

# Архитектура программного обеспечения автоматизированной системы анализа поведения ЖИВОТНЫХ

А. М. Синица<sup>1</sup>, Д. И. Каплун<sup>2</sup>

Кафедра автоматизации и процессов управления,  
Санкт-Петербургский электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)

<sup>1</sup>amsinitca@etu.ru, <sup>2</sup>dikaplun@etu.ru

В. К. Ковригин

Кафедра систем автоматизированного проектирования,  
Санкт-Петербургский электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)

kovrigin.v.k@gmail.com

A. Zamansky

Департамент информационных систем, Университет Хайфы

Хайфа, Израиль

annazam@is.haifa.ac.il

**Аннотация.** Одним из наиболее важных инструментов для изучения поведения животных является анализ, основанный на видео с их участием. Несмотря на то, что системы автоматического анализа существуют для многих видов, для высших животных, таких как собаки, эффективных систем нет. В таких случаях, ученые используют ручной анализ. Данная работа направлена на решение этой проблемы путем разработки программного обеспечения, автоматизирующего анализ поведения животных, а именно анализ сна собак. В статье описывается архитектура системы.

**Ключевые слова:** анализ поведения животных; автоматизация анализа; машинное обучение; обработка видео; архитектура

## I. ВВЕДЕНИЕ

Повышение сложности современных исследований приводит к увеличению перерабатываемого объема данных. Современные технологии обработки данных способны удовлетворить возрастающие запросы на обработку данных, включая фото и видео, но подобные исследования становятся междисциплинарными и более сложными, совмещая знания из основной области исследований и области автоматической обработки данных.

Наблюдение за животными является основным методом анализа поведения животных в арсенале ученых. Этот метод основан на длительной записи видео, которые обычно обрабатываются вручную. Подобный подход требует много времени от исследователей и может привести к ошибкам. Поэтому для ускорения обработки видео используются различные инструменты автоматизации. Существует большое количество систем

для анализа поведения животных. Их можно разделить на три категории.

Первая категория — инструменты кодирования событий. При работе с данными программами оператору необходимо просматривать видео и отмечать происходящие события: начало и конец поведенческого акта (поиск еды, сон и т.д.). Такие программы как Observer TX [1], BORIS [2], CowLog [3] попадают в эту категорию. Но подобные программы не распознают состояние животных, возлагая эту задачу на оператора.

Вторая категория — инструменты для количественных измерений. Данные программы фокусируются на автоматическом определении количественных переменных (координат, направления, расстояния, углов, длительности, скорости).

Такие программы как JWatcher и Kinovea попадают в эту категорию. Они так же не распознают состояние животных.

Третья категория — системы автоматизированного анализа поведения. Они автоматически определяют поведение животных. Существуют системы анализа для диких животных [4], свиней [5], домашней птицы [6], насекомых [7] и грызунов [8].

Программы способные распознавать поведение собак (BARK) единичны и являются академическими экспериментами [9], они не доступны большинству пользователей. Также в этих системах используются дорогостоящие и сложные в эксплуатации 3D камеры.

Ранее было объявлено о крупном проекте, посвященном разработке общей платформы для автоматического анализа поведения [10]. Одним из его этапов является изучение анализа режимов сна как показателя благосостояния животных [11]. В этом проекте

разрабатывался программный комплекс для анализа режимов сна с помощью машинного обучения.

В данной статье описывается архитектура системы, которая самостоятельно определяет состояние собак (сон или бодрствование) на основе видеофайлов с дешевых камер наблюдения, установленных в приютах для собак.

Разработанное программное обеспечение предназначено для оценки различных показателей поведения на основе видеозаписей. Поскольку список индикаторов не известен, а количество видео может быть большим, система должна быть масштабируемой за счет увеличения вычислительной мощности и должна обладать гибким алгоритмом анализа. Для этого была использована клиент-серверная архитектура на основе плагинов.

II. ОПИСАНИЕ СИСТЕМЫ

Как было сказано ранее, система имеет клиент-серверную архитектуру (рис. 1). В связи с необходимостью масштабирования процедур с высокой нагрузкой, таких как работа нейронной сети, они реализованы в выделенных процессах, которые могут быть связаны между собой через TCP / IP для возможности построения системы распределенных вычислений. Такая обработка изображений обычно схожа для различных задач, включает в себя обнаружение объектов (локализацию), классификацию или сегментацию и может быть легко использована повторно для других задач без изменения модели обработки.

Классы программы разработаны в соответствии с архитектурой, описанной ниже.

Клиентская часть разбита на четыре логических модуля:

- GUI — модуль графического интерфейса;
- PRE — модуль предварительной обработки изображения;
- C.PROCESS — модуль ответственный за взаимодействие с сервером (соединение, отправка изображений, прием данных);
- POST — модуль постобработки изображений и анализа полученных данных классификации.

Клиентская часть основана на плагинах, каждый из которых унаследован от базового плагина и реализует эти логические модули как виртуальные методы, исключая модуль взаимодействия, который реализуется в одном из базовых модулей, в качестве основной части обработки кадра. Такой подход позволяет нам быстро разрабатывать алгоритмы обработки видео как новые или унаследованные плагины.

Серверная часть разбита на два логических модуля:

- S.PROCESS — модуль ответственный за взаимодействие с клиентом (соединение, прием изображений, отправка данных);
- NN — модуль обработки изображений.

Модуль обработки изображений представляет собой модель нейронной сети на основе библиотеки TensorFlow.

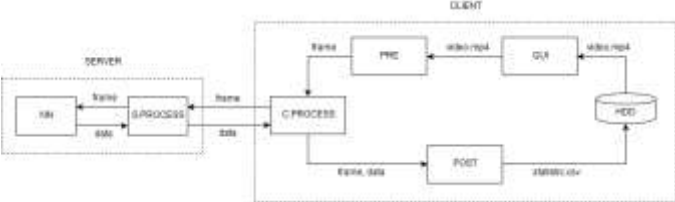


Рис. 1. Архитектура системы

A. Модуль графического интерфейса (GUI)

Модуль графического интерфейса (GUI) отвечает за интерфейс программы и за обработку действий пользователя. На рис. 2 представлен графический интерфейс программы.

Основные GUI элементы:

- меню с кнопками: «Open file», «Exit», «Save statistic», «Go to unnamed objects», и пользовательское меню из плагина;
- видео виджет для отображения кадров, кнопки «Back», «Play», «Next» и слайдер;
- виджет статистики для каждого кадра;
- виджет суммарной статистики.



Рис. 2. Графический интерфейс программы

ТАБЛИЦА I Действия кнопок в СИСТЕМЕ

Кнопка	Расположение	Действие
Open file	Внутри кнопки "File" в меню	Открытие файла
Exit	Внутри кнопки "File" в меню	Выход из программы
Save statistic	Меню	Сохранение статистики
Go to unnamed object	Меню	Переход к не именованным объектам
Back	Видео виджет	Переход на один кадр назад
Play	Видео виджет	Старт / остановка отображения кадров
Next	Видео виджет	Переход на один кадр вперед

Все типы виджетов, кроме виджета для видеоплеера, могут быть реализованы в плагинах или унаследованы от одного из существующих плагинов

#### *B. Модуль предварительной обработки кадров (PRE)*

Основной задачей модуля предварительной обработки изображений (PRE) является предварительный анализ кадров.

Перед отправкой на сервер кадр сравнивается с предыдущим. Если объекты в кадре не совершали никаких движений, кадр не отправляется. Благодаря этому одинаковые и испорченные кадры не проходят через нейронную сеть, значительно увеличивая производительность программы.

Другая задача, которая может быть решена на этом этапе, – это предварительная обработка изображений, например, фильтрация или обрезка.

#### *C. Модуль взаимодействия с сервером (C.PROCESS)*

Модуль C.PROCESS отвечает за взаимодействие с сервером: устанавливает соединение, отправляет кадры и принимает данные, полученные в результате обработки сервера. Данные между клиентом и сервером передаются с помощью протокола TCP/IP.

Протокол данных, основанный на json, содержит аннотацию изображения и двоичные данные, которые содержат кадры.

#### *D. Модуль постобработки изображений (POST)*

Модуль постобработки изображений (POST) состоит из последовательных фильтров, обрабатывающих следующие ошибки нейронной сети:

- двойное обнаружение – обнаружение одной собаки как спящей и как бодрствующей,
- неправильный порядок индексов собак,
- высокочастотный шум координат ограничительной рамки,
- одиночная ошибка классификации.

На рис. 3 представлена последовательность фильтров в модуле постобработки.

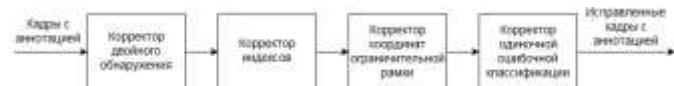


Рис. 3. Последовательность фильтров постобработки

Входные данные для модуля постобработки: изображение и аннотация для каждой собаки, полученная от нейронной сети.

Корректор двойного обнаружения вычисляет евклидово расстояние между центрами ограничительных рамок собак и сравнивает его. Если расстояние совпадает, значит одна собака была определена и как спящая и как бодрствующая. Далее сравнивается вероятность

определения двух разных состояний и выбирается состояние с большей вероятностью.

Корректор индексов исправляет неправильно классифицированный индекс собак с помощью евклидова расстояния между центрами ограничительных рамок собак. Если индекс собаки изменился, ему присваивается ближайший индекс с прошлого кадра.

Корректор координат ограничительной рамки сглаживает высокочастотные дрожания координат ограничительной рамки, предоставляя пользователю более комфортный режим наблюдения.

Корректор ошибочной классификации исправляет одиночные ошибки, которые являются одной из фундаментальных особенностей нейронных сетей. Для исправления ошибок применяются два подхода. Первый – анализ движения. Если собака двигается, то вероятнее всего, она бодрствует. И если текущее состояние – сон, то его следует исправить. Второй подход основан на предположении, что частота чередования между состояниями сна и бодрствования у животных не может быть слишком высокой. Следовательно, если встречаются короткие последовательности состояний (меньше 4 кадров), их также нужно исправить.

После того как кадр прошел через последовательность фильтров, по нему формируется и сохраняется статистика.

#### *E. Модуль взаимодействия с клиентом (S.PROCESS)*

Модуль S.PROCESS отвечает за взаимодействие с клиентом: устанавливает соединение, принимает кадры и отправляет данные, полученные в результате обработки сервера.

#### *F. Image Processing Module (NN)*

На серверной части кадры обрабатываются сверточной нейронной сетью, которая определяет положение собак в кадре и их состояние (спит или бодрствует). Для определения объектов используется TensorFlow Object Detection API [12]. Изначально, для лучшей производительности использовалась модель `ssd_mobilenet_v1` [13], но она не смогла предоставить необходимую точность, вследствие этого была выбрана `faster_rcnn_resnet101` [14].

Входом в нейронную сеть является изображение и его индекс. Выходом – аннотация с индексом собаки, координатами углов прямоугольника, описывающего положение собаки в кадре, состояние собаки (спит или бодрствует) и вероятность нахождения собаки в этом состоянии.

### III. ЗАКЛЮЧЕНИЕ

Несмотря на то, что собаки являются хорошо изученным видом в науке о животных, очень немногие работы посвящены проблеме автоматического анализа поведения собак и других животных в целом. В этой статье мы представили архитектуру системы для автоматического анализа поведения собак, которая, в настоящее время,

используется для измерения показателей благосостояния в текущем исследовательском проекте.

Данная архитектура была выбрана для того, чтобы разделить модуль с высокой вычислительной нагрузкой (серверная часть) и модуль с низкой вычислительной нагрузкой (клиентская часть). Это дает возможность запускать серверную часть на устройствах с высокой производительностью, а клиентскую часть на устройствах с низкой производительностью, что в свою очередь экономит ресурсы пользователей.

Внутренняя архитектура клиента и сервера позволяет изменять отдельные модули без изменения остальной системы. Подобный подход делает систему гибкой и удобной для дальнейшего развития.

Одними из будущих направлений исследований являются исследования более сложных подходов, в которых можно моделировать зависимости во времени (рекуррентные системы, вероятностные модели), внедрять многосерверные архитектуры для увеличения скорости обработки, включать асинхронную обработку видео с сохранением в кэш.

В ближайшем будущем планируется расширить опыт использования системы в других проектах по анализу поведения животных.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Noldus L.P.J.J. The Observer: A software system for collection and analysis of observational data // *Behavior Research Methods, Instruments, & Computers*. 1991. Т. 23. № 3. С. 415–429.
- [2] Friard O., Gamba M. BORIS: a free, versatile open-source event-logging software for video/audio coding and live observations // *Methods in Ecology and Evolution*. 2016. Т. 7. № 11. С. 1325–1330.
- [3] Hänninen L., Pastell M. CowLog: Open-source software for coding behaviors from digital video // *Behavior Research Methods*. 2009. Т. 41. № 2. С. 472–476.
- [4] Burghardt T., Čalić J. Analysing animal behaviour in wildlife videos using face detection and tracking // *IEEE Proceedings - Vision, Image, and Signal Processing*. 2006. Т. 153. № 3. С. 305.
- [5] Ahrendt P., Gregersen T., Karstoft H. Development of a real-time computer vision system for tracking loose-housed pigs // *Computers and Electronics in Agriculture*. 2011. Т. 76. № 2. С. 169–174.
- [6] Sergeant D., Boyle R., Forbes M. Computer visual tracking of poultry // *Computers and Electronics in Agriculture*. 1998. Т. 21. № 1. С. 1–18.
- [7] Noldus L.P.J., Spink A.J., Tegelenbosch R.A. Computerised video tracking, movement analysis and behaviour recognition in insects // *Computers and Electronics in Agriculture*. 2002. Т. 35. № 2–3. С. 201–227.
- [8] H.A Van de Weerd, R.J.A Bulthuis, A.F Bergman, F Schlingmann, J Tolboom, P.L.P Van Loo, R Remie, V Baumans, L.F.M Van Zutphen. Validation of a new system for the automatic registration of behaviour in mice and rats // *Behavioural Processes*. 2001. Т. 53. № 1–2. С. 11–20.
- [9] Barnard S, Calderara S, Pistocchi S, Cucchiara R, Podaliri-Vulpiani M., Messori S., Ferri N.. Quick, Accurate, Smart: 3D Computer Vision Technology Helps Assessing Confined Animals' Behaviour // *PLOS ONE*. 2016. Т. 11. № 7. С. e0158748.
- [10] Kaplun D., Sinitca A., Zamansky A., Blauer-Elsner S., Plazner M., Fux A., & van der Linden, D. Animal Health Informatics: Towards a Generic Framework for Automatic Behavior Analysis Position Paper // *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies. SCITEPRESS - Science and Technology Publications*, 2019.
- [11] Kaplun D., Sinitca A., Zamansky A., Schork I., Young R., de Azevedo, C. (2019) Time budget of sleep analysis as welfare indicator using machine learning. *European Journal of Clinical Investigation*, 49(S1), 101–221.
- [12] Huang V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors // *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR): IEEE*, 2017.
- [13] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neuralnetworks for mobile vision applications // *arXiv e-prints*, 2017
- [14] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition // *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.