DEEP LEARNING-BASED TRACKING OF MULTIPLE OBJECTS IN THE CONTEXT OF FARM ANIMAL ETHOLOGY

R. Ali[1,*], M. Dorozynski[1], J. Stracke[2], M. Mehltretter[1]

[1]Institute of Photogrammetry and GeoInformation, Leibniz University Hannover, Germany
(ali, dorozynski, mehltretter)@ipi.uni-hannover.de

[2]Institute of Animal Science, University of Bonn, Germany
jenny.stracke@itw.uni-bonn.de

Commission II, WG II/5

KEY WORDS: Image Sequence Analysis, Multi-Object Tracking, Tracktor, Animal Science, Poultry Tracking

ABSTRACT:

Automatic detection and tracking of individual animals is important to enhance their welfare and to improve our understanding of their behaviour. Due to methodological difficulties, especially in the context of poultry tracking, it is a challenging task to automatically recognise and track individual animals. Those difficulties can be, for example, the similarity of animals of the same species which makes distinguishing between them harder, or sudden changes in their body shape which may happen due to putting on or spreading out the wings in a very short period of time. In this paper, an automatic poultry tracking algorithm is proposed. This algorithm is based on the well-known tracktor approach and tackles multi-object tracking by exploiting the regression head of the Faster R-CNN model to perform temporal realignment of object bounding boxes. Additionally, we use a multi-scale re-identification model to improve the re-association of the detected animals. For evaluating the performance of the proposed method in this study, a novel dataset consisting of seven image sequences that show chicks in an average pen farm in different stages of growth is used.

1. INTRODUCTION

Analysing the behaviour of farm animals is a fundamental prerequisite for defining their needs and thus ensuring animal welfare. In this context, it is of great interest to determine the movement of animals in their habitat as a function of time, to analyse their behaviour in groups, during foraging and in relation to the use of space. In animal science, it is currently common practice to use video recordings of animal behaviour which are manually analysed. Since such manual procedures are extremely time-consuming, the goal of this work is to provide automatically extracted trajectories of all animals in an image sequence. To the best of our knowledge, there is only little research in the area of image-based tracking of farm animals (Zhang et al., 2019; Bergamini et al., 2021), which is particularly true for tracking of poultry (Li et al., 2020; Neethirajan, 2022). Up to now, there is no effective method to identify individual animals as a function of time. The research regarding animal tracking concentrates predominantly on animal movements. This can be done, for example, by analysing the optical flow patterns at flock level, to determine if animals are infected by a human pathogen (Colles et al., 2016), or by applying background subtraction to detect the lying-down behaviour of individual broiler chicken (Aydin, 2017). On the other hand, many approaches have been developed to track pedestrians as illustrated by the extensive review on this topic by Luo et al. (2022).

While pedestrian and animal tracking have similar goals, i.e., automatically detecting and associating each individual in an image sequence to a track, some differences arise due to animal anatomy and behaviour: The first difference concerns the similarity of individuals of the same species. While pedestrians can often be distinguished from each other by their appearance, size, as well as colour and shape of clothing, this is not always possible for animals of the same species. This is especially relevant when re-identification becomes necessary, for example, to connect two partial trajectories resulting from a temporary occlusion of an animal. Further challenges in tracking poultry arise with re-identification over longer time periods, as young animals grow much faster compared to humans resulting in relatively fast changes in appearance (Wurtz et al., 2019). Second, in poultry, sudden and significant changes in appearance can be observed when the wings are put on or spread out in a very short time. This makes it challenging to detect the animals in the image sequence and to re-identify all the detections of the individual animals in the course of the entire image sequence to determine those animal's trajectories (Kashiha et al., 2013). Third, in contrast to pedestrians who move in an almost straight path on sidewalks, or move in small groups from which some motion models can be derived by using a social force model (Helbing and Molnár, 1995), it is harder to justify any such assumption to describe the motion of poultry as a function of time (Colantonio et al., 2007). As such animals, especially young ones, are very energetic and playful in their movements, it is harder to model their motion. Fourth, the recording configuration often differs between pedestrian tracking and animal tracking. While the footage of pedestrians used for tracking is often captured by cameras at street level with the optical axis parallel to the ground level, cameras used to track livestock are usually mounted below the barn ceiling and result in nadir or oblique images.

In this paper, we investigate the potential of animal tracking using the tracktor approach presented in (Bergmann et al., 2019).

* Corresponding author

This approach is based on the Faster R-CNN model (Ren et al., 2015) and thus, it can detect objects that are small and close to each other well since it is a region proposal-based method (Zhao et al., 2019). Furthermore, tracking using the tracktor approach is achieved via bounding box regression, i.e., no explicit motion model is needed. To improve the re-identification of animals that have been occluded temporary, we use MuDeep (Qian et al., 2017). The MuDeep network architecture is based on a Siamese network. It learns features at different scales and evaluates their importance for object matching. The underlying assumption for the usage of this method is that MuDeep improves the re-association of the animals by focusing on the small differences in their appearance given that those animals are similar in many respects.

The main contribution of the present work is a method for the tracking of multiple animals in a confined space for research purposes, focusing on the tracking of poultry. For this intent, we use a tracking model that does not require a defined explicit motion model by exploiting the regression head of a detector to perform temporal realignment of object bounding boxes. Additionally, we use a multi-scale re-identification model to address problems that originate from occlusions.

2. RELATED WORK

Much research has been carried out on pedestrian tracking during the last years (Klinger et al., 2014; Tang et al., 2017; Chen et al., 2018; Ristani and Tomasi, 2018; Nguyen et al., 2019); these algorithms are predominantly based on the tracking-by-detection paradigm (Chen et al., 2018; Bergmann et al., 2019). In tracking-by-detection the tracking problem is broken down into two steps: i) object detection in each frame, ii) object association between adjacent frames. It should be noted that the quality of the tracking algorithm is limited by the performance of the underlying detection method (Luo et al., 2022). Recently, neural network-based detectors have outperformed conventional methods for detection (Ren et al., 2015; Redmon and Farhadi, 2018), making them the main choice for tracking-by-detection approaches (Bergmann et al., 2019).

Many methods can be used for the second step of tracking-by-detection in the context of pedestrian tracking. One of those methods is motion modelling and trajectory prediction (Shafique et al., 2008). This method captures the dynamic behaviour of an object to estimate its potential position in future frames by expressing assumptions about the movements of the individuals to be tracked in form of a motion model that can be integrated into a filter approach (Luo et al., 2022). One could, for example, make a simple constant velocity assumption. Alternatively, the motion model can be made more complex by applying prior knowledge from a social force model (Helbing and Molnár, 1995), being an example for interaction modelling (Luo et al., 2022). The social force model describes the behaviour of crowds as a result of the interactions of individuals. Concerning our task, it is hard to adapt a social force model from pedestrian tracking to poultry tracking due to difficulties in modelling animal motion and the limited understanding of the interaction of stock individuals (Colantonio et al., 2007). Another method is to use an appearance-based model to create links between the detected object in the individual frames. Such information on the appearance can be particularly helpful in crowded scenes with many object-object occlusions where an ID-switch is probable to happen. Such an appearance-based model can exploit optical flow (Ali and Shah, 2008), point features (Ommer et al., 2009) or gradient-based features such as features of a histogram of oriented gradients (Dalal and Triggs, 2005). Due to the significant advances of machine learning approaches in recent years, many re-identification models based on Convolutional Neural Networks (CNNs) have been developed in the context of pedestrian tracking (Li et al., 2018; Yu et al., 2018). However, such methods are not necessarily reliable for animal tracking since animals often have similar shapes, and colour statistics are often contaminated by background pixels and illumination changes, while the differences in appearance between the animals are often subtle and only detectable at particular locations and scales.

The proposed algorithms for animal detection in the literature mainly tackle specific animal behaviour aspects, such as eating or drinking (Li et al., 2020). This is achieved by detecting the position of an animal relative to the position of the feeder, water or the nest (Li et al., 2020). While those systems could in principle also be used for tracking, they exclusively concentrate on detecting different behaviours of the animals and neglect tracking. By using wireless wearable sensors (e.g., accelerometer, RFID microchip) (Chien and Chen, 2018) the position of the animal can be detected if the animal is near the feeder, water or the nest. Due to the expense and invasive nature of wireless wearable sensors, visual-based monitoring systems have been used more and more in recent years (Li et al., 2020). Bergamini et al. (2021) use an image-based tracking algorithm to extract long-term behaviour changes of pigs. They use the YOLO v3 (Redmon and Farhadi, 2018) model to detect pigs in each frame independently. In the first frame, a new track is created and initialised for each detection. While this tracker is based on a Minimum Output Sum of Squared Error (Bolme et al., 2010), it uses an adaptive correlation for object tracking which produces stable correlation filters once initialised. In the following frames, the updated tracks and single-frame detections are matched together comparing the Intersection over Union (IoU) and their appearance. The best assignments are found with the Hungarian algorithm (Kuhn, 1955) and a track is removed, if no match is found. On the other hand, if a detection is not matched to any track, a new track is created and initialised. The algorithm in (Bolme et al., 2010) is easy to calculate and can quickly track objects, but it does not guarantee accurate results when the object's appearance changes. Neethirajan (2022) uses the YOLO v5 model, which is based on YOLO v3 (Redmon and Farhadi, 2018), to detect chickens in each frame. To track each individual chicken, a Kalman filter is applied. The downside of this approach is that the accuracy of the Kalman filter depends on the assumption of linear motion for any chick to be tracked. If a chick takes some abrupt turns, which often happens, the nonlinear movement cannot be handled well by the Kalman filter framework.

Another possibility to achieve the stated goals is Tracktor (Bergmann et al., 2019). Tracktor is a non-motion-model based tracking approach, simplifying tracking by eliminating the need for any knowledge or assumptions about animal motion behaviour. Tracktor requires an image sequence of the animals to track as input. It tackles multi-object tracking by exploiting the regression head of a detector to perform temporal realignment of object bounding boxes; objects are detected and classified using Faster R-CNN (Ren et al., 2015). The data association step in the tracktor method is done by combining object classification scores from the detection step and IoU between bounding boxes of two subsequent time steps. Tracktor gen-

erates a region suggestion for each animal in each frame and combines the suggestions of different frames to form a trajectory for the observed animal over the entire image sequence.

3. TRACKTOR FOR ANIMAL TRACKING

Our methodology is based on the tracktor approach (Bergmann et al., 2019), where a schematic description of our method can be found in Figure 1. In contrast to Tracktor, we use the MuDeep model (Qian et al., 2017) for re-identification, aiming to increase the accuracy and the reliability of the data association step and to better handle occlusions. Further, we employ additional augmentations during training. To make this paper self-contained, the subsequent sections will contain a review of the approach in (Bergmann et al., 2019), highlighting our contributions and extensions of the original approach.

3.1 Animal Detection

In the first step, detection and classification of all animals are carried out using Faster R-CNN. Faster R-CNN can be separated into three different steps. The first step is image feature-map extraction in which a backbone CNN is employed, with ResNet-50-FPN (He et al., 2016) being used as backbone in the present work. In the next step, the network learns whether a pixel belongs to a certain object and estimates the size of that object. This is done by sliding a window over the feature maps and placing a set of "anchors" on corresponding positions between the input image and the window over the feature maps. Following (Bergmann et al., 2019), we use nine anchors with three different aspect ratios and three different sizes. This ensures that animals of different sizes are detected. This step generates a multitude of bounding box proposals for each potential animal. In the third step, feature maps for each proposal are extracted via Region of Interest pooling (Girshick, 2015) and passed to the classification and regression heads. The classification head assigns an object score s_k^t to each proposal. This score represents the probability s in frame t of a proposal k showing an object of interest, i.e., an animal. The regression head refines the coordinates of the proposals that contain animals. Next, non-maximum-suppression is applied to obtain the final set of detected animals.

3.2 Animal Tracking

Tracktor tackles multi-object tracking by exploiting the regression head of the detector to perform temporal realignments of object bounding boxes, which is possible in case of a high frame rate. Tracktor extracts the trajectories of objects in a video sequence. Each trajectory is given as a list of ordered object bounding boxes $T_k = \{b_0^k, b_1^k, ...\}$, where b_t^k is a bounding box b of object k in frame $t \in \{0, 1, ...\}$. In each frame t, the list of detected objects assigned to a trajectory is defined as $B_t = \{b_t^0, b_t^1, ..., b_t^{K_t-1}\}$ listing the bounding boxes b of all K_t objects $k \in \{0, 1, ..., K_t - 1\}$ in frame t. At $t = 0$ the tracker initialises tracks from the first set of detections D_0 as $B_0 := D_0 = \{d_0^0, d_0^1, ...\}$, where d_0^j is the j^{th} bounding box d delivered by the detector in frame $t = 0$ that is not yet assigned to a trajectory and has an object score s_j^0 bigger than a threshold $\lambda_{detect} = 0.5$, as defined in the original approach of Bergmann et al. (2019). For all frames $t > 0$ the following two steps are carried out to determine B_t:

- Bounding box regression (red arrows in Figure 1): given the assumption that an object k moves only slightly between two subsequent frames, its trajectory list $T_k = \{b_0^k, ..., b_{t-1}^k\}$ is extended from the preceding frame $t-1$ to the current frame t, leading to $T_k = \{b_0^k, ..., b_{t-1}^k, b_t^k\}$. This is achieved by exploiting bounding box regression, i.e. by regressing the bounding box b_{t-1}^k in frame $t-1$ to the object's new bounding box position b_t^k at frame t. This is conducted for all K_{t-1} objects, where K_{t-1} is the number of objects in frame $t-1$, leading to the list of bounding boxes $\{b_t^0, ..., b_t^{K_{t-1}-1}\}$ for the current frame t referred to as active trajectories.

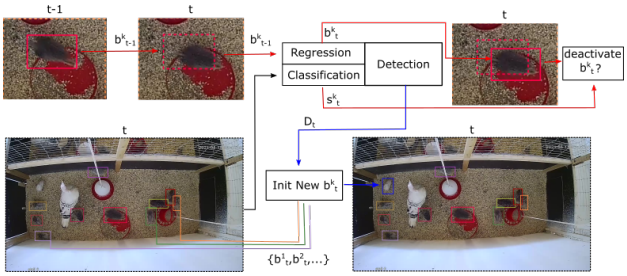

Figure 1. The proposed methodology performs simultaneous tracking of multiple animals using the tracktor approach. The two basic processing steps, bounding box regression and initialisation, are shown by red and blue arrows, respectively, for a single image at time t using an example from our poultry dataset. In the bounding box regression step, the object detector regression head adjusts the existing bounding boxes b_{t-1}^k of frame $t-1$ to the new positions of the objects at frame t. The corresponding object classification scores s_t^k of the new bounding boxes are used to deactivate potentially occluded tracks. To determine whether a new bounding box has to be initialised, the Intersection over Union (IoU) is calculated between each element of the set of detections D_t of frame t and the elements of the active tracks $B_t = \{b_t^0, b_t^1, ...\}$. If the IoU between a detection and the bounding boxes of all active tracks is smaller than a threshold, a new track is initialised for this detection and is added to the active tracks. (Figure adapted from Bergmann et al., 2019). ©IPI, TiHo Hannover.

- Bounding box initialisation (blue arrows in Figure 1): a new trajectory of an object i that is not yet contained in the list of objects $\{b_t^0, ..., b_t^{K_{t-1}-1}\}$ resulting from bounding box regression can be initialised using the list of detections D_t for the current frame t, assuming that there is a detection $d_i \in D_t$ representing object i. A new trajectory is initialised for that object if the IoU of d_i with any of the $\{b_t^0, ..., b_t^{K_{t-1}-1}\}$ is smaller than a threshold $\lambda_{new} = 0.3$, leading to $B_t = \{b_t^0, ..., b_t^{K_{t-1}-1}, d_i\} =: \{b_t^0, ..., b_t^{K_t-1}\}$.

A trajectory is deactivated, if the IoU between two objects in B_t is larger than a threshold $\lambda_{active} = 0.6$. which means that the object with the smaller classification score is occluded by the other object. Alternatively, a trajectory is also deactivated if the classification score s_k^t of any object k in frame t resulting from Faster R-CNN is below a threshold $\lambda_{score} = 0.5$, which means that the object has left the frame or is occluded by another part of the scene.

3.3 Animal Re-Identification

In tracking, especially in the context of poultry tracking, occlusions are a common problem. This problem can, for example,

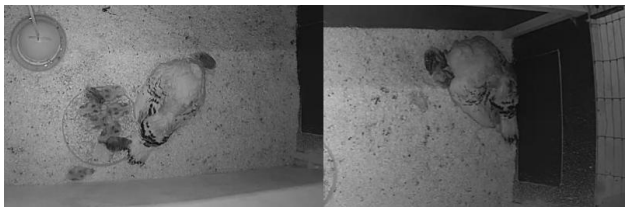

Figure 2. An example of chick-to-chick occlusion while feeding on the left and while sleeping on the right. ©TiHo Hannover

be seen when all the animals group together for food or sleeping as depicted in Figure 2.

To re-identify the animals after occlusions, we propose the use of the MuDeep re-identification model (Qian et al., 2017). The MuDeep model is capable of detecting subtle differences between objects and thus, it is suitable to distinguish very similarly-looking animals. This model requires two object images as input and has two branches, one for processing each image. Each branch consists of the following components: tied convolutional layers, multi-scale stream layers, saliency-based learning fusion layers and a verification subnetwork, which is made up of fully connected layers. After each convolutional and fully connected layer, a batch normalisation and a Rectified Linear Unit activation (Agarap, 2018) are used. The weights are shared between the two branches, resulting in a Siamese network structure. An overview of the network is shown in Figure 3.

The following steps are carried out by the MuDeep model to check whether two images belong to the same object. In the first step, two image extracts each containing an object are pre-processed by the two tied convolutional layers. The generated feature maps are passed to the second step, the multi-scale stream layers. The purpose of the multi-scale layers is to extract high-level features. In these layers, the data stream is down sampled with four different sized convolution masks. In the third step, a saliency-based learning fusion layer is used to combine the output of the multi-scale stream layer and emphasise the channels with highly discriminative high-level features. Such features may, for example, be associated with the head of animals or feathers with a unique colour. This layer is connected to a fully connected layer that takes the high-level features and delivers a feature vector with a lower dimensionality being the outpt of the verification network. We use the output of the verification subnetwork, the last step in the MuDeep model, to decide whether the input image-pair belongs to the same animal. To allow the re-identification of animals that have not been seen by the detector for multiple frames, we save deactivated tracks for 50 frames such as in the original approach of Bergmann et al. (2019). Then, the MuDeep model is used to compare the deactivated with the newly initialised tracks and connects them, if the associated animal can be re-identified, meaning that both tracks belong to the same animal.

4. DATASET

For the evaluation of the method presented in this work, a novel dataset was captured. The dataset was recorded by TiHo Hannover (Tierärztliche Hochschule Hannover) over a period of 48 days. This dataset is composed of a total of seven videos that were acquired during different times of the day and with varying lighting conditions. The dataset comprises grey and RGB

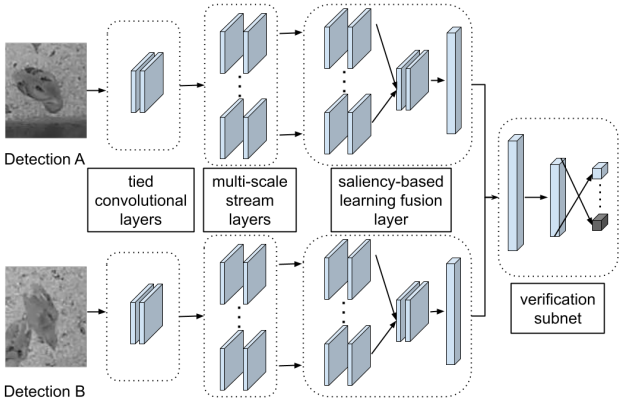

Figure 3. Overview of the MuDeep architecture.

Figure 4. An example of the scene recorded by the used video sequences. The red arrow shows the adult turkey, blue arrows show the chicks. The yellow rectangle shows the main pen, the orange rectangles show the adjacent pens. ©IPI, TiHo Hannover

video sequences. Each video sequence is recorded via a stationary BERGHOCH 8MP Aussen I Basic camera, mounted over a pen observing the scene from a nadir view. The pen dimensions are 2.6 x 1.0 meters. In each sequence, one pen is fully visible, containing various chicks and an adult hen. Note that in this work, we are focusing on tracking the chicks only. The chicks in the different recordings are of different ages and thus of different sizes. We split the dataset accordingly into three sections: small, medium and big. In addition to the fully visible pen, several partially visible pens can be seen at the boundary of the images (see Figure 4). To provide a reference, all the data was manually annotated, using axis-aligned bounding boxes for all chicks in all frames. Those boxes are set such that they are the smallest possible boxes containing the chicks completely. The bounding boxes are defined by the coordinates of the top left and bottom right corners.

The video recordings were acquired at 30 frames per second with a resolution of 1280 x 720 pixels. The recordings are split into three different chick sizes and two camera colour settings (see Table 1). While Figure 5 shows examples of RGB and grey value images, Figure 6 shows the chicks of different sizes in the different stages of their growth. In total, the dataset consists of seven annotated video sequences, of which five have a length between 1800 and 1850 frames, one video sequence has 2000 frames and one has 4000 frames.

ID	#Frames	Chick size	Colour setting	Use
1a	1000	Small	RGB	train
1b	1000	Small	RGB	**val**
1c	2000	Small	RGB	**test**
2	1850	Small	RGB	train
3	1815	Small	RGB and Grey	**test**
4a	900	Medium	Grey	train
4b	900	Medium	Grey	**val**
5	1808	Medium	RGB	train
6a	1000	Big	Grey	train
6b	1000	Big	Grey	**test**
7	1819	Big	RGB	train

Table 1. Description of our novel dataset. ID: the number indicates the ID of a video sequence, while the letters denote that a sequence has been split. For example, the subsets 1a and 1b are both part of video sequence 1, where 1a contains frames 0 to 999 and 1b the frames 1000 to 1999. #Frames: number of frames in the video sequence. Size: the growth status of the chicks in the video sequence. Colour setting: colour settings used while recording (RGB or grey). Use: indicate whether a video sequence is used for training, validation or testing.

Figure 5. Example of the different colour settings used. RGB video sequence on the left, grey video sequence on the right. ©TiHo Hannover

Figure 6. Different stages of chick growth. From left to right: small, medium, big. ©TiHo Hannover

5. EXPERIMENTS

In this section, we evaluate the method presented in this work using the dataset introduced in Section 4. To that extent, we will use the split shown in Table 1 to create the training, validation and test sets. For the training of the Faster R-CNN model, we use a maximum of 30 epochs with early stopping to avoid overfitting. Thus, the model with the best mean average precision score on the validation set is used for testing. The model is trained with a batch size of 4 and a learning rate of 0.001. The MuDeep model is trained with a maximum of 40 epochs, a batch size of 32 and a learning rate of 0.0001.

5.1 Metrics

Three metrics are used for evaluating the detection model, namely Precision, Recall and Average Precision (AP). Precision is the number of true positives divided by the number of true positives plus the number of false positives, while Recall is the number of true positives divided by the number of true positives plus the number of false negatives. Average Precision is the harmonic mean of Precision and Recall.

For the quantitative evaluation of the tracking approach, the following metrics, which are common in the tracking domain, are used: IDF1 (Ristani et al., 2016), MOTA and MOTP (Bernardin and Stiefelhagen, 2008). The combination of these three metrics allows, on the one hand, to evaluate the results in terms of the correctness of the object IDs and thus the consistency of the trajectories (IDF1), as well as the accuracy of the determined position (MOTP). On the other hand, this combination allows us to put the tracking results obtained into the context of other work in terms of classification accuracy (MOTA). IDF1 combines ID precision (IDP) and ID recall (IDR) into a single value by using the harmonic mean, using the definitions of precision and recall given above. MOTP represents the average of all IoU errors of the chick detections. MOTA, on the other hand, combines three different error metrics, namely the number of ID switches, false positives and false negatives in one score. Summing up these three metrics and dividing the sum by the total number of objects present in all frames, gives us the total error rate E_{tot}. MOTA is then defined as $MOTA = 1 - E_{tot}$.

5.2 Augmentations

While the footage of pedestrians used for tracking is often captured by cameras at street level with the optical axis parallel to the ground level, cameras used to track the chicks are mounted below the barn ceiling and result in nadir or oblique images. This gives us the possibility to extend the used augmentation setting for pedestrian tracking in (Bergmann et al., 2019), where only random flipping along the vertical axis is applied. In contrast to Bergmann et al. (2019), we suggest to add Gaussian noise, to augment the brightness and to randomly flip along the horizontal axis. The use of the added Gaussian noise is motivated by the assumption that very fast motions of the chicks result in motion blur. We apply Gaussian noise with a zero mean and a standard deviation of $\sigma = 1.7$. The brightness augmentations are motivated by the observation that the pen is unevenly illuminated. Accordingly, brightness augmentations are supposed to simulate the situations in which the chicks are in shadow. For this purpose, a brightness factor is randomly drawn from the interval $[0.5, 1.5]$ and each channel of the image is multiplied by this factor. If a colour value exceeds the upper or lower bound of possible values after applying the brightness factor, this value is set to the respective bound. By randomly applying flipping along the vertical axis, we exploit that the chicks are observed in a top view. An example of the augmentation strategies applied can be seen in Figure 7.

5.3 Experiments

To define a baseline for tracking the chicks and to obtain a better understanding of the influence of the MuDeep re-identification model on the tracking results in the context of poultry tracking, we carry out multiple experiments. Those experiments can be split into two groups: detection experiments and re-identification experiments. In the first group of experiments, i.e., the detection experiments, we train the Faster R-CNN model with two different data augmentation methods. For the first experiment, we used the same augmentation that has been used in the tracktor approach for pedestrian tracking, namely random vertical flip. We call this experiment the baseline-detection test. In the second experiment, we apply the extended augmentation strategy mentioned above to the dataset. We call this experiment the extended-detection test. In the second group of experiments, i.e., the re-identification experiments, we apply both the baseline-detector and the extended-detector to obtain

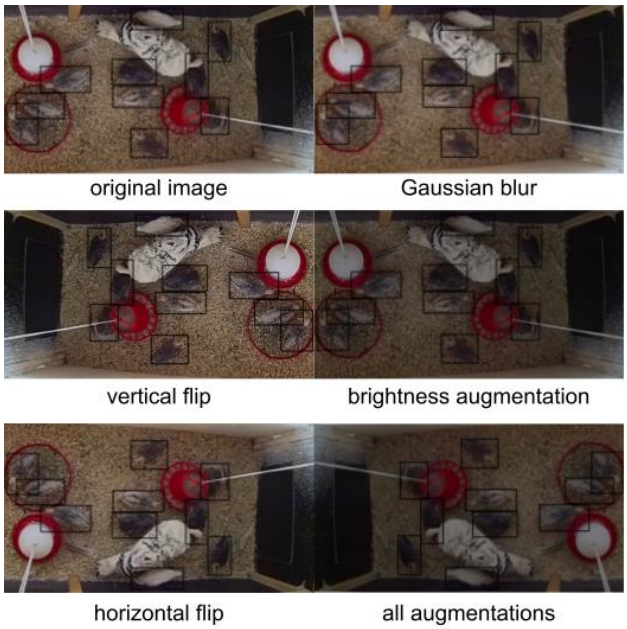

Figure 7. Example of the different augmentation strategies used.

Augmentation	Re-identification model	Experiment ID
baseline	none	baseline-none
	ResNet-50	baseline-ResNet-50
	MuDeep	baseline-MuDeep
extended	none	extended-none
	ResNet-50	extended-ResNet-50
	MuDeep	extended-MuDeep

Table 2. Overview of the experiments carried out (for details see text).

detections in the context of Tracktor. Furthermore, Tracktor is used with different re-identification models, i.e., without any re-identification model (denoted as none), with the Siamese re-identification model based on ResNet-50 (denoted as ResNet-50), which has also been used in the tracktor approach for pedestrian tracking, and with MuDeep re-identification (denoted as MuDeep). We combine each detector with each of the re-identification models in our experiments, whereas an overview of the experiments is given in Table 2.

6. RESULTS

For testing, we use the three different video sequences 1c, 3 and 6b. The chicks and the barn in sequences 1c and 6b have been seen by the model in the training and validation phase, respectively, where different frames have been used there, i.e., the disjoint sequences 1a, 1b and 6a. On the other hand, the chicks and the barn in sequence 3 have neither been seen during training nor while validating the detection model.

6.1 Detection Results

Comparing the results of the baseline detector and the extended detector shown in Table 3, it can be seen that the Average Precision (AP) of the extended model is slightly higher for both sequences 1c and 6b, which can be explained by the fact that similar sequences have been used in the training and validation of the detector. In contrast, the results for sequence 3

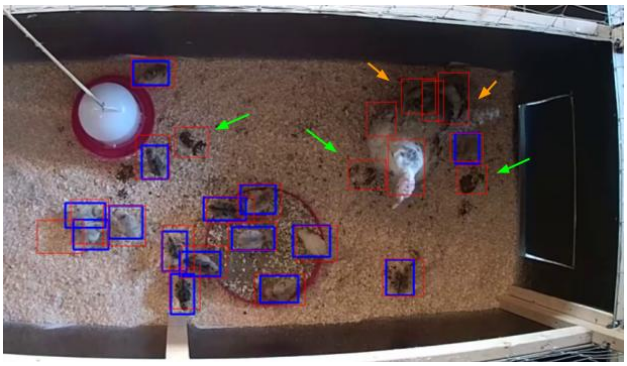

Figure 8. An example of the detections in sequence 3. Green arrows indicate dirt on the floor. Orange arrows indicate black feathers on the tail of the chicken. ©IPI, TiHo Hannover

ID	Augmentation	AP	Precision	Recall	FP
1c	baseline	78.2%	85.5%	91.4%	2321
	extended	79%	86.3%	91.5%	2178
3	baseline	29.1%	55.4%	49.6%	10846
	extended	28.8%	54.6%	52.5%	11866
6b	baseline	93.9%	95.7%	97.8%	463
	extended	95.5%	96.4%	98.9%	411
mean	baseline	67.1%	78.9%	79.6%	4543
	extended	67.8%	79.1%	81%	4818

Table 3. Detection results. ID: the ID of the used video sequence. Augmentation: the type of the used augmentation. AP: average precision score. FP: number of false positive detections.

are slightly worse for the extended version, which is probably caused by its differences to the sequences used for training and validation. An example of detections in sequence 3 can be seen in Figure 8. It shows that the condition of the barn namely the dirt on the floor cause false-positive detections. The adult turkey causes false-positive detections too, due to its black tail feathers. In the training sequences, the adult turkey has fewer of these black feathers in the tail area and the barn is cleaner with no visible dirt on the floor. The use of extended augmentation increased the AP in sequences 1c and 6b by 0,8% and 1,6%, respectively. In sequence 3 an increase in Recall by 2,9% coupled with a decrease of the Precision by 0,8% can be seen which means more true-positive and false-positive detections are made. Overall, the detection model with extended augmentations shows a slight improvement compared to the detection model with the baseline augmentation, while testing on data that is similar to the training and validation sets.

6.2 Tracking Results

For the six different tracking experiments, the results corresponding to the IDF1, MOTA and MOTP metrics are given for each of the test sequences 1c, 3 and 6b in Table 4. These results show that using the detection model with the extended augmentation strategy increases the IDF1 score for tracking for all three re-identification variants: Tracking without a re-identification model, tracking with ResNet-50-based re-identification and tracking with MuDeep are improved by 2,4%, 2,6% and 0,8% in IDF1, respectively. Also, for the same detection model, the results show an increase in the IDF1 score from not using any re-identification (*extended-None*) to the use of ResNet-50 (*extended-ResNet-50*) and MuDeep (*extended-*

Experiment ID	Sequence ID	IDF1↑	MOTA↑	MOTP↓
baseline-None	1c	58.5%	78%	0.23
	3	26.5%	9%	0.36
	6b	88.1%	95.5%	0.14
	mean	57.7%	60.8%	**0.2**
extended-None	1c	63.7%	81.2%	0.24
	3	23.8%	8.9%	0.37
	6b	92.8%	97.3%	0.14
	mean	60.1%	62.5%	0.21
baseline-ResNet-50	1c	64.3%	78.2%	0.23
	3	27.7%	9.4%	0.36
	6b	94.7%	95.7%	0.14
	mean	62.2%	61.1%	**0.2**
extended-ResNet-50	1c	69.6%	81.5%	0.24
	3	26.8%	9.3%	0.37
	6b	98%	97.4%	0.14
	mean	64.8%	**62.7**%	0.21
baseline-MuDeep	1c	69%	78.2%	0.23
	3	28.3%	9.4%	0.36
	6b	96%	95.7%	0.14
	mean	64.4%	61.1%	**0.2**
extended-MuDeep	1c	69.2%	81.5%	0.24
	3	28.4%	9.3%	0.37
	6b	98%	97.4%	0.14
	mean	**65.2**%	**62.7**%	0.21

Table 4. Results of all the tracking experiments. For each experiment, we report the scores of each video sequence. The last row of each experiment shows the average scores of all the three video sequences.

MuDeep). The detection model with the extended augmentation in combination with the MuDeep re-identification delivers the best IDF1 score of 65.2% on average.

The MOTA score, similarly to IDF1, shows an improvement when using the detection model with the extended augmentation. Unlike IDF1, there are no differences between ResNet-50 and MuDeep for re-identification; both *extended-ResNet-50* and *extended-MuDeep* achieve an average MOTA of 62.7%. Compared to MOTA, IDF1 is better at expressing the consistency of ID matching, by measuring how long the identification is correct (Huang et al., 2020), which means that the use of MuDeep improves the ID matching of the tracked chicks. The reason why the tracking results of Sequence 3 are much worse than the results of the other two test sequences, i.e., around 30% worse in IDF1 and around 70% worse in MOTA compared to 1c, is the high number of false-positive detections. Those false-positive detections create false-positive tracks, which decreases the IDF1 and MOTA score of Sequence 3.

The MOTP metric only varies slightly comparing the corresponding experiments using the two different augmentation strategies. Furthermore, no changes between the results of the different re-identification models while using the same detector variant can be observed, which is reasonable, because MOTP measures the detection precision error and thus, it measures the quality of the detection model output and not the tracking model output.

7. CONCLUSIONS AND FUTURE WORK

It is of significant interest to track animals to analyse their behaviour and thus, to improve their welfare. In this paper, we presented an approach based on Tracktor to track poultry; Tracktor does not use a motion-model for tracking. To improve the detection step we extended the used augmentation strategy. Additionally, we use a multi-scale model to improve the re-identification of detected chicks that have been temporarily occluded. The results showed improvements in the IDF1 and MOTA metrics while using the detection model with the extended augmentation in combination with the MuDeep re-identification model compared to the results delivered by the original tracktor approach without any tracking extensions.

It is a question of future research to investigate the influence of different augmentation methods on the detection and tracking models in more detail, for example, in form of a more differentiated analysis of the improvements resulting from each individual augmentation method, and based on that, the introduction of new augmentation methods. In addition, we believe that by increasing the variety of the used dataset, like the use of sequences from barns with different conditions and with more samples from different growth statuses of the chicks, we can further improve the tracking performance.

REFERENCES

Agarap, A. F., 2018. Deep Learning Using Rectified Linear Units (ReLU). *CoRR*, abs/1803.08375. http://arxiv.org/abs/1803.08375.

Ali, S., Shah, M., 2008. *Floor Fields for Tracking in High Density Crowd Scenes*. Springer Berlin Heidelberg.

Aydin, A., 2017. Using 3D vision camera system to automatically assess the level of inactivity in broiler chickens. *Computers and Electronics in Agriculture*, 135, 4-10.

Bergamini, L., Pini, S., Simoni, A., Vezzani, R., Calderara, S., D'Eath, R. B., Fisher, R. B., 2021. Extracting accurate long-term behavior changes from a large pig dataset. *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 4, SciTePress, 524–533.

Bergmann, P., Meinhardt, T., Leal-Taixé, L., 2019. Tracking Without Bells And Whistles. *CoRR*, abs/1903.05625. http://arxiv.org/abs/1903.05625.

Bernardin, K., Stiefelhagen, R., 2008. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP Journal on Image and Video Processing*, 2008(1), 1-10.

Bolme, D. S., Beveridge, J. R., Draper, B. A., Lui, Y. M., 2010. Visual object tracking using adaptive correlation filters. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2544–2550.

Chen, L., Ai, H., Zhuang, Z., Shang, C., 2018. Real-Time Multiple People Tracking With Deeply Learned Candidate Selection And Person Re-Identification. *2018 IEEE International Conference on Multimedia and Expo (ICME)*, 1-6.

Chien, Y.-R., Chen, Y.-X., 2018. An RFID-Based Smart Nest Box: An Experimental Study of Laying Performance and Behavior of Individual Hens. *Sensors (Basel, Switzerland)*, 18.

Colantonio, S., Benvenuti, M., Di Bono, M., Pieri, G., Salvetti, O., 2007. Object tracking in a stereo and infrared vision system. *Infrared Physics & Technology*, 49(3), 266-271.

Colles, F. M., Cain, R. J., Nickson, T., Smith, A. L., Roberts, S. J., Maiden, M. C. J., Lunn, D., Dawkins, M. S., 2016. Monitoring Chicken Flock Behaviour Provides Early Warning Of Infection By Human Pathogen Campylobacter. *Proceedings of the Royal Society B: Biological Sciences*, 283(1822), 20152323. http://dx.doi.org/10.1098/rspb.2015.2323.

Dalal, N., Triggs, B., 2005. Histograms Of Oriented Gradients For Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 886-893 vol. 1.

Girshick, R., 2015. Fast r-cnn. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1440–1448.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Helbing, D., Molnár, P., 1995. Social Force Model For Pedestrian Dynamics. *Phys. Rev. E*, 51, 4282–4286. https://link.aps.org/doi/10.1103/PhysRevE.51.4282.

Huang, Y., Zhu, F., Zeng, Z., Qiu, X., Shen, Y., Wu, J., 2020. Sqe: A self quality evaluation metric for parameters optimization in multi-object tracking. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8303–8311.

Kashiha, M., Bahr, C., Ott, S., Moons, C. P., Niewold, T. A., Ödberg, F., Berckmans, D., 2013. Automatic Identification Of Marked Pigs In A Pen Using Image Pattern Recognition. *Computers and Electronics in Agriculture*, 93, 111–120.

Klinger, T., Rottensteiner, F., Heipke, C., 2014. A Dynamic Bayes Network for visual Pedestrian Tracking. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-3, 145–150. https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-3/145/2014/.

Kuhn, H. W., 1955. The Hungarian Method For The Assignment Problem. *Naval Research Logistics Quarterly*, 2(1-2), 83-97. https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109.

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(3), 617–625.

Li, W., Zhu, X., Gong, S., 2018. Harmonious attention network for person re-identification. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2285–2294.

Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., Kim, T.-K., 2022. Multiple Object Tracking: A Literature Review. *CoRR*.

Neethirajan, S., 2022. ChickTrack – A Quantitative Tracking Tool For Measuring Chicken Activity. *Measurement*, 191, 110819.

Nguyen, U., Rottensteiner, F., Heipke, C., 2019. Confidence-aware Pedestrian Tracking Using a Stereo Camera. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4(2/W5), 53–60.

Ommer, B., Mader, T., Buhmann, J. M., 2009. Seeing The Objects Behind The Dots: Recognition In Videos From A Moving Camera. *International Journal of Computer Vision*, 83(1), 57–71. http://dx.doi.org/10.1007/s11263-009-0211-7.

Qian, X., Fu, Y., Jiang, Y.-G., Xiang, T., Xue, X., 2017. Multi-scale Deep Learning Architectures for Person Re-identification. *2017 IEEE International Conference on Computer Vision (ICCV)*, 5409-5418.

Redmon, J., Farhadi, A., 2018. YOLOv3: An Incremental Improvement. *CoRR*, abs/1804.02767.

Ren, S., He, K., Girshick, R. B., Sun, J., 2015. Faster R-CNN: Towards Real-Time Object Detection With Region Proposal Networks. *CoRR*, abs/1506.01497. http://arxiv.org/abs/1506.01497.

Ristani, E., Solera, F., Zou, R. S., Cucchiara, R., Tomasi, C., 2016. Performance Measures And A Data Set For Multi-Target, Multi-Camera Tracking. *CoRR*.

Ristani, E., Tomasi, C., 2018. Features for multi-target multi-camera tracking and re-identification. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6036–6046.

Shafique, K., Lee, M. W., Haering, N., 2008. A rank constrained continuous formulation of multi-frame multi-target tracking problem. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8.

Tang, S., Andriluka, M., Andres, B., Schiele, B., 2017. Multiple People Tracking by Lifted Multicut and Person Re-identification. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3701-3710.

Wurtz, K., Camerlink, I., D'Eath, R. B., Fernández, A. P., Norton, T., Steibel, J., Siegford, J., 2019. Recording Behaviour Of Indoor-Housed Farm Animals Automatically Using Machine Vision Technology: A Systematic Review. *PLOS ONE*, 14(12).

Yu, Q., Chang, X., Song, Y.-Z., Xiang, T., Hospedales, T. M., 2018. The Devil Is In The Middle: Exploiting Mid-Level Representations For Cross-Domain Instance Matching. *CoRR*.

Zhang, L., Gray, H., Ye, X., Collins, L., Allinson, N., 2019. Automatic Individual Pig Detection and Tracking in Pig Farms. *Sensors*, 19(5), 1188. http://dx.doi.org/10.3390/s19051188.

Zhao, Z.-Q., Zheng, P., Xu, S.-T., Wu, X., 2019. Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11), 3212-3232.