

Article

Identifying and Counting Avian Blood Cells in Whole Slide Images via Deep Learning

Markus Vogelbacher ^{1,*}, Finja Strehmann ², Hicham Bellafkir ¹, Markus Mühling ¹, Nikolaus Korfhage ¹, Daniel Schneider ¹, Sascha Rösner ², Dana G. Schabo ², Nina Farwig ² and Bernd Freisleben ¹

¹ Department of Mathematics and Computer Science, University of Marburg, Hans-Meerwein-Str. 6, 35043 Marburg, Germany

² Department of Biology, University of Marburg, Karl-von-Frisch-Str. 8, 35043 Marburg, Germany; finja.strehmann@biologie.uni-marburg.de (F.S.); sascha.roesner@biologie.uni-marburg.de (S.R.)

* Correspondence: vogelbacher@informatik.uni-marburg.de

Simple Summary: Avian blood analysis is crucial for understanding the health of birds. Currently, avian blood cells are often counted manually in microscopic images, which is time-consuming, expensive, and prone to errors. In this article, we present a novel deep learning approach to automate the quantification of different types of avian red and white blood cells in whole slide images of avian blood smears. Our approach supports ornithologists in terms of hematological data acquisition, accelerates avian blood analysis, and achieves high accuracy in counting different types of avian blood cells.

Abstract: Avian blood analysis is a fundamental method for investigating a wide range of topics concerning individual birds and populations of birds. Determining precise blood cell counts helps researchers gain insights into the health condition of birds. For example, the ratio of heterophils to lymphocytes (H/L ratio) is a well-established index for comparing relative stress load. However, such measurements are currently often obtained manually by human experts. In this article, we present a novel approach to automatically quantify avian red and white blood cells in whole slide images. Our approach is based on two deep neural network models. The first model determines image regions that are suitable for counting blood cells, and the second model is an instance segmentation model that detects the cells in the determined image regions. The region selection model achieves up to 97.3% in terms of F1 score (i.e., the harmonic mean of precision and recall), and the instance segmentation model achieves up to 90.7% in terms of mean average precision. Our approach helps ornithologists acquire hematological data from avian blood smears more precisely and efficiently.

Keywords: cell segmentation; bird blood analysis; microscopy images; blood smear images; object detection; ornithology



Citation: Vogelbacher, M.; Strehmann, F.; Bellafkir, H.; Mühling, M.; Korfhage, N.; Schneider, D.; Rösner, S.; Schabo, D.G.; Farwig, N.; Freisleben, B. Identifying and Counting Avian Blood Cells in Whole Slide Images via Deep Learning. *Birds* **2024**, *5*, 48–66. <https://doi.org/10.3390/birds5010004>

Academic Editor: Jukka Jokimäki

Received: 21 December 2023

Revised: 12 January 2024

Accepted: 15 January 2024

Published: 19 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Automated visual and acoustic monitoring methods for birds can provide information about the presence and the number of bird species [1] or individuals [2] in certain areas, but analyzing the physiological conditions of individual birds allows us to understand potential causes of negative population trends. For example, measuring the physiological stress of birds can serve as a valuable early warning indicator for conservation efforts. The physiological conditions and the stress of birds can be determined in several ways, e.g., by assessing the body weight or the fat and muscle scores in migratory birds [3,4]. Other frequently used methods are investigating the parasite loads, measuring the heart rates, and measuring the levels of circulating stress hormones, such as corticosterone [5–9]. Depending on the research questions studied, these methods can be a good choice for assessing long-term stress or the investment in immunity.

The method investigated in this article comprises analyzing blood smears and counting blood cells [10]. Not only are white blood cells, i.e., leukocytes, an important part of the immune system of vertebrates such as mammals or birds, but also the composition of leukocytes is known to change in response to elevated stress hormones (glucocorticoids) and can, therefore, be used to assess stress levels [10]. In particular, the ratio of heterophils to lymphocytes (H/L ratio) is considered to be a well-established stress index for assessing long-term stress in birds [10,11]. Since the H/L ratio changes only 30 to 60 min after the onset of an acute stress event, it is possible to measure stress without mirroring the influence of the capture event [12]. It is also possible to calculate the leukocyte concentration (leukocytes per 10,000 erythrocytes) or the concentration of specific leukocyte cell types for gaining an understanding of the current health status of a bird and the investment in immunity [13–15].

Leukocyte counts are quite cost-effective since they do not require complex laboratory techniques. However, evaluation under the microscope often requires manual interpretation by human experts, is time-consuming, and can only assess small portions of the entire smear. Typically, leukocytes are counted until 100 leukocytes are reached [16]. Consequently, the counted values and subsequent ratios are not always reproducible, and the result depends on the section counted. Furthermore, the method is prone to observer errors. Therefore, there is an urgent need for automated methods to perform leukocyte counts in avian blood smears.

Bird and human blood cell analysis have some aspects in common. The counted leukocytes are similar: lymphocytes, eosinophils, basophils, and monocytes can be found in mammalian as well as avian blood. However, there are some significant differences that make the automated counting of avian blood cells more difficult [17]. The neutrophils in human blood are equivalent to heterophils in birds. One of the main differences between bird and human blood, however, is the presence of nuclei in bird erythrocytes (i.e., red blood cells) and thrombocytes, whereas there is no nucleus in mammalian erythrocytes and thrombocytes [17]. The presence of a nucleus in erythrocytes makes the cell identification process more complicated since lysed and ruptured erythrocytes can be mistaken for other cell types. Lastly, during ornithological field studies, the bird blood samples are usually not taken in a sterile environment, leading to dirt contaminating the smears. Such contaminants and stain remnants can further lead to confusion. Because of these differences from human blood and the associated challenges, it is necessary to develop dedicated solutions instead of relying on existing machine learning approaches for human blood analysis to automatically analyze bird blood samples.

A solid understanding of the different leukocyte types is necessary when analyzing avian blood samples since some are quite similar to each other. Figure 1 shows examples of each blood cell type as well as two challenging anomalies, i.e., stain remnants and lysed cells (Figure 1d) as well as ruptured cells (Figure 1h). Lymphocytes appear small, round, and blue in blood smears and are most common in passerine birds. Their nuclei usually take up more than 90% of the cell (see Figure 1b). Heterophils can be identified by their lobed cell nuclei and rod-shaped granules in the cytoplasm, as shown in Figure 1e. In birds, heterophils and lymphocytes make up approximately 80% of the leukocytes [18]. Eosinophils are similar to heterophils but have round granules (see Figure 1c). Basophils can be recognized by their purple-staining granules, as shown in Figure 1f, but they are rare to find. Monocytes are larger cells that can be confused with lymphocytes, but their nucleus often has a kidney-shaped appearance and takes up only up to 75% of the cell (see Figure 1g) [19]. Additionally, it is important to be aware of possible variations regarding the morphology and staining characteristics of these cell types between different avian species, which may affect their identification and interpretation.

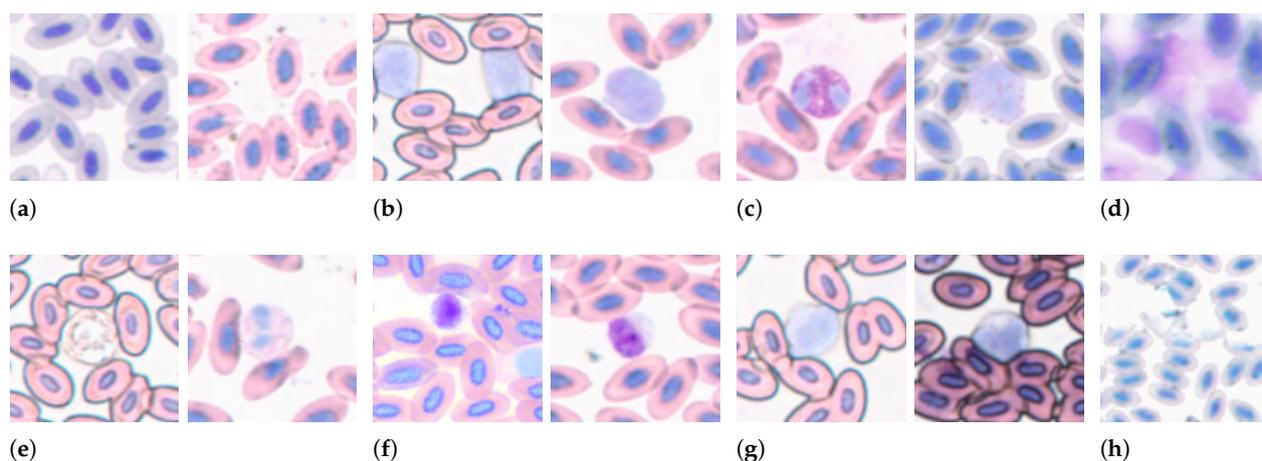


Figure 1. Examples of different cell types: (a) erythrocytes of a blue tit (*Cyanistes caeruleus*) and a Eurasian blackcap (*Sylvia atricapilla*); (b) lymphocytes of a common blackbird (*Turdus merula*) and a common buzzard (*Buteo buteo*); (c) eosinophils of a common buzzard and a European robin (*Erithacus rubecula*); (d) stain remnants and lysed cells in the blood sample of a black woodpecker (*Dryocopus martius*); (e) heterophils of a common blackbird and a common buzzard; (f) basophils of a common blackbird and a common buzzard; (g) monocytes of a common blackbird; and a European robin and (h) ruptured cells in a blood sample of a Eurasian blue tit. The blood smears were stained with Giemsa and scanned at 40× magnification.

Avian blood counts are still mostly obtained manually. However, there are several approaches for more systematic, automated ways of counting avian blood cells. For instance, Meechart et al. (2020) [20] developed a simple computer vision algorithm based on Otsu's thresholding method [21] to automatically segment and count erythrocytes in chicken blood samples. Beaufrère et al. (2013) [22] used image cytometry, i.e., the analysis of blood in microscopy images, in combination with the open-source software CellProfiler [23,24] to classify each cell using handcrafted features as well as machine learning algorithms. However, they stated their results were not satisfactory.

Another way of automating avian blood counts is the use of hardware devices for blood analysis. For example, the Abbott Cell-Dyn 3500 hematology analyzer [25] (Abbott, Abbott Park, IL, USA) was used in studies analyzing chicken blood samples [26,27]. The Cell-Dyn 3500 works on whole blood samples and relies on flow cytometry, i.e., the analysis of a stream of cells by a laser stream and electric impedance measurements. The device was standardized for poultry blood.

The CellaVision[®] DC-1 analyzer [28] (CellaVision AB, Lund, Sweden) scans blood smears and pre-classifies erythrocytes as well as leukocytes. In combination with the proprietary CellaVision[®] VET software [29], the device can be used to analyze animal blood, including bird blood. However, the pre-classification results still need to be verified by a human expert. The device has a limited capacity of a single slide and is able to process roughly 10 slides per hour, according to the manufacturer [28]. This throughput does not appear to reduce turnaround times in (human) blood analysis [30]. Yet, in a distributed laboratory network, the device could indeed contribute to reduced turnaround times [31].

In the last decade, deep learning models, in particular convolutional neural networks (CNNs), have become the state of the art in many computer vision tasks, such as image classification, object detection, and semantic segmentation. These deep neural networks are highly suitable for image processing since they can learn complex image features directly from the image data in an end-to-end manner. Apart from their success in natural image processing, they have also contributed to biological and medical imaging tasks, e.g., in cell detection and segmentation [32,33], blood sample diagnostics [34,35], histopathological sample diagnostics [36], such as breast cancer detection [37], and magnetic resonance imaging (MRI) analysis [38].

However, only a few deep learning approaches are available for avian blood cell analysis. For instance, Govind et al. (2018) [39] presented a system for automatically detecting and classifying avian erythrocytes in whole slide images. Initially, they extract optimal areas from the whole slide images for analyzing erythrocytes. In the first step, regions are chosen from low-resolution windows using a quadratic determinant analysis classifier. These optimal areas are then refined at higher resolution using an algorithm based on binary object sizes. This algorithm identifies overlapping cells that need to be split. The actual separation is conducted in a multi-step handcrafted algorithm. Intensity- and texture-based features are used to distinguish between erythrocytes and leukocytes, but the latter are not actually detected. In the final step, all detected erythrocytes, i.e., solitary and separated from clumps, are classified. This is the only part of the approach that relies on deep learning. Each detected cell is cropped and fed to a GoogLeNet deep neural network [40]. The resulting model can classify the detected erythrocytes as mammalian, reptilian, or avian. Furthermore, the model can categorize erythrocytes into one of thirteen species. However, only one of these is a bird species.

Kittichai et al. (2021) [41] used different CNN models to detect infections of an avian malaria parasite (*Plasmodium gallinaceum*) in domestic chickens. Initially, a YOLOv3 [42] deep learning model was used to detect erythrocytes in thin blood smear images. Then, four CNN architectures were employed for the classification of the detected cells to characterize the different avian malaria blood stages.

However, to the best of our knowledge, there is no hardware-independent and publicly available approach for the automated segmentation and classification of avian blood cells, i.e., erythrocytes as well as leukocytes.

In this article, we present a novel deep learning approach for the automated analysis of avian blood smears. It is based on two deep neural networks to automatically quantify avian red and white blood cells in whole slide images, i.e., digital images produced by scanning microscopic glass slides [43]. The first neural network model determines image regions that are suitable for counting blood cells. The second neural network model performs instance segmentation to detect blood cells in the determined image regions. For both models, we investigate different neural network architectures and different backbone networks for feature extraction in cell instance segmentation. We provide an open-source software tool to automate and speed up blood cell counts in avian blood smears. We make the annotated dataset used in our work publicly available, along with the trained neural network models and source code [44]. In this way, we enable ornithologists and other interested researchers to build on our work.

2. Materials and Methods

We present a deep learning approach for automatically identifying and counting avian blood cells. The approach is divided into three main phases. Figure 2 gives an overview of the entire process from acquiring blood smears to automatically emitting a blood cell count for a whole slide image. In the first phase, avian blood samples are acquired and digitized. Next, the resulting images are split into tiles that are uploaded to a web-based annotation tool used by a human expert to thoroughly annotate tiles for both tasks, i.e., tile selection and cell instance segmentation. In the second phase, adequate deep neural networks are trained for both tasks. The deep neural network models assist a human expert during annotation by providing pre-annotations for further labeling. In the third phase, the trained deep neural network models are applied to process whole slide images that were not used during training. These images are split into tiles that are analyzed by the tile selection model. Only tiles approved as countable are forwarded to the instance segmentation model. The final blood cell counts are determined based on the outputs of the instance segmentation model.

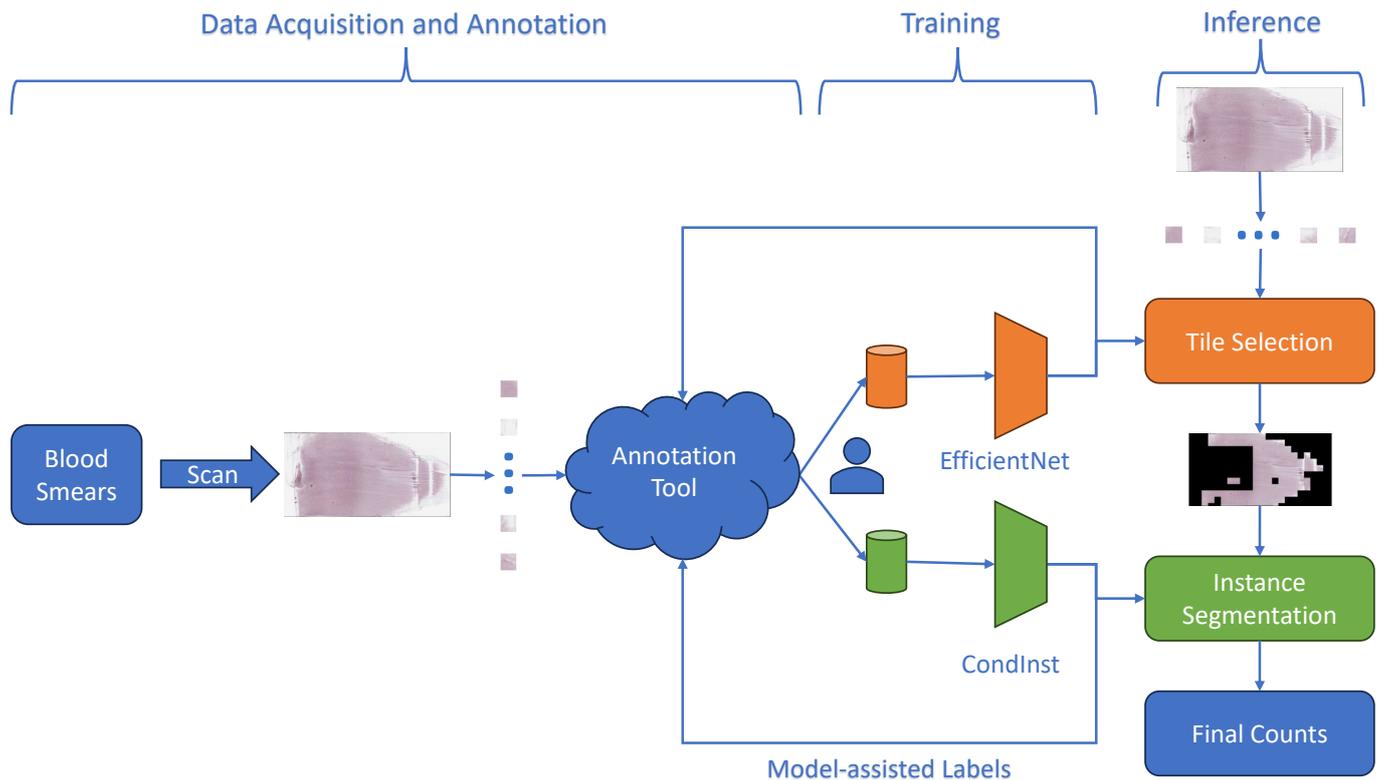


Figure 2. Overview of our approach for counting avian blood cells in whole slide images. Data acquisition: Blood smears are prepared and scanned. The images are then cut into tiles and annotated by a human expert via a web-based annotation tool. Training: The neural network models are trained using annotated data for tile selection and instance segmentation. The models also assist in iteratively annotating images. Inference: An input image is tiled and fed into the tile selection model. Countable tiles are passed to the instance segmentation model before the final counts are determined.

2.1. Data Acquisition and Annotation

To be able to train deep learning models at scale, we used avian blood smear samples from an ornithological field study in the Marburg Open Forest (MOF), a 250-hectare beech-dominated forest in Central Hesse, Germany. The data collection took place over four consecutive years, from 2019 to 2022, during the breeding seasons of the entire forest bird community (29 species of 16 families and 5 orders) between mid-March and August. The birds were captured using mist nets 12 m in length and 2.5 m in height; the mesh size was 16×16 mm. Each bird was marked with a ring for re-capture identification with the necessary permissions obtained from the Heligoland Bird Observatory (Institut für Vogelforschung Heligoland, Germany). A blood sample was taken within 30 min from capture by puncturing the brachial vein and using heparinized capillary tubes, following the animal testing approval granted by the regional council of Giessen, Hesse, Germany (V 54–19 c 20 15 h 01 MR 20/15 Nr. G 10/2019). The blood was then used to create one or two whole blood air dry smears of each bird right after sampling in the field. In the laboratory, the blood smears were fixed in methanol within 24 h and stained with Giemsa within 21 days, following standard protocols [45].

Our work relies on two data sources. We created a first, small dataset consisting of 160 images through manual acquisition. The images were digitized with a Zeiss [46] AxioCam ERc 5s Rev.2 (Carl Zeiss AG, Oberkochen, Germany) in combination with a Zeiss Primo Star microscope (Carl Zeiss AG, Oberkochen, Germany) at $100\times$ magnification with oil immersion and saved in the PNG image format with a resolution of 2560×1920 px. However, this method is not suitable for creating a large dataset since there is a significant manual effort involved, which makes the method very time-consuming.

To create a second, larger dataset, we digitized one or two blood smears per bird during the initial capture and, in recapture cases, we digitized another one or two blood smears of the same bird by scanning the blood smears with a Leica Aperio AT2 Scanner [47] (Leica Biosystems Nussloch GmbH, Nussloch, Germany) at 40× magnification. We selected the highest quality smear of each bird per capture for our analysis. Aperio scanners generate high-resolution images that are stored in the SVS file format, which consists of a series of TIFF images. The first image is always the full-resolution scan. In subsequent layers, it is split into tiles that are decreasing in size with each layer. Overall, we obtained 527 whole slide images from 459 individual birds of 29 species. These images range from 47,807 px to 205,176 px in width and from 35,045 px to 93,827 px in height. To be able to process these huge images with deep learning models, we used OpenSlide [48] to crop them into tiles.

The complete and accurate annotation of the dataset is crucial for training a high-quality deep learning model. To reduce the impact of human errors and eliminate consistency issues in the annotations, we relied on a single human expert for the labeling task. In the following, we describe the annotation process in more detail for both of our tasks, i.e., tile selection and cell detection and segmentation.

There are several criteria for classifying an avian blood smear image tile as either countable or non-countable. Figure 3 shows examples of positive (i.e., countable) and negative (i.e., non-countable) image crops. Cells should be equally distributed, as shown in Figure 3b, without large empty spaces, as shown in Figure 3a. Furthermore, there should be only a few overlapping cells and especially no overlapping nuclei, as contrasted in Figure 3c,d. In general, good image quality is desirable (Figure 3f,e). To be able to train a deep learning model to classify blood smear image tiles as countable or non-countable, a human expert manually selected image tiles and classified them as countable or non-countable. This process led to a dataset consisting of 2288 positive and 2372 negative examples. While it is sufficient for the tile selection task to simply annotate each sample to be countable or not, we fully annotated the dataset for detection with segmentation masks instead of using bounding boxes. Hence, each single cell instance needs to be precisely covered by a mask and tagged with the corresponding class label. Although this annotation method takes even more time, it improves the performance of the final model [49]. Providing exact cell boundaries is particularly beneficial in crowded image regions, where several bounding boxes may overlap.

We used the web platform Labelbox [50] for annotating images of our datasets. The instance segmentation dataset was annotated in an iterative, model-assisted manner. This means that we used the tile selection network to propose regions to be annotated and eventually selected them based on how many rare cells had been detected by an intermediate instance segmentation model. In the very first iteration, we used a superpixel algorithm to generate simple instance masks. In each iteration, we uploaded the corresponding instance segmentation masks to Labelbox to be refined by our human expert. This procedure significantly reduces the time needed to fully annotate an image file with masks and class labels compared to annotating from scratch. Overall, we went through four iterations of labeling. For the annotated cell instances, we established two primary categories: erythrocyte, with only the nucleus annotated, and leukocyte. The latter was further split into five subtypes, namely, lymphocyte, eosinophil, heterophil, basophil, and monocyte. Thrombocytes were not explicitly annotated; they were considered to be part of the background during training. Thus, our trained neural network model can distinguish between non-relevant thrombocytes and other annotated cell types, e.g., erythrocytes. By annotating only the nucleus of each erythrocyte rather than the entire cell including the cytoplasm, we maintained the option to label parasite-infected instances individually in future work. Cells infected with parasites may be annotated by masking the entire cell including the cytoplasm. One erythrocyte can be simultaneously counted as both an erythrocyte and a cell with blood parasite because of the distinct annotation regions.

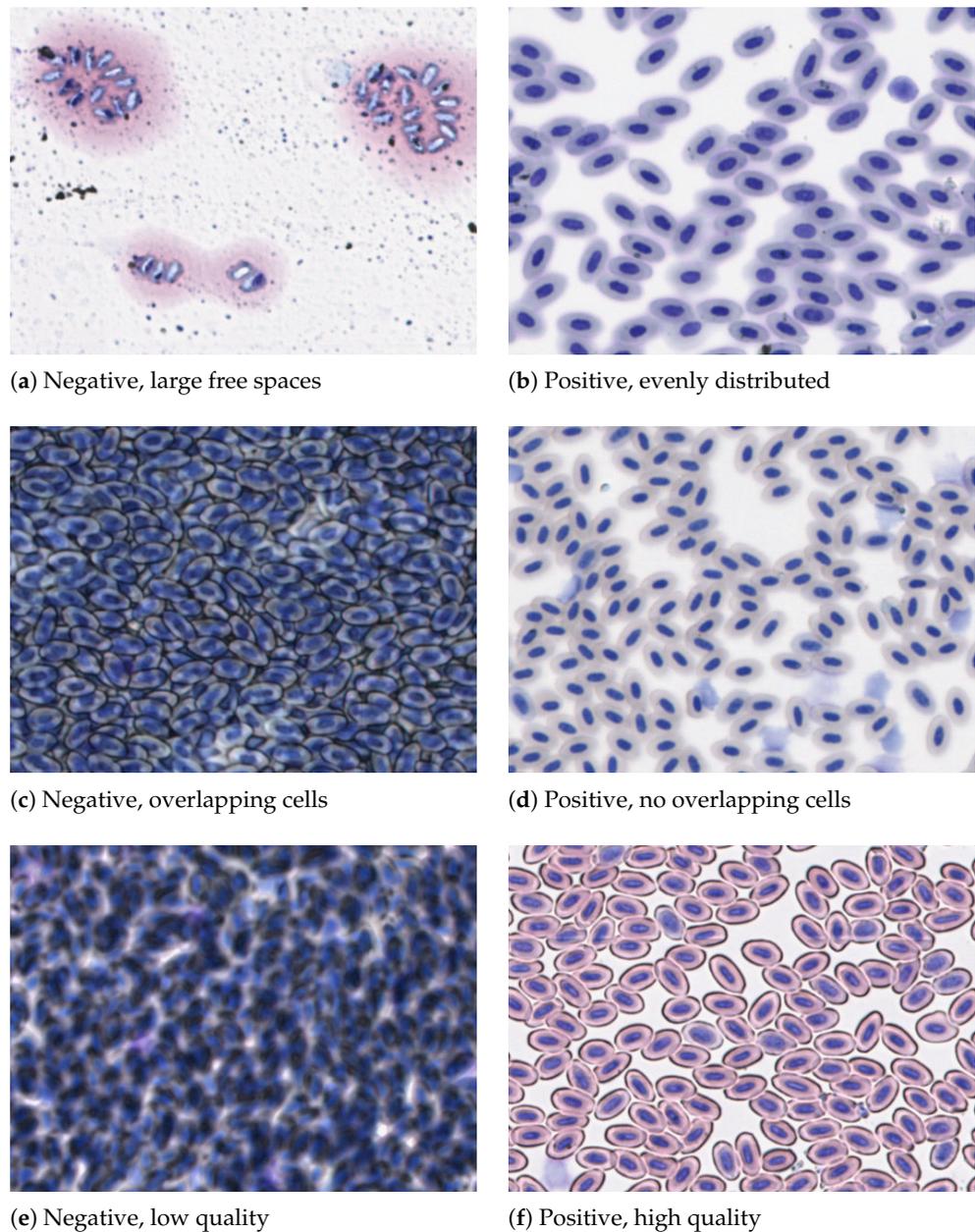


Figure 3. Positive and negative examples for image tiles that are suitable for counting cells. Subfigures (a,c,e) show non-countable examples, while Subfigures (b,d,f) show countable tiles.

Overall, our segmentation dataset consisted of 1810 fully annotated images. As Table 1 shows, the dataset contained 226,781 annotated cell instances that were unevenly spread across 5 taxonomic bird orders, namely, *Accipitriformes*, *Columbiformes*, *Falconiformes*, *Passeriformes*, and *Piciformes*. The orders *Passeriformes* and *Accipitriformes* dominated the dataset. Moreover, with a share of 98% of all cells, erythrocytes were by far the most frequent type of blood cells in our dataset. Among the leukocytes, the subtypes were also not distributed equally. While lymphocytes, eosinophils, and heterophils numbers were between 1000 and 2000 samples, basophils and monocytes were expectedly very rare, as Figure 4 demonstrates. This imbalance is often challenging for machine learning approaches. The annotation of the instance segmentation dataset took our human expert more than 70 h.

Table 1. Summary of annotated bird blood cells across 24 bird species and 1810 blood smear images. Numbers are given for each order of birds occurring in the dataset and for each blood cell type, i.e., erythrocytes as well as five subtypes of leukocytes.

Order	Erythrocyte	Lymphocyte	Eosinophil	Heterophil	Basophil	Monocyte	Overall
<i>Accipitriformes</i>	26,867	254	504	208	28	5	27,866
<i>Columbiformes</i>	329	1	1	5	0	0	336
<i>Falconiformes</i>	245	2	0	0	0	0	247
<i>Passeriformes</i>	192,188	1638	611	815	130	76	195,458
<i>Piciformes</i>	2573	15	9	12	0	0	2609
Σ	222,202	1910	1125	1040	158	81	226,781

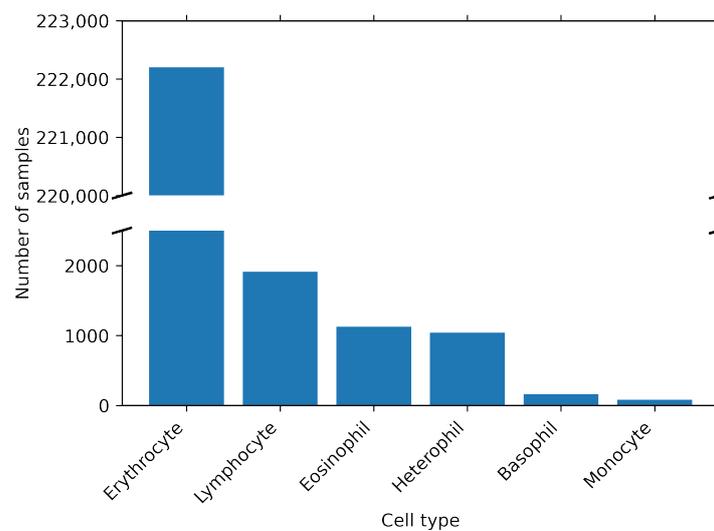


Figure 4. Distribution of annotated bird blood cell types. The plot shows how the annotated cell instances are spread among different cell types, i.e., erythrocytes, lymphocytes, eosinophils, heterophils, basophils, and monocytes.

2.2. Deep Learning Approach

Our novel approach for analyzing whole slide avian blood smear images consists of two stages, i.e., tile selection and instance segmentation. The tile selection process is shown in Figure 5. Initially, we decompose the input whole slide image into tiles. For each tile, we perform a binary classification to ensure that the contained cells fulfill the requirements to be countable. Next, an instance segmentation model is applied to all tiles that are classified as countable. This model detects all cells in the image and classifies each one as either an erythrocyte or as one of the subtypes of leukocytes. Figure 6 illustrates this procedure. Each step is explained in more detail in the following sections.

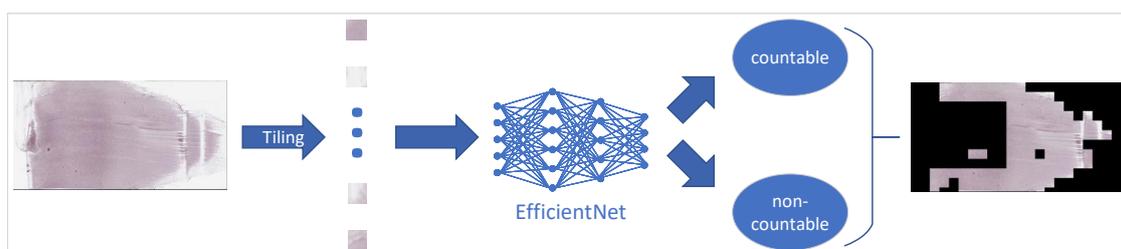


Figure 5. Overview of the tile selection phase. The original image is split up into single tiles. Each tile is fed into the EfficientNet CNN model that classifies it as either countable or non-countable. We finally visualize the results by blacking out all tiles classified as non-countable. For visualization purposes, we aggregated 16×16 tiles to one patch in the depicted output.

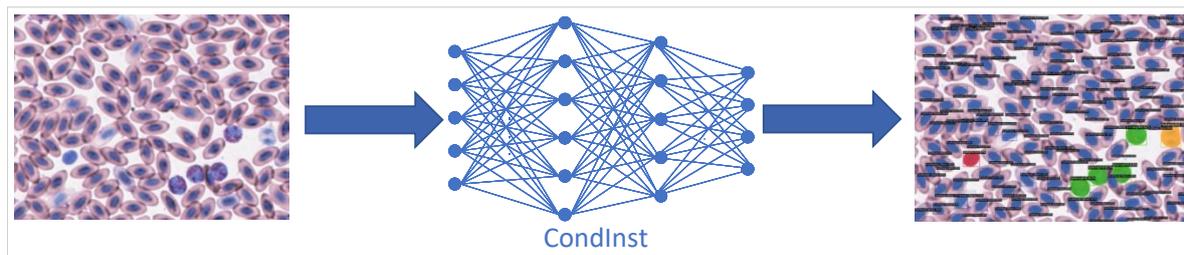


Figure 6. Overview of the cell instance segmentation phase. Countable image tiles serve as inputs to the CondInst instance segmentation model. On the right, we visualize the predictions made by the model. Each color corresponds to one cell type. In the example, the model recognized one lymphocyte (red), one heterophil (orange), four eosinophils (green), and many erythrocytes (blue) reliably.

2.2.1. Tile Selection

For the tile selection model, we used EfficientNet [51] as our architecture. EfficientNet is a family of neural network architectures that has proven to be an excellent choice for image classification. We experimented with the two smallest versions of EfficientNet, namely EfficientNet-B0 and EfficientNet-B1. In a pre-processing step, we randomly applied data augmentation to prevent our model from overfitting the training data. Besides random contrast, random hue, random cropping, and horizontal as well as vertical flipping, these augmentations included elastic transformations that have proven to be very beneficial for cell recognition tasks [32,33]. The input size of our model was 512×384 px. Our training was based on a well-established pre-trained ImageNet model and fine-tuned in two phases using the Adam optimizer [52] and binary cross-entropy loss. By experimenting with different learning rates, we found an initial learning rate of 1×10^{-4} to work best in our case. During the first training phase, we kept the majority of the model parameters fixed. We made an exception for the last layer, where we introduced a new set of weights. This approach ensured that the new layer would not interfere with the pre-trained weights in the rest of the model. In this way, the randomly initialized last layer could adapt to the training data. We trained the network for 20 epochs, i.e., iterating with the whole training dataset 20 times. In the second training phase, we lowered the initial learning rate by a factor of 10 and trained the last 20 layers of the model to ensure that the model would learn useful features for the tile selection task. The loss converged again after up to 30 more epochs. Furthermore, in both phases, the learning rate was reduced whenever the validation accuracy stagnated in order to help the model find the optimal set of weights.

2.2.2. Detection and Segmentation

For the instance segmentation task, we used the CondInst architecture [53], as shown in Figure 6. This neural network is based on the anchor-free object detector FCOS [54] that tackles object detection in a fully convolutional way. To solve object detection or instance segmentation tasks, many approaches rely on anchor boxes and proposals, e.g., Faster R-CNN [55] and most versions of YOLO [56].

For instance, Faster R-CNN generates bounding box proposals based on pre-defined anchor boxes. Anchor boxes of different scales and aspect ratios are placed in each area of the image and are assigned a score based on how likely they are to contain a relevant object. High-scored proposals are resized to a common size and processed in parallel in two different branches of the network, the so-called heads. One refines the proposed bounding boxes, i.e., the box regression head, while the other predicts the corresponding class label, i.e., the classification head.

However, using anchor boxes has several drawbacks. First, deep neural networks based on anchor boxes are sensitive to the choice of hyperparameters. For example, since anchor box scales and aspect ratios are fixed, they cannot easily adapt to new tasks. Furthermore, such networks are computationally inefficient. They produce many negative boxes and have to calculate many Intersection over Union (IoU) values to find the optimal

proposals. The object detector FCOS [54] relies on neither anchor boxes nor proposals. Instead, one of its heads simply predicts a 4D vector regressing the bounding box centered at this location.

While fully convolutional networks have been commonly used for semantic segmentation for many years, the CondInst architecture [53] successfully applies this type of neural network to instance segmentation. To be able to predict instance segmentation masks rather than bounding boxes, CondInst adds a mask branch to FCOS and inserts a so-called controller head, along with the classification and bounding box heads for each location (x, y) . The controller head dynamically generates convolutional filters specifically built for each instance in an image by predicting its parameters. In this way, the mask branch becomes instance aware, i.e., it can predict one segmentation mask for each object instance in the image. This strategy yields very good results for irregular shapes that are challenging to tightly enclose within a bounding box.

To train our model, we used two different kinds of input data. The first part of our training dataset consisted of 138 images with a resolution of 2560×1920 px from the manually acquired data source. Since these images were captured with a magnitude different from the one used for the whole slide images, we needed to choose the crop tile size such that they contained a comparable number of similarly sized cells. This resulted in a tile size of 512×384 px.

In a pre-processing step, we resized each image to 1066×800 px, matching the maximal short-edge input size of CondInst. Furthermore, we applied extensive data augmentation to enrich the dataset and prevent the model from overfitting to the training data. In particular, we applied random horizontal and vertical flipping as well as random adaptations of brightness, contrast, and saturation in an empirically chosen range of 0.7 to 1.3, respectively. Additionally, we applied random elastic transformations that are explicitly useful for cell segmentation tasks [32,33] since they produce realistic alternations of the cells by using an $x \times x$ grid overlay and distorting it with random displacement vectors. We empirically chose $x \in \{6, 7, 8, 9\}$. The order of magnitude ranged from 5 to 10. Through visual inspection, we made sure that deformations based on these parameters did not produce unrealistic cell structures.

We built our CondInst model based on a ResNet-101 backbone architecture. To enable transfer learning, the entire CondInst network was initialized with weights pre-trained on the COCO (Common Objects in Context) object detection dataset [57]. Hence, we lowered the initial learning rate by a factor of 10, which resulted in a learning rate of 0.001. To match the number of classes, we modified the classification head accordingly. We optimized the network for 16,600 iterations at a batch size of 4, i.e., more than 50 epochs. When approaching the end of the training, the learning rate was decreased twice according to the scheduler used in CondInst [53].

2.3. Hardware and Software

We implemented our method using the AdelaiDet [58] and Detectron2 [59] frameworks and utilized the Augmentor Python library [60] for pre-processing and, in particular, for generating elastic transformations. All our experiments were conducted on a workstation equipped with an AMD EPYC™ 7702P 64-Core CPU, 256 GB RAM, and four NVIDIA® A100-PCIe-80GB GPUs.

For runtime measurements, we used only a single GPU to run the instance segmentation model. The tile selection model was applied as a parallelized pre-processing step on the CPU.

2.4. Quality Metrics

We evaluated the EfficientNet approach for choosing countable image regions by using the accuracy score, a widely used metric defined as the proportion of true positive results, i.e., both true positives and true negatives in all predictions made by the model. Furthermore, we calculated the F1 score, which is the harmonic mean of recall and precision,

i.e., $2 \cdot \frac{p \cdot r}{p+r}$. Recall and precision were computed as $p = \frac{TP}{TP+FP}$ and $r = \frac{TP}{TP+FN}$, where TP, TN, FP, and FN are true positives, true negatives, false positives, and false negatives, respectively. We evaluated our instance segmentation models in terms of average precision (AP), a common metric for object detection tasks. The AP is defined as the mean precision for equally spaced recall values. The metric corresponds to the area under the precision-recall curve, where predicted bounding boxes with an Intersection over Union (IoU) of more than a threshold t are considered to be true positives (TPs). For two sets of pixels, the IoU is defined as $IoU(A, B) = \frac{A \cap B}{A \cup B}$. Predictions that have no matching ground truth boxes are false positives (FPs) and ground truth boxes with no matching prediction are false negatives (FNs). For a given IoU threshold t , the AP is computed as an interpolation based on 101 values [57]:

$$AP(t) = \frac{1}{101} \sum_{r \in \{0, 0.01, \dots, 1\}} p_{interp}(r), \quad (1)$$

where $p_{interp}(r) = \max_{\tilde{r} \geq r} p(\tilde{r})$ and $p(\tilde{r})$ is the measured precision at recall \tilde{r} . In our experiments, we set $t = 0.5$, i.e., an IoU of 50% between ground truth and the predicted bounding box or segmentation is required for a proposal to be counted as a true positive. We denote this metric as AP@50.

To evaluate the overall performance, we calculated the mean AP (mAP) score by taking the mean value of the AP scores from the different classes.

We performed each experiment five times with random seeds and report the standard deviation to make sure that we present reliable results.

3. Results

3.1. Tile Selection

The models were evaluated on a held-out dataset consisting of 298 positive and 346 negative examples.

As Table 2 shows, both of the models performed very well with accuracies and F1 scores above 96%. The smaller version, i.e., EfficientNet-B0, performed better with an accuracy of 97.5% and an F1 score of 97.3%.

Table 2. Results for tile selection model. We present accuracy and F1 score for the smallest available EfficientNet architectures, i.e., B0 and B1. All experiments were performed five times, and the standard deviation is reported. Values in bold indicate the best results for each metric.

Model	Accuracy	F1 Score
EfficientNet-B0	0.975 ± 0.002	0.973 ± 0.003
EfficientNet-B1	0.970 ± 0.006	0.968 ± 0.006

3.2. Detection and Segmentation

First, we performed the training with no augmentation at all. The results are summarized in Table 3 for the detection task. Adding the default data augmentation of CondInst, i.e., random horizontal flipping, improved the results by roughly 1.9% in terms of mAP. The application of further data augmentation, namely, random vertical flipping, random brightness, random contrast, and random saturation, again improved the score by roughly 2.8%. If we instead applied horizontal flipping and elastic deformations, the models still achieved an mAP of 87.7%. Combining all data augmentation methods in one model resulted in the best model, achieving 98.9% for erythrocytes, 90.2% for lymphocytes, 87.3% for eosinophils, and 86.3% for heterophils in terms of AP and, hence, an mAP score of 90.7%. Overall, combining all data augmentation methods resulted in an improvement of roughly 5.2% in terms of mAP.

Table 3. Detection results for erythrocytes and several subtypes of leukocytes. We experimented with different methods for data augmentation, i.e., random horizontal flipping (HFlip), random combinations of vertical flipping, brightness, contrast and saturation (DA), and elastic deformations (ED). We experimented with different architectures, i.e., CondInst and Mask R-CNN, and backbone models, i.e., ResNet-50 (R-50) and ResNet-101 (R-101). We report average precision at an Intersection over Union (IoU) of 50% (AP@50) for each class and the corresponding mean average precision (mAP). Values in bold indicate the best results for each metric.

Model	Backbone	HFlip	DA	ED	Detection (AP@50)				
					Erythrocyte	Lymphocyte	Eosinophil	Heterophil	mAP
CondInst	R-101				0.989 ± 0.000	0.847 ± 0.020	0.837 ± 0.006	0.765 ± 0.016	0.860
	R-101	✓			0.989 ± 0.000	0.869 ± 0.009	0.849 ± 0.007	0.799 ± 0.014	0.877
	R-101	✓	✓		0.989 ± 0.000	0.896 ± 0.004	0.869 ± 0.004	0.852 ± 0.009	0.902
	R-101	✓		✓	0.989 ± 0.000	0.882 ± 0.004	0.862 ± 0.008	0.813 ± 0.009	0.887
	R-101	✓	✓	✓	0.989 ± 0.000	0.902 ± 0.006	0.873 ± 0.006	0.863 ± 0.011	0.907
CondInst	R-50	✓	✓	✓	0.989 ± 0.000	0.877 ± 0.007	0.853 ± 0.004	0.818 ± 0.007	0.884
Mask-RCNN	R-101	✓	✓	✓	0.938 ± 0.004	0.844 ± 0.006	0.785 ± 0.007	0.812 ± 0.005	0.845

The results of the instance segmentation shown in Table 4 are similar to those obtained for the corresponding bounding box detections. Each additional data augmentation step increased the mAP scores, and the best result was achieved by applying all random data augmentation techniques. However, the best model was not as dominant as in the detection task and could not outperform the other approaches in every category.

We observe that erythrocytes were continuously recognized almost perfectly with 98.9% and 99.0% AP for detection and segmentation, respectively. Thus, the model learned to not confuse thrombocytes or immature erythrocytes with erythrocytes. However, there was also an obvious drop in performance for all leukocyte subclasses. On the one hand, erythrocytes were easiest to identify because of the characteristic nucleus, and on the other hand, they were by far the most frequent cell type in avian blood samples, i.e., roughly 98% of all instances in our dataset. Therefore, the model could learn better features from this large set of samples. Among the leukocytes, this trend was evident as well. The most frequent leukocyte class, i.e., lymphocytes, still achieved 90.0% in terms of AP, while eosinophils and heterophils achieved 87.3% and 85.2%, respectively. Thus, there appears to be a correlation between the number of training samples and performance, which is, however, not statistically significant.

To further analyze the relation of precision and recall in more detail, we plotted the precision–recall curve of our best model in Figure 7. The graph of the erythrocyte (blue line) class is almost perfect, as expected, with an AP and, hence, an area under the curve of 99%. The graphs for the other classes start descending sooner, but by choosing the best threshold, in our case 0.5, a good balance between precision and recall could be achieved.

CondInst with a smaller backbone, namely ResNet-50, performed very well but could not compete with the model based on ResNet-101. In comparison, the performance deteriorated by roughly 2.5% in terms of mAP. However, the anchor-based Mask R-CNN approach using a ResNet-101 backbone showed a clear drop in performance of roughly 6.8% compared to the anchor-free CondInst approach using the identical backbone.

Table 4. Instance segmentation results for erythrocytes and several subtypes of leukocytes. We experimented with different methods for data augmentation, i.e., random horizontal flipping (HFlip), random combinations of vertical flipping, brightness, contrast and saturation (DA), and elastic deformations (ED). We experimented with different architectures, i.e., CondInst and Mask R-CNN, and backbone models, i.e., ResNet-50 (R-50) and ResNet-101 (R-101). We report average precision at an IoU of 50% (AP@50) and the corresponding mean average precision (mAP). Values in bold indicate the best results for each metric.

Model	Backbone	HFlip	DA	ED	Segmentation (AP@50)				
					Erythrocyte	Lymphocyte	Eosinophil	Heterophil	mAP
CondInst	R-101				0.989 ± 0.000	0.846 ± 0.019	0.835 ± 0.006	0.766 ± 0.017	0.859
	R-101	✓			0.990 ± 0.000	0.871 ± 0.010	0.846 ± 0.007	0.798 ± 0.014	0.876
	R-101	✓	✓		0.990 ± 0.000	0.895 ± 0.004	0.869 ± 0.004	0.852 ± 0.009	0.902
	R-101	✓		✓	0.990 ± 0.000	0.883 ± 0.007	0.860 ± 0.008	0.812 ± 0.010	0.886
	R-101	✓	✓	✓	0.990 ± 0.000	0.903 ± 0.007	0.873 ± 0.006	0.862 ± 0.013	0.907
CondInst	R-50	✓	✓	✓	0.990 ± 0.000	0.877 ± 0.009	0.853 ± 0.004	0.817 ± 0.006	0.884
Mask R-CNN	R-101	✓	✓	✓	0.938 ± 0.004	0.844 ± 0.006	0.783 ± 0.007	0.812 ± 0.005	0.844

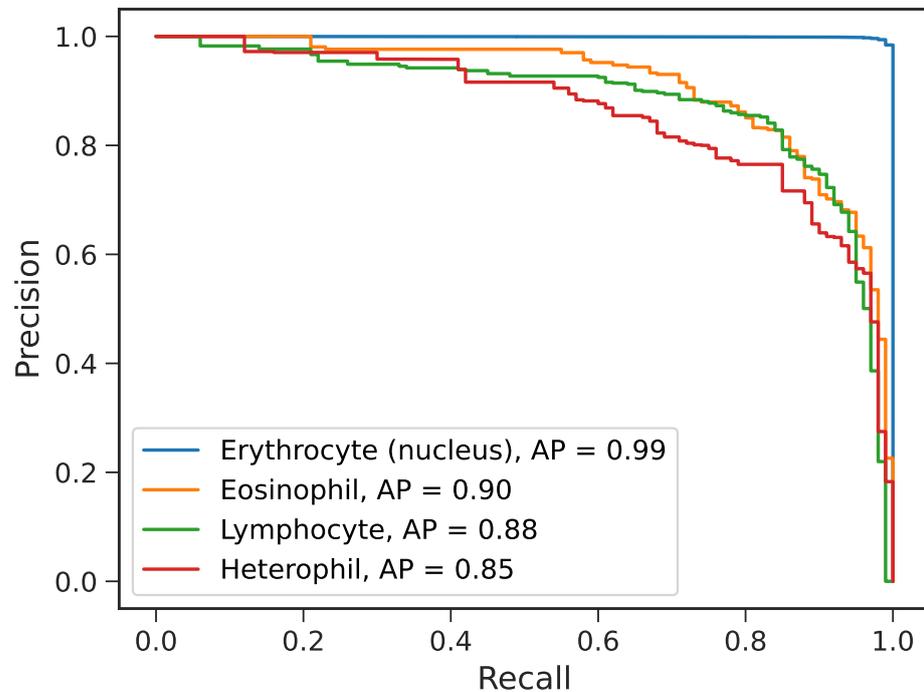


Figure 7. Precision–recall curve. The curves for the corresponding cell types are drawn in different colors, i.e., blue for erythrocytes, orange for eosinophils, green for lymphocytes, and red for heterophils.

We did not have enough samples of basophils and monocytes for a comprehensive evaluation of their respective classes, but these samples could be aggregated into their superclass leukocytes. We trained a binary CondInst model that could classify avian blood cells into erythrocytes and leukocytes. As Tables 5 and 6 show, our model could perform very well on this task, achieving more than 93% and 98.8% in terms of AP for leukocytes and erythrocytes, respectively. As before, the larger backbone, i.e., the ResNet-101, pushed the model to a better performance on leukocytes.

The mAP score regarding all leukocytes was higher than for any of the subclasses. Presumably, the multi-class model confused cell instances of the different subclasses.

Table 5. Detection results for binary model. We report the AP at an IoU of 50% (AP@50) based on the predicted bounding boxes for the CondInst architecture with two different backbone models, namely, ResNet-50 (R-50) and ResNet-101 (R-101). Values in bold indicate the best results for each metric.

Model	Backbone	Detection (AP@50)		
		Erythrocyte	Leukocyte	mAP
CondInst	R-50	0.989 ± 0.000	0.933 ± 0.002	0.961
	R-101	0.989 ± 0.000	0.936 ± 0.002	0.963

Table 6. Segmentation results for binary model. We report the AP at an IoU of 50% (AP@50) based on the predicted segmentation masks for the CondInst architecture with two different backbone models, namely, ResNet-50 (R-50) and ResNet-101 (R-101). Values in bold indicate the best results for each metric.

Model	Backbone	Detection (AP@50)		
		Erythrocyte	Leukocyte	mAP
CondInst	R-50	0.990 ± 0.000	0.932 ± 0.002	0.961
	R-101	0.990 ± 0.000	0.937 ± 0.002	0.964

3.3. Inference Runtimes

The inference runtimes for samples of different sizes are shown in Table 7.

Table 7. Inference runtimes. We present details on the samples regarding their size and countability along with the inference time of the respective image. The number of countable tiles is determined by our best area-selection model based on the EfficientNet-B0 architecture.

Sample	Width (px)	Height (px)	Area (px)	Overall Tiles	Countable Tiles	Fraction (%)	Time (mm:ss)
1_023	195,126	91,974	17,947 B	91,059	49,369	54.22	52:01
1_044	155,375	92,066	14,305 B	72,417	61,998	85.61	57:07
2_013	63,743	65,396	4,169 B	21,080	4140	19.64	05:26
5_055	53,784	46,887	2,511 B	12,705	535	4.21	01:58
8_036	205,167	93,585	19,201 B	97,200	24,743	25.46	25:09
8_040	195,216	88,232	17,224 B	87,249	9697	11.11	13:39

We included the largest whole slide image (i.e., sample 8_036) consisting of more than 19 billion pixels as well as the smallest sample (5_055) with only roughly 2.5 billion pixels. However, in addition to the size of the image, the fraction of actual countable tiles played a crucial role in the processing times. For the largest file (8_036) containing 97,200 tiles with a countable tiles fraction of roughly one-fourth, our approach took roughly 25 min. Yet, another sample (1_023) with only 91,059 tiles, but more than half of them classified as countable, took roughly 52 min. Processing the three smaller samples took less than 15 min each. In general, none of the selected images needed more than one hour to determine the cell counts in the corresponding blood smear. Depending on the mentioned factors, processing took mostly less than a tenth of a second for one countable tile, including tile selection, segmentation, and identification, as well as counting of the respective cell instances. In contrast, our human expert took an average of roughly two minutes to annotate a tile with labels and segmentation masks in our semi-automated setting.

4. Discussion

Our novel approach offers a proficient assessment of avian blood scans, which speeds up the workflow of blood cell counting significantly compared to the traditional method of visually counting on microscopes. Compared to existing hardware devices for automated blood analysis [25,28], which are usually quite expensive, our approach is freely available. Hence, we enable researchers who do not have access to such devices used in veterinarian

laboratories to utilize an automated cell-counting method. The CellaVision® DC-1 analyzer has been evaluated for mammalian, reptilian, and avian blood by comparing its pre-classification to the final results after review by veterinarians [61]. The agreement was very good for neutrophils, heterophils, and lymphocytes (each > 90%) and good for monocytes (81%). However, eosinophils and basophils needed massive re-classification by human experts. Interestingly, while we agree that achieving good performance for basophils is a challenge, our model appears to be more reliable for eosinophils. However, we could not evaluate our model on monocytes that were recognized in a satisfactory way by the CellaVision® DC-1. Moreover, our approach can be more efficient than hardware-based approaches. The DC-1 [28] analyzer processes given slides sequentially, achieving a throughput of no more than roughly 10 slides an hour. Our approach allows users to scan slides with various methods, e.g., with microscope cameras or high throughput scanners, like the Leica Aperio AT2 Scanner [47] with a capacity of 400 slides, as used in our study. The Leica Aperio AT2 Scanner can be used to digitize a large number of slides in a very time-efficient manner. Our approach can be arbitrarily scaled by processing several slide images in parallel and is only limited by the available hardware resources. Furthermore, our approach can handle low-quality blood smears because it has been trained under such conditions, while the CellaVision® DC-1 analyzer is primarily designed for usage in veterinary laboratories. Moreover, because of its proprietary design, it is not possible to use custom training data to adapt the classification approach. Hence, regarding large numbers of avian blood data sampled in ornithological field studies, our approach opens new possibilities for bird-related research.

While our approach shows that it is feasible to automatically count not only red but also white avian blood cells with open-source software, it still has some downsides. Because of the low number of samples in our training set, our neural network model is not yet able to reliably recognize basophils or monocytes. Furthermore, the model is trained on a limited number of bird species. Because of potential variations in staining intensity, coloration, and cell morphology, it may be a challenge to detect cells of other bird species as reliably as for the given species [62]. In particular, eosinophils may be quite different between bird species.

However, these issues indicate several areas for future work. The model performance can be further improved by extending the dataset in general and particularly for the rare classes, i.e., for basophils and monocytes. Instead of bluntly annotating more images that barely contain any of these cells, this can be done using an active learning approach, which reliably provides unlabeled images that contain these types of avian white blood cells. Moreover, it is a promising direction to generate more training samples by generative deep learning approaches, like GANs [63], or image generation models based on latent diffusion [64]. Furthermore, our approach can be extended to recognize and count blood parasites (e.g., *Haemosporida* and *Trypanosoma*). Another interesting aspect is investigating and improving the generalization ability of our neural network model in cross-domain scenarios. This can include different techniques when creating blood smears for different bird species. We plan to include further bird species into our model, e.g., penguins.

Several studies have indicated that extreme ecological conditions can significantly increase hematocrit levels in birds. For example, a female great tit from the northernmost populations in Northern Finland showed a hematocrit level of 0.83 [65]. This makes the blood viscous and leads to densely packed cells in the blood smear image, which can be challenging for automated counting approaches. Since we trained our model to count only areas matching human quality standards, we only counted tiles from the monolayer. Hence, a high hematocrit level may lead to significantly more rejected tiles. However, our approach is adaptable to new annotated data sources. Thus, providing our models with manually labeled images with high hematocrit levels in future training iterations will improve their ability to process and count cells with such rare conditions better. In general, our approach is based on open-source software. Therefore, the models can easily be adapted to other datasets or extended to recognize further cell types.

So far, our approach aims to automate the tedious task of manually counting avian blood cells. Furthermore, it eliminates inter-observer errors. However, it still counts cells only in the monolayer. Future work may expand the countable areas, as achieved by handcrafted feature algorithms [39]. For a deep learning approach like ours, this can be achieved by training the model with data involving lower-quality areas. By learning useful features from the annotated samples, the resulting models may be capable of achieving superhuman performance.

Our deep learning model opens up new opportunities in ornithology and ecology for documenting and evaluating the stress levels and health conditions of bird populations and communities efficiently and can, therefore, be used as an early warning indicator to detect physiological changes within populations or communities even before a population declines. With this fast, reliable, and automated approach, even old collection samples may retrospectively be incorporated into modern ornithological research. Our approach is currently used in practice for research on the relative stress load of forest birds by automatically determining H/L ratios.

5. Conclusions

We presented a fully automated open-source software approach to determine not only the total erythrocyte count but also the total and differentiated leukocyte counts of avian blood samples. Our approach operates on whole slide blood smear images. First, we select tiles using a deep neural network model to determine the areas of the images that are suitable for counting contained cells by classifying all tiles into countable and non-countable ones. Each tile classified as countable is then fed into another deep neural network model. This model is capable of detecting and classifying avian blood cells, i.e., erythrocytes and leukocytes, with 96.1% in terms of mAP. Furthermore, if the model is trained to also recognize subtypes of leukocytes, it achieves up to 98.9%, 90.2%, 87.3%, and 86.3% in terms of AP for erythrocytes (with nuclei), lymphocytes, eosinophils, and heterophils, respectively.

Author Contributions: Conceptualization, M.V. and F.S.; methodology, M.V., F.S., H.B., M.M., N.K. and D.S.; software, M.V.; validation, M.V.; investigation, M.V. and F.S.; resources, F.S., S.R. and D.G.S.; data curation, M.V. and F.S.; writing—original draft preparation, M.V. and F.S.; writing—review and editing, all authors; visualization, M.V.; supervision, M.M., N.F., D.G.S., S.R. and B.F.; project administration, N.F. and B.F.; funding acquisition, N.F. and B.F.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Hessian State Ministry for Higher Education, Research and the Arts (HMWK) (LOEWE Natur 4.0 and hessian.AI Connectom AI4Birds, AI4BirdsDemo).

Institutional Review Board Statement: The animal study protocol was approved by the Institutional Review Board (or Ethics Committee) of Regional Council of Giessen, Hesse, Germany (protocol code V 54–19 c 20 15 h 01 MR 20/15 Nr. G 10/2019 and date of approval 26 February 2019).

Data Availability Statement: Data are contained within the article. All relevant software and data presented in this article, including the trained neural network models, are openly available at <https://data.uni-marburg.de/handle/dataumr/250> (accessed on 17 January 2024). Furthermore, our software repository is publicly available at <https://github.com/umr-ds/avibloodcount> (accessed on 17 January 2024).

Acknowledgments: We thank Yvonne R. Schumm, Petra Quillfeldt, and Juan F. Masello of the University of Giessen, Germany, for their valuable feedback. Furthermore, we would like to thank the Institute of Pathology at the University of Marburg, especially Mareike Meier, for providing the slide scanner and scanning the slides. The field campaign for blood sampling was supported by numerous members and students of the Conservation Ecology group at the University of Marburg. The software development and data annotation process were supported by Valeryia Kizik, a student assistant at the University of Marburg.

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

References

- Höchst, J.; Bellafkir, H.; Lampe, P.; Vogelbacher, M.; Mühling, M.; Schneider, D.; Lindner, K.; Rösner, S.; Schabo, D.G.; Farwig, N.; et al. Bird@Edge: Bird Species Recognition at the Edge. In Proceedings of the 10th International Conference on Networked Systems (NETYS), Virtual Event, 17–19 May 2022; pp. 69–86. [CrossRef]
- Ferreira, A.C.; Silva, L.R.; Renna, F.; Brandl, H.B.; Renoult, J.P.; Farine, D.R.; Covas, R.; Doutrelant, C. Deep Learning-based Methods for Individual Recognition in Small Birds. *Meth. Eco. Evol.* **2020**, *11*, 1072–1085. [CrossRef]
- Johnson, D.H.; Krapu, G.L.; Reinecke, K.J.; Jorde, D.G. An Evaluation of Condition Indices for Birds. *J. Wildl. Manage.* **1985**, *49*, 569–575. [CrossRef]
- Balen, J.H.V. The Significance of Variations in Body Weight and Wing Length in the Great Tit, *Parus Major*. *Ardea* **1967**, *55*, 1–59. [CrossRef]
- Thompson, C.; Hillgarth, N.; Leu, M.; McClure, H. High Parasite Load in House Finches (*Carpodacus mexicanus*) is Correlated with Reduced Expression of a Sexually Selected Trait. *Am. Nat.* **1997**, *149*, 270–294. [CrossRef]
- Schoenle, L.A.; Kernbach, M.; Haussmann, M.F.; Bonier, F.; Moore, I.T. An Experimental Test of the Physiological Consequences of Avian Malaria Infection. *J. Anim. Ecol.* **2017**, *86*, 1483–1496. [CrossRef] [PubMed]
- Froget, G.; Butler, P.J.; Handrich, Y.; Woakes, A.J. Heart Rate as an Indicator of Oxygen Consumption: Influence of Body Condition in the King Penguin. *J. Exp. Biol.* **2001**, *204*, 2133–2144. [CrossRef] [PubMed]
- Rösner, S.; Schabo, D.; Palme, R.; Lorenc, T.; Mussard-Forster, E.; Brandl, R.; Müller, J. High-quality Habitats and Refuges from Tourism Reduce Individual Stress Responses in a Forest Specialist. *Wildl. Res.* **2023**, *50*, 1071–1084. [CrossRef]
- Sorenson, G.H.; Dey, C.J.; Madliger, C.L.; Love, O.P. Effectiveness of Baseline Corticosterone as a Monitoring Tool for Fitness: A Meta-analysis in Seabirds. *Oecologia* **2016**, *183*, 353–365. [CrossRef]
- Davis, A.; Maney, D.; Maerz, J. The Use of Leukocyte Profiles to Measure Stress in Vertebrates: A Review for Ecologists. *Funct. Ecol.* **2008**, *22*, 760–772. [CrossRef]
- Skwarska, J. Variation of Heterophil-To-Lymphocyte Ratio in the Great Tit *Parus major*-a Review. *Acta Ornithol.* **2019**, *53*, 103–114. [CrossRef]
- Cirule, D.; Krama, T.; Vrublevska, J.; Rantala, M.J.; Krams, I. A Rapid Effect of Handling on Counts of White Blood Cells in a Wintering Passerine Bird: A More Practical Measure of Stress? *J. Ornithol.* **2011**, *153*, 161–166. [CrossRef]
- Masello, J.F.; Choconi, R.G.; Helmer, M.; Kremberg, T.; Lubjuhn, T.; Quillfeldt, P. Do Leucocytes Reflect Condition in Nestling Burrowing Parrots *Cyanoliseus patagonus* in the Wild? *Comp. Biochem. Physiol. Part A Mol. Integr. Physiol.* **2009**, *152*, 176–181. [CrossRef] [PubMed]
- Salvante, K.G. Techniques for Studying Integrated Immune Function in Birds. *Auk* **2006**, *123*, 575–586. [CrossRef]
- Wojczulanis-Jakubas, K.; Jakubas, D.; Czujkowska, A.; Kulaszewicz, L.; Kruszewicz, A.G. Blood Parasite Infestation and the Leukocyte Profiles in Adult and Immature Reed Warblers (*Acrocephalus scirpaceus*) and Sedge Warblers (*Acrocephalus schoenobaenus*) During Autumn Migration. *Ann. Zool. Fenn.* **2012**, *49*, 341–349. [CrossRef]
- Ruiz, G.; Rosenmann, M.; Novoa, F.F.; Sabat, P. Hematological Parameters and Stress Index in Rufous-Collared Sparrows Dwelling in Urban Environments. *Condor* **2002**, *104*, 162–166. [CrossRef]
- Hawkey, C.M.; Dennett, T.B.; Peirce, M.A. *A Colour Atlas of Comparative Veterinary Haematology: Normal and Abnormal Blood Cells in Mammals, Birds and Reptiles*; State University Press: Ames, IA, USA, 1989.
- Rupley, A.E. *Manual of Avian Practice*; W B Saunders: London, UK, 1997.
- Davis, A.K. The Wildlife Leukocytes Webpage: The Ecologist's Source for Information about Leukocytes of Wildlife Species. Available online: <http://wildlifehematology.uga.edu> (accessed on 12 January 2024).
- Meechart, K.; Auethavekiat, S.; Sa-ing, V. An Automatic Detection for Avian Blood Cell based on Adaptive Thresholding Algorithm. In Proceedings of the 12th Biomedical Engineering International Conference (BMEiCON), Ubon Ratchathani, Thailand, 19–22 November 2019; pp. 1–4. [CrossRef]
- Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern. Syst.* **1979**, *9*, 62–66. [CrossRef]
- Beaufrère, H.; Ammersbach, M.; Tully, T.N. Complete Blood Cell Count in Psittaciformes by Using High-Throughput Image Cytometry: A Pilot Study. *J. Avian Med. Surg.* **2013**, *27*, 211–217. [CrossRef]
- Kamentsky, L.; Jones, T.; Fraser, A.; Bray, M.; Logan, D.; Madden, K.; Ljosa, V.; Rueden, C.; Harris, G.; Eliceiri, K.; et al. Improved Structure, Function, and Compatibility for CellProfiler: Modular High-throughput Image Analysis Software. *Bioinformatics* **2011**, *27*, 1179–1180. [CrossRef]
- Jones, T.R.; Carpenter, A.E.; Lamprecht, M.R.; Moffat, J.; Silver, S.J.; Grenier, J.K.; Castoreno, A.B.; Eggert, U.S.; Root, D.E.; Golland, P.; et al. Scoring Diverse Cellular Morphologies in Image-based Screens with Iterative Feedback and Machine Learning. *Proc. Natl. Acad. Sci. USA* **2009**, *106*, 1826–1831. [CrossRef]
- Abbott. CELL-DYN Family Hematology Analyzers and Systems | Abbott Core Laboratory. Available online: <https://www.corelaboratory.abbott/int/en/offering/brands/cell-dyn.html> (accessed on 6 January 2024).
- Lilliehöök, I.; Wall, H.; Tauson, R.; Tvedten, H. Differential Leukocyte Counts Determined in Chicken Blood Using the Cell-Dyn 3500. *Vet. Clin. Pathol.* **2004**, *33*, 133–138. [CrossRef]
- Post, J.; Rebel, J.; ter Huurne, A. Automated Blood Cell Count: A Sensitive and Reliable Method to Study Corticosterone-related Stress in Broilers. *Poult. Sci.* **2003**, *82*, 591–595. [CrossRef] [PubMed]

28. CellaVision. CellaVision® DC-1. Available online: <https://www.cellavision.com/products/analyzers/cellavisionr-dc-1> (accessed on 2 January 2024).
29. CellaVision. CellaVision® VET. Available online: <https://www.cellavision.com/products/software/cellavisionr-vet> (accessed on 2 January 2024).
30. Lee, G.H.; Yoon, S.; Nam, M.; Kim, H.; Hur, M. Performance of Digital Morphology Analyzer CellaVision DC-1. *Clin. Chem. Lab. Med. (CCLM)* **2023**, *61*, 133–141. [[CrossRef](#)] [[PubMed](#)]
31. Mayes, C.; Gwilliam, T.; Mahe, E.R. Improving Turn-around Times in Low-throughput Distributed Hematology Laboratory Settings with the CellaVision® DC-1 Instrument. *J. Lab. Med.* **2023**. [[CrossRef](#)]
32. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Proceedings of the 18th International Conference, Munich, Germany, 5–9 October 2015, Part III*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2015; Volume 9351, pp. 234–241. [[CrossRef](#)]
33. Korfhage, N.; Mühling, M.; Ringshandl, S.; Becker, A.; Schmeck, B.; Freisleben, B. Detection and Segmentation of Morphologically Complex Eukaryotic Cells in Fluorescence Microscopy Images via Feature Pyramid Fusion. *PLoS Comput. Biol.* **2020**, *16*, e1008179. [[CrossRef](#)] [[PubMed](#)]
34. Chola, C.; Maaad, A.Y.; Heyat, M.B.B.; Jv, B.B.; Naji, W.; Hemachandran, K.; Mahmoud, N.; Abdelsamee, N.; Al-antari Aisslab, M.A.; Kadah, Y.; et al. BCNet: A Deep Learning Computer-Aided Diagnosis Framework for Human Peripheral Blood Cell Identification. *Diagnostics* **2022**, *12*, 2815. [[CrossRef](#)]
35. Kittichai, V.; Kaewthamasorn, M.; Thanee, S.; Sasisaowapak, T.; Naing, K.; Jomtarak, R.; Tongloy, T.; Chuwongin, S.; Boonsang, S. Superior Auto-identification of Medically Important Trypanosome Parasites by Using a Hybrid Deep-learning Model. *J. Vis. Exp.* **2023**. [[CrossRef](#)]
36. Guo, R.; Xie, K.; Pagnucco, M.; Song, Y. SAC-Net: Learning with Weak and Noisy Labels in Histopathology Image Segmentation. *Med. Image Anal.* **2023**, *86*, 102790. [[CrossRef](#)]
37. Rashmi, R.; Prasad, K.; Udupa, C. Breast Histopathological Image Analysis using Image Processing Techniques for Diagnostic Puposes: A Methodological Review. *J. Med. Syst.* **2021**, *46*, 7. [[CrossRef](#)]
38. Chattopadhyay, A.; Maitra, M. MRI-based Brain Tumour Image Detection Using CNN based Deep Learning Method. *Neurosci. Inform.* **2022**, *2*, 100060. [[CrossRef](#)]
39. Govind, D.; Lutnick, B.; Tomaszewski, J.; Sarder, P. Automated Erythrocyte Detection and Classification from Whole Slide Images. *J. Med. Imaging* **2018**, *5*, 027501. [[CrossRef](#)]
40. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
41. Kittichai, V.; Kaewthamasorn, M.; Thanee, S.; Jomtarak, R.; Klanboot, K.; Naing, K.M.; Tongloy, T.; Chuwongin, S.; Boonsang, S. Classification for Avian Malaria Parasite Plasmodium gallinaceum Blood Stages by Using Deep Convolutional Neural Networks. *Sci. Rep.* **2021**, *11*, 16919. [[CrossRef](#)] [[PubMed](#)]
42. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
43. Parwani, A.V. (Ed.) *Whole Slide Imaging: Current Applications and Future Directions*; Springer International Publishing: Cham, Switzerland, 2022. [[CrossRef](#)]
44. umr-ds. Avibloodcount. Available online: <https://github.com/umr-ds/avibloodcount> (accessed on 12 January 2024).
45. Robertson, G.W.; Maxwell, M.H. Modified Staining Techniques for Avian Blood Cells. *Br. Poul. Sci.* **1990**, *31*, 881–886. [[CrossRef](#)] [[PubMed](#)]
46. Zeiss. Microscopes, Software & Imaging Solutions. Available online: <https://www.zeiss.com/microscopy> (accessed on 2 January 2024).
47. Leica Biosystems. Digital Pathology Microscope Slide Scanners - Whole Slide Imaging. Available online: <https://www.leicabiosystems.com/digital-pathology/scan> (accessed on 2 January 2024).
48. Goode, A.; Gilbert, B.; Harkes, J.; Jukic, D.; Satyanarayanan, M. OpenSlide: A Vendor-neutral Software Foundation for Digital Pathology. *J. Pathol. Inform.* **2013**, *4*, 27. [[CrossRef](#)]
49. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.B. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988. [[CrossRef](#)]
50. Labelbox. Labelbox. Available online: <https://labelbox.com> (accessed on 2 January 2024).
51. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning (ICML), PMLR, Long Beach, CA, USA, 9–15 June 2019; Volume 97, pp. 6105–6114.
52. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
53. Tian, Z.; Shen, C.; Chen, H. Conditional Convolutions for Instance Segmentation. In *Computer Vision—ECCV 2020, Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2020, Part I*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12346, pp. 282–298. [[CrossRef](#)]
54. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: A Simple and Strong Anchor-free Object Detector. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 1922–1933. [[CrossRef](#)] [[PubMed](#)]

55. Ren, S.; He, K.; Girshick, R.B.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
56. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525. [CrossRef]
57. Lin, T.; Maire, M.; Belongie, S.J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014, Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014, Part V*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2014; Volume 8693, pp. 740–755. [CrossRef]
58. Tian, Z.; Chen, H.; Wang, X.; Liu, Y.; Shen, C. AdelaiDet: A Toolbox for Instance-Level Recognition Tasks. 2019. Available online: <https://git.io/adelaidet> (accessed on 17 January 2024).
59. Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.Y.; Girshick, R. Detectron2. 2019. Available online: <https://github.com/facebookresearch/detectron2> (accessed on 17 January 2024).
60. Bloice, M.D.; Stocker, C.; Holzinger, A. Augmentor: An Image Augmentation Library for Machine Learning. *J. Open Source Softw.* **2017**, *2*, 432. [CrossRef]
61. Leclerc, A. Evaluation of the Cellavision® DC-1 Hematology Analyzer in Assisting with Differential White Blood Cell Counts in Zoo Practice. In Proceedings of the The Joint AAZV/EAZWV Conference, Virtual Event, 4, 12, 20, 28 October and 5 November 2021.
62. Feldman, B.; Jain, N.; Stein, C.S.; Zinkl, J. *Schalm's Veterinary Hematology*, 5th ed.; Blackwell: London, UK, 2000.
63. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
64. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis With Latent Diffusion Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 10684–10695.
65. Krama, T.; Suraka, V.; Hukkanen, M.; Rytönen, S.; Orell, M.; Cirule, D.; Rantala, M.; Krams, I. Physiological Condition and Blood Parasites of Breeding Great Tits: A Comparison of Core and Northernmost Populations. *J. Ornithol.* **2013**, *154*, 1019–1028. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.