Context-aware Mouse Behaviour Recognition using Hidden Markov Models

Zheheng Jiang, Danny Crookes *Senior Member, IEEE*, Brian Desmond Green, YunFeng Zhao, HaiPing Ma, Ling Li, Shengping Zhang *Member, IEEE*, Dacheng Tao *Fellow, IEEE* and Huiyu Zhou

***Abstract*—Automated recognition of mouse behaviours is crucial in studying psychiatric and neurologic diseases. To achieve this objective, it is very important to analyse temporal dynamics of mouse behaviours. In particular, the change between mouse neighbouring actions is swift in a short period. In this paper, we develop and implement a novel Hidden Markov Model (HMM) algorithm to describe the temporal characteristics of mouse behaviours. In particular, we here propose a hybrid deep learning architecture, where the first unsupervised layer relies on an advanced spatial-temporal segment Fisher Vector (SFV) encoding both visual and contextual features. Subsequent supervised layers based on our segment aggregate network (SAN) are trained to estimate the state dependent observation probabilities of the HMM. The proposed architecture shows the ability to discriminate between visually similar behaviours and results in high recognition rates with the strength of processing imbalanced mouse behaviour datasets. Finally, we evaluate our approach using JHuang's and our own datasets, and the results show that our method outperforms other state-of-the-art approaches.**

***Index Terms*—Mouse behaviours, Hidden Markov Model, spatial-temporal segment, Fisher Vector, segment aggregate network.**

I. Introduction

STUDYING neurobehavioural phenotypes can be of great interest because the first symptom of neurological, psychiatric or neurodegenerative disorders is often identifiable through subtle changes in day-to-day human behaviours (e.g., food intake, sleeping and activity patterns) [1]. For example, the activity/rest cycles of Alzheimer's patients gradually deteriorate in the early stage of the disease. Mouse and rat disease models are a valuable resource in studying these psychiatric and neurologic diseases [2]–[8]. However, the studies need prolonged systematic observation of mice or rats carrying the diseases, e.g. several days or months, which is highly labour intensive and subject to human error and varying interpretations. Furthermore, human observers may fail to detect behavioural events that are either too quick or slow, and miss some events because of dwindling attention span. Automated home-cage systems can facilitate neurobehavioural analysis of mouse phenotypes over a long period of time. The application of such systems to the mouse models of human diseases has provided new insights into the pathophysiology and treatment of these disorders [9]–[11].

Previously developed automated systems [12]–[14] rely mostly on the use of sensor equipment such as infrared beams. Although these systems have demonstrated good performance in monitoring locomotor activities, they cannot be used to study home-cage behaviours such as grooming, hanging and micro-movements. Visual analysis is thus being used to recognise subtle animal behaviours.

In the scientific literature some systems have been described to automatically recognize animal behaviours by visual analysis. For instance, Dankert et al. [15] tracked an insect body using a Gaussian Mixture Model (GMM) and then performed recognition of aggression and courtship behaviours of insects. Unfortunately this system is unsuitable for analysing micro-behaviours such as micro-movements of the head, grooming or rearing. Rousseau et al. [16] may be the first group to report that the detection of specific behaviours was possible. They applied neural network techniques to recognise 9 solitary rat behaviours from the body shape and position of rats, recorded from the side view. However, their method of tracking the nose is not sufficiently developed to draw conclusions concerning its sensitivity and reliability. In 2005, Dollár et al. [17] recognised mouse behaviours by classifying sparse spatio-temporal features. However, they only considered visual features of the interest points (e.g. image gradient) without using contextual information such as the spatial relationship between two interest points. This method can only be used to classify short video clips, each of which contains only one subject behaviour. In 2010 Jhuang et al. [18] extracted image features based on a computational model of motion processing in the human brain [19], followed by classification using a Hidden Markov Model Support Vector Machine (SVMHMM) method. Their method to locate the mouse is dependent on a good background model, which may not be achievable in noisy environments. Burgos-Artizzu et al. [20] designed a system for recognising social behaviours of mice using the top and side views. They applied AdaBoost with spatio-temporal and trajectory features to classify mouse behaviours. As with the method of Dollár et al. [17], this method also ignored the spatio-temporal contextual features. Furthermore, their trajectory features are based on a tracking algorithm which

Z. Jiang and H. Zhou are with School of Informatics, University of Leicester, United Kingdom. E-mail: {zj53;hz143}@leicester.ac.uk.

D. Crookes and YunFeng Zhao are with School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, United Kingdom. E-mail: {d.crookes;y.zhao}@qub.ac.uk.

B. Green is with School of Biological Sciences, Queen's University Belfast, United Kingdom. E-mail: b.green@qub.ac.uk.

Haiping Ma is with the Department of Electrical Engineering, Shaoxing University, Shaoxing, Zhejiang, 312000, China. E-mail: Mahp@usx.edu.cn

L. Li is with University of Kent. E-mail: C.Li@kent.ac.uk

S. Zhang is with School of Computer Science and Technology, Harbin Institute of Technology, China. E-mail: s.zhang@hit.edu.cn.

D. Tao is with School of Information Technologies, The University of Sydney. E-mail: dacheng.tao@sydney.edu.au.

Manuscript received on 19 December, 2017; revised xxxx.

was not detailed in their paper. Recently, CNNs have been considered to classify individual behaviours of mice [21]. The proposed models are designed to study interactive behaviours with objects, so it is not yet powerful enough to recognise fine-grained behaviours of a mouse itself based on the designed features. Kramida et al. [22] presented a mouse behavior classification method using VGG features and a long short-term memory (LSTM) model. Existing neural networks trained for action recognition typically learned from human subjects in natural scenes which are intrinsically different from the rodents in laboratory environments, and it is still a challenge to transfer networks from the human to the rodent domain.

Many mouse actions have pairwise relationships in the temporal domain. For example, it is very unlikely to have a hang or rest action immediately after a drink action. Recently, LSTMs have demonstrated satisfactory performance for modelling sequential data but they are usually suitable for learning long-term dynamics of sequential data, e.g. speech recognition and handwriting recognition with long-distance dependency. In this paper, we use a generative HMM [23] to model the temporal transition of mouse actions, where the parameters are obtained using our proposed SFV+SAN network. In our proposed framework, shown in Fig.1, we treat a video sequence as a set of action clips. Each clip is represented as a set of feature vectors employing our spatial-temporal segment Fisher vectors (SFV), corresponding to an observed variable in the HMM. The main contribution of our proposed approach includes:

1. We propose a novel HMM model learning framework. Its first unsupervised layers of video clips representation relies on our SFV and involves feature encoding of both appearance and contextual features. The subsequent supervised classification layers comprise our advanced segment aggregate network (SAN) trained through back-propagation to support the HMM for inferring the most likely state sequence. Our proposed system is a hybrid learning architecture as it stacks several unsupervised and supervised layers. The motivation behind this hybrid learning architecture is twofold: Firstly, we want to explore a mid-level representation to mine discriminative action parts. Secondly, we model the transformation of adjacent actions in time using a HMM that considers the contextual relationship of mouse behaviours.

2. We introduce a novel interest point detector, based on the Dollár's interest point detector, using frame differencing and Laplacian of Gaussian (LoG) filtering. Inspired by the use of context in local features such as [23], [24], we propose to exploit spatial-temporal context which can characterise the spatial location, pose and temporal changes of a mouse. This is the first attempt to encode contextual features from actions rather than simply concatenate them with the extracted appearance features [18], [20]. Our contextual features are an important feature which characterises both spatial location and temporal changes of a mouse. We retain absolute and relative positions of each interest point and then concatenate them to form the contextual features.

3. Unlike the traditional FV pipelines [25]–[28] that encode and aggregate the local features by sum pooling over the entire video, our SFV performs sum pooling over each subvolume to increase the discrimination ability of the extracted feature.

4. In order to cooperate with SFV, we also propose an advanced SAN network in the supervised layers. This network is employed to aggregate subvolume-level FVs so that our framework is capable of modelling the spatial-temporal structure over the entire video clip (or dataset).

5. We conduct a comprehensive evaluation of the proposed algorithm, and compare our method against several state-of-the-art techniques for mouse behaviour recognition. The proposed architecture results in high recognition rates with the strength of processing imbalanced mouse behaviour datasets.

II. Modelling with HMM

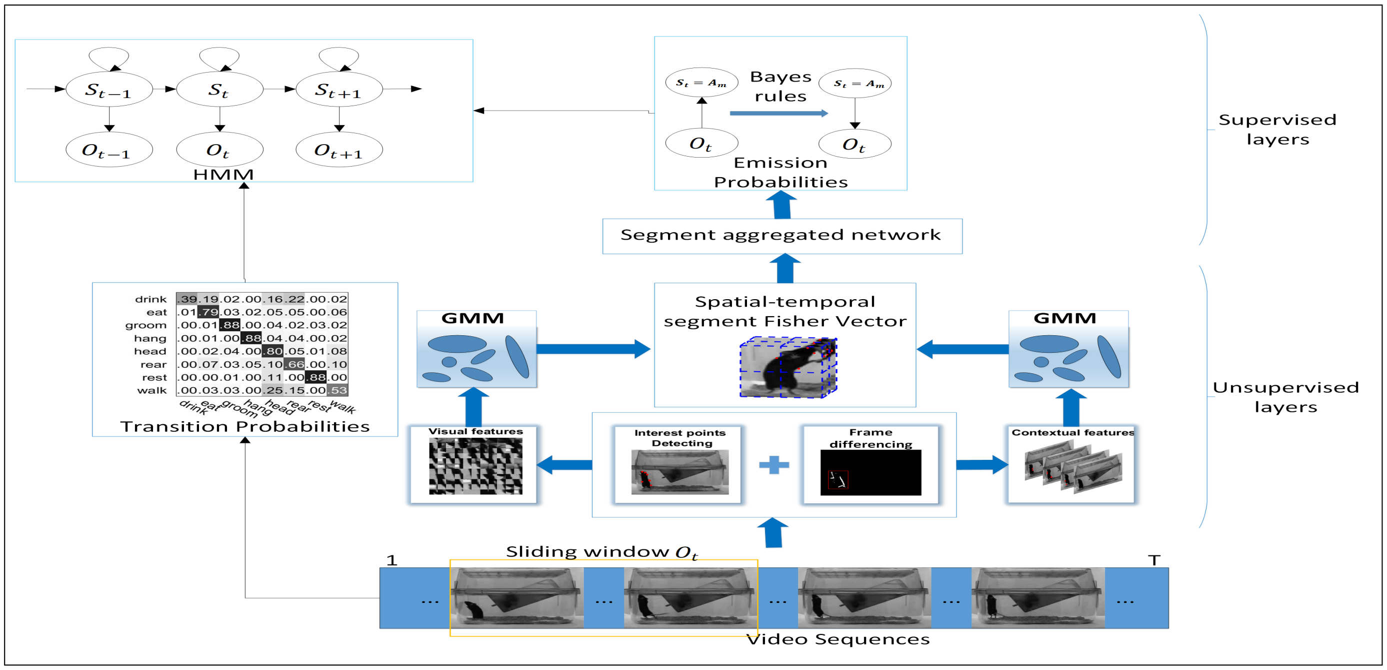
In this section, we introduce how we model mouse behaviours using HMM and estimate the parameters of the proposed infrastructure. Fig. 1 shows the proposed HMM framework.

A. Model set-up

HMM in our approach is used 1) to infer latent or hidden states from the observed sequential data, and 2) to account for the dynamics of the observed sequential data according to the dynamics of the hidden states (see e.g., [29], [30]). Here, we assume that an observation O_t in the observed sequential data $O^* = \{O_1, O_2, ..., O_T\}$ is generated by an underlying and hidden state S_t. The underlying states follow a Markov chain. HMM is a discrete time model where we receive an observation generated by a hidden state at each time instance t. The sequence of the underlying states $S^* = \{S_1, S_2, ..., S_T\}$ takes possible values from a countable finite set $S_t \in A^* = \{A_1, A_2, ..., A_M\}$, where A^* is an action set with M actions. The sequence of the underlying states S^* forms the Markov chain and satisfies the Markov property:$P(S_{t+1}|S_t, S_{t-1}, ..., S_1) = P(S_{t+1}|S_t)$, i.e., the probability of the transition to the next state S_{t+1} only depends on the current state S_t. The transitions between the actions (e.g. walk and rest) are represented by the transition probability matrix, in which element α_{mn} denotes the transition probability from action A_m in the current state S_t to action A_n in the next state S_{t+1}:$\alpha_{mn} = P(S_{t+1} = A_n|S_t = A_m), \sum_j \alpha_{mn} = 1$ and $A_m, A_n \in A^*$.

In our application, suppose there are M actions (without loss of generality, in this paper we look at the example of $M = 8$), each of which corresponds to an underlying state of the HMM. Examples for the transition probabilities between actions and the probabilities of self-transition are illustrated in Fig. 2. In our experiments, we observe that the probability of a self-transition is usually larger than that of a transition between actions. Interestingly, a mouse often switches back and forth between 'walk' and 'rear', or 'walk' and 'head'.

The probability of the current observation O_t is conditioned on the current latent state S_t:$P(O_t|O_{t-1}, ...O_1, S_t, ..., S_1) = P(O_t|S_t)$. In our case, suppose there are T sliding windows partitioned from an entire video, then we have a sequential observation O^* with T elements. Given any of the underlying states $S_t = A_m$, O_t has a probability estimated by our hybrid learning architecture.



	drink	eat	groom	hang	head	rear	rest	walk
drink	.39	.19	.02	.00	.16	.22	.00	.02
eat	.01	.79	.03	.02	.05	.05	.00	.06
groom	.00	.01	.88	.00	.04	.02	.03	.02
hang	.00	.01	.00	.88	.04	.04	.00	.02
head	.00	.02	.04	.00	.80	.05	.01	.08
rear	.00	.07	.03	.05	.10	.66	.00	.10
rest	.00	.00	.01	.00	.11	.00	.88	.00
walk	.00	.03	.03	.00	.25	.15	.00	.53

Fig. 1. Illustration of the proposed hybrid learning framework.

B. Model estimation

In a HMM model, there are three sets of parameters: the initial probabilities of the first states π_m in the sequence of the underlying states S^*, the probability matrix including transition probabilities α_{nm} between hidden states (i.e. actions), and the probability distribution of generating observation O_t (given S_t). Typically the feature vectors of the sequential data O^* are assumed to follow Gaussian Mixture Models (GMMs) [31]–[34]. The Expectation-Maximisation (EM) algorithm is normally used to derive the parameters of GMMs and transition probabilities. However, GMMs of several states may overlap and lack the capacity to discriminate one class from the others. Furthermore, estimating a GMM of high dimensionality requires a large amount of training data and generally limits the dimension of the targeted feature space to avoid this difficulty. To overcome these limitations, people started using standard Artificial Neural Networks instead of Gaussian mixtures for a better discrimination ability [35], [36]. However, these ANN-HMM architectures are not deep enough to identify complex actions.

In contrast to a common HMM paradigm which estimate HMM parameters from only the sequence of observation O^*, our model utilizes the ground truth $Y^* = \{Y_1, Y_2, ..., Y_T\}$ about the sequence of latent actions to directly initialize the initial state and transition probabilities. In particular, we estimate the initial probabilities π_m by $\pi_m = P\left(S_1 = A_m\right)$ with $\sum_m \pi_m = 1$. For estimating the transition probability α_{nm} from action A_n to action A_m, we assume that q_{nm} is the times that action state A_n changes to A_m. Then we have: $\alpha_{nm} = \frac{q_{nm}}{\sum_{m'} q_{nm'}}$. For the estimation the probability distribution of generating observation O_t (given S_t), we propose a hybrid learning architecture consisting of unsupervised and supervised layers to estimate the probability distribution

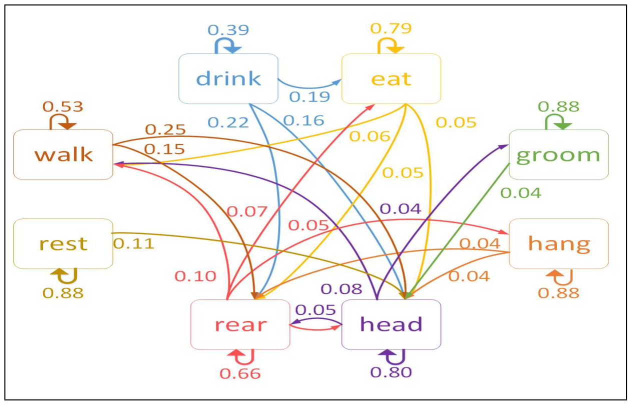


Fig. 2. An example to show a Markov model that depicts the transition probabilities between different actions and the probabilities of keeping the same action (i.e., self-transition). Transition probabilities < 0.04 are omitted to avoid clutter.

of generating observation O_t (i.e. an emission probability of the HMM). The first set of the unsupervised layers involves three layers of interest point detection, local feature extraction and SFV encoding, followed by supervised layers including SAN layers, which consists of several subvolume-level networks and an aggregation layer (see III-B-1 for detail), and a HMM layer. In the last layer of SAN, we use a softmax function to estimate $P\left(S_t = A_m|O_t\right)$ that is transformed to the emission probability of the next HMM layer. The posteriors $P\left(S_t = A_m|O_t\right)$ have to be transformed into emission probabilities using Bayes' rule $P\left(O_t|S_t = A_m\right) \propto \frac{P(S_t=A_m|O_t)}{P(S_t=A_m)}$, where the prior probability $P\left(S_t = A_m\right)$ of action A_m is estimated by: $\frac{u_m}{\sum_{m'} u_{m'}}$, where u_m is the occurrence of action A_m in the training data. For simplicity, here we assume all the observations have a uniform prior probability.

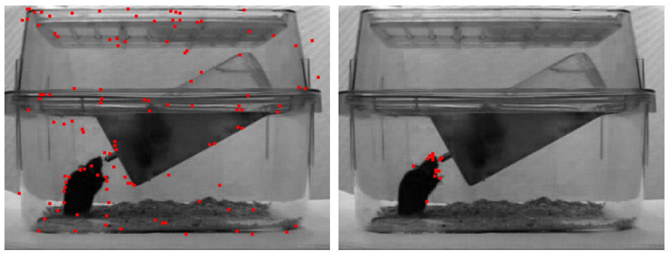


Fig. 3. Comparison between interest points detected using our detector (right) and the Dollár detector (left) under the same illumination.

Given a well-fitted HMM and the observation sequence $O^* = \{O_1, O_2, ..., O_T\}$, our interest is to infer the most likely sequence of action states $S^* = \{S_1, S_2, ..., S_T\}$. In order to find an optimal state sequence S^* over time, we deploy the standard Viterbi algorithm based on the output of SAN. Defining μ_t^m as the probability of the most probable state sequence $S_{1:t}$ with $S_t = A_m \in A^*$ as its final state, the recursion of the Viterbi algorithm [37] can be described as follows:

Initializing:

$$\begin{aligned} \mu_1^m &= P\left(O_t | S_t = A_m\right) \cdot \pi_m \\ \psi_1^m &= 0. \end{aligned} \tag{1}$$

Recursing:

$$\begin{aligned} \mu_t^m &= \max_{1 \le n \le M} \left(P\left(O_t | S_t = A_m\right) \alpha_{nm} \cdot \mu_{t-1}^n\right) \\ \psi_t^n &= \operatorname*{argmax}_{1 \le n \le M} \left(P\left(O_t | S_t = A_m\right) \alpha_{nm} \cdot \mu_{t-1}^n\right). \end{aligned} \tag{2}$$

Terminating:

$$\begin{aligned} m_T &= \operatorname*{argmax}_{1 \le m \le M} \left(\mu_T^m\right), \\ S_T &= A_{m_T}. \end{aligned} \tag{3}$$

Path backtracking:

$$\begin{aligned} m_{t-1} &= \psi_t^{m_t}, \\ S_{t-1} &= A_{m_{t-1}}. \end{aligned} \tag{4}$$

where ψ_t^n is a backpointer which is used to retrieve the final Viterbi path S^* in Eqs. (3) and (4). In section III-B-2, we will introduce how to import the output of SFV to the Viterbi algorithm in further detail.

III. Hybrid learning architecture

As shown in Fig.1, our proposed hybrid learning architecture SFV-SAN aims to estimate the probability distribution of generating observation O_t in the HMM framework. Our framework is constituted of several unsupervised and supervised layers to be detailed as follows.

A. Unsupervised layers

There are three unsupervised layers that sequentially perform interest point detection, local feature extraction and SFV encoding.

1) Points detection layer: Our interest point detector is the improved version of [17]. The traditional Dollárs detector uses solely local information within a small region, and it is prone to false detection under illumination variations. To overcome these shortcomings, we here propose a different interest point detector including two steps: 1) a LoG filter is used instead of a single Gaussian filter [17] for reducing the impact of illumination change, and 2) frame differencing is used to eliminate spurious interest points on the background. This two-step approach facilitates saliency detection in both the temporal and spatial domains and produces a combined filtering response. Therefore, our response function is $R = \left(I\left(x,y,t\right) * g\left(x,y,\sigma\right) * L * h_{ev}\left(t\right)\right)^2 + \left(I\left(x,y,t\right) * g\left(x,y,\sigma\right) * L * h_{od}\left(t\right)\right)^2$, where $I\left(x,y,t\right)$ is the image at time t, $g\left(x,y,\sigma\right)$ is the 2D Gaussian smoothing kernel which is applied only along the spatial dimension, L is the operator of Laplace used on the spatial dimension, and h_{ev} and h_{od} are a quadrature pair of 1-D Gabor filters defined as: $h_{ev}\left(t;\tau,\omega\right) = -\cos\left(2\pi t\omega\right) e^{-t^2/\tau^2}$ and $h_{od}\left(t;\tau,\omega\right) = -\sin\left(2\pi t\omega\right) e^{-t^2/\tau^2}$. With the constraints $\omega = 4/\tau$, σ and τ are two parameters of the spatial and temporal scales, respectively. They are empirically set by $\sigma = 2.5$ and $\tau = 2$. Fig. 3 shows that, under the same illumination, our detector can extract precise interest points on the mouse.

2) Local feature extraction layer.: Like most of the existing action recognition methods [17], [28], [38], [39], we extract visual features from the cuboids around the interest points in the 3-D spatio-temporal volume. For simplicity, we extract the brightness gradients of three channels (G_x, G_y, G_z) from each cuboid and flatten the cuboid into a vector [17]. To eliminate noise and retain principle information, Principle Component Analysis (PCA) is used to reduce the dimensionality of the visual feature vector. Besides appearance information, spatial-temporal context information is also important for mouse classification. As we have known, some behaviours look very similar but may locate at different areas (e.g. drinking and eating). Using spatial-temporal context information will help further improve the discriminability of the extracted features. We also exploit the contextual information of the interest points to characterise both spatial location and temporal changes of a mouse. Two types of features are computed: the relative and absolute spatial positions of the interest points [38]. If there are Q interest points in an action clip, then our contextual feature vector has the form: $F_q = \frac{[X_q - X_c; Y_q - Y_c; T_q - T_c; X_q; Y_q]}{\|[X_q - X_c; Y_q - Y_c; T_q - T_c; X_q; Y_q]\|_2}, q = 1, 2, ..., Q$ where $[X_c; Y_c; T_c]$ and $[X_q; Y_q; T_q]$ represent the center coordinate of all the interest points and the absolute coordinate of the qth interest point respectively in the spatial-temporal domain.

3) SFV encoding layer.: The Fisher vector can be derived from the Fisher kernel [40] and has been applied by Perronnin et al. [25] for large-scale image categorisation. Assuming that all interest points are independent and given a GMM with parameters $\lambda = \{\omega_k, \mu_k, \sigma_k, k = 1, ..., K\}$ where $\omega_k, \mu_k, \sigma_k$ and K respectively denote the mixture weight, mean vector, standard deviation vector (diagonal covariance) and the num-

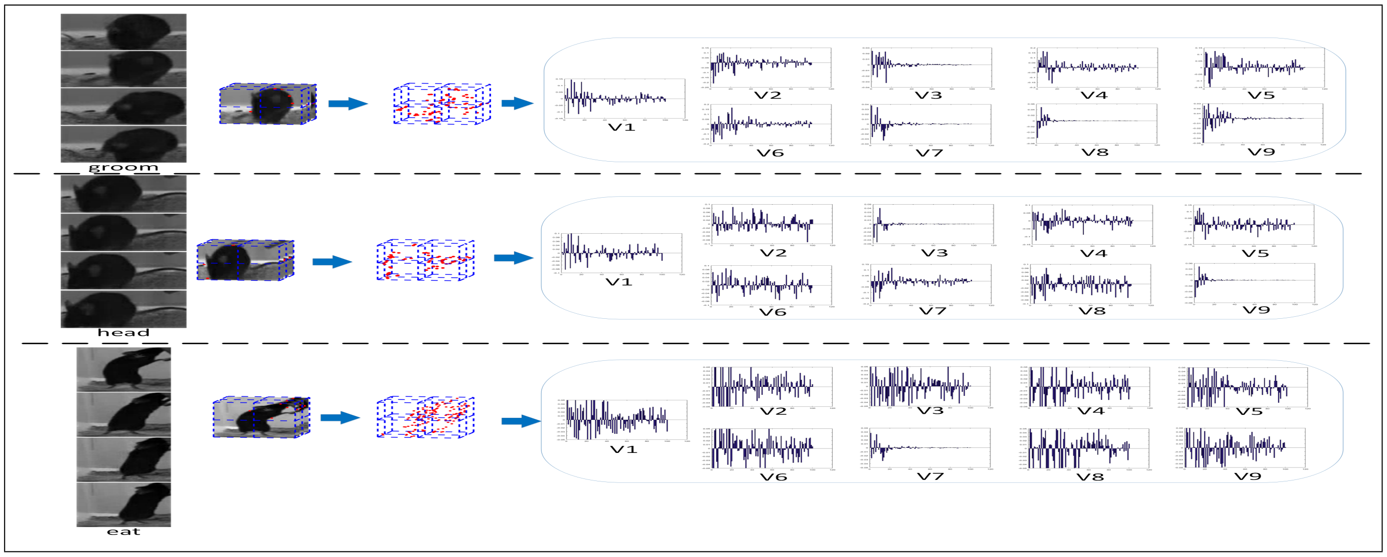


Fig. 4. Visualization of SFV representation for different action clips.

ber of Gaussians. FV has the form:

$$\mathcal{G}_{\mu,k}^{X} = \frac{1}{Y\sqrt{\omega_k}} \sum_{y=1}^{Y} \gamma_y(k) \left(\frac{x_y - \mu_k}{\sigma_k}\right) \tag{5}$$

$$\mathcal{G}_{\sigma,k}^{X} = \frac{1}{Y\sqrt{\omega_k}} \sum_{y=1}^{Y} \gamma_y(k) \left[\frac{(x_y - \mu_k)^2}{\sigma_k^2} - 1\right] \tag{6}$$

where Y is the number of the interest points detected from an action clip. Parameter $\gamma_d(k)$ is the weight of x_y to the kth Gaussian: $\gamma_y(k) = \frac{\omega_k u_k(x_n;\mu_k,\Sigma_k)}{\sum_{k=1}^{K}\omega_k u_k(x_n;\mu_k,\Sigma_k)}$. To correct for the independence assumption of interest points, we apply power normalization followed by L_2 normalization [25] to each $\mathcal{G}_{\mu,k}^{X}$ and $\mathcal{G}_{\sigma,k}^{X}$ before concatenating them together. It is observed that a critical step in the encoding layer is to learn the GMM from a set of local features $X^* = \{x_1, x_2, ..., x_M\}$. These features are randomly sampled from the universal set of local features which are extracted from the interest points of all the training action clips. Given the feature set X^*, the optimal parameters of GMM are learned through maximum likelihood estimation [37]:

$$argmax_{\lambda} \sum_{m=1}^{M} \ln \left[\sum_{k=1}^{K} \omega_k \mathcal{N}(x_m;\mu_k,\sigma_k)\right], \tag{7}$$

where $\mathcal{N}(x;\mu_k,\sigma_k)$ is D-dimensional Gaussian distribution and D is the dimensionality of the feature vector. We use the iterative EM algorithm of [37] to render the solution. The traditional FV coding only learns the GMM from the appearance features, e.g. [25]–[28]. To capture the distribution of appearance and contextual features, we learn two GMMs for these two features respectively. Based on Eqs. (5) and (6), two FVs are used to represent an action clip. Note that the global sum-pooling used in Eqs. (5) and (6) ignores the relative location of the aggregated features. To derive an optimal spatial-temporal structure of the action class, inspired by the methods reported in [41]–[44], we trim the entire video clip into one which only contains all the interest points and then empirically segment this small video clip to 2*2*2 subvolumes. Both contextual and appearance features encoded by two Fisher vectors are then computed for each subvolume and the whole video clip. After having normalised these two FVs for each subvolume and the whole video clip by power and L2 normalisation, we concatenate them into one feature vector in the subvolume domain and then give this as the input of the next supervised layers. Note, contextual and appearance features are complementary and jointly boost the recognition rate (see Section IV A for justification). We finally obtain a sequence of FV representations $\{V_1, V_2, ...V_H\}$. Here V_1 are computed from a whole video clip, $\{V_2, ...V_H\}$ come from several subvolumes, and $H = 9$ in our experiment. Fig.4 illustrates the SFV representation for some examples of action clips. Actions 'groom' and 'head' are easily misclassified by the traditional FV pipeline as they always obtain a similar FV representation (V_1) which aggregates codes in an entire clip. However subvolume-level features in our SFV layer can provide more clues to distinguish these two challenging behaviours, for example V_2, V_6 and V_8. Algorithm 1 summarises the steps of our SFV scheme.

Algorithm 1 Algorithm for the proposed SFV scheme.

Input: Input action clip O and the number of Gaussian K.
Output: feature representation V.

```
1: Detect interest points from the input $O$ using proposed
   LOG filter + frame differencing method;
2: Extract appearance and contextual features for each inter-
   est point;
3: Encode each interest point using appearance and contex-
   tual GMMs which are learned through Eq. (7);
4: Sum-pooling, normalise and fuse features as described in
   the SFV encoding layer of Section III-A to obtain feature
   representations.
   $V = \{V_h, h = 1, ..., H\}$.
```

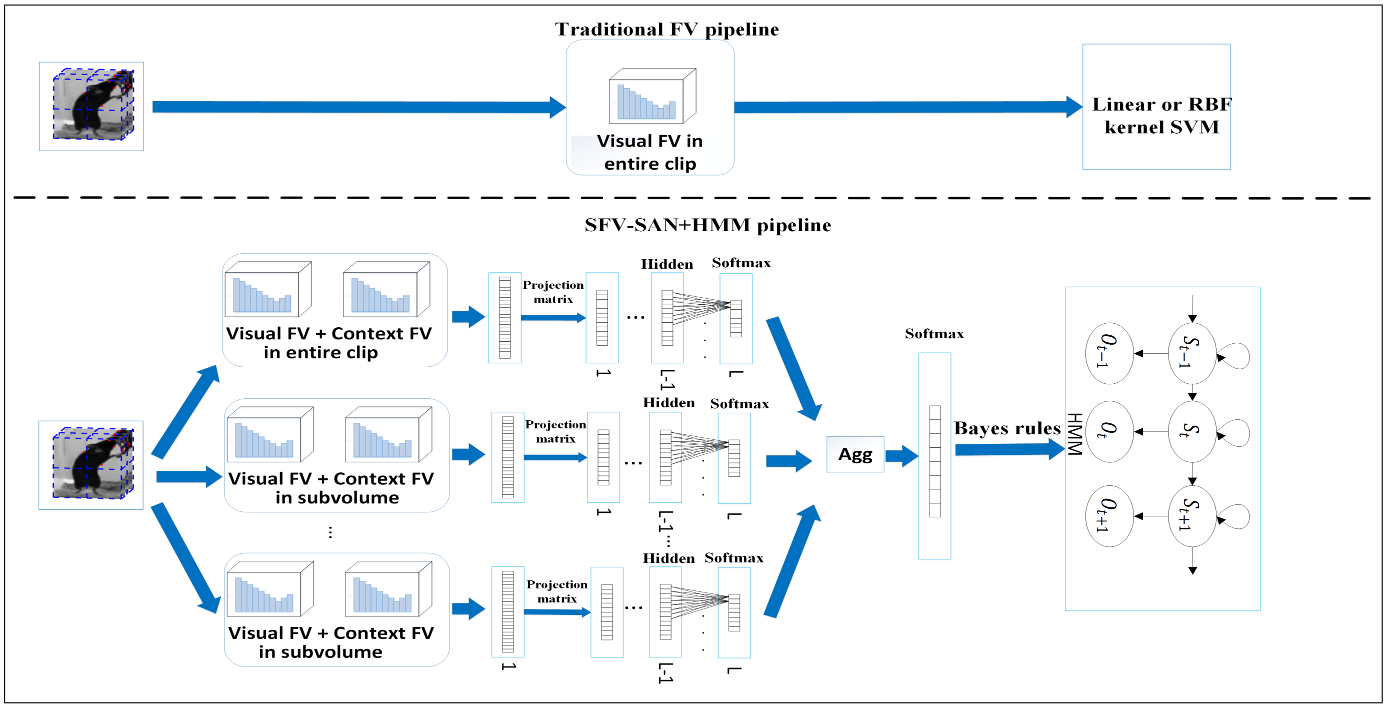


Fig. 5. Comparison between the proposed SFV-SAN pipeline and the traditional FV pipeline.

B. Supervised layers

To infer the best state sequence from the sequence of sliding windows, we establish several supervised layers upon the previous unsupervised layers, whose parameters are learned by using the sequence of sliding windows and action labels from the training dataset

1) SAN layers: In the supervised layers, the SAN cooperates with SFV and finally obtains probability $P\left(S_t = A_m | O_t\right)$ that is transformed to the emission probability of the next HMM layer. Given a sequence of FVs $\{V_1, V_2, ...V_H\}$, the SAN models them as follows:

$$SAN\left(V_1, V_2, ...V_H\right) = \mathcal{O}\left(Agg\left(\mathcal{F}\left(V_1^T P_1; W\right), \mathcal{F}\left(V_2^T P_2; W\right), ...\mathcal{F}\left(V_H^T P_H; W\right)\right)\right) \quad (8)$$

where $\mathcal{F}\left(V_h^T P_h; W\right)$ is the function operating on the FV representation V_h and produces classification results for all the classes. The parameter P_h is a projection matrix for dimensionality reduction of the FV representation V_h. The projection matrix $\{P_1, P_2, ...P_H\}$ and W of all the connected layers are jointly learned. The segment aggregate function Agg (average pooling) combines the outputs from multiple segments to obtain a consensus of class hypothesis among them. For obtaining the probability of each action class for the entire video clip, we use a softmax function as the prediction function $\mathcal{O}$. Combined with the standard categorical cross-entropy loss, the final loss function is formed below:

$$J = E + \frac{\lambda}{2}\|W\|_2^2 + \frac{\mu}{2}\sum_{h=1}^{H}\|P_h\|_F^2, \quad (9)$$

$$E = -\sum_{m=1}^{M} y_m\left[G_m - \log\left(\sum_{i=1}^{M}\exp\left(G_m\right)\right)\right]. \quad (10)$$

λ and μ are the regularization constants of the $L2$ norm of W and the Frobenius norm of all P_h respectively. These two regularization terms prevent the model from overfitting the training data. Eq. (10) is different from the traditional cross-entropy loss because we introduce the segment aggregate function into our loss function. We denote $y_m = 1$ if a clip is tagged as a numerical action label m of M actions and $y_m = 0$ otherwise. $G_m = Agg\left(\mathcal{F}_m\left(V_1^T P_1\right), ..., \mathcal{F}_m\left(V_H^T P_H\right)\right)$. Note that, $\mathcal{F}_m$ represents the classification result of class m, which can be obtained from the last softmax layer of each segment. Here a class score G_m is inferred from the scores of the same class and the same sample on all the FV representations. We use an average pooling function to achieve this score aggregation. In the back-propagation process, the gradients of parameters w_{ij}^h and P_h with respect to the loss value J can be derived as follows:

$$\frac{\partial J}{\partial w_{ij}^h} = \sum_m^M \frac{\partial J}{\partial G_m}\sum_h^H \frac{\partial G_m}{\partial \mathcal{F}_m\left(V_h^T P_h\right)}\frac{\partial \mathcal{F}_m\left(V_h^T P_h\right)}{\partial w_{ij}^h}. \quad (11)$$

$$\frac{\partial J}{\partial P_h} = \sum_m^M \frac{\partial J}{\partial G_m}\sum_h^H \frac{\partial G_m}{\partial \mathcal{F}_m\left(V_h^T P_h\right)}\frac{\partial \mathcal{F}_m\left(V_h^T P_h\right)}{\partial\left(V_h^T P_h\right)}\frac{\partial\left(V_h^T P_h\right)}{\partial P_h}. \quad (12)$$

Here, w_{ij}^h means the weight from the i_{th} neuron in the l_{th} layer to the j_{th} neuron in the $l_{th}+1$ layer. l_{th} and $l_{th}+1$ two are layers after the projection layer, serving the same subvolume h. Eqs. (11) and (12) ensure that the parameter updating uses the segmental aggregation G derived from all the subvolume-level prediction. Figure 5 shows the comparison between the traditional FV pipeline and our proposed SFV-SAN+HMM approach. The traditional FV pipeline encodes the local features of the action video and aggregates the codes over the entire video by sum pooling. Subsequently this representation is usually fed to a Linear or RBF kernel SVM. Since the traditional FV pipeline represents and classifies the video in the local feature space, it cannot mine spatial-temporal structures. Such a pipeline is not good enough for the applications which require a high discriminative ability, e.g., the behaviour recognition task studied in this paper. To overcome this limitation, we propose a SFV layer, which encodes and aggregates the local features by sum pooling over each subvolume, followed by a SAN for capturing spatial-temporal structures and aggregating information from these subvolumes. For learning our model parameters, we firstly initialise P_h by PCA and w_{ij}^h from a zero-mean Gaussian. We then use a mini-batch (set to 8) stochastic gradient descent (SGD) paradigm, a popular and efficient method to solve stochastic optimization problems which arise in machine learning [41], [45]–[47], to update these parameters, where we experimentally set the learning rate to 0.03 and the momentum to 0.95. The regularization parameters λ and μ in Eq. (9) are set to $2 \cdot 10^{-6}$ and $2 \cdot 10^{-5}$, respectively. The SGD algorithm is an iterative method for minimizing the loss function Eq. (9) and updates the parameters by

$$w_{ij}^h = \widehat{w}_{ij}^h + \Delta w_{ij}^h + \xi \Delta \widehat{w}_{ij}^h, \tag{13}$$

where w_{ij}^h and Δw_{ij}^h are the weight and the update weight in the current iteration. The hat above the variable denote that its value is computed from the previous iteration. ξ is the momentum. In order to construct the above update equation, we need to compute the gradient of our objective function J with respect to w_{ij}^h, which is determined by its location in the neural network and can be derived from the chain rule. For the softmax layer of the h segment: The update weight Δw_{ij}^h has the following form:

$$\begin{aligned}\Delta w_{ij}^h &= -\eta \frac{1}{N}\sum_n^N \frac{\partial J_n}{\partial w_{n,ij}^h} \\ &= -\eta \frac{1}{N}\sum_n^N \left(\frac{\partial E_n}{\partial z_{n,j}^h}\frac{\partial z_{n,j}^h}{\partial w_{n,ij}^h} + \lambda w_{n,ij}^h\right),\end{aligned} \tag{14}$$

where $z_{n,j}^h = \sum_i w_{n,ij}^h b_{n,i}^h$, $b_{n,i}^h$ is the i_{th} output of the former hidden layer, and N is the size of the mini-batch. Then we can get:

$$\frac{\partial z_{n,j}^h}{\partial w_{n,ij}^h} = b_{n,i}^h, \tag{15}$$

$$\frac{\partial E_n}{\partial z_{n,j}^h} = \left(\frac{\exp\left(G_{n,j}\right)}{\sum_{n,j'}\exp\left(G_{n,j'}\right)} - y_{n,j}\right)\sum_h^H\sum_{n,j'} \frac{\exp\left(z_{n,j'}^h\right)\exp\left(z_{n,j}^h\right) - \exp\left(z_{n,j}^h\right)\sum_{n,j'}\exp\left(z_{n,j'}^h\right)}{\left(\sum_{j'}\exp\left(z_{n,j'}^h\right)\right)^2}. \tag{16}$$

For the hidden layer of the h segment: To avoid confusion, we replace j in Eq. (16) with k, and define j and k as the neuron number in the last hidden layer and the softmax layer, respectively. We also define:

$$\theta_{n,k}^h = \frac{\partial E_n}{\partial z_{n,k}^h}. \tag{17}$$

The update Δw_{ij}^h in the last hidden layer follows:

$$\begin{aligned}\Delta w_{ij}^h &= -\eta \frac{1}{N}\sum_n^N \frac{\partial J_n}{\partial w_{n,ij}^h} \\ &= -\eta \frac{1}{N}\sum_n^N \left(\frac{\partial E_n}{\partial z_{n,j}^h}\frac{\partial z_{n,j}^h}{\partial w_{n,ij}^h} + \lambda w_{n,ij}^h\right) \\ &= -\eta \frac{1}{N}\sum_n^N \left(\sum_k^K \frac{\partial E_n}{\partial z_{n,k}^h}\frac{\partial z_{n,k}^h}{\partial b_{n,}^h}\frac{\partial b_{n,j}^h}{\partial z_{n,j}^h}\frac{\partial z_{n,j}^h}{\partial w_{n,ij}^h} + \lambda w_{n,ij}^h\right),\end{aligned} \tag{18}$$

where η is the learning rate. $b_{n,j}^h = \sigma\left(z_{n,j}^h\right)$. We use a rectified Linear Unit (reLU) $\sigma(x) = max(0, x)$, which makes the training faster and less error-prone with respect to the involved $tanh$ and sigmoid units [48]. Finally, we simplify Eq. (18) as follows:

$$\Delta w_{ij}^h = \begin{cases} 0 & \text{if} \quad z_{n,j}^h \le 0 \\ -\eta\frac{1}{N}\sum_n^N\left(\sum_k^K \right. & \\ \left.\theta_{n,k}^h w_{n,jk}\left(z_{n,j}^h\right)b_{n,i}^h + \lambda w_{n,ij}^h\right) & \text{if} \quad z_{n,j}^h > 0 \end{cases} \tag{19}$$

Similar to Eq. (17), we define $\theta_{n,j}^h = \frac{\partial E_n}{\partial z_{n,j}^h}$, i.e.,

$$\theta_{n,j}^h = \begin{cases} 0 & \text{if} \quad z_{n,j}^h \le 0 \\ \sum_k^K \theta_{n,k}^h w_{n,jk}\left(z_{n,j}^h\right) & \text{if} \quad z_{n,j}^h > 0 \end{cases} \tag{20}$$

We then extend the above equation to the other hidden layers in a recursive manner:

$$\theta_{n,j}^h = \begin{cases} 0 & \text{if} \quad z_j^h \le 0 \\ \sum_{c=1}^{C_{l+1}} \theta_{n,c}^h w_{n,jc}\left(z_{n,j}^h\right) & \text{if} \quad z_{n,j}^h > 0 \end{cases} \tag{21}$$

where C_{l+1} is the total number of the neurons in the $l_{th}+1$ layer. In the hidden layer, the update Δw_{ij}^h can be described as

$$\Delta w_{ij}^h = -\eta\frac{1}{N}\sum_n^N\left(\theta_{n,j}^h b_{n,i}^h + \lambda w_{n,ij}^h\right). \tag{22}$$

To study the influence of the number of the hidden layers, we conduct experiments in Section V-A setting this number from 0 to 2.

For the projection layer of the h segment, we update P using the follow equation:

$$\begin{aligned}P_h &= P_h - \eta\frac{\partial J_n}{\partial P_{n,h}} \\ &= P_h - \eta\frac{1}{N}\sum_n^N\left(\frac{\partial E_n}{\partial\left(V_{n,h}^T P_{n,h}\right)}\frac{\partial\left(V_{n,h}^T P_{n,h}\right)}{\partial P_{n,h}} + \mu P_{n,h}\right) \\ &= P_h - \eta\left(\Theta_{n,h}V_{n,h}^T + \mu P_{n,h}\right), \Theta_{n,h} = [\theta_{n,1}^h, ..., \theta_{n,C_1}^h].\end{aligned} \tag{23}$$

2) HMM layer: The goal of the HMM layer is to infer the most likely sequence of action states $S^* = \{S_1, S_2, ..., S_T\}$, based on the class scores of SAN. Let $S_t = A_m \in A^*$ denote the final state of the most probable state sequence $S_{1:t}$, the posteriors $P(S_t = A_m|O_t)$ can be transformed into emission probabilities using Bayes' rule $P(O_t|S_t = A_m) \propto \frac{P(S_t=A_m|O_t)}{P(S_t=A_m)}$. By introducing the class scores of SAN, Eqs. (1) and (2) of the standard Viterbi algorithm can be modified as:

Initializing:

$$\begin{aligned}\mu_1^m &= P(S_t = A_m|O_t) \cdot \frac{1}{P(S_t = A_m)} \cdot \pi_m \\ &= \frac{exp(G_{1,m})}{\sum_{m'} exp(G_{1,m'})} \cdot \frac{\sum_{m'} u_{m'}}{u_m} \cdot \pi_m \\ \psi_1^m &= 0.\end{aligned} \tag{24}$$

Recursing:

$$\begin{aligned}\mu_t^m &= \max_{1\le m\le M}\left(P(S_t = A_m|O_t) \cdot \frac{1}{P(S_t = A_m)} \cdot \alpha_{nm} \cdot \mu_{t-1}^n\right) \\ &= \max_{1\le m\le M}\left(\frac{exp(G_{t,m})}{\sum_{i'} exp(G_{t,m'})} \cdot \frac{\sum_{m'} u_{m'}}{u_m} \cdot \frac{q_{nm}}{\sum_{m'} q_{nm'}} \cdot \mu_{t-1}^n\right) \\ \psi_t^n &= \operatorname*{argmax}_{1\le n\le M}\left(\frac{exp(G_{t,m})}{\sum_{i'} exp(G_{t,m'})} \cdot \frac{\sum_{m'} u_{m'}}{u_m} \cdot \frac{q_{nm}}{\sum_{m'} q_{nm'}} \cdot \mu_{t-1}^n\right).\end{aligned} \tag{25}$$

Terminating:

$$\begin{aligned}m_T &= \operatorname*{argmax}_{1\le m\le M}(\mu_T^m), \\ S_T &= A_{m_T}.\end{aligned} \tag{26}$$

Path backtracking:

$$\begin{aligned}m_{t-1} &= \psi_t^{m_t}, \\ S_{t-1} &= A_{m_{t-1}}.\end{aligned} \tag{27}$$

where $G_{t,m}$ is the class score from the output of our SAN at time t. The action occurrence u_m and the action transition occurrence q_{nm} can be estimated with action labels Y^* in the training data as described in Section II-B. The algorithm for training and testing SAN+HMM is outlined in Algorithms 2 and 3.

Algorithm 2 Algorithm for training the proposed SAN+HMM.

```
Input: Sequence of sliding windows $O^* = \{O_1, O_2, ..., O_T\}$
    and action labels $Y^* = \{Y_1, Y_2, ..., Y_T\}$ for training, SGD
    parameters $\mu, \lambda, \eta, \xi, N$.
Output: SAN, $\Pi = \{\pi_1, ..., \pi_M\}$, transition matrix $\dot{A}$.
    Initialisation: initialize $P_h$ by PCA and $w_{ij}^h$ from a zero-
    mean Gaussian.
 1: Compute feature representations $V^* = \{V_h^t, h = 1, ..., H, t = 1, ..., T\}$ for $O^*$ using Algorithm 1;
 2: Compute prior probabilities $\Pi$ and transition matrix $\dot{A}$ of
    HMM using $Y^*$, as described in section II-B;
 3: Randomly shuffle $V^*$ on $t$ dimensions;
 4: for $iter = 1$ to $\frac{T}{N}$ do
 5:     Propagation forward through the network to generate
        $b_{n,i}^h$ and $z_{n,j}^h$ of all layers;
 6:     Updates for all the weights from the softmax layer using
        Eqs. (14), (15) and (16);
 7:     Update for all the weights from the hidden layer using
        Eqs. (20), (21) and (22);
 8:     Update network weights using Eq. (13);
 9:     Update $P$ in projection layer using Eq. (23);
10:     Record the update $\Delta w_{ij}^h$.
11: end for
12: return SAN, $\Pi$, $\dot{A}$.
```

Algorithm 3 Algorithm for testing the proposed SAN+HMM.

```
Input: Sequence of sliding windows $O^* = \{O_1, O_2, ..., O_T\}$
    for testing; the SAN and HMM with parameters $\Pi$ and $\dot{A}$
    using Algorithm 2.
Output: State sequence $S^* = \{S_1, S_2, ..., S_T\}$.
 1: for $t = 1$ to $T$ do
 2:     Compute feature representation $V^* = \{V_h, h = 1, ..., H\}$ of $O_t$ using Algorithm 1;
 3:     Propagation forward through the network to generate
        the class scores $G_{t,m}$;
 4:     if $t = 1$ then
 5:         Initialize $\mu_1^m$ and backpointer $\psi_t^n$ using Eq. (24);
 6:     else
 7:         Compute $\mu_1^m$ and backpointer $\psi_t^n$ using Eq. (25).
 8:     end if
 9: end for
10: Compute the last state $S_T$ using Eq. (26).
11: Compute the optimal state sequence $S^*$ using path back-
    tracking described in Eq. (27).
```

IV. Experimental Setup

A. Datasets:

In our experimental work, we firstly use the publicly accessible Jhuang databases [18]. The public Jhuang database has two parts: 'clipped database' and 'full database'. The 'clipped database' contains 4200 clips in which only the best instances of specific behaviours are included. It consists of 8 mouse behaviour classes: rear (399 cases), groom (1477), eat (374), drink (61), hang (521), rest (868), walk (233) and

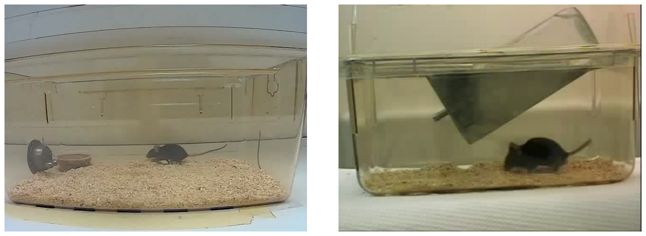

(a) Our Database. (b) Jhuang Database.

Fig. 6. Database used in our experiment.

head (180). Each clip records a single mouse from a side-view camera (see Figure 6 for examples of video frames). The 'full database' contains 12 frame-by-frame labelled videos that last over 10 hours in total. In this paper, experiments on the 'clipped database' use a half-by-half cross-validation procedure. A leave-one-out strategy is used in our experiments on the 'full database'.

Apart from the public database described above, we also recorded four videos of a single mouse using a Sony Action camera (HDR-AS15) with a frame rate of 30 fps and 640 by 480 pixels' VGA video resolution. The mouse used throughout this study was housed under constant climatic conditions with free access to food and water. All experimental procedures were performed in accordance with the Guidance on the Operation of the Animals (Scientific Procedures) Act, 1986 (UK) and approved by the Queen's University Belfast Animal Welfare and Ethical Review Body. Our recognition system has been designed to suit different mouse cages. The location of mouse foods and drinks in our mouse cage is changing (see Figure 6). To facilitate the 'dig' behaviour in the experiment, we covered the ground of the cage with sawdust. However, this sawdust leads to severe background clutters, a big challenge for mouse detection and tracking. This database includes 4 frame-by-frame labelled videos, each of which lasts 30 minutes and contains 6 different behaviours: rear (defined by an upright posture and forelimbs off the ground), groom (defined by the forelimbs sweeping across the face or torso), eat (defined by the mouse reaching and acquiring food from the food bin), walk (defined by ambulation), micro-movement (defined by small movements of the mouse's head or limbs) and dig (defined by raising of sawdust with the forelimbs and/or head).

B. Baseline features:

In recent years, trajectory-based approaches, e.g. Improved Dense Trajectory (IDT) [39], have attracted attention due to their resilience to noise and illumination change. Their approach allows image points to be densely sampled which are tracked using optical flow. We here use the default trajectory length of 15 frames. The trajectory descriptor describes its shape by a sequence of displacement vectors. Histograms of Oriented Gradients (HOG), Histograms of Optical Flow (HOF) and Motion Boundary Histograms (MBH) are computed in the spatio-temporal volume aligned with each trajectory. For each trajectory, we compute HOG, HOF and MBH descriptors with the same parameters as shown in [48]. The final dimensions of the descriptors are 30 for Trajectory, 96 for HOG, 108 for HOF and 192 for MBH.

Other state-of-the-art approaches with respect to deep learning features include Gated Restricted Boltzmann Machines [49], 3D Convolutional Neural Networks [50], Deep Convolutional Neural Networks [51], Two-Stream Convolutional Networks [46]. Wang et.al [47] argued that the current network architectures for action recognition in videos are relatively shallow compared with those very deep models in the image domain [52], [53]. They therefore proposed a very deep two-stream ConvNets and achieved the state of the art results. We use their trained VGG-16 CNNs, which are pretrained on ImageNet [52] and fine-tuned on the UCF-101 dataset. We fix the parameters of the first 13 convolutional layers and re-train the last 3 fully connected layers on the mouse dataset. Wang et.al [26] designed a trajectory-pooled deep convolutional descriptor (TDD), whose goal is to combine the benefits of both trajectory-based and deep-learned features. This local trajectory-aligned descriptor is computed from the spatial and temporal nets. We load the parameters of the convolutional layers from the networks that have already been trained on the ImageNet dataset [45]. We choose the descriptors from conv4 and conv5 layers for the spatial nets, and conv3 and con4 layers for the temporal nets. Finally, we de-correlate TDD with PCA and reduce its dimensions. The reason for choosing the TDD feature as our baseline is that the feature shows great performance on several datasets of human action and can be effectively integrated into our SFV-SAN+HMM network to improve the system performance.

C. Baseline encoding method:

The traditional Fisher Vector pipeline [25] is chosen as our baseline encoding method. We set the number of Gaussians to $K = 20$ for the best results in the experiments, and each feature is represented with a $2KD$ dimensional Fisher vector, where D is the dimension of the feature vector. To normalise a Fisher vector, we apply power and L_2 normalisation. Finally, we concatenate the normalised Fisher vectors of different descriptors. A linear SVM is used for classification. Besides, we also compare our SFV-SAN pipeline with the stacked FVs, a video-level representation with multi-layer nested Fisher vector encoding, which was proposed by Peng et al. [27].

D. Baseline recurrent neural network(RNN):

Recently, recurrent neural networks, especially Long Short-Term Memory (LSTM) networks, have demonstrated their large success in speech recognition [54] and human action recognition [55]. We use a LSTM network trained using Adaptive Moment Estimation (Adam) instead of linear SVM in the traditional FV pipeline. The learning rate of the model was initially set to 0.01 and the hidden unit was set to 100, experimentally.

V. Experimental Results

A. Experiments on 'clipped database' of JHuang

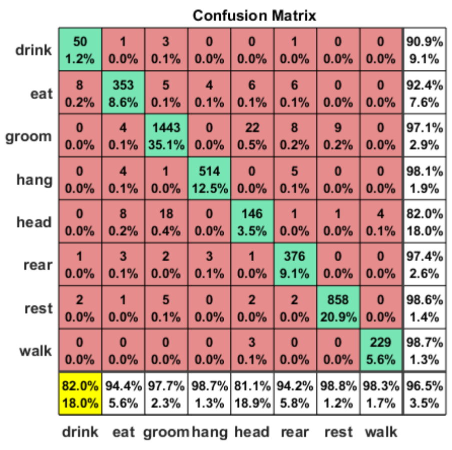
We compare the performance of our visual features (VF) and contextual features (CF) against that of the state-of-the-

TABLE I
PERFORMANCE (ACCURACY) OF DIFFERENT FEATURES FOR CLIPPED DATASET.

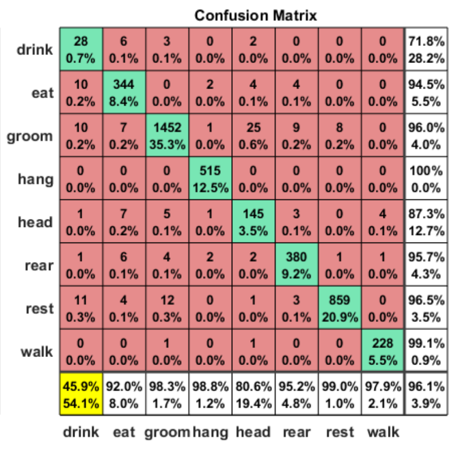
Features			Traditional FV pipeline [25]		SFV-SAN pipeline		
			linear SVM	RBF SVM	0 NN	1 NN	2 NN
IDT [48]	Trajectory		73.8%	74.5%	73.5%	**78.1%**	75.7%
	HOG		91.6%	93.1%	88.8%	**94.7%**	93.7%
	HOF		83.2%	86.9%	82.7%	**88.3%**	86.0%
	MBH		87.9%	88.1%	87.6%	**91.0%**	89.9%
	Combined IDT with Traj.		91.8%	92.1%	91.9%	**93.8%**	93.5%
	Combined IDT without Traj.		92.3%	92.7%	93.0%	**94.5%**	94.0%
TDD [26]	Spatial conv4 and conv5		93.1%	93.1%	93.8%	**95.3%**	94.7%
	Temporal conv3 and conv4		93.2%	92.7%	93.0%	**93.7%**	93.4%
	Combined TDD		95.1%	94.0%	95.5%	**96.1%**	95.4%
ours	*With frame differencing*	Visual features	91.4%	89.4%	89.2%	**93.1%**	90.7%
		Contextual features	92.2%	90.8%	90.4%	**93.3%**	92.0%
		Combined VF&CF	95.4%	91.1%	94.9%	**96.5%**	96.3%
	Without frame differencing	Combined VF&CF	94.7%	90.2%	94.2%	**96.0%**	95.7%
itcombined	*ours+IDT without Traj.*		95.5%	94.5%	97.2%	**97.4%**	97.3%
	TDD+ours		95.3%	94.5%	97.0%	**97.5%**	96.8%
	TDD+IDT without Traj.		95.3%	94.3%	96.0%	**96.7%**	96.1%
	TDD+ours+IDT with Traj.		95.1%	94.1%	96.4%	**97.0%**	96.7%
	TDD+ours+IDT without Traj.		95.4%	94.7%	97.6%	**97.9%**	97.0%
Dollár [17]			81.0%(Bag of visual words + linear SVM)				
Jhuang [18]			93.0%(linear SVM)				

TABLE II
PERFORMANCE (ACCURACY) OF DIFFERENT FEATURES FOR CLIPPED DATASET.

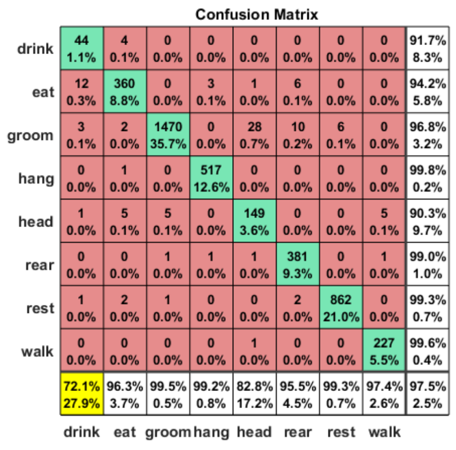
Action	our visual features	our contextual features	Combined VF&CF	Combined IDT	Combined TDD	Combined IDT+TDD	Combined IDT+TDD +ours
drink	42.6%	70.5%	**80.3%**	50.8%	45.9%	62.3%	77.0%
eat	85.3%	92.2%	93.3%	90.1%	92.0%	94.1%	**97.3%**
groom	96.5%	96.4%	98.2%	97.0%	98.3%	98.0%	**99.1%**
hang	96.7%	94.6%	98.5%	96.0%	98.8%	99.2%	**99.2%**
head	71.7%	58.9%	80.6%	77.8%	80.6%	86.1%	**86.1%**
rear	85.7%	88.5%	93.5%	93.2%	95.2%	96.0%	**96.7%**
rest	98.3%	98.8%	99.1%	97.1%	99.0%	98.8%	**99.1%**
walk	99.6%	92.3%	97.9%	99.1%	97.9%	97.9%	**99.6%**
all	93.1%	93.3%	96.5%	94.5%	96.1%	96.7%	**97.9%**



(a) Combined VF and CF



(b) TDD.



(c) Combined TDD+ours.

Fig. 7. Confusion matrixes of our system using different features.The diagonal cells show the number and percentage of correct classifications. The non-diagonal cells contain the number and percentage of incorrectly classified behaviors. The proportion of each actual behavior that were correctly or incorrectly predicted is shown in the bottom row. The proportion of each predicted behavior that were correct or incorrect is shown in the rightmost column. Overall, the proportion of correct predictions is shown in the bottom right corner.

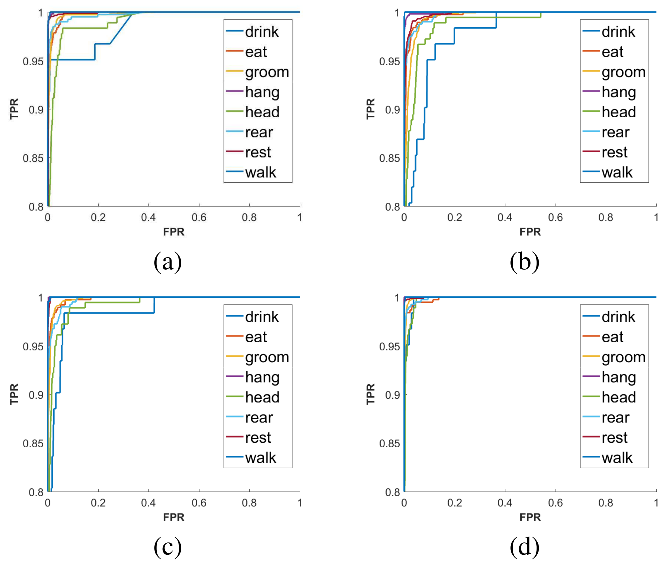


Fig. 8. ROC curve shows the true positive rate (TPR) against the false positive rate (FPR) for the features: (a) combined visual and contextual features; (b) IDT combined features; (c) TDD combined features; (d) all combined features.

art IDT and TDD features using the same encoding method. In our empirical study, we set 100 units per hidden layer and use reLU non-linearity as the activation function between two layers. We experiment with a set of neural network layers operating on the FV representation of each subvolume and the entire video clip. The number of the hidden layers varies from 0 to 2, compared against the traditional FV pipeline with radial basis function (RBF) and linear kernels SVM. Table I shows the performance of the different features, where the traditional FV pipeline with linear or RBF kernel SVM perform significantly worse than our SFV-SAN pipeline. Especially, the neural network with one hidden layer has the best performance no matter what features it takes. More hidden layers seem to make the model overfit. In JHuang's dataset, the CNN based TDD features lead to the performance similar to our combined VF and CF features. Table VI shows that our combined VF and CF features lead to 3.2% better performance than the TDD features for our datasets. We also explore the the complementary capacity of these three methods by concatenating their Fisher vectors in each subvolume, and observe that the integration can further improve the system performance to 97.9%. However, it is interesting to notice that the combined features without IDT trajectory have better performance that that with IDT trajectory. That is to say, it is not always true that adding more features will produce better classification results. In fact, it is complementary features rather than a random combination of features to boost the system performance. Table I also shows that our approach significantly outperforms Jhuang (93.0%) and Dollár (81.0%), both of which use much larger proportions of the dataset for training.

Table II illustrates the performance of the different features for specific behaviours. In Table II, we observe that the system performance for 'head' is clearly worse than the others. The main reason is that 'head' is affected by the micro-movements of the mouse's head or limbs so it is easily confused with similar behaviours such as 'groom'. Regarding the low accuracy for 'drink', we believe that this is due to the imbalanced database where 'drink' has a very small training set (1% of the whole dataset). However, we find that our VF+CF features still achieve sufficient accuracy on the 'drink' behaviour, meanwhile more than half of the 'drink' clips are incorrectly classified using the TDD features, shown in yellow cell of Fig.7(b). Fig.7(c) shows that adding our VF+CF features can help improve the system performance in discriminating between visually similar behaviours, for which the location of the mouse in the cage provides critical information. For example, drinking (versus eating) occurs at the water bottle spout. Figure 8 clearly shows that the combined features are able to achieve significantly higher accuracy for each behaviour than individual uses of them.

B. *Experiments on 'full database' of Jhuang*

Our system performance is evaluated here based on a leave-one-out cross-validation procedure. This procedure employs all the videos except one to train the system (via random sampling) and the remaining video for testing. We repeat this procedure $n = 12$ times and report the weighted average accuracy in Table III. There are two annotations: 'Group 1' and 'Group 2', where 'Group 1' is used as the ground truth to train and test the system while 'Group 2' is used for measuring the human agreement between two independent human annotators. All the local features are computed from a short sliding video (40 frames), which are centered at each frame. As shown in Table III, our features have a better ability to represent mouse behaviours. Although the metric used might suggest that the performance difference between our proposed features and Jhuang's is small, analyzing each behaviour separately shows that ours outperforms Jhuang's system in 5 out of the 8 behaviours (eat, groom, hang, rear, rest). In particular, Table II shows a very poor performance by FV+LSTM (17.1% by FV+LSTM vs 60.3% by SFV-SAN+HMM) for the recognition of the 'drink. We consider the imbalance training dataset (only 0.3% of the video is annotated as 'drink') influences the performance of LSTM. It is true that SFV-SAN+HMM has a better capability than LSTM to deal with the bias towards the majority class in this example. Overall, our system trained with the fusion of all the features achieves 81.5% agreement with 'Group 1', which is higher than the commercial system [56] , Jhuang's system and humans (71.6%). To study the important role of our SFV-SAN+HMM pipeline, we train it with different features and report results in Table IV. The columns on the left side show the performance of each local feature and their combination with the traditional FV pipeline. The columns in the middle show the results after applying our SFV-SAN pipeline. Additionally, incorporating our HMM model is shown in the columns on the right side. In comparison to the baseline of the traditional FV pipeline, our SFV-SAN pipeline results in 4.9% better performance on the combined features, 6.7% better on average. The HMM model further improves the performance and lead to a final performance improvement of 6.4% and 8.7% on the combined features and average compared to the baseline method.

TABLE III
COMPARISON OF DIFFERENT SYSTEMS.

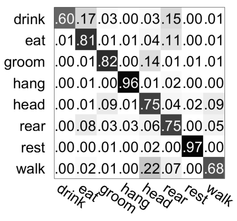
Action	VF&CF (SFV-SAN+HMM)	IDT (traditional FV)	TDD (traditional FV)	IDT+TDD (SFV-SAN+HMM)	VF&CF+IDT+TDD		JHuang	Commercial system	Human
					SFV-SAN+HMM	FV+LSTM			
drink	59.4%	21.9%	24.7%	61.6%	60.3%	17.1%	72.0%	63.0%	78.0%
eat	80.3%	62.0%	69.7%	77.7%	81.1%	78.6%	75.0%	73.0%	87.0%
groom	79.9%	70.3%	64.9%	82.9%	82.2%	80.9%	70.0%	30.0%	57.0%
hang	95.2%	86.5%	90.0%	94.3%	95.8%	94.8%	92.0%	82.0%	91.0%
head	68.6%	57.0%	63.3%	73.0%	74.6%	73.8%	83.0%	64.0%	64.0%
rear	69.8%	60.3%	66.2%	70.8%	74.9%	72.6%	70.0%	35.0%	66.0%
rest	98.2%	86.5%	93.0%	96.3%	97.1%	88.3%	94.0%	96.0%	95.0%
walk	66.3%	55.3%	59.3%	68.5%	67.7%	65.8%	55.0%	69.0%	68.0%
all	**78.1%**	67.7%	70.8%	**79.8%**	**81.5%**	78.8%	77.3%	60.9%	71.6%

TABLE IV
COMPARISON OF TRADITIONAL FV, OUR SFV-SAN AND OUR SFV-SAN+HMM PIPELINE.

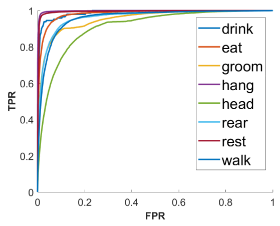
	Traditional FV pipeline				SFV-SAN pipeline				SFV-SAN+HMM pipeline			
	VF&CF	IDT	TDD	Combined	VF&CF	IDT	TDD	Combined	VF&CF	IDT	TDD	Combined
drink	25.3%	21.9%	24.7%	31.5%	37.0%	17.8%	24.0%	28.1%	59.4%	56.2%	54.8%	60.3%
eat	65.8%	62.0%	69.7%	73.0%	74.2%	72.2%	75.2%	77.2%	80.3%	78.6%	80.0%	81.1%
groom	74.3%	70.3%	64.9%	76.8%	77.9%	78.7%	70.6%	80.8%	79.9%	81.9%	80.0%	82.2%
hang	89.3%	86.5%	90.0%	92.3%	92.5%	91.1%	92.6%	94.2%	95.2%	95.0%	95.8%	95.8%
head	59.4%	57.0%	63.3%	65.1%	67.0%	68.7%	72.1%	73.7%	68.6%	70.4%	73.4%	74.6%
rear	61.7%	60.3%	66.2%	67.6%	70.3%	71.1%	73.2%	74.7%	69.8%	72.2%	73.4%	74.9%
rest	97.0%	86.5%	93.0%	94.4%	97.4%	92.1%	95.5%	95.2%	98.2%	96.6%	97.2%	97.1%
walk	57.6%	55.3%	59.3%	60.9%	64.2%	64.0%	65.0%	66.1%	66.3%	66.4%	67.0%	67.7%
all	71.7%	67.7%	70.8%	75.1%	77.0%	76.5%	76.7%	**80.0%**	78.1%	78.8%	79.3%	**81.5%**

TABLE V
COMPARISON OF DIFFERENT SYSTEMS.

Action	VF&CF(SFV-SAN+HMM)	IDT (traditional FV)	TDD (traditional FV)	TDD+IDT (SFV-SAN+HMM)	VF&CF+TDD+IDT		Stacked FV	Two-stream	Dollár
					SFV-SAN+HMM	FV+LSTM			
dig	75.8%	46.9%	49.1%	59.5%	67.2%	61.0%	46.5%	47.8%	52.1%
eat	92.1%	35.2%	43.7%	59.3%	88.0%	61.5%	31.6%	73.1%	19.1%
groom	71.8%	40.9%	42.5%	52.2%	66.3%	51.4%	38.4%	49.2%	30.7%
head	67.6%	54.5%	67.8%	76.7%	74.6%	78.0%	74.4%	64.6%	76.1%
rear	85.5%	73.6%	79.1%	84.6.0%	86.0%	84.5%	74.2%	86.2%	63.6%
walk	73.5%	62.0%	67.0%	71.6%	70.8%	66.1%	63.5%	71.0%	64.3%
all	**74.2%**	53.2%	61.2%	70.1%	**74.7%**	70.3%	61.0%	64.6%	59.1%



(a) Confusion matrix.



(b) ROC curve.

Fig. 9. Performance of our proposed system against 'Group 1'.

Fig. 9 shows the confusion matrix of our system against 'Group 1'. A confusion matrix allows the visualization of the agreement between two entities, where the diagonal cells show the percentage of the agreement with 'Group 1' and the non-diagonal cells contain the percentage of the disagreement with 'Group 1'. In Figure 9 for example, the cell value in the fourth row and fourth column indicates that our system correctly classifies 96% of the 'hanging' behaviours as labelled by human observers of 'Group 1', whereas 4% of the behaviours

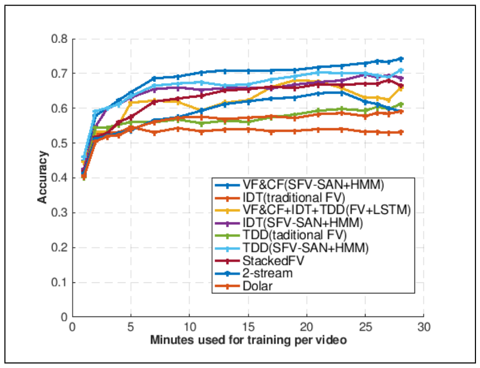


Fig. 10. The system performance with varying sizes of training examples.

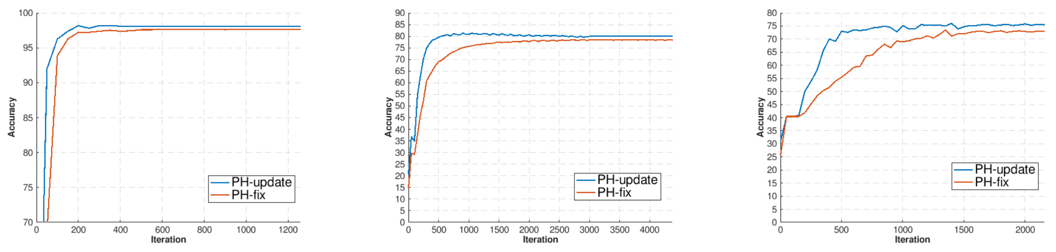
are incorrectly classified as 'rear' (2%), 'head' (1%), 'eat' (1%).

C. Experiments on our database

The aim of this experiment is threefold. First, while the proposed system has good performance on Jhuang's Database,

TABLE VI
COMPARISON OF TRADITIONAL FV, OUR SFV-SAN AND OUR SFV-SAN+HMM PIPELINE.

	Traditional FV pipeline				SFV-SAN pipeline				SFV-SAN+HMM pipeline			
	VF&CF	IDT	TDD	Combined	VF&CF	IDT	TDD	Combined	VF&CF	IDT	TDD	Combined
dig	59.6%	46.9%	49.1%	59.2%	63.5%	50.4%	52.4%	61.5%	75.8%	59.3%	60.0%	67.2%
eat	73.6%	35.2%	43.7%	75.3%	78.9%	39.8%	43.2%	78.6%	92.1%	58.2%	71.7%	88.0%
groom	59.3%	40.9%	42.5%	62.9%	64.0%	41.5%	42.9%	59.4%	71.8%	59.3%	62.2%	66.3%
head	67.2%	54.5%	67.8%	69.9%	74.3%	73.3%	76.5%	76.6%	67.6%	71.7%	73.2%	74.6%
rear	81.2%	73.6%	79.1%	82.8%	82.3%	80.6%	80.7%	83.3%	85.5%	86.2%	85.0%	86.0%
walk	65.1%	62.0%	67.0%	66.8%	67.6%	66.6%	66.1%	67.6%	73.5%	71.4%	70.0%	70.8%
all	67.2%	53.2%	61.2%	69.3%	72.2%	63.3%	65.4%	72.4%	**74.2%**	69.0%	71.0%	**74.7%**



(a) JHuang's clipped dataset. (b) JHuang's full dataset. (c) Our dataset.

Fig. 11. Impact of updating and learning projection layer

(a) eat (b) dig

Fig. 12. Example images with annotation results.

we expected the proposed system was generalised well with many different laboratory settings. For this reason, we collected and annotated a new dataset of videos with an entirely different mouse cage recorded by a different camera system (see Figure 6). Second, we also expected the proposed system could be used to recognise additional behaviours. To this end, we covered the ground of the mouse cage with sawdust so that we can collect a new behaviour i.e. dig. Third, we wanted to understand how many training examples are sufficient for good system performance. To investigate this, we systematically evaluate the performance of the system as a function of the amount of minutes available for training. We select a representative set of video segments from the first x minute of each video for training; testing is carried out on the remaining of the video from the $x-th$ minute to the end of the same video. Figure 10 shows the performance of several systems with varying sizes of the training examples. The performance of the proposed SFV-SAN+HMM pipeline with our features is consistently higher than that of the other methods including FV+LSTM pipeline. The most likely reason for the inferior performance of the LSTM is that the low quantity and imbalanced training dataset is quite a challenge to the LSTM. Besides, we also believe RNNs (e.g. LSTM based) are usually used for learning long-term dynamics of sequential data such as speech recognition and handwriting recognition. However, in our application, the change between neighbouring actions is swift in a short period, adding an extra dimensionality of technical challenges and thereby RNNs do not perform satisfactorily. After analysing the performance curve of our system, we observe that satisfactory performance can be achieved with only 7 minutes of annotation for each training video, corresponding to 90% of the performance obtained using 28 minutes of annotations (these results are the average ones of ten trials). Some example images with their annotation results are shown in Figure 12.

Table V depicts the results of each behaviour class using several baseline methods. In the conclusion of the previous experiments, our SFV-SAN+HMM pipeline with the combined features has the best result (74.7%). In Table VI, we compare the traditional FV, our SFV-SAN and our SFV-SAN+HMM pipeline with different features. In comparison to the traditional one, the SFV-SAN pipeline improves performance 6.7% and 4.8% on average and combined features. The integration of our SFV-SAN pipeline and HMM model leads to the final performance improvement of 7.0%, 15.8%, 10.0% and 5.4% for ours, IDT, TDD and combined features respectively, in comparison to the baseline algorithm. During the training of SFV-SAN or SFV-SAN+HMM, we generate the validation accuracy curves after having updated the projection layer with the PCA initialisation and after simply having taken the PCA initialisation, whilst repeating the same experiments for our dataset, JHuang's clipped and full datasets. From Fig.11, we observe that updating the projection layer is a better way to improve the validation accuracy. Furthermore, as Fig.11(b) and (c) show, updating the projection layer can boost the performance of the SFV-SAN+HMM network during the train-

TABLE VII
TIME (MIN) OF TRAINING DIFFERENT SYSTEM

Action	Combined VF&CF	IDT (traditional)	TDD (traditional)	Stacked FV	Two-stream	Dollár
CPU	191	229	882	250	/	179
GPU	/	/	674	/	1198	/

TABLE VIII
TIME (SECOND/FRAME) OF TESTING DIFFERENT SYSTEM

Action	Combined VF&CF	IDT (traditional)	TDD (traditional)	Stacked FV	Two-stream	Dollár
CPU	0.8	1.1	4.2	1.19	/	0.8
GPU	/	/	3.2	/	5.7	/

ing. For example, the network reaches the best performance after 500 and 700 iterations for JHuang's full dataset and our dataset, respectively, whilst the validation accuracy, after we simply use the PCA initialisation, is relatively lower. To compare the computational costs, we calculate the time of training and testing the individual systems using a segment of 7 minutes of each video.

All the algorithms are implemented on a PC with a 3.6-GHz Intel Core i7 processor and a 4-GB memory. To speed up the training of TDD and deep two-stream ConvNets, we employ a NVIDIA GRID M60-8Q GPU. The time cost (min) of training different systems is reported in Table VII. From Table VII, we can see that the SFV-SAN+HMM pipeline only consumes 191 minutes for training. Table VIII shows the SFV-SAN+HMM can also reach the testing speed of 0.8 second per frame. Although it is implemented on the CPU machine, it is still faster than the other compared systems except Dollár's method.

VI. Conclusion

This paper has presented an SFV-SAN+HMM framework for automated recognition of mouse behaviours. In order to estimate the emission probabilities of the HMM, an efficient hybrid architecture including a combination of SFV and SAN has been introduced. Results on the 'clips' database of Jhuang show that our feature extraction method, a key component in the unsupervised layers of our hybrid architecture, gives better accuracy than the other state-of-the-art methods. Our feature extraction method achieves weighted average accuracy of 96.5% (using visual and context features) and 97.9% (incorporated with IDT and TDD features) compared to the others that have the best accuracy of 93%. On the 'full' database of Jhuang, our SAN-SAN+HMM pipeline also obtained the best results (81.5%) with combined feature, higher than the traditional FV pipeline with IDT (67.7%)or TDD (70.8%) Jhuang's system (77.3%) and human annotation (71.6%). The experiment on our database showed the robustness of the proposed system for recognition of the additional behaviours and different settings of the mouse cage. Meanwhile, the proposed system still achieved the accuracy of 74.7% using the combined features, while the best performance of the traditional FV pipeline with TDD only reached the accuracy of 61.2% and FV+LSTM with the combined features reached the accuracy of 70.3%. Moreover, the proposed system shows a better ability than LSTM to avoid the bias towards the majority class on the imbalance dataset. Using the same dataset for training on CPU, our system is 4 times faster than the traditional FV pipeline with TDD. Even though this baseline method was implemented on a GPU, our system was still 3 times faster. Our future work will include the exploration of social interactions between multiple mice using the proposed system.

Acknowledgment

This research is supported by UK EPSRC under Grant EP/N011074/1. S. Zhang was also supported by the National Natural Science Foundation of China (No. 61872112).

References

[1] Dziegielewski and S. F, *DSM-IV-TR in Action*. John Wiley & Sons, 2010. 1

[2] L. H. Tecott and E. J. Nestler, "Neurobehavioral assessment in the information age," *Nature Neuroscience*, vol. 7, no. 5, pp. 462–466, 2004. 1

[3] D. Brunner, E. Nestler, and E. Leahy, "In need of high-throughput behavioral systems," *Drug discovery today*, vol. 7, no. 18, pp. S107–S112, 2002. 1

[4] D. Houle, D. R. Govindaraju, and S. Omholt, "Phenomics: the next challenge," *Nature reviews genetics*, vol. 11, no. 12, pp. 855–866, 2010. 1

[5] J. Askenasy, "Approaching disturbed sleep in late parkinson's disease: first step toward a proposal for a revised updrs," *Parkinsonism & related disorders*, vol. 8, no. 2, pp. 123–131, 2001. 1

[6] A. Vogel-Ciernia, D. P. Matheos, R. M. Barrett, E. A. Kramár, S. Azzawi, Y. Chen, C. N. Magnan, M. Zeller, A. Sylvain, J. Haettig *et al.*, "The neuron-specific chromatin regulatory subunit baf53b is necessary for synaptic plasticity and memory," *Nature neuroscience*, vol. 16, no. 5, pp. 552–561, 2013. 1

[7] L. Lewejohann, A. M. Hoppmann, P. Kegel, M. Kritzler, A. Krüger, and N. Sachser, "Behavioral phenotyping of a murine model of alzheimers disease in a seminaturalistic environment using rfid tracking," *Behavior research methods*, vol. 41, no. 3, pp. 850–856, 2009. 1

[8] A. V. Kalueff, A. M. Stewart, C. Song, K. C. Berridge, A. M. Graybiel, and J. C. Fentress, "Neurobiology of rodent self-grooming and its value for translational neuroscience," *Nature Reviews Neuroscience*, vol. 17, no. 1, pp. 45–59, 2016. 1

[9] R. de Heer, M. Schenke, W. Kuurman, and B. Spruijt, "Learning (in) the phenotyper: an integrative approach to conducting cognitive behavioural challenges in a home cage environment," *Measuring Behavior 2008*, p. 57, 2008. 1

[10] A. D. Steele, W. S. Jackson, O. D. King, and S. Lindquist, "The power of automated high-resolution behavior analysis revealed by its application to mouse models of huntington's and prion diseases," *Proceedings of the National Academy of Sciences*, vol. 104, no. 6, pp. 1983–1988, 2007. 1

[11] A. Weissbrod, A. Shapiro, G. Vasserman, L. Edry, M. Dayan, A. Yitzhaky, L. Hertzberg, O. Feinerman, and T. Kimchi, "Automated long-term tracking and social behavioural phenotyping of animal colonies within a semi-natural environment," *Nature communications*, vol. 4, p. ncomms3018, 2013. 1

[12] E. H. Goulding, A. K. Schenk, P. Juneja, A. W. MacKay, J. M. Wade, and L. H. Tecott, "A robust automated system elucidates mouse home cage behavioral structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 52, pp. 20 575–20 582, 2008. 1

[13] G. Dell'Omo, E. Vannoni, A. L. Vyssotski, M. A. Di Bari, R. Nonno, U. Agrimi, and H. P. Lipp, "Early behavioural changes in mice infected with bse and scrapie: automated home cage monitoring reveals prion strain differences," *European Journal of Neuroscience*, vol. 16, no. 4, pp. 735–742, 2002. 1

[14] L. P. Noldus, A. J. Spink, and R. A. Tegelenbosch, "Ethovision: a versatile video tracking system for automation of behavioral experiments," *Behavior Research Methods, Instruments, & Computers*, vol. 33, no. 3, pp. 398–414, 2001. 1

[15] H. Dankert, L. Wang, E. D. Hoopfer, D. J. Anderson, and P. Perona, "Automated monitoring and analysis of social behavior in drosophila," *Nature Methods*, vol. 6, no. 4, pp. 297–303, 2009. 1

[16] J. B. I. Rousseau, P. B. A. Van Lochem, W. H. Gispen, and B. M. Spruijt, "Classification of rat behavior with an image-processing method and a neural network," *Behavior Research Methods, Instruments, & Computers*, vol. 32, no. 1, pp. 63–71, 2000. 1

[17] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *In Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop*, 2005, pp. 65–72. 1, 4, 10

[18] H. Jhuang, E. Garrote, X. Yu, V. Khilnani, T. Poggio, A. D. Steele, and T. Serre, "Automated home-cage behavioural phenotyping of mice," *Nature communications*, vol. 1, no. 5, pp. 1–9, 2010. 1, 2, 8, 10

[19] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," in *Proc. ICCV*, 2007. 1

[20] X. P. Burgos-Artizzu, P. Dollár, D. Lin, D. J. Anderson, and P. Perona, "Social behavior recognition in continuous video," in *Proc. CVPR*, 2012. 1, 2

[21] Z. Ren, A. N. Annie, V. Ciernia, and Y. J. Lee, "Who moved my cheese? automatic annotation of rodent behaviors with convolutional neural networks," in *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*. IEEE, 2017, pp. 1277–1286. 2

[22] G. Kramida, Y. Aloimonos, C. M. Parameshwara, C. Fermüller, N. A. Francis, and P. Kanold, "Automated mouse behavior recognition using vgg features and lstm networks," in *Visual Observation and Analysis of Vertebrate And Insect Behavior Workshop (VAIB)*, 2016. 2

[23] H. Ling and S. Soatto, "Proximity distribution kernels for geometric context in category recognition," 2007. 2

[24] Y. J. Lee and K. Grauman, "Object-graphs for context-aware visual category discovery," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 346–358, 2012. 2

[25] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proc. ECCV*. Springer, 2010, pp. 143–156. 2, 4, 5, 9, 10

[26] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. CVPR*, 2015, pp. 4305–4314. 2, 5, 9, 10

[27] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action recognition with stacked fisher vectors," in *European Conference on Computer Vision*. Springer, 2014, pp. 581–595. 2, 5, 9

[28] X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *Computer Vision and Image Understanding*, vol. 150, pp. 109–125, 2016. 2, 4, 5

[29] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989. 2

[30] Y. Ephraim and N. Merhav, "Hidden markov processes," *IEEE Transactions on information theory*, vol. 48, no. 6, pp. 1518–1569, 2002. 2

[31] J. L. Gauvain and C. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, 1994. 3

[32] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech & Language*, vol. 9, no. 2, pp. 171–185, 1995. 3

[33] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, "Object trajectory-based activity classification and recognition using hidden markov models," *IEEE transactions on Image Processing*, vol. 16, no. 7, pp. 1912–1919, 2007. 3

[34] L. Piyathilaka and S. Kodagoda, "Gaussian mixture based hmm for human daily activity recognition using 3d skeleton features," in *IEEE Conference on Industrial Electronics and Applications*. IEEE, 2013, pp. 567–572. 3

[35] S. Espana-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid hmm/ann models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 767–779, 2011. 3

[36] Y. Yen, M. Fanty, and R. Cole, "Speech recognition using neural networks with forward-backward probability generated targets," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4. IEEE, 1997, pp. 3241–3244. 3

[37] C. M. Bishop, "Pattern recognition," *Machine Learning*, vol. 128, pp. 1–58, 2006. 4, 5

[38] I. Laptev, "On space-time interest points," *International journal of computer vision*, vol. 64, no. 2-3, pp. 107–123, 2005. 4

[39] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. ICCV*, 2013. 4, 9

[40] T. S. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," *Advances in Neural Information Processing Systems*, pp. 487–493, 1999. 4

[41] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep fisher networks for large-scale image classification," in *Advances in neural information processing systems*, 2013, pp. 163–171. 5, 7

[42] S. Zhang, X. Lan, H. Yao, H. Zhou, D. Tao, and X. Li, "A biologically inspired appearance model for robust visual tracking," *IEEE transactions on neural networks and learning systems*, 2016. 5

[43] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, vol. 2. IEEE, 2006, pp. 2169–2178. 5

[44] D. Lin, C. Lu, R. Liao, and J. Jia, "Learning important spatial pooling regions for scene classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3726–3733. 5

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105. 7, 9

[46] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576. 7, 9

[47] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao, "Towards good practices for very deep two-stream convnets," *arXiv preprint arXiv:1507.02159*, 2015. 7, 9

[48] H. Wang, A. Klaser, C. Schmid, and C. L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 60–79, 2013. 7, 9, 10

[49] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Proc. ECCV*, 2010. 9

[50] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013. 9

[51] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. CVPR*, 2014. 9

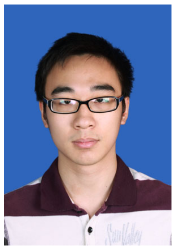
[52] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 9

[53] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9. 9

[54] O. Vinyals, S. V. Ravuri, and D. Povey, "Revisiting recurrent neural networks for robust asr," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4085–4088. 9

[55] C. Li, P. Wang, S. Wang, Y. Hou, and W. Li, "Skeleton-based action recognition using lstm and cnn," in *Multimedia & Expo Workshops (ICMEW), 2017 IEEE International Conference on*. IEEE, 2017, pp. 585–590. 9

[56] J. Roughan, S. Wright-Williams, and P. Flecknell, "Automated analysis of postoperative behaviour: assessment of homecagescan as a novel method to rapidly identify pain and analgesic effects in mice," *Laboratory animals*, vol. 43, no. 1, pp. 17–26, 2009. 11



Zheheng Jiang received the B.Sc. degree in Electrical Engineering and Automation (Grid Monitoring) from Nanjing Institute of Technology and the M.Sc. degree in Software Development from Queens University of Belfast, Belfast, U.K. He is currently pursuing the Ph.D. degree with the School of Electronic, Electrical Engineering and Computer Science, Queens University of Belfast, Belfast, U.K.

His current research interests include machine learning for vision, object detection and recognition, video analysis and event recognition.

Danny Crookes obtained a BSc (1st class honours) and PhD in 1977 and 1980 respectively. He was appointed to the Chair of Computer Engineering in 1993 at Queens University Belfast, and was Head of Computer Science from 1993-2002. He was Director of Research for Speech, Image and Vision Systems at the Institute for Electronics, Communications and Information Technology (ECIT) at Queens University Belfast. His research interests include computer visin, speech enhancement and separation, and the use of novel architectures for high performance signal processing. Professor Crookes has published over 240 scientific papers in journals and international conferences.

Dr. Brian Green is a Principal Investigator within the School of Biological Sciences. His research interests and expertise include development of therapeutic molecules for metabolic disease, and the application of metabolomic platforms for the study and diagnosis of human disease. He has published more than 200 research articles and chaired the Alzheimer's Research UK Network Centre for Northern Ireland (2013-2016). His group pioneered the use of a range of LC-MS metabolomics technologies for the study of Alzheimers and Huntingtons disease. Research projects have been funded by Diabetes UK, The British Heart Foundation, InvestNI, KTP, NIHPSS, Innovate UK, EPSRC, Alzheimers Research UK and a number of commercial companies.

Yunfeng Zhao received the B.S. degree in Computer Science from Georgia Southwestern State University and the B. Eng. degree in Network Engineering from Ningbo University of Technology. He is currently a researcher at Institute for Global Food Security and pursuing his Ph.D. degree with School of Electronics, Electrical Engineering and Computer Science, Queens University Belfast, Northern Ireland. His research interests include artificial intelligence, color and hyperspectral image analysis.

Haiping Ma received the B. S. degree from Shaoxing University, Shaoxing, China, the M. S. degree from the Taiyuan University of Technology, Taiyuan, China, and the Ph.D. degree from Shanghai University, Shanghai, China, in 2004, 2007, and 2014, respectively, all in control theory and control engineering. He is currently a visiting scholar with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, UK, and an Associate Professor with the College of Mathematics, Physics and Information, Shaoxing University. In 2015, he received the Outstanding Ph.D. Dissertation Award, Chinese Association of System Simulation, China. He has published over 30 research papers on evolutionary algorithms and applications. His current research interests include evolutionary computation, information fusion, and intelligent control. He is the author of the textbook Evolutionary Computation with Biogeography-based Optimization (John Wiley & Sons, 2017).

Dr. Ling Li is the Director of Internationalisation at the School of Computing and also the founding coordinator of Laboratory of Brain — Cognition — Computing (BC2 Lab) of the school responsible for coordinating multidisciplinary research between Computing, Sports and local NHS hospitals. She had six-year research experience at Imperial College London with a focus to understand body sensor data (EEG, EMG, ECG, eAR-sensor, and etc.). She participated in large scale projects. She also involved in projects from government and industry (i.e. Samsung GRO award). She now serves at the editorial board of Brain Informatics and the secretary of IEEE Computing Society in UK and Ireland.

Shengping Zhang received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2013. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology at Weihai. He had been a postdoctoral research associate with Brown University and with Hong Kong Baptist University, and a visiting student researcher with University of California at Berkeley. He has authored or co-authored over 50 research publications in refereed journals and conferences. His research interests include deep learning and its applications in computer vision. Dr. Zhang is also an Associate Editor of the Signal Image and Video Processing, and Journal of Electronic Imaging.

Dacheng Tao is with the UBTECH Sydney Artificial Intelligence Centre and the School of Information Technologies, the Faculty of Engineering and Information Technologies, the University of Sydney, 6 Cleveland St, Darlington, NSW 2008, Australia. Dacheng Tao (F'15) is Professor of Computer Science and ARC Laureate Fellow in the School of Information Technologies and the Faculty of Engineering and Information Technologies, and the Inaugural Director of the UBTECH Sydney Artificial Intelligence Centre, at the University of Sydney. He mainly applies statistics and mathematics to Artificial Intelligence and Data Science. His research results have expounded in one monograph and 200+ publications at prestigious journals and prominent conferences, such as IEEE T-PAMI, T-IP, T-NNLS, IJCV, JMLR, NIPS, ICML, CVPR, ICCV, ECCV, ICDM; and ACM SIGKDD, with several best paper awards, such as the best theory/algorithm paper runner up award in IEEE ICDM'07, the best student paper award in IEEE ICDM'13, the distinguished paper award in the 2018 IJCAI, the 2014 ICDM 10-year highest-impact paper award, and the 2017 IEEE Signal Processing Society Best Paper Award. He is a Fellow of the Australian Academy of Science, AAAS, IEEE, IAPR, OSA and SPIE.

Huiyu Zhou received a Bachelor of Engineering degree in Radio Technology from Huazhong University of Science and Technology of China, and a Master of Science degree in Biomedical Engineering from University of Dundee of United Kingdom, respectively. He was awarded a Doctor of Philosophy degree in Computer Vision from Heriot-Watt University, Edinburgh, United Kingdom. Dr. Zhou currently is a Reader at Department of Informatics, University of Leicester, United Kingdom. He has published over 180 peer-reviewed papers in the field. His research work has been or is being supported by UK EPSRC, MRC, EU, Royal Society, Leverhulme Trust, Puffin Trust, Invest NI and industry.