An Efficient Classifier using Machine Learning Technique for Individual Action Identification

G.L.Sravanthi[1], M.Vasumathi Devi[2], K.Satya Sandeep[3], A.Naresh[4], A.Peda Gopi[5]
Vignan's Nirula Institute of Technology & Science for Women, Peda Palakaluru, Guntur-522009, Andhra Pradesh

***Abstract*—Human action recognition is an important branch of computer vision and is getting increasing attention from researchers. It has been applied in many areas including surveillance, healthcare, sports and computer games. This proposed work focuses on designing a human action recognition system for a human interaction dataset. Literature research is conducted to determine suitable algorithms for action recognition. In this proposed work, three machine learning models are implemented as the classifiers for human actions. An image processing method and a projection-based feature extraction algorithm are presented to generate training examples for the classifier. The action recognition task is divided into two parts: 4-class human posture recognition and 5-class human motion recognition. Classifiers are trained to classify input data into one of the posture or motion classes. Performance evaluations of the classifiers are carried out to assess validation accuracy and test accuracy for action recognition. The architecture designs for the centralized and distributed recognition systems are presented. Later these designed architectures are simulated on the sensor network to evaluate feasibility and recognition performance. Overall, the designed classifiers show a promising performance for action recognition.**

***Keywords*—Human action recognition; machine learning; neural networks**

I. Introduction

Human action recognition has a variety of applications which require automated recognition of human behaviors. By using motion recognition, movements of a person in image frames can be detected and regenerated into a 3D model, and this can be useful for sports experts to analyze the performance of athletes [3]. Besides, it can be implemented in surveillance system which can automatically monitor human presences and behaviors at public areas like shopping mall, train station and airport to detect abnormal or suspicious activities. The capability to understand the meaning of human interactions by machines enables the development of advanced human-computer interfaces which can be used for computer games or designs. Apart from that, analysis of human activities can also be used in video conferencing, healthcare monitoring, and virtual reality.

Much research has been done in the field of human action recognition. Before 2000, researchers paid more attention to body tracking and posture recognition. W. Kay et al. [1] proposed a model-based approach by which stick figures are obtained from ordinary images to derive a model of the lower limbs. C. Gu et al. [2] introduced an approach to estimate human postures by infrared images in real-time. Space-time methods have been widely used which can model 3D volumes of human actions by combining 2D images in a space-time dimension. R. Goyal et al. [3] introduced a spatio-temporal approach which can recognize a wide range of actions by using over-segmented videos with correlation techniques.

To recognize human activities from image sequences, state-space [7, 8] and template matching [9, 10] are also applied by many researchers. Traditional approaches usually need a series of specifically designed algorithms to preprocess the images and extract feature sets [11]. Designing of such complicated algorithms can be quite time consuming. Besides, these manually designed algorithms cannot be easily reproduced for new use cases. Compared with traditional approaches, machine learning methods do not need complex preprocessing algorithms and only depend on good training datasets. Apart from that, when more patterns need to be considered, modification of the classifier can be easily applied on machine learning methods [12]. This is why traditional approaches are gradually replaced by machine learning methods.

The purpose of our project is to design human action recognition.

In order to produce dynamic visuals to interact with the human, the installation operates a visualization program which simply loads the depth images, renders them by visual lights and displays them on the screen in order [13]. This program works well and enables simple interactions between human and the installation. However, it can only conduct binary classifications on presence of human or absence of human. The natural elements are depicted in Fig. 1. To analyze human actions in the depth images, a better solution needs to be found.

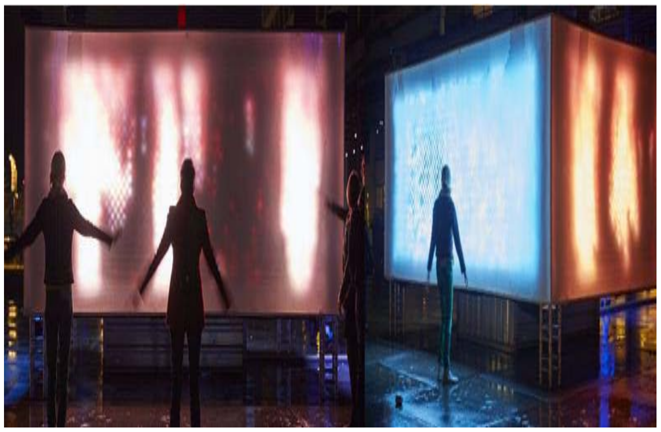


Fig 1. Natural Elements.

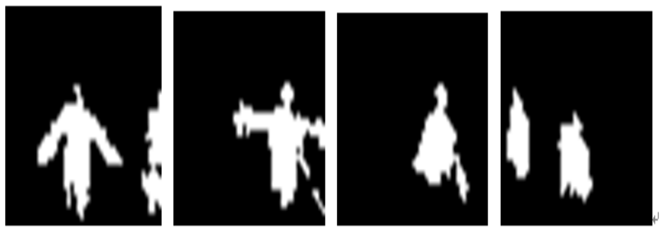


Fig 2. Depth Images.

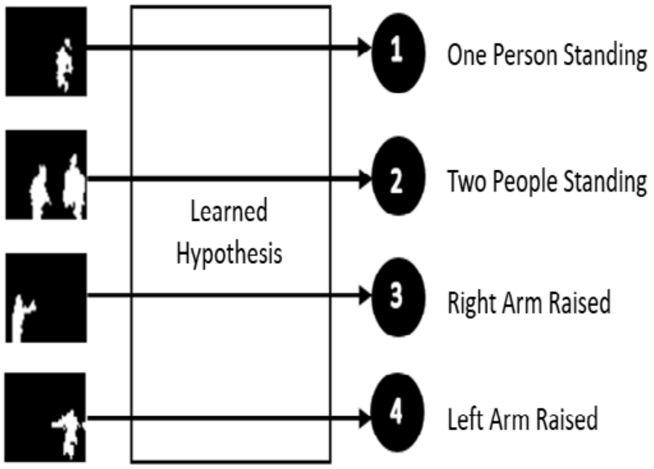


Fig 3. A Example for Mapping Input Data to Predetermined Posture Classes.

The depth images are depicted in Fig. 2 that represents natural images. As an example, we use four single images as inputs of the classifier. The classifier uses a learned "hypothesis work" for mapping inputs to predetermined classes. Fig. 3 shows an example for mapping input data to predefined posture classes.

To be specific, the classifier implements a mathematical model to read input data, transform the data, and compare the result with standard values to determine which action it belongs to. As the original data is not labeled, it cannot be directly used as input to the classifier in the learning stage. Before recognition, the original data needs to be visualized to investigate what kinds of human actions can be found. The recognizable human actions need to be classified and each action is noted by a unique label. Then the images are processed by algorithms and manually labeled to generate training examples for the classifier.

A. Machine Learning Models

Machine learning [14] has been widely used in spam detection, speech recognition, robot control, object recognition and many other domains. It is a type of artificial intelligence technology which enables computers to learn without being explicitly programmed [15]. Machine learning can be considered as the development of a system that can process a large volume of data, extract meaningful and useful information and exploit such information in practical problems [16]. Machine learning has two common learning styles: supervised learning and unsupervised learning.

Supervised learning [17] is usually used for regression, classification and ranking. In case of classification, the task of supervised learning is to establishing a relationship function from training data which consists of labeled training examples to their respective labels. Each training example contains input data and a corresponding labeled output value. The output of the relationship function is a logical value used for classification. The trained function should be able to predict the correct value of any input data. By analyzing each pair of training examples, the relationship function is produced and adjusted step by step until the underlying relationship between inputs and outputs can be appropriately expressed by the function. Common approaches for supervised learning includes logistic regression, Bayesian statistics and artificial neural networks [16].

Unsupervised learning [18] can be used for clustering, anomaly detection and dimensionality reduction. Unlike supervised learning, the task of this machine learning method is discovering the hidden structure of unlabeled training data [19]. As the training data is unlabeled, unsupervised learning does not have such a reward system to evaluate predicted output values [20]. It mainly focuses on exploring hidden patterns or intrinsic structure of training data [21]. The intrinsic structure can be used to organize these unlabeled data into similarity groups, which are also called clusters. K-means, hierarchical clustering and hidden Markov models are common cluster algorithms for unsupervised learning [22] [23].

The goal of this proposed work is to develop a human action recognition system based on the interaction dataset [24]. The algorithm used for this system should be able to recognize each human action which needs to be predefined before training. Thus, compared with unsupervised learning, supervised learning which can use labeled training set for classification tasks is the optimal solution for designing such a recognition system [25].

Fig. 4 shows the workflow of supervised learning methods. For classification tasks, the classifier is developed by three phases: training, validation and performance test. Feature extraction is the first step in the training phase. Raw data are preprocessed by feature extraction algorithms to get the feature matrix. Then the correct outputs are created for the derived feature data.

After that, feature data together with correct outputs are randomly selected and separated into three datasets: training set, validation set and test set. The training set is fed to the model iteratively to train the parameters of the model [26]. The validation set is used to determine when to stop the training by estimating the performance of the model during the training process. The test set is a set of examples that never take part in the training process, which means that is totally new data for the trained model and is finally used to evaluate the quality and the generalizability of the trained model by calculating the prediction accuracy on it. The work flow of the supervised learning is depicted in Fig. 4.

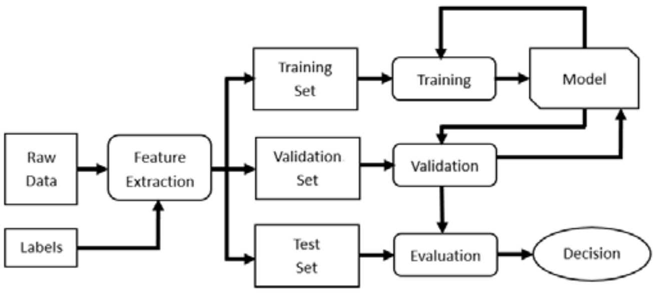


Fig 4. Workflow for Supervised Learning.

According to the literature study, some supervised learning methods are selected for our classifier. The reason why these methods are suitable for our project is discussed as follows.

Logistic Regression (LR) is one of the most commonly used method for applied statistics and discrete data analysis and often works surprisingly well as a classifier [27]. When the target outcome has only two types, it can be called a binary classification problem [28]. For binary classification, this algorithm learns from the relationship between the target outcome and a given set of predictors by estimating probabilities using the logistic function [29]. When the target outcome has more than two types, it is called a multiclass problem. Multinomial Logistic Regression (MLR) is a classification method which generalizes logistic regression to multiclass problems. We implement the MLR model as the classifier for human action recognition tasks [30].

II. Literature Survey

H. Zhao, et al. [4] present Motion History Image (MHI) to represent human motion. An MHI is a kind of temporal template which can be created by recording the motion in an image sequence. Human postures in the image sequence are accumulated in a way that records the corresponding motion history. MHI can be used in our project to extract motion features from humans. Aaron F. Bobick, et al. use a matching based method to recognize predefined motions by comparing new templates with labeled MHI instances. This matching approach performs well because of the high quality of motion examples.

Training examples for different motions are designed and created by the algorithm developers as reference [31]. However, we want to make the classification algorithms suitable for training and test datasets that are randomly collected, with unpredictable variations for each kind of motion [32]. Our algorithms will classify human actions by learning from data without the need of standard templates created by experts.

M. Monfort et al. [5] apply ANN and Bayesian framework for human action representation and classification on a multi-camera setup. Action recognition is achieved by using Multilayer Perceptron (MLP) which is a feed forward neural network model. The ability of this ANN based method to correctly classify human actions is shown by the experiments on multi-view database in which the highest recognition rate is 94.87%. The proposed method demonstrates the capacity of ANN in human action representation and recognition and shows the effectiveness of ANN model in challenging experimentations. Thus, we envision that ANN is a good candidate classifier for human action recognition.

S. Lai et al. [6] present a vision-based technique for static hand gesture recognition. Multilayer neural networks with back-propagation algorithms are used to recognize gesture features into predefined classes. The neural network based method performs well in the testing experiment and reaches sufficient recognition accuracy. What we can learn from this paper is how they analyze the problems and determine proper techniques for each step of recognition process. They divide the process into four steps: data gathering, data processing, feature extraction and classification. This type of work flow can be applied in the development of algorithms. Apart from that, this paper gives a detailed introduction of how to train the multilayer neural networks and how to design the experiments to test the classifier.

K. Tang et al. [7] propose a real-time human action recognition system based on fusion features of depth images and inertial signals. The system is trained by a public human action dataset and evaluated for real-time and offline recognition performance. The recognition accuracy is more than 97% which demonstrates the effectiveness of the system. This paper gives us a good example of data analysis in which bar charts, tables and confusion matrix are used to analyze different aspects of recognition performance.

J. Wu et al. [9] conduct investigation on different types of distributed neural networks in terms of communication cost and memory usage. They propose centralized, horizontal and vertical decomposition approaches for distributing neural networks in a Wireless Sensor Network (WSN). Compared with vertical decomposition, horizontal decomposition gets a more promising result for communication costs. This article gives a good example of how neural network can be modified to a distributed structure.

III. Proposed Method

According to the requirement of training examples, original data should be processed to fit the format of input layer of the classifier. For human posture recognition, still images are used as posture representation. For human motion recognition, a motion history image algorithm is applied to generate a motion representation from successive posture images. Pixel values of generated depth images can be directly used as features to train on machine learning models [33]. The classifier based on pixel features performs well on Intel Core i5 platform but it is not applicable on embedded platform which is limited on computational workload and memory capacity. Thus, it needs feature extraction methods to extract higher level human action features from depth images [34].

A projection-based feature extractor is presented to generate smaller feature matrix for the classifier used on embedded platforms. Machine learning models are built on TensorFlow platform. MLR is used for action recognition tasks followed by the SGD-MLR as a contrast method. A multi-layer ANN is also implemented as the classifier which uses flexible configuration of parameters to optimize recognition performance. This ANN model is then modified to a distributed structure which can be deployed on sensor networks.

A posture separation method is designed to detect posture of a single person in the image, separate it out and resize the image to a predefined size. The resized posture images are used as training examples for the human posture recognition. Pixel values are extracted as features of human postures. The feature matrix created by all feature vectors of training examples will be used as the training dataset for machine learning models. Since large matrix computation is hard to implement on embedded devices, a projection-based feature extraction method is used to extract high level features from pixel features. By using this feature extraction method, the size of the motion feature vector shrinks from 1728 to 84 and the size of posture feature shrinks from 480 to 44. Thus, this method can reduce computation workload and memory capacity of machine learning models and satisfy the requirement of embedded devices.

After data processing, four training datasets are created – two for human posture recognition and two for human motion recognition. These datasets can be easily used on machine learning models for training, validation and testing. The detail of data processing and feature extraction are explained in the following sections.

A. *Data Preprocessing*

At the beginning of the implementation phase, raw data should be transformed to a format which can be easily processed and readily retrieved. Data preprocessing methods are applied to solve this problem. The raw data are zipped log files which contain depth values of human interactions [31]. By using data preprocessing methods, these zipped log files are converted to binary value depth images and saved in text files. Pixel values are extracted from each frame and written in a single line of a text file [32]. Timestamps related to the frames are saved in another text file as index of frames. All empty frames are filtered and only valid frames are kept.

B. *Feature Extraction*

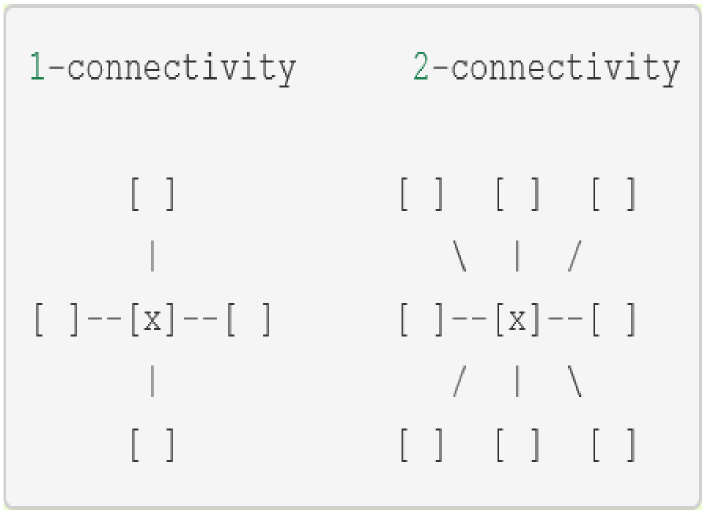
As mentioned before, features need to be extracted from labeled images. The classifier is first trained and used on Intel Core I5 platform and then transplanted to embedded platform which has limited computation capacity. Thus, feature extraction methods that require complex computations should not be applied. Since raw data has already been converted to binary value depth images, those methods designed for the RGB image are not applicable in our project. According to the literature study, learning methods can directly learn from pixel values of images. Thus, pixel values of each training example are extracted as the feature of human action patterns.

Since the size of a depth image is 36x48, the corresponding feature vector can be denoted as x[1,k] (k=1728) where k represents the number of features. When we have 1000 training examples for posture recognition, the input matrix can be denoted as X[n,k] (n=1000,k=1728) where n represents number of examples. If we use MLR as the classifier, the weight parameter used in LR is denoted by W[k,m] (k=1728,m=4) where m represents the number of output classes. The output can be denoted by $h\mathrm{W}(XW)$ where $h(\cdot)$ is the hypothesis work. During the training process, partial derivatives of cost function are calculated to update the weight parameter. The scale of feature matrix affects the speed of matrix operation and the larger feature matrix needs longer computation time.

In order to reduce computational workload and accelerate convergence of the objective function, we introduce a posture separation method that can detect a human body and cut it out by a bounding box. Only pixel values in the detected area are considered as effective features. This method can be achieved by image processing algorithm based on the Scikit-image library [4]. First, connected regions in the depth image are set to different colors. As can be seen in Fig. 5, connectivity refers to the maximum distance between neighbors. In our project, connectivity is set to 2 which means that the pixels with same value in 2 hops can be considered as neighbors. A connected region is a complete set of neighbors in which pixels are linked in range of connectivity. The connectivity of the pixels for behavior analysis is depicted in Fig. 5.

After coloring the connected regions, the number of pixels in each filled area is calculated and the largest region (except the background) is considered as the target region. The target region is then detected and cut out by a bounding box vector (min_row,min_col,max_row,max_col). The separated image cannot be used by learning models because they vary in size and contain different number of pixels. To solve this problem, we resize the separated image to 24x20 which is the best resolution compatible for most images. Pixel values of resized images can now be extracted as features.

Depth images only contain static information of human postures which is not sufficient for motion representation. According to [2], motion history image, a temporal template, can be used for motion feature representation. MHI not only records the presence of motion but also saves the history of a movement from the start frame to the end frame in sequence of images. MHI is created by past successive images using a weighted sum algorithm. Latest image contributes most and produces the brightest part of the MHI. The algorithm is shown below:


```
1-connectivity     2-connectivity

     [ ]           [ ]  [ ]  [ ]
      |              \   |   /
 [ ]--[x]--[ ]     [ ]--[x]--[ ]
      |              /   |   \
     [ ]           [ ]  [ ]  [ ]
```

Fig 5. Connectivity of Pixels.

$$H_T(\mathrm{x,y,t}) = \frac{1}{\gamma}\begin{cases} \tau & if\ D(x,y,t) = 1 \\ \max(0, H_T(\mathrm{x,y,t}-1)-1) & otherwise \end{cases}$$

where $D(x,y,t)$ is a depth image, x and y are locations of pixels, t is the index of frame, τ is the duration, and $H\tau(x,y,t)$ is the generated MHI. Basically, MHI is a vector-image which contains direction information of a motion by combining and vanishing past images step by step. In order to create desirable MHIs which have clear borders and complete motions, key parameters of the algorithm should be determined.

In our implementation, the batch size is tested in the range of {5,10,20,50,100,200} to see how it affects the performance. The mini-batch selection procedure is shown below:

Offset=step*batch$_{size}$ %(n- batch$_{size}$)

batchData = trainDataset[offset: (offset+ batch$_{size}$),:]

In our MLP neural networks, the neuron is a Rectified Linear Unit (ReLU) which employs the rectifier as a non-linear activation function. This rectifier function can be donated by

Relu(x)=max (0,x)

where x is the input of a neuron and the output is the maximum value between 0 and x. This means that it only uses positive values as activations and sets all negative values to 0.

For logistic regression, the cost function with L2-regularization can be denoted by

$$d(\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{Cost}(\mathrm{h}\,\theta\big(x(i), y(i)\big) + \frac{\beta}{2\eta}\sum_{i=1}^{m} h\,\theta_i^2 co$$

Fitting the training set too well is a problem for the classifier. When the classifier is over-fitting to the training examples, it will decrease the prediction accuracy on new examples. To solve this problem, we use regularization to prevent over-fitting and improve the generalization of the classifier. Regularization is one of the most common optimization techniques. It adds a penalty term associated with weight parameters to the cost function of hypothesis work. In this way, it makes a tradeoff between weight shrinking and minimum cost to find the model which has optimal prediction performance on all possible input examples.

IV. RESULTS

This proposed work explains the experimental methods that were to find an optimal classifier and evaluate the results of the experiments conducted using different classifiers. Key parameters of these the classifiers are tested on Intel I5 platform to find the best configuration. To verify the modified classifiers based on architecture 2 and 3, the neural network model is redesigned in the c language and simulated on the sky notes by using Cooja. Performance evaluation is conducted for the simulated classifiers.

This section gives a brief introduction for experiments and explains the setups and measurements. The experiments were tested on 4-class posture recognition, 3-class motion recognition and 5-class motion recognition. Examples of four classes of postures and five classes of motions are shown in figure. The 3-class motion recognition is a simplified version of 5-class motion recognition. We consider “Moving Left” and “Moving Right” as the same class – “Moving”, and consider “Waving Up” and “Waving Down” as the same class – “Waving”. The third class of motion is “Standing”.

At the beginning of the experiment, labeled examples of human actions are grouped into three parts: training set, validation set and test set. The training set is a set of examples used to tune the parameters of the classifier. The validation set is a set of examples used to estimate the performance of the model during the training process. The test set is a set of examples used to assess the performance of a fully-trained classifier. Table I illustrates the configuration of datasets for each recognition tasks. For 3-class and 5-class human motion recognition, the training/validation/test set ratio is 80:10:10 which is a commonly used settings. For 4-class human posture recognition, as there exists big variances between different examples of the same posture, to guarantee the generalizability of the classifier, the training dataset is divided with more validation and test examples.

Fig. 6 shows the result of experiment 1. We choose pixel values as feature of training examples and test the number of training step on 4-class posture recognition task. As can be seen in the figure, when the number of steps increases, the classifier achieves higher validation accuracy and test accuracy. Training accuracy is fluctuating because it is affected by the regularization term which tradeoff training accuracy to guarantee the generalizability. When the training step is larger than 1600, the test accuracy stops increasing and stays around 94.8%. Thus, for this specific task, 1600 steps is sufficient for training the classifier.

TABLE I. CONFIGURATION OF DATASETS

Recognition Tasks	Actions	No.of Examples in the Training set	No.of Examples in the Validation set	No.of Examples in the Test set
4-Class Human Posture Recognition	Left Arm Raised	200	100	150
	Right Arm Raised	200	100	150
	Single Person Standing	200	100	150
	Two people Standing	200	100	150
4-Class Human Motion Recognition	Moving	320	40	40
	Waving	320	40	40
	Standing	320	40	40
4-Class Human Motion Recognition	Moving Left	160	20	20
	Moving Right	160	20	20
	Waving up	160	20	20
	Waving down	160	20	20
	Standing	160	20	20

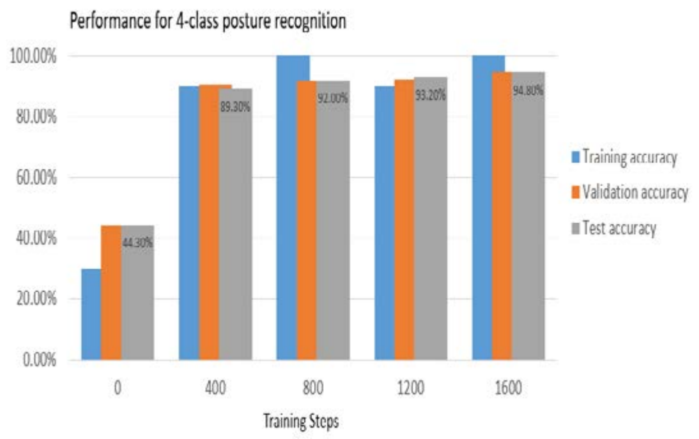


Fig 6. Impact of Training Steps on the Performance of the Classifier.

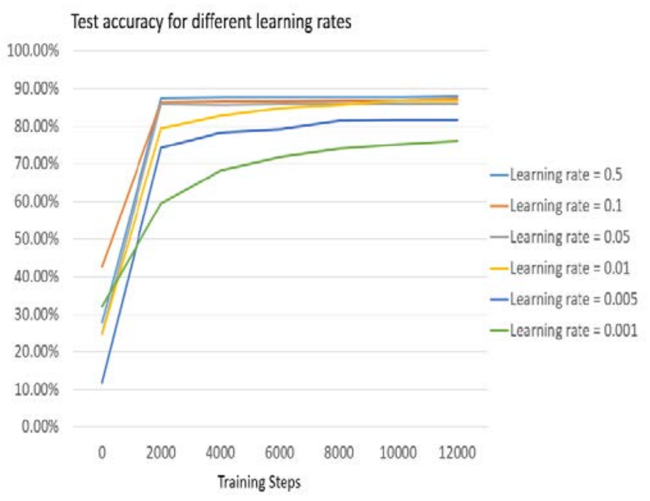


Fig 7. Impact of Learning Rate on the Test Accuracy.

When the learning rate (alpha) is set to a large value like 0.5, the model is unable to find the optimal spot so the resulting test accuracy keeps stably at 90.7%. When the alpha is set to a small value like 0.001, the training process is quite slow and needs more than 12000 steps. We find that 0.1 is a good value for learning rate that reaches the best test accuracy (91.2%). Fig. 7 represents the test accuracy levels.

Batch size is tested on 3-layer neural network for 3-class motion recognition problem using pixel features. When the batch size is set to an extremely low value like 1, the classifier is under-fitting and only gets 33.3% for test accuracy. For other batch sizes, the classifier gets similar performance about 95% ~ 97%. We find 10 is a good value for batch size that leads to the optimal test accuracy (97.5%) on this specific recognition task. Fig. 8 indicates the Impact of beta on the performance of the SGD-MLR classifier.

For a 4-layer neural network, the best configuration is for hidden nodes which leads to the optimal performance – 88% test accuracy. Fig. 9 represents the Impact of dropout on the performance of 3-layer neural networks

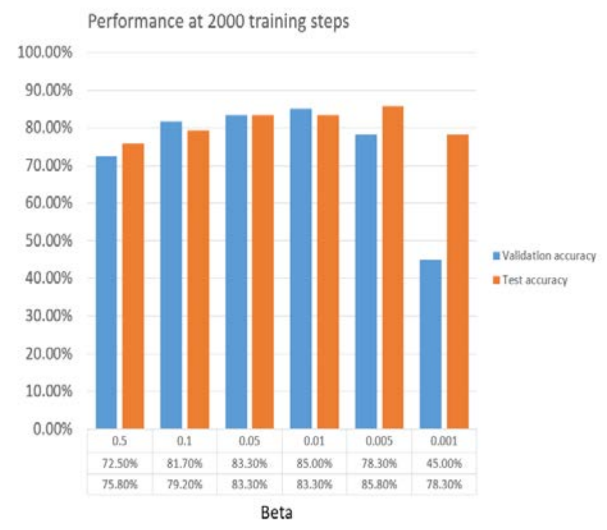


Fig 8. Impact of Beta on the Performance of the SGD-MLR Classifier.

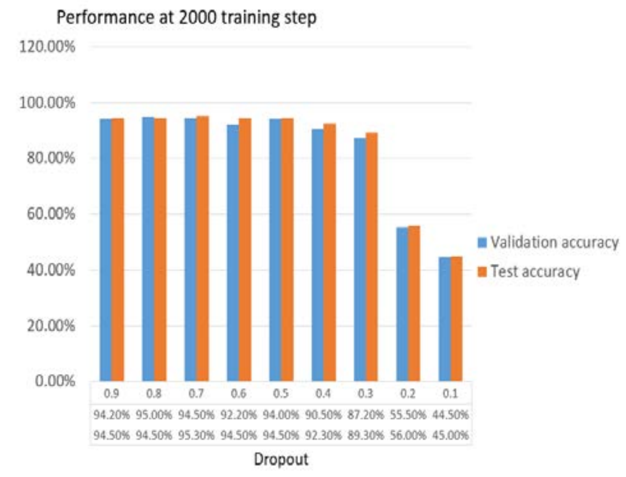


Fig 9. Impact of Dropout on the Performance of 3-Layer Neural Networks.

However, when we add one more hidden layer to the neural network, the performance does not improve a lot. In addition, tuning a 5-layer neural network is much tougher because of the various possible combinations of parameters. Thus, the 4-layer neural network is sufficient for this recognition task. The accuracy levels of the proposed method are illustrated in Table II.

TABLE II. ACCURACY LEVELS

No.of Hidden Layers	No.of Hidden Nodes	Validation Accuracy	Test Accuracy	No.of Hidden Layers	No.of Hidden Nodes	Validation Accuracy	Test Accuracy
1	5	75.0%	72.0%	1	50	89.0%	83.0%
1	20	84.0%	81.0%	1	75	90.0%	84.0%
2	10,5	80.0%	76.0%	2	50,20	87.0%	86.0%
2	20,10	79.0%	83.0%	2	75,50	84.0%	88.0%
3	20,10, 5	77.0%	78.0%	3	75,20, 10	85.0%	86.0%
3	50,20, 10	82.0%	85.0%	3	75,50, 20	87.0%	88.0%

V. Conclusion

The first task of the proposed work is to investigate human action patterns and process training examples. By using the visualization tool, the most common postures and motions are found. Examples of postures and motions are quickly selected and labeled by using the labeled tool. An image processing method is designed to improve the performance of posture recognition. The MHI is implemented to properly represent the human motion. For feature extraction, a projection based feature is presented to efficiently extract high level features from pixel values. Compared with the pixel feature, it reduces the computation workloads and the memory footprints of recognition process. Secondly, the classifiers based on machine learning models are successfully created and trained by using the TensorFLow platform. For both posture and motion recognitions, the trained classifiers show high performance in terms of validation accuracy and test accuracy. Seven experiments are conducted on Intel I5 platform and the optimal configurations of the key parameters are found based on each specific recognition tasks. Several modifications are designed for the neural network model to implement the classifier in a distributed way. We evaluate the performance of the modified neural network models, and it shows a high recognition accuracy for human posture and motion recognitions. This proposed work presents the architecture designs for centralized and distributed recognition systems and evaluate them in terms of extra-functional requirements including the performance and scalability. The designed architectures are simulated on the sensor network using the Cooja simulator and the Sky nodes. We evaluate the simulation tasks by using the confusion matrix and the result shows a quite good recognition accuracy for the distributed recognition system. However, as the Sky node has limited computation capacity, the distributed recognition system has lower throughput and longer response time compared with the centralized one. This problem can be solved in the future work by using more powerful devices and applying better communication mechanisms in the system.

References

[1]. W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev et al., “The kinetics human action video dataset,” arXiv preprint arXiv:1705.06950, 2017.

[2]. C. Gu, C. Sun, S. Vijayanarasimhan, C. Pantofaru, D. A. Ross, G. Toderici, Y. Li, S. Ricco, R. Sukthankar, C. Schmid et al., “Ava: A video dataset of spatio-temporally localized atomic visual actions,” arXiv preprint arXiv:1705.08421, 2017.

[3]. R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag et al., “The” something something” video database for learning and evaluating visual common sense,” in Proc. ICCV, 2017.

[4]. H. Zhao, Z. Yan, H. Wang, L. Torresani, and A. Torralba, “Slac: A sparsely labeled dataset for action classification and localization,” arXiv preprint arXiv:1712.09374, 2017.

[5]. M. Monfort, B. Zhou, S. A. Bargal, T. Yan, A. Andonian, K. Ramakrishnan, L. Brown, Q. Fan, D. Gutfruend, C. Vondrick et al., “Moments in time dataset: one million videos for event understanding.”

[6]. S. Lai, W.-S. Zhang, J.-F. Hu, and J. Zhang, “Global-local temporal saliency action prediction,” IEEE Transactions on Image Processing, vol. 27, no. 5, pp. 2272–2285, 2018.

[7]. K. Tang, V. Ramanathan, L. Fei-Fei, and D. Koller, “Shifting weights: Adapting object detectors from image to video,” in Advances in Neural Information Processing Systems, 2012.

[8]. S. Satkin and M. Hebert, “Modeling the temporal extent of actions,” in ECCV, 2010.

[9]. J. Wu, I. Yildirim, J. J. Lim, W. T. Freeman, and J. B. Tenenbaum, “Galileo: Perceiving physical object properties by integrating a physics engine with deep learning,” in Advances in Neural Information Processing Systems, 2015, pp. 127–135.

[10]. J. Hou, X. Wu, J. Chen, J. Luo, and Y. Jia, “Unsupervised deep learning of mid-level video representation for action recognition,” in AAAI, 2018.

[11]. Fatima, I.; Fahim, M.; Lee, Y.K.; Lee, S. A unified framework for activity recognition-based behavior analysis and action prediction in smart homes. Sensors **2013**, 13, 2682–2699.

[12]. Chen, L.; Nugent, C.D.; Wang, H. A knowledge-driven approach to activity recognition in smart homes. IEEE Trans. Knowl. Data Eng. **2012**, 24, 961–974.

[13]. Aloulou, H.; Mokhtari, M.; Tiberghien, T.; Biswas, J.; Yap, P. An adaptable and flexible framework for assistive living of cognitively impaired people. IEEE J. Biomed. Health Inform. **2014**, 18, 353–360.

[14]. Krüger, F.; Nyolt, M.; Yordanova, K.; Hein, A.; Kirste, T. Computational state space models for activity and intention recognition. A feasibility study. PLoS ONE 2014, 9, e109381, doi:10.1371/journal.pone.0109381.

[15]. K. K. Reddy and M. Shah, “Recognizing 50 human action categories of web videos,” Machine Vision and Applications Journal, 2012.

[16]. Kurakin, Z. Zhang, and Z. Liu, “A real-time system for dynamic hand gesture recognition with a depth sensor,” in EUSIPCO, 2012.

[17]. O. Kliper-Gross, T. Hassner, and L. Wolf, “The action similarity labeling challenge,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 3, 2012.

[18]. H. S. Koppula, R. Gupta, and A. Saxena, “Learning human activities and object affordances from rgb-d videos,” International Journal of Robotics Research, 2013.

[19]. G. Yu, Z. Liu, and J. Yuan, “Discriminative orderlet mining for real-time recognition of human-object interaction,” in ACCV, 2014.

[20]. Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang, “Exploiting feature and class relationships in video categorization with regularized deep neural networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 2, pp. 352–364, 2018. Available: https://doi.org/10.1109/TPAMI.2017.2670560.

[21]. Arepalli, Gopi & Erukula, Suresh & Gopi, A.P. & Nagaraju, Chiluka. (2016). Secure multicast routing protocol in MANETs using efficient ECGDH algorithm. International Journal of Electrical and Computer Engineering (IJECE). 6. 1857-1865. 10.11591/ijece.v6i4.9941.

[22]. K. Sarada, V. Lakshman Narayana,(2020), “Improving Relevant Text Extraction Accuracy using Clustering Methods”, TEST Engineering and Management, Volume 83, Page Number: 15212 – 15219.

[23]. K. Sarada, V. Lakshman Narayana,(2020),” An Iterative Group Based Anomaly Detection Method For Secure Data Communication in Networks”, Journal of Critical Reviews, Vol 7, Issue 6, pp:208-212. doi: 10.31838/jcr.07.06.39.

[24]. Banavathu Mounika, P. Anusha, V. Lakshman Narayana,(2020), “ Use of BlockChain Technology In Providing Security During Data Sharing”, Journal of Critical Reviews, Vol 7, Issue 6, pp:338-343. doi: 10.31838/jcr.07.06.59.

[25]. Lakshman narayana, b. Naga Sudheer,(2020),” fuzzy base artificial neural network model for text extraction from images”, journal of critical reviews, vol 7, issue 6,pp:350-354, doi: 10.31838/jcr.07.06.61.

[26]. V. Lakshman Narayana, A. Peda Gopi,(2020),” Accurate Identification And Detection Of Outliers In Networks Using Group Random Forest Methodoly”, Journal of Critical Reviews, Vol 7, Issue 6,pp:381-384, doi: 10.31838/jcr.07.06.67.

[27]. Sandhya Pasala, V. Pavani, G. Vidya Lakshmi, V. Lakshman Narayana,(2020),” Identification Of Attackers Using Blockchain Transactions Using Cryptography Methods”, Journal of Critical Reviews, Vol 7, Issue 6,pp:368-375, doi: 10.31838/jcr.07.06.65

[28]. C.R.Bharathi, Vejendla. Lakshman Narayana , L.V. Ramesh, (2020),” Secure Data Communication Using Internet of Things”, International Journal of Scientific & Technology Research, Volume 9, Issue 04,pp:3516-3520.

[29]. Lakshman Narayana Vejendla and Bharathi C R ,(2018),“Multi-mode Routing Algorithm with Cryptographic Techniques and Reduction of Packet Drop using 2ACK scheme in MANETs”, Smart Intelligent Computing and Applications, Vo1.1, pp.649-658. DOI: 10.1007/978-981-13-1921-1_63 DOI: 10.1007/978-981-13-1921-1_63

[30]. Chaitanya, K., and S. Venkateswarlu,(2016),"Detection Of Blackhole & Greyhole Attacks In Manets Based On Acknowledgement Based Approach." Journal of Theoretical and Applied Information Technology 89.1: 228.

[31]. Lakshman Narayana Vejendla , A Peda Gopi and N.Ashok Kumar,(2018),“ Different techniques for hiding the text information using text steganography techniques: A survey”, Ingénierie des Systèmes d'Information, Vol.23, Issue.6,pp.115-125. DOI: 10.3166/ISI.23.6.115-125

[32]. A Peda Gopi and Lakshman Narayana Vejendla (2018), “Dynamic load balancing for client server assignment in distributed system using genetic algorithm”, Ingénierie des Systèmes d'Information, Vol.23, Issue.6, pp. 87-98. DOI: 10.3166/ISI.23.6.87-98

[33]. B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles, “Activitynet: A large-scale video benchmark for human activity understanding,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 961–970.

[34]. G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari, “Charades-ego: A large-scale dataset of paired third and first person videos,” arXiv preprint arXiv:1804.09626, 2018.