Classification of Human Activities on Videos Using Deep Learning

…..

….

Abstract: With the number of millions of surveillance cameras, the need for something like these systems increase every day. As a result, the number of videos increases on social platforms. This massive number of videos call for systems able to classify. In this study, a system has been developed for classifying human activities from videos. CNN and LSTM models have been used in the developed system. Experimental studies on 20000 images of ten user activities show that CNN is more successful than LSTM. We hope that the benchmarks and conclusions from this study can offer assistance researchers in related fields to rapidly set up a great premise for encourage investigations along this exceptionally empowering course.

Keywords: Human activity classification, Deep learning, CNN, LSTM

Reference to this paper should be made as follows: Gençaslan, S., Utku, A. and Akcayol, M.A. (2020) 'Classification of Human Activities on Videos Using Deep Learning', *International Journal of Computational Science and Engineering*, Vol. X, No. Y4, pp.000–000.

1. Introduction

With the popularity of personally cameras, phones, survelliance cameras and content sharing social platforms, there is an urgent need for technologies that can naturally analyze and classify video data. Video classification is the primary step in the process of analyzing video content. Unnecessary to say, the job of a traditional security operator is to observe some extended video streams to detect, classify, categorize and recognize human activities[1].

In recent years, deep learning based models have been demonstrated to be more competitive than the traditional methods for fathoming and solving complex problems. Moreover, Convolutional Neural Networks (CNN) have exhibited incredible success on different tasks, particularly image classification, image based object localization, speech recognition, etc. Moreover, Long Short Term Memory (LSTM) is appropriate to classify, process and predict time series.

In the continuation of this section, the studies in the literature on video classification has been investigated. Video classification is a vital research topic in multimedia and computer vision. Successful classification systems rely on the extracted video features heavily.

In the most studies, video classification has been extensively researched and analyzed in detail such as [13, 3, 4]. Most of these examples are about recognizing and classification activities and objects detection in the videos. For example, Ma has detected that LSTM for detecting actions is better in terms of the performance. Furthermore LSTM, LSTM-s and LSTM-m have been used [13]. In addition, Canotiho has handled a problem about recognition of some action, such as walking, running, fighting etc [1]. Especially recognition of the common human actions is harder because human has a complex structure. About this topic, Kim has used The Hidden Markov Model (HMM) for recognition basic actions and used Conditional Random Field (CRF) for more complex actions [2].

Many related works have been investigated recognition and classification of human actions/activities. For instance, Anguita has developed a system that recognize 30 fundamental human activities such as walking, going up stairs, going down stairs, sitting, standing, laying etc. by using Support Vector Machines (SVM) with the data that taken from sensors of smartphones [3]. Some related works have used Category Feature Vectors (CFV). For example, Lin has used it and described each activity as a combination of Gaussian Mixture Models (GMM). Consequently,

the model has gained more flexible for unusual actions [4]. In some related works, the localization of human has been used. For instance, Dai has used Temporal Context Network (TCN) which is structurally similar to Faster-RCNN. Consequently, some classification has been done and LSTM which is a method that is effective for classification on videos has been used [5].

In the literature, because LSTM and CNN algorithms have more effective results, they are used commonly for video classification applications. Furthermore, those algorithms have been observed many various accuracy results between %30 and %96 in related works.

In this study, these two algorithms have been used and compared each other and the rest of this paper has been sorted out as follows. Section 2 describes the proposed approach in the literature in detail. Experimental results, examinations and comparisons have been discussed in Section 3, followed by conclusions in Section 4.

2. METHODOLOGY

In this section, deep learning and machine learning based algorithms have been investigated. Decision Trees (DT), SVM, Recurrent Neural Networks (RNN) and especially CNN and LSTM. Additionally, the key components of the proposed algorithms for our study such as CNN and LSTM have been described in detail.

2.1. Deep Learning Based Algorithms

CNN is one of the variants of the neural network widely used in the field of computer vision. In the deep learning, CNN is a type of deep neural network mainly used to search and analyze images [13]. It has applications in image and video recognition, recommendation systems [14], image classification, medical image analysis, natural language processing [15] and financial time series [16].

It gets its name from the kind of hidden layers that is comprises of. For the most part, those hidden layers comprise of convolutional layers, pooling layers, fully connected layers, non-linearity layers, flattening layers and normalization layers. Fig. 1 shows the structure of CNN basically.

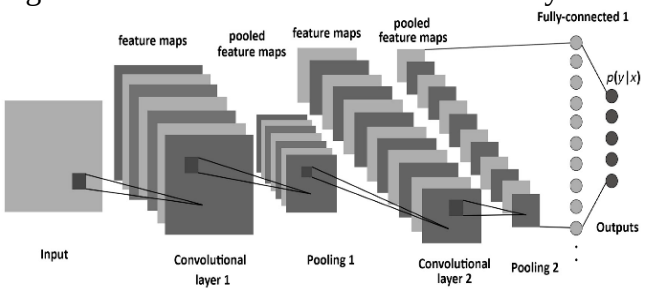

Figure 1. The structure of CNN

Mathematically, convolution of two functions f and g is defined as Eqn. 1.

$$(f * g)(i) = \sum_{j=1}^{m} g(j) f(i - j + \frac{m}{2}) \qquad (1)$$

Another of the deep learning based algorithms is RNN. RNN is a structure which produces output by applying a number of mathematical operations to the information coming to the neurons on the layers. RNN are a class of neural networks in which connections between units form a directed loop. When describing the RNN model, it is necessary to consider the directed loop statement which is mentioned in the previous sentence. For that statement, it can be said as a structure that works forward. This is referred to as feedforward in the literature. Feedforward passes the input data through the network and an output value is acquired. An error is obtained by comparing the obtained output values with the correct values. The weight values on the network are changed depending on the error and in this way, a model that can output the most accurate result is created. Fig. 2 shows the structure of a feedforward neural network.

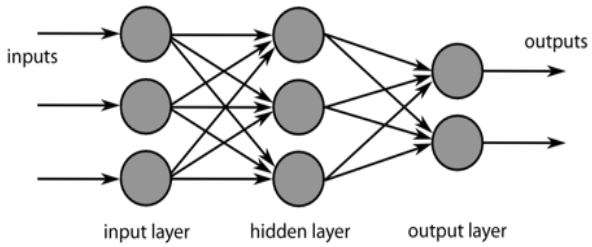

Figure 2. The structure of feedforward

In the training of a feedforward network, the error should be sufficiently reduced. Thus, a structure that will output appropriate to the input is created by regenerating the weights going to the neurons [17].

In the structure of RNN, the result is drawn not only in the current input but also in other inputs. In addition to the input data at time *t* in RNN, results of the hidden layer from the moment *t*-1 are the input of the hidden layer at time *t*. The decision made for the input at *t*-1 affects the decision in which will make at time *t*. In other words, the inputs in these networks combine current and previous information to obtain an output.

Recurrent structures are different from the process of feedforward structures because they use their output as an input in subsequent processes. Besides, it can be said that recurrent networks have a memory. The reason for adding memory to the network is that a series of inputs arranged in a certain order make sense for the output. Feedforward networks are not adequate for such data sets. At this point, RNN comes into play. Recurrent networks are utilized to get it the structure of sequential data, for example using the outputs from the previous transactions.

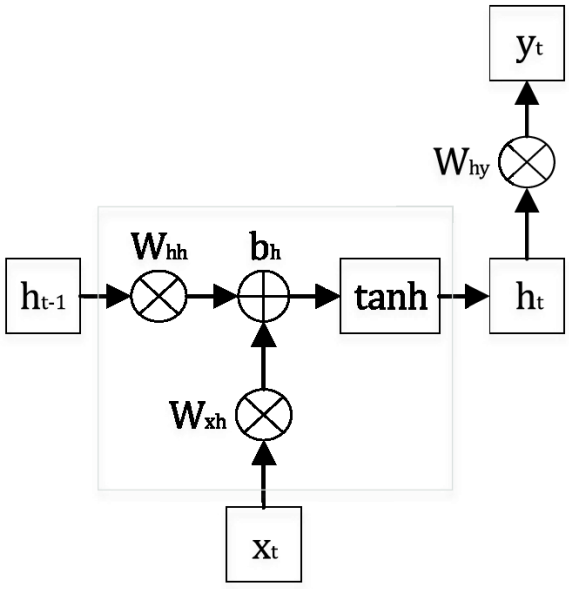

Figure 3. The process of RNN

As seen in Fig. 3, the result from the hidden layer in the process loop of RNN both produces output and written to the content units. In this way, each new input is processed with content units produced as a result of processing previous inputs. If there is a correlation between the data which is memorized at different times, this is called long term dependency.

RNN is a network that can calculate the relationship between these long term dependencies. While doing this calculation, the following mathematical formula is used.

$$h_t = \varnothing(W_{x_t} + U_{h_{t-1}}) \qquad (2)$$

h_t is the result of the hidden layer at *t*. The X_t input is multiplied by the *W* weight. At time *t*-1, h_{t-1} value kept in content unit is multiplied by *U* weight and summed with W_{X_t}. *W* and *U* values are the weights between the input and the layer. Here, the weight matrix takes values according to which of the previous and current data has more or less effect on the result.

The result of the error from these operations is calculated and new weight values are rearranged with backpropagation. The process of backprop continues until the error is sufficiently minimized. Sum of $W_{X_t} + U_{h_{t-1}}$ is put into activation function like sigmoid, tanh. Thus, very large or very small values are placed in a logical range. In this way, non-linearity is also achieved.

Our last deep learning based algorithm is LSTM. This is a unique type of RNN that can understand and learn long-term conditions. All recurrent neural networks pass through a chain of repeating neural network modules. In the standard RNN, this repeating module has a straightforward structure like a single tanh layer.

The difference between the repeating module in the LSTM structure is that instead of a single neural network layer, there are 4 specially associated layers. These layers are also called gates. It can be a structure that gets data exterior the typical stream. This data can be put away, composed to the cell or examined. The cell chooses what to store, when it will permit it to examine, compose or delete. These gates have a network structure and activation function. Similar to a neuron, it transmits or stops incoming information according to its weight and also these weights are calculated by learning a recursive network. Thanks to this structure, the cell learns whether to receive the data, leave it or delete it. Fig. 4 shows the structure of LSTM basically.

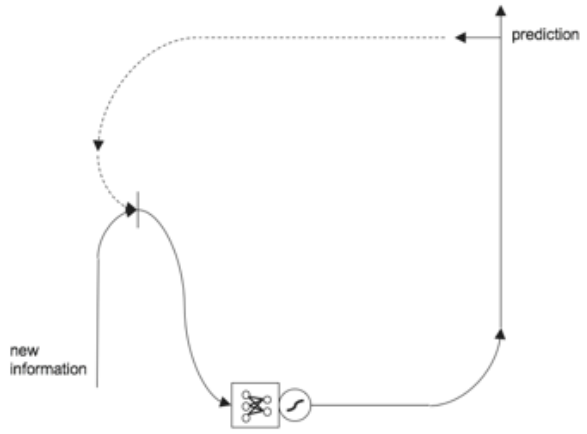

Figure 4. Basic LSTM representation

The memory unit stores them for use in long-time data. The + sign in the diagram shows the addition process on an element basis. The x sign is the product on the basis of elements. With which multiplication is done on an element basis, how much of the data is used and how much is used is calculated by multiplying the data in memory by weights. Then an estimate is generated by summing from memory and probability with + operation. Fig. 5 shows the memory unit of LSTM.

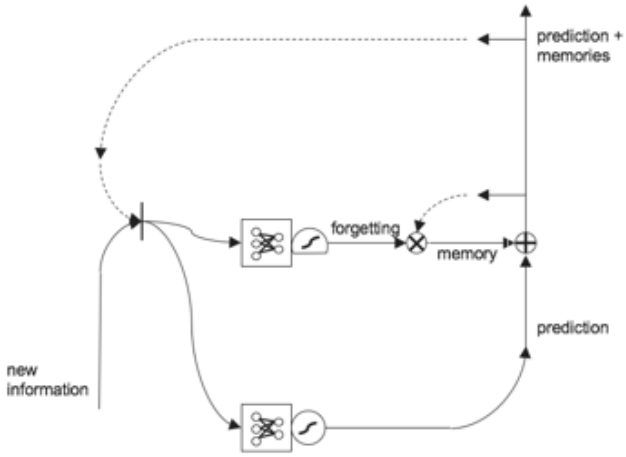

Figure 5. LSTM memory unit

Fig. 6 shows the LSTM selection unit and a new door has been added. This is a filter section to choose which one to use and how much after gate picking. In this section, keeping the data in memory and separating it from the guesswork is done. This door also has a unique neural network.

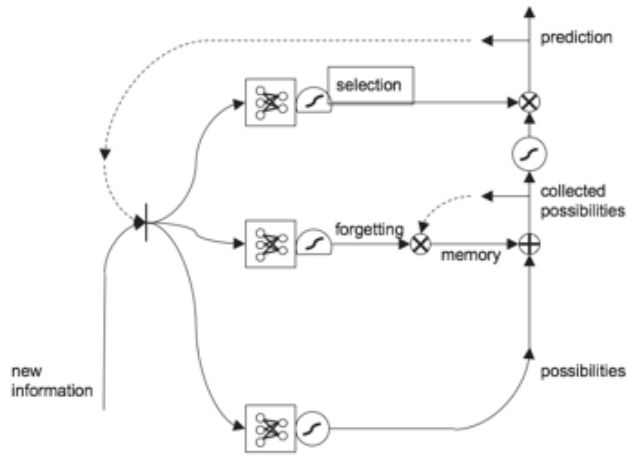

Figure 6. LSTM selection unit

Another door is used for filtering before collecting the first possibilities from memory. there is also a network structure. The incoming results are multiplied on an element basis and proceed to the other process. Thus, the data that does not have to go into memory are filtered. Fig. 7 shows this ignoring unit of LSTM.

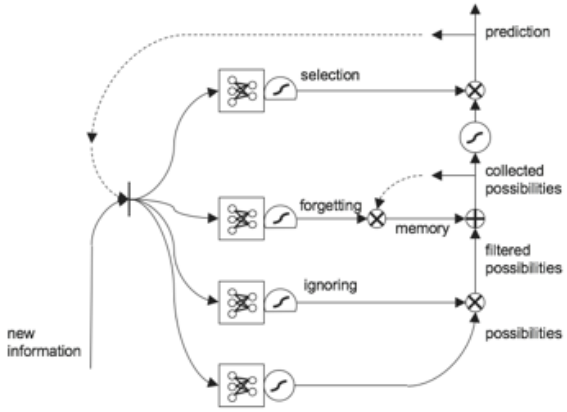

Figure 7. LSTM ignoring unit

2.2. Machine Learning Based Algorithms

Tree-based learning algorithms are one of the most widely used supervised learning algorithms. Generally, these algorithms can be adjusted to solve all classification and regression problems. DT is one of the data mining classification algorithms. They have a predefined target variable. Structurally, they put forward a top down approach. DT may be a structure for applying a set of decision rules to divide a dataset containing a large nulmber of records into smaller sets. In other words, it is a structure that can be used by applying simple decision steps and

dividing a large number of large records into very small recording groups.

DT have many advantages such as being easy to understand and interpreting, processing both numerical and categorical data, and handling multi-output problems. On the other hand, there are disadvantages such as being able to produce extremely complex trees that cannot explain the data well and experiencing an over fit situation.

Entropy is a measure of vulnerability about our data. Instinctively, it can be thought that a data set has a lower entropy if it has only one label. So, our data needs to divide in a way that minimizes entropy. The better the splits, the better our forecast.

$$H = -p(x)\log p(x) \qquad (3)$$

Here, $p(x)$ denotes the percentage of the group belonging to a particular class and H denotes the entropy. it can be said that DT wants to make splits that minimize the entropy value. Knowledge gains are used to determine the best division. Information gain is calculated with the following equation:

$$Gain(S,D) = H(S) - \sum \frac{|V|}{|S|} H(V) \qquad (4)$$

Here, S is the original dataset, and D is a split part of the set. Each V is a subset of S. V is all discrete and forms S. In this case, knowledge gain is defined as the difference between the entropy of the original data set before the split and the entropy value of each attribute.

SVM is fundamentally used for separating the two categories of data in the best way. For this, the boundary or in other words the hyper plane in determined. SVM is utilized in numerous classification problems from facial recognition systems to speech classification, investigation and analysis problems. SVM has many advantages such as being successful in high-dimensional spaces, being effective in cases where the measure number is more than the number of test samples. SVM is divided into two parts depending on whether the data set can be separated linearly or not.

When using SVM for classification, it can be assumed that the instances of the two categories are distributed linearly. In this case, the separation of the two types is specified in the linear SVM with the help of the decision function obtained from the training data. This is the decision line that divides the data set into two parts. While infinite decision lines can be drawn, the key is to identify the ideal decision line. Fig. 8 shows the support vectors.

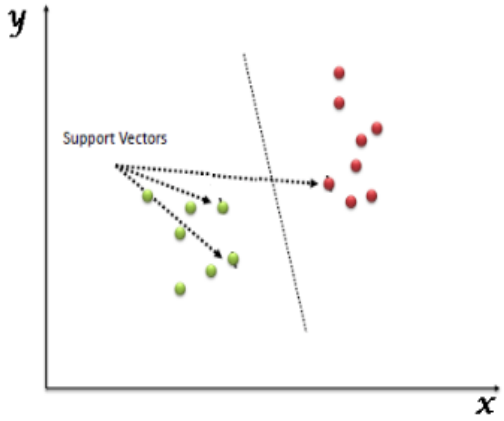

Figure 8. Linear SVM

In a nonlinear dataset, SVM cannot draw a linear hyper-plane. Consequently, kernel tricks called kernel numbers are utilized. Center strategy extraordinarily builds machine learning in nonlinear data. Fig. 9 shows the nonlinear SVM.

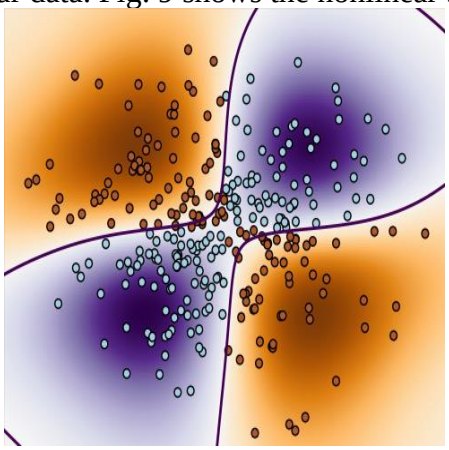

Figure 9. Nonlinear SVM

3. PROPOSED APPROACH

In this section, the proposed approaches have been tested and the results have been reported on one of the popular datasets for action recognition and classification.

3.1. Dataset

The release of the Kinetics dataset [6] in 2017 prompted stamped enhancements in best in class performance on an assortment of action recognition datasets: UCF-101 [7], HMDB-51 [8], Charades [9], AVA [10], Thumos [11]. The Kinetics 600 dataset is a multi-class dataset and consists of 5 columns such as label, youtube_id, time_start, time_end, split. In total, the dataset consists of approximately 500000 videos with 600 different human action labels. The dataset has 5 columns such as the dataset is open-domain and covers a wide range of lots of different topics including sports, animals, music, basic activities and complex activities.

10 different classes have been selected including fundamental human activities such as {Brushing Teeth, Dining, Laughing, Reading Newspaper, Singing, Sleeping, Swimming Backstroke, Waking Up, Washing Hair, Writing} and split the dataset into a training set of 335 videos and a test set of 165 videos. Consequently, 500 videos of different lengths have been obtained.

These 500 videos of different lengths have been cut in 10 seconds each and set the videos which have been chosen a fixed frame rate of 4 FPS with a spatial resolution of 128x128 pixels. In totally, 20000 frames have been obtained from 500 videos. These 20000 frames have been splitted into a training set of 13334 frames and a test of 6666 frames. Fig.10 shows the size of our categories for this study.

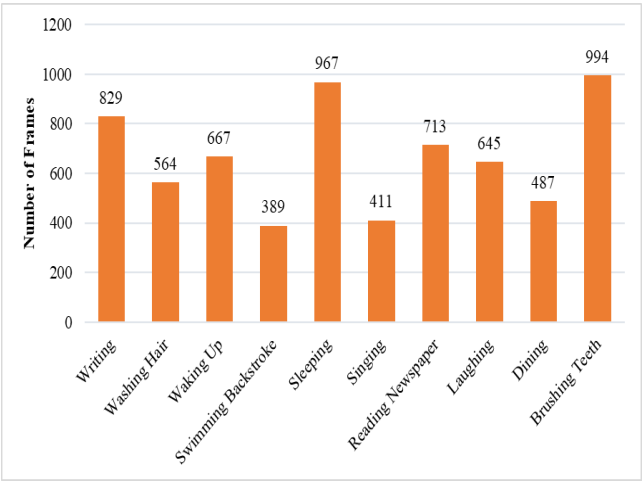

Figure 10. Number of frames in each category

3.2. Developed Model

The CNN model has 3 convolutional layers with 32-64-128 units which its kernel_size is 3, padding is “same” and activation function is ReLU, 3 pooling layers, 4 dropouts, 1 flattening layer and 2 fully connected layers. Additionally, Adam has been used as an optimizer.

The LSTM model 2 LSTM layers with 64 hidden units, 2 dropouts and 2 fully connected layers. Table 1 shows more detail about CNN and LSTM model.

Table 1. Configuration of two networks

	CNN	LSTM
Convolutional Layers	128x128x32 (stride:1, padding:1) x2 pooling dropout 64x64x64 (stride:1, padding:1) x2 pooling dropout 32x32x128 (stride:1, padding:1) x2 pooling dropout flatten	None,16384,64 dropout None,64 dropout None,32 dropout None,10
Layer	2,048 neurons	1,024 neurons
FC	softmax	softmax

The performance of our proposed methods has been reported using test set. Two different variations have been tested for video classification on Kinetics and the results have been presented in Fig. 11 and Table 2.

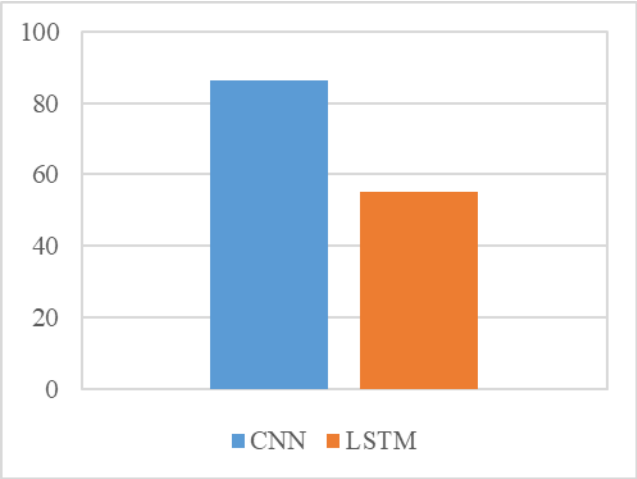

Figure 11. Accuracy of proposed methods (%)

Table 2. Performance of proposed methods

Model	Kinetics
CNN	86.5%
LSTM	55%

As seen in Table 2 and Figure 11, CNN has 86.5% success rate and LSTM has 55% accuracy rate. These accuracy rates show that CNN is more successful than LSTM for the Kinetics dataset.

3.3. Experimental Results

K fold cross validation which is a type of cross validation have been tried in this work. Furthermore, 3-5-7 fold cross validation have been attempted and the best result has been gained from 5 fold cross validation. The result of that process is 83.55%. Table 3 shows the result of each fold, sum of those and our result accuracy of cross 3-5-7 fold cross validation.

Table 3. The results of 3-5-7 fold cross validation

	3 Fold CV	5 Fold CV	7 Fold CV
Number of Train/Test Sample	Train= 13334 Test= 6666	Train= 16000 Test= 4000	Train= 17143 Test= 2857
Fold 1	0.479676	0.46975	0.440167
Fold 2	0.8159	0.9215	0.98949
Fold 3	0.91314	0.93	0.96079
Fold 4	-	0.917	0.89289
Fold 5	-	0.9395	0.91774
Fold 6	-	-	0.65593
Fold 7	-	-	0.94097
Sum	2.20877	4.17775	5.80313
Accuracy	0.73625	0.83554	0.82901
	73.62%	83.55%	82.9%

As in the Table 3, the last results of 3-fold, 5-fold and 7-fold cross validation have been obtained such as 73.62%, 83.55%, 82.9%. After these results, 5-fold cross validation has been chosen for this study. According to 5-fold cross validation (test sample=4000), Fig. 12 shows the number of sample which is true/false classified.

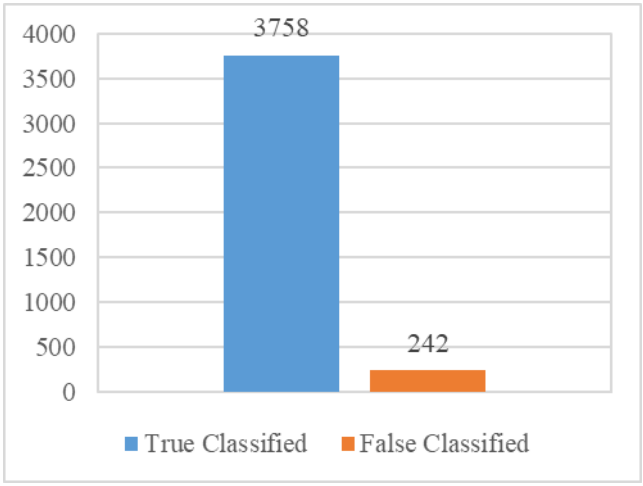

Figure 12. True/False classified sample numbers

Various methods can be used to measure classification performances and in this study, a confusion matrix has been created for test results and Table 4 shows the confusion matrix. According to the confusion matrix, the values of True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN) have been calculated. Accuracy, Precision, Recall and F1-Score values have been calculated to measure the performance of the classification. These values are important in measuring the classification performance and can be calculated with the formulas given as Eqn. 5, 6, 7 and 8.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

$$F1Score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (8)$$

After the confusion matrix has been created, the accuracy values have been calculated for each action class according to the formula in Eqn. 6 and has been given in Fig. 13.

Table 4. Confusion matrix which has been obtained in test set

	Brushing Teeth	Dining	Laughing	Reading Newspaper	Singing	Sleeping	Swimming Backstroke	Waking Up	Washing Hair	Writing
Brushing Teeth	987	0	0	0	0	28	0	4	2	19
Dining	0	487	0	2	7	24	0	0	0	40
Laughing	5	0	641	0	1	137	0	14	0	28
Reading Newspaper	2	0	0	711	0	5	0	2	0	0
Singing	0	0	0	0	400	40	0	0	0	0
Sleeping	0	0	0	0	0	560	0	0	0	0
Swimming Backstroke	0	0	0	0	0	11	389	0	0	0
Waking Up	0	0	0	0	0	31	0	638	0	11
Washing Hair	0	0	4	0	3	51	0	1	562	19
Writing	0	0	0	0	0	80	0	8	0	712

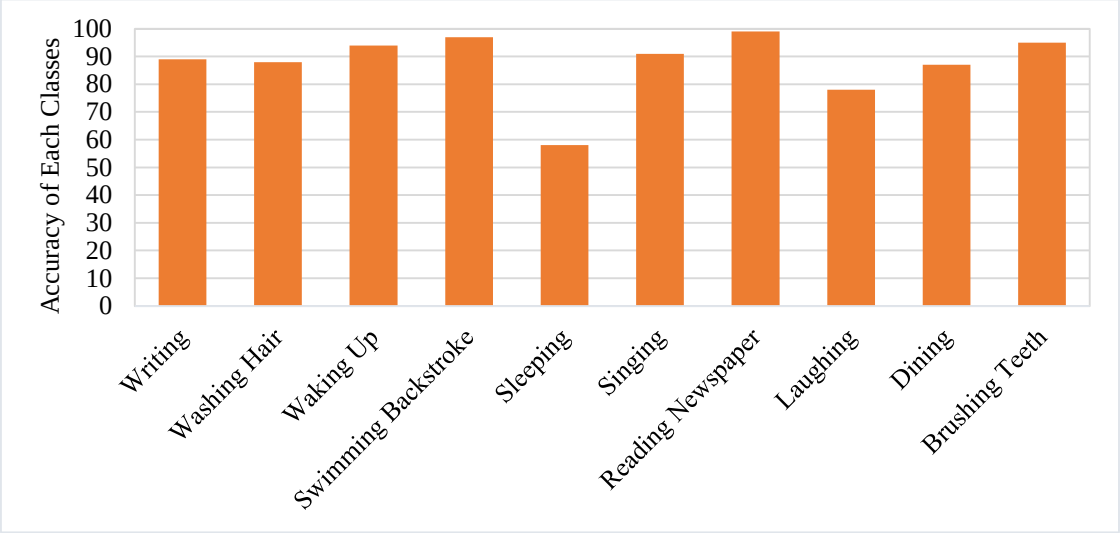

Figure 13. Accuracy of each classes (%)

After the confusion matrix has been created, the average Precision, Recall and F1-Score values have been given in Table 5.

Table 5. The results of precision, recall and f-measure

	Precision	Recall	F1-Score
Result	93%	92%	92%

As in the Table 5, the results of Precision, Recall and F-Measure have been gained as 93%, 92%, 92%, respectively.

According to confusion matrix, the values of True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN) have been calculated. Fig. 14 shows the number of frames which are true/false classified according to confusion matrix.

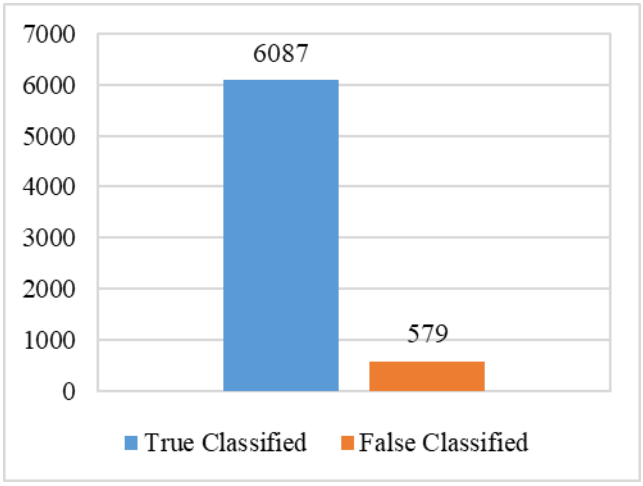

Figure 14. The number of true/false classified

As in the Fig. 14, the result of the number of true classified samples has been calculated by summing the values of True Positive and True Negative. Similarly, result of the number of false classified samples has been calculated by summing the values of False Positive and False Negative.

4. CONCLUSIONS

Image and video classification processes are becoming widespread in the IT sector and are very important. Classification can be done in many subjects such as catching criminals, fast and unusual movements of vehicles, abnormal movements of people, line violations in pedestrian crossings of vehicles. In this image and video classification process, deep learning, and therefore artificial neural networks (ANN), play an important role. Deep learning methods can give very successful results in image processing applications, as it can make calculations used in machine learning in numerous layers immediately and that it can make evaluations with better parameters, finding even the parameters that we need to define in machine learning.

In this study, CNN and LSTM have been used for image classification and have been tried on many models. The most appropriate parameters have been determined with the trials. First of all, LSTM models has been created and the best success has been measured as 55% and determined that this success has been insufficient. After this determination, CNN models has been tried. Most of these models have been in over fit condition. In the CNN model, which has been last tried, the result of 3-5-7-fold cross validation procedures for 5 folds is the most successful result and the accuracy value of the model has been measured as 83.55%.

5. REFERENCES

1. Ribeiro, P. C., Santos-Victor, J. and Lisboa, P. (2005). 'Human activity recognition from video: modeling, feature selection and classification architecture'. *Proceedings of International Workshop on Human Activity Recognition and Modelling.* Oxford, UK. pp.61-78.

2. Kim, E., Helal, S. and Cook, D. (2009). 'Human activity recognition and pattern discovery'. *IEEE pervasive computing*, Vol. 9 No.1, pp.48-53.

3. Anguita, D., Ghio, A., Oneto, L., Parra, X. and Reyes-Ortiz, J. L. (2013). 'A public domain dataset for human activity recognition using smartphones'. *Esann*, Vol. 3, pp. 3.

4. Lin, W., Sun, M. T., Poovandran, R. and Zhang, Z. (2008). 'Human activity recognition for video surveillance'. *2008 IEEE International Symposium on Circuits and Systems*, Washington, USA, pp. 2737-2740.

5. Dai, X., Singh, B., Zhang, G., Davis, L. S. and Qiu Chen, Y. (2017). 'Temporal context network for activity localization in videos'. *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, pp. 5793-5802.

6. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S. and Suleyman, M. (2017). 'The kinetics human action video dataset'. *arXiv preprint arXiv:1705.06950*.

7. Soomro, K., Zamir, A. R. and Shah, M. (2012). 'UCF101: A dataset of 101 human actions classes from videos in the wild'. *arXiv preprint arXiv:1212.0402*.

8. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T. and Serre, T. (2011). 'HMDB: a large video database for human motion recognition'. *2011*

International Conference on Computer Vision, Barcelona, Spain, pp. 2556-2563.

9. Sigurdsson, G. A., Varol, G., Wang, X., Farhadi, A., Laptev, I. and Gupta, A. (2016). 'Hollywood in homes: Crowdsourcing data collection for activity understanding'. *European Conference on Computer Vision*. Amsterdam, Holland, pp. 510-526.

10. Gu, C., Sun, C., Ross, D. A., Vondrick, C., Pantofaru, C., Li, Y. and Schmid, C. (2018). 'Ava: A video dataset of spatio-temporally localized atomic visual actions'. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Utah, USA, pp. 6047-6056.

11. Idrees, H., Zamir, A. R., Jiang, Y. G., Gorban, A., Laptev, I., Sukthankar, R. and Shah, M. (2017). 'The THUMOS challenge on action recognition for videos in the wild'. *Computer Vision and Image Understanding*, Vol. 155, pp. 1-23.

12. Ma, S., Sigal, L. and Sclaroff, S. (2016). 'Learning activity progression in lstms for activity detection and early detection'. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* Las Vegas, NV, USA, pp. 1942-1950.

13. Valueva, M. V., Nagornov, N. N., Lyakhov, P. A., Valuev, G. V. and Chervyakov, N. I. (2020). 'Application of the residue number system to reduce hardware costs of the convolutional neural network implementation'. *Mathematics and Computers in Simulation*.

14. Van den Oord, A., Dieleman, S. and Schrauwen, B. (2013). 'Deep content-based music recommendation'. *Advances in neural information processing systems.* pp. 2643-2651).

15. Collobert, R. and Weston, J. (2008). 'A unified architecture for natural language processing: Deep neural networks with multitask learning'. *Proceedings of the 25th international conference on Machine learning*. Helsinki Finland, pp. 160-167.

16. Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M. and Iosifidis, A. (2017). 'Forecasting stock prices from the limit order book using convolutional neural networks'. *2017 IEEE 19th Conference on Business Informatics (CBI)*. Thessaloniki, Greece, Vol. 1, pp. 7-12.

17. Zaremba, W., Sutskever, I., and Vinyals, O. (2014). Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.