Applied Soft Computing xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Hybrid deep neural network model for human action recognition

Earnest Paul Ijjina*, C. Krishna Mohan

Visual Learning and Intelligence (VIGIL) Lab, Department of Computer Science & Engineering, Indian Institute of Technology Hyderabad, Telangana 502205, India

ARTICLE INFO

Article history:
Received 15 June 2015
Received in revised form 10 August 2015
Accepted 13 August 2015
Available online xxx

Keywords:
Deep neural network
Convolutional neural network (CNN)
Classifier fusion
Action bank features

ABSTRACT

In this paper, we propose a hybrid deep neural network model for recognizing human actions in videos. A hybrid deep neural network model is designed by the fusion of homogeneous convolutional neural network (CNN) classifiers. The ensemble of classifiers is built by diversifying the input features and varying the initialization of the weights of the neural network. The convolutional neural network classifiers are trained to output a value of one, for the predicted class and a zero, for all the other classes. The outputs of the trained classifiers are considered as confidence value for prediction so that the predicted class will have a confidence value of approximately 1 and the rest of the classes will have a confidence value of approximately 0. The fusion function is computed as the maximum value of the outputs across all classifiers, to pick the correct class label during fusion. The effectiveness of the proposed approach is demonstrated on UCF50 dataset resulting in a high recognition accuracy of 99.68%.

1. Introduction

The famous 'no free lunch' theorem [1] proposed by Wolpert suggests that there is no single computational view that solves all pattern recognition tasks. This lead to an increased interest in combining several classifier systems that perform information fusion of classification decisions thereby over-coming the limitations of using a single classifier. Several techniques like hybrid intelligent systems, multi-classifier systems, information fusion were proposed in the literature for classification, employing several computational views. While information fusion techniques combine information from different sources to recognize a new view for better classification, multi-classifier systems focus on combining different classifier models for effective classification. Hybrid intelligent systems employs intelligent techniques in various computational phases from data normalization to final decision making to obtain a blend of heterogeneous fundamental views for effective classification.

A true function that cannot be modeled by a single hypothesis can now be modeled as a combination of hypotheses. One of the advantage of using a hybrid intelligent systems is its ability to handle the two extreme cases in availability of training data. The scenario where data samples are scarce can be effectively handled by considering bootstrapping methods like boosting [2] and the scenario with huge number of data samples by combining decisions of classifiers trained on partitions of data [3]. A multi-classifier system can outperform the best individual classifier [4] and this was analytically proved in [5] by considering majority voting on a group of independent classifiers. In case of classifiers using heuristic approaches for optimization, that does not ensure an optimal solution, but a combined approach may increase the probability of finding an optimal model. As stated by Wolpert [1], each classifier has its specific competence domain and choosing an ensemble of heterogeneous classifiers would result in an effective classification model.

The general structure of a multiple classifier system (MCS) consists of a classifier ensemble with a set of diverse classifiers. The most discriminative features are given as input to the classifier ensemble and a fusion method is used to optimally combine the individual classifier outputs for classification. Thus, the main design issues in a MCS are: (1) *system topology*: describing the interconnection between classifiers, (2) *ensemble design*: defining the generation and selection of a pool of classifiers, and (3) *fuser design*: a decision combination function that optimally combines the outputs of classifiers. Some of the multiple classifier system (MCS) proposed in the literature are discussed in the following section. The two popular MCS system topologies are a parallel topology [6] and a serial (or conditional) topology. In a parallel topology, the same input data is fed to all the classifiers and the output generated by these classifiers is used for decision making. As the classification output of one classifier is independent of the output of other classifiers, this approach is more suited when considering classifiers with

* Corresponding author. Tel.: +91 9494466490.
E-mail addresses: cs12p1002@iith.ac.in (E.P. Ijjina), ckm@iith.ac.in (C.K. Mohan).

http://dx.doi.org/10.1016/j.asoc.2015.08.025
1568-4946/

Please cite this article in press as: E.P. Ijjina, C.K. Mohan, Hybrid deep neural network model for human action recognition, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.08.025

low support/confidence in classification. The sequential approach is considered when the cost of classifier exploration is high. The classifiers are arranged in sequence of increasing computation cost i.e., the classifier with the least computation cost will be the first classifier in the pipeline. In [7], each classifier gives an estimate of the certainty of classification and the uncertain data samples are sent to the next classifier in the pipeline. A reject-option [8] can also be used in serial topology and Adaboost [9] is a special case of sequential topology.

Ensemble design in a MCS aims to include mutually complementary classifiers that are characterized by low classifier output correlation [10] and high accuracy [11]. Dietterich in [12] empirically validated that a robust classifier can be built by combining the evidences of complementary classifiers. Brown in [13] suggests that diversity can be achieved using implicit or explicit approaches. Implicit techniques involves use of random techniques to generate individual classifiers while explicit approaches focus on optimizing a diversity metric in an ensemble of classifiers. The wide range of experimental results in [14] suggests that increasing diversity should result in a combined system with better accuracy. According to [6,15], diversity of classifiers can be enforced by manipulation of either individual classifier inputs, outputs, or models. Some of the approaches for diversifying input data are: (1) using different data partitions, (2) using different set of features, and (3) taking local specialization of individual classifiers into consideration. Local specialization is a classifier selection approach that selects the best classifier from a pool of classifiers trained on partitions of the features space. Diversity in MCS can also be achieved by considering classifiers designed to classify only a subset of classes and applying combination technique to restore the whole class label set. Finally, the diversified models in the ensemble should be combined to take advantage of the homogeneous/heterogeneous combination of models. As some classifiers are more efficient for some domains, an ensemble of heterogeneous classifiers would result in an solution well-addressed in multiple domains. As most machine learning algorithms (like neural networks [16]) would try to find an optimal solution from a given initial setup, combining homogeneous (identical) models with various initializations may improve classification performance.

An effective fuser is a crucial requirement for an efficient classifier. A fuser combines the outputs of the selected classifiers from the ensemble to give a final decision of the MCS system. The outputs of the classifier could be the class label associated with the test instance or the support (confidence value) for test instance to belong to a class. Early implementations of fusion models considered majority voting [6] that determined the final class label by (1) *unanimous voting* where the decision is unanimous, or (2) *simple majority* decision made if majority is more than half of the selected classifiers, or (3) *majority voting* where decision is to select the class with highest number of votes. Later, alternate voting methods [6,17] were proposed that assign different weights to the outputs of the classifiers. Fusion models based on support, use a support function to compute the confidence of a classifier in its decision. Some of the well know approaches are the ranking based approach of Borda count [18], the posterior probability approaches [19–21] and combination of accuracy of neural networks [22]. Trainable fusers were proposed by considering the weights used to combine classifier outputs as a learning process [23,24]. Perception learning with evolutionary approaches were used by Wozniak in [25] to train a fuser and Zheng used data envelopment analysis in [26]. A experimental comparison of various fusion functions along with their sensitivity analysis was done in [27].

Among the high-dimension data, human action recognition in videos poses a unique challenge due to the existence of temporal dimension whose length varies with each instance and subject executing the action. The inconsistencies in the execution of actions, the environment and capturing conditions further complicates the observed data, that is in-turn used as input for recognition algorithms. Some of the most commonly used features for human action recognition are histogram of oriented gradients (HOG) [28], histogram of optical flow (HOF) [28], motion boundary histograms (MBH) [29] and motion interchange patterns (MIP) [30]. These features are used with some classical approaches like support vector machine (SVM), neural networks and k-nearest neighbor to compute the base results for most of the action recognition datasets. Nazli Ikizler-Cinbis et al. used different features with multiple instance learning (MIL) framework to utilize the entities related to an action like the scene, objects and people for action recognition in [31]. In [32], Fabian Caba Heilbron et al. used dense point trajectories to extract context from foreground motion to recognize actions in videos with camera motion. In [33], Salah Althloothi et al. also used multiple features for human action recognition in RGB-D videos by using multiple kernel learning. An ensemble of homogeneous models are used by Karen Simonyan et al. [34] for object recognition in ImageNet Large Scale Visual Recognition Competition (ILSVRC) [35]. Samira Ebrahimi Kahou et al. [36] used fusion of models trained on different modalities to improve the efficiency of their model in Emotion Recognition In The Wild (EmotiW) [37] challenge. Mengyi Liu et al. in [38] combined multiple kernel methods trained for different modalities using a trained fusion function. Multi-resolution CNN architecture with time information fusion is used by Andrej Karpathy et al. in [39] for human action recognition. These approaches assert the need for using multiple features and classifiers to design an effective classification model.

In this paper, we propose a hybrid classifier for action recognition by fusion of evidences generated by homogeneous models arranged in a parallel topology. A convolutional neural network classifier designed to recognize human actions from action bank features is used to build the ensemble of classifiers. The novelty of the proposed approach lies in achieving the diversity of models by manipulation of the input data using complementary features and by varying the initialization of neural network weights. Also, we use a fusion function that exploits the high confidence value of classifiers for correct prediction, to pick the correct class label across outputs. The reminder of this paper is organized as follows: Section 1 gives an introduction to multi-classifier systems and the various approaches in the literature for classifier fusion. Section 2 introduces the proposed hybrid system along with the approaches used to diversify the models and the fusion function. Section 3 covers the experiment setup and results followed by analysis of results. Section 4 gives the conclusions and future directions of this work.

2. Human action recognition using fusion of CNN classifiers

This section presents the CNN classifier architecture used to generate the ensemble of classifiers in the proposed fusion model. Different initial weights are used to generate multiple CNN classifiers. These weights are determined by a random number generator that is initialized by a seed value. By using n unique seed values, n different weight initializations of CNN classifier are constructed. Corresponding to each weight initialization, we train one CNN classifier on action bank features and another CNN classifier on complementary action bank features. The complementary action bank features are computed by taking the complement of action bank features interpreted as an image. As a result, $2n$ number of CNN classifiers (that are assumed to be implicitly diverse) are constructed in the ensemble. The block diagram of the proposed model to recognize 'c' classes using fusion of $2n$ models is shown in Fig. 1. The CNN classifier initialized with seed value i and using action bank features (AB) as input is represented by CNN_i. Similarly, the

Please cite this article in press as: E.P. Ijjina, C.K. Mohan, Hybrid deep neural network model for human action recognition, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.08.025

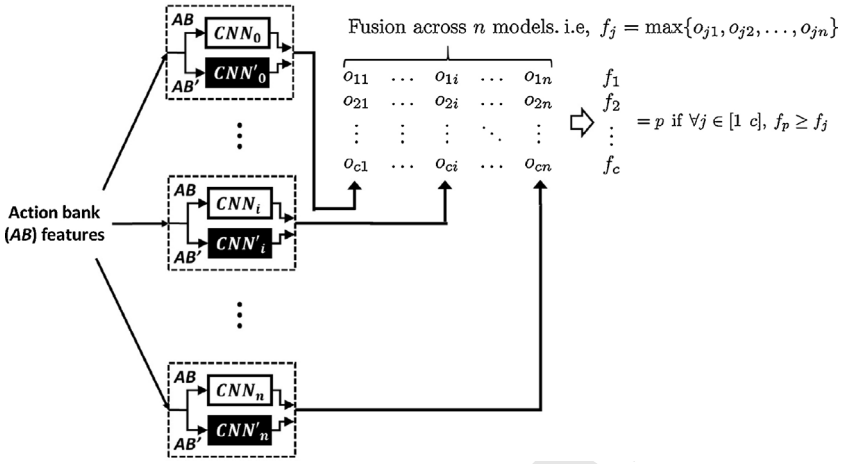


Fig. 1. Block diagram of the proposed hybrid CNN model for human action recognition.

CNN classifier initialized with seed value i and using complementary action bank features (AB') as input is represented by CNN'_i. Here, initializing a CNN classifier with seed value i refers to passing seed value i to the random number generator that initializes the weights in the CNN classifier. To simplify the description and analysis of the proposed model, we first consider the fusion of models with identical initial weights i.e., CNN_i and CNN'_i to compute the performance of the combined model CNN^*_i. Next, we compute the performance of the fusion model using the outputs of the combined models. The CNN classifiers are trained to generate binary decoded outputs i.e., a value of 1 corresponding to the index of the predicted class and 0's for the remaining classes. To select the outputs with a confidence value of approximately 1 (i.e., correct predictions), we use the maximum value function across models as the fusion function to generate the outputs f_j as shown in the model. The outputs of fusion function $f_1, f_2, \ldots, f_c$ are interpreted as binary decoded outputs to compute the predicted class label of the fusion model. The following subsections explain the various aspects of the proposed fusion model in detail.

2.1. Convolutional neural network classifier for human action recognition

In this work, we consider the CNN classifier architecture proposed by Ijjina et al. in [40] that considers action bank features extracted from videos as input to a CNN classifier for action recognition. Action bank features [41] of a video are computed using an action bank which is a fixed set of template videos. To compute the action bank features of a video, the similarity information of the new video against each video in the action bank is captured in a vector of size 73. If an action bank of size n is used, the action bank features generated will be of size $n \times 73$. The 202×73 action banks features of videos with boxing and running action from KTH dataset are shown in Fig. 2.

From Fig. 2, it can be observed that videos of same action will have similar action bank features. The emergence of this similarity in action bank features is due to the use of same action bank template videos for the generation of action bank features for all videos. The CNN classifier architecture proposed in [40] aims to learn and utilize the local linear patterns associated with each action in action bank features for human action recognition. The next section explains how multiple CNN classifiers are generated using the architecture introduced in this section.

2.2. Generation of multiple CNN classifiers

The architecture of the CNN classifier used in this work is shown in Fig. 3. The performance of this CNN classifier depends on the inputs used for action recognition and the initial weights of the neural network. As explained in Section 1, the approaches used in the literature to enforce diversity of classifiers are manipulation of individual classifier inputs, outputs, or models. We aim to

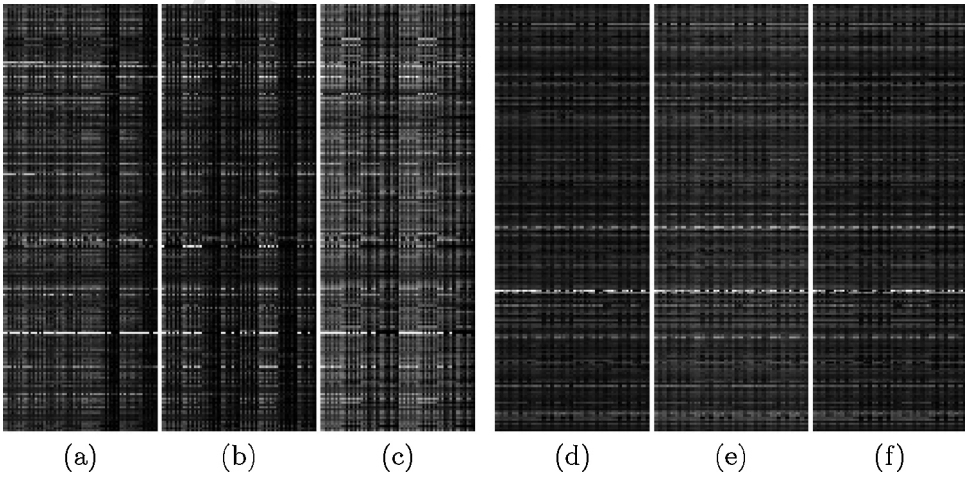


Fig. 2. Action bank features of boxing and running videos in KTH dataset: (a–c) are for boxing and (d–f) are for running action.

Please cite this article in press as: E.P. Ijjina, C.K. Mohan, Hybrid deep neural network model for human action recognition, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.08.025

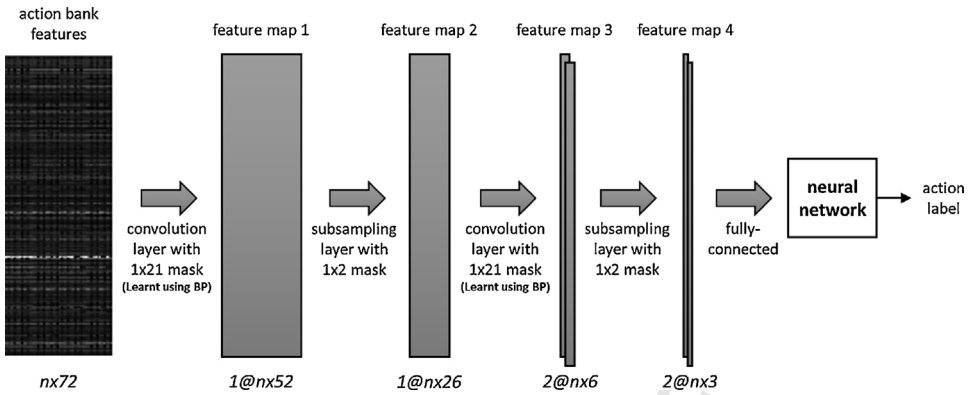


Fig. 3. CNN classifier architecture considered.

achieve diversity by (1) diversifying the input data through utilization of complementary action bank features, and (2) diversifying the model by generating different versions of the same model by varying model-initialization.

In the first approach, we diversify the input data by utilizing complementary action bank (AB') features for action recognition. As explained in Section 2.1, the input to the CNN classifiers are the action bank features (AB) interpreted as gray scale images shown in Fig. 2. The complementary action bank features (AB') computed by inversion of gray scale image generated using action bank features, gives an alternate representation of action bank features that preserve the local patterns. Thus, we aim to achieve diversity of classifiers by using complementary action bank (AB') features for action recognition. The second approach is to generate different versions of the same model by varying model-initialization. As the CNN classifier used in this work (Fig. 3) is implemented using a neural network, the performance of the trained classifier depends upon the initial weights of the neural network. By changing the seed value of the random number generator that initializes the neural network weights, we can generate different versions of the same model thereby diversifying the classifiers in the ensemble. The next section explains how the outputs of these classifiers are fused for classification by the hybrid model.

2.3. Fusion of CNN classifiers

The previous sections explained the CNN classifier used in this work and how multiple diverse models of this classifiers are generated for classifier fusion. Using maximum value as fusion function across outputs of binary decoded CNN classifiers will result in a hybrid system with binary decoded output. As neural networks are used to implement the CNN classifier and back-propagation is used for training, a well trained classifier will generate an output close to one for the correct class label and zero for the remaining classes. Thus, using maximum value across classifier outputs as the fusion function selects the correct class label across models due to high confidence ($\approx$1) for correct classification and low confidence ($\approx$0) associated with incorrect classification. The next section discusses the experimental results on UCF50 dataset.

3. Experimental results

The hybrid model introduced in the previous section is evaluated on UCF50 dataset by 5-fold cross validation. The action bank features of size 207×72 generated for videos in UCF50 dataset are used as an input to the CNN classifier. The range of the seed value (n in Fig. 1) is empirically determined to be between 0 and 8, resulting in 18 models for each set during the 5-fold cross validation. Experiments are conducted using the proposed hybrid model with various fusion functions. The performance of the proposed approach using different fusion functions is given in Table 1. From the table, it can be observed that the classification error is minimum when *Avg* and *Max* rules are used for fusion of models. This could be due to the design of models to have high confidence for correct classification and low confidence for misclassification. The misclassification of 139 observations using *majority voting* suggests that most of the base classifiers misclassified these observations. The misclassification of 436 observations by *Median* rule suggests that more than half of the base classifiers misclassified these observations. The number of misclassifications using *Prod* rule depends on the (product of) outputs of all classifiers used in fusion. As a result, classification with low confidence affects the corresponding fusion value (f_j in Fig. 1) used to assign class label. Fusion using *Min* rule has maximum error and is not suitable for this model as the base classifiers are trained to have high confidence for correct class labels.

Due to the low classification error of the proposed fusion model for *Avg* and *Max* rules, we analyze the variation in misclassified observations for these two cases with increase in the seed value (i) in Table 2. The number of observations misclassified by the generated CNN classifiers is shown in third and fourth columns of this table. The CNN_i column corresponds to the models trained on original action bank features and the CNN'_i column corresponds to the

Table 1
Performance of the proposed model using various fusion functions for 5-fold cross-validation of UCF50 dataset.

Data split	# of observations	Fusion function					Majority voting
		Min	Max	Avg	Prod	Median	
Set 1	1345	1287	1	0	158	109	16
Set 2	1320	578	10	10	141	69	27
Set 3	1325	1288	5	5	129	78	35
Set 4	1315	1085	3	3	180	80	28
Set 5	1312	1243	2	2	173	100	33
Total	6617	5481	**21**	**20**	781	436	139
Error (in %)		82.83	**0.317**	**0.302**	11.80	6.59	2.1

Please cite this article in press as: E.P. Ijjina, C.K. Mohan, Hybrid deep neural network model for human action recognition, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.08.025

Table 2
Classification performance (in # of misclassified observations) for the five splits of UCF50 dataset.

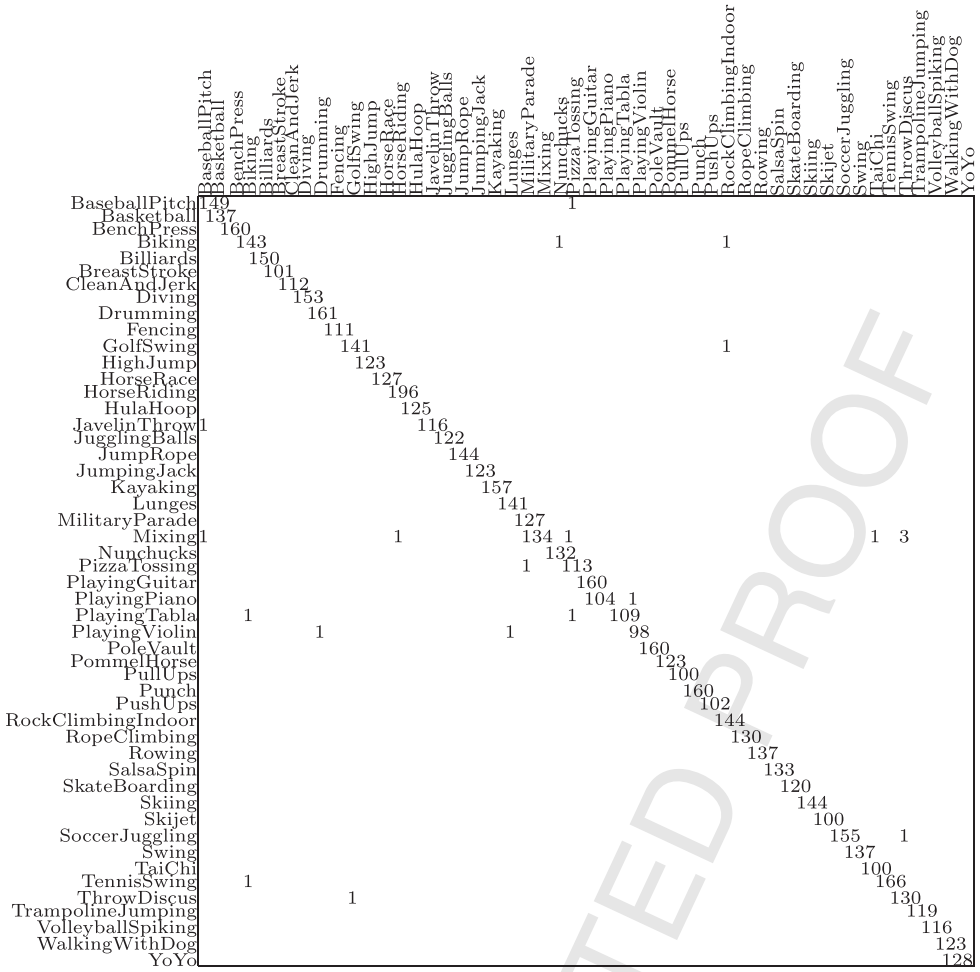
Data split #	Seed value (i)	CNN_i	CNN'_i	CNN^*_i ($CNN_i \cup CNN'_i$)		Fusion $\left(\bigcup_{0\le k\le i} CNN^*_k\right)$	
				Max	Avg	Max	Avg
1	0	130	202	95	95	95	95
	1	163	211	74	74	22	22
	2	141	169	50	50	14	13
	3	157	86	28	28	4	3
	4	180	155	43	43	4	3
	5	89	1303	89	89	1	0
	6	164	1303	164	164	1	0
	7	167	178	87	87	1	0
	8	178	183	46	46	1	0
2	0	164	228	89	89	89	89
	1	182	154	53	53	22	22
	2	131	155	52	52	20	20
	3	206	94	49	49	19	19
	4	210	143	34	34	15	15
	5	108	85	54	54	14	14
	6	213	94	31	31	10	10
	7	160	175	87	87	10	10
	8	174	162	88	88	10	10
3	0	147	196	79	79	79	79
	1	219	131	43	43	28	28
	2	211	138	60	60	20	20
	3	157	130	42	42	12	12
	4	218	1291	218	218	11	11
	5	111	100	57	57	5	5
	6	191	127	67	67	5	5
	7	121	145	69	69	5	5
	8	167	145	72	72	5	5
4	0	147	221	93	93	93	93
	1	1010	131	76	76	20	20
	2	133	140	41	41	13	13
	3	304	200	103	103	8	8
	4	173	164	64	64	5	5
	5	80	124	46	46	4	4
	6	207	100	36	36	4	4
	7	133	193	60	60	4	4
	8	284	111	45	45	3	3
5	0	139	159	63	63	63	63
	1	145	84	46	46	29	29
	2	162	134	62	62	23	23
	3	169	88	37	37	13	13
	4	219	122	83	83	12	12
	5	107	1263	107	107	10	10
	6	171	1287	171	171	8	8
	7	266	982	226	226	4	4
	8	209	78	21	21	2	2

performance of models trained on complementary action bank features. Here, i represents the seed value used to initialize the CNN classifier. The columns labeled CNN^*_i contains the performance of fusion of CNN_i and CNN'_i models i.e., ($CNN_i \cup CNN'_i$). Entries in the columns labeled $\bigcup_{0\le k\le i} CNN^*_k$ denote the fusion of combined models as the seed value increases from 0 to 8. The number of observations misclassified by various fusion models for *Avg* and *Max* rules are computed for comparative study.

In Table 2, for split 1 and seed value 0, the classifier CNN_0 has 9.66% misclassification error (130 misclassified observation) and the classifier CNN'_0 has 15.01% misclassification error (202 misclassified observation). The classifier CNN^*_0 which is a fusion of CNN_0 and CNN'_0 gives an improved performance of misclassification error of 7.06% (95 misclassified observation). This could be due to the dissimilarity of the observations recognized by CNN_0 and CNN'_0. Similarly, for split 1 and seed value 1, the classifier CNN_1 achieves a misclassification error of 12.11% (163 misclassified observation) and the classifier CNN'_1 achieves a misclassification error of 15.68% (211 misclassified cases). The classifier CNN^*_1 which is a fusion of CNN_1 and CNN'_1 gives a misclassification error of 5.50% (74 misclassified observation). The same analysis can be extended to the remaining combined models $CNN^*_2, \ldots, CNN^*_8$ to obtain their corresponding misclassification error. For split 1, the fusion across the combined models i.e., $\bigcup_{0\le k\le i} CNN^*_k$ using *Max*-rule, results in a misclassification error of 0.07% (1 misclassified observation). Similar analysis can be extended for the fusion of models in data splits 2, 3, 4 and 5. From these results, it can be observed that for any given data split and seed value, the performance of the combined model $CNN^*_i = (CNN_i \cup CNN'_i)$ is better than or equal to the individual models (CNN_i and CNN'_i). Also, the performance of the fusion model either improves or remains same with the addition of each combined model. The performance of the final fusion model i.e.,

Please cite this article in press as: E.P. Ijjina, C.K. Mohan, Hybrid deep neural network model for human action recognition, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.08.025

Table 3
Confusion matrix of the proposed approach for 5-fold cross validation of UCF50 dataset.



$\bigcup_{0\le i\le 8} CNN_i^*$ is significantly better than the performance of individual combined models (CNN_i^*). This could be due to the dissimilarity in the observations recognized by the models. Also, fusion using models with high misclassification (like CNN'_5, CNN'_6 for set 1 and CNN'_4 for set 3) would not deteriorate the performance.

The number of test cases and the number of misclassified observations for various fusion functions in each data split of UCF50 dataset is given in Table 1. Using *Max* rule as fusion function, less than 11 observations are misclassified in each split and 21 observations are misclassified across all the splits. The misclassification of 21 observations among 6617 test cases results in a misclassification error of 0.317% i.e., a recognition accuracy of 99.68%. Similarly, using *Avg* rule as fusion function, 20 observations among 6617 test cases gets misclassified resulting in a misclassification error of 0.302% i.e., a recognition accuracy of 99.7%. The confusion matrix of the proposed hybrid model for UCF50 dataset is shown in Table 3. The labels on the vertical axis indicate the actual class labels and the labels on the horizontal axis indicate the predicted class labels. It can be observed that among the 145 test instances of *Biking*, 143 were recognized as *Biking*, 1 as *Nunchucks* and 1 as *RockClimbingIndoor*. The diagonal elements represent the correctly predicted test cases and the non-diagonal elements represent the misclassified test cases.

Table 4 depicts the comparison of the proposed approach with the existing work in literature for 5-fold cross validation of UCF50 dataset. Among the existing approaches, the state of the art approach for human action recognition on UCF50 dataset has a recognition accuracy of 94.1%. This approach was proposed by Nicolas Ballas et al. in [42] and it uses spatio-temporal context and weighted SVM to build an action model from salient regions. Even though the proposed approach may be computationally more expensive than existing approaches, the computation time can be significantly reduced by using a GPU-accelerated implementation like *NVIDIA CUDA Deep Neural Network (cuDNN)* [43] library.

The proposed approach outperforms the current state of the art approach by around 4%, achieving a near ideal recognition accuracy of 99.7%. The improvement in performance could be due to the high recognition accuracy of the proposed fusion model, the diversity of

Table 4
Performance comparison of the proposed approach with existing techniques on UCF50 dataset.

Approach	Accuracy (in %)
Sadanand and Corso [41]	57.9
Kliper-Gross et al. [44]	68.51
Shi Feng et al. [45]	71.7
LiMin Wang et al. [46]	71.7
Wang et al. [47]	75.7
Qiang Zhou et al. [48]	80.2
Ijjina Earnest et al. [40]	94.02
Nicolas Ballas et al. [42]	94.1
Proposed fusion model (with *Max*-rule)	**99.68**
Proposed fusion model (with *Avg*-rule)	**99.7**

Table 5
Variation in the number of correctly classified test cases between CNN classifiers considered in fusion for UCF50 dataset set 1 (in %).

Set 1	CNN_0	CNN_1	CNN_2	CNN_3	CNN_4	CNN_5	CNN_6	CNN_7	CNN_8	CNN'_0	CNN'_1	CNN'_2	CNN'_3	CNN'_4	CNN'_5	CNN'_6	CNN'_7	CNN'_8
CNN_0	0	7	6	7	8	3	6	6	7	8	12	9	4	8	87	87	8	11
CNN_1	5	0	5	7	9	4	7	8	9	9	10	8	3	7	85	85	8	9
CNN_2	5	6	0	8	10	5	7	9	8	11	11	9	4	7	86	86	9	10
CNN_3	5	8	7	0	8	5	7	7	7	9	13	11	4	9	85	85	9	11
CNN_4	4	8	7	6	0	3	7	7	6	8	13	10	4	8	83	83	7	10
CNN_5	6	9	9	10	10	0	9	8	11	11	14	11	5	9	90	90	11	12
CNN_6	3	7	5	6	8	4	0	6	5	7	12	9	4	8	85	85	6	11
CNN_7	3	7	7	6	8	2	6	0	7	8	12	10	4	7	85	85	7	11
CNN_8	3	7	6	6	6	4	4	6	0	7	12	9	4	8	84	84	6	10
CNN'_0	3	6	6	6	6	3	5	5	6	0	12	10	3	8	82	82	7	11
CNN'_1	6	7	6	9	11	5	9	9	10	12	0	3	3	4	81	81	10	4
CNN'_2	7	8	7	10	10	5	9	10	10	12	6	0	3	4	84	84	11	4
CNN'_3	7	9	8	10	11	5	10	10	11	12	13	9	0	9	90	90	11	11
CNN'_4	6	7	6	9	10	4	8	8	10	11	8	5	4	0	85	85	10	6
CNN'_5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CNN'_6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CNN'_7	4	7	7	7	7	4	5	6	6	9	13	10	4	9	84	84	0	11
CNN'_8	7	7	7	9	10	5	9	10	10	13	6	3	4	4	83	83	11	0

models used in fusion and training of classifiers to generate high confidence value for correct class labels. The next section analyzes these factors to verify their contribution to the overall effectiveness of the propose model.

3.1. Detailed analysis of results

The working principle behind fusion of classifiers approach is explained using a Venn diagram shown in Fig. 4. In this figure, if circular regions A and B denote the observations correctly classified in the universe of samples S by models CNN_A and CNN_B, respectively, then the fusion of these two models is defined by the region $(A \cup B)$ that is equal to $(A \cap B) + (A - B) + (B - A)$. This implies that the number of correctly classified samples by a fusion model will be minimum when $(A = B)$ and will be maximum when $(A \cap B) = \emptyset$. Here, $\emptyset$ denotes the empty set. Hence, if the observations classified by the models CNN_A and CNN_B are mutually exclusive, the fusion of the models will be maximum. But, in a practical scenario where $(A \cap B) \neq \emptyset$, the number of observations correctly classified by the fusion model can be increased by increasing the number of samples correctly classified by only one of the models i.e., the region $((A - B) \cup (B - A))$. Therefore, misclassification by fusion model can be decreased by increasing the number of samples correctly classified by only one of the classifiers rather than by both the classifiers.

From Fig. 4, some of the conditions for building an effective fusion model when considering the maximum value as fusion function are: (a) *diversity of classifiers:* the existence of observations that are correctly classified by only one model, denoted by region $((A - B) \cup (B - A))$ and, (b) the confidence value generated by the classifier for a correct classification should be high (≈ 1) and for an

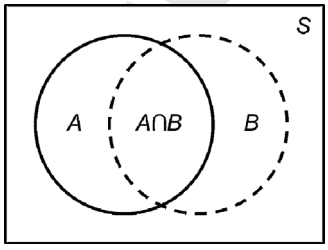


Fig. 4. Venn diagram demonstrating fusion of models.

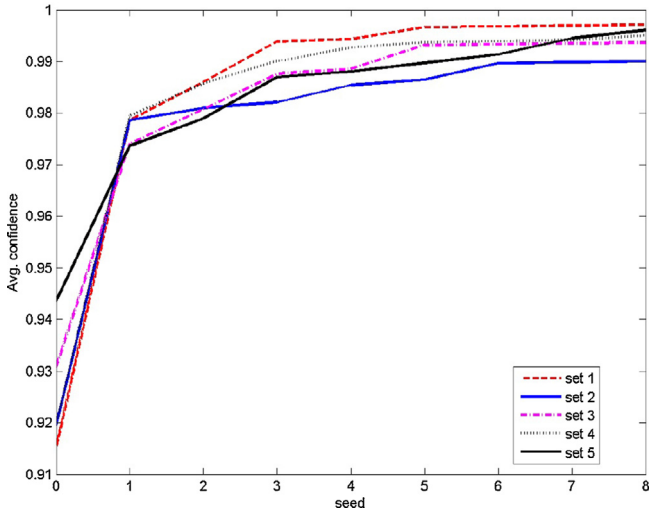


Fig. 5. Variation in average confidence value of hybrid model against seed value (i).

Table 6
Variation in the number of correctly classified test cases between CNN classifiers considered in fusion for UCF50 dataset set 2 (in %).

Set 2	CNN_0	CNN_1	CNN_2	CNN_3	CNN_4	CNN_5	CNN_6	CNN_7	CNN_8	CNN'_0	CNN'_1	CNN'_2	CNN'_3	CNN'_4	CNN'_5	CNN'_6	CNN'_7	CNN'_8
CNN_0	0	7	5	9	10	4	10	5	7	11	9	8	4	8	4	4	7	5
CNN_1	5	0	5	8	9	4	7	6	5	8	8	7	3	7	4	4	6	5
CNN_2	8	9	0	12	12	3	12	7	10	12	8	8	3	7	3	4	9	8
CNN_3	6	6	6	0	6	4	6	5	5	8	8	8	3	8	3	4	5	5
CNN_4	6	6	6	5	0	4	4	5	3	8	9	9	4	8	3	5	5	6
CNN_5	8	10	5	11	12	0	11	6	9	12	9	9	4	8	2	5	9	8
CNN_6	6	5	6	6	4	3	0	5	2	7	9	8	4	8	3	5	4	6
CNN_7	6	7	4	9	9	2	9	0	6	8	9	9	4	8	2	4	7	6
CNN_8	7	6	6	7	6	4	5	5	0	9	9	9	4	8	3	5	6	6
CNN'_0	6	5	5	6	6	3	6	3	5	0	9	8	4	8	2	5	6	5
CNN'_1	9	10	6	12	13	5	13	9	11	14	0	3	1	4	4	2	11	9
CNN'_2	9	9	6	12	13	5	13	9	11	14	3	0	1	3	4	2	10	8
CNN'_3	10	10	6	12	13	5	13	9	10	14	6	6	0	6	5	3	11	9
CNN'_4	9	10	6	12	13	5	13	9	11	14	5	4	2	0	4	2	11	8
CNN'_5	10	11	6	12	12	4	13	7	10	13	10	9	5	9	0	6	10	9
CNN'_6	10	10	7	13	14	6	14	9	11	15	7	7	3	6	5	0	11	9
CNN'_7	6	7	6	7	8	4	7	6	6	10	9	9	5	8	3	5	0	5
CNN'_8	5	7	6	9	9	4	9	5	7	10	8	8	3	7	3	4	6	0

Table 7
Variation in the number of correctly classified test cases between CNN classifiers considered in fusion for UCF50 dataset set 3 (in %).

Set 3	CNN_0	CNN_1	CNN_2	CNN_3	CNN_4	CNN_5	CNN_6	CNN_7	CNN_8	CNN'_0	CNN'_1	CNN'_2	CNN'_3	CNN'_4	CNN'_5	CNN'_6	CNN'_7	CNN'_8
CNN_0	0	11	11	5	10	4	9	5	7	9	6	7	7	87	5	3	6	6
CNN_1	6	0	8	6	7	4	5	4	7	7	7	7	7	81	5	5	5	5
CNN_2	6	8	0	7	9	4	7	4	5	11	6	6	7	82	4	5	5	6
CNN_3	4	11	11	0	10	3	9	4	7	9	6	7	7	86	4	2	5	5
CNN_4	5	7	8	6	0	4	3	3	5	9	6	7	7	81	4	4	4	4
CNN_5	6	12	11	7	12	0	10	5	7	10	6	6	7	90	3	5	7	6
CNN_6	6	7	9	6	6	4	0	3	5	8	7	7	7	83	4	5	4	4
CNN_7	7	11	11	6	11	4	9	0	7	10	7	7	7	89	4	5	6	5
CNN_8	5	11	8	6	9	3	7	4	0	10	6	7	7	86	4	4	5	6
CNN'_0	5	9	12	6	10	3	7	4	7	0	6	7	7	83	5	5	5	5
CNN'_1	8	13	12	7	13	4	11	6	9	10	0	4	5	88	5	6	8	8
CNN'_2	8	13	11	8	13	4	11	6	9	12	4	0	2	87	5	6	8	8
CNN'_3	8	14	13	9	13	5	12	7	9	12	5	3	0	88	6	7	9	8
CNN'_4	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0
CNN'_5	9	14	13	9	13	4	11	6	9	12	8	8	8	90	0	7	8	8
CNN'_6	5	12	11	5	11	4	9	4	7	10	6	7	7	88	5	0	6	6
CNN'_7	6	10	10	6	9	5	8	4	7	9	7	7	7	87	5	5	0	3
CNN'_8	6	10	11	6	10	4	8	4	7	9	7	7	7	87	5	5	3	0

Please cite this article in press as: E.P. Ijjina, C.K. Mohan, Hybrid deep neural network model for human action recognition, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.08.025

Table 8
Variation in the number of correctly classified test cases between CNN classifiers considered in fusion for UCF50 dataset set 4 (in %).

Set 4	CNN_0	CNN_1	CNN_2	CNN_3	CNN_4	CNN_5	CNN_6	CNN_7	CNN_8	CNN'_0	CNN'_1	CNN'_2	CNN'_3	CNN'_4	CNN'_5	CNN'_6	CNN'_7	CNN'_8
CNN_0	0	67	5	16	8	2	8	4	14	10	7	7	8	8	6	5	9	6
CNN_1	2	0	1	2	1	1	2	1	2	2	4	3	2	1	1	2	1	3
CNN_2	6	68	0	18	9	3	9	5	15	10	7	8	10	7	6	5	9	6
CNN_3	4	56	5	0	8	2	8	4	11	9	7	7	7	7	6	4	7	5
CNN_4	6	64	5	18	0	3	8	4	11	8	7	8	5	8	7	5	6	6
CNN_5	7	72	7	19	10	0	12	6	18	13	8	8	12	8	6	5	11	6
CNN_6	3	63	4	16	5	3	0	4	9	4	7	7	6	6	7	5	3	5
CNN_7	5	67	5	17	8	2	10	0	14	10	7	7	10	7	6	4	10	5
CNN_8	3	57	3	12	3	2	3	3	0	3	7	7	4	5	6	4	2	5
CNN'_0	4	62	4	16	4	3	3	4	7	0	7	7	5	6	6	5	3	5
CNN'_1	8	71	7	20	11	4	12	7	19	14	0	3	12	10	7	3	12	3
CNN'_2	8	69	7	19	10	4	12	7	18	14	3	0	12	9	6	2	12	3
CNN'_3	4	63	5	15	3	3	7	4	10	7	6	7	0	8	7	5	5	6
CNN'_4	7	65	5	17	8	2	9	5	14	10	7	7	11	0	5	5	9	6
CNN'_5	8	68	7	19	10	3	13	6	18	13	7	7	13	8	0	5	12	5
CNN'_6	8	71	7	20	11	4	13	7	18	14	5	5	13	10	6	0	12	4
CNN'_7	5	63	4	16	5	3	4	6	9	5	7	8	6	7	7	5	0	6
CNN'_8	8	71	8	20	11	4	12	7	18	13	5	5	12	10	6	3	12	0

Table 9
Variation in the number of correctly classified test cases between CNN classifiers considered in fusion for UCF50 dataset set 5 (in %).

Set 5	CNN_0	CNN_1	CNN_2	CNN_3	CNN_4	CNN_5	CNN_6	CNN_7	CNN_8	CNN'_0	CNN'_1	CNN'_2	CNN'_3	CNN'_4	CNN'_5	CNN'_6	CNN'_7	CNN'_8
CNN_0	0	7	9	7	12	5	5	13	9	7	3	6	4	6	86	88	65	4
CNN_1	6	0	7	8	11	4	8	16	11	6	3	5	3	4	86	87	66	3
CNN_2	7	5	0	9	8	5	9	17	12	6	3	5	3	4	84	86	64	3
CNN_3	5	6	9	0	11	5	7	14	9	7	3	6	4	6	84	86	63	4
CNN_4	6	5	4	7	0	4	8	15	10	5	3	5	3	3	80	82	60	4
CNN_5	8	7	9	9	12	0	10	17	13	8	4	6	4	6	89	90	68	4
CNN_6	3	6	9	6	12	5	0	12	7	7	3	6	4	6	83	85	63	4
CNN_7	3	7	9	6	12	5	5	0	6	8	3	6	4	7	76	78	58	4
CNN_8	4	6	8	6	11	5	4	10	0	6	3	6	3	6	81	82	62	4
CNN'_0	6	5	6	8	10	4	8	16	10	0	1	3	3	2	85	86	64	2
CNN'_1	7	8	9	10	13	6	10	17	13	7	0	5	4	5	90	92	69	3
CNN'_2	6	6	8	9	12	4	9	17	11	4	1	0	3	3	86	88	65	3
CNN'_3	8	8	9	10	13	6	11	18	13	8	3	6	0	6	90	91	69	4
CNN'_4	7	6	7	9	10	5	10	18	13	5	2	4	3	0	87	89	67	4
CNN'_5	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	4	0	0
CNN'_6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
CNN'_7	1	2	2	1	1	1	1	3	3	1	1	1	1	1	22	24	0	1
CNN'_8	8	8	10	11	14	6	11	18	14	8	3	7	5	7	91	92	70	0

Please cite this article in press as: E.P. Ijjina, C.K. Mohan, Hybrid deep neural network model for human action recognition, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.08.025

Table 10
Average confidence value of the outputs of CNN classifiers for correct and incorrect classification on the 5 sets of UCF50 dataset.

Model	Set 1		Set 2		Set 3		Set 4		Set 5	
	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
CNN_0	0.991755	0.006445	0.992068	0.006301	0.992464	0.006449	0.992071	0.005492	0.993101	0.005019
CNN_1	0.992244	0.006305	0.991472	0.005521	0.993078	0.004896	0.969150	0.001880	0.992983	0.007011
CNN_2	0.992205	0.005604	0.992102	0.005607	0.992772	0.004784	0.991919	0.005764	0.992396	0.006240
CNN_3	0.992187	0.005110	0.992407	0.005374	0.992124	0.006449	0.992111	0.005028	0.993000	0.004643
CNN_4	0.991863	0.006414	0.992169	0.004349	0.992799	0.005263	0.992613	0.004656	0.993971	0.004318
CNN_5	0.992642	0.005903	0.992441	0.005936	0.992834	0.007039	0.993045	0.005213	0.993259	0.005580
CNN_6	0.992861	0.005082	0.991380	0.005540	0.992730	0.005773	0.992744	0.004551	0.994054	0.004492
CNN_7	0.992727	0.005106	0.992264	0.006024	0.992844	0.005710	0.993036	0.004874	0.993314	0.004493
CNN_8	0.992889	0.005075	0.991708	0.005725	0.992510	0.005838	0.992008	0.005027	0.992816	0.004430
CNN'_0	0.992497	0.006210	0.991806	0.005479	0.993412	0.004569	0.992367	0.004778	0.992373	0.007551
CNN'_1	0.992490	0.008148	0.986333	0.022612	0.992328	0.007232	0.992570	0.008507	0.992608	0.006640
CNN'_2	0.991742	0.014028	0.992329	0.009129	0.992566	0.008016	0.992389	0.007318	0.993095	0.007210
CNN'_3	0.991628	0.006536	0.991503	0.010240	0.992836	0.007417	0.992572	0.004699	0.992672	0.007332
CNN'_4	0.991923	0.008534	0.990999	0.012043	0.000000	0.000000	0.992452	0.005571	0.994054	0.006829
CNN'_5	0.000000	0.000000	0.991815	0.006222	0.992300	0.006458	0.992070	0.005754	0.000000	0.000000
CNN'_6	0.000000	0.000000	0.992290	0.007369	0.992714	0.006077	0.992076	0.007471	0.000000	0.000000
CNN'_7	0.991977	0.005676	0.991757	0.005797	0.993386	0.004892	0.992848	0.004758	0.994249	0.001019
CNN'_8	0.990216	0.017193	0.992103	0.006182	0.993249	0.005266	0.990255	0.011173	0.992016	0.007502

incorrect classification should be low (≈0), thereby ensuring the selection of correct class label due to fusion across models. The following sections analyze the results in previous section to verify if these two conditions are satisfied.

3.2. Pairwise variation of models

To analyze the variation between models generated in Section 3, we compute the pairwise difference across models by considering the model along vertical axis as CNN_A and along horizontal axis as CNN_B. The percentage of observations correctly classified by CNN_A and not by CNN_B is computed as $(\frac{A-B}{S}) \times 100$, that is stored in Ath row Bth column of the table. The variation of models for the 5 sets in UCF50 dataset rounded to the nearest integer is shown in Tables 5–9.

From the values in these tables, it can be observed that model diversity is achieved between the models. A high diversity is observed when the classification error of CNN_B is high. For example, when CNN'_4 in set 3 is considered as CNN_B, the pairwise variation with other models is more than 50%. This confirms the generation and use of classifiers with diversity in the proposed fusion model. This satisfies the *diversity of classifiers* condition for designing an effective classifier. The next section analyzes the confidence values of the classifiers generated in Section 3.

3.3. Classification confidence of CNN classifiers

As explained in the previous sections, the proposed hybrid model uses CNN classifiers with binary decoded outputs. If the outputs of the trained classifier are ideal i.e., a value of 1 for correct classification and a 0 for incorrect classification, the second condition for designing an effective classifier will be satisfied. The average confidence values for correct and incorrect classification for the CNN classifiers generated in Section 3 for the 5 sets in UCF50 dataset rounded to 6 digits after decimal point are shown in Table 10.

From Table 10, it can be observed that for most of the models the average confidence value for correct prediction is ≈1 and for incorrect prediction is ≈0. Even though some of the models have 0 confidence value for correct prediction (like CNN'_5 and CNN'_6 for set 1 and 5), the use of maximum value as fusion function ensures that correct class labels are selected during fusion. The next section analyzes the effect of classifier fusion on the average confidence value of prediction.

3.4. Fusion across CNN classifiers

The proposed fusion model, as explained in Section 2.3, uses maximum value of classifier outputs as fusion function. Therefore, the average confidence value associated with correct predictions should increase with the number of models included in fusion. The change in average confidence of the proposed model for the 5 sets in UCF50 datasets with the number of models included in fusion i.e., seed value (i) is shown in Fig. 5.

From Fig. 5, it can be observed that the average confidence value increases with increase in number of models considered in fusion, thereby suggesting the selection of correct class labels over models. The next section analyzes the impact of variation of models and high confidence in correct classification on the overall accuracy of the proposed hybrid model.

Table 11
Number of test-cases in UCF50 dataset for 5-fold cross validation.

Split #	# of test cases (p)	Padding (1350-p)
1	1345	5
2	1320	30
3	1325	25
4	1315	35
5	1312	38

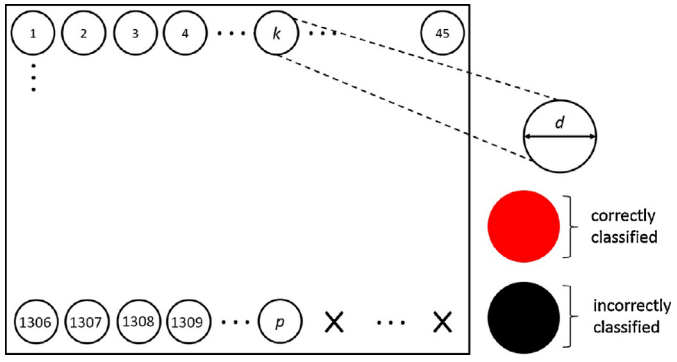
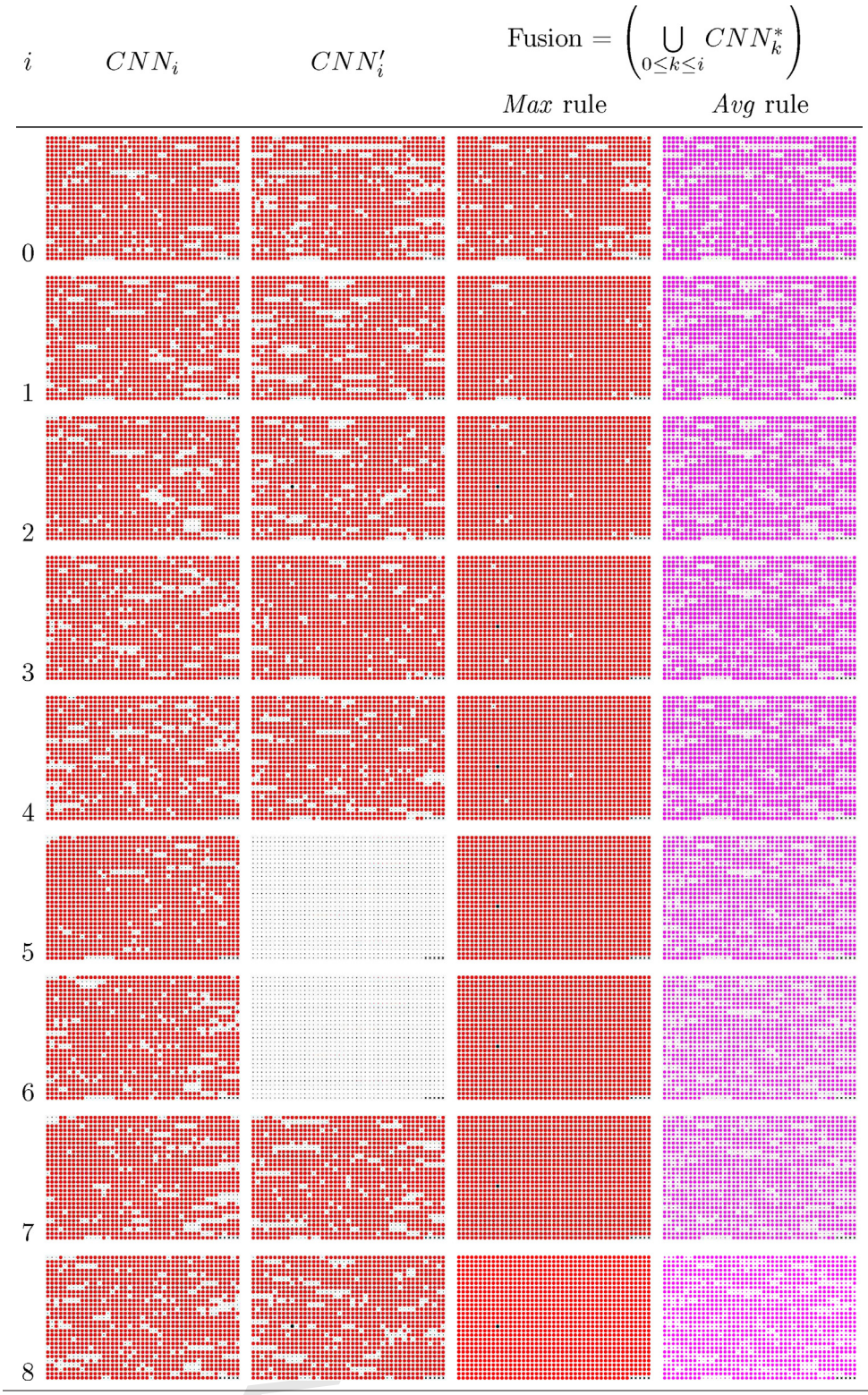


Fig. 6. Representation used to visualize correctness and confidence of recognition by a classifier. Best viewed in color.

Table 12
Visualization of outputs of classifiers due to fusion for UCF50 dataset split 1. Best viewed in color.

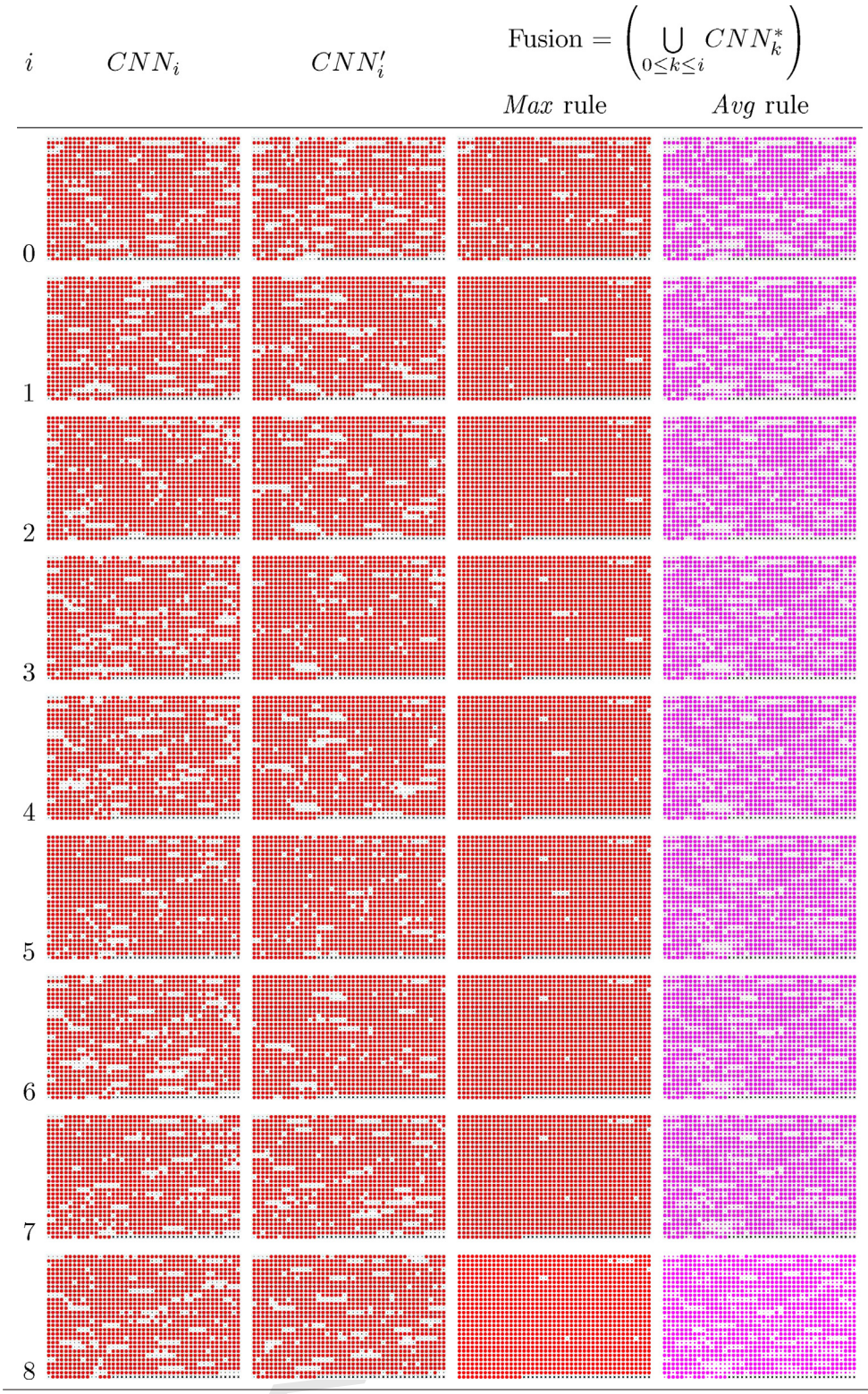


3.5. Analysis of the proposed hybrid model

In this section, we analyze how the number of misclassified instances change with increase in number of models considered in the proposed fusion model. For effective visualization of misclassified observations, we consider the representation shown in Fig. 6, where a 30×45 matrix of circles is used to represent the output of a classifier. By considering row major ordering of elements, the circle at kth location corresponds to the output of classifier for the kth test instance. If the test instance is classified correctly, the circle will be in red (or) pink color, otherwise it will be in black color. The diameter d of the circle is proportional to the confidence value generated

Please cite this article in press as: E.P. Ijjina, C.K. Mohan, Hybrid deep neural network model for human action recognition, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.08.025

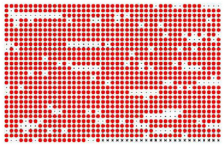
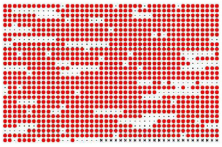
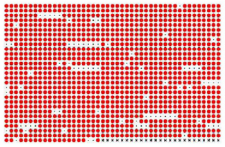
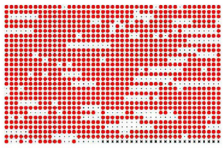
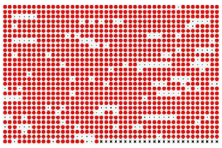
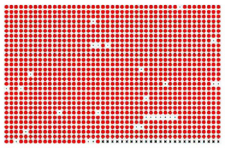
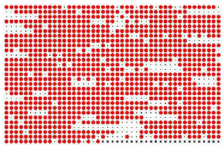
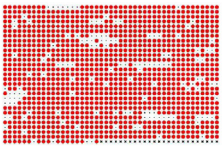
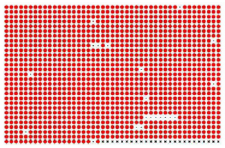
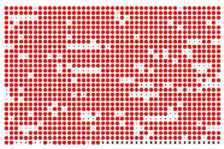
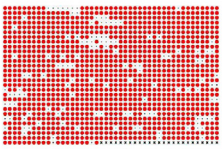
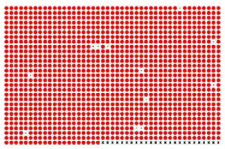
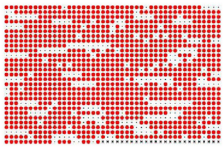
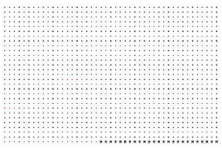
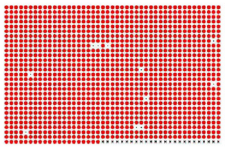
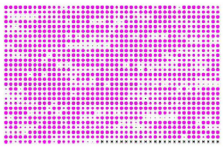
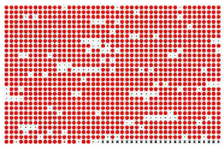
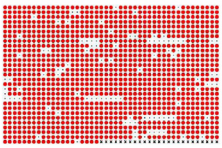
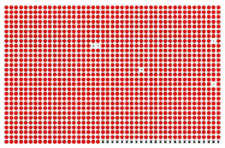
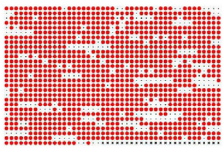
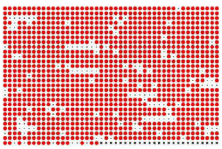
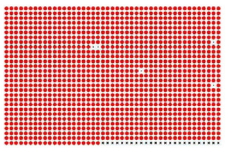
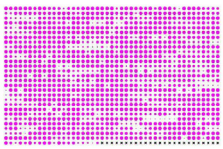
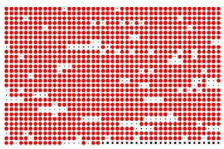
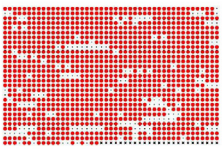
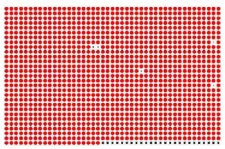
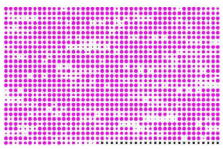
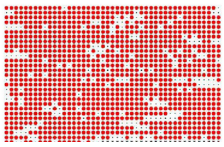
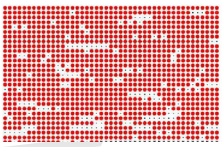
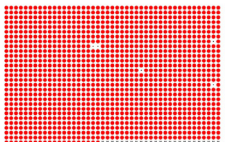
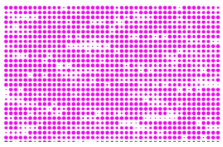
Table 13
Visualization of outputs of classifiers due to fusion for UCF50 dataset split 2. Best viewed in color.



by the fusion model for the test instance. In this representation, the color of the circle will be red if *Max*-rule is used as fusion function and pink if *Avg*-rule is used as fusion function. In case there are p ($1350 = 30 \times 45$) test cases, the last (1350-p) locations in this representation will be marked by cross symbols. When *Max*-rule is used as fusion function, the color and diameter of the circle depicts the correctness and confidence, respectively in recognizing a test instance. There by, the fusion across models is same as selecting the biggest circle at each location across outputs of classifiers. From the average confidence values of correct and incorrect prediction given

Please cite this article in press as: E.P. Ijjina, C.K. Mohan, Hybrid deep neural network model for human action recognition, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.08.025

Table 14
Visualization of outputs of classifiers due to fusion for UCF50 dataset split 3. Best viewed in color.

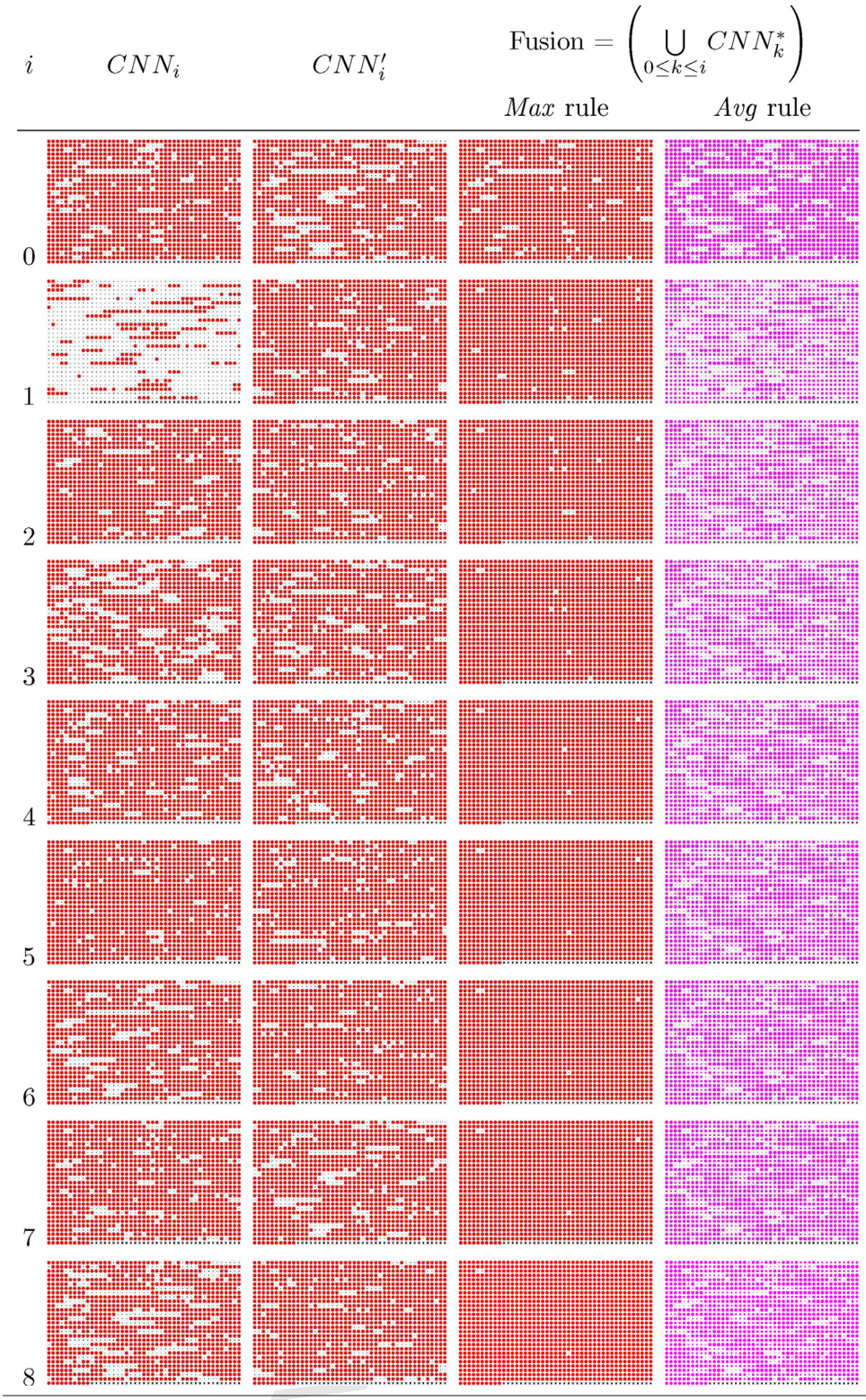
i	CNN_i	CNN'_i	Fusion $= \left(\bigcup_{0 \le k \le i} CNN^*_k \right)$	
			Max rule	*Avg* rule
0				
1				
2				
3				
4				
5				
6				
7				
8				

in Table 10, the red circles with average confidence value ≈1 will have a bigger diameter than black circles with average confidence value ≈0 (as diameter of a circle is proportional to the confidence value generated by the classifier for its test instance). As a result, fusion across models will result in the elimination (minimization) of black circles and selection of bigger red circles, thereby reducing misclassification.

The number of test instances in the 5-splits of UCF50 dataset is given in Table 11. It can be observed that for split 1 with 1345 (p) test cases, the last 5 (1350-p) locations will be marked by cross

Please cite this article in press as: E.P. Ijjina, C.K. Mohan, Hybrid deep neural network model for human action recognition, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.08.025

Table 15
Visualization of outputs of classifiers due to fusion for UCF50 dataset split 4. Best viewed in color.

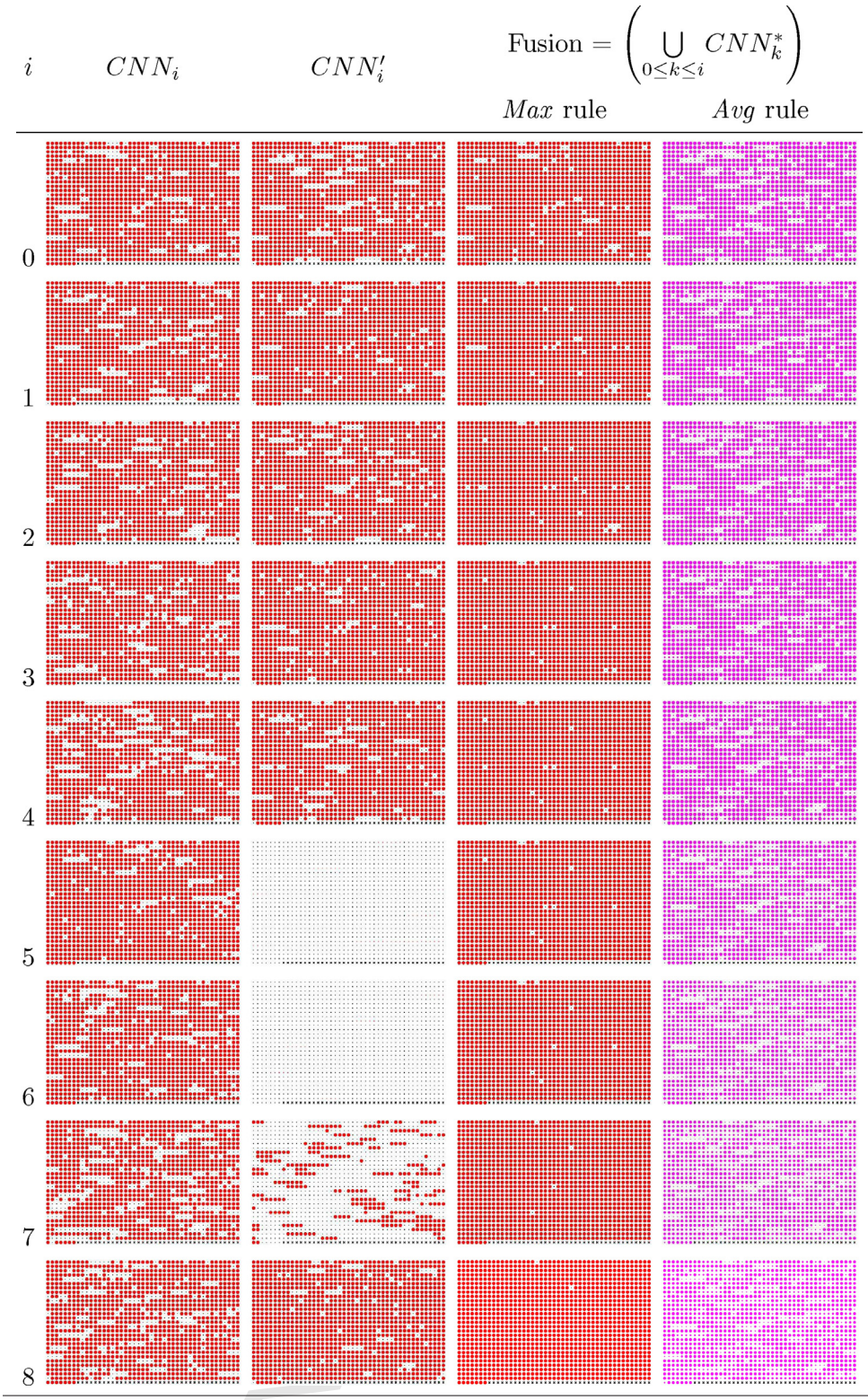


symbols. Similarly, as the number of test cases in each split (p) is less than 1350, the last (1350-p) locations will have cross symbols for quick identification.

The visualization of outputs of classifiers for split 1 is shown in Table 12. Similarly, the visualization of outputs of classifiers for split 2, 3, 4 and 5 are shown in Tables 13–16, respectively. The visualization of misclassified cases shown in Tables 12–16 correspond to the number of misclassified cases by the proposed hybrid model for *Min* and *Avg* rules given in Table 2. The improvement in classification accuracy with increase in the number of models considered in fusion can be observed visually by the increase in the number of red and pink circles as we move from top to bottom

Please cite this article in press as: E.P. Ijjina, C.K. Mohan, Hybrid deep neural network model for human action recognition, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.08.025

Table 16
Visualization of outputs of classifiers due to fusion for UCF50 dataset split 5. Best viewed in color.



in the fusion column of Tables 12–16. From the representation of fusion model outputs for *Max* and *Avg* rules, it can be observed that the same observations are misclassified by both the rules. There is one exception to this observation, where an observation is misclassified with high confidence by *Max* rule is correctly classified by using *Avg* rule as fusion function in split 1. This may be due to the normalization effect caused by averaging the outputs across models, there by reducing the high peak caused by one model.

As the proposed fusion model has almost same performance for *Max* and *Avg* rules, we analyze the difference (gap) in confidence value between the top two class labels assigned to the test cases. The histogram of this difference in confidence values for *Max* and

Please cite this article in press as: E.P. Ijjina, C.K. Mohan, Hybrid deep neural network model for human action recognition, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.08.025

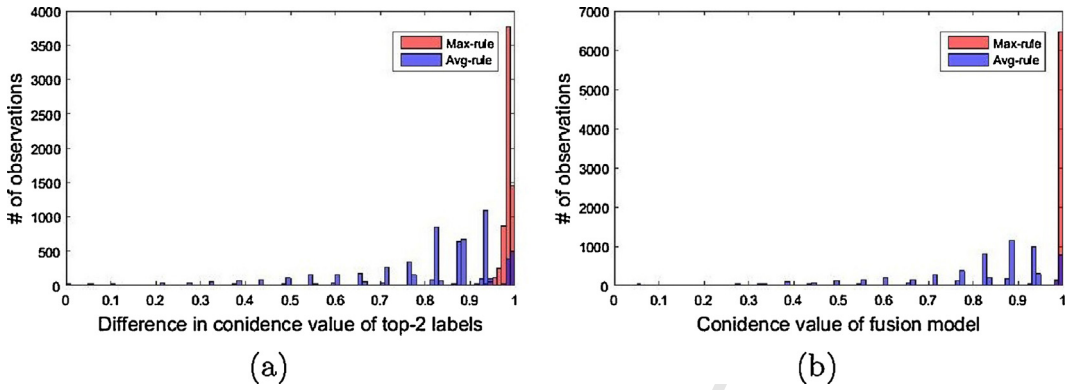


Fig. 7. Comparison of confidence values of fusion model for *Max* and *Avg* rules: (a) The histogram of difference in confidence value between top-2 labels of fusion model for *Max* and *Avg* fusion rules and (b) the histogram of confidence values of fusion model for *Max* and *Avg* fusion rules.

Avg rules is shown in Fig. 7(a). The histogram of confidence value of fusion model for the two fusion rules is shown in Fig. 7(b).

From the results shown in Fig. 7, it can be observed that the outputs of the fusion model will have high confidence and maximum difference in confidence value with the next class label when *Max*-rule is used as the fusion function. Hence, the fusion model will be noise tolerant when *Max*-rule is used as fusion function even though it mis-classifies one observation more than *Avg*-rule.

4. Conclusions

In this work, we proposed a hybrid deep neural network model using fusion of CNN classifier with binary decoded outputs. The high confidence of classifiers for correct prediction and the variation of models (achieved though manipulation of input features and initialization of models) are leveraged to design an effective classifier fusion model. Fusion using maximum value ensures the selection of correct class label making this an effective classifier model. This model achieved a near accurate recognition of 99.68% on UCF50 dataset. The future work will consider other spatio-temporal representations of videos similar to action bank features.

References

[1] D.H. Wolpert, The supervised learning no-free-lunch theorems, in: Proceedings of the 6th Online World Conference on Soft Computing in Industrial Applications, 2001, pp. 25–42.
[2] G.L. Marcialis, F. Roli, Fusion of Face Recognition Algorithms for Video-Based Surveillance Systems, Kluwer Academic, 2003.
[3] R. Polikar, Ensemble learning, Scholarpedia 4 (1) (2009) 27–76.
[4] R. Clemen, Combining forecasts: a review and annotated bibliography, Int. J. Forecast. 5 (4) (1989) 559–583.
[5] K. Tumer, J. Ghosh, Analysis of decision boundaries in linearly combined neural classifiers, Pattern Recognit. 29 (2) (1996) 341–348.
[6] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.
[7] M. Termenon, M. Graña, A two stage sequential ensemble applied to the classification of Alzheimer's disease based on MRI features, Neural Process. Lett. 35 (1) (2012) 1–12.
[8] P.L. Bartlett, M.H. Wegkamp, Classification with a reject option using a hinge loss, J. Mach. Learn. Res. 9 (2008) 1823–1840.
[9] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[10] G. Zenobi, P. Cunningham, Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error, in: Proceedings of the12th European Conference on Machine Learning, Springer Verlag, 2001, pp. 576–587.
[11] A. Krogh, J. Vedelsby, Neural network ensembles, cross validation, and active learning, in: Proceedings of the Advances in Neural Information Processing Systems (NIPS), MIT Press, 1995, pp. 231–238.
[12] T.G. Dietterich, Ensemble methods in machine learning, in: J. Kittler, F. Roli (Eds.), Multiple Classifier Systems, LNCS, vol. 1857, Springer, 2001, pp. 1–15.
[13] G. Brown, J. Wyatt, R. Harris, X. Yao, Diversity creation methods: a survey and categorisation, J. Inf. Fusion 6 (1) (2005) 5–20.
[14] Y. Bi, The impact of diversity on the accuracy of evidential classifier ensembles, Int. J. Approx. Reason. 53 (4) (2012) 584–607.
[15] G. Giacinto, F. Roli, G. Fumera, Design of effective multiple classifier systems by clustering of classifiers, in: Proceedings of the 15th International Conference on Pattern Recognition, vol. 2, IEEE, 2000, pp. 160–163.
[16] Y.H. Hu, Handbook of Neural Network Signal Processing, 1st ed., CRC Press, Inc., Boca Raton, FL, USA, 2000.
[17] M. van Erp, L. Vuurpijl, L. Schomaker, An overview and comparison of voting methods for pattern recognition, in: Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR'02), IWFHR'02, IEEE Computer Society, 2002, pp. 195–200.
[18] T.K. Ho, J.J. Hull, S.N. Srihari, Decision combination in multiple classifier systems, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1) (1994) 66–75.
[19] L.A. Alexandre, A.C. Campilho, M.S. Kamel, Combining independent and unbiased classifiers using weighted average, in: Proceedings of the 15th International Conference on Pattern Recognition, ICPR'00, Barcelona, Spain, September 3–8, 2000, 2000, pp. 2495–2498.
[20] B. Biggio, G. Fumera, F. Roli, Bayesian analysis of linear combiners, in: Proceedings of the 7th International Conference on Multiple Classifier Systems, MCS'07, Springer-Verlag, 2007, pp. 292–301.
[21] J. Kittler, F.M. Alkoot, Sum versus vote fusion in multiple classifier systems, IEEE Trans. Pattern Anal. Mach. Intell. 25 (1) (2003) 110–115.
[22] D.W. Opitz, J.W. Shavlik, Generating accurate and diverse members of a neural-network ensemble, in: Proceedings of the Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, November 27–30, 1995, 1995, pp. 535–541.
[23] S. Hashem, Optimal linear combinations of neural networks, Neural Netw. 10 (4) (1997) 599–614.
[24] H. Inoue, H. Narihisa, Optimizing a multiple classifier system, in: Proceedings of the 7th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence, PRICAI'02, Springer-Verlag, 2002, pp. 285–294.
[25] M. Wozniak, Experiments with trained and untrained fusers, in: E. Corchado, J.M. Corchado, A. Abraham (Eds.), Innovations in Hybrid Intelligent Systems, vol. 44 of Advances in Soft Computing, Springer, 2008, pp. 144–150.
[26] Z. Zheng, B. Padmanabhan, Constructing ensembles from data envelopment analysis, INFORMS J. Comput. 19 (4) (2007) 486–496.
[27] J. Kittler, M. Hatef, R. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 226–239.
[28] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), 2008.
[29] N. Dalal, B. Triggs, C. Schmid, Human detection using oriented histograms of flow and appearance, in: Proceedings of the 9th European Conference on Computer Vision – Volume Part II, ECCV 06, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 428–441.
[30] O. Kliper-Gross, Y. Gurovich, T. Hassner, L. Wolf, Motion interchange patterns for action recognition in unconstrained videos, in: Proceedings of the 12th European Conference on Computer Vision – Volume Part VI, ECCV'12, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 256–269.
[31] N. Ikizler-Cinbis, S. Sclaroff, Object, scene and actions: combining multiple features for human action recognition, in: K. Daniilidis, P. Maragos, N. Paragios (Eds.), Computer Vision ECCV 2010, vol. 631 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010, pp. 494–507.
[32] F.C. Heilbron, A. Thabet, J.C. Niebles, B. Ghanem, Camera motion and surrounding scene appearance as context for action recognition, in: ACCV 2014, Springer, 2014, pp. 583–597.
[33] S. Althloothi, M.H. Mahoor, X. Zhang, R.M. Voyles, Human activity recognition using multi-features and multiple kernel learning, Pattern Recognit. 47 (5) (2014) 1800–1812.
[34] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014, CoRR abs/1409.1556.

Please cite this article in press as: E.P. Ijjina, C.K. Mohan, Hybrid deep neural network model for human action recognition, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.08.025

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, Int. J. Comput. Vis. (IJCV) (2015).
[36] S.E. Kahou, C. Pal, X. Bouthillier, P. Froumenty, C. Gülçehre, R. Memisevic, P. Vincent, A. Courville, Y. Bengio, R.C. Ferrari, M. Mirza, S. Jean, P.-L. Carrier, Y. Dauphin, N. Boulanger-Lewandowski, A. Aggarwal, J. Zumer, P. Lamblin, J.-P. Raymond, G. Desjardins, R. Pascanu, D. Warde-Farley, A. Torabi, A. Sharma, E. Bengio, M. Côté, K.R. Konda, Z. Wu, Combining modality specific deep neural networks for emotion recognition in video, in: Proceedings of the 15th ACM on International Conference on Multimodal Interaction, ICMI'13, ACM, New York, NY, USA, 2013, pp. 543–550.
[37] A. Dhall, R. Goecke, J. Joshi, M. Wagner, T. Gedeon, Emotion recognition in the wild challenge (EmotiW) challenge and workshop summary, in: Proceedings of the 15th ACM on International Conference on Multimodal Interaction, ACM, 2013, pp. 371–372.
[38] M. Liu, R. Wang, S. Li, S. Shan, Z. Huang, X. Chen, Combining multiple kernel methods on Riemannian manifold for emotion recognition in the wild, in: Proceedings of the 16th International Conference on Multimodal Interaction, ICMI'14, ACM, New York, NY, USA, 2014, pp. 494–501.
[39] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
[40] E.P. Ijjina, C.K. Mohan, Human action recognition based on recognition of linear patterns in action bank features using convolutional neural networks, in: Proc. of the 13th International Conference on Machine Learning and Applications (ICMLA), IEEE, 2014, pp. 178–182.
[41] S. Sadanand, J.J. Corso, Action bank: a high-level representation of activity in video, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 1234–1241.
[42] N. Ballas, Y. Yang, Z.-Z. Lan, B. Delezoide, F. Preteux, A. Hauptmann, Space–time robust representation for action recognition, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2013.
[43] NVIDIA cuDNN, GPU Accelerated Deep Learning, 2014 https://developer.nvidia.com/cudnn (accessed: 08.08.15.
[44] O. Kliper-Gross, Y. Gurovich, T. Hassner, L. Wolf, Motion interchange patterns for action recognition in unconstrained videos, in: Proceedings of the 12th European Conference on Computer Vision – Volume Part VI, ECCV'12, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 256–269.
[45] F. Shi, E. Petriu, R. Laganiere, Sampling strategies for real-time action recognition, in: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 2595–2602.
[46] L. Wang, Y. Qiao, X. Tang, Motionlets: mid-level 3D parts for human motion recognition, in: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 2674–2681.
[47] H. Wang, A. Klaser, C. Schmid, C.-L. Liu, Action recognition by dense trajectories, in: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR'11, IEEE Computer Society, Washington, DC, USA, 2011, pp. 3169–3176.
[48] Q. Zhou, G. Wang, K. Jia, Q. Zhao, Learning to share latent tasks for action recognition, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2013.

Please cite this article in press as: E.P. Ijjina, C.K. Mohan, Hybrid deep neural network model for human action recognition, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.08.025