Canine Action Recognition: Exploring Keypoint and Non-Keypoint Approaches Enhanced by Synthetic Data

Barbora Bezáková[a] and Zuzana Berger Haladová[b]
Faculty of Mathematics, Physics and Informatics, Comenius University, Bratislava, Slovakia

Keywords: Convolutional Neural Networks, Action Recognition, Generative Models, Pose Estimation.

Abstract: This study focuses on the implementation of deep neural networks capable of recognizing actions from dog photographs. The objective is to implement and compare two approaches. The first approach uses pose estimation, where keypoints and their positions on photographs are analyzed to recognize the performed action. The second approach focuses on recognizing actions in photographs without the need of pose estimation. The image dataset was created using generative models and augmented to increase variability. Results show that combining synthetic and real data effectively addresses the challenge of limited amount of annotated datasets in the field of dog action recognition. It is demonstrated that the integration of artificially generated data into training process can lead to effective results when tested on real-world photographs.

1 INTRODUCTION

Recognizing animal actions is a critical yet inherently complex task due to the wide variability in movement patterns, environmental conditions, and the scarcity of annotated data specific to this application. Despite these challenges, the significance of action recognition cannot be overstated, as it forms the foundation for numerous applications aimed at improving animal welfare and management. From veterinary care to animal training and wildlife monitoring, accurate action recognition enables better decision-making, enhances safety, and promotes animal well-being.

This study focuses on recognizing canine actions in four primary categories: sit, stand, lay down, and run, by employing deep neural networks. We compare methodologies that incorporate keypoint detection with those that operate without it.

Due to limited annotated data for canine actions, we investigate synthetic data generation to augment existing datasets. This approach could reduce reliance on extensive manual annotations, showing how synthetic data can enhance model training and generalization on real data.

The goal of pose estimation is to determine the position and orientation of humans or animals in space. This process works by localizing keypoints, whose number and placement depend on the specific task. In pose estimation, the keypoints are typically various points on the head and major joints of the body. Action recognition involves identifying the movements performed by the observed subject.

A significant challenge in action recognition from still images, as opposed to videos, is the absence of temporal features, which can make it difficult to accurately represent actions. This limitation is why pose estimation is often integrated into action recognition models. Integrating pose estimation into an action recognition model can help improve the accuracy of action identification, as it captures the positioning of objects within the photograph (Girish et al., 2020).

The structure of this study is organized as follows: Section 2 reviews relevant prior work in the fields of action recognition and pose estimation. Section 3 outlines the pipeline used throughout the process. Section 4 details the datasets, both existing and newly created, while Section 5 covers the implementation and evaluation of the deep learning models. Finally, the study concludes with key findings in Section 6.

2 PREVIOUS WORK

Computer vision systems for tracking animals enable continuous, non-intrusive monitoring, which helps detect signs of stress, illness, or environmental changes early. By reducing the need for human pres-

[a] https://orcid.org/0009-0003-7275-6843
[b] https://orcid.org/0000-0002-5947-8063

Bezáková, B. and Haladová, Z. B.
Canine Action Recognition: Exploring Keypoint and Non-Keypoint Approaches Enhanced by Synthetic Data.
DOI: 10.5220/0013195000003912

In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2025) - Volume 3: VISAPP*, pages 584-591
ISBN: 978-989-758-728-3; ISSN: 2184-4321

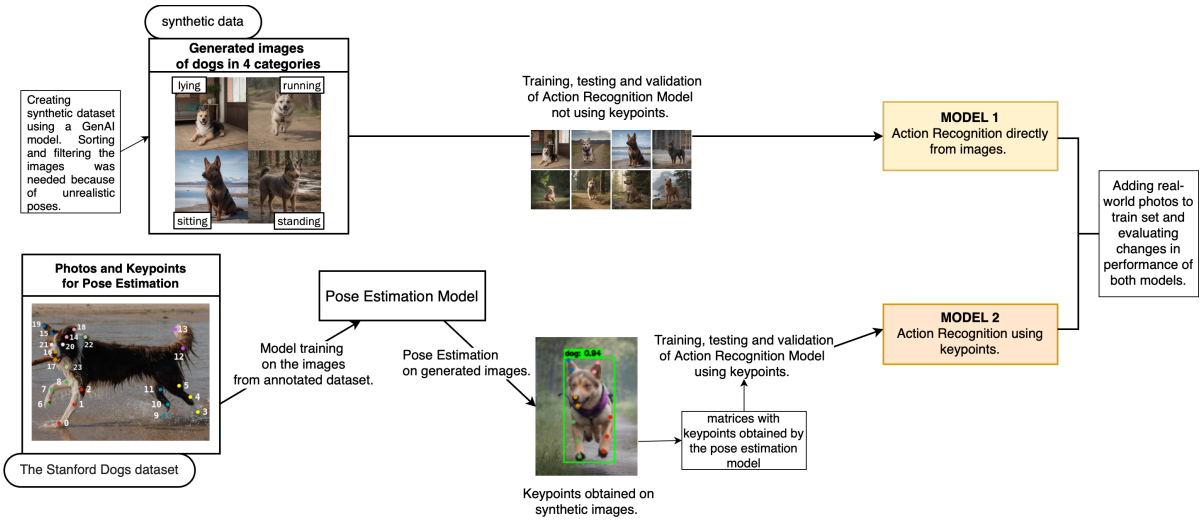

Figure 1: Pipeline of the process described in this study.

ence, it minimizes stress and allows remote observation, making it ideal for large-scale animal management and improved welfare (Fernandes et al., 2020).

The Stanford Dog Dataset (Khosla et al., 2011) was created to support computer vision tasks related to dog breeds. It contains images of 120 different dog breeds captured in various environments and lighting conditions. The dataset was subsequently annotated with 24 keypoints marking the positions of different points on the head, body, tail and legs (Biggs et al., 2020) and made it a valuable resource for training pose estimation models.

To address the current challenges in animal pose estimation, a comprehensive overview highlights key issues, including the scarcity of annotated datasets (Jiang et al., 2022). This work classifies 2D pose estimation, which focuses on detecting the 2D coordinates of keypoints, based on the number of animals in the scene, distinguishing between single-animal and multi-animal pose estimation.

The use of advanced approaches, such as leveraging GPT-inspired architectures and datasets like AnimalML3D for generating realistic animal motions (Yang et al., 2024), offers valuable insights. While not applied here, such methods could enrich work like ours in future research.

The widely popular and effective YOLO algorithm (Jocher et al., 2023a) is currently used in the field of computer vision for object detection, image segmentation, pose estimation, and classification. In this paper, we will use a pre-trained model for classification.

Action recognition from RGB data presents significant challenges due to background variability, differing viewpoints, and lighting conditions, all of which can reduce model accuracy (Sun et al., 2022).

3 PIPELINE

The pipeline shown in Figure 1 is divided into two distinct sections representing the development of two action recognition models. In the upper part, we present the steps undertaken to train an action recognition model that classifies actions directly from images. Initially, synthetic images were generated using deep learning model, categorized into four classes: lying, running, sitting, and standing. Following this, the training, testing, and validation of the action recognition model were performed. A pre-trained YOLO classification model yolo served as the backbone for extracting image features relevant to action recognition (Jocher et al., 2023b).

The lower part illustrates the process of developing an action recognition model using keypoint-based pose estimation. The initial step consisted of utilizing a dataset containing canine images annotated with keypoints, which was essential for training the pose estimation model. This model was applied to the synthetic images to detect body keypoints. Subsequently, the extracted keypoint coordinates were used to train, test, and validate an action recognition model using keypoint coordinates.

While the pipeline itself illustrates the development of these models, the subsequent sections of this paper will delve into the limitations of each method, providing an evaluation of their accuracy. In the end of paper, we will show how we can combine real photographs and generated data to improve the performance of action recognition models.

4 DATASETS

Large datasets with publicly available photos, like ImageNet (Deng et al., 2009), are commonly used to train deep learning models. However, there is no annotated database specifically suited to the task we address in this paper. To create a sufficiently large dataset, we would need to source images from multiple collections. For instance, we examined the COCO dataset (Lin et al., 2015), which includes image captions such as the following:

```
{"image_id": 65485,"id": 18520,
"caption": "A black dog sitting on
top of a boat. this is a black dog
riding on a boat. A dog that is
sitting on a boat. A black dog is sitting
on the bed of a boat The black dog sits
on a boat that is on the water."}
```

The text corresponds to the image shown in Figure 2b. Using an online search tool COCO Captions Explorer (Department of Computer Science at Rice University, 2020), we estimated the number of images available for each category. While categories such as "sit," "stand", and "lay" contained around 1000 to 1600 images, we encountered difficulties with "run", as searching for the phrase "running dog" returned only 192 images. Finding enough images would require searching for synonyms, and even then, irrelevant results would likely need filtering.

This limitation, combined with the presence of mislabeled or irrelevant images, made the task of building a reliable dataset difficult. When we searched for "standing dog" we found some images like shown in Figure 2a. The same problem is with other webpages like Flickr or Shutterstock. For these reasons, we chose to generate our own data using deep learning models. This approach allowed us to create a dataset of images featuring dogs in four specific poses: standing, lying, sitting, and running.

(a) (b)

Figure 2: Examples of dog images from the COCO dataset.

4.1 Synthetic Image Generation

Scraping and sorting datasets requires not only collecting large volumes of images but also manually annotating them, which is time-consuming. In contrast, generating synthetic images produces pre-annotated data directly aligned with specified actions. Although some filtering is required to exclude unrealistic poses, this approach appears to be a more efficient and scalable solution.

To generate images for training of the network focused on action recognition, we utilized the StabilityMatrix (LykosAI, 2024) interface. It allowed us to load a model for image generation from civitai.com and huggingface.co, where we identified and selected the Realistic Vision V6.0 B1 model (Realistic Vision, 2024).

The settings used for generating images were as follows: the CFG scale, which controls how strictly the image generation process follows the given text input, was set to 2. Higher values make the image conform more closely to the text, but maximum settings do not always yield the best results, as strict conformity may reduce diversity and quality. The steps parameter was set to 100, determining how many iterations the process would undergo, with noise being gradually removed at each step. The sampler method chosen was Euler Ancestral, referring to the approach used for noise removal during image generation (getimg.ai, 2024). The image dimensions were set to 512×512 pixels.

4.1.1 Prompt Creation

In prompt engineering for generating images from text, subject terms are essential for precise image creation. Prompt modifiers are used to enhance image quality and gain greater control over the generation process. Negative prompts can filter out specific subjects or styles, while positive prompts allow for style blending. Prompt writing is an iterative process, beginning with subject terms followed by adding modifiers and solidifiers (Oppenlaender, 2022).

Initially, we used simple prompts like "a sitting dog" for image generation, but this method proved ineffective, resulting in significantly lower quality images, such as dogs with distorted body parts. We found that longer, more detailed prompts yielded better results. Prompts such as "a photorealistic sitting dog in a photorealistic environment" and "a photorealistic dog sitting on a street, side view, floppy ears", along with negative prompts like "disfigured, deformed". These refined prompts were applied to generate images across all four categories.

Despite specifying poses, errors occurred, such

(a) (b) (c) (d)

Figure 3: Examples of synthetic images with unusual features and misplaced body parts are shown. In Figure 3a, two dogs are merged into a single body. Figure 3b depicts a dog with extra limbs, which could also pose challenges during pose estimation. The dog in Figure 3c resembles a human figure, while the dog in Figure 3d has a completely distorted body.

as dogs lying instead of sitting, body deformations, missing or extra limbs, blurred parts, or human-like appearances (see Figures 3a–3d). To ensure each category contained at least 1000 images, we generated 1.5–2 times this amount and manually selected distortion-free images for training.

4.2 Data for Pose Estimation

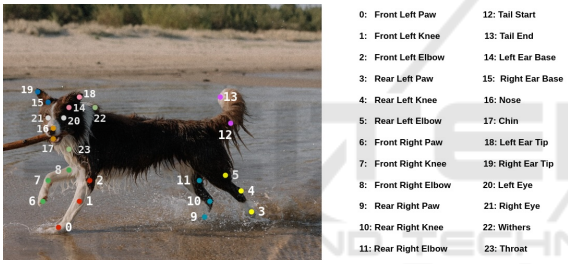

Figure 4: Keypoints and their positions. The keypoints are consistently ordered, ensuring data uniformity throughout the processing pipeline.

We used the previously mentioned Stanford Dog Dataset (Khosla et al., 2011) to train a pose estimation model. In addition to obtaining data with images of dogs and corresponding keypoint files from StanfordExtra (Biggs et al., 2020), several further steps were necessary for our work. Since the keypoint data was annotated separately and were not part of the original database, they had to be obtained independently, following the instructions provided in the original repository (Biggs et al., 2020).

The specific keypoints selected by the annotation creators in the utilized dataset are shown in Figure 4. These 24 keypoints were chosen to accurately capture the position of the body. The order of these keypoints is consistently maintained, meaning that keypoint number 0 (left front paw) will always represent the first point in any representation.

After obtaining all the necessary data, we split them into training, validation, and test sets. The data was divided by the authors into corresponding groups, with 6773, 4062, and 1703 samples, respectively. Each folder contained images and their corresponding text files with keypoint coordinates. The dataset, which included keypoint coordinates, was not always entirely accurate, and errors likely occurred during the annotation process.

5 IMPLEMENTATION

5.1 Pose Estimation Model

The pose estimation model was trained with the parameters listed in Table 1; this process was adapted from the method described by Dawn (Dawn, 2023).

Table 1: Training parameters for the pose estimation model.

Parameter	Value
IMAGE_SIZE	512
BATCH_SIZE	16
CLOSE_MOSAIC	10
MOSAIC	0.4
FLIP_LR	0.0
EPOCHS	150

We used the trained model to obtain keypoints for dogs on the generated images. The normalized keypoints were then saved into text files. The keypoints obtained for the entire database of generated images will be used for training and testing a model designed for action recognition using keypoints. The goal is to determine the performed action based on the position of the keypoints. We transformed the keypoints into matrices. The dimensions of each matrix are 24×3, where the first column always represents the x-coordinate, the second represents the y-coordinate, and the third indicates the visibility of the corresponding keypoint. Each row of the matrix thus represents a single keypoint on the body, as shown in the example in Figure 5.

$$A = \begin{bmatrix} x_1 & y_1 & v_1 \\ x_2 & y_2 & v_2 \\ \vdots & \vdots & \vdots \\ x_{24} & y_{24} & v_{24} \end{bmatrix} \qquad B = \begin{bmatrix} 0.54 & 0.2 & 2 \\ 0.0 & 0.0 & 0 \\ \vdots & \vdots & \vdots \\ 0.12 & 0.56 & 2 \end{bmatrix}$$

Figure 5: Matrices with keypoints. Matrix A represents the general form with variables x, y, and v, while matrix B shows a specific example with real values. YOLO algorithm uses visibility values where 0 represents hidden keypoints, 1 represents partially visible keypoints, and 2 represents fully visible keypoints. We focus exclusively on the values 0 and 2, as the original dataset does not contain any keypoints labeled as partially visible.

In Figure 6, we can see example of a generated photo along with the keypoints estimated by the pose estimation model. The coordinates of these keypoints are stored in matrices for subsequent analysis.

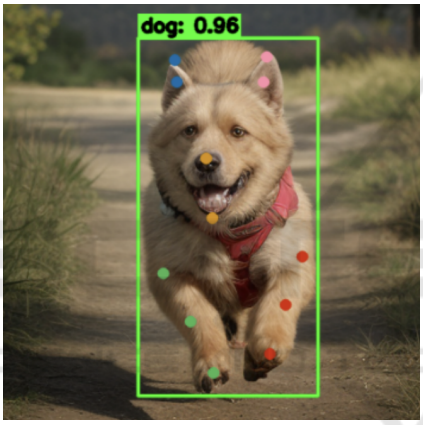

Figure 6: Keypoints obtained on a generated image using the Pose Estimation model described in 5.1.

5.2 Action Recognition Models

We present two approaches in this section: one that relies on keypoint detection to identify the performed action and another that uses only raw image data for action classification. Both methods are essential for understanding the role of feature extraction in recognizing actions. The following subsections will delve into the implementation details and performance of these models.

5.2.1 Action Recognition Model Using Keypoints

Our input data consists of matrices with keypoints mentioned in 5.1, and the output data contains information about the actions being performed, which are assigned to the respective matrices. These data pairs form the basis of our model and are used to train the neural network with the goal of recognizing actions based on the position of keypoints obtained from the input images.

Compared to the model without keypoint usage, we used approximately 75% of the data for training, as the model occasionally misidentified keypoints during pose estimation, detecting two dogs in an image where there was only one. We did not want to include such text files in the training process, as we are focusing on action classification for a single dog. After removing the wrong files, we evenly split the data into training, testing, and validation sets in a ratio of 75 : 15 : 10. Our experiments showed that training the model for 30 epochs was the most effective approach.

To address the issue, we propose an approach where we first use YOLO for dog detection. This step allows us to detect and isolate individual dogs in an image, even when there are multiple dogs present. Once detected, we crop the dogs from the original image and send them to the model for pose estimation. This way, we could ensure that only the relevant portion of the image is processed by the pose estimation model, reducing the risk of misidentification.

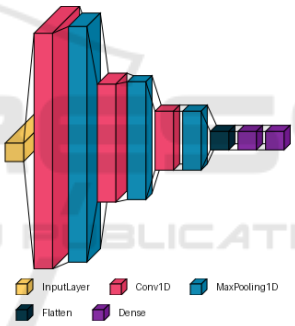

Figure 7: Model architecture: The input layer processes a 24×3 matrix representing the keypoints, where each row corresponds to a keypoint's coordinates and visibility.

Using tensorflow.keras.models, we implemented a deep convolutional neural network model shown in figure 7. The model contains three convolutional layers with 64, 32, and 16 filters, respectively, followed by pooling layers, a Flatten layer, and Dense layers with 16 and 4 neurons (the output layer). The total number of parameters is 8,708.

5.2.2 Action Recognition Model not Using Keypoints

For each action, we selected the top 1,000 images, organized them in a split-directory format required for neural network training (Jocher, 2023), and divided them randomly into training (700 images), validation (150 images), and testing (150 images) sets with no overlap. Then we used a pre-trained model yolov8n-cls.pt (Jocher et al., 2023b), designed for classifica-

tion into categories, and trained it on the synthetic dataset for 30 epochs.

5.3 Evaluation

The model trained on synthetic data without the use of keypoints was tested using a dataset consisting of real photographs divided into four separate categories, each containing around 50 images.

Some examples of classification are shown in Figure 8. Figure 8a represents the true labels and Figure 8b represents the classification done by the model. Certain errors identified in the model underscore the complexities involved in recognizing canine actions from images. Differentiating between a standing and running dog can pose challenges even for humans, given that photographs do not convey the temporal progression of movement. The lack of dynamic context requires the model to base its judgments solely on a single static image, which can result in confusion between visually similar poses. Additionally, when legs or other body parts were obscured, the model exhibited a tendency to make more significant mistakes. This limited visibility can hinder the model's ability to accurately identify actions, as it does not possess all the requisite information for precise decision-making.

The model using keypoints coordinates was the most successful in predicting the category "sit", although it incorrectly classified some instances as belonging to the category "lay" or "run". The model also performed well in identifying the category of photographs from the "run" class, with this category most frequently misclassified as "lay". We show several examples of these misclassifications in Figure 9.

The action recognition model that relies on keypoint coordinates is highly dependent on the pose estimation model. Inaccuracies in the training data for the pose estimation model can lead to minor errors that affect the learning and classification accuracy of the action recognition model.

To compare both approaches, we calculated relevant metrics for the models—accuracy, precision, sensitivity, and F1 score. These metrics allowed us to evaluate and compare the performance of the individual approaches on real photographs. In Table 2, we can see the calculated metrics for the two models. The model using keypoints coordinates achieved higher values in all observed metrics. Although the keypoint model achieves slightly better results, the differences are not significant, and we cannot claim that it is unequivocally superior.

Table 2: Comparison of two models: Model 2 incorporates keypoint coordinates, while Model 1 does not. The differences between them are minor, and further statistical analysis is needed to evaluate their significance.

Metric	Model 1	Model 2
Accuracy	0.665	**0.672**
Precision	0.677	**0.693**
Recall	0.666	**0.672**
F1 Score	0.665	**0.676**

5.4 Integrating Real Photographs for Enhanced Model Training

The action recognition models were initially trained exclusively on synthetic data generated by deep learning models. However, synthetic data often lack the complexity and diversity inherent in real-world images, which can exhibit various forms of noise and artifacts, including motion blur, low lighting, and images taken from significant distances. These limitations may impede the models' ability to generalize effectively to real-world scenarios, potentially reducing accuracy and reliability in applied settings. To address this, we designed an experiment to integrate real images into the training dataset. These real images were sourced from the COCO dataset, selectively filtered based on descriptive tags to match our target categories. This approach aims to evaluate whether augmenting the synthetic dataset with real-world images can enhance the models' capacity to adapt to real conditions, despite synthetic data remaining the dominant component in the training set.

Initially, the dataset for each action category contained approximately 700 images or sets of keypoints. To increase data variability, we added 100 additional real photographs for the model without keypoint detection and 100 corresponding keypoint files for the keypoint-based model. These new data was placed in the appropriate category folders, and the training process was carried out using the same conditions and parameters as in previous sessions.

The inclusion of real photographs in the training dataset for the model that does not utilize keypoints led to a significant improvement in its performance.

Table 3: Comparison of the performance of 2 models without keypoint detection - before and after adding real data to train set. Tested on real photographs.

Metric	Before	After
Accuracy	0.665	**0.800**
Precision	0.677	**0.813**
Recall	0.666	**0.813**
F1 Score	0.665	**0.801**

(a) Ground truth categories (b) Predicted categories

Figure 8: Comparison of actual and predicted categories by non-keypoint based model.

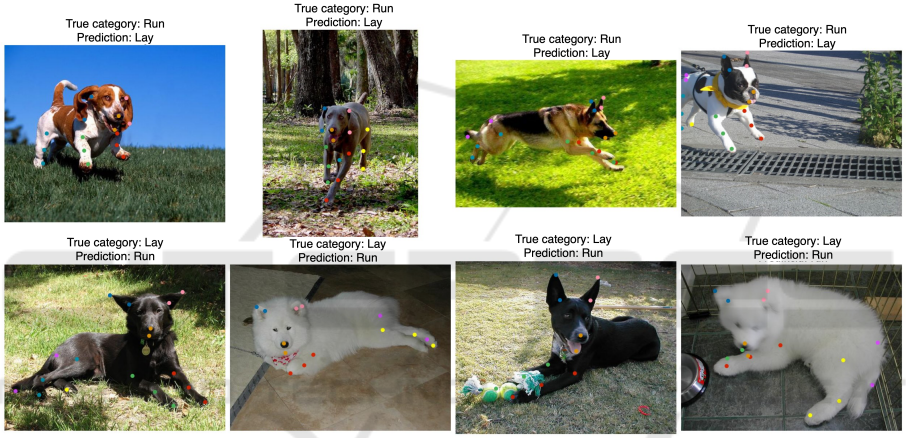

Figure 9: Examples of misclassification for *lay* and *run* by keypoint-based model. The model struggled to differentiate between these two categories, likely due to occluded body parts and the absence of temporal sequence information.

In contrast, the keypoint-based model performed better in some categories but got worse in others. Adding real data while maintaining 30 epochs negatively impacted performance. To address this, we increased the number of epochs to 50, as keypoints from real data can be more challenging to process due to factors like occluded body parts. Extending the training period seems to have allowed the model to better adapt to these real-world conditions, resulting in improved accuracy and other metrics. The gains were more modest than those observed in the non-keypoint model.

To confirm that the improvement wasn't solely due to the increased number of epochs, Table 5 presents the validation results of a model trained only on synthetic data and validated on real photos, first after 30 epochs and then after 50 epochs. The results indicate a decline in performance, suggesting the model may have overfitted after training on generated data for 50 epochs.

Table 4: Comparison of the performance of the model using skeleton detection before and after adding real data to training set, tested on real data.

Metric	Before (30 epochs)	After (50 epochs)
Accuracy	0.672	**0.721**
Precision	0.693	**0.722**
Recall	0.672	**0.721**
F1 Score	0.676	**0.720**

Nevertheless, the incorporation of real data improved the model's performance, indicating that further increases in real data amount may continue to enhance its accuracy and robustness. Additionally, expanding the model's exposure to diverse scenarios is likely to improve its generalization capabilities in practical applications.

Table 5: Comparison of the performance of the model using skeleton detection after 30 epochs and 50 epochs, trained only with generated data and tested on real data.

Metric	30 epochs	50 epochs
Accuracy	**0.672**	0.602
Precision	**0.693**	0.635
Recall	**0.672**	0.6019
F1 Score	**0.676**	0.606

6 CONCLUSIONS

Action recognition in animals presents a complex challenge due to the variability in movement and the diverse environments in which animals operate. Creating a high-quality dataset for training deep learning models involved several challenges, particularly due to a lack of annotated data for this specific task. We, therefore, investigated the use of existing datasets and assessed their suitability for our needs, while an alternative approach was also used as we generated image data with deep learning models.

Comparing the two approaches to action recognition revealed that the model using keypoints performed marginally better than the non-keypoint model. However, after including real data in the training set, we found that the model without keypoint detection achieved significantly better results on real photos. This suggests that non-keypoint models have strong potential in canine action recognition and may adapt more readily to the variability and complexity of real-world images. We believe that using generated data to augment the training set, combined with real photographs, can substantially aid in training models that require large datasets.

ACKNOWLEDGEMENTS

This work was supported by the grant KEGA 004UK-4/2024 “DICH: Digitalization of Cultural Heritage”.

REFERENCES

Biggs, B., Boyne, O., Charles, J., Fitzgibbon, A., and Cipolla, R. (2020). Who left the dogs out?: 3D animal reconstruction with expectation maximization in the loop. In *ECCV*.

Dawn, K. (2023). Animal Pose Estimation: Fine-tuning YOLOv8 Pose Models. https://learnopencv.com/animal-pose-estimation/#Data-Configuration. Last Access: 13.11.2023.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Department of Computer Science at Rice University (2020). COCO Captions Explorer. https://vislang.ai/coco-explorer. Last Access: 8.11.2024.

Fernandes, A. F. A., Dórea, J. R. R., and Rosa, G. J. d. M. (2020). Image analysis and computer vision applications in animal sciences: an overview. *Frontiers in Veterinary Science*, 7:551269.

getimg.ai (2024). Parameters. https://getimg.ai/guides/stable-diffusion-parameters. Last Access: 31.3.2024.

Girish, D., Singh, V., and Ralescu, A. (2020). Understanding action recognition in still images.

Jiang, L., Lee, C., Teotia, D., and Ostadabbas, S. (2022). Animal pose estimation: A closer look at the state-of-the-art, existing gaps and opportunities. *Computer Vision and Image Understanding*, 222:103483.

Jocher, G. (2023). Image Classification Datasets Overview. https://docs.ultralytics.com/datasets/classify/. Last Access: 20.3.2024.

Jocher, G., Chaurasia, A., and Qiu, J. (2023a). Ultralytics YOLOv8.

Jocher, G., Chaurasia, A., and Qiu, J. (2023b). Ultralytics YOLOv8 - yolov8n-cls.pt.

Khosla, A., Jayadevaprakash, N., Yao, B., and Fei-Fei, L. (2011). Novel Dataset for Fine-Grained Image Categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2015). Microsoft COCO: Common Objects in Context.

LykosAI (2024). StabilityMatrix. https://github.com/LykosAI/StabilityMatrix/tree/main.

Oppenlaender, J. (2022). A taxonomy of prompt modifiers for text-to-image generation. arxiv. *arXiv preprint arXiv:2204.13988*.

Realistic Vision (2024). Realistic Vision V6.0 B1. https://civitai.com/models/4201/realistic-vision-v60-b1.

Sun, Z., Ke, Q., Rahmani, H., Bennamoun, M., Wang, G., and Liu, J. (2022). Human action recognition from various data modalities: A review. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3200–3225.

Yang, Z., Zhou, M., Shan, M., Wen, B., Xuan, Z., Hill, M., Bai, J., Qi, G.-J., and Wang, Y. (2024). Omnimotiongpt: Animal motion generation with limited data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1249–1259.