INTERNATIONAL DESIGN CONFERENCE – DESIGN 2024
https://doi.org/10.1017/pds.2024.224

Automatic movement pattern analysis for data-driven system optimisation – an example for fattening livestock farming monitoring system

Gurubaran Raveendran ✉, Sören Meyer zu Westerhausen, Johanna Wurst and Roland Lachmayer

Leibniz University Hannover, Germany

✉ Raveendran@ipeg.uni-hannover.de

Abstract

This paper introduces a method for analysing motion patterns that can be utilised to optimise data-driven systems. The aim is to use surveillance cameras and artificial intelligence to track multiple objects in a reliable manner, thereby preserving the authenticity of movement patterns for numerous and similar objects. In a case study, this method is applied to optimize lighting conditions in animal husbandry. Furthermore, this approach can be utilized not only in animal husbandry but also in other domains.

Keywords: artificial intelligence (AI), big data analysis, design methodology, design analysis

1. Introduction

In the context of engineering design, the application of sensor systems for continuous monitoring has gained importance in various applications (Lachmayer et al., 2014). This development is related to a current trend towards sustainable and welfare-oriented animal husbandry observed in recent years (Ohashi et al., 2023). The use of technology to individually monitor and track animals in livestock farming has the potential to significantly improve animal welfare, promote their health and increase overall profitability. However, methods based on Radio-Frequency Identification (RFID) and Global Positioning System (GPS) involve physically mounting devices on the animals, which can alter their behaviour and may not be feasible for certain farm animal species (Guzhva et al., 2018; Li et al., 2020; Neethirajan, 2022). In contrast, video-based tracking systems present a promising solution by eschewing physical attachments (Guzhva et al., 2018). This technological innovation holds significant promise for animal husbandry, providing comprehensive insights into animal behaviour, facilitating early identification of injuries or health issues, and detecting deviations from normative behavioural patterns (Bartels et al., 2020). Through continuous monitoring and analysis of animal behaviour, livestock managers can make informed decisions to optimize animal welfare and refine environmental conditions within the barns (Guzhva et al., 2018; Schmidt et al., 2022).

The research project 'Optimization of Light Management in Husbandry of Fattening Turkeys (acronym: OptiLiMa)' also focuses on improving animal welfare, especially in poultry rearing by preventing harmful feather pecking, as emphasised by a central source of the project (Raveendran et al., 2023). The project addresses the synthesis of design theory and advanced technology to real-time analyse and classify movement patterns in over 6,000 animals, emphasising the use of design methods and tools. The integration of Artificial Intelligence (AI) into the animal housing lighting management system enables an automated optimisation process for key environmental factors based on real-time analysis of animal behaviour. This data-driven design solution automatically adjusts the lighting

https://doi.org/10.1017/pds.2024.224 Published online by Cambridge University Press

system to promote animal activity and support their well-being, increasing efficiency and effectiveness in animal husbandry without the physical presence of animal keepers. Section 2 clarifies whether existing modern technologies can already solve the problem or whether a modified method is required for the development of a video-based AI animal monitoring system. Section 3 then proposes a specific design method that can be used in the development of such systems. Section 4 uses a use case to demonstrate the development of an observation system in a fattening turkey house, including an analysis framework for optimising the lighting system. Finally, Section 5 summarises the findings and creates a design guideline that can serve as a guide for optimising other livestock systems.

2. State of the art

With increasing digitalisation in animal husbandry, schematic tasks and problems in herd management can be overcome. The individual observation and tracking of animals using technology in the barn is an important step towards precision livestock farming. With such technology, injured or sick animals can be identified and deviations from normal behaviour in the barn can be detected. This has a positive effect on animal welfare, improves animal health and increases profitability.

Existing approaches are based on RFID, Wi-Fi, Bluetooth, GPS, or Ultra-Wideband, but they all have the disadvantage that they require a tracking module to be attached to the animal. Such an intervention may affect the behaviour of the animal and the herd (Li et al., 2020).

The large amount of work of these technologies and impracticality for marking of poultry due to anatomical and ethological characteristics make them unfeasible for poultry farms, as well as the high acquisition costs for thousands of birds kept in commercial farms (Neethirajan, 2022). The implementation of video-based tracking systems for the real-time analysis of large animal populations in agriculture, especially for the monitoring of fattening turkeys, is a complex challenge that goes far beyond the capacities of conventional industrial systems. In contrast to the predictable movements of machines, the monitoring of animals requires the interpretation of diverse, unpredictable and complex behaviours. This requires the development of advanced algorithms and a robust data processing infrastructure capable of handling the large amounts of data generated by tracking 5,000 to 16,000 animals. In this context, analysing crowds using machine learning and AI, especially by using Convolutional Neural Networks (CNNs) for real-time tracking and counting as well as anomaly identification, offers a valuable starting point (Zhan et al., 2008). Techniques originally developed for monitoring human movement encounter specific challenges when applied in a turkey barn, compounded by the unique dynamics, rapid growth phases and homogeneity in shape and size of the turkeys. While humans are relatively easy to distinguish by individual characteristics such as clothing, size or posture, the uniformity of the animals makes quick and precise identification difficult. All turkeys look similar, which complicates the application of algorithms developed for human movement patterns. The rapid movements and drastic physical changes of the animals from around 500 g to up to 21 kg in just 140 days require the algorithms to be highly adaptable (Raveendran et al., 2023). Therefore, specially adapted models and analysis methods are required that not only consider the dynamic movements and rapid growth, but can also master the challenge of the similar appearance of the animals in the barn. The successful implementation of such systems requires innovative approaches that are specifically tailored to the needs and characteristics of animal monitoring. Despite existing challenges, the application of advanced animal monitoring technologies offers significant potential, which is discussed in detail in Section 3.

3. Methodology for designing system for movement pattern analysis

A flexible design methodology is proposed to create an adaptive design framework for the development of AI-based monitoring and optimisation systems that have applications beyond the specific use in livestock farming. The proposed methodology consists of the following three steps:

1. ***Hardware configuration:*** The process begins with the selection and provision of sensors or cameras that meet the specific requirements of the monitoring area. This is followed by setting

https://doi.org/10.1017/pds.2024.224 Published online by Cambridge University Press

up a robust hardware infrastructure that enables real-time data processing and analysis, with a particular focus on a powerful graphics card for AI computing power.

2. ***Software architecture:*** Developing an integrated monitoring system based on AI, the goal is to collect and analyse data in real time, facilitating optimizations in specific application areas. The goal and purpose of AI is to help, and facilitate tasks for humans, or even solve tasks that a human or programmed algorithm may not be able to. Deep Learning is a subfield of Machine Learning, which in turn is a subfield of AI (Campesato, 2020). The classic image processing based on a predetermined sequence of algorithms applied to an image to recognize specific features or objects in that image has become outdated by the Deep Learning-Object Recognition approach. These algorithms can be developed based on rules and heuristics and usually require extensive knowledge of underlying mathematics and signal processing. In contrast, Deep Learning-Object Recognition uses a form of machine learning where neural networks are used to automatically extract and recognize features in an image (Don Yitong Ma, 2022). Neural networks can process a large amount of data and recognize patterns and information in the data. One of the most commonly used network architectures is the CNN, which is often used in object recognition in images, resulting in more efficient and accurate recognition (Zhang et al., 2019). In case of dynamically fast-growing properties, a CNN model is used for each growth phase, switching to the next higher CNN model at regular intervals. With the help of such a model trained on large datasets, animal detection can take place. In a confined poultry environment, where animals are closely packed and look similar, linking data points across multiple frames poses a significant challenge. To overcome this challenge, precise methods are employed. Methods for linking data points across multiple frames include optical flow, feature tracking, and the nearest-neighbour algorithm. Optical flow enables precise tracking of movements within image sequences by analysing pixel movements between successive frames. This method offers the advantage of continuous motion tracking and can be used to identify and track animals in the image. However, it is sensitive to scene changes and requires precise feature detection, which may limit the detection of dynamic animal movements (DeCarlo and Metaxas, 2000). Feature tracking is a method for tracking specific features, such as characteristic points or patterns, across multiple frames. Compared to optical flow, it is more robust to scene changes and is particularly suitable for precise tracking of individual animals. However, it is more susceptible to errors in densely packed environments and may have difficulty tracking dynamic animal movements (Bretzner and Lindeberg, 1998). To further improve animal tracking in image sequences, the nearest-neighbour algorithm is applied. This algorithm enables efficient and consistent linking of animals from one frame to the next by calculating the minimum distance between the animals' features. The nearest-neighbour algorithm offers the advantage of improved accuracy and reliability in linking data points across multiple frames. It enables precise identification and tracking of animals, even in challenging environments with similar-looking animals. By applying the nearest-neighbour algorithm, the quality of data linkage is further improved, and animal identification is optimized (Dick et al., 2013). In summary, linking data points across multiple frames in confined poultry environments is a complex task. However, using methods such as optical flow, feature tracking, and particularly the nearest-neighbour algorithm enables precise identification and tracking of animals. The nearest-neighbour algorithm stands out as an efficient, consistent, and accurate method for linking data points across multiple frames.

3. ***Database management and thread scheduling algorithms:*** Playing a vital role in various domains, the storage and management of substantial volumes of structured or unstructured data provide a structured framework for efficient organization and retrieval, simplifying data management and facilitating access to information (Üreten et al., 2019). As a result, databases have become indispensable in various applications, playing a central role in modern software development and supporting fundamental business processes (Hagen et al., 2021). Specifically designed for secure and efficient processing of large data volumes, databases become particularly important when dealing with real-time data streams from a variety of sources.

https://doi.org/10.1017/pds.2024.224 Published online by Cambridge University Press

They facilitate rapid data retrieval for analysing animal behaviour and enable seamless data sharing among different applications or systems, promoting integration and collaboration in research initiatives (Lachmayer and Mozgova, 2022). The choice between a relational database like MySQL and a NoSQL database like MongoDB depends on specific requirements and the nature of the application. Relational databases are suitable for processing structured data and complex relationships, while NoSQL databases support more flexible data models. MySQL, with its B-tree indexing, is often used for optimized query performance, utilizing B-trees for efficient access to extensive datasets (Kieseberg et al., 2019). On the other hand, if the focus is on unstructured data and flexibility in complex queries, a NoSQL database like MongoDB may prove to be better suited. Efficient processing of large volumes of data in data queries requires scalable systems to overcome time-related challenges. The use of thread scheduling algorithms offers a promising approach to optimize the performance of data queries and ensure system scalability. These algorithms enable the efficient utilisation of processor resources by determining which threads should be executed based on various factors such as thread priority and processor availability. In the state of the art, various approaches to thread scheduling have been proposed. A hardware-based thread-list scheduler describes the assignment of threads to processor contexts based on their readiness and processor idle time. This approach aims to optimize thread execution and efficiently utilize processor resources (Giorgi and Scionti, 2015). The use of dynamic multithread languages like Cilk is discussed as a means to simplify task parallelization and improve application performance. These languages provide low-overhead fork-join primitives and efficient runtime systems for parallel execution (Dimakopoulos, 2014). A thread-based software synthesis technique is proposed to reduce communication overhead in a system. This technique employs a mixed static-dynamic thread scheduler to optimize thread communication and enhance the efficiency of parallel execution (Shin and Choi, 1996). These articles provide valuable insights into the design and implementation of thread scheduling algorithms for efficient parallel execution. They present different approaches and techniques to maximise the performance of multithreaded applications and optimize resource utilisation.

The implementation of AI-supported systems takes place in five core phases, starting with the preparation phase, in which a needs analysis is carried out to identify the specific requirements and select suitable technologies. In the installation phase, the necessary hardware and software infrastructure is set up, including the nearest-neighbour algorithm and the configuration of databases and thread scheduling. The operation starts with data acquisition and real-time analysis, using the nearest neighbour algorithm for precise object tracking and thread scheduling for performance optimisation. In the Optimisation phase, the system is fine-tuned based on feedback and insights gained. The final phase of maintenance and evaluation is used to regularly review and adjust the system to ensure its long-term efficiency and effectiveness. This compact approach enables the flexible and effective implementation of AI systems in various application areas.

4. Case study: AI in livestock environmental optimization

The method was used for an animal monitoring system with minimal influence on animal behaviour. Strategic camera arrangements and efficient AI algorithms enable seamless integration into operational processes and take ethical aspects into account. Deep learning enables the detection of up to 6,000 animals without tagging. The Section covers advanced methods for livestock monitoring and management, focussing on key areas:

- Subsection 4.1 discusses specialised algorithms for the growth of fattening turkeys over 140 days.
- Subsection 4.2 describes an approach that combines parallel processing and data retrieval to accurately track animal movements.
- Subsection 4.3 introduces an algorithm to classify movement patterns and contributes to system optimisation through an activity index.

https://doi.org/10.1017/pds.2024.224 Published online by Cambridge University Press

4.1. Design for data collection with video-based detection system

The goal is a real-time surveillance system for a 120 x 20 metre barn to monitor over 6,000 animals, achieved through twelve IP cameras with 70% coverage. Network access enables local and remote monitoring, while specialised workstations with acceleration units efficiently process the video data (Figure 1). To ensure efficient parallel processing of the video streams, each camera uses a dedicated graphics card so as not to share the computing power (Navarro et al., 2021). With twelve cameras, twelve individual graphics cards are therefore required. This leads to the need for three workstations, as there is room for a maximum of four graphics cards in one housing and this is also in line with price-performance considerations. The workstations mentioned use solid-state drives (SSDs) for operating systems and processing and hard disc drives (HDDs) for data collection.

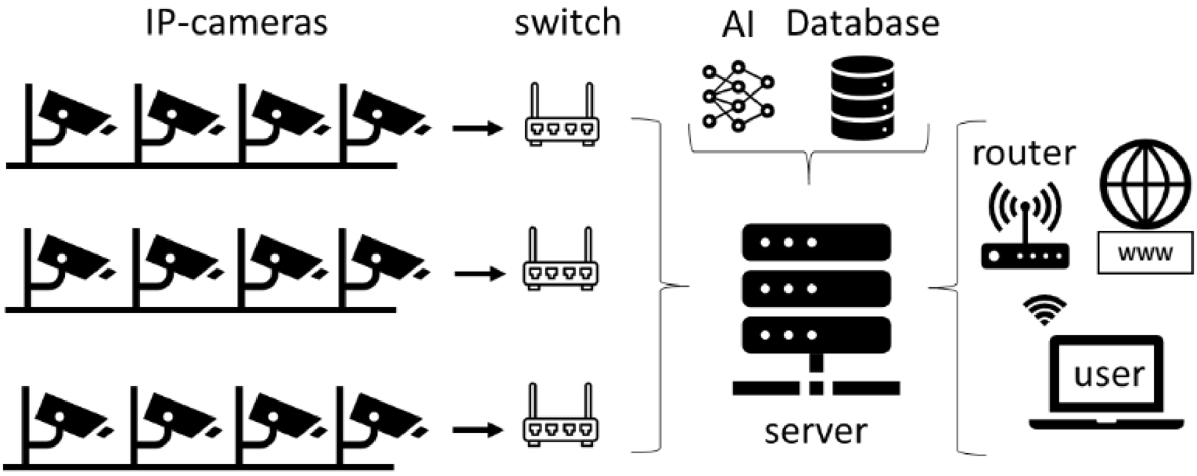

Figure 1. Automatic animal observation system with 12 IP-cameras

Each workstation is equipped with at least 96 GB of memory to meet the memory-intensive requirements of the Python software environment. The choice of the Ubuntu server operating system is based on criteria such as cost-effectiveness, a robust security infrastructure, regular updates and scalability to meet the needs of the users (van Vugt, 2008).The software architecture (Figure 2) is the key element of this system. Through advanced algorithms for object detection and data storage, it enables a holistic capture of animal behaviour, as discussed in the next chapter. The findings derived from Section 3 serve as the basis for the design of a scientifically sound software solution. On the hardware level, twelve cameras are employed to monitor the barn and turkeys. These cameras utilize the Real-Time Streaming Protocol (RTSP) for real-time transmission of video streams. The video streams are divided into individual frames to enable a detailed analysis. On the software level, a checksum calculation is initially performed for each individual frame to ensure data integrity. Frames that do not match the calculated checksums are identified as defective and can consequently be removed. Furthermore, the video content undergoes dynamic evaluation through a real-time decision function, which determines whether storage in the mp4 format is warranted. The operator's interaction with the control console serves as a decision criterion in this process. For animal recognition, a CNN model is employed, based on a 50-layered Deep Residual Network (ResNet). This CNN model is complemented by a Single Shot MultiBox Detector, facilitating efficient object recognition without the need for a separate object proposal. To enhance the quality of the detection algorithms, the system can allow manual annotations of the objects to be recognized, thereby adding training data. Furthermore, an automatic adjustment of the CNN model occurs after twelve weeks to maintain high detection accuracy. Factors such as the exponential body growth of turkeys and the age of the animals, determined through a clock function, are considered during this adjustment. The training of the CNN models is facilitated by a web-based platform developed by Wolution, providing easy handling and integration of training data. The captured positional data is stored in a MySQL database, offering a stable and reliable solution for data storage. All twelve cameras access a common database and structure. To efficiently process large datasets in a MySQL database for Big Data analytics, various measures are implemented to

https://doi.org/10.1017/pds.2024.224 Published online by Cambridge University Press

optimise data processing. These include the creation of indexes, table partitioning, and query optimisation. A specific index structure known as B-Tree indexing is utilized to expedite access to vast datasets. This index structure consists of pointers that reference specific rows or columns in the database. Similar to a keyword index in a book, B-Tree indexing enables rapid searching for values without the need to scan the entire table. For the described system, the "Camera" and "timestamp_server" columns are chosen as index columns. The "objectCount" column typically contains between 200 and 500 objects per row, resulting in the "objectBoxes" array containing between 1,000 and 2,500 entries. During the fattening phase, the database can reach a size of 22 TB. To ensure efficient processing of Big Data, methods for data reduction and intelligent download options are considered to adjust the size of downloads and the included rows. This approach aims to optimise the performance and access times of the MySQL database. By employing B-Tree indexes, optimizing queries, and incorporating data reduction methods, the MySQL database can efficiently handle Big Data. This facilitates effective Big Data analysis and customisation of downloads to specific requirements. To maintain a continuous 24/7 connection to the database, the configuration file includes parameters for wait time, connection timeout, and hardware monitoring.

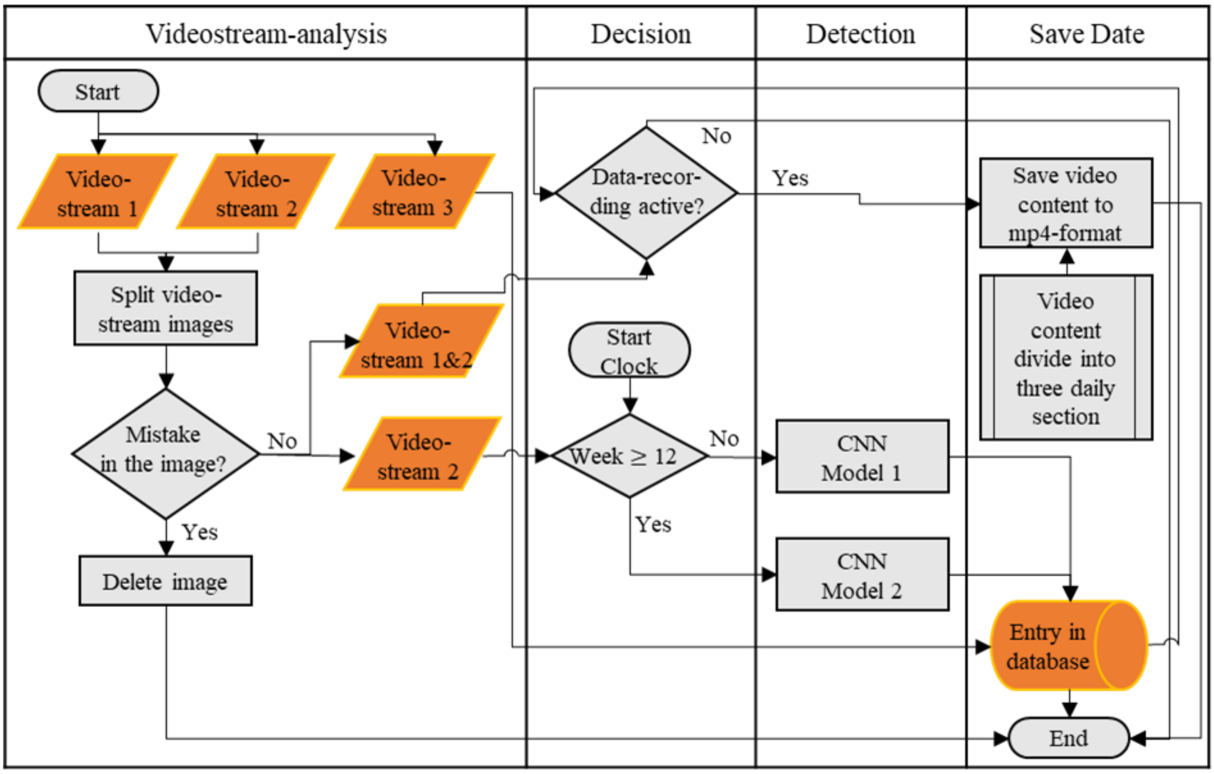

Figure 2. Flowchart of the software architecture for detection the animals

4.2. Data preparation for movement pattern analysis

The present conceptualization addresses the efficient execution of data queries employing parallel processing and thread scheduling algorithms. Research requirements encompass swift, scalable data queries, processing substantial data volumes within acceptable time frames, ensuring data integrity in case of interruptions, and dynamic adaptability to expand database sizes and user counts. The design methodology indicates that efficient data queries facilitated by these algorithms optimize data query performance and ensure system scalability. Various approaches, including hardware-based thread list scheduling and software-based synthesis techniques, provide valuable insights into the design and implementation of thread scheduling algorithms. The findings encompass enhanced system performance through efficient resource utilization in parallel execution using Thread-Scheduling. Simultaneous processing of multiple queries reduces response times and increases efficiency for users. Additionally,

https://doi.org/10.1017/pds.2024.224 Published online by Cambridge University Press

thread-based parallel processing with scheduling enables scalable SQL data queries to handle growing database sizes without causing bottlenecks. The implemented workflow of the concept for efficient data querying through Thread-Scheduling can be divided into several consecutive steps. Initially, users input specific download parameters such as time range, columns, and login credentials, triggering the initiation of the download process. In this context, a Thread-Scheduling algorithm is employed, determining the start time and priority of threads based on various criteria. These criteria typically include factors such as available resources (e. g. , CPU utilisation, memory), the size of the data query, user priority, and user wait time. The initial thread (Thread0) is created in a separate thread for establishing the database connection, ensuring optimal resource utilization. Concurrently, additional threads (Thread X) are launched based on stability considerations and in accordance with the Thread-Scheduling algorithm to perform background database queries. The writing process occurs immediately after each query, with data chunks being written directly to the corresponding file to ensure data integrity even in the event of interruptions. Dynamic adjustment of chunk sizes also takes place within the framework of Thread-Scheduling. Upon completion of a thread, the program checks for additional data availability and initiates new threads according to the Scheduling algorithm to ensure efficient resource utilisation. The program is only completed after the full download and processing of all chunks according to the thread scheduling algorithm. A dedicated algorithm detects and processes empty files to ensure data integrity and ensure consistent conclusions. Introducing a three-stage method for nearest neighbour analysis, this systematic approach excels in generating trajectories and deriving movement patterns from positional data. Their wide range of applications ranges from animal behaviour analysis to crowd monitoring. By implementing this method, comprehensive analyses of movement patterns can be performed, which in turn can help improve systems based on these movement patterns. In the initial step of this three-stage method, nearest neighbour analysis is used to record fine movement intensities. The position data is extracted frame by frame from the database. Each data point in a frame is analysed for its nearest neighbours using a small search radius that corresponds to the diagonal of the measurement object. The resulting neighbouring points are connected to each other to generate detailed trajectories. Optionally, simple movement patterns such as speed or acceleration can be extracted for each point. In the second stage, the search radius is expanded to twice the diagonal of the object in order to record medium intensities of movement. The nearest neighbour analysis is carried out again for this extended search radius, connecting the identified neighbours to form further trajectories. In the third stage, the search radius is expanded again to three times the diagonal of the object to adequately account for large movements. The nearest neighbour analysis is repeated for this expanded search radius, and the identified neighbours are linked into comprehensive trajectories. Finally, the visual analysis enables a detailed review and interpretation of the results of this method for motion pattern recognition based on the nearest neighbour analysis.

4.3. Framework analysis for movement pattern

In the context of movement analysis, the division of movement patterns into different classes becomes crucial as it forms the basis for a meaningful analysis. In this context, three categories for movement patterns are established: "resting - blue", "active - green" and "very active - red". An algorithm is developed to classify the trajectories into these predefined classes, using the prediction error of the linear best-fit line as the main feature for the categorization. The speed along the straight line is considered to differentiate between uniform movements and those with changes in direction. To validate this approach, fitting algorithms are applied to the motion trajectories to determine the degree of the polynomial. This specifies that movements above the third degree are classified as “very active”. The results of these two methods are compared, and if there are discrepancies, the affected trajectories are classified as “active” instead of “very active”. To ensure that the classification meets expected standards, a thorough visual analysis of the results is performed. Two scenarios are examined: one illustrating a lower intensity of movement in the objects (Figure 3, right), and another demonstrating a higher intensity of movement in the objects (Figure 3, left). This study is intended to ensure the reliability of the classification method under different conditions and to make possible adjustments.

https://doi.org/10.1017/pds.2024.224 Published online by Cambridge University Press

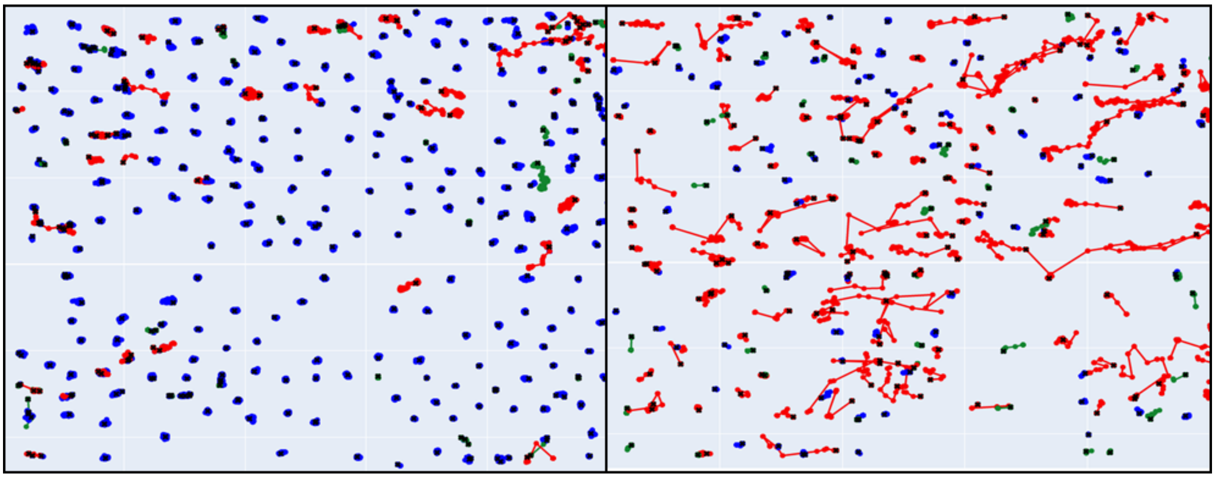

Figure 3. Lower (right) and Higher (left) intensity of movement in the objects

Upon close examination of Figure 3, a clear activity of the objects moving from the right to the left side is observable. It is noticeable that only a few objects remain at rest; in fact, all objects exhibit extremely intense movement dynamics. The increased motion activity, as seen in Figure 3, highlights the system's sensitivity to subtle motion changes. This conclusion implies that the algorithms and classification methods implemented have the ability to capture the subtle nuances in the animals' movement patterns and distinguish between resting and very active states. This result strengthens the validity and performance of the developed analysis framework in the context of movement pattern recognition and classification. The analysis of movement patterns based on different scenarios is a crucial step in developing a deeper understanding of these patterns. A key factor in this process is the activity level, which plays a pivotal role in data-driven system optimization. By applying an activity index ($I_{Aktivität}$), it becomes possible to quantify and evaluate activity, enabling an objective assessment of movements and behaviours.

The use of an activity index brings several benefits, particularly in the context of automated control systems. Precise measurement and evaluation of the activity level not only facilitate informed analysis but also allow for the implementation of targeted optimization strategies. These strategies can be developed based on the captured activity data, ensuring efficient control and adaptation of the systems. Another advantage lies in the ability to detect deviations in movement behaviour at an early stage. Continuous monitoring of the activity index enables the identification of irregularities and the initiation of appropriate measures. This helps address potential issues or disruptions in movement patterns early on, thereby enhancing system reliability. Figure 4 depicts an elevated activity level, averaging at 54.9 %. It is evident that a large number of objects are in motion, indicating an abnormal situation.

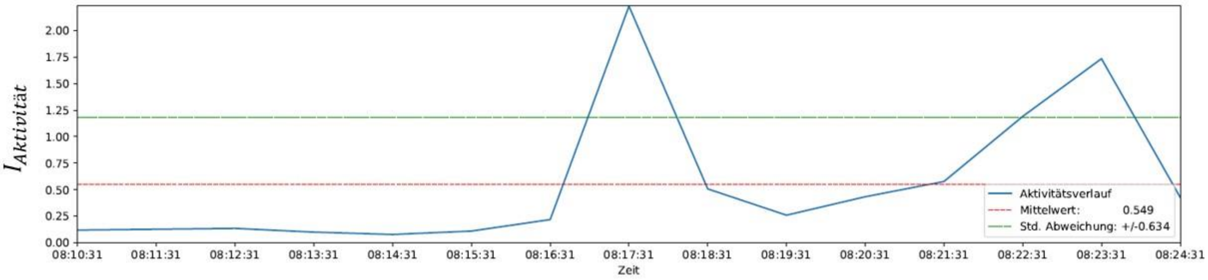

Figure 4. Optimising data-driven systems through activity level analysis

Integrating an activity index into automated control systems also enables dynamic adaptation to changing environmental conditions or user behaviour. This leads to optimized system performance and efficient resource utilisation.

https://doi.org/10.1017/pds.2024.224 Published online by Cambridge University Press

5. Conclusion and outlook

The "OptiLiMa" project is an example of how design research and AI methods can revolutionise animal husbandry by using behavioural analysis of fattening turkeys to provide a data-driven basis for optimising animal husbandry systems. This design method, which has been proven to reliably track of over 6,000 animals using a commercially available surveillance camera without sensors on the animal, also ensures that natural animal behaviour remains unaffected. Especially in animal husbandry, this contactless detection and analysis of movements enables a natural representation of normal movement, which supports targeted improvement of their well-being. The proposed design methodology pushes the boundaries of agricultural practice through design research and methodology, while offering a universal approach to cross-industry challenges. By using movement pattern analysis, "OptiLiMa" enables precise optimisation of operations, such as improving processes in large railway stations through more efficient people flow analysis. This optimisation leads to reduced delays and faster departure times. From these findings, a design guideline for the optimisation of operational parameters can be derived, which supports both specific and general use cases and can therefore increase performance and efficiency in various industries. The design guidelines include progressive approaches:

1. Analysis of the initial situation: Identify needs and challenges, considering the environment and requirements.
2. Integration of technology: Utilise modern sensors and AI for data collection.
3. Real-time data processing: Quickly analyse large amounts of data for immediate adaptation.
4. Customisable systems: Implement smart lighting and climate control based on data.
5. Flexible optimisation: Drive sustainable performance through adaptive optimisation strategies that improve efficiency and adaptability to change.
6. Privacy: Keep collected data secure and private.
7. User friendliness: Intuitive operation and training for staff.

References

Bartels, T., Stuhrmann, R.A., Krause, E.T., Schrader, L., 2020. Research Note: Injurious pecking in fattening turkeys (Meleagris gallopavo f. dom.)—video analyses of triggering factors and behavioral sequences in small flocks of male turkeys. Poultry Science 99, 6326–6331. https://doi.org/10.1016/j.psj.2020.09.016

Bretzner, L., Lindeberg, T., 1998. Feature Tracking with Automatic Selection of Spatial Scales. Computer Vision and Image Understanding 71, 385–392. https://doi.org/10.1006/cviu.1998.0650

Campesato, O., 2020. Artificial Intelligence, Machine Learning, and Deep Learning. Mercury Learning and Information.

DeCarlo, D., Metaxas, D., 2000. Optical Flow Constraints on Deformable Models with Applications to Face Tracking. International Journal of Computer Vision 38, 99–127. https://doi.org/10.1023/A:1008122917811

Dick, T., Perez, C., Jagersand, M., Shademan, A., 2013. Realtime Registration-Based Tracking via Approximate Nearest Neighbour Search. https://doi.org/10.15607/RSS.2013.IX.044

Dimakopoulos, V.V., 2014. Parallel Programming Models, in: Torquati, M., Bertels, K., Karlsson, S., Pacull, F. (Eds.), Smart Multicore Embedded Systems. Springer, New York, NY, pp. 3–20. https://doi.org/10.1007/978-1-4614-8800-2_1

Don Yitong Ma, 2022. Summary of Research on Application of Deep Learning in Image Recognition. Highlights in Science, Engineering and Technology 1, 72–77. https://doi.org/10.54097/hset.v1i.429

Giorgi, R., Scionti, A., 2015. A scalable thread scheduling co-processor based on data-flow principles. Future Generation Computer Systems 53, 100–108. https://doi.org/10.1016/j.future.2014.12.014

Guzhva, O., Ardö, H., Nilsson, M., Herlin, A., Tufvesson, L., 2018. Now You See Me: Convolutional Neural Network Based Tracker for Dairy Cows. Frontiers in Robotics and AI 5.

Hagen, S., Brinker, J., Gembarski, P.C., Lachmayer, R., Thomas, O., 2021. Integration von Smarten Produkten und Dienstleistungen im IoT-Zeitalter – Ein Graph-basierter Entwicklungsansatz, in: Meinhardt, S., Wortmann, F. (Eds.), IoT – Best Practices: Internet der Dinge, Geschäftsmodellinnovationen, IoT-Plattformen, IoT in Fertigung und Logistik, Edition HMD. Springer Fachmedien, Wiesbaden, pp. 245–258. https://doi.org/10.1007/978-3-658-32439-1_14

Kieseberg, P., Schrittwieser, S., Frühwirt, P., Weippl, E., 2019. Analysis of the Internals of MySQL/InnoDB B+ Tree Index Navigation from a Forensic Perspective, in: 2019 International Conference on Software Security and Assurance (ICSSA). Presented at the 2019 International Conference on Software Security and Assurance (ICSSA), pp. 46–51. https://doi.org/10.1109/ICSSA48308.2019.00013

Lachmayer, R., Mozgova, I., 2022. Technical Inheritance as an Approach to Data-Driven Product Development, in: Krause, D., Heyden, E. (Eds.), Design Methodology for Future Products: Data Driven, Agile and Flexible. Springer International Publishing, Cham, pp. 47–64. https://doi.org/10.1007/978-3-030-78368-6_3

Lachmayer, R., Mozgova, I., Reimche, W., Colditz, F., Mroz, G., Gottwald, P., 2014. Technical Inheritance: A Concept to Adapt the Evolution of Nature to Product Engineering. Procedia Technology, 2nd International Conference on System-Integrated Intelligence: Challenges for Product and Production Engineering 15, 178–187. https://doi.org/10.1016/j.protcy.2014.09.070

Li, N., Ren, Z., Li, D., Zeng, L., 2020. Review: Automated techniques for monitoring the behaviour and welfare of broilers and laying hens: towards the goal of precision livestock farming. animal 14, 617–625. https://doi.org/10.1017/S1751731119002155

Navarro, C.A., Carrasco, R., Barrientos, R.J., Riquelme, J.A., Vega, R., 2021. GPU Tensor Cores for Fast Arithmetic Reductions. IEEE Transactions on Parallel and Distributed Systems 32, 72–84. https://doi.org/10.1109/TPDS.2020.3011893

Neethirajan, S., 2022. ChickTrack – A quantitative tracking tool for measuring chicken activity. Measurement 191, 110819. https://doi.org/10.1016/j.measurement.2022.110819

Ohashi, T., Saijo, M., Suzuki, K., Arafuka, S., 2023. Deciphering the Drivers of Smart Livestock Technology Adoption in Japan: A Scoping Review, Expert Interviews, and Grounded Theory Approach. https://doi.org/10.48550/arXiv.2307.03338

Raveendran, G., Homeyer, K., Raabe, J., Bartels, T., Lachmayer, R., 2023. Beleuchtung als Einflussfaktor für eine tiergerechte Mastputenhaltung. . Juni.

Schmidt, T.B., Lancaster, J.M., Psota, E., Mote, B.E., Hulbert, L.E., Holliday, A., Woiwode, R., Pérez, L.C., 2022. Evaluation of a novel computer vision-based livestock monitoring system to identify and track specific behaviors of individual nursery pigs within a group-housed environment. Translational Animal Science 6, txac082. https://doi.org/10.1093/tas/txac082

Shin, Y., Choi, K., 1996. Thread-based software synthesis for embedded system design, in: Proceedings ED&TC European Design and Test Conference. Presented at the Proceedings ED&TC European Design and Test Conference, pp. 282–286. https://doi.org/10.1109/EDTC.1996.494314

Üreten, S., Eisenmann [2, M., Nelius [2, T., Cao [3, S., Matthiesen [2, S., Krause [1, D., 2019. A Concept Map for Design Method Experiments in Product Development – A Guideline for Method Developers. DS 98: Proceedings of the 30th Symposium Design for X (DFX 2019) 147–158. https://doi.org/10.35199/dfx2019.13

van Vugt, S., 2008. Installing Ubuntu Server, in: van Vugt, S. (Ed.), Beginning Ubuntu LTS Server Administration: From Novice to Professional. Apress, Berkeley, CA, pp. 1–27. https://doi.org/10.1007/978-1-4302-1081-8_1

Zhan, B., Monekosso, D., Remagnino, P., Velastin, S., Xu, L.-Q., 2008. Crowd analysis: A survey. Mach. Vis. Appl. 19, 345–357. https://doi.org/10.1007/s00138-008-0132-4

Zhang, Z., Liu, H., Meng, Z., Chen, J., 2019. Deep learning-based automatic recognition network of agricultural machinery images. Computers and Electronics in Agriculture 166, 104978. https://doi.org/10.1016/j.compag.2019.104978

https://doi.org/10.1017/pds.2024.224 Published online by Cambridge University Press