Behavior Recognition in Mouse Videos using Contextual Features Encoded by Spatial-temporal Stacked Fisher Vectors

Zheheng Jiang[1], Danny Crookes[1], Brian Desmond Green[2], Shengping Zhang[3] and Huiyu Zhou[1]
[1]*School of EEECS, Queen's University Belfast, Belfast, U.K.*
[2]*School of Biological Sciences, Queen's University Belfast, Belfast, U.K.*
[3]*School of Computer Science and Technology, Harbin Institute of Technology, Weihai, China*
{zjiang01, d.crookes, b.green}@qub.ac.uk, shengping.zhang@gmail.com, h.zhou@ecit.qub.ac.uk

Keywords: Mouse Behavior Recognition, Spatial-temporal Stacked Fisher Vector, Gaussian Mixture Model, Contextual Features, Spatio-temporal Interest Points.

Abstract: Manual measurement of mouse behavior is highly labor intensive and prone to error. This investigation aims to efficiently and accurately recognize individual mouse behaviors in action videos and continuous videos. In our system each mouse action video is expressed as the collection of a set of interest points. We extract both appearance and contextual features from the interest points collected from the training datasets, and then obtain two Gaussian Mixture Model (GMM) dictionaries for the visual and contextual features. The two GMM dictionaries are leveraged by our spatial-temporal stacked Fisher Vector (FV) to represent each mouse action video. A neural network is used to classify mouse action and finally applied to annotate continuous video. The novelty of our proposed approach is: (i) our method exploits contextual features from spatio-temporal interest points, leading to enhanced performance, (ii) we encode contextual features and then fuse them with appearance features, and (iii) location information of a mouse is extracted from spatio-temporal interest points to support mouse behavior recognition. We evaluate our method against the database of Jhuang et al. (Jhuang et al., 2010) and the results show that our method outperforms several state-of-the-art approaches.

1 INTRODUCTION

Mice are extensively employed in biomedical science and their responses to disease or therapy are frequently detected by measurement of their behavior patterns. In most cases this monitoring is performed manually using video recordings. Recording of diverse behaviors of home-cage mice generates a large amount of information for researchers (Steele et al., 2007; Roughan et al., 2009) in pathology, psychology, ethology, neuroscience and medicine. However, manual annotation of mouse recordings is a highly labor intensive process which is error-prone and subject to individual interpretation. Furthermore, human observers may fail to detect behavioral events that are very quick or too slow, and humans may miss events because of dwindling attention span.

In the literature some systems which automatically recognize animal behaviors have been described. For instance, Rousseau et al. (Rousseau et al., 2000) were the first to show that the detection of specific behaviors was possible. They applied neural network techniques to recognize 9 solitary rat behaviors from body shape and position, 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. (Dollár et al., 2005) recognized mouse behavior using the classification of sparse spatio-temporal features. However, they only considered visual features of the interest points (e.g. image gradient) without the contextual information such as the spatial relationship between different interest points. In 2010 Jhuang et al. (Jhuang et al., 2010) used background subtraction to get a subwindow of the mouse in each frame from the side-view. From the mouse subwindow, the features that they used were generated based on a computational model of motion processing in the human brain (Jhuang et al., 2007), followed by classification using a Hidden Markov Model Support Vector Machine (SVMHMM). Their method to locate the mouse is dependent on a good background model, which it turns out can be problematic. Recently, Burgos-Artizzu et al. (Burgos-

Jiang, Z., Crookes, D., Green, B., Zhang, S. and Zhou, H.
Behavior Recognition in Mouse Videos using Contextual Features Encoded by Spatial-temporal Stacked Fisher Vectors.
DOI: 10.5220/0006244602590269
In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2017)*, pages 259-269
ISBN: 978-989-758-222-6

Artizzu et al., 2012) created a system for recognizing the social behavior of mice, both from the top and side views. They applied AdaBoost with spatio-temporal and trajectory features to classify mouse behaviors. As with the method of Dollár et al. (Dollár et al., 2005), this method also ignored the spatio-temporal contextual features. Furthermore, their trajectory features are based on a tracking algorithm which was not detailed in their paper.

A common feature of all of the above studies is that the location information of mice is computed by tracking (Rousseau et al., 2000); (Burgos-Artizzu et al., 2012) or detection (Jhuang et al., 2007) algorithms. Also, their extracted features are derived from studies of human behavior recognition, such as spatio-temporal, trajectory and shape features. Low-level local features have become popular in action recognition due to their robustness to background noise and independence of the detection and tracking algorithms. Among these local features, spatio-temporal interest points (Dollár et al., 2005); (Laptev, 2005) and Improved Dense Trajectories (IDT) (Wang and Schmid, 2013); (Wang et al., 2015) are widely used because of their ease of use and good performance. Spatio-temporal interest points are used by some systems (Dollár et al., 2005); (Laptev, 2005) to extract visual features around interest points, but contextual features also imply a large amount of information about spatial location and temporal changes of the mouse.

In our system, we propose to exploit contextual features of interest points, which also potentially describe mouse location without using an independent tracking or detecting algorithm. These features are then encoded as spatial-temporal stacked Fisher vectors which are the input to the neural network. The main contributions of this study are:

1. We improve upon the performance of Dollar's interest point detector especially under illumine-tion using frame differencing and Laplacian of Gaussian filtering.
2. We explore new contextual features from the spatio-temporal interest points for behavior recognition. It is the first attempt to encode this contextual feature rather than simply concatenate them after appearance features like (Jhuang et al., 2010), (Burgos-Artizzu et al., 2012) and (Laptev, 2005). Our contextual features are an important feature which can characterize both spatial location and temporal changes in mice. We compute the absolute and relative positions of each interest point and then concatenate them to form the contextual features.
3. We compute spatial-temporal stacked Fisher vectors for both contextual and visual features that help improve behavior recognition accuracy. We generate two GMM dictionaries for contextu-al and visual features respectively and then compute spatial-temporal stacked Fisher vectors for each of them.
4. We conduct a comprehensive evaluation of the proposed algorithm, and compare it with several state-of-the-art techniques.

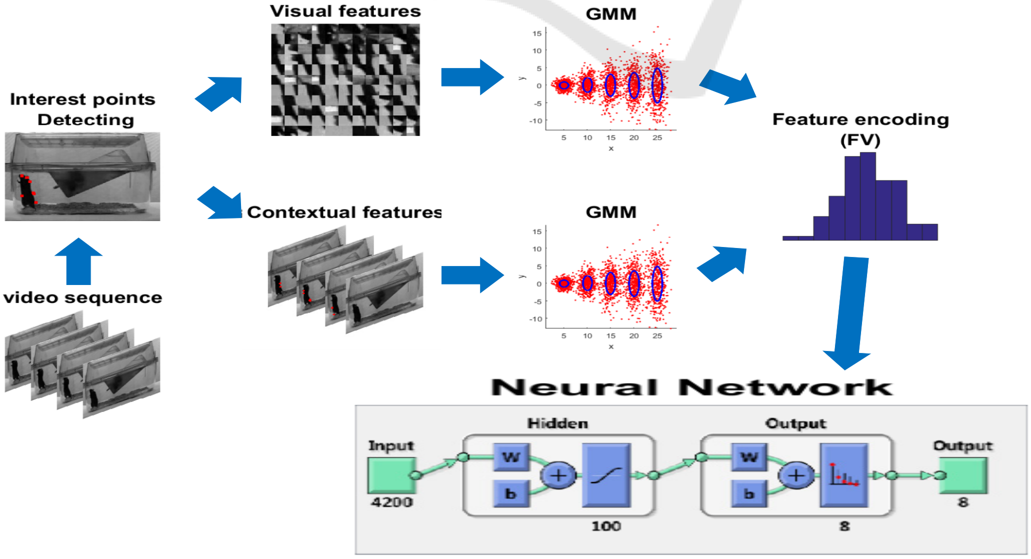


Figure 1: The proposed framework.

2 FRAMEWORK OF OUR APPROACH

As shown in Fig. 1 the pipeline for our method consists of five steps: (i) detection of interest points, (ii) description of interest points, (iii) generation of two Gaussian Mixture Model (GMM) dictionaries, (iv) feature encoding with spatial-temporal stacked Fisher vector (FV), and (v) classification with a neural network (NN). In the following sections, we will describe each step in more detail.

2.1 Detection of Interest Points

Interest points are local spatio-temporal features considered to be salient or characteristic of the action captured in a 3D spatio-temporal volume (see Fig. 2). Spatio-temporal interest points are those points where the local neighborhood has a significant variation in both the spatial and the temporal domains. Laptev (Laptev, 2005) extended the 2D Harris corner detector to 3D. However the main drawback of this method is the relatively small number of stable interest points. Willems et al. (Willems et al., 2008) identify saliency as the determinant of a 3D Hessian matrix, which is faster and denser than Harris 3D but less dense than Dollar's detector. Another trend is to use dense sampling (Wang and Schmid, 2013), which extracts video blocks at regular positions and scales in space and time. Obviously dense sampling is able to produce many more interest points than the above detector. However, it is more difficult to ensure that all interest points are on the object. Among various interest point detection methods, the one proposed by Dollar et al. (Dollár et al., 2005) is perhaps the most suitable for mouse action recognition. They calculate a response function to locate interest points. Their response function has the form:

$$\boldsymbol{R} = (\boldsymbol{I} * \boldsymbol{g} * \boldsymbol{h_{ev}}(\boldsymbol{t}))^2 + (\boldsymbol{I} \ * \ \boldsymbol{g} * \boldsymbol{h_{od}}(\boldsymbol{t}))^2 \quad (1)$$

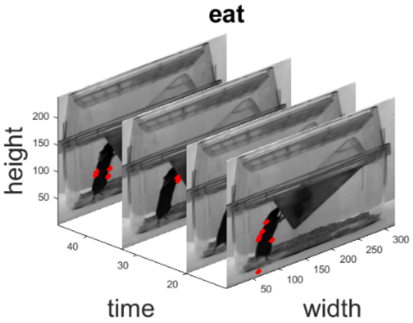


Figure 2: Some examples of detected interest points (depicted by red dots) in a 3D spatio-temporal volume.

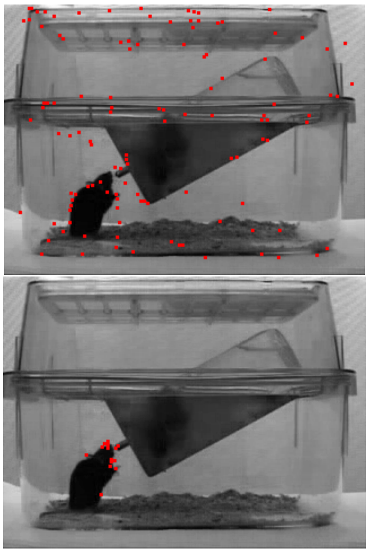


Figure 3: Comparison between interest points detected using our detector (bottom) and the Dollar detector (top) under illumination change.

where g is the 2D Gaussian smoothing kernel which is applied only along the spatial dimensions, and $\boldsymbol{h_{ev}}$ and $\boldsymbol{h_{od}}$ are a quadrature pair of 1D Gabor filters applied temporally, defined as: $\boldsymbol{h_{ev}}(\boldsymbol{t};\boldsymbol{\tau},\boldsymbol{w}) = -\boldsymbol{cos}(2\boldsymbol{\pi t w})\boldsymbol{e}^{-\boldsymbol{t}^2/\boldsymbol{\tau}^2}$, and $\boldsymbol{h_{od}}(\boldsymbol{t};\boldsymbol{\tau},\boldsymbol{w}) = -\boldsymbol{sin}(2\boldsymbol{\pi t w})\boldsymbol{e}^{-\boldsymbol{t}^2/\boldsymbol{\tau}^2}$.

Despite this method's popularity, since it uses solely local information within a small region, it is prone to false detection under illumination variation; it also tends to generate spurious interest points around highly textured background areas. Some drawbacks are highlighted in the examples in Fig. 3.

To overcome these shortcomings, we propose here a different interest point detector. In particular, although the 1-D Gabor filter applied in the temporal domain is effective for capturing the dynamics of actions, it is sensitive to both illumination and highly textured background. To overcome this problem, the proposed detector explores different filters for detecting salient spatio-temporal interest points. More specifically, our detector consists of two steps: 1) Laplacian of the Gaussian filtering in the spatial domain replacing single Gaussian in Dollar et al. (Dollár et al., 2005) for reducing the influence of illumination and 2) frame differencing for eliminating spurious interest points on the background. This two-step approach facilitates saliency detection in both the temporal and spatial domains to give a combined filter response. Hence our response function has the form:

$$\boldsymbol{R} = (\boldsymbol{I} * \boldsymbol{g} * \boldsymbol{L} * \boldsymbol{h_{ev}}(\boldsymbol{t}))^2 + (\boldsymbol{I} \ * \ \boldsymbol{g} * \boldsymbol{L} * \boldsymbol{h_{od}}(\boldsymbol{t}))^2 \quad (2)$$

in which $\boldsymbol{L}$ is the operator of Laplace used on the space. Due to introduction of the Laplace operator, our interest points detector can effectively reduce the influence of illumination, followed by frame differencing for ensuring all interest points are on the mouse. Fig. 3 also shows that, under the same illumination change, our detector can extract more precise interest points on the mouse.

2.2 Contextual and Visual Features

2.2.1 Spatio-temporal Contextual Feature Extraction

Most existing mouse behavior recognition systems (Jhuang et al., 2010; Rousseau et al., 2000; Burgos-Artizzu et al., 2012) extract position features from established trackers. However these tracking algorithms do not seem to be very reliable. For example, Jhuang et al. (Jhuang et al., 2010) used background subtraction to get a subwindow of the mouse in each frame, but their foreground detection algorithm assumes the background is constant, which cannot be guaranteed in a real experiment. Besides, trajectory features extracted by Burgos-Artizzu et al. (Burgos-Artizzu et al., 2012) are from their undetailed tracking algorithm, and the mouse nose tracking algorithm applied by Rousseau et al. (Rousseau et al., 2000) seems insensitive and unreliable. Unlike their approach, we propose a novel method to extract contextual information from the detected interest points, which also imply the location of the mouse without using any extra mouse tracking or detection algorithm.

Our spatio-temporal contextual information of interest points is an important action representation, because they characterize both spatial location and temporal changes of the mouse. There are two types of features that are computed: the relative position, and absolute spatial position of interest points. The position of each interest point in the 3D spatio-temporal volume is represented by its XYT coordinates. Fig. 2 intuitively shows the distribution of interest points. Suppose there are R interest points detected in an action video. In order to compute relative positions, we firstly compute a center interest point defined by: $[\boldsymbol{X_c}; \boldsymbol{Y_c}; \boldsymbol{T_c}] = \frac{\boldsymbol{1}}{\boldsymbol{R}} * \sum_{i=1}^{R} [\boldsymbol{X_i}; \boldsymbol{Y_i}; \boldsymbol{T_i}]$, where $[\boldsymbol{X_c}; \boldsymbol{Y_c}; \boldsymbol{T_c}]$ and $[\boldsymbol{X_i}; \boldsymbol{Y_i}; \boldsymbol{T_i}]$ represent the coordinates of the center and the i^{th} interest point respectively in an action video. Consequently, the relative position of interest points is represented by the coordinates of R interest points relative to the center interest point: $\boldsymbol{P_i} = [\boldsymbol{X_i} - \boldsymbol{X_c};\ \boldsymbol{Y_i} - \ \boldsymbol{Y_c};\ \boldsymbol{T_i} - \boldsymbol{T_c}], \boldsymbol{i} = \boldsymbol{1}, \boldsymbol{2}, \dots, \boldsymbol{R}.$

Using relative position efficiently describes the distribution in the 3D spatio-temporal volume, because it concentrates on different behavior patterns while ignoring outliers. The absolute spatial position of each interest point is able to characterize the place where the action happens (which can be important for location-based behaviors such as drinking). To capture this information, we concatenate the XY coordinates to the relative position. Overall, the contextual feature vector has the form: $\boldsymbol{F_i} = [\boldsymbol{X_i} - \boldsymbol{X_c};\ \boldsymbol{Y_i} - \ \boldsymbol{Y_c};\ \boldsymbol{T_i} - \boldsymbol{T_c};\ \boldsymbol{X_i};\ \boldsymbol{Y_i}], \boldsymbol{i} = \boldsymbol{1}, \boldsymbol{2}, \dots, \boldsymbol{R}.$

2.2.2 Spatio-temporal Visual Feature Extraction

After detecting the interest points, we extract the visual features (see Fig. 4) from the cuboids around the interest points in the 3D spatio-temporal volume. For simplicity, we extract the brightness gradients with three channels (*G_x*, G_y, G_t) from each cuboid and flatten the cuboid into a vector as (Dollár et al., 2005). To eliminate noise and retain some principle information, Principle Component Analysis (PCA) is used to reduce the dimensionality of the visual feature vector.

2.3 Generation of GMM Dictionaries for Contextual and Visual Features

The aim of dictionary generation is to describe the local feature space and provide a partition for local descriptors (Peng et al. 2014). In some existing mouse behavior systems (Dollár et al., 2005; Burgos-Artizzu et al., 2012), a mouse action is modeled as a bag of independent and unordered visual words; however, the spatio-temporal contextual information of interest points is ignored. In these approaches, the k-means clustering algorithm is used to construct the dictionary. In our work, instead of k-means, we use Gaussian Mixture Model (GMM), which is a probabilistic model to characterize the distribution of the given feature space.

For each type of dictionary, we suppose a *K*-component GMM, and each Gaussian *k* has the form (Perronnin et al. 2010):

$$\boldsymbol{u_k} = \frac{\boldsymbol{1}}{\sqrt{(\boldsymbol{2\pi})^{\boldsymbol{D}}|\boldsymbol{\Sigma_k}|}} \boldsymbol{exp}\left\{-\frac{\boldsymbol{1}}{\boldsymbol{2}}(\boldsymbol{x} - \ \boldsymbol{\mu_k})^{\mathrm{T}} \boldsymbol{\Sigma_k^{-1}} (\boldsymbol{x} - \boldsymbol{\mu_k})\right\} \tag{3}$$

where $\boldsymbol{\mu_k}$ and $\boldsymbol{\Sigma_k}$ are the D dimension of mean vector and diagonal covariance matrix respectively, $\boldsymbol{k} = \boldsymbol{1}, \boldsymbol{2}, \dots, \boldsymbol{K}$. Then the GMM can be defined as:

$$\boldsymbol{p(x;\theta)} = \sum_{\boldsymbol{k=1}}^{\boldsymbol{K}} \boldsymbol{\omega_k u_k(x;\mu_k,\Sigma_k)} \qquad (4)$$

where $\boldsymbol{\theta = \{\omega_k,\mu_k,\Sigma_k,\mathrm{k}=1,\dots,\mathrm{K}\}}$, $\boldsymbol{\omega_k}$ is the mixture weight of Gaussian $\boldsymbol{u_k}$ and subject to $\forall_{\boldsymbol{k}}\colon \boldsymbol{\omega_k \geq 0}, \boldsymbol{\sum_{k=1}^{K}\omega_k = 1}$.

Given the feature set $\mathbf{X} = \{\boldsymbol{x_1},\dots,\boldsymbol{x_M}\}$, we apply the Expectation-Maximization (EM) algorithm to optimize parameters of GMM, which is learned through maximum likelihood (Bishop, 2006).

There are two benefits in our approach with two dictionaries: (1) The two dictionaries consider both contextual and visual features of interest points. (2) Unlike k-means, GMM delivers not only the mean information of code words, but also the shape of their distribution.

2.4 Feature Encoding and Fusion

Feature encoding aims to leverage the dictionary to integrate all local descriptors into a feature vector, which can ensure all video clips have the same dimension of feature vector, and efficiently improve classification performance. Although feature encoding and fusion are very important procedures in mouse action recognition, related papers discussing this are rare. For example, (Dollár et al., 2005) and (Laptev, 2005) only use the traditional encoding method of Vector Quantization. For feature fusion, some existing mouse behavior recognition systems (Jhuang et al., 2010; Rousseau et al., 2000; Burgos-Artizzu et al., 2012) simply append positional features after appearance features without encoding. In our opinion, appearance features and contextual features are two different kinds of feature and vary in value range. So it is more reasonable to encode them separately. In recent evaluations (Peng et al., 2014; Chatfield et al., 2011; Wang et al., 2009), the Fisher Vector performs consistently better than bag of features, where it is popular to encode features for both image and video classification. We also apply this encoding method and show that it can improve the performance of our features as well (see section 4.1). Unlike bag of features, Fisher Vector leverages GMM as its dictionary for encoding more information than the mean of code words. It calculates the gradient of the log-likelihood with respect to a parameter of GMM, which can describe how that parameter contributes to the process of generating a particular example (Perronnin et al. 2010). Let $X = \{x_n, \mathrm{n} = 1 \dots N\}$ be the set of N descriptors of interest points in an action video. Then this video can be represented by the gradient vector of log likelihood (Jaakkola et al. 1999):

$$\boldsymbol{G_\theta^X} = \frac{\boldsymbol{1}}{\boldsymbol{N}}\boldsymbol{\nabla_\theta logp(X;\theta)} \qquad (5)$$

where $p(X;\theta) = \prod_{n=1}^{N} p(x_n;\theta)$ and θ is the parameter of this function. This is a generative model to characterize an action video with a gradient vector derived from a probability density function. On the basis of this generative model, Perronnin et al. (Perronnin et al. 2010) introduced the GMM to replace the probability density function $p(x_n;\theta)$ and developed an improved Fisher vector as follows:

$$\boldsymbol{\mathcal{G}_{\mu,k}^X} = \frac{\boldsymbol{1}}{\boldsymbol{N\sqrt{\omega_k}}}\sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \boldsymbol{\Upsilon_n(k)}\left(\frac{\boldsymbol{x_n - \mu_k}}{\boldsymbol{\sigma_k}}\right) \qquad (6)$$

$$\boldsymbol{\mathcal{G}_{\sigma,k}^X} = \frac{\boldsymbol{1}}{\boldsymbol{N}\sqrt{\omega_k}}\sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \boldsymbol{\Upsilon_n(k)}[\frac{\boldsymbol{(x_n - \mu_k)^2}}{\boldsymbol{\sigma_k^2}} - \boldsymbol{1}] \qquad (7)$$

where σ_k^2 has D dimensions and represents the diagonal covariance matrices, *i.e.* the diagonal of $\boldsymbol{\Sigma_k}$. In other words, $\boldsymbol{\mathcal{G}_{\mu,k}^X}$ and $\boldsymbol{\mathcal{G}_{\sigma,k}^X}$ are the *D*-dimensional gradients with respect to the mean $\boldsymbol{\mu_k}$ and standard deviation $\boldsymbol{\sigma_k}$ of Gaussian k. Eqs. (6) and (7) are the mathematical derivations of Eq. (4) replacing the $p(x_n;\theta)$ of GMM. In addition, $\boldsymbol{\Upsilon_n(k)}$ is the weight of $x_n$to the *k^th^* Gaussian:

$$\boldsymbol{\Upsilon_n(k)} = \frac{\boldsymbol{\omega_k u_k(x_n;\mu_k,\Sigma_k)}}{\boldsymbol{\sum_{k=1}^{K}\omega_k u_k(x_n;\mu_k,\Sigma_k)}} \qquad (8)$$

If we suppose there are *K* Gaussians and *D* dimensions of a descriptor after performing PCA in our system, then the Fisher vector is the concatenation of $\boldsymbol{\mathcal{G}_{\mu,k}^X}$ and $\boldsymbol{\mathcal{G}_{\sigma,k}^X}$ with a total of 2*KD* vector dimensions, which describes how the parameters of the generative model $p(X;\theta)$ should be modified to better fit the data X.

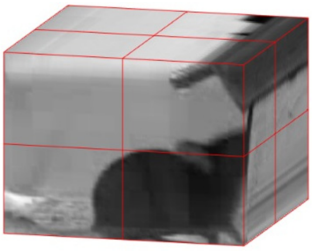


Figure 4: Spatial-temporal stacking.

In our approach, as mentioned in section 2.2, we have two GMM dictionaries, one for visual and the other for contextual features, so we can compute two Fisher vectors for both of them. Note that local sum-pooling, which is in the form of (6) and (7), is agnostic to the relative location of aggregated features. To capture the spatial-temporal structure within each feature's neighborhood, inspired by spatial stacking of (Simonyan et al., 2013), we incorporate the stacking sub-layer, which concatenates the spatial-temporal adjacent features in the 2*2*2 cuboid which encompasses all the detected interest points (Fig. 4). After normalizing these spatial-temporal stacked Fisher vector by power and L2 normalization, we fuse contextual and appearance Fisher vectors to give the input to the classifier. In particular, contextual and appearance are complementary features, and they jointly boost the performance of the classifier (see Section 4.2).

2.5 Classification with a Neural Network

In our study, the fusion FV of contextual and visual features is the final feature vector which needs to be classified. Although FV are designed to work well with a linear classifier (because they correspond to the explicit feature map of the Fisher Kernel), we still wanted to understand whether FV classification could be improved with non-linear classifiers. Perronnin et al. (Perronnin et al., 2015) recently gave a positive answer for image classification. In our experiment, we also apply a non-linear classifier and find that a combination of neural networks and Fisher vectors can give better results than the conventional combination of Fisher vector and linear support vector machine (SVM). In our approach, a feed-forward neural network was constructed with two layers of sigmoid hidden neurons and softmax output neurons. The feed-forward neural network allows the one-way transmission of the data from input to output. The hidden layer was fed with the fused and normalized Fisher Vector of contextual and visual features. In each layer, the input is weighted and transformed by an activation function (sigmoid in the hidden layer and softmax in the output layer) and is then passed to the neurons in the next layer.

3 EXPERIMENTAL SETUP

3.1 Benchmark Experiment

To quantify the improvement obtained by our spatio-temporal visual and contextual features, we compared our method first to the state-of-the-art approach using Trajectory Features proposed by Wang et al. (Wang et al. 2015), because it has been a popular action representation in recent years, albeit for human behaviors.

3.1.1 Spatio-temporal Visual and Contextual Features

The interest point detector used in this experiment was proposed by Dollar et al. (Dollár et al., 2005). For parameter setting, the spatial and temporal scale parameters σ and τ are empirically set to 2 and 3, respectively. After detecting the interest points, we extract XYT relative and absolute locations of each interest point. Afterwards we construct a visual feature vector using brightness gradients from cuboids, which are centered on the interest points and have default size 13*13*19. To eliminate noise and retain some principle information, Principle Component Analysis (PCA) is then used to reduce the dimensionality of visual feature vector by preserving 98% of the energy.

3.1.2 Trajectory Features

Improved Dense Trajectories (IDT) (Wang et al., 2015) is another widely used local feature. This approach densely samples points in each frame. Tracking points are achieved using optical flow. We used the default trajectory length of 15 frames. For each trajectory, we computed descriptors of Trajectory, HOG, HOF and MBH (Wang et al., 2013). The Trajectory descriptor describes its shape by a sequence of displacement vectors. The other descriptors are computed in the spatio-temporal volume aligned with the trajectory. HOG represents the static appearance information by the orientation of image gradients. Both HOF and MBH measure motion information, and are based on optical flow. HOF directly quantizes the orientation of flow vectors. MBH quantizes the derivatives by splitting the optical flow into horizontal and vertical components. The final dimensions of the descriptors are 30 for Trajectory, 96 for HOG, 108 for HOF and 192 for MBH.

3.2 Feature Encoding and Classification

To encode features, we compared bag of features and Fisher vector. We used 1500 randomly sampled features with k-means to train a codebook for each descriptor type including HOG, HOF, MBH, spatio-temporal visual and contextual features. The size of the codebook is set to K=50. Unlike bag of features, Fisher vector (Perronnin et al. 2010) encodes both first and second order statistics between the video descriptors and a Gaussian Mixture Model (GMM). In order to estimate the GMM for each descriptor, we randomly sample 1500 features from the training set and set the number of Gaussians to K=20. Each descriptor type has 2KD dimensional Fisher vector as described in (Perronnin et al. 2010). To normalize a Fisher vector, we apply power and L2 normalization as in (Perronnin et al. 2010). Finally we concatenate normalized Fisher vectors of different descriptor types and compare the performance of different combinations of them.

For classification, we use neural network (NN), linear SVM, radial basis function SVM and K-nearest neighbor (kNN) for comparing the performance of the trajectory features with our spatio-temporal features. For parameter settings of each classification method, we fix the number of hidden nodes in NN to 100, use a one-against-the-rest strategy for designing multi-class classification of SVM and set K=1 in kNN. For the other parameters, we follow the default setup in Matlab. After the experiments, we choose the best results as the evidence of the comparison and analysis. In all experiments we divided all datasets into two parts: half is used for training and half for testing. Additionally, to evaluate our system on continuous videos, we used leave-one-out procedure on a frame-by-frame comparison with human ground truth. During the leave-one-out procedure, all except one video are used to train a neural network and the trained neural network was used to test the one remaining video. The procedure is repeated n times for all videos and the average performance is reported.

3.3 Datasets

The Jhuang database (Jhuang et al., 2010**)** was used for our experimental test. The first type of database called the ‘clipped database’ contains 4200 clips in which only the best instances of specific behaviors are included. This dataset is the largest of the current publicly available datasets. It consists of eight mouse behavior classes: rear (399 cases), groom (1477), eat (374), drink (61), hang (521), rest (868), walk (233) and head (180). Each clip records a single mouse from a side-view camera. The second database denoted as the ‘full database’ involves 12 frame-by-frame labeled videos lasting over 10 hours in total. In order to make the recognition system more robust during the training process, they varied the camera angles and lighting conditions. They also used many mice of different size, gender, and coat color in experiments. In this paper, experiments of 4.1, 4.2, 4.3 and 4.4 are measured on the ‘clipped database’ using a half-by-half cross-validation procedure. The ‘full database’ is used to train and test our system evaluated by a leave-one-out strategy in the last experiment.

4 EXPERIMENTAL RESULTS

4.1 Comparison with Trajectory Features

In this section, we evaluated the performance of our visual features (VF) and contextual features (CF) using different feature encoding methods, compared with the state-of-the-art IDT features approach. Table 1 compares the final performance of the different features. In Table 1, we can observe that the combined features have better accuracy than just one. However, for IDT features, trajectory shapes seem not to be suitable for mouse behavior recognition. The reason may be that differences between behaviors can be subtle, and the trajectory shape may not give enough fine detail. The results also show that IDT features without trajectory shapes have better performance than with trajectory shapes (93.4% vs 92.6%). Furthermore, a Fisher vector representation always results in a better performance than bag of features for each type of feature and combined features. Taking all the results together it is clear that visual features and contextual features give best results and their combination provides the best overall accuracy (95.9% compared with 93.4% for IDT features.

Table 1: Comparison of the performance (accuracy %) of IDT features and spatio-temporal features.

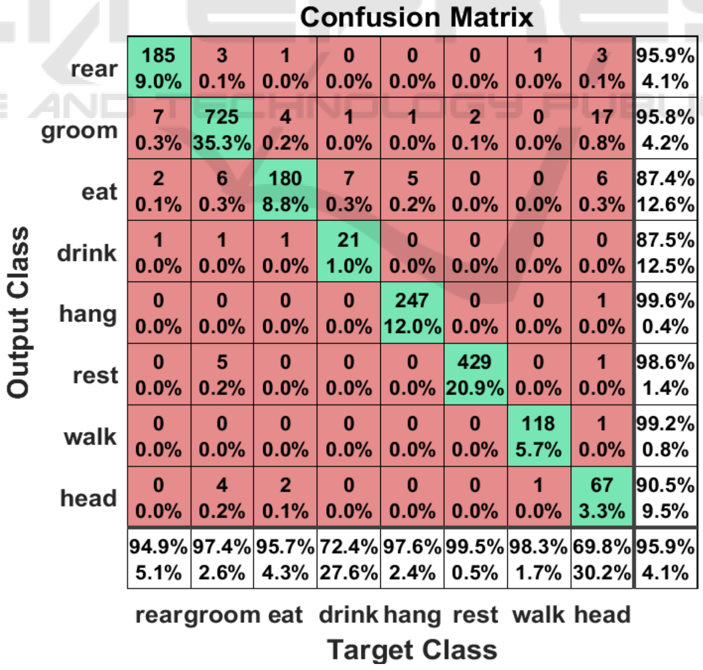
Features	BOF+NN	FV+linear SVM	FV+NN
IDT			
Trajectory	69.1%	73.6%	73.3%
HOG	84.8%	91.6%	91.9%
HOF	77.2%	83.2%	84.9%
MBH	79.3%	87.9%	89.5%
Combined with trajectory	85.5%	91.9%	**92.6%**
Combined without trajectory	88.5%	92.3%	**93.4%**
Spatio-temporal			
Visual features	87.3%	91.4%	91.3%
Contextual features	89.4%	92.2%	93.0%
Combined	**93.1%**	**95.4%**	**95.9%**

4.2 Evaluation of Spatio-temporal Visual and Contextual Features on Specific Behaviors

Table 2 compares the performance of spatio-temporal visual and contextual features for specific behaviors. This experiment is tested on the same feature encoding and classification (FV+NN). In Table 2, we see that, except for “walk”, “head” and “groom”, the contextual features seem to result in better accuracy. The possible explanation is that the contextual features are more effective for distinguishing behaviors which are more localized, such as “eat”, “drink”, “rear” and “hang”. These often happen near the feeder, tube, wall and ceiling respectively. Although the interest points of “groom”, “walk”, “rest”, “groom” and “head” can happen at any place except the ceiling, each behavior has a particular distribution in both the spatial and temporal domains. So this contextual distribution can also contain evidence to help distinguish behaviors. However in the ROC curve (see Fig. 6) of contextual features, the performance for “head” is obviously worse than for other behaviors. “Head” is easily confused with similar spatio-temporal contextual and visual information. The small proportion of “drink” in the dataset also influences the final accuracy; it is reasonable to suppose that if we had more “drink” action videos for training (see section 4.4), the accuracy would be greatly improved. We also note that the combined features are able to achieve significantly higher accuracy for each behavior than either the contextual and visual features on their own. Fig. 5 shows the confusion matrix for the combined features for more detail.

Table 2: Comparison of the performance (accuracy %) of visual features, contextual features and their combination.

Action	visual features	contextual features	combined features
rear	83.1%	84.0%	94.9%
groom	96.2%	96.2%	97.4%
eat	76.8%	87.5%	95.7%
drink	56.3%	84.8%	72.4%
hang	93.6%	96.3%	97.6%
rest	98.8%	99.1%	99.5%
walk	98.2%	96.5%	98.3%
head	64.5%	61.5%	69.8%
all	91.3%	93.0%	95.9%



	rear	groom	eat	drink	hang	rest	walk	head	
rear	185 9.0%	3 0.1%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	3 0.1%	95.9% 4.1%
groom	7 0.3%	725 35.3%	4 0.2%	1 0.0%	1 0.0%	2 0.1%	0 0.0%	17 0.8%	95.8% 4.2%
eat	2 0.1%	6 0.3%	180 8.8%	7 0.3%	5 0.2%	0 0.0%	0 0.0%	6 0.3%	87.4% 12.6%
drink	1 0.0%	1 0.0%	1 0.0%	21 1.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	87.5% 12.5%
hang	0 0.0%	0 0.0%	0 0.0%	0 0.0%	247 12.0%	0 0.0%	0 0.0%	1 0.0%	99.6% 0.4%
rest	0 0.0%	5 0.2%	0 0.0%	0 0.0%	0 0.0%	429 20.9%	0 0.0%	1 0.0%	98.6% 1.4%
walk	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	118 5.7%	1 0.0%	99.2% 0.8%
head	0 0.0%	4 0.2%	2 0.1%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	67 3.3%	90.5% 9.5%
	94.9% 5.1%	97.4% 2.6%	95.7% 4.3%	72.4% 27.6%	97.6% 2.4%	99.5% 0.5%	98.3% 1.7%	69.8% 30.2%	95.9% 4.1%

Figure 5: The confusion matrix for the combination of visual and contextual 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.

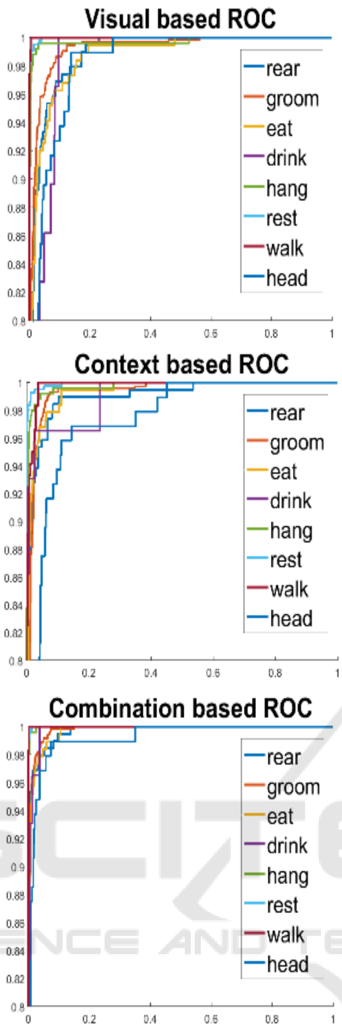


Figure 6: The ROC curve of visual features visual features, contextual features and their combination.

4.3 Evaluation of Classification

In this section we compare our combined spatio-temporal (ST) features with combined IDT features using different encoding methods and classifiers. Table 3 shows that our combined ST features always outperform the IDT features using different combinations of encoding methods and classifiers. The table also shows that the combination of FV and NN or linear SVM appears to achieve higher accuracy for both combined IDT features and combined ST features. Moreover, the combination of FV+NN, FV+linear SVM and BOF+NN have better results than the others. The results also suggest that the performance of FV+NN is a little better than FV+linear SVM which is used to classify IDT features in (Wang et al. 2015). Typically the selection of the SVM kernel is based on experience.

Table 3: Comparison of results (accuracy %) using different encoding methods and classifiers.

Classification	IDT (no trajectory)	IDT (with trajectory)	our ST
FV + NN	**93.4%**	92.6%	**95.9%**
FV + linear SVM	**92.3%**	91.9%	**95.4%**
FV + RBF SVM	90.7%	88.6%	91.1%
BOF+NN	88.5%	85.5%	**93.1%**
BOF+kNN	79.4%	78.4%	90.9%
BOF+ linear SVM	87.1%	86.8%	92.4%
BOF + RBF SVM	85.9%	84.7%	92.7%

However, NN seems to be more robust to different encoding methods, because regardless of the features and encoding method used in our experiment the NN generally outperforms the other classifiers.

4.4 Comparison with State-of-the-art

In this section we compare our method to the method proposed by Dollar et al. (Dollár et al., 2005) and Wang et al. (Wang et al. 2015) for each specific mouse behavior. We use the same validation strategy (half-by-half) for each state-of-the art method and compare the results in Table 4. Interestingly, all methods, including ours, struggle to recognize “drink” and “head”. In particular, the method proposed by (Wang et al. 2015) achieves very low accuracy (5%). The most likely reason is that “drink” and “head” have only a small proportion of the training set (1.5% and 4.3% respectively). We also see that the trajectory features including trajectory shapes and descriptors used by (Wang et al. 2015) cannot correctly represent “drink” behavior, because their interest points detecting method (Improving Dense trajectory) struggles to detect useful feature points

Table 4: Comparison of accuracy with state-of-the-art methods.

Action	Dollar	Wang	our method	Jhuang
rear	57.9%	89.7%	94.9%	-
groom	88.4%	96.2%	97.4%	-
eat	69.0%	88.8%	95.7%	-
drink	41.0%	5.0%	72.4%	-
hang	80.8%	96.9%	97.6%	-
rest	98.8%	95.8%	99.5%	-
walk	96.1%	97.0%	98.3%	-
head	32.2%	64.8%	69.8%	-
all	82.2%	92.3%	95.9%	93%

from the drinking mouse which maintains its posture but uses only its mouth (see Fig. 2). Overall, our method significantly outperforms the current state-of-the-art methods for each specific mouse behavior. In terms of the final accuracy our method has an improvement of 13.7%, 3.6% and 2.9% over (Dollár et al., 2005), (Wang et al. 2015) and (Jhuang et al., 2010), respectively.

4.5 Continuous Video Annotation

To annotate continuous videos, sliding windows are centered at each frame and both appearance features and contextual features are computed inside them. Once spatio-temporal features are computed for all the sliding windows, Fisher vector is then computed for each frame by focusing on a sliding window centered in the current frame. These fisher vectors are finally classified by a trained neural network and their classification results are regarded as labels of all the frames. To explore an optimal sliding window size, we establish an experiment to compare the percentage agreements with human annotation using different sliding window sizes, illustrated in Fig. 7.

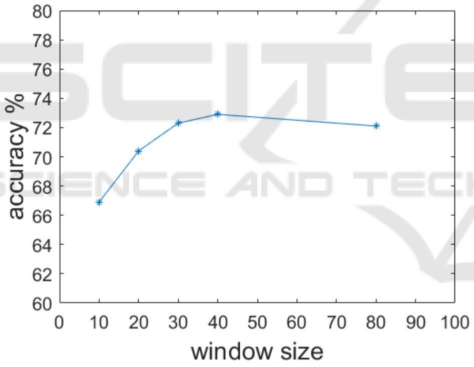


Figure 7: Continuous video annotation with different window sizes.

5 CONCLUSION

This paper has presented a new approach to automatically recognizing specific mouse behaviors. We show that our interest detector is stable under illumination. Our appearance and contextual fusion features encoded by spatial-temporal stacked fisher vector significantly outperform the other state-of-the-art features. Also, the combination of Fisher vector and neural networks improves the performance of our system and gives higher accuracy than the other state-of-the art systems. Overall, our method achieves an average of 95.9% accuracy compared to the previous best test of 93%. Final experiments on annotation of continuous video also obtain results (72.9%) that are on a par with human annotation, which is evaluated as 71.6% in (Jhuang et al., 2010). Future work will include exploring more distinguishing features, combining temporal model and extending the range of behaviors. We also plan to study social behavior between multiple mice.

ACKNOWLEDGEMENT

This project is supported by UK EPSRC under Grant EP/N011074/1 and National Natural Science Foundation of China under Grant 61300111.

REFERENCES

Bishop, C. M. (2006). In *Pattern Recognition and Machine Learning*.

Burgos-Artizzu, X. P., Dollár, P., Lin, D., Anderson, D. J., and Perona, P. (2012, June). In *Social behavior recognition in continuous video*. IEEE Conference on Computer Vision and Pattern Recognition.

Chatfield, K., Lempitsky, V. S., Vedaldi, A., and Zisserman, A. (2011, September). In *The devil is in the details: an evaluation of recent feature encoding methods*. British Machine Vision Conference (Vol. 2, No. 4, p. 8).

Dankert, H., Wang, L., Hoopfer, E. D., Anderson, D. J., and Perona, P. (2009). In *Automated monitoring and analysis of social behavior in Drosophila*. Nature methods, 6(4), 297-303.

Dollár, P., Rabaud, V., Cottrell, G., and Belongie, S. (2005, October). In *Behavior recognition via sparse spatio-temporal features*. 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance.

Jaakkola, T. S., and Haussler, D. (1999). In *Exploiting generative models in discriminative classifiers*. Advances in neural information processing systems, 487-493.

Jhuang, H., Garrote, E., Yu, X., Khilnani, V., Poggio, T., Steele, A. D., and Serre, T. (2010). In *Automated home-cage behavioural phenotyping of mice*. Nature communications, 1, 68.

Jhuang, H., Serre, T., Wolf, L., and Poggio, T. (2007, October). In *A biologically inspired system for action recognition*. IEEE 11th International Conference on Computer Vision (pp. 1-8).

Laptev, I. (2005). In *On space-time interest points*. International Journal of Computer Vision, 64(2-3), 107-123.

Roughan, J. V., Wright-Williams, S. L., and Flecknell, P. A. (2009). In *Automated analysis of postoperative behaviour: assessment of HomeCageScan as a novel* *method to rapidly identify pain and analgesic effects in mice.* Laboratory animals, 43(1), 17-26.

Rousseau, J. B. I., Van Lochem, P. B. A., Gispen, W. H., and Spruijt, B. M. (2000). In *Classification of rat behavior with an image-processing method and a neural network*. Behavior Research Methods, Instruments, and Computers,32(1), 63-71.

Sánchez, J., Perronnin, F., Mensink, T., and Verbeek, J. (2013). In *Image classification with the fisher vector: Theory and practice.* International journal of computer vision, 105(3), 222-245.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). In *Deep fisher networks for large-scale image classification*. Advances in neural information processing systems (pp. 163-171).

Steele, A. D., Jackson, W. S., King, O. D., and Lindquist, S. (2007). In *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, 104(6), 1983-1988.

Perronnin, F., and Larlus, D. (2015). In *Fisher vectors meet neural networks: A hybrid classification architecture.* Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3743-3752).

Perronnin, F., Sánchez, J., and Mensink, T. (2010). In *Improving the fisher kernel for large-scale image classification.* European conference on computer vision (pp. 143-156). Springer Berlin Heidelberg.

Peng, X., Wang, L., Wang, X., and Qiao, Y. (2014). In *Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice*. arXiv preprint arXiv:1405.4506.

Peng, X., Wang, L., Qiao, Y., and Peng, Q. (2014, August). In *A joint evaluation of dictionary learning and feature encoding for action recognition*. 22nd International Conference on Pattern Recognition (pp. 2607-2612).

Wang, H., Kläser, A., Schmid, C., and Liu, C. L. (2013). In *Dense trajectories and motion boundary descriptors for action recognition*. International journal of computer vision, 103(1), 60-79.

Wang, H., and Schmid, C. (2013). In *Action recognition with improved trajectories*. Proceedings of the IEEE International Conference on Computer Vision (pp. 3551-3558).

Wang, H., Ullah, M. M., Klaser, A., Laptev, I., and Schmid, C. (2009). In *Evaluation of local spatio-temporal features for action recognition*. British Machine Vision Conference (pp. 124-1).

Wang, H., Oneata, D., Verbeek, J., and Schmid, C. (2015). In *A robust and efficient video representation for action recognition*. International Journal of Computer Vision, 1-20.

Willems, G., Tuytelaars, T., and Van Gool, L. (2008). In *An efficient dense and scale-invariant spatio-temporal interest point detector*. European conference on computer vision (pp. 650-663). Springer Berlin Heidelberg.