Review

Can MiceGPT be True? A Survey on AI-empowered Mice Behavior Analysis Applications and Solutions

Chaopeng Guo , Yuming Chen , Chengxia Ma , Shuang Hao * , Jie Song *

Posted Date: 5 July 2023

doi: 10.20944/preprints202307.0271.v1

Keywords: Mice Behavior Analysis; Mice Model; AI; Computer Vision

Review

Can MiceGPT Be True? A Survey on AI-empowered Mice Behavior Analysis Applications and Solutions

Chaopeng Guo [1], Yuming Chen [1], Chengxia Ma [2], Shuang Hao [2] * and Jie Song [1,] *

[1] Software College, Northeastern University, Shenyang 110169, China; guochaopeng@swc.neu.edu.cn (C.G.); 2290126@stu.neu.edu.cn (Y.C.)
[2] College of Life and Health Sciences, Northeastern University, Shenyang 110169, China; 2101365@mail.neu.edu.cn (C.M.)
* Correspondence: haoshuang@mail.neu.edu.cn (H.S.); songjie@mail.neu.edu.cn (J.S.)

Abstract: Mice are one of the frequently used animal models in science research whose behavioral characteristics can provide much valuable information in biology, neuroscience, and pharmacology. Nowadays, artificial intelligence is widely used in mice behavior analysis. Integrated AI systems such as ChatGPT and VisualGPT are already available, and we discuss the feasibility of MiceGPT to help researchers identify and classify mouse behavior more easily. We review the applications of mice behavior analysis, analyze the tasks of deep learning on these applications based on an AI pyramid, and finally summarize the AI approaches to solve these tasks. Based on these summaries, we propose three MiceGPT architectures to demonstrate the theoretical feasibility of MiceGPT.

Keywords: mice behavior analysis; mice model; AI; computer vision

1. Introduction

Mice are one of the animal models in the biology and medical fields. It has been used for many years and has many advantages, including similarity to humans in many physiological functions and many methods of functional intervention through genetic modification. Researchers conducted various experiments on mice and observed the experimental phenomena of mice for biological and medical study, such as gene identification [1], cell classification [2] and protein prediction [3]. Among the in vivo and in vitro experiments, mice behavior analysis is an essential topic and plays key roles in the medicine, neuroscience, biology, genetics, and educational psychology field. For example, researchers study behavioral patterns of mice to investigate the effect of a gene mutation, understand the efficacy of potential pharmacological therapies, or uncover the neural underpinnings of behavior for further treatment of mental disorders. Nowadays, mice behavior analysis has become a common approach in a wide range of biomedical research fields.

In the early stages of research, traditional behavioral analysis approaches allow for quantification of behavior by tracking the animal's position in space, such as three-chamber assay [4], open-field arena [5] and water maze [6]. However, with the development of technologies, traditional approaches face challenges in emphasizing important details of behavior involving subtle actions [7]. Fine-grained behavioral feature data cannot be obtained through visual observation or subjective evaluation. Traditional approaches are time-consuming on high-precision feature computation work, and the results are also variable [8]. A novel, automated, quantifiable approach for extracting fine-grained behavioral features is essential. Along with the development of the artificial intelligence (AI) field, AI can learn from large amounts of data and extract quantitative features automatically. "AI-empowered" has become a research and application trend today. Researchers also have applied AI to mice behavior analysis by analyzing the video or video frame data, such as by machine learning methods [9] and by deep learning methods [10]. AI empowers mice behavior analysis and makes some creative research possible now.

Recently, with the rapid spread of ChatGPT [11], a more convenient and intelligent AI system has become a popular trend in the AI field. Compared with traditional AI studies, a GPT-integrated

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

system can fulfill various objectives, such as translation, Q&A, dialogue, and text generation. However, there are no AI systems like MiceGPT for biology-related researchers. The Researchers must choose appropriate methods from a wide range of AI approaches to accomplish their research. This not only fails to demonstrate the convenience of AI but also increases their extra study tasks, which reduces research efficiency. Therefore, biology-related researchers require a system like **"MiceGPT"**, shown in Figure 1, in which it contains diverse mice behavior analysis apps combined with lots of state-of-arts AI models. Researchers can input their query requirements of analysis in the system, and MiceGPT can automatically classify the queries into specific applications in AI methods, and divide the application into the AI task, which is trained by different state-of-the-art AI models with mice behavior data, and finally response the query results to the researchers. So, can a "MiceGPT" be true? Concretely, what applications of mice behavior analysis can "MiceGPT" support? What tasks can the applications be divided into? What AI models can empower the tasks? We want to answer these questions in this paper.

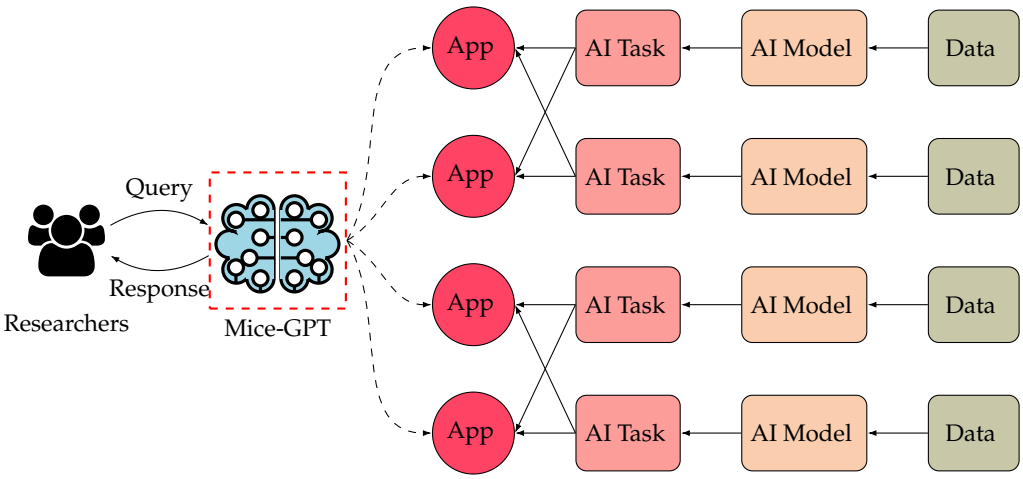

Figure 1. MiceGPT Overview.

This paper aims to make a survey to answer the above questions. Based on Figure 1, we summarize the applications of mice behavior analysis, classify the applications into several well-known tasks of the AI field, and propose state-of-the-art AI-empowered approaches to solve the tasks. Finally, we propose our prospect architecture on "MiceGPT" with the content of the survey. We also propose two improved MiceGPT architectures with state-of-the-art Natural Language Processing (NLP) and AI generation technologies.

The rest of the paper is as follows: Section2 introduces our motivations for this survey. Section 3 summarizes all the applications on the mice behavior analysis and proposes the relationship between applications and AI tasks. Section 4 summarizes the suitable AI-empowered task approaches. Section 5 introduces the iteration of MiceGPT's architecture. Section 6 concludes the paper.

2. Research Questions

This section introduces the main research questions of the survey. We first retrieve the AI-based papers of mice behavior analysis to ensure that all the studies in this survey are all AI-based. The paper starts with a general question of "Can MiceGPT be true", which is subsequently divided into four Research Questions (RQs) based on Figure 1. The paper answers the RQs through literature surveys and makes summaries. Research questions include:

- **RQ1:** What applications can AI empowers in the mice behavior analysis studies? (Answered in Section 3)
- **RQ2:** How to taxonomize the applications into AI tasks? (Answered in Section 3)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

- **RQ3:** What AI methods can be used for executing AI tasks? (Answered in Section 4)
- **RQ4:** How can MiceGPT trains the AI methods, classify the AI tasks, and identify the applications? (Answered in Section 5)

3. Applications

In this section, we conduct a preliminary search about mice behavior with AI approaches using the Google Scholar and SCI Expanded library with the keywords "mice behavior AND machine learning AND deep learning". In Google Scholar, the keywords are chiefly matched in the body of papers instead of the abstract, and the search results contain the patents and research reports. They are not our main focus. In the SCI Expanded library, we search the same keywords in the title, abstract, and keywords. The search scope is "Article AND Meetings." The initial number of retrieved documents amounted to around 85 publications. We selected 26 papers as state-of-the-art works, according to the following rules:

- Including studies whose data are videos or video frames;
- Including studies that have exact application goals instead of technical goals;
- Excluding studies that focus on machine learning instead of deep learning;

In the end, we obtained 26 related papers and grouped them into four applications. This section summarizes the state-of-art AI-empowered mice behavior research on applications to summarize and taxonomize AI-empowered mice behavior analysis applications for the further study of MiceGPT.

3.1. Disease Detection

Changes in daily human behavior (e.g., food intake, sleep, and activity patterns) can often reflect symptoms of several diseases. Mice disease models [12,13] are a valuable resource in studying the diseases [14]. However, these studies require long and systematic observations of disease-carrying mice, which requires much labor work and is subject to human error. Fortunately, AI can be a powerful tool for diagnosing disease in mice [15–17]. As shown in Figure 2, mice behaviors, such as scraching and gait, are recorded as video data with high-speed cameras. AI methods, such as semantic segmentation, pose estimation, and action recognition, diagnose disease in mice through the video data. AI provides new insights into the pathophysiology and treatment of diseases.

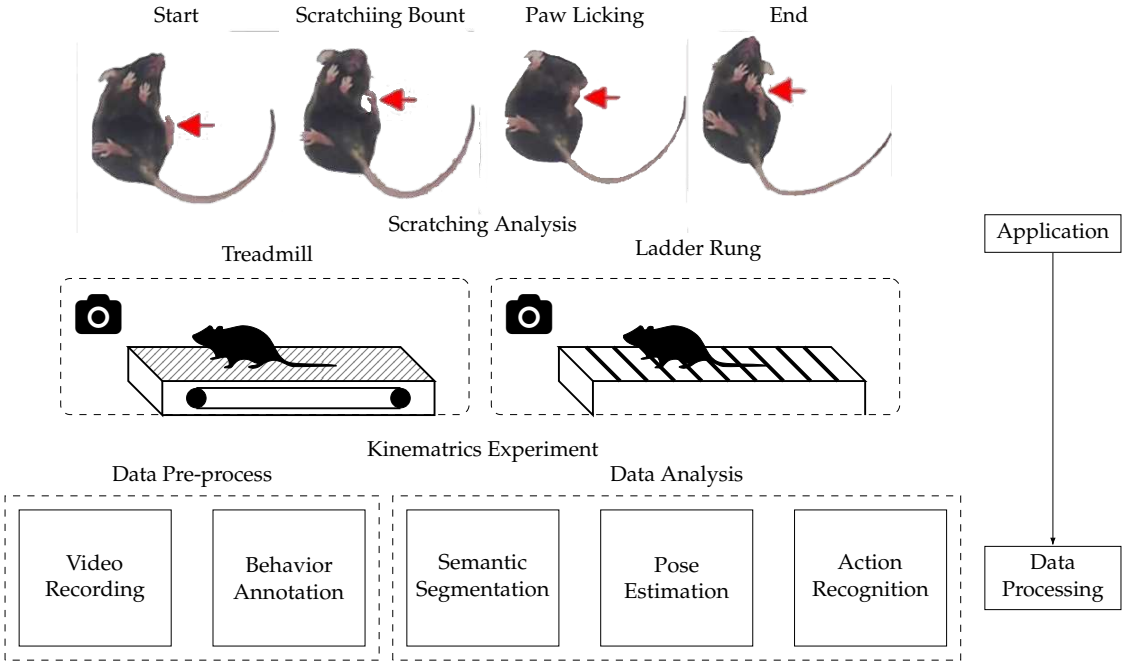

Figure 2. Disease Example.

 doi:10.20944/preprints202307.0271.v1

Most of the existing studies on AI-empowered mice disease detection are based on video data, while a few are based on text data. Yu et al. [18] record the mice behavior from the bottom of a mouse videotaping box with a camera. Compared to the top or side views, the bottom view can clearly capture the key body parts involved in scratching behavior. Weber et al. [19] customize a free-walking runway with two mirrors that allow 3D recording of the mice from the lateral/side and down perspectives. Aljovic et al. [20] film all videos with a GoPro 8 camera positioned parallel to and at a fixed distance and angle from the treadmill and ladder. Alexandrov et al. [9] generate a large, content-rich behavioral data set using a series of HET Htt CAG-repeat-KI mice with a range of CAG repeat lengths, assessed at different ages.

Weber et al. [19] reveal gait abnormalities and motor deficits in rodents after a focal ischemic stroke with key point detection and pose estimation based on deep learning. They provide a comprehensive 3D gait analysis of mice. They further refined the widely used ladder rung test using deep learning and compared its performance to human annotators. The results show that deep learning-based motion tracking with comprehensive post-analysis provides accurate and sensitive data to describe the complex recovery of rodents following a stroke.

Yu et al. [18] develop a new system, Scratch-AID (Automatic Itch Detection), based on image classification and action recognition. The system could automatically identify and quantify mice scratching behavior with high accuracy. They design a CRNN (Convolutional Recurrent Neural Network) by combining CNN (Convolutional Neural Network) and RNN (Recurrent Neural Network). The CNN extracts static features, and the RNN extracts dynamic features. Finally, a classifier combines the features extracted by both CNN and RNN and generates the prediction output (scratching or non-scratching). The best-trained network achieves 97.6% recall and 96.9% precision on test videos.

Sakamoto et al. [16] develop an accurate automated prediction method for black mice with image classification and action recognition. Same with Yu et.al's research [18], they also used CRNN. The CRNN outputs a decimal value between zero and one for pre-processed images. They define an image whose value is more than 0.5 as "scratching". They set a posterior filter that removes the predictions for nine or fewer frames, which could easily be wrong, to improve the predictive performance. The results show that the established CRNN and posterior filter successfully predicted the scratching behavior in black mice.

Aljovic et al. [20] develop an open-source computational "toolbox" with pose estimation and image classification functions. The toolbox can be applied to neurological conditions affecting the brain and spinal cord. The toolbox is based upon pose estimation obtained from DeepLabCut [21]. It can be used for automated kinematic parameter computation, automated footfall detection, and kinematic data analysis with random forest classification and principal component analysis. The results show that the automated comprehensive analysis could delineate the specific parameters of the locomotor function that are best suited to track injuries of the brain or spinal cord or are sensitive enough to predict disease onset during the prodromal phase of a multiple sclerosis model.

Alexandrov et al. [9] use a computational method based on SVMs (Support Vector Machines) to analyze the large-scale phenotypic information generated by the three systems. They select the phenotypes that best-distinguished mice with CAG repeats of different lengths. The final model, which incorporates about 200 behavioral features, accurately predicts the CAG-repeat length of a blinded mouse line. The results demonstrate the potential to predict underlying disease mutations by measuring subtle variations at the level of behavioral phenotypes.

3.2. External Stimuli Effective Assessment

External stimuli effective assessment is a basic experiment approach for mice. Compared with the mice without the stimuli, researchers make external stimuli on the specific mice organ to analyze the stimuli effect by analyzing mice behaviors. The types of external stimuli are various, such as medicine [22], artificial stimulus [23], and genetic alteration [24]. Due to the high-speed behavior of mice, traditional approaches cannot exactly obtain the video frames with the complete organ. With

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

the development of AI technology, researchers apply the AI-empowered approaches to evaluate the external stimuli effect on mice automatically, which presents the basic research steps in Figure 3. Researchers make various external stimuli on the mice, such as drug stimuli, artificial stimuli and gene knockout to observe the actions of mice. With the AI techniques empowering, the AI models extract the features from the video timeline, and make classification, detection and tracing tasks by the train process. Then the researchers can get the expected outputs from the AI models. Studying external stimuli effect in mice can contribute to exploring disease treatment and neuroscience. They generally focus on the detection, classification, segmentation, and tracing tasks.

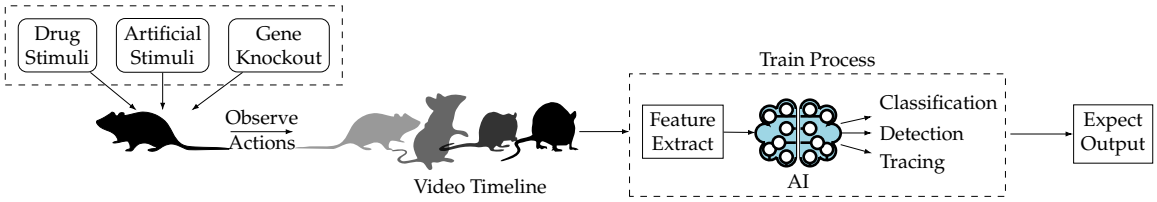

Figure 3. External Stimuli Example.

Current AI-empowered studies on external stimuli effective assessment adopt video data as the training and testing data. Wotton et al. [25] collect video data of mice behavior in response to a hind paw formalin injection. Kathote et al. [26] record the bottom view videos of the mice behavior with acetazolamide and baclofen. Vidal et al. [27] create a video database including the behavioral data of 8 different white-haired mice collected multiple times at different times. Abdus-Saboor et al. [28] use high-speed videography to record sub-second, full-body move videos. Marks et al. [29] collect raw video frames in complex environments directly. Torabi et al. [30] collect neonatal (10-days-old) rat pup video recordings using standard locomotor-derived kinematic measures. Martins et al. [31] collected videos of the tail suspension test (TST) in a controlled environment. Wang et al. [32] collect mice behavior with an overhead camera during video recording.

Wotton et al. [25] aim to make key point detection and licking action recognition of mice and propose an automated rating system for rapid, yet clinically relevant nociception assays in the formalin assay. They take advantage of the key point detection by DeepLabCut [21] with a pre-trained ResNet50 [33], and use the GentleBoost classifier to identify the behavior of licking of each frame. The results show that the automated system easily scores over 80 videos and reveals strain differences in both response timing and amplitude.

Vidal et al. [27] focus on automating the prediction of the grimace scale on white-furred mice by AI-empowered object detection, semantic segmentation, and image classification. They create a video database including the behavioral data of 8 different white-haired mice collected multiple times, use YOLO to detect frames that provide a stable frontal face of the mice, and propose a Dilated CNN to segment the mice eyes region and a Grimace Scale Prediction Network to classify the grimace scale into dilatation, activation, and dropout. The results show that this process is possible to differentiate among the pain scale of the mice.

Abdus-Saboor et al. [28] analyze sub-second behavioral features following hind paw stimulation with both noxious and innocuous stimuli to assess pain sensation in mice by AI-empowered action recognition. They apply four mechanical stimuli to the plantar surface of a randomly chosen hind paw of fully acclimated mice, apply machine learning to make classifies withdrawal action behaviors as a probability of being pain-like, and obtain the probability by regression analysis. The results indicated that a sensitive pain sensation assessment could be feasibly achieved based on the calibration of the animal's own behavior.

 doi:10.20944/preprints202307.0271.v1

Kathote et al. [26] develop an AI-empowered pose estimation method to quantify Glucose transporter 1 deficiency syndrome mice behavior to infer potential therapeutic value on cancer. They make automation of pose estimation by deep neural networks to analyze more subtle changes that the drugs may potentially cause, use K-means to cluster and select usable frames, and train these frames for automated tracking of body parts in the recorded videos. The results indicate that this in vivo approach can estimate preclinical suitability from the perspective of G1D locomotion.

Marks et al. [29] propose a novel deep learning architecture to study brain function, the effects of pharmacological interventions, and genetic alterations by quantification of behaviors. The architecture consists of four neural networks. It made instance segmentation to find the mask and bounding box for each animal by SegNet. Based on the segmentation, the architecture can make key point detection by PoseNet, object tracing by IdNet, and action recognition by BehaveNet based on different types of input data. The results show that the architecture successfully recognized multiple behaviors of freely moving individual mice and socially interacting non-human primates in three dimensions.

Torabi et al. [30] study the effect of maternal nicotine exposure before conception on 10-day-old rat pup motor behavior and propose a deep neural network by action recognition. They train the model for classifying the videos into maternal preconception nicotine exposed groups and control them. The results suggest novel findings that maternal preconception nicotine exposure delays and alters offspring motor development.

Martins et al. [31] develop a novel computerized approach, based on AI and video analysis of the experimentation procedure, to standardize the TST by object detection and action classification. They propose a CNN network to detect the bounding boxes of the rear paws in the videos. Based on this, they apply some machine learning techniques to classify the movement status of the rodent, such as SVMs, decision trees, and kNNs (k-nearest neighbors). The results show that the CNN achieved 87.7% success in the paw identification problem, and the classifier achieved 95% accuracy in classifying the animal's mobility states.

Wang et al. [32] seek to develop a hybrid machine learning workflow to understand the brain more by accurate and effective quantification of animal behavior. They use DeepLabCut to trace the mice body key points and detect the mice behavior during a video period by random forest and hidden Markov model models. The results show that the workflow represented a balanced approach for improving the depth and reliability of machine learning classifiers in chemosensory and other behavioral contexts.

3.3. Social Behavior Analysis

The study of social behavior in mice [34,35] holds significant importance in the field of medicine. By gaining a deep understanding of the neurobiological basis of social behavior in mice, people can unravel the mechanisms underlying social behavioral disorders and provide clues for the diagnosis and treatment of related diseases [36,37]. Additionally, research has revealed the impact of social stress and stress on social behavior in mice, highlighting the interaction between stress and social behavior and offering new strategies for treating stress-related disorders. The study of social behavior in mice also contributes to exploring the influence of social interaction on health, providing important clues to understanding the association between social isolation and health issues. Social behavior analysis in mice generally includes object detection, key point detection, post estimation, action recognition, and other tasks, as shown in Figure 4. The process mainly consists of two steps: data extraction and data analysis. In the data extraction step, the researchers estimate the posture of the mice in the video data and extract the mouse trajectories from the image data. In the data analysis step, they extract the pose features and classify behaviors. Finally, the results are visualized.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

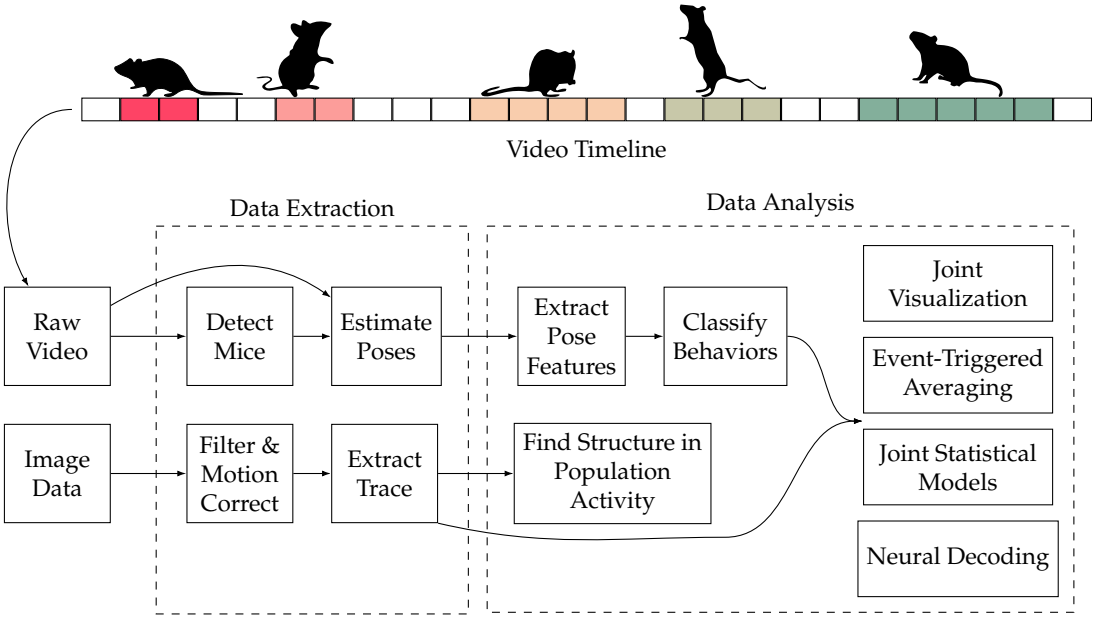

Figure 4. Social Behavior Example.

Current studies of social behavior in mice are vision-based, which means they study the behaviors of mice by analyzing the video data of mice activity. Video data are classified as single-view or multi-view. The studies that rely on analyzing single-view video recordings [38,39] can be ambiguous when the basic information about the behavior is occluded. Multi-view video can provide more behavioral information about mice, which is easier to identify their behavioral characteristics. Therefore, multi-view video recordings for mouse observations are increasingly receiving much attention [40–43].

Segalin et al. [38] introduce the Mouse Action Recognition System (MARS), an automated pose estimation and behavior quantification pipeline in pairs of freely interacting mice. MARS achieves human-level performance in pose estimation and behavior classification. Moreover, it uses computer vision to track and detect the pose of the mice and the XGBoost [44] algorithm to classify their behavior. The authors also provide custom Python code to train novel behavior classifiers.

Agbele et al. [39] present a system that uses local binary patterns and cascade AdaBoost [45] classifier to detect and classify mice behavioral movement in videos with minimal supervision, helping animal behaviorists in their research by providing a non-invasive and non-intrusive way to study mice behavior. The developed cascade AdaBoost algorithm was able to detect eight different mice movements.

Winters et al. [46] present a new automated method for assessing maternal care in laboratory mice using machine learning algorithms and aim to improve the reliability and reproducibility of the pup retrieval test performance assessment. The results show that the proposed automated procedure was able to estimate retrieval success with an accuracy of 86.7%. They bred primiparous c57bl/6JRJ mice and housed them in groups for time-controlled breeding in standard type II cages. They use the puppy retrieval test to evaluate puppy-oriented maternal care in laboratory mice. Automatic tracking of dams and one pup is established in DeepLabCut, and “maternal approach”, “handling” and "digging" for automatic behavioral classification are established in simple behavior analysis.

Jiang et al. [40] propose a novel multi-view latent-attention and dynamic discriminative model for identifying social behaviors from various views. The proposed model outperforms other state-of-the-art technologies and effectively solves the imbalanced data problem. The model jointly learns view-specific and view-shared sub-structures, where the former captures the unique dynamics of each view while the latter encodes the interaction between the views. Additionally, a multi-view latent-attention variational autoencoder model is introduced in learning the acquired features, enabling

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

them to learn discriminative features in each view. Also, the graphical model models the correlation between the neighboring labels, which has shown superior performance in recognizing mouse behaviors in a long video recording.

Hong et al. [47] present a new integrated hardware and software system for automatically estimating pose and classifying social behaviors involving close and dynamic interactions between two mice. The experiment proves that their integrated approach allows for rapid, automated measurement of social behaviors and allows the ability to develop new, objective behavioral metrics. They design a hardware setup and software to produce an accurate representation and segmentation of inaccurately represented segments. Then they develop a computer vision tool that extracts a representation of the location and body pose (orientation, posture, etc.) of individual animals and use the representation to train a supervised machine learning algorithm to detect specific social behaviors.

Burgos-Artizzu et al. [42] present a novel method for analyzing social behavior in continuous videos by segmenting them into action "bouts" using a temporal context model that combines features from spatio-temporal energy and agent trajectories. The method is tested on a dataset of videos of interacting pairs of mice, reaching a mean recognition rate of 61.2% compared to the expert's agreement rate of about 70%. The authors find that their novel trajectory features, used in a discriminative framework, are more informative than widely used spatio-temporal features. Furthermore, temporal context plays an important role in action recognition of continuous videos. The authors compare their method with other approaches and show that their approach outperforms them regarding recognition rate.

Tanas et al. [48] discuss using multidimensional analysis to evaluate the behavioral phenotype of mice with Angelman syndrome and wild-type littermates. The approach was able to predict the genotype of mice based on their behavioral profile with high accuracy and detect behavioral improvement as a function of treatment in Angelman syndrome model mice. They define multidimensional analysis as the multi-step process of (a) reducing the dimensionality of large behavioral datasets using principal component analysis, (b) clustering data in principal component space using k-means clustering, and (c) assessing whether behaviorally defined clusters align with animal genotype.

3.4. Neurobehavioral Assessment

Neurology is a major research direction of biology and medical science. Traditional approaches measure the representational mice behavior information for the neurologic study, such as dynamic weight-bearing test [49], metabolic parameters test [50] and grip strength test [51]. However, some micro features of mice behaviors can promote the research of neurology, which can not be discovered by manual observation. Therefore, researchers apply AI methods in analyzing certain mice behaviors to study mice's nervous systems further. To make the neurobehavioral assessment, researchers commonly collect the video of mice behavior, then transfer the video data into image frames, and make AI models for training the images for classification, segmentation, key point detection, and context action prediction.

In the neurobehavioral assessment, all the studies collect video data and divide videos into image frames to train AI models. Ren et al. [52], Jiang et al. [53], and Tong et al. [54] collected mice action behavior videos. Geuther et al. [10] collected the mice sleep behavior videos. Cai et al. [55] recorded the mice freezing behavior videos. Jhuang et al. [56] provided software and an extensive manually annotated video database for data training and testing. Lara-Dona et al. [57] collected the mice pupil behavior videos of both eyes.

Ren et al. [52] find that automated annotation of mice behavior could help study the neuroscience of long-term memory in mice. Then they treat the annotation task as a per-frame image classification problem and fine-tune a powerful CNN network pre-trained on ImageNet for recognizing annotate animal behaviors automatically to save human annotation costs. The results show that the powerful CNN can provides more accurate annotations than alternate automatic methods.

 doi:10.20944/preprints202307.0271.v1

Cai et al. [55] study the reward & punishment mechanism of dopamine neurons by mice freezing behaviors, and eliminate the need for human scoring by pre-trained ResNet. They further train on the pixel-by-pixel intensity difference between consecutive pairs of frames and classify each frame into a certain behavior classification. The results show that each classifier achieved optimal training within 50 training epochs and yielded 92–96% accuracy.

Jhuang et. al [56] aime to make the neurobehavioural analysis of mice phenotypes and classify every frame of a video sequence by semantic segmentation and image classification, even for those frames that are ambiguous and difficult to categorize. They first made the semantic segmentation to get the foreground mask by the background subtraction procedure. Then they train and test a multiclass SVM model on single isolated frames to recognize high-quality unambiguous behavior. The results show that their model can lead to 93% accuracy, which is significantly higher than the performance of a representative computer vision system.

Lara-Dona et al. [57] analyze the changes in pupil diameter by semantic segmentation, which reflects neural activity in the locus coeruleus. They built up the SOLOv2 to segment mice pupils from each photo frame, and output the range of mice pupils. The results confirm a high accuracy that makes the system suitable for real-time pupil size tracking.

Geuther et al. [10] treat the nerve signals and the mice behavior videos to analyze mice's sleep quality. They segment the mice mask from the video and use the human expert-scored EEG/EMG data to train a visual classifier, and finally make action recognition, which classified each 10s video into categories, such as wake, sleep NREM, and sleep REM. The results show that their classifier can reach the overall accuracy of 0.92 ± 0.05, which can replace the manual classification.

Tong et al. [54] apply both segmentation and key point detection in their study. They aim to analyze optomotor response to evaluate animals' visual function and nervous system. They use binarization to make the semantic segmentation of mouse contour and propose a powerful CNN network to detect the position of the mouse's nose and track the orientation of the mouse's head. The results show that their CNN network can achieve a recognition rate of 94.89%.

Jiang et al. [53] propose a hybrid deep learning architecture with a novel hidden Markov model algorithm to describe the temporal characteristics of mice behaviors by action prediction. The architecture contains an unsupervised layer and a supervised layer. The unsupervised layer relies on an advanced spatial-temporal segment Fisher vector encoding both visual and contextual features, and the supervised layer is trained to estimate the state-dependent observation probabilities of the hidden Markov model. The results show that the accuracy of their architecture can get 96.5% on average.

3.5. AI Tasks Taxonomy

After summarizing the behavior analysis applications in mice, we also summarized the AI tasks during behavior analysis in mice, as shown in Table 1. The table also summarizes the data types and characteristics of the study.

To summarize the behavior analysis applications in mice, we first read the collected literature and classified them according to their research purposes and applications. We found that most of the studies could be grouped into 4 categories, namely disease detection, external stimuli effective assessment, social behavior analysis, and neurobehavioral assessment, in which we introduce 4, 8, 6, and 7 literature, respectively. In addition, in the mice behavior analysis studies, the AI-empowered mice behavior applications can divide into multiple AI tasks because of different applications and research methods. We summarized nine tasks in total. We also summarize the characteristics of the data analyzed in the studies. Almost all of them analyze video data, which are broadly classified as having single-view and multiple-view, i.e., whether the data were collected from a single camera or multiple cameras.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

Table 1. AI tasks taxonomy: MV=Multi-view, SV=Single-view; -T=Top-bottom, -B=Bottom-top, -F=Front-Back, -S=Side-Side.

Application	Literature	AI Task	Data Attribute
Neurobehavioral Assessment	[54]	Semantic Segmentation, Key Point Detection	SV-T
	[52]	Image Classification	SV-T
	[55]	Image Classification	SV-T
	[10]	Semantic Segmentation, Action Recognition	SV-T
	[56]	Semantic Segmentation, Image Classification	SV-F
	[53]	Action Prediction	SV-F
	[57]	Semantic Segmentation	SV-F
Social Behavior Analysis	[39]	Object Detection, Action Recognition	SV-S
	[47]	Pose Estimation, Action Recognition	MV-TFS
	[46]	Pose Estimation, Action Recognition Object Tracing	SV-T
	[40]	Action Recognition, Key Point Detection	MV-TS
	[48]	Action Recognition	MV-TS
	[38]	Object Detection, Pose Estimation, Action recognition	MV-TF
External Stimuli Effective Assessment	[25]	Key Point Detection, Action Recognition	MV-B
	[26]	Pose Estimation	SV-B
	[27]	Object Detection, Semantic Segmentation, Image Classification	SV-F
	[28]	Action Recognition	SV-T
	[29]	Instance Segmentation, Key Point Detection, Object Tracing, Action Recognition	SV-T
	[30]	Action Recognition	SV-T
	[31]	Object Detection, Action Recognition	SV-F
	[32]	Object Tracing, Action Recognition	SV-T
Disease Detection	[18]	Semantic Segmentation, Action Recognition	SV-B
	[16]	Semantic Segmentation, Action Recognition	SV-T
	[19]	Key Point Detection, Pose Estimation	MV-BS
	[20]	Key Point Detection, Pose Estimation	SV-S

4. AI-empowered Approaches

In this section, we focus on the techniques behind mice behavior analysis in biology fields. We first build up an AI pyramid according to the AI task's dependency relationship. Then, we introduce several general backbones, namely the fundamental architectures of AI models. In the last, we introduce the AI models in each AI task. Noted that, except some models used to couple with mice video data are introduced, we also introduce some state-of-art approaches used for human-related recognition.

4.1. AI Pyramid

The architecture of AI tasks is organized as Figure 5. It is a "pyramid" structure including four layers: top layer, middle layer, fundamental layer, and backbone layer. The topper layers may take advantage of the techniques of the lower layers.

The backbone layer contains the backbone models and networks. The backbone is the major network of a model. It helps abstract the features of images or videos and generate the feature map for the following network structure. Researchers mainly use the pre-trained backbone and fine-tune it for their study. The common backbones include CNNs, such as ResNet, ResNeXt, DarkNet, MobileNet, Yolo, HourGlass, and Transformers.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

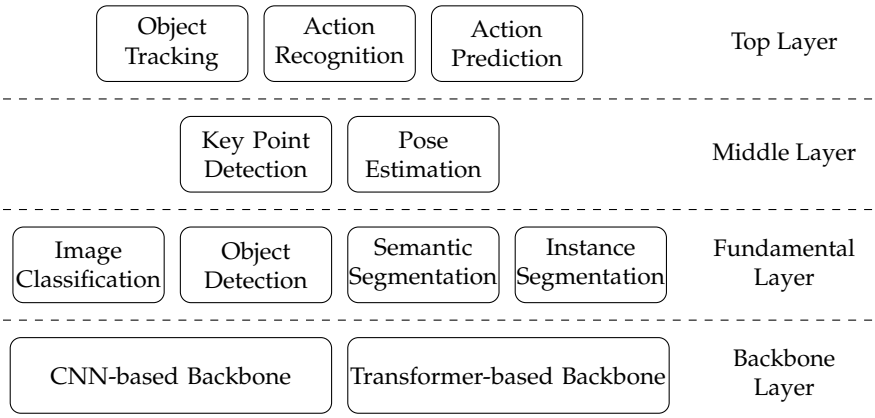

Figure 5. Task Pyramid.

The fundamental layer contains the basic AI tasks, including image classification, object detection, semantic segmentation, and instance segmentation. These tasks are atomic and can not be further divided into other AI tasks and take advantage of backbone networks from the backbone layer. For example, object detection can select YoloV5 as the backbone network.

The middle layer contains key point detection and poses estimation. Both tasks may need support from the fundamental layer. For example, the key point detection model may combine the semantic and instance segmentation as the first step, and apply the object detection as the final step. Also, the tasks of the middle layer may apply to the backbone networks from the backbone layer, such as DarkNet and MobileNet.

The top layer tasks may integrate both the middle and fundamental layers' tasks in the model. For example, the action prediction can combine the key point detection(Middle layer) and the semantic segmentation(Fundamental layer) tasks. The task of the top layer can also integrate the backbone network into its model.

4.2. Backbone

The backbone is the major architecture of the AI models. It helps to extract the modular structure of image features and transform the images into high-dimensional feature representations. Existing known backbones for mice behavior analysis can be categorized into two categories: CNN-based and Transformer-based.

In CNN-based backbones, the backbones contain multiple convolutional layers and pooling layers. The convolutional layers help extract the features of images. The pooling layers reduce the number of parameters and improve the robustness of features. Common CNN-based backbones include DarkNet, ResNet, MobileNet, HourGlass, and YoloV5. The dependency and relationship of these CNN-based backbones are shown in Figure 6. The CNN-based backbones are all based on convolutions and poolings. In detail, the MobileNet requires depthwise convolution to achieve lightweight, and others require residual techniques to improve performance. In the residual techniques, HourGlass, ResNet family, and DarkNet family can be categorized by the iterative module, skip connection, and the darknet module and the darknet module can be furtherly dividied into DarkNet and Yolo families.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

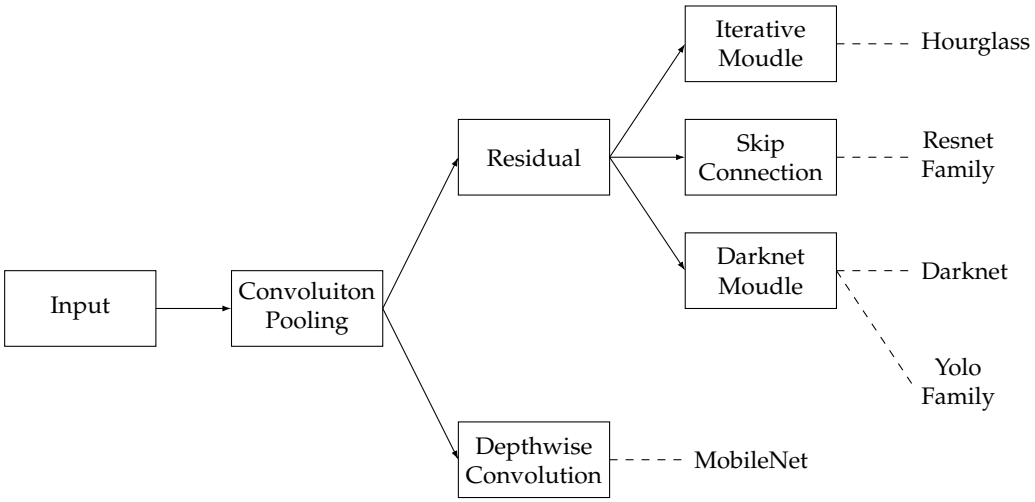

Figure 6. CNN-based Backbone Overview.

DarkNet [58] is a lightweight CNN network. The structure of the backbone adopts multiple convolution layers and downsampling layers. A Batch Normalization layer and a Leaky ReLU layer follow each convolution layer. In detail, it contains an input layer, 19 convolutional layers, 2 upsampling layers, a fully connected layer, 26 batch normalization layers, 19 leaky ReLU layers, and 5 max pooling layers.

ResNet50 [59] is a typical backbone in the Resnet family, which is a deep residual network. It has 50 layers in total and avoids the problem of disappearing gradients. In detail, it contains an input layer, a 7*7 convolutional layer, a pooling layer, 16 residual blocks (3 convolutional layers in each block), a global average pooling layer, a fully connected layer, and a softmax layer.

MobileNet [60] is a lightweight convolutional neural network proposed by Google. It can make rapid image classification and object detection on mobile devices. It has an input layer, 13 convolutional layers, 13 depthwise separable convolutional layers, a global average layer, a fully connected layer, and a softmax layer.

HourGlass [61] is a CNN-based backbone for human pose estimation. It consists of 4 HourGlass modules. Each module contains an input layer, a convolutional layer (64 filters, 7*7 kernel size, stride 2), some residual blocks (64 filters, 3*3 kernel size), a max pooling layer (2*2 kernel size, stride 2), an Hourglass (recursive), some residual blocks (128 filters, 3*3 kernel size), an upsampling layer (2*2 kernel size, nearest-neighbor interpolation), some residual blocks (64 filters, 3*3 kernel size), a convolutional layer (specific filters, 1*1 kernel size), and an output layer.

YoloV5 [62] is the scaled-YoloV4 in fact. contains a convolutional layer, a feature pyramid layer, and a detection head. The convolutional layer takes the CSPNet as the backbone, including 9 convolutional layers. The feature pyramid layer applies a spatial pyramid pooling module and fuses multi-scale feature maps to improve the detection ability of micro targets. The detection head has three branches for detecting targets of any size. Each branch has a convolutional layer and an output layer.

Recently, the transformer-based backbone has become a popular backbone architecture in computer vision tasks. However, it hasn't been used in the mice behavior analysis. Considering its state-of-art performance and accuracy. It is essential to utilize the transformer-based in the biology field. Transformer is mainly applied in Sequence-to-Sequence tasks, such as translation and speech recognition. It contains an input embedding layer, some encoder layers, and some decoder layers (each includes three sub-layers, Multi-head Self-attention, multi-head attention, and feedforward neural network). However, in the transformer-based, the encoders are mostly used as a backbone, such as ViT [63] and Swin [64].

 doi:10.20944/preprints202307.0271.v1

ViT is proposed by Google Brain in 2020. It aims to apply a transformer to the computer vision field. ViT contains four layers: patch embedding layer, transformer encoder layer, global average pooling layer, and the fully connection layer. The patch embedding layer divides the image into pieces of fixed size and maps them into a vector. The transformer encoder layers help to extract the features of the vector. The global average pooling layer and the fully connection layer are used for the feature presentation and the output presentation.

Swin is proposed by Microsoft Research Asia in 2021. It has three parts: Swin transformer block for extracting the local feature, stage segmentation for dividing the image into multiple sub-figures, and the cross-stage connection for transmitting the features among different parts.

4.3. Fundamental Layer Tasks

The fundamental layer tasks mainly make basic image analysis. The goal of these tasks is to extract information about objects or features from images or videos, such as their location, size, shape, and category.

4.3.1. Image Classification

Image Classification is a fundamental task in the field of computer vision. Its goal is to assign a label to an input image from a predefined set of categories. The training methods of image classification can be divided into supervised learning, unsupervised learning, semi-supervised learning, self-supervised learning, and weakly supervised learning. Supervised learning is the model learning labeled data, learning a mapping relationship between data and labels. Unsupervised learning is learning completely unlabeled data from which models learn patterns. Semi-supervised learning is data that includes both labeled and unlabeled parts. Self-supervised model learning is also learning unlabeled data. The difference is that these unlabeled data can be labeled by learning.

Ren et al. [52] used the supervised learning training model. They take a pre-trained CNN trained on ImageNet and fine-tune it for their rodent behavior classification task. They use C_k, F_k, P, D, C to represent a convolutional layer with k filters (C_k), a fully-connected layer with k neurons (F_k), a down-sampling max-pooling layer (P) with kernel size 3 and stride 2, a dropout layer (D), and a soft-max classifier (C). They transfer AlexNet into use by replacing its last 1000-dimensional classification layer with a 5-dimensional classification layer. The AlexNet network architecture is: $C96(11)$-P-$C256(5)$-P-$C384(3)$-$C384(3)$-$C256(3)$-$PF4096$-D-$F4096$-D-C. They also transferred C3D, which simultaneously learns spatial and temporal features by performing 3D convolutions, and has been shown to outperform alternate 2D CNNs for video classification tasks. The C3D network architecture is $C64$-P-$C128$-P-$C256$-$C256$-P-$C512$-$C512$-P-$C512C512$-P-$F4096$-D-$F4096$-D-C.

Cai et al. [55] also use the supervised learning. They develop an analysis pipeline based on a CNN model to identify freezing behavior in mice. The CNN is initialized on the pre-trained ResNet18 architecture and further trained on 'difference images,' the pixel-by-pixel intensity difference between consecutive pairs of frames. The rationale for inputting different images to the CNN was to capture frame-by-frame motion. Each difference image is human-labeled as 1 or 0 to signify 'freeze' or 'no freeze,' and the network learned to predict labels for new difference images. The CNN allowes accurate and automated classification of freezing behavior throughout the duration of their experiments with minimal labor and enables them to determine that the precise temporal relationship between dopamine neuron activity and freezing behavior depends on the VTA subregion.

 doi:10.20944/preprints202307.0271.v1

At present, the image classification of mice is basically supervised learning. It is worth noting that labeling data usually takes a lot of manpower and material resources, and there are a lot of unlabeled data in real life. Although supervised learning is the most commonly used method in image classification, other training methods have their applications, particularly when large amounts of labeled data are unavailable or when labeling is costly. Most of the current popular image classification methods combine supervised and unsupervised learning. The following introduces the current advanced image classification algorithms. The summary of image classification is shown in Table 2.

Du et al. [65] propose a novel semi-supervised efficient contrastive learning classification method for esophageal disease. They use pre-trained ResNet50 as the CNN backbone. First, they propose an efficient contrastive pair generation module to generate efficient contrastive pairs. Then, an unsupervised visual feature representation containing the general feature of esophageal gastroscopic images is learned by unsupervised efficient contrastive learning. Finally, they transfer the feature representation to the downstream esophageal disease classification task. The experimental results have demonstrated that the classification accuracy is 92.57%. The proposed method can reduce the reliance on large labeled datasets and the burden of data annotation.

Xue et al. [66] propose a generative self-supervised pretraining and few-shot land cover classification method for multimodal remote sensing data. The approach contains two stages: generative self-supervised pretraining and few-shot land cover classification. In the pretraining procedure, local multiview observed images are divided into image patches, which are masked randomly, and unmasked patches are embedded for the encoder to learn high-level feature representations. After the self-supervised pretraining process, the learned spatial features are normalized and combined with corresponding spectral information. These are employed as an input of the lightweight SVM for classification. The transformer structure is employed as the backbone.

Li et al. [67] present a self-supervised learning framework for retinal disease diagnosis that reduces the annotation efforts by learning the visual features from the unlabeled images. The framework is based on ResNet18. The workflow of the overall architecture of the self-supervised method involves randomly sampling images from the training dataset, applying random data augmentation twice to generate rotated images, assigning rotation labels to each image, and utilizing a feature embedding network to map the input to a high-level feature vector that is decoupled into two parts: rotation-related and rotation-invariant features. The experimental results demonstrate that with a large amount of unlabeled data available, the proposed method could surpass the supervised baseline for pathologic myopia and is very close to the supervised baseline for age-related macular degeneration, showing the potential benefit of the method in clinical practice.

Taleb et al. [68] propose using self-supervised learning methods to learn from unlabeled data for dental caries classification. The backbone of the methods is CNNs. They train with three self-supervised algorithms on a large corpus of unlabeled dental images, which contain 38K bitewing radiographs. They then apply the learned neural network representations on tooth-level dental caries classification, using labels extracted from electronic health records. The experimental results demonstrate improved caries classification performance and label efficiency.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

Table 2. Summary of Studies on Image Classification.

	Architecture	Type	Category	Dataset	Performance
[52]	AlexNet,C3D	Mice	Supervised learning	Private	The model not only provides more accurate annotations than alternate automatic methods, but also provides reliable annotations that can replace human annotations for neuroscience experiments.
[55]	ResNet18	Mice	Supervised learning	Private	The CNN allows accurate and automated classification of freezing behavior throughout the duration of our experiments with minimal labor
[65]	ResNet50	Stomach	Semi-supervised learning	Private,Kvasir [69]	The classification accuracy is 92.57%, which is better than that of the other state-of-the-art semi-supervised methods and is also higher than the classification method based on transfer learning by 2.28%.
[66]	Transformer	Remote sensing	Self-supervised learning	Private	The generative self-supervised model achieves superior performance in terms of feature learning and land cover classification, especially in the small sample classification case.
[67]	ResNet18	Retina	Self-supervised learning	Ichallenge-AMD dataset [70], Ichallenge-PM dataset [71]	The method outperforms other self-supervised feature learning methods (around 4.2% area under the curve and can surpass the supervised baseline for pathologic myopia
[68]	ResNet18	Dental caries	Self-supervised learning	Private	Using as few as 18 annotations can produce 45% sensitivity, which is comparable to human-level diagnostic performance

4.3.2. Object Detection

Object detection aims to solve the problem of identifying and positioning the set goal. Its solutions can be classified into two categories: one-stage and two-stage. The two-stage method splits the object detection task into a location task and a classification task. A series of candidate boxes as samples are generated through the region propose networks (RPN) first, and then classification regression is carried out through the network. The one-stage method directly regresses the distribution probability and position coordinates of the target instead of the RPN. It obtains the location information and target categories over the backbone network. The major processes of the two methods are shown in Figure 7.

Existing mice behavior analysis studies apply one-stage and two-stage methods. For the one-stage method, Vidal et al. [27] applied YoloV3 trained on the Open Images datasets to detect the mouse faces. Their modified YOLO model is trained for 100 epochs on the corpus. For the first 50 epochs, the entire model is frozen except for the output layer. Then, they unfreeze all the parameters in the model, training the model for another 50 epochs.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

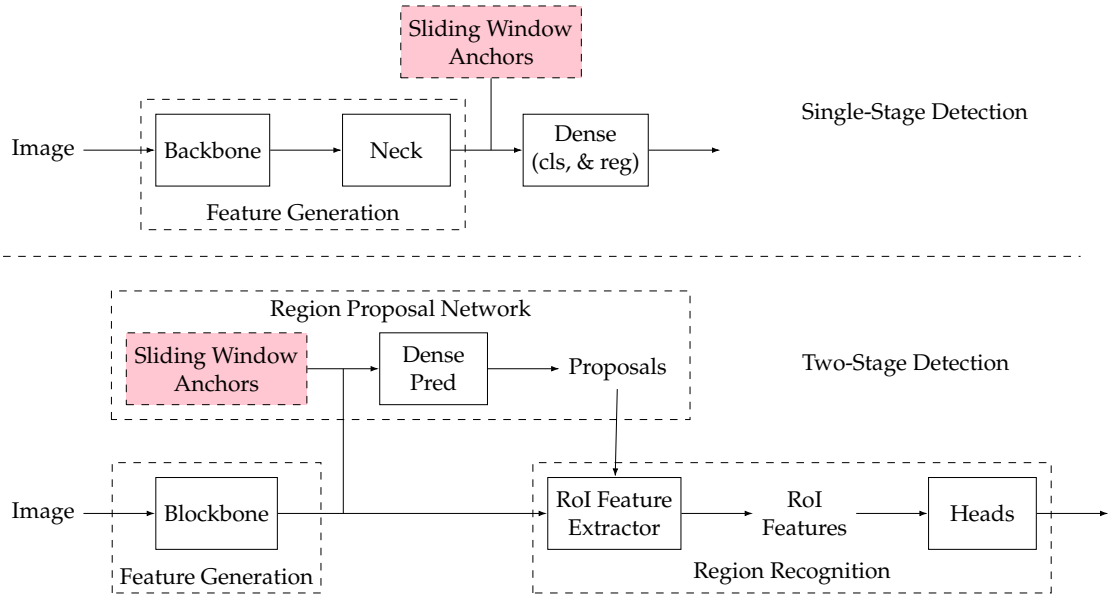

Figure 7. Object detection: One-stage and Two-stage Methods.

For the two-stage method, Martins et al. [31] apply Inspection ResNetV2 with Faster R-CNN to detect the rear paws of mice. They apply Faster R-CNN to locate the rear paws by RPN networks and obtain the region of interest (ROIs). Then, the extracted ROIs are integrated with the feature map, and classification and box regression are carried out by the Inspection ResNetV2. Besides, Segalin et al. [38] also applied Inspection ResNetV2 with ImageNet pre-trained weights to detect the mice location. In their study, the network model computes a short list of up to K possible object detectors proposal (bounding boxes) and associate confidence scores denoting the likelihood of that box containing a target object, in this case, the black or white mice. During training, their network model seeks to optimize the location and maximize confidence scores of predicted bounding boxes that best match the ground truth, while minimizing confidence scores of those that do not match the ground truth. The bounding box location was encoded as the coordinates of the box's upper-left and lower-right corners, normalized with respect to the image dimensions. Finally, the network output is the confidence score scaled between 0 (lowest) and 1 (highest).

With the development of deep learning, state-of-the-art object detection techniques can be further divided into anchor-based and anchor-free methods. The anchor is used for label allocation. In the anchor-based method, boxes of different sizes and aspect ratios are preset either manually or by clustering methods, which can cover the whole image. It can be applied in both one-stage and two-stage methods. The anchor-free method can be divided into two sub-methods. The first one determines the object's center and the predictions for the four borders (called center-based). The second one locates to multiple predefined or self-learning key points and then constrains the spatial range of the object (called key point-based). The state-of-the-art studies on object detection apply the anchor-based and anchor-free mode, which are summarized in Table 3.

Hu et al. [72] propose a one-stage anchor-free network for improving the detection accuracy of the one-stage method. The whole network takes a point cloud input and voxelized it. They apply AFDet as the backbone, which has two stages, and each stage has a convolutional layer and three blocks. To fully explore the potential of the single-stage framework, they apply the self-calibrated convolutions for each block. Besides, they devise an intersection over union (IoU) aware confidence score prediction as the anchor-free head of the network. The head belongs to the key point-based anchor-free method. The authors devise a keypoint prediction sub-head as auxiliary supervision in the detection head. They add another heatmap that predicts 4 corners and the center of every object in bird's eye view during training.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

Table 3. Summary of Studies on Object Detection.

	Architecture	Type	Category	Dataset	Performance
[27]	YoloV3	Mice	One-stage	Open Images dataset	A mean intersection over union (IoU) score of 0.87
[31]	Inspection ResNetV2 with Faster R-CNN	Mice	Two-stage	Private	Approximately 95% accuracy
[38]	Inspection ResNetV2 with ImageNet pretrained weights	Mice	Two-stage	Behavior Ensemble and Neural Trajectory Observatory (BENTO)	Good efficiency on Precision-Recall (PR) curves
[72]	Point Cloud Voxelization, 3D Feature Extractor, backbone(AFDet) and the Anchor-Free Detector	Object detection from point clouds	One-stage, anchor-free	Waymo Open Dataset, nuScenes Dataset	Accuracy:73.12, latency:60.06ms
[73]	YOLOv5, the feature fusion layer, and the multiscale detection layer	Industrial defect detection	Two-stage, anchor-based	VOC2007, NEU-DET, Enriched-NEU-DET	83.3% mean average precision (mAP)
[74]	The location prior network (LPN) and the size prior network (SPN)	Video object detection	One-stage	ImageNet VID	54.1 AP and 60.1 APl
[75]	ResNet backbone, a FPN, an ARM cascade network with rotated IoU prediction branch, and the two-stage sample selective strategy	Rotating object detection	Two-stage	UAV-ROD	96.65 mAP and 98.84 accurancy under the plane category

Li et al. [73] propose a two-stage anchor-based network to make the first-stage recognition more effective at locating insignificant small defects with high similarity on the steel surface. The network structure contains input, backbone, neck, and output parts. The input terminal mainly contains the preprocessing of the data, including mosaic data augmentation and adaptive image filling. In the neck network, the feature pyramid structures of feature pyramid network (FPN) and pixel aggregation network (PAN) were used. The FPN structure conveys strong semantic features from the top feature maps into the lower feature maps. At the same time, the PAN structure conveys strong localization features from lower feature maps into higher feature maps. The head output is mainly used to predict targets of different sizes on feature maps. The backbone is YoloV5 with improved feature extraction capability of the backbone network for steel defects. They remove the Conv and C3 layer that obtained 1/32 scale feature information in the original YOLOv5, and replace it with a Conv and C3 layer that extracted feature information at a 1/24 scale. Besides, they embed an efficient channel attention network mechanism into the backbone network and connect it in parallel to the C3 module.

Sun et al. [74] present a simple yet efficient framework to address the computational bottlenecks and achieve efficient one-stage VOD. They proposed two modules to achieve an efficient one-stage video object detector called the location prior network and the size prior network. The location prior network has two steps. First, the foreground region selection is guided by the detected bounding boxes from the previous frame. Second, the partial feature aggregation enhances the selected foreground pixels using attention modules. Besides, the authors apply an attention mechanism in the one-stage method and solve the bottlenecks, including efficiency and detection heads on low feature levels. The input of the attention is foreground pixels on the current frame and the reference frames.

Zhou et al. [75] propose an anchor-based two-stage model called TS4Net for rotating object detection solely. Benefiting from the ARM and TS4, the TS4Net can achieve superior performance with one preset horizontal anchor. The architecture of TS4Net adopts the vanilla one-stage detector RetinaNet as the baseline model. In the RetinaNet, two parallel fully convolutional networks are connected after FPN to perform the classification and regression tasks, respectively. It can also add an extra IoU prediction head to train jointly with the classification head and regression head, which improves the detection performance during inference. To select the positive samples from the horizontal anchors with large IoU values, authors adopted an ARM cascade network including a two-stage cascade network, which is stacked by four convolutional layers with 3*3 convolution kernels as classification and regression networks in the first stage. Besides, the authors propose the two-stage

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

sample selective strategy. The first stage of ARM refines the horizontal anchors to high-quality rotated anchors, and then the second stage adjusts the rotating anchor to a more accurate prediction box.

Recently, Zhou et al. [76] propose a state-of-the-art two-stage study on video object detection field with transformer technique. They propose an end-to-end model based on spatial-temporal transformer architecture, improving the efficiency of the detection transformer and deformable DETR. The model started from a ResNet backbone extracting features of multiple frames, Then, a series of shared spatial transformer encoders produce the feature memories, which are linked and fed into the temporal deformable transformer encoder, and the spatial transformer decoder decodes the spatial object queries. Next, the model used a temporal query encoder to model the relations of different queries and aggregate these queries supporting the object query of the current frame. Both the temporal object query and the temporal feature memories are fed into the temporal deformable transformer decoder to learn the temporal contexts across different frames. The input is video frames, and the output is the shared weights.

4.3.3. Semantic Segmentation

Semantic segmentation is a computer vision task that assigns each pixel in an image to a specific semantic category. In mice behavior analysis studies, by applying semantic segmentation to mice behavior video data, it can be used for behavior recognition and tracking, spatial localization and trajectory analysis, environmental interaction, behavioral context association, disease model, and drug effect evaluation. The application of semantic segmentation in mice behavior analysis research can achieve fine classification and quantification of behavior, provide more comprehensive and accurate behavioral characterization, and promote a deeper understanding of mouse behavior patterns and biological mechanisms.

Vidal et al. [27] propose a machine-learning approach to automate the prediction of the grimace scale on white-furred mice, which is used to understand the suffering of a mouse in the presence of interventions. The approach involves face detection, landmark region extraction, and expression recognition. For eye region extraction and grimace pain prediction, a novel structure based on a dilated convolutional network is proposed. Dilated convolutional neural networks [77] were proposed as effective tools to perform semantic segmentation.

Wu et al. [78] propose a boosting semantic segmentation framework that performs state-of-the-art segmenting of somata and vessels in the mouse brain. The proposed framework consists of a CNN for multilabel semantic segmentation, a fusion module combining the annotated labels and the corresponding predictions from the CNN, and a boosting algorithm to update the sample weights sequentially. It improves the quality of the annotated labels for deep learning-based segmentation tasks.

Geuther et al. [10] propose a machine learning-based visual classification of sleep in mice, which provides a path to high-throughput studies of sleep. The authors collect synchronized high-resolution video and EEG/EMG data in 16 male C57BL/6J mice, extract features from the video that are time and frequency-based, and use the human expert-scored EEG/EMG data to train a visual classifier. When processing the video data, they apply a segmentation neural network architecture [79] to produce mice masks.

Existing semantic segmentation methods are divided into four categories according to different network architectures: CNN-based architectures, transformer-based architectures, multi-layer perception-based (MLP-based) architectures, and others.

In the CNN-based architecture, the deep network has a strong representation ability of semantic information, and the shallow network contains rich spatial detail information. Zhang et al. [80] proposed an EncNet model, which designed a context encoding module to capture global semantic information and calculated the scaling factor of the feature graph based on the coding information to highlight the information categories that need to be emphasized. Some of the most important works include the DeepLap family proposed by Chen et al. [81] and the densely connected atrous spatial

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

pyramid pooling (DenseASPP) proposed by Yang et al. [82]. They all use dilated convolution to replace the original down-sampling method and expand the receptive field to obtain more context information without increasing the number of parameters and calculations.

Transformer is a deep neural network based on self-attention. In the recent two years, transformer structure and its variants have been successfully applied to segmentation. Zheng et al. [83] first performed semantic segmentation based on the transformer and constructed a segmentation transformer network to extract global semantic information. Inspired by the segmentation transformer network, trudel et al. [84] design a pure transformer model, named Segmenter, to apply to semantic segmentation tasks. The model leverages pre-trained models for image classification and fine-tunes them on moderate-sized datasets available for semantic segmentation. Segmenter outperforms the state-of-the-art on both ADE20K and Pascal Context datasets and is competitive on Cityscapes.

MLP-based architecture is simple in design since it abandons convolution and self-attention. The performance in many visual tasks is comparable to the CNN-based and Transformer-based architectures. Yu et al. [85] propose a novel pure MLP architecture, spatial-shift MLP (S2-MLP), which only contains channel-mixing MLP. The proposed S2-MLP attains higher recognition accuracy than MLP-mixer when training on the ImageNet-1K dataset.

Table 4. Summary of Studies on Semantic Segmentation.

Reference	Architecture	Type	Category	Dataset	Performance
[27]	YoloV3	Mice	CNN-based	Open Images dataset	Achieves a performance of 97.2% in terms of accuracy
[78]	DCNN based on U-Net	Mice	CNN-based	MOST dataset	Improves the network performance by about 3–10%
[10]	-	Mice	-	Private	Achieves an overall accuracy of 0.92 ± 0.05 (mean ± SD)
[80]	Context Encoding Network based on ResNet	Semantic segmentation framework	CNN-based	CIFAR-10 dataset	Achieves an error rate of 3.45%
[81]	DCNN (VGG-16 or ResNet-101)	Semantic image segmentation model	CNN-based	PASCAL VOC 2012, PASCAL-Context, PASCALPerson-Part, and Cityscapes dataset	Reaching 79.7 percent mIOU
[82]	DenseASPP, consists of a base network followed by a cascade of atrous convolution layers	Semantic image segmentation in autonomous driving	CNN-based	Cityscapes dataset	Achieve state-of-the-art performance
[83]	Transformer	Segmentation model	Transformer-based	ADE20K, Pascal Context, and Cityscapes dataset	Achieves new state of the art on ADE20K (50.28% mIoU), Pascal Context (55.83% mIoU) and competitive results on Cityscapes
[84]	Vision Transformer	Segmentation model	Transformer-based	ADE20K, Pascal Context, and Cityscapes dataset	Outperforms the state of the art on both ADE20K and Pascal Context datasets and is competitive on Cityscapes
[85]	Spatial-shift MLP (S2-MLP), containing only channel-mixing MLPs	Segmentation model	MLP-based	ImageNet-1K dataset	Attains considerably higher recognition accuracy than MLP-mixer on ImageNet-1K dataset.

4.3.4. Instance Segmentation

Instance segmentation is a computer vision technique that involves identifying and delineating individual objects within an image. Unlike semantic segmentation, which assigns a single label to each pixel in an image, instance segmentation identifies different objects within an image and assigns each object a unique label. The methods of instance segmentation can be divided into three categories: top-down, bottom-up, and one-stage. In mice behavior studies, instance segmentation can be used to track mouse movement trajectories and postures, allowing for the analysis of activity patterns and behavioral characteristics. Instance segmentation provides researchers with accurate and efficient data analysis tools to promote the development and progress of mouse research, whose studies are summarized in Table 5.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

Marks et al. [29] use top-down methods. They propose SIPEC:SegNet, which is based on the Mask R-CNN architecture, to segment instances of animals. SIPEC:SegNet is optimized for analyzing multiple animals. They further apply transfer learning onto the weights of the Mask R-CNN ResNet-backbone pre-trained on the Microsoft Common Objects in Context (COCO) dataset. Moreover, they apply image augmentation to increase network robustness against invariances (for example, rotational invariance) and therefore increase generalizability. The experimental results demonstrate that SIPEC:SegNet achieved a mean average precision of 1.0 ± 0 (mean ± s.e.m.). For single-mouse videos, the model achieves 95% of its mean peak performance (MAP of 0.95 ± 0.05) using as few as a total of three labeled frames for training. SIPEC:SegNet could robustly segment animals despite occlusions, multiple scales, and rapid movement, and enable tracking of animal identities within a session.

Although instance segmentation can play a significant role in mice behavior recognition, there are not many studies on mice behavior that utilize instance segmentation. The following introduces some popular instance segmentation methods of the above three method categories, which can provide a reference for the subsequent research on mice behavior.

Shen et al. [86] propose a parallel detection and segmentation, a framework to learn instance segmentation with only image-level labels. The framework draws inspiration from both top-down and bottom-up instance segmentation approaches. The detection module is the same as the typical design of any weakly supervised object detection. In contrast, the segmentation module leverages self-supervised learning to model class-agnostic foreground extraction, followed by self-training to refine class-specific segmentation. The paper further proposes an instance-activation correlation module to improve the coherence between detection and segmentation branches. The experimental results demonstrate that the proposed method outperforms baselines and achieves state-of-the-art results on PASCAL VOC and COCO.

Korfhage et al. [87] present a CNN architecture based on Mask R-CNN for cell detection and segmentation (top-down) that incorporates previously learned nucleus features. A novel fusion of feature pyramids for nucleus detection and segmentation with feature pyramids for cell detection and segmentation is used to improve performance on a microscopic image dataset created by the authors and provided for public use, containing both nucleus and cell signals. The proposed feature pyramid fusion architecture clearly outperforms a state-of-the-art Mask R-CNN approach for cell detection and segmentation with relative mean average precision improvements of up to 23.88% and 23.17%, respectively. No post-processing was carried out in the experiments when compared to other methods to ensure a fair comparison.

Zhou et al. [88] propose a bottom-up regime to learn category-level human semantic segmentation and multi-person pose estimation in a joint and end-to-end manner. They adopt ResNet-101 [33] as the backbone. The proposed method exploits structural information over different human granularities and eases the difficulty of person partitioning. A dense-to-sparse projection field is learned and progressively improved over the network feature pyramid for robustness. By formulating joint association as maximum-weight bipartite matching, a differentiable solution is developed to exploit projected gradient descent and Dykstra's cyclic projection algorithm. This makes the method end-to-end trainable and allows back-propagating the grouping error to supervise multi-granularity human representation learning directly. Experiments on three instance-aware human parsing datasets show that the proposed model outperforms other bottom-up alternatives with much more efficient inference.

Wang et al. [89] propose a framework called segmenting objects by locations (SOLO), which is based on ResNet-50. SOLO is a one-stage, end-to-end instance segmentation method that can perform detection and segmentation simultaneously with high efficiency and accuracy. The main idea of the SOLO is to transform the instance segmentation problem into a dense prediction problem. Specifically, SOLO divides the image into a set of position-sensitive small grids and predicts the object category and instance segmentation mask in each grid. In this way, each pixel can be assigned to an instance,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

and the object edge can be accurately segmented. The experimental results demonstrated that the proposed SOLO framework achieves state-of-the-art results for the instance segmentation task in terms of both speed and accuracy while being considerably simpler than the existing methods.

Li et al. [90] propose PaFPN-SOLO, a SOLO-based image instance segmentation algorithm. They enhanced the ResNet backbone by incorporating a Non-local operation, effectively preserving more feature information from the image during the extraction process. In addition, they employ a method known as bottom-up path augmentation. This method was designed to extract more precise positional information from the lower feature layers. This dual improvement not only boosted the network model's ability to localize the feature structure but also reduced the distance over which information needed to propagate between feature layers. When the modified algorithm was tested on two datasets, COCO2017 and Cityscapes, it produced significantly improved segmentation results. The average segmentation accuracy on these datasets reached 56% and 47.3% respectively, marking an increase of 4.4% and 7.4% over the performance of the original SOLO network.

Table 5. Summary of Studies on Instance Segmentation.

Reference	Architecture	Type	Category	Dataset	Performance
[29]	Mask R-CNN	Mice	Top-down method	Private	SIPEC successfully recognizes multiple behaviours of freely moving individual mice as well as socially interacting non-human primates in three dimensions
[86]	PDSL framework	-	Top-down method	PASCAL VOC 2012 [91], MS COCO [92]	PDSL framework outperforms baselines and achieves state-of-the-art results on PASCAL VOC and MS COCO.
[87]	Mask R-CNN	Cell	Top-down method	Private	The proposed architecture clearly outperforms a state-of-the-art Mask R-CNN approach for cell detection and segmentation with relative mean average precision improvements ofup to23.88% and 23.17%, respectively.
[88]	ResNet101	Human	Bottom-up method	MHPv2 [93], DensePose-COCO [94], PASCAL-Person-Part [95]	Experiments on three instance-aware human parsing datasets show that the proposed model outperforms other bottom-up alternatives with much more efficient inference.
[89]	ResNet50	-	Top-down method	LVIS [96]	The proposed framework achieves state-of-the-art results for instance segmentation in terms of both speed and accuracy, while being considerably simpler than the existing methods.
[90]	ResNet	-	Bottom-up method	COCO2017, Cityscapes [97]	The average segmentation accuracy on COCO2017 and Cityscapes reached 56% and 47.3% respectively, marking an increase of 4.4% and 7.4% over the performance of the original SOLO network.

4.4. Middle Layer Tasks

The middle-layer tasks mainly focus on pose estimation in both humans and other animals. They can be used in motion recognition, human-computer interaction, and motion capture applications. To make the estimation more accurate, they need higher accuracy and real-time than other tasks, so they also cost more computing resources than other tasks.

4.4.1. Key Point Detection

Key point detection is a major technology of deep learning. It is a basic task in computer vision. It is the pre-task of human action recognition and action prediction. In the mice behavior analysis studies, the key point detection also contains fundamental techniques, such as object detection and semantic segmentation. The input is an image, and the output is the expected key points. Normally, key point detection can be categorized into 2D and 3D detection, and all the studies of mice key point detection apply 2D detection methods.

Tong et al. [54] make key point detection based on the semantic segmentation of the mice contour. They proposed a CNN architecture to detect the snout point of the mice. The CNN contains four convolutional layers, an average pooling layer after the convolutional layers, a flattened layer, and

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

three fully connected layers. The input of CNN is an area near snouts, and the output is the snout point position.

Wotton et al. [25], Weber et al. [19], Winters et al. [46], and Aljovic et al. [20] all make key point detection for body-part detection. Wotton et al. [25] propose a ResNet50-based CNN to learn specific features and the skipping function to minimize information loss. Weber et al. [19], Aljovic et al. [20], and Winters et al. [46] make the key point for detecting distinct body parts of mice. They proposed a ResNet-50 from the DeepLabCut by manually labeling 120 frames selected using k-means clustering from multiple videos of different mice. The former one detects the body parts, including the head, right front toe, left front toe, center front, right back toe, left back toe, center back, and tail base. The middle one detected 14 body parts configuration for the mother and pup together. The latter labels six body parts (toe, MTP joint, ankle, knee, hip, and iliac crest) in 450 image frames, and trained for 400,000 iterations.

Besides mice, key point detection is mostly applied in humans. Human key point detection can be categorized into single-person and multi-person detection. The multi-person detection algorithms can be further divided into Top-Down and Bottom-Up two parts. All the studies are summarized in Table 6.

Wen et al. [98] make multi-person key point detection based on the pre-trained network and SHNet. The pre-trained network was used for object detection. SHNet is used for keypoint detection. It consisted of four stages and the attention mechanism. The first stage consists of four remaining units, which are the same as ResNet50 and are composed of a bottleneck with a width of 64, followed by a 3*3 convolution feature graph whose width is reduced to 4. The second, third, and fourth stages contained 1, 4, and 3 communicative blocks. Besides, the model required paying more attention to the channel features with the largest amount of information and suppressing unimportant channel features. The attention mechanism contains information input, calculation of attention distribution, and calculation of weight average of input information. The input is the vector of each image, and the output is the weights of each feature.

Gong et al. [99] propose a retrained AlphaPose model to make multi-person key point detection in the upper human body. The AlphaPose method detects human key points based on the regional multiplayer pose estimation (RMPE) framework proposed by the AlphaPose method, containing three components: symmetric spatial transformation network (SSTN), parametric pose non-maximum suppression (NMS) and pose guided proposals generator (PGPG). The SSTN network consists of a spatial transformation network (STN), single-person pose estimation (SPPE), and spatial de-transformer network (SDTN). STN is used to acquire high-quality human proposals and exclude inaccurate input frames. SPPE is used to estimate the pose of the input human candidates. SDTN maps the pose estimated by SPPE back to the original image coordinates and adjusts the input frames to make the detected frames more accurate. The AlphaPose model can detect 17 human upper body key points.

Zang et al. [100] propose a lightweight multi-stage attention network (LMANet) to detect the key points of a single person at night. LMANet contains a backbone network and some subnets for identifying key points that are not obvious or hidden through the characteristics of different receptive fields and the association between key points. The backbone network is pruned MobileNet. The input of the backbone is 334*384. The first layer is a 3*3 convolution, and layer 2 to layer 6 are the classic bottleneck structure. The expected output is 12*12. For the subnets, there are 2 subnets, each of which contains only 2 bottlenecks. The input is 48*48, and the output is 12*12, which is the spatial attention module in the revised feature representation. Besides, the second bottleneck and the fourth bottleneck of the LMANet backbone network have added the channel attention mechanism, which is used to enhance the local features of each feature map at the spatial level. The attention module can get a refined output after the two-stage networks, and finally obtain a heatmap of 14 key points of the human body.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

Hong et al. [101] proposed a PGNet for single-person key-point detection. PGNet consists of three main components: Pipeline Guidance Strategy (PGS), Cross-Distance-IoU Loss (CIoU), and Cascaded Fusion Feature Model (CFFM). The backbone network in PGNet is ResNet-50, which is divided into 5 stages using CFFM. The feature-guided network after the image is convolved is used to extract key-point features, while CFFM is utilized to extract high-level and low-level features from the conv1-5 layers of ResNet-50. The middle three layers of CFFM are specifically used to avoid consuming a large amount of spatial information during convolution calculations. The feature-guided network combines traditional data parallelism with model parallelism enhanced with pipelining, which partitions the layers of the object being trained into multiple stages. After feature extraction, a convolution operation is used to fuse the features of the two branches, which completes the key points.

Table 6. Summary of Studies on Key Point Detection.

Reference	Architecture	Type	Category	Dataset	Performance
[54]	CNN	Mice	2D	Private	Achieve the recognition rate of 94.89%
[25]	ResNet-50	Mice	2D	Private	Reveal strain differences in both response timing and amplitude
[19]	ResNet-50	Mice	2D	Private	A 98% accuracy when compared baseline to animals at 3 dpi
[46]	ResNet-50	Mice	2D	Private	An accuracy of 86.7%
[20]	ResNet-50	Mice	2D	Private	Predict the acute injury status with 90% accuracy and long-term defcits with 85% accuracy.
[98]	SHNet, MaskedNet	Human	multi-person	MPII, COCO2017	Achieve high accuracy on all 16 joint points
[99]	AlphaPose	Human	multi-person	Private, Halpe-FullBody136	Detection precision is improved by 5.6%, and the false detection rate is reduced by 13%
[100]	LMANet	Human	single-person	Private, MPII, AI Challenger	PCKh value is 83.0935
[101]	PGNet	Human	single-person	COCO	Improve the accuracy of the COCO dataset by 0.2%

4.4.2. Pose Estimation

Quantifying mice behaviors from videos or images remains a challenging problem, where pose estimation plays an important role in describing mice behaviors. Although deep learning-based methods have made promising advances in human pose estimation, they cannot be directly applied to pose estimation of mice due to different physiological natures. Particularly, since the mouse body is highly deformable, it is a challenge to accurately locate different keypoints on the mouse body. The mice pose estimation can be divided into 2D and 3D.

Zhou et al. [102] propose a novel Hourglass network-based model, defined as graphical model based structured context enhancement network (GM-SCENet) where two effective modules, structured context mixer (SCM) and cascaded multi-level supervision (CMLS) are subsequently implemented. SCM can adaptively learn and enhance the proposed structured context information of each mouse part by a novel graphical model that takes into account the motion difference between body parts. Then, the CMLS module is designed to jointly train the proposed SCM and the Hourglass network by generating multi-level information, increasing the robustness of the whole network. Using the multi-level prediction information from SCM and CMLS, they develop an inference method to ensure the accuracy of the localization results.

Xu et al. [103] propose a symmetry approach and design a CNN for mice pose estimation under scale variation. The network architecture consists of a UNet structure with residual structure to extract features, Atrous Spatial Pyramid Pooling (ASPP) module to expand the perceptual field, and deep and shallow feature fusion to capture the various spatial relationships related to body parts. The model generates a set of prediction results based on heat map and coordinate offset. The paper also discusses the use of dilation convolution and loss function design. The authors use their own built mice dataset and obtained state-of-the-art results.

 doi:10.20944/preprints202307.0271.v1

Salem et al. [43] propose a systematic approach to accurately estimate the 3D pose of the mice from single-monocular fisheye-distorted images. The approach employs a novel adaptation of a structured forest algorithm. The authors benchmark their algorithm against existing methods and demonstrate the utility of the pose estimates in predicting mice behavior in a continuous video. The full text information provides a review of literature works with respect to pose representation and pose estimation/detection method.

In addition to the above mice pose estimation studies, we also present some state-of-the-art human pose estimation studies, which are expected to be applied to mice pose estimation. The human pose estimation techniques can be categorized into 2D and 3D pose estimation. In 2D human pose estimation, joints and body parts are tracked across the surface of an image, whereas 3D human pose estimation also estimates the depth of the joints and body parts in the image [104].

2D human pose estimation has been a fundamental yet challenging problem in computer vision. The goal is to localize human anatomical keypoints (e.g., elbow, wrist, etc.) or parts. Sun et al. [105] propose a High-resolution net (HRNet) for human pose estimation that maintains high-resolution representations throughout the process. Cheng et al. [106] and Yu et al. [107] both propose novel methods based on HRNet. The former presents HigherHRNet, which uses high-resolution feature pyramids to learn scale-aware representations and solve the scale variation challenge in bottom-up multi-person pose estimation. The feature pyramid in HigherHRNet consists of feature map outputs from HRNet and upsampled higher-resolution outputs through a transposed convolution. The latter presents an efficient high-resolution network, Lite-HRNet. The authors start by applying the efficient shuffle block in ShuffleNet to HRNet, which yields stronger performance over popular lightweight networks such as MobileNet, ShuffleNet, and Small HRNet. They introduce a lightweight unit, conditional channel weighting, to replace costly pointwise (1×1) convolutions in shuffle blocks.

To date, most of the efforts for 3D pose estimation are focused on monoculars. Iskakov et al. [108] present two novel solutions for multi-view 3D human pose estimation based on new learnable triangulation methods that combine 3D information from multiple 2D views. The first solution is a basic differentiable algebraic triangulation with an addition of confidence weights estimated from the input images. The second solution is based on a novel method of volumetric aggregation from intermediate 2D backbone (ResNet-152) feature maps. Both approaches are end-to-end differentiable, which allows direct optimization of the target metric. He et al. [109] proposes a method called 'epipolar transformer' which enables a 2D detector to leverage 3D-aware features to improve 2D pose estimation. The method leverages epipolar constraints and feature matching to approximate the features at a corresponding point in a neighboring view. This helps to resolve depth ambiguity and accurately estimate the 3D position of joints. They adopt ResNet-50 with image resolution 256×256 proposed in simple baselines for human pose estimation as our backbone. network. They use the ImageNet pre-trained model for initialization. Weinzaepfel et al. [110] propose a method called DOPE that detects and estimates whole-body 3D human poses, including bodies, hands, and faces, in the wild. The method takes advantage of previously annotated or generated datasets to train independent experts for each part and distills their knowledge into a single deep network designed for whole-body 2D-3D pose detection. They follow the Faster RCNN implementation and adopt ResNet50 as the backbone. The resulting estimations are combined to obtain whole-body pseudo-ground-truth poses. A distillation loss encourages whole-body predictions to mimic the experts' outputs. DOPE outperforms the same whole-body model trained without distillation while staying close to the performance of the experts.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

Table 7. Summary of Studies on Pose Estimation.

Reference	Architecture	Type	Category	Dataset	Performance
[102]	Hourglass network	Mice	2D	Parkinson's Disease Mouse Behaviour	The superior performance over the other state-of-the-art methods in terms of PCK@0.2 score.
[103]	ResNet, ASPP	Mice	2D	Private	Overall performance has achieved superior performance at various thresholds
[43]	Structured forests	Mice	3D	Private	Precision 86%
[105]	HRNet	Human	2D	COCO, MPII human pose estimation, and PoseTrack dataset	Achieves a 92.3 PCKh@0.5 score
[106]	HigherHRNet	Human	2D	COCO dataset	Achieves new state-of-the-art result on COCO test-dev (70.5% AP), surpasses all top-down methods on CrowdPose test (67.6% AP)
[107]	Lite-HRNet	Human	2D	COCO and MPII human pose estimation datasets	Achieves 87.0 PCKh @0.5
[108]	ResNet-152	Human	3D	Human3.6M and CMU Panoptic datasets	Achieve state-of-the-art performance on the Human3.6M dataset
[109]	ResNet-50	Human	3D	InterHand and Human3.6M datasets	Outperforms state-of-the-art by 4.23mm and achieves MPJPE 26.9 mm
[110]	ResNet-50	Human	3D	MPII, MuPoTs-3D, and RenderedH datasets	Outperforms the same whole-body model while staying close to the performance of the experts, less demanding than the ensemble of experts and can achieve real-time performance

4.5. Top Layer Tasks

Top layer tasks mostly take multiple steps including the lower layer's tasks. They are used to analyze and understand the motion in applications such as surveillance, robotics, and sports analysis.

4.5.1. Object Tracking

Object tracking refers to the process of automatically detecting and tracking a specific object in a video or image sequence. The input of object tracking algorithm is usually a video sequence, and the output is the information of the target's position, size, and motion status in different frames of the input video, which is used to achieve continuous tracking of the target. In neuroscience research on mice, object tracking technology can help researchers better understand the mouse's behavioral patterns and neural activity through monitoring and analyzing mouse's behavior. Furthermore, target tracking technology can be used to evaluate mouse behavior performance in drug treatment or nervous system disease models.

Marks et al. [29] introduce SIPEC:SegNet, a Mask R-CNN architecture designed to enable tracking of animal identities within a session. To improve temporal continuity-based tracking, SIPEC:IdNet is developed with a DenseNet backbone that generates visual features, which are integrated over time using a gated-recurrent-unit network to reidentify animals when the temporal-continuity-based tracking fails. This allows SIPEC to identify primates over the course of weeks and outperform both idtracker.ai's identification module within and across sessions, as well as PrimNet.

Wang et al. [32] and Winters et al. [46] both use DeepLabCut to track mice behaviors. DeepLabCut is an open-source software package for markerless pose estimation of animals and humans in video data using deep learning, which can be used for tracking the pose of mice. The DeepLabCut architecture is a deep neural network based on ResNet, referred to as a "multi-residual network". Wang et al. [32] use DeepLabCut to estimate the positions of mouse body parts. Positional features are calculated using DeepLabCut outputs and are used to train random forest and hidden markov models with equal number of states, separately. Winters et al. [46] use DeepLabCut to create a dam-pup tracking algorithm in the pup retrieval protocol and classified variables such as "maternal approach", "carrying" and "digging" using simple behavioral analysis.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

SIPEC: SegNet and DeepLabCut essentially track mice through object detection rather than actual object tracking models. There are many existing object tracking algorithms for tracking humans and vehicles, which can be classified into single-branch and multi-branch models. These model can provide inspiration for object tracking of mice, shown in Table 8.

Single-branch models use a single model or algorithm for object tracking, typically based on a linear or nonlinear model. Wang et al. [111] propose an online multi-object tracking framework based on a hierarchical single-branch network, based on Faster R-CNN [112] with a ResNet-50 [33] backbone. The proposed single-branch network utilizes an improved Hierarchical Online Instance Matching (iHOIM) loss to explicitly model the inter-relationship between object detection and Re-ID. The iHOIM loss function unifies the objectives of the two subtasks and encourages better detection performance and feature learning even in extremely crowded scenes. Moreover, the paper introduce the object positions, predicted by a motion model, as region proposals for subsequent object detection. The object trajectories are obtained using a DeepSort framework. Experimental results show that compared with the two-stage methods on MOT16 and MOT20 datasets, their model achieves a state-of-the-art performance even in crowded tracking scenes.

Multi-branch models use multiple models or algorithms for object tracking, typically by combining multiple linear or nonlinear models to track the object. Vaquero et al. [113] develop a complete detection and tracking system for vehicles in driving scenarios using a dual-branch CNN architecture. The system utilizes LiDAR data and a deconvolutional neural network to segment vehicles from a front projection, and then apply Euclidean clustering to extract bounding boxes for tracking over time. The authors further enhance the system by introducing a dual-view deep-learning pipeline to segment vehicles from LiDAR information, as well as novel techniques such as adaptive threshold recursive clustering and a bounding box growing algorithm guided by contextual information. They evaluate their method extensively on the Kitti benchmark [114] for both detection and tracking tasks, and demonstrate superior performance compared to existing methods through quantitative analysis. Jiang et al. [115] propose a multi-branch and multi-scale perception object tracking framework based on Siamese Convolutional Neural Networks denoted as MultiBSP. To achieve different task goals for each branch, a tower-structured relation network is created to learn the non-linear relation function between a template and search area. By using a multi-branch architecture, the system is able to combine and verify the results from each branch, resulting i n a powerful performance. The experimental results show that the MultiBSP achieved state-of-the-art performance on six benchmarks.

Table 8. Summary of Studies on Object Tracking.

Reference	Architecture	Type	Category	Dataset	Performance
[29]	Mask R-CNN	Mice	-	Private	SIPEC:SegNet robustly segment animals despite occlusions, multiple scales and rapid movement, and enable tracking of animal identities within a session.
[32]	ResNet	Mice	-	Private	DeepLabCut can estimate the positions of mouse body parts.
[46]	ResNet	Mice	-	Private	Automated tracking of a dam and one pup was established in DeepLabCut and was combined with automated behavioral classification of "maternal approach", "carrying" and "digging" in Simple Behavioral Analysis (SimBA).
[111]	Faster R-CNN, ResNet-50	Human	Single-branch	MOT16 [116], MOT20 [117]	Compared with the two-stage methods on MOT16 and MOT20 datasets, the model achieves a new state-of-the-art performance even in crowded tracking scenes.
[113]	DNN	Vehicle	Multi-branch	Kitti [118]	The dualbranch classifier consistently outperforms previous single-branch approaches, improving or directly competing to other state of the art LiDAR-based methods.
[115]	ResNet50	-	Multi-branch	VOT-2018 [119], VOT-2019 [120], OTB-100 [121], UAV123 [122], GOT10k [123], LASOT [124]	MultiBSP can achieve robust tracking and have state-of-the-art performance and the effectiveness of each module and the tracking stability is proved by qualitative and quantitative analyses.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

4.5.2. Action Recognition

Action recognition in mice plays a crucial role in biomedical research, as it can be employed for studying disease models, evaluating drug efficacy, investigating the functioning of the nervous system, exploring behavioral genetics, and assessing environmental toxicity. By observing and analyzing the behavioral patterns of mice, insights into disease mechanisms, drug effects, neural network functionality, genetic foundations, and the impact of the environment on organism behavior can be revealed. We summary the action recognition related research in this seciton, the sumamrizing reuslts can be refered to Table 9.

Segalin et al. [38] present the Mouse Action Recognition System (MARS), a quartet of software tools for automated behavior analysis, training and evaluation of novel pose estimator and behavior classification models, and joint visualization of neural and behavioral data. This software is accompanied by three datasets aimed at characterizing inter-annotator variability for both pose and behavior annotation. Together, the software and datasets introduced in this paper provide a robust computational pipeline for the analysis of social behavior in pairs of interacting mice and establish essential measures of reliability and sources of variability in human annotations of animal pose and behavior.

Le et al. [125] propose a framework that uses a 3D Convolutional network (ConvNet) to extract short-term spatio-temporal features from overlapped short clips. Then those local features are fed to a Long Short Term Memory network to learn long-term features which are used for classification. The framwork is denoted as LSTM-3DCNN, and the paper shows how to learn local spatio-temporal behavioral features using a 3D ConvNet and recognize behaviors in long videos with an LSTM network.

Kramida et al. [126] presents a mice behavior classification method based on LSTM. The method employs an end-to-end learning approach where visual features from pre-trained CNN are extracted from each image frame and is used to train a customized LSTM-based model in weakly-supervised fashion to recognize different behaviors of the mice in the videos. The classification framework relies on two deep learning mechanisms: pre-trained VGG features and LSTM. In a preprocessing step, the independent multimodal background subtraction algorithm is used to segment out the mouse.

We also present some state-of-the-art research in human action recognition. Deep learning-based human action recognition methods can be simply classified as skeleton-based and video-based according to whether or not to detect human keypoints first.

For video-based action recognition methods, most of the network structures are based on Two-stream/Multi-stream 2D CNN [127–129], RNN [130,131], and 3D CNN [132,133]. The two-stream 2D CNN framework generally contains two 2D CNN branches taking different input features extracted from the RGB videos for Human Action Recognition (HAR), and the final result is usually obtained through fusion strategies. Zong et al. [127] present Motion Saliency based multi-stream Multiplier ResNets (MSM-ResNets) method for action recognition. They extended the two-stream CNN in [128] to a three-stream CNN by adding the motion saliency stream to better capture the salient motion information. Zhang et al. [129] propose two video super-resolution methods producing high resolution videos, fed to the spatial and temporal streams to predict the action class. RNN-based models usually employ 2D CNNs, which serve as feature extractors, followed by an LSTM model for HAR. Majd et al. [130] proposed a C^2 LSTM which incorporates convolution and cross-correlation operators to learn motion and spatial features while modeling temporal dependencies. He et al. [131] adopt the Bi-directional LSTM, which consists of two independent LSTMs to learn both the forward and backward temporal information. The 3D CNN-based methods are very powerful in modeling discriminative features from both the spatial and temporal dimensions for HAR. 3D CNN model (C3D) [132] learns the spatio-temporal features from raw videos in an end-to-end learning framework. Fayyaz et al. [133] address the problem of dynamically adapting the temporal feature resolution within the 3D CNNs to reduce their computational cost. A Similarity Guided Sampling (SGS) module is

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

proposed to enable 3D CNNs to dynamically adapt their computational resources by selecting the most informative and distinctive temporal features.

For skeleton-based action recognition methods, most of the network structures used in them are based on RNN [134], CNN [135], and GCN [136]. RNNs and their gated variants (e.g., LSTMs) are capable of learning the dynamic dependencies in sequential data. Various methods have applied and adapted RNNs and LSTMs to effectively model the temporal context information within the skeleton sequences for HAR. Li et al. [134] propose a new type of RNNs called Independently Recurrent Neural Network (IndRNN) with the recurrent connection formulated as Hadamard product. IndRNN with regulated recurrent weights effectively addresses the gradient vanishing and exploding problems and thus long-term dependencies can be learned. CNNs have achieved great success in 2D image analysis due to their superior capability in learning features in the spatial domain. Zhang et al. [135] propose a novel view adaptation scheme for skeleton-based human action recognition. They introduce two view adaptive neural networks, VA-RNN and VA-CNN, which are respectively built based on the recurrent neural network (RNN) with the Long Short-term Memory (LSTM) and the convolutional neural network (CNN). Skeleton data is naturally in the form of graphs. Hence, simply representing skeleton data as a vector sequence processed by RNNs, or 2D/3D maps processed by CNNs, cannot fully model the complex spatio-temporal configurations and correlations of the body joints. As a result, many GNN and GCN-based HAR methods have been proposed to treat the skeleton data as graph structures of edges and nodes. Song et al. [136] propose a multistream GCN model, which fuses the input branches including joint positions, motion velocities, and bone features at early stage, and utilized separable convolutional layers and a compound scaling strategy to extremely reduce the redundant trainable parameters while increasing the capacity of model.

Table 9. Summary of Studies on Action Recognition.

Reference	Architecture	Type	Category	Dataset	Performance
[38]	Hourglass network	Mice	video-based	Private	Provide a robust computational pipeline for the analysis of social behavior in pairs of interacting mice
[125]	3D ConvNet, LSTM network	Mice	video-based	Private	Obtain accuracy on par with human assessment
[126]	LSTM	Mice	video-based	Private	Producing errors of 3.08%, 14.81%, and 7.4% on the training, validation, and testing sets respectively
[127]	2D CNN	Human	video-based	UCF101 and HMDB51 datasets	Outperforms other compared state-of-the-art models
[129]	2D CNN	Human	video-based	UCF101 and HMDB51 datasets	Improve the recognition performance of LR video from 42.81% to 53.59% on spatial stream and from 56.54% to 61.5% on temporal stream.
[131]	RNN	Human	video-based	UCF101 and HMDB51 datasets	Outperforms the state-of-the-art approaches for action recognition
[133]	3D CNN	Human	video-based	Kinetics-600, Kinetics-400, mini-Kinetics, Something-Something V2, UCF101, and HMDB51 datasets	SGS decreases the computation cost (GFLOPS) between 33% and 53% without compromising accuracy.
[134]	RNN	Human	skeleton-based	Penn Treebank (PTB-c), and NTU RGB+D datasets	Performs much better than the traditional RNN, LSTM, and Transformer models on sequential MNIST classification, language modeling, and action recognition tasks.
[135]	CNN	Human	skeleton-based	NTU RGB+D, the SYSU Human-Object Interaction, the UWA3D, the Northwestern-UCLA, and the SBU Kinect Interaction datasets	Superior performance over state-of-the-art approaches
[136]	GCN	Human	skeleton-based	NTU RGB+D 60 and 120 datasets	Outperforms other SOTA methods

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

4.5.3. Action Prediction

In some studies, mice behavior needs long-term or short-term observation. The features of mice behavior relate to temporal information. Therefore, the action prediction task needs to relate the context of the former mice behavior to predict the future mice behavior. Nowadays, temporal context information prediction can be categorized into three parts: short-term temporal context, long-term temporal context, and temporal semantic context. Existing studies of mice context behavior focus on the LSTM models. They also combine the lower layers' techniques, such as semantic segmentation and key point detection. For example, Kramida et al. [126] present a long-term mice behavior prediction method based on a LSTM model. Before the LSTM model, Pre-trained VGG and the independent multimodal background subtraction algorithm help segment the mice from the video. It combines with the semantic segmentation techniques. Then the LSTM is set up to predict the behavior sequences. The LSTM contains its own sets of input-unit weight, hidden-layer weight, and bias matrices, a time-propagating cell unit, input, output, and forget gates. Jiang et al. [53] improv e the LSTM model of [105], and propose a hidden Markov model to describe the short-term temporal characteristics of mice behavior. Before hidden Markov model, they make key point detection to detect the interest points of mice, and transforme the points into spatial-temporal segment Fisher Vector as the input of segment aggregated network. Then, hidden Markov model is used to infer latent or hidden states from the observed sequential data, and to account for the dynamics of the observed sequential data according to the dynamics of the hidden states. It is a discrete-time model where they receive an observation generated by a hidden state at each time instance. In summary, the action prediction task aims to connect the data context to extract the feature.

The state-of-the-art studies on temporal context prediction apply the attention mechanism and transformer framework, increasing the prediction accuracy and efficiency. The studies are summarized in Table 10. For short-term temporal context, Zang et al. [137] propose a MultiParallel Attention Network (MPAN) model to learn users' short-term interests by capturing contextual information and temporal signals simultaneously in a recommendation system. They propose a interest learning module and a interest fusion module to accurately capture users' short-term interests. The interest learning module consists of three parts: an embedding layer, a short-term interest generator and a long-term interest generator. The short-term interest generator utilizes a time-aware attention mechanism to learn short-term interests. The long-term interest generator employs the multi-head attention mechanism to extract the long-term purpose within the session from different semantic aspects. In the interest fusion module, a bi-linear similarity function is utilized to compute the recommendation score for each candidate item. The input is the session prefix, and the output is a one-hot encoding vector. At last, they utilize MPAN to predict the user's short-term interest.

For the long-term temporal context, Guo et al. [138] propose a transformer-based spatial-temporal graph neural network (ASTGNN) for long-term traffic forecasting. ASTGNN follows an encoder-decoder structure. The encoder and decoder in this model utilize multiple temporal trend-aware self-attention blocks and spatial dynamic GCN blocks alternatingly. The model is auto-regressive, meaning that it uses previously generated data as additional input when generating the next step. To capture the temporal dynamics of traffic data and have global receptive fields, a novel self-attention mechanism is designed for numerical sequence representation transformation. This self-attention mechanism is specialized for utilizing local context and maps a query and a set of key-value pairs to an output. The output is a weighted sum of the values, with the weight for each value determined by the corresponding key and the query. In the spatial dimension, they develop a dynamic graph convolution module, employing self-attention to capture the spatial correlations dynamically. The module employs self-attention and contains both a spatial-temporal encoder and a decoder. The encoder comprises a stack of identical layers, each containing two basic blocks: a temporal trend-aware multi-head self-attention block and a spatial dynamic GCN block. The decoder generates output sequences in an auto-regressive manner.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

For the temporal semantic context, Zhang et al. [139] propose a multi-temporal resolution pyramid structure model (MTSCANet) to realize temporal action localization efficiently. MTSCANet utilizes temporal semantic context fusion (TSCF) to fuse three feature sequences with different temporal resolutions into temporal and semantic contexts, respectively. The local-global attention module (LGAM) is used to encode the input temporal features in local-global temporal order, while the norm and location regularization are used to produce the final result. TSCF is employed to extract temporal semantic features, which are then input to LGAM for local-global timing coding to enhance feature robustness and enrich feature information. The three feature sequences are merged, processed again by LGAM and TSCF, and finally output proper vectors.

Table 10. Summary of Studies on Action Prediction.

Reference	Architecture	Type	Category	Dataset	Performance
[126]	RNN with LSTM	Mice	long-term	COCO, MPII	PCKh value is 92.3 in MPII and AP value is 75.5 in COCO
[53]	hidden Markov model (HVV)	Mice	long-term	Private, JHuang's datasets	Achieve weighted average accuracy of 96.5% (using visual and context features) and 97.9% (incor porated with IDT and TDD features)
[137]	MultiParallel Attention Network (MPAN)	Recommendation	short-term	YOOCHOSE and DIGENTICA	Obtain the best ISLF
[138]	Spatial-Temporal Graph Neural Network (ASTGNN)	Traffic forecasting	long-term	Caltrans Perfor mance Measurement System (PeMS)	Get the best performance in MAE, RMSE and MAPE.
[139]	Multi-temporal resolution pyramid structure model (MTSCANet)	Videos	temporal semantic context	THUMOS14, ActivityNet-1.3, HACS	An average mAP of 47.02% on THUMOS14, an average mAP of 34.94% on ActivityNet-1.3 and an average mAP of 28.46% on HACS

5. MiceGPT Design

In the above sections, we introduce the AI-empowered mice analysis applications and the corresponding state-of-art approaches to enhance the research process in biology fields. However, we still lack integrated AI systems, such as ChatGPT and VisualGPT [140], to perform the autonomous mice behavior analysis. In this section, we propose an architecture called MiceGPT and its variations to fufill the AI-empowered automated mice behavior anlaysis in the biolog realted fields.

5.1. Fundamental Architecture Design

The MiceGPT architecture overview is shown in Figure 8, which consists of five layers, namely, a query layer, an application layer, a storage layer, an AI model layer, and a data layer.

The data layer provides the interfaces to connect with different data sources. In the architecture, most of the data types are images and videos. Therefore, the data layer should support the different encoders and decoders of images and videos, such as jpg, png for images, and h264 videos h265 videos [141]. Furthermore, in practice, the data source might be a media stream, which the data layer should consider.

The AI model layer is the core component of the GPT framework, encompassing the design, training, and inference of deep learning models. This layer employs various deep learning models and architectures, such as Transformer, to process and analyze data, extract features, and perform model training and prediction. The objective of the AI model layer is to leverage the data provided by the data layer for model training, enabling the ability to make predictions or generate outputs for the given task, such as object tracking, pose estimation, object detection etc.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

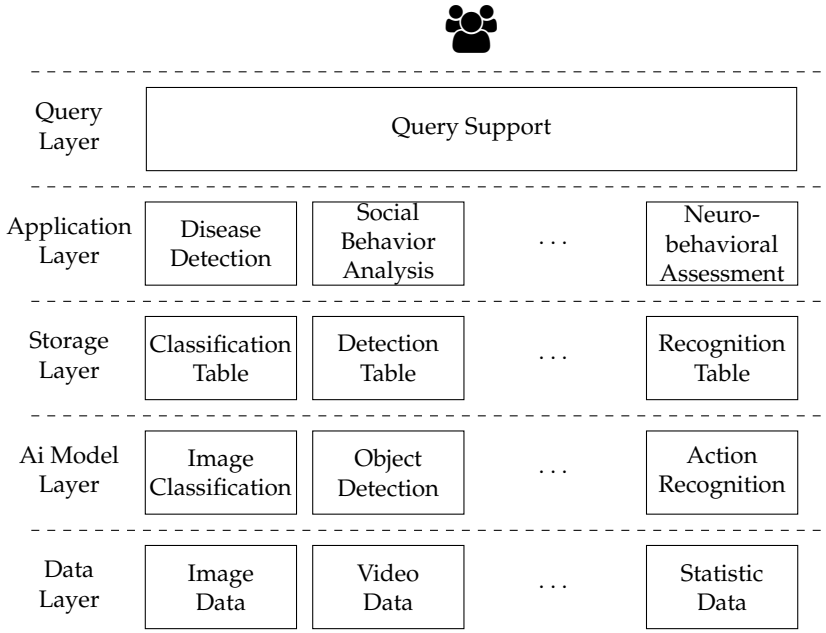

Figure 8. MiceGPT Architecture Overview.

The storage layer is responsible for managing and storing various data and models within the GPT framework. This includes data storage, access, and retrieval, as well as storage and management of model parameters. The storage layer can employ various technologies and tools, such as databases and distributed file systems, to efficiently handle large-scale data and models.

The application layer consists of functional components built upon the AI model layer, addressing specific problems. This layer utilizes the capabilities provided by the AI model layer to develop various applications or services based on specific application requirements, such as disease detection, social behavior analysis, neuro-behavioral assessment, and so on. The application layer can include applications such as image recognition, natural language processing, recommendation systems, and other types of applications.

The query layer serves as the interface layer for user interactions, responsible for receiving user requests and forwarding them to the respective application layer. This layer processes user inputs, providing functionalities such as result querying, question answering, and information retrieval, and presents the results returned by the application layer to the user. The query layer can encompass various user interfaces, such as command-line interfaces, graphical interfaces, or web services.

Through the collaborative work of the data layer, AI model layer, storage layer, application layer, and query layer, the GPT framework accomplishes data preparation and processing, training and inference of deep learning models, storage and management of data and models, as well as interaction with users and implementation of application functionalities. This layered architecture enhances the flexibility, scalability, and adaptability of the GPT framework, enabling it to cater to various tasks and application requirements.

5.2. AI-empowered Query Layer

In MiceGPT, the query layer is designed to fulfill the user's query request. The query layer is implemented by the SQL-like language [142]. However, SQL-like language needs a certain learning process about the database and language itself, which might be difficult for researchers in biology-related fields. Therefore, in this section, we introduce an AI-empowered query layer to enhance the usability and interactivity of MiceGPT.

The AI-empowered query layer can be analogized to an intelligent application's request handler. When a user submits a request, the AI-empowered query layer goes through parsing and classification

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 July 2023 doi:10.20944/preprints202307.0271.v1

processes. Ultimately, the query layer assigns the query to an appropriate application. Note that the input of the AI-empowered query layer is nature languages rather than SQL. Users only need to describe their requirements and desired results. Then, the query layer would invoke the proper AI application and configure the data source automatically. The request analysis process resembles the self-attention and multi-head attention mechanisms like VisualGPT [140], which determine attention weights based on input data features and relationships to understand better and process the input information.

Once the request is classified into the corresponding application, the application invokes a pre-trained AI model to process the request and generate results. The functionality is like the visual feature modulation of VisualGPT, which incorporates visual features into the generation process. The AI-empowered query layer, guided by pre-trained natural language models, processes the input data and produces suitable output.

Finally, the application returns the processed results to the user. This mirrors VisualGPT's transmission of fused image-text feature representations to the decoder, resulting in the final output generation. The entire process, involving the query layer's request parsing and classification, as well as the application's invocation of pre-trained AI model results, enables intelligent applications to generate appropriate responses based on user requests.

5.3. AI-empowered Application Layer

In MiceGPT, the application layer is designed to fulfill research objectives. In the fundamental architecture design, the application layer is implemented by software developers and biology field experts. However, with the increasing research requirements, the application layer may grow exponentially. Besides, the requirement analysis process is time-consuming because of the gap between the computer field and the biology-related fields.

The application layer connects with the AI model layer and the storage layer. The analysis applications can be considered as processes, in which the AI models are called, and data is operated in the storage layer. Namely, the application ties to using tools provided by the AI model and storage layers to fulfill the user's research task. Combining existing AI techniques, such as AutoGPT [143], we propose an AI-empowered application layer to enhance MiceGPT.

Unlike the application layer in the fundamental architecture design, the AI-empowered application layer focuses on iterative prompt learning to finish the general tasks rather than specific processes for each application. The iterative prompt learning process includes the following steps: (1) The AI-empowered application layer automatically generates prompts based on specific strategies, initially including the user's input of name, role, and objectives. (2) The AI-empowered application layer communities with a generative large language model, such as ChatGPT, to ask the command prompts for the next step to fulfill the user's objectives. (3) The commands generated in step 2 are highly extensible, with each command representing a distinct external capability, such as web scraping, google search, calling a pre-defined AI model, and communicating with the storage layer. The result obtained through invoking these commands then becomes a constituent element of the command prompt. (4) The process returns to step 1 and iterates until the final result is obtained with the state being "complete".

With the AI-empowered application layer, MiceGPT is able to finish the user's research task ultimately. The difference between the AI-empowered application layer and AutoGPT is that the application layer is defined to fulfill mice behavior analysis research and it must have the ability to call for pre-defined AI models and the storage layer.

 doi:10.20944/preprints202307.0271.v1

6. Conclusions

In this paper, we mainly focus on the AI-empowered mice behavior analysis field. Firstly, we summarize applications that use AI-empowered mice behavior analysis methods, including disease detection, external stimuli effective assessment, social behavior analysis, and neurobehavioral assessment. Then, we analyze the AI techniques behind these applications. Furthermore, we introduce some related state-of-art deep learning models to inspire the following research. Last, we propose a MiceGPT architecture that integrates AI techniques and mice behavior analysis applications to provide easy-to-use tools for biology-related researchers.

While summarizing, we figure out there are still open challenges in AI-empowered mice behavior analysis research.

Firstly, AI technology is widely applied in the field of behavior analysis research, specifically in the study of human behavior patterns and psychological states, providing in-depth analysis and understanding. However, the application of AI technology in mice behavior analysis research is relatively limited, resulting in certain constraints on the detailed interpretation of mice behavior and the in-depth analysis of behavior patterns. This disparity limits our comprehensive understanding of mice behavior and cognition, as well as restricts the application of mice models in areas such as disease research and drug development.

Secondly, there is a lack of sufficient datasets and benchmarks, and on the other hand, different applications have diverse requirements for datasets. This shortcoming of datasets and benchmarks restricts the training and evaluation of AI models, hinders research progress, and comparisons across different application domains. Therefore, establishing comprehensive and diverse datasets and benchmarks, tailored to specific application needs, becomes a crucial measure for advancing the development and application of AI technology on mice behavior analysis.

Thirdly, the lacking of an AI testbed in the mice behavior analysis field is a challenge. Currently, the common AI platform for mice behavior analysis is DeepLabCut. However, DeepLabCut only offers the fundamental steps of mice behavior analysis. It supports the AI model layer and part of the application layer of MiceGPT system. Therefore, integrating AI with mice behavior analysis is a challenge now.

Lastly, current large language models are widely used in the field of Natural language processing, and the related technologies and methods of large language models are gradually introduced into the field of computer vision. However, in biology-related fields and mice behavior recognition field, there has been no relevant application or research on large language models. In our MiceGPT design, we introduce the large language models as our query layer to simplify the usage of MiceGPT for biology-related researchers. Considering the potential of big language models, it is necessary to increase the research and application of large language models in the field of biology.

Author Contributions: Conceptualization, C.G., and J.S.; methodology, C.G. and Y.C.; software, Y.C.; validation, C.G, and Y.H.; formal analysis, C.G.; investigation, Y.H.; resources, C.X.; data curation, C.X.; writing—original draft preparation, C.G., and Y.C.; writing—review and editing, J.S. and H.S.; visualization, Y.C.; supervision, J.S., and H.S.; project administration, C.G.; funding acquisition, C.G. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported by Fundamental Research Funds for the Central Universities (No. N2317005).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

Abbreviations

The following abbreviations are used in this manuscript:

Abbreviation	Full Name
AI	Artificial Intelligence
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
CRNN	Convolutional Recurrent Neural Network
FC	Fully Connected
SVMs	Support Vector Machines
KNN	K-nearest Neighbours
MARS	Mouse Action Recognition System
RPN	Region Propose Network
ROI	Region of Interest
IoU	Intersection over Union
RNN	Recurrent Neural Networks
GCN	Graph Convolution Network

References

1. Manavalan.; Basith.; Shin.; Lee.; Wei.; Lee. 4mCpred-EL: An Ensemble Learning Framework for Identification of DNA N4-methylcytosine Sites in the Mouse Genome. *Cells* **2019**, *8*, 1332. doi:10.3390/cells8111332.
2. Koehler, C.C.; Hall, L.M.; Hellmer, C.B.; Ichinose, T. Using Looming Visual Stimuli to Evaluate Mouse Vision. *Journal of Visualized Experiments* **2019**, *148*, 59766. doi:10.3791/59766.
3. Taherzadeh, G.; Yang, Y.; Xu, H.; Xue, Y.; Liew, A.W.C.; Zhou, Y. Predicting Lysine-Malonylation Sites of Proteins Using Sequence and Predicted Structural Features. *Journal of Computational Chemistry* **2018**, *39*, 1757–1763. doi:10.1002/jcc.25353.
4. Pearson, B.L.; Defensor, E.B.; Blanchard, D.C.; Blanchard, R.J. C57BL/6J Mice Fail to Exhibit Preference for Social Novelty in the Three-Chamber Apparatus. *Behavioural Brain Research* **2010**, *213*, 189–194. doi:10.1016/j.bbr.2010.04.054.
5. Kulesskaya, N.; Voikar, V. Assessment of Mouse Anxiety-like Behavior in the Light–Dark Box and Open-Field Arena: Role of Equipment and Procedure. *Physiology & Behavior* **2014**, *133*, 30–38. doi:10.1016/j.physbeh.2014.05.006.
6. Seo, M.K.; Jeong, S.; Seog, D.H.; Lee, J.A.; Lee, J.H.; Lee, Y.; McIntyre, R.S.; Park, S.W.; Lee, J.G. Effects of Liraglutide on Depressive Behavior in a Mouse Depression Model and Cognition in the Probe Trial of Morris Water Maze Test. *Journal of Affective Disorders* **2023**, *324*, 8–15. doi:10.1016/j.jad.2022.12.089.
7. Bohnslav, J.P.; Wimalasena, N.K.; Clausing, K.J.; Dai, Y.Y.; Yarmolinsky, D.A.; Cruz, T.; Kashlan, A.D.; Chiappe, M.E.; Orefice, L.L.; Woolf, C.J.; Harvey, C.D. DeepEthogram, a Machine Learning Pipeline for Supervised Behavior Classification from Raw Pixels. *eLife* **2021**, *10*, e63377. doi:10.7554/eLife.63377.
8. Egnor, S.R.; Branson, K. Computational Analysis of Behavior. *Annual Review of Neuroscience* **2016**, *39*, 217–236. doi:10.1146/annurev-neuro-070815-013845.
9. Alexandrov, V.; Brunner, D.; Menalled, L.B.; Kudwa, A.; Watson-Johnson, J.; Mazzella, M.; Russell, I.; Ruiz, M.C.; Torello, J.; Sabath, E.; Sanchez, A.; Gomez, M.; Filipov, I.; Cox, K.; Kwan, M.; Ghavami, A.; Ramboz, S.; Lager, B.; Wheeler, V.C.; Aaronson, J.; Rosinski, J.; Gusella, J.F.; MacDonald, M.E.; Howland, D.; Kwak, S. Large-Scale Phenome Analysis Defines a Behavioral Signature for Huntington's Disease Genotype in Mice. *Nature Biotechnology* **2016**, *34*, 838–844. doi:10.1038/nbt.3587.
10. Geuther, B.; Chen, M.; Galante, R.J.; Han, O.; Lian, J.; George, J.; Pack, A.I.; Kumar, V. High-Throughput Visual Assessment of Sleep Stages in Mice Using Machine Learning. *Sleep* **2022**, *45*, zsab260. doi:10.1093/sleep/zsab260.
11. Taecharungroj, V. "What Can ChatGPT Do?" analyzing early reactions to the innovative AI chatbot on twitter. *Big Data Cogn. Comput.* **2023**, *7*, 35. doi:10.3390/bdcc7010035.

12. Vogel-Ciernia, A.; Matheos, D.P.; Barrett, R.M.; Kramár, E.A.; Azzawi, S.; Chen, Y.; Magnan, C.N.; Zeller, M.; Sylvain, A.; Haettig, J.a. The neuron-specific chromatin regulatory subunit BAF53b is necessary for synaptic plasticity and memory. *Nature Neuroscience* **2015**, *16*, 552–61.
13. Kalueff, A.V.; Stewart, A.M.; Song, C.; Berridge, K.C.; Graybiel, A.M.; Fentress, J.C. Neurobiology of Rodent Self-Grooming and Its Value for Translational Neuroscience. *Nature Reviews Neuroscience* **2016**, *17*, 45–59. doi:10.1038/nrn.2015.8.
14. Houle, D.; Govindaraju, D.R.; Omholt, S. Phenomics: The next Challenge. *Nature Reviews Genetics* **2010**, *11*, 855–866. doi:10.1038/nrg2897.
15. Lee, K.; Park, I.; Bishayee, K.; Lee, U. Machine-Learning Based Automatic and Real-Time Detection of Mouse Scratching Behaviors. *IBRO Reports* **2019**, *6*, S414–S415. doi:10.1016/j.ibror.2019.07.1317.
16. Sakamoto, N.; Haraguchi, T.; Kobayashi, K.; Miyazaki, Y.; Murata, T. Automated Scratching Detection System for Black Mouse Using Deep Learning. *Frontiers in Physiology* **2022**, *13*, 939281. doi:10.3389/fphys.2022.939281.
17. Viglione, A.; Sagona, G.; Carrara, F.; Amato, G.; Totaro, V.; Lupori, L.; Putignano, E.; Pizzorusso, T.; Mazziotti, R. Behavioral Impulsivity Is Associated with Pupillary Alterations and Hyperactivity in CDKL5 Mutant Mice. *Human Molecular Genetics* **2022**, *31*, 4107–4120. doi:10.1093/hmg/ddac164.
18. Yu, H.; Xiong, J.; Ye, A.Y.; Cranfill, S.L.; Cannonier, T.; Gautam, M.; Zhang, M.; Bilal, R.; Park, J.E.; Xue, Y.; Polam, V.; Vujovic, Z.; Dai, D.; Ong, W.; Ip, J.; Hsieh, A.; Mimouni, N.; Lozada, A.; Sosale, M.; Ahn, A.; Ma, M.; Ding, L.; Arsuaga, J.; Luo, W. Scratch-AID, a Deep Learning-Based System for Automatic Detection of Mouse Scratching Behavior with High Accuracy. *eLife* **2022**, *11*, e84042. doi:10.7554/eLife.84042.
19. Weber, R.Z.; Mulders, G.; Kaiser, J.; Tackenberg, C.; Rust, R. Deep Learning-Based Behavioral Profiling of Rodent Stroke Recovery. *BMC Biology* **2022**, *20*, 232. doi:10.1186/s12915-022-01434-9.
20. Aljovic, A.; Zhao, S.; Chahin, M.; De La Rosa, C.; Van Steenbergen, V.; Kerschensteiner, M.; Bareyre, F.M. A Deep Learning-Based Toolbox for Automated Limb Motion Analysis (ALMA) in Murine Models of Neurological Disorders. *Communications Biology* **2022**, *5*, 131. doi:10.1038/s42003-022-03077-6.
21. Mathis, A.; Mamidanna, P.; Cury, K.M.; Abe, T.; Murthy, V.N.; Mathis, M.W.; Bethge, M. DeepLabCut: Markerless Pose Estimation of User-Defined Body Parts with Deep Learning. *Nature Neuroscience* **2018**, *21*, 1281–1289. doi:10.1038/s41593-018-0209-y.
22. Cai, H.; Luo, Y.; Yan, X.; Ding, P.; Huang, Y.; Fang, S.; Zhang, R.; Chen, Y.; Guo, Z.; Fang, J.; Wang, Q.; Xu, J. The Mechanisms of Bushen-Yizhi Formula as a Therapeutic Agent against Alzheimer's Disease. *Scientific Reports* **2018**, *8*, 3104. doi:10.1038/s41598-018-21468-w.
23. Iino, Y.; Sawada, T.; Yamaguchi, K.; Tajiri, M.; Ishii, S.; Kasai, H.; Yagishita, S. Dopamine D2 Receptors in Discrimination Learning and Spine Enlargement. *Nature* **2020**, *579*, 555–560. doi:10.1038/s41586-020-2115-1.
24. Merlini, M.; Rafalski, V.A.; Rios Coronado, P.E.; Gill, T.M.; Ellisman, M.; Muthukumar, G.; Subramanian, K.S.; Ryu, J.K.; Syme, C.A.; Davalos, D.; Seeley, W.W.; Mucke, L.; Nelson, R.B.; Akassoglou, K. Fibrinogen Induces Microglia-Mediated Spine Elimination and Cognitive Impairment in an Alzheimer's Disease Model. *Neuron* **2019**, *101*, 1099–1108.e6. doi:10.1016/j.neuron.2019.01.014.
25. Wotton, J.M.; Peterson, E.; Anderson, L.; Murray, S.A.; Braun, R.E.; Chesler, E.J.; White, J.K.; Kumar, V. Machine Learning-Based Automated Phenotyping of Inflammatory Nocifensive Behavior in Mice. *Molecular Pain* **2020**, *16*, 174480692095859. doi:10.1177/1744806920958596.
26. Kathote, G.; Ma, Q.; Angulo, G.; Chen, H.; Jakkamsetti, V.; Dobariya, A.; Good, L.B.; Posner, B.; Park, J.Y.; Pascual, J.M. Identification of Glucose Transport Modulators In Vitro and Method for Their Deep Learning Neural Network Behavioral Evaluation in Glucose Transporter 1–Deficient Mice. *Journal of Pharmacology and Experimental Therapeutics* **2023**, *384*, 393–405. doi:10.1124/jpet.122.001428.
27. Vidal, A.; Jha, S.; Hassler, S.; Price, T.; Busso, C. Face Detection and Grimace Scale Prediction of White Furred Mice. *Machine Learning with Applications* **2022**, *8*, 100312. doi:10.1016/j.mlwa.2022.100312.
28. Abdus-Saboor, I.; Fried, N.T.; Lay, M.; Burdge, J.; Swanson, K.; Fischer, R.; Jones, J.; Dong, P.; Cai, W.; Guo, X.; Tao, Y.X.; Bethea, J.; Ma, M.; Dong, X.; Ding, L.; Luo, W. Development of a Mouse Pain Scale Using Sub-second Behavioral Mapping and Statistical Modeling. *Cell Reports* **2019**, *28*, 1623–1634.e4. doi:10.1016/j..2019.07.017.
29. Marks, M.; Jin, Q.; Sturman, O.; Von Ziegler, L.; Kollmorgen, S.; Von Der Behrens, W.; Mante, V.; Bohacek, J.; Yanik, M.F. Deep-Learning-Based Identification, Tracking, Pose Estimation and Behaviour Classification

of Interacting Primates and Mice in Complex Environments. *Nature Machine Intelligence* **2022**, *4*, 331–340. doi:10.1038/s42256-022-00477-5.

30. Torabi, R.; Jenkins, S.; Harker, A.; Whishaw, I.Q.; Gibb, R.; Luczak, A. A Neural Network Reveals Motoric Effects of Maternal Preconception Exposure to Nicotine on Rat Pup Behavior: A New Approach for Movement Disorders Diagnosis. *Frontiers in Neuroscience* **2021**, *15*, 686767. doi:10.3389/fnins.2021.686767.
31. Martins, T.M.; Brown Driemeyer, J.P.; Schmidt, T.P.; Sobieranski, A.C.; Dutra, R.C.; Oliveira Weber, T. A Machine Learning Approach to Immobility Detection in Mice during the Tail Suspension Test for Depressive-Type Behavior Analysis. *Research on Biomedical Engineering* **2022**, *39*, 15–26. doi:10.1007/s42600-022-00246-8.
32. Wang, J.; Karbasi, P.; Wang, L.; Meeks, J.P. A Layered, Hybrid Machine Learning Analytic Workflow for Mouse Risk Assessment Behavior. *eneuro* **2023**, *10*, ENEURO.0335–22.2022. doi:10.1523/ENEURO.0335-22.2022.
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Las Vegas, NV, USA, 2016; pp. 770–778. doi:10.1109/CVPR.2016.90.
34. Bermudez Contreras, E.; Sutherland, R.J.; Mohajerani, M.H.; Whishaw, I.Q. Challenges of a Small World Analysis for the Continuous Monitoring of Behavior in Mice. *Neuroscience & Biobehavioral Reviews* **2022**, *136*, 104621. doi:10.1016/j.neubiorev.2022.104621.
35. Gharagozloo, M.; Amrani, A.; Wittingstall, K.; Hamilton-Wright, A.; Gris, D. Machine Learning in Modeling of Mouse Behavior. *Frontiers in Neuroscience* **2021**, *15*, 700253. doi:10.3389/fnins.2021.700253.
36. Van Dam, E.A.; Noldus, L.P.; Van Gerven, M.A. Deep Learning Improves Automated Rodent Behavior Recognition within a Specific Experimental Setup. *Journal of Neuroscience Methods* **2020**, *332*, 108536. doi:10.1016/j.jneumeth.2019.108536.
37. Robie, A.A.; Seagraves, K.M.; Egnor, S.E.R.; Branson, K. Machine Vision Methods for Analyzing Social Interactions. *Journal of Experimental Biology* **2017**, *220*, 25–34. doi:10.1242/jeb.142281.
38. Segalin, C.; Williams, J.; Karigo, T.; Hui, M.; Zelikowsky, M.; Sun, J.J.; Perona, P.; Anderson, D.J.; Kennedy, A. The Mouse Action Recognition System (MARS) Software Pipeline for Automated Analysis of Social Behaviors in Mice. *eLife* **2021**, *10*, e63720. doi:10.7554/eLife.63720.
39. Agbele, T.; Ojeme, B.; Jiang, R. Application of Local Binary Patterns and Cascade AdaBoost Classifier for Mice Behavioural Patterns Detection and Analysis. *Procedia Computer Science* **2019**, *159*, 1375–1386. doi:10.1016/j.procs.2019.09.308.
40. Jiang, Z.; Zhou, F.; Zhao, A.; Li, X.; Li, L.; Tao, D.; Li, X.; Zhou, H. Multi-View Mouse Social Behaviour Recognition With Deep Graphic Model. *IEEE Transactions on Image Processing* **2021**, *30*, 5490–5504. doi:10.1109/TIP.2021.3083079.
41. Sheets, A.L.; Lai, P.L.; Fisher, L.C.; Basso, D.M. Quantitative Evaluation of 3D Mouse Behaviors and Motor Function in the Open-Field after Spinal Cord Injury Using Markerless Motion Tracking. *PLoS ONE* **2013**, *8*, e74536. doi:10.1371/journal.pone.0074536.
42. Burgos-Artizzu, X.P.; Dollár, P.; Lin, D.; Anderson, D.J.; Perona, P. Social Behavior Recognition in Continuous Video. 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1322–1329. doi:10.1109/CVPR.2012.6247817.
43. Salem, G.; Krynitsky, J.; Hayes, M.; Pohida, T.; Burgos-Artizzu, X. Three-Dimensional Pose Estimation for Laboratory Mouse From Monocular Images. *IEEE Transactions on Image Processing* **2019**, *28*, 4273–4287. doi:10.1109/TIP.2019.2908796.
44. Su, W.; Jiang, F.; Shi, C.; Wu, D.; Liu, L.; Li, S.; Yuan, Y.; Shi, J. An XGBoost-Based Knowledge Tracing Model. *International Journal of Computational Intelligence Systems* **2023**, *16*, 13. doi:10.1007/s44196-023-00192-y.
45. Huang, X.; Li, Z.; Jin, Y.; Zhang, W. Fair-AdaBoost: Extending AdaBoost Method to Achieve Fair Classification. *Expert Systems With Applications* **2022**, *202*, 117240. doi:10.1016/j.eswa.2022.117240.
46. Winters, C.; Gorssen, W.; Ossorio-Salazar, V.A.; Nilsson, S.; Golden, S.; D'Hooge, R. Automated Procedure to Assess Pup Retrieval in Laboratory Mice. *Scientific Reports* **2022**, *12*, 1663. doi:10.1038/s41598-022-05641-w.
47. Hong, W.; Kennedy, A.; Burgos-Artizzu, X.P.; Zelikowsky, M.; Navonne, S.G.; Perona, P.; Anderson, D.J. Automated Measurement of Mouse Social Behaviors Using Depth Sensing, Video Tracking, and Machine Learning. *Proceedings of the National Academy of Sciences* **2015**, *112*. doi:10.1073/pnas.1515982112.

48. Tanas, J.K.; Kerr, D.D.; Wang, L.; Rai, A.; Wallaard, I.; Elgersma, Y.; Sidorov, M.S. Multidimensional Analysis of Behavior Predicts Genotype with High Accuracy in a Mouse Model of Angelman Syndrome. *Translational Psychiatry* **2022**, *12*, 426–434. doi:10.1038/s41398-022-02206-3.
49. Yamamoto, M.; Motomura, E.; Yanagisawa, R.; Hoang, V.A.T.; Mogi, M.; Mori, T.; Nakamura, M.; Takeya, M.; Eto, K. Evaluation of Neurobehavioral Impairment in Methylmercury-Treated KK-Ay Mice by Dynamic Weight-Bearing Test: Neurobehavioral Disorders in Methylmercury-Treated Mice. *Journal of Applied Toxicology* **2019**, *39*, 221–230. doi:10.1002/jat.3710.
50. Delanogare, E.; Bullich, S.; Barbosa, L.A.D.S.; Barros, W.D.M.; Braga, S.P.; Kraus, S.I.; Kasprowicz, J.N.; Dos Santos, G.J.; Guiard, B.P.; Moreira, E.L.G. Metformin Improves Neurobehavioral Impairments of Streptozotocin-treated and Western Diet-fed Mice: Beyond Glucose-lowering Effects. *Fundamental & Clinical Pharmacology* **2023**, *37*, 94–106. doi:10.1111/fcp.12825.
51. McMackin, M.Z.; Henderson, C.K.; Cortopassi, G.A. Neurobehavioral Deficits in the KIKO Mouse Model of Friedreich's Ataxia. *Behavioural Brain Research* **2017**, *316*, 183–188. doi:10.1016/j.bbr.2016.08.053.
52. Ren, Z.; Annie, A.N.; Ciernia, V.; Lee, Y.J. Who Moved My Cheese? Automatic Annotation of Rodent Behaviors with Convolutional Neural Networks. 2017 IEEE Winter Conference on Applications of Computer Vision (WACV); IEEE: Santa Rosa, CA, USA, 2017; pp. 1277–1286. doi:10.1109/WACV.2017.147.
53. Jiang, Z.; Crookes, D.; Green, B.D.; Zhao, Y.; Ma, H.; Li, L.; Zhang, S.; Tao, D.; Zhou, H. Context-Aware Mouse Behavior Recognition Using Hidden Markov Models. *IEEE Transactions on Image Processing* **2019**, *28*, 1133–1148. doi:10.1109/TIP.2018.2875335.
54. Tong, M.; Yu, X.; Shao, J.; Shao, Z.; Li, W.; Lin, W. Automated Measuring Method Based on Machine Learning for Optomotor Response in Mice. *Neurocomputing* **2020**, *418*, 241–250. doi:10.1016/j.neucom.2020.08.009.
55. Cai, L.X.; Pizano, K.; Gundersen, G.W.; Hayes, C.L.; Fleming, W.T.; Holt, S.; Cox, J.M.; Witten, I.B. Distinct Signals in Medial and Lateral VTA Dopamine Neurons Modulate Fear Extinction at Different Times. *eLife* **2020**, *9*, e54936. doi:10.7554/eLife.54936.
56. Jhuang, H.; Garrote, E.; Yu, X.; Khilnani, V.; Poggio, T.; Steele, A.D.; Serre, T. Correction: Corrigendum: Automated Home-Cage Behavioural Phenotyping of Mice. *Nature Communications* **2012**, *3*, 654. doi:10.1038/ncomms1399.
57. Lara-Doña, A.; Torres-Sanchez, S.; Priego-Torres, B.; Berrocoso, E.; Sanchez-Morillo, D. Automated Mouse Pupil Size Measurement System to Assess Locus Coeruleus Activity with a Deep Learning-Based Approach. *Sensors* **2021**, *21*, 7106. doi:10.3390/s21217106.
58. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection, 2016, [arxiv:cs/1506.02640].
59. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition, 2015, [arxiv:cs/1512.03385].
60. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, 2017, [arxiv:cs/1704.04861].
61. Newell, A.; Yang, K.; Deng, J. Stacked Hourglass Networks for Human Pose Estimation, 2016, [arxiv:cs/1603.06937].
62. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. Scaled-YOLOv4: Scaling Cross Stage Partial Network, 2021, [arxiv:cs/2011.08036].
63. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; Houlsby, N. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale, 2021, [arxiv:cs/2010.11929].
64. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows, 2021, [arxiv:cs/2103.14030].
65. Du, W.; Rao, N.; Yong, J.; Wang, Y.; Hu, D.; Gan, T.; Zhu, L.; Zeng, B. Improving the Classification Performance of Esophageal Disease on Small Dataset by Semi-supervised Efficient Contrastive Learning. *Journal of Medical Systems* **2022**, *46*, 4. doi:10.1007/s10916-021-01782-z.
66. Xue, Z.; Yu, X.; Yu, A.; Liu, B.; Zhang, P.; Wu, S. Self-Supervised Feature Learning for Multimodal Remote Sensing Image Land Cover Classification. *IEEE Transactions on Geoscience and Remote Sensing* **2022**, *60*, 1–15. doi:10.1109/TGRS.2022.3190466.

67. Li, X.; Hu, X.; Qi, X.; Yu, L.; Zhao, W.; Heng, P.A.; Xing, L. Rotation-Oriented Collaborative Self-Supervised Learning for Retinal Disease Diagnosis. *IEEE Transactions on Medical Imaging* **2021**, *40*, 2284–2294. doi:10.1109/TMI.2021.3075244.
68. Taleb, A.; Rohrer, C.; Bergner, B.; De Leon, G.; Rodrigues, J.A.; Schwendicke, F.; Lippert, C.; Krois, J. Self-Supervised Learning Methods for Label-Efficient Dental Caries Classification. *Diagnostics* **2022**, *12*, 1237. doi:10.3390/diagnostics12051237.
69. Pogorelov, K.; Randel, K.R.; Griwodz, C.; Lange, T.D.; Halvorsen, P. KVASIR: A Multi-Class Image Dataset for Computer Aided Gastrointestinal Disease Detection. Acm on Multimedia Systems Conference, 2017.
70. Fu, H.; Li, F.; Orlando, J.; Bogunovic, H.; Sun, X.; Liao, J.; Xu, Y.; Zhang, S.; Zhang, X. Adam: Automatic detection challenge on age-related macular degeneration. *IEEE Dataport* **2020**.
71. Fang, H.; Li, F.; Wu, J.; Fu, H.; Sun, X.; Orlando, J.I.; Bogunović, H.; Zhang, X.; Xu, Y. PALM: Open Fundus Photograph Dataset with Pathologic Myopia Recognition and Anatomical Structure Annotation. *arXiv preprint arXiv:2305.07816* **2023**.
72. Hu, Y.; Ding, Z.; Ge, R.; Shao, W.; Huang, L.; Li, K.; Liu, Q. AFDetV2: Rethinking the Necessity of the Second Stage for Object Detection from Point Clouds. *Proceedings of the AAAI Conference on Artificial Intelligence* **2022**, *36*, 969–979. doi:10.1609/aaai.v36i1.19980.
73. Li, Z.; Tian, X.; Liu, X.; Liu, Y.; Shi, X. A Two-Stage Industrial Defect Detection Framework Based on Improved-YOLOv5 and Optimized-Inception-ResnetV2 Models. *Applied Sciences* **2022**, *12*, 834. doi:10.3390/app12020834.
74. Sun, G.; Hua, Y.; Hu, G.; Robertson, N. Efficient One-Stage Video Object Detection by Exploiting Temporal Consistency. In *Computer Vision – ECCV 2022*; Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G.M.; Hassner, T., Eds.; Springer Nature Switzerland: Cham, 2022; Vol. 13695, pp. 1–16. doi:10.1007/978-3-031-19833-5_1.
75. Zhou, J.; Feng, K.; Li, W.; Han, J.; Pan, F. TS4Net: Two-stage Sample Selective Strategy for Rotating Object Detection. *Neurocomputing* **2022**, *501*, 753–764. doi:10.1016/j.neucom.2022.06.049.
76. Zhou, Q.; Li, X.; He, L.; Yang, Y.; Cheng, G.; Tong, Y.; Ma, L.; Tao, D. TransVOD: End-to-End Video Object Detection With Spatial-Temporal Transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2023**, *45*, 7853–7869. doi:10.1109/TPAMI.2022.3223955.
77. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions, 2016, [arxiv:cs/1511.07122].
78. Wu, X.; Tao, Y.; He, G.; Liu, D.; Fan, M.; Yang, S.; Gong, H.; Xiao, R.; Chen, S.; Huang, J. Boosting Multilabel Semantic Segmentation for Somata and Vessels in Mouse Brain. *Frontiers in Neuroscience* **2021**, *15*, 610122. doi:10.3389/fnins.2021.610122.
79. Webb, J.M.; Fu, Y.H. Recent Advances in Sleep Genetics. *Current Opinion in Neurobiology* **2021**, *69*, 19–24. doi:10.1016/j.conb.2020.11.012.
80. Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; Agrawal, A. Context Encoding for Semantic Segmentation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; IEEE: Salt Lake City, UT, USA, 2018; pp. 7151–7160. doi:10.1109/CVPR.2018.00747.
81. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2018**, *40*, 834–848. doi:10.1109/TPAMI.2017.2699184.
82. Yang, M.; Yu, K.; Zhang, C.; Li, Z.; Yang, K. DenseASPP for Semantic Segmentation in Street Scenes. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; IEEE: Salt Lake City, UT, USA, 2018; pp. 3684–3692. doi:10.1109/CVPR.2018.00388.
83. Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P.H.; Zhang, L. Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Nashville, TN, USA, 2021; pp. 6877–6886. doi:10.1109/CVPR46437.2021.00681.
84. Strudel, R.; Garcia, R.; Laptev, I.; Schmid, C. Segmenter: Transformer for Semantic Segmentation. 2021 IEEE/CVF International Conference on Computer Vision (ICCV); IEEE: Montreal, QC, Canada, 2021; pp. 7242–7252. doi:10.1109/ICCV48922.2021.00717.
85. Yu, T.; Li, X.; Cai, Y.; Sun, M.; Li, P. S^2-MLP: Spatial-Shift MLP Architecture for Vision. 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV); IEEE: Waikoloa, HI, USA, 2022; pp. 3615–3624. doi:10.1109/WACV51458.2022.00367.

86. Shen, Y.; Cao, L.; Chen, Z.; Zhang, B.; Su, C.; Wu, Y.; Huang, F.; Ji, R. Parallel detection-and-segmentation learning for weakly supervised instance segmentation. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 8198–8208.
87. Korfhage, N.; Mühling, M.; Ringshandl, S.; Becker, A.; Schmeck, B.; Freisleben, B. Detection and Segmentation of Morphologically Complex Eukaryotic Cells in Fluorescence Microscopy Images via Feature Pyramid Fusion. *PLOS Computational Biology* **2020**, *16*, e1008179. doi:10.1371/journal.pcbi.1008179.
88. Zhou, T.; Wang, W.; Liu, S.; Yang, Y.; Van Gool, L. Differentiable Multi-Granularity Human Representation Learning for Instance-Aware Human Semantic Parsing. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Nashville, TN, USA, 2021; pp. 1622–1631. doi:10.1109/CVPR46437.2021.00167.
89. Wang, X.; Zhang, R.; Shen, C.; Kong, T.; Li, L. SOLO: A Simple Framework for Instance Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2021**, pp. 1–1. doi:10.1109/TPAMI.2021.3111116.
90. Li, B.r.; Zhang, J.k.; Liang, Y. PaFPN-SOLO: A SOLO-based Image Instance Segmentation Algorithm. 2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML). IEEE, 2022, pp. 557–564.
91. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *International journal of computer vision* **2010**, *88*, 303–338.
92. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer, 2014, pp. 740–755.
93. Zhao, J.; Li, J.; Cheng, Y.; Zhou, L.; Sim, T.; Yan, S.; Feng, J. Understanding Humans in Crowded Scenes: Deep Nested Adversarial Learning and A New Benchmark for Multi-Human Parsing. *ACM* **2018**.
94. Güler, R.A.; Neverova, N.; Kokkinos, I. DensePose: Dense Human Pose Estimation In The Wild. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
95. Xia, F.; Wang, P.; Chen, X.; Yuille, A.L. Joint Multi-person Pose Estimation and Semantic Part Segmentation. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Honolulu, HI, 2017; pp. 6080–6089. doi:10.1109/CVPR.2017.644.
96. Gupta, A.; Dollár, P.; Girshick, R. LVIS: A Dataset for Large Vocabulary Instance Segmentation. *IEEE* **2019**.
97. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
98. Wen, J.; Chi, J.; Wu, C.; Yu, X. Human Pose Estimation Based Pre-training Model and Efficient High-Resolution Representation. 2021 40th Chinese Control Conference (CCC); IEEE: Shanghai, China, 2021; pp. 8463–8468. doi:10.23919/CCC52363.2021.9549849.
99. Gong, F.; Li, Y.; Yuan, X.; Liu, X.; Gao, Y. Human Elbow Flexion Behaviour Recognition Based on Posture Estimation in Complex Scenes. *IET Image Processing* **2023**, *17*, 178–192. doi:10.1049/ipr2.12626.
100. Zang, Y.; Fan, C.; Zheng, Z.; Yang, D. Pose Estimation at Night in Infrared Images Using a Lightweight Multi-Stage Attention Network. *Signal, Image and Video Processing* **2021**, *15*, 1757–1765. doi:10.1007/s11760-021-01916-3.
101. Hong, F.; Lu, C.; Liu, C.; Liu, R.; Jiang, W.; Ju, W.; Wang, T. PGNet: Pipeline Guidance for Human Key-Point Detection. *Entropy* **2020**, *22*, 369. doi:10.3390/e22030369.
102. Zhou, F.; Jiang, Z.; Liu, Z.; Chen, F.; Chen, L.; Tong, L.; Yang, Z.; Wang, H.; Fei, M.; Li, L.; Zhou, H. Structured Context Enhancement Network for Mouse Pose Estimation. *IEEE Transactions on Circuits and Systems for Video Technology* **2022**, *32*, 2787–2801. doi:10.1109/TCSVT.2021.3098497.
103. Xu, Z.; Liu, R.; Wang, Z.; Wang, S.; Zhu, J. Detection of Key Points in Mice at Different Scales via Convolutional Neural Network. *Symmetry* **2022**, *14*, 1437. doi:10.3390/sym14071437.
104. Topham, L.K.; Khan, W.; Al-Jumeily, D.; Hussain, A. Human Body Pose Estimation for Gait Identification: A Comprehensive Survey of Datasets and Models. *ACM Computing Surveys* **2023**, *55*, 1–42. doi:10.1145/3533384.
105. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep High-Resolution Representation Learning for Human Pose Estimation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Long Beach, CA, USA, 2019; pp. 5686–5696. doi:10.1109/CVPR.2019.00584.
106. Cheng, B.; Xiao, B.; Wang, J.; Shi, H.; Huang, T.S.; Zhang, L. HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation. 2020 IEEE/CVF Conference on

Computer Vision and Pattern Recognition (CVPR); IEEE: Seattle, WA, USA, 2020; pp. 5385–5394. doi:10.1109/CVPR42600.2020.00543.

107. Yu, C.; Xiao, B.; Gao, C.; Yuan, L.; Zhang, L.; Sang, N.; Wang, J. Lite-HRNet: A Lightweight High-Resolution Network. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Nashville, TN, USA, 2021; pp. 10435–10445. doi:10.1109/CVPR46437.2021.01030.
108. Iskakov, K.; Burkov, E.; Lempitsky, V.; Malkov, Y. Learnable Triangulation of Human Pose. 2019 IEEE/CVF International Conference on Computer Vision (ICCV); IEEE: Seoul, Korea (South), 2019; pp. 7717–7726. doi:10.1109/ICCV.2019.00781.
109. He, Y.; Yan, R.; Fragkiadaki, K.; Yu, S.I. Epipolar Transformers. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Only, 2020; pp. 7779–7788.
110. Weinzaepfel, P.; Brégier, R.; Combaluzier, H.; Leroy, V.; Rogez, G. DOPE: Distillation of Part Experts for Whole-Body 3D Pose Estimation in the Wild. In *Computer Vision – ECCV 2020*; Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J.M., Eds.; Springer International Publishing: Cham, 2020; Vol. 12371, pp. 380–397. doi:10.1007/978-3-030-58574-7_23.
111. Wang, F.; Luo, L.; Zhu, E.; Wang, S. Multi-Object Tracking with a Hierarchical Single-Branch Network. In *MultiMedia Modeling*; Þór Jónsson, B.; Gurrin, C.; Tran, M.T.; Dang-Nguyen, D.T.; Hu, A.M.C.; Huynh Thi Thanh, B.; Huet, B., Eds.; Springer International Publishing: Cham, 2022; Vol. 13142, pp. 73–83. doi:10.1007/978-3-030-98355-0_7.
112. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2017**, *39*, 1137–1149. doi:10.1109/TPAMI.2016.2577031.
113. Vaquero, V.; Del Pino, I.; Moreno-Noguer, F.; Sola, J.; Sanfeliu, A.; Andrade-Cetto, J. Dual-Branch CNNs for Vehicle Detection and Tracking on LiDAR Data. *IEEE Transactions on Intelligent Transportation Systems* **2021**, *22*, 6942–6953. doi:10.1109/TITS.2020.2998771.
114. Geiger, A.; Lenz, P.; Urtasun, R. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. 2012 IEEE Conference on Computer Vision and Pattern Recognition; IEEE: Providence, RI, 2012; pp. 3354–3361. doi:10.1109/CVPR.2012.6248074.
115. Jiang, J.; Yang, X.; Li, Z.; Shen, K.; Jiang, F.; Ren, H.; Li, Y. MultiBSP: Multi-Branch and Multi-Scale Perception Object Tracking Framework Based on Siamese CNN. *Neural Computing and Applications* **2022**, *34*, 18787–18803. doi:10.1007/s00521-022-07420-0.
116. Milan, A.; Leal-Taixe, L.; Reid, I.; Roth, S.; Schindler, K. MOT16: A Benchmark for Multi-Object Tracking, 2016, [arxiv:cs/1603.00831].
117. Dendorfer, P.; Rezatofighi, H.; Milan, A.; Shi, J.; Cremers, D.; Reid, I.; Roth, S.; Schindler, K.; Leal-Taixé, L. MOT20: A benchmark for multi object tracking in crowded scenes. *arXiv* **2020**.
118. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research* **2013**, *32*, 1231–1237.
119. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; ˇCehovin Zajc, L.; Vojir, T.; Bhat, G.; Lukezic, A.; Eldesokey, A.; others. The sixth visual object tracking vot2018 challenge results. Proceedings of the European conference on computer vision (ECCV) workshops, 2018, pp. 0–0.
120. Kristan, M.; Berg, A.; Zheng, L.; Rout, L.; Zhou, L. The Seventh Visual Object Tracking VOT2019 Challenge Results. 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), 2019.
121. Wu, Y.; Lim, J.; Yang, M.H. Object tracking benchmark. *IEEE transactions on pattern analysis and machine intelligence* **2015**, *37*, 1834–1848.
122. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for uav tracking. Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer, 2016, pp. 445–461.
123. Huang, L.; Zhao, X.; Huang, K. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE transactions on pattern analysis and machine intelligence* **2019**, *43*, 1562–1577.
124. Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; Ling, H. Lasot: A high-quality benchmark for large-scale single object tracking. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 5374–5383.

125. Le, V.A.; Murari, K. Recurrent 3D Convolutional Network for Rodent Behavior Recognition. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); IEEE: Brighton, United Kingdom, 2019; pp. 1174–1178. doi:10.1109/ICASSP.2019.8683238.
126. Kramida, G.; Aloimonos, Y.; Parameshwara, C.M.; Fermuller, C.; Francis, N.A.; Kanold, P. Automated Mouse Behavior Recognition Using VGG Features and LSTM Networks. Visual Observation and Analysis of Vertebrate And Insect Behavior; , 2016; pp. 1–3.
127. Zong, M.; Wang, R.; Chen, X.; Chen, Z.; Gong, Y. Motion Saliency Based Multi-Stream Multiplier ResNets for Action Recognition. *Image and Vision Computing* **2021**, *107*, 104108. doi:10.1016/j.imavis.2021.104108.
128. Feichtenhofer, C.; Pinz, A.; Wildes, R.P. Spatiotemporal Multiplier Networks for Video Action Recognition. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Honolulu, HI, 2017; pp. 7445–7454. doi:10.1109/CVPR.2017.787.
129. Zhang, H.; Liu, D.; Xiong, Z. Two-Stream Action Recognition-Oriented Video Super-Resolution. 2019 IEEE/CVF International Conference on Computer Vision (ICCV); IEEE: Seoul, Korea (South), 2019; pp. 8798–8807. doi:10.1109/ICCV.2019.00889.
130. Majd, M.; Safabakhsh, R. Correlational Convolutional LSTM for Human Action Recognition. *Neurocomputing* **2020**, *396*, 224–229. doi:10.1016/j.neucom.2018.10.095.
131. He, J.Y.; Wu, X.; Cheng, Z.Q.; Yuan, Z.; Jiang, Y.G. DB-LSTM: Densely-connected Bi-directional LSTM for Human Action Recognition. *Neurocomputing* **2021**, *444*, 319–331. doi:10.1016/j.neucom.2020.05.118.
132. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning Spatiotemporal Features with 3D Convolutional Networks. 2015 IEEE International Conference on Computer Vision (ICCV); IEEE: Santiago, Chile, 2015; pp. 4489–4497. doi:10.1109/ICCV.2015.510.
133. Fayyaz, M.; Bahrami, E.; Diba, A.; Noroozi, M.; Adeli, E.; Van Gool, L.; Gall, J. 3D CNNs with Adaptive Temporal Feature Resolutions. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Nashville, TN, USA, 2021; pp. 4729–4738. doi:10.1109/CVPR46437.2021.00470.
134. Li, S.; Li, W.; Cook, C.; Gao, Y. Deep Independently Recurrent Neural Network (IndRNN), 2020, [arxiv:cs/1910.06251].
135. Zhang, P.; Lan, C.; Xing, J.; Zeng, W.; Xue, J.; Zheng, N. View Adaptive Neural Networks for High Performance Skeleton-Based Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2019**, *41*, 1963–1978. doi:10.1109/TPAMI.2019.2896631.
136. Song, Y.F.; Zhang, Z.; Shan, C.; Wang, L. Constructing Stronger and Faster Baselines for Skeleton-Based Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2023**, *45*, 1474–1488. doi:10.1109/TPAMI.2022.3157033.
137. Zang, T.; Zhu, Y.; Zhu, J.; Xu, Y.; Liu, H. MPAN: Multi-parallel Attention Network for Session-Based Recommendation. *Neurocomputing* **2022**, *471*, 230–241. doi:10.1016/j.neucom.2021.11.030.
138. Guo, S.; Lin, Y.; Wan, H.; Li, X.; Cong, G. Learning Dynamics and Heterogeneity of Spatial-Temporal Graph Data for Traffic Forecasting. *IEEE Transactions on Knowledge and Data Engineering* **2022**, *34*, 5415–5428. doi:10.1109/TKDE.2021.3056502.
139. Zhang, H.; Ma, C.; Yu, D.; Guan, L.; Wang, D.; Hu, Z.; Liu, X. MTSCANet: Multi Temporal Resolution Temporal Semantic Context Aggregation Network. *IET Computer Vision* **2023**, *17*, 366–378. doi:10.1049/cvi2.12163.
140. Wu, C.; Yin, S.; Qi, W.; Wang, X.; Tang, Z.; Duan, N. Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models, 2023, [arxiv:cs/2303.04671].
141. Fernández, D.G.; Barrio, A.A.D.; Juan, G.B.; García, C.; Prieto, M.; Hermida, R. Complexity Reduction in the HEVC/H265 Standard Based on Smooth Region Classification. *Digital Signal Processing* **2018**, *73*, 24–39. doi:10.1016/j.dsp.2017.11.001.
142. Marathe, A.P. Towards Intelligent Database Systems Using Clusters of SQL Transactions. *Knowledge and Information Systems* **2023**, *65*, 2863–2894. doi:10.1007/s10115-023-01850-5.
143. Significant-Gravitas. AutoGPT, 2023.