applied sciences

MDPI

Article

Active Learning Based on Crowdsourced Data

Tomasz Maria Boiński *, Julian Szymański and Agata Krauzewicz

Faculty of Electronics, Telecommunication and Informatics, Gdańsk University of Technology, 80-233 Gdańsk, Poland; julian.szymanski@eti.pg.gda.pl (J.S.); s160511@student.pg.edu.pl (A.K.)
* Correspondence: tomboins@pg.edu.pl

Abstract: The paper proposes a crowdsourcing-based approach for annotated data acquisition and means to support Active Learning training approach. In the proposed solution, aimed at data engineers, the knowledge of the crowd serves as an oracle that is able to judge whether the given sample is informative or not. The proposed solution reduces the amount of work needed to annotate large sets of data. Furthermore, it allows a perpetual increase in the trained network quality by the inclusion of new samples, gathered after network deployment. The paper also discusses means of limiting network training times, especially in the post-deployment stage, where the size of the training set can increase dramatically. This is done by the introduction of the fourth set composed of samples gather during network actual usage.

Keywords: active learning; sample assessment; crowdsourcing

Citation: Boiński, T.M.; Szymański, J.; Krauzewicz, A. Active Learning Based on Crowdsourced Data. *Appl. Sci.* **2022**, *12*, 409. https://doi.org/10.3390/app12010409

Academic Editors: Jerry Chun-Wei Lin, Stefania Tomasiello and Gautam Srivastava

Received: 21 November 2021
Accepted: 19 December 2021
Published: 1 January 2022

1. Introduction

Nowadays the world is all about data [1]. Traditional algorithms, due to the constant development of neural networks, are losing their significance. We tend to rely on such approaches more and more due to the complexity of current-day problems and the sheer size of the problem space. These methods, however, require large volumes of properly annotated data.

Pure data gathering usually can be done easily and can be fully automated. For example, we can quite easily record bees entering and exiting a hive. On the other hand, without annotation raw data is in many cases just useless. Data annotation itself usually is what makes the training process costly and time consuming.

The sheer number of samples needed usually makes this a lengthy task, and the difficulty of the annotation process is domain-dependent and usually cannot be automated. In most cases, the process needs to be done by a team of humans. We can rely on the transductive learning [2] approaches but this requires an already annotated set of data and would require verification of the newly annotated set. In both cases, either the manual annotation or the verification of automated annotations, humans have to be involved. In some cases, they have to be experts in the field who need to be able to distinguish objects belonging to the particular class that might be very similar to each other, e.g., when annotating medical data [3]. Decisions on noisy or, e.g., overlapping objects within the same class can also sometimes be difficult.

In some cases, we can lower the cost and the time needed to prepare the data by reducing the size of the training set. One of the solutions is an Active Learning [4,5] approach. It allows better sample preparation by covering the domain with fewer, although more informative, samples. This way, the cost of generating such a set is also lower. In this method, the network is trained using only informative samples, both positive and negative, that is, samples that provide some new information allowing the creation of class prototypes and their distinction. This way, we can skip the annotation of samples that are similar to others already present in the training set.

In this approach, we need still need to prepare the data, albeit in a small volume, and, more importantly, we need a method to distinguish meaningful samples those not provide any new information.

In this paper, we propose a crowdsourcing-based approach for not only annotated data acquisition but also means to support Active Learning in the form of the oracle able to judge whether the given sample is informative or not. The approach is implemented as a general-purpose framework for data acquisition and verification. We also propose a method for improving the classification quality by network-trained Active Learning approach. Furthermore, thanks to the implementation of the crowdsourcing approach, we can limit the amount of work needed to annotate large sets of data. We also proposed the creation of an additional, fourth training set, obtained during the post-deployment stage when the pre-trained network is actually used by utility applications. Thus, after initial training, we can perpetually increase the quality of the network by the inclusion of new samples. The paper also discusses means of limiting network training times, especially in the post-deployment stage, where the size of the training set can increase dramatically.

The structure of this paper is as follows. First, in Section 2, the Active Learning approach and the state of the art for Active Learning are presented. This section also presents the CenHive system and its validation. Next, in Section 3, possible interconnections with crowdsourcing and our custom crowdsourcing framework are described. Section 4 shows the proposed approach and the results are being discussed. Finally, some conclusions are given.

2. The State of the Art

2.1. Active Learning Approaches

Traditionally in supervised learning, we train the model using a set of input–output pairs generated by an unknown target function. We aim at achieving generalization by providing as many samples as possible so that the network will be trained with different cases. The generalization capability depends on a number of factors such as network architecture, the training procedure, and the training data set itself. Traditionally, the training data are randomly selected according to the probability distribution on the input set. The current state of the network is completely ignored, and the training data are not adjusted to the current knowledge of the network. As a result, during the learning process, new samples provide less and less new information [6]. This leads to increased training time as redundant training samples slow down the process. Unnecessarily large training sets also generate additional costs as the acquisition of labeled training data can be expensive.

The aforementioned problems are addressed by Active Learning [6]. In this approach, the network is first trained with a small subset of the training set. Next, based on the current state of the network, the most informative samples are selected. Those samples are selected by the so-called oracle—the network asks the oracle whether the given sample is informative or not, and, if yes, such sample is used to train the network. The process is repeated until we reach a satisfactory state of the network or until there are no new informative samples available.

Active learning can use multiple sample selection algorithms. The most common are membership query synthesis, stream-based selective sampling, and pool-based sampling [5].

Pool-based sampling is used for problems, where a large un-tagged set is easily and cheaply available. In stream-based selective sampling, each sample is analyzed by the network and the network decides whether the given sample should be tagged or not. The third approach, Membership Query Synthesis, allows the network to request a sample not only from the pool of already available samples but any from the problem space. The implementation of such an approach is, however, problematic due to the availability of such samples or the tagging process itself.

One of the aspects of Active Learning is a way to judge whether a given sample is informative or not. The most common methods are Uncertainty Sampling, Expected Model Change, Expected Error Reduction, and Query By Committee. Uncertainty Sampling [7] allows selection of those samples where the model does not trust the labels.

Expected Model Change [8] selects in turn samples that should introduce the biggest changes in the model. Similarly Expected Error Reduction [9] prefers the samples that would introduce the greatest reduction of generalization error. In the Query By Committee, a set of models present labels that are the most appropriate for the given sample. For training, the samples are selected for which the committee disagrees the most.

Labels for selected informative samples are provided by the so-called oracle. This can be implemented using different approaches. The oracle can be a single or group of experts or some kind of automated system or even a process, depending on the domain. For classification and detection systems, the oracle, however, is usually implemented in form of human experts. An automatic system (e.g., a neural network) able to mark the samples correctly would already be a very well-trained model thus eliminating the need to create a new system. In some cases, the persons taking part in tagging process have to be experts in the problem domain, e.g., in medicine. In other cases, we can rely on an average human being, as usual, a typical, adult human is able to mark certain objects, such as plants, animals, pedestrians on the street, vehicles, bees, etc., correctly.

Active Learning allows so-called batch-mode learning. Traditionally in each iteration, only one new sample should be added to the training set. In some cases, this can lead to inefficient learning and thus to a slow and lengthy training process. Furthermore, in convolutional neural networks, a single new sample might not have any impact on the network quality due to the local optimization methods used during training [10]. In batch-mode Active Learning during each iteration, the training set is extended by a set (batch) of new informative samples. The batch selection method is also important as not always selecting the most informative samples is the best approach, as some samples can carry similar information or they can be unique and not representative or carry a lot of noise. One of the simplest and most effective solutions is to select samples randomly from the set of informative objects [5].

Over the recent years Active Learning approach have proved to be highly efficient in solving specific problems [11]; however, in most cases, the selection algorithms are not general. Usually they are hand-picked. Some work was done on a generalization of the selection algorithms, i.e., to "learn how to learn" [12–14]. This field needs to be explored further in the future as the generalization can still be improved. In the paper, we propose generalizing the selection algorithm with crowdsourcing. We focus on incremental pool-based sampling approach and batch-mode Active Learning.

2.2. The CenHive System

Typically deep neural networks learning includes the usage of three sets of data: the training set, the test set, and the evaluation set. Each of those sets plays an important role in the training process. In many cases, we have access to the additional, fourth set, which can be gathered during the actual usage of the neural network, as after training the network is usually deployed in a real-life environment. During the usage of the network, many new data can be gathered, as samples need to be passed to the network, e.g., object detection. If the detection could be somehow verified, the new data could be added to the training set for retraining and improvement of the network quality by feeding it with new data and thus reacting to changing conditions within the problem domain. For annotation verification, the knowledge of the crowd can be used by employing a crowdsourcing approach via Games with a Purpose [15] for data verification. Our and other research [16–21] shows that the approach is viable and can be successfully used.

Constant addition of the new data to the training set would, however, greatly impact the training time. The annotated data also need to be verified efficiently for the training process to be valid and not replace one bottleneck with another. In the paper, we aim

at providing a method for efficient verification of such new data samples and a way to limit the size of the training set while not sacrificing the quality and generality of the trained network. Our observation shows that the limitation of the training set can be done by using the new samples combined with only a subset of the samples used in previous training iterations. This way, the number of samples remains constant for each iteration thus limiting the time needed for the whole training process. The question remains what the proportions should be and how to gather annotated data quickly and without the requirement of large investments.

CenHive is a web-based solution allowing distribution and aggregation of mappings verification and creation tasks to human users using a Games with a Purpose approach [15]. The system allows multiple clients and data types. It supports not only text data but audio, video, images, and any combination of aforementioned input types both in questions presented to the users and answers generated by them. The amount of answers sent to the client is also configurable and the client, instead of an answer to a question presented, can create a new set of data for verification, e.g., by creating some mapping by marking objects in the image.

The data transferred to the users are aggregated into so-called contents. They can take the form of e.g., mappings (e.g., sets of connections between Wikipedia articles and a WordNet synsets), tags (e.g., images with labeled objects), or raw data suitable for labeling, etc. The tags or descriptions associated with the given content are called phrase. They can be in any form, including yes/no answers, a set of tags, etc.

The system evaluation was done with help of the faculty students over the span of 3 years, each year consisting of 60 students. With the aforementioned procedure, we tested the approach in two domains, where we tested the level of annotators agreement on the provided data:

- Mapping verification [20]—using TGame (https://play.google.com/store/apps/details?id=pl.gda.eti.kask.tgame accessed on 21 December 2021), an Android platform game, designed for Wikipedia–WordNet mapping verification. It supports audio, video, text, and images for both the contents and phrases. The player has to fulfill the task to activate checkpoints in the game. Our tests showed that the solution is viable for mapping verification. During the two-month-long test period, players gave 3731 answers in total to 338 questions. The total number of answers for different mapping types is shown in Table 1.

  During the tests, we established a procedure for dealing with malicious users. As in the crowdsourcing approach, we have no control over the user actions, and we decided to consider only tasks with multiple solutions. To eliminate blind shots, we took into account only questions that were displayed for at least 5 s—the time needed to actually read the question.
- Image tagging/verification—using 2048 clone (https://kask.eti.pg.gda.pl/cenhive2/2048/ accessed on 21 December 2021). The player, during the game, donates some of his or her computer computation power to create new data in the volunteer computing approach. The downloaded photos were analyzed using Viola-Jones Feature Detection Algorithm using the Haar Cascades [22] algorithm (more specifically using HAAR.js implementation (https://github.com/foo123/HAAR.js accessed on 21 December 2021)). Detected faces were sent back to CenHive and stored as tags, where the original image was tagged with coordinates of the rectangle containing the detected face. The player, when he or she wanted to undo a move, needed to verify the correctness of faces detects. He or she was presented with a random image selected from all of the detection done by any player.

  During the test period, where tests were done with the help of the research team and students, for 64 multi-person photos, the game players generated 628 face-detects (giving 303 distinct face detects). In this case, multiple face-detects are not necessary as they are done arithmetically using the player's device's processing power. The

detected faces were further verified—for 92 images, we received 181 verification responses and 7 reports.

Table 1. Number of answers for different type of questions.

Question Type	Questions	Answers	Answers per Question	Reports	Blind Shots	Blind Shots %
Extended mappings	239	3308	13.84	625	16	0.48
Yes/No	99	423	4.27	10	12	2.84
Total	338	3731	11.04	685	62	1.34

Two other clients are implemented that support general image tagging and aim at producing data for training neural networks using an active learning approach. Through both clients, users can verify bee detection done by a neural network and have the ability to tag bees on random images. One of the clients is a captcha element, which in its current form can be seen on the CenHive registration page (https://kask.eti.pg.gda.pl/cenhive2/?page=register accessed on 21 December 2021). The other client is an Android game that should be available for the general public in the coming weeks. This work is a part of our wider idea for a beehive monitoring system using both audio and video for hive supervision [23]. The approach was validated and shows promising results [18], which allows us to propose a generalized model for crowdsourcing supported Active Learning approach.

3. Crowdsourcing Enabled Active Learning

The proposed solution is aimed at data engineers, who can use a crowdsourcing approach, based on unqualified users, to obtain high-quality annotated data. We maintain the credibility of the obtained data by handing out the same task to multiple users, thus eliminating incorrect, either intentionally or by accident, answers. The significance of the proposed approach is backed up by the universal difficulty of obtaining proper volumes of training data. The proposed approach allows the data engineers to gather the data relatively fast and cheaply.

The introduction of the fourth set, mentioned in the previous section, allows neural network optimization and further generalization beyond the initial training stage. To further limit the data needed, we decided to use Active Learning, and the data annotation verification is done using crowdsourcing approach via CenHive system [18–21] (https://kask.eti.pg.gda.pl/cenhive2/ accessed on 21 December 2021).

In our tests, we used the fourth-set approach for image recognition. The proposed solution can be, however, applied to multiple domains and types of data. This is independent of whether the network is used for image detection, signal recognition, or other applications, and it is also independent of whether the data can be annotated easily without the use of an expert or if some expert knowledge is needed, such as medical expertise, to properly annotate the data. Our previous research shows that crowdsourcing can be successfully applied for mapping verification and data tagging. Furthermore, the knowledge gathered from the crowd can be successfully used in the Active Learning approach [18]. In this section, we present the general system for sample acquisition and its role in the Active Learning approach.

3.1. The Data

During our research, we used video recordings of bees entering and exiting from the hive [24] that were recorded in Spring 2017 and 2018. The videos were available in .mov and .mp4 formats and thus were of different quality. The images were tagged using the CenHive system using the crowdsourcing approach.

The bees were marked on Full HD images that were too big to serve as direct input for neural networks. As such the images were split into fragments measuring 500 × 500 pixels. Each image could have any number of bees. The total number of images and bees in training, validation, and test sets can be seen in Table 2. Sample images are shown in Figure 1.

Table 2. The training data.

	Training Set	Validation Set	Test Set
Images	60,422	17,292	7458
Bees	71,105	19,841	6741

The tagged images included some hard cases such as the following:

- Over 17% of the bees located near the entrances to the hive are partially visible;
- In around 4.2% of cases, the bees overlap;
- About 3% of spotted bees are hard to distinguish from the background due to the color of the surrounding environment;
- Almost 37% of the bees were blurred as they moved faster than the camera frame rate.

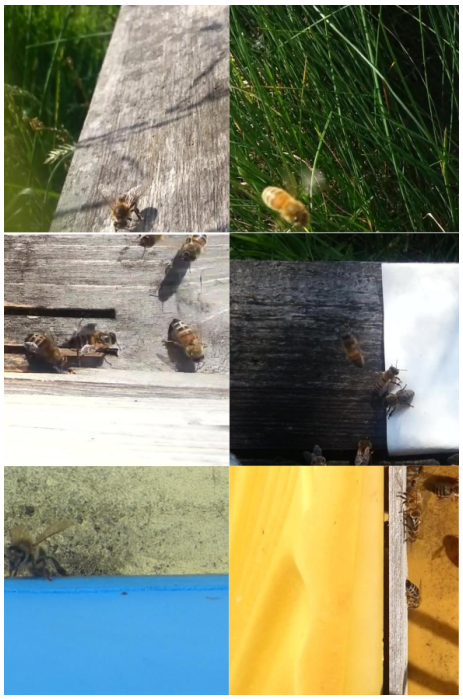

Figure 1. Sample images used as a training set.

3.2. Network Topology and the Training Process

The network topology used for object recognition is constantly changed and improved; thus, one of the aims of the proposed approach is to develop a solution that is topology-independent. During our research, we decided to use Faster R-CNN [25] network as it is often used as a base for other modern solutions and thus will be easier to replace in the future. It also scores very high in different rankings [25,26]. The network was implemented using the Facebook Detectron system [27]. As a base for the network, we decided to use the ResNet-50 based model shipped with the Detectron system, which was trained on the ImageNet-1k set [28,29].

The training process is iterative in nature. Before the first iteration, all available samples are put into the global set, and the network is initialized. With each iteration, the network quality is increased. Each iteration consists of the following steps:

1. All images available in the global set are passed to the network, and the bee detection is done;
2. All network annotated images are passed to CenHive platform;
3. The network annotated images are forwarded to the users, who verify the correctness of the annotation;
4. The user answers are used to determine the informative value of each sample and prepare the corrected samples;
5. Random subset of user verified and annotated images are added to the training set and removed from the global set;
6. The network is trained using the training set obtained in the previous step.

3.3. Active Learning Variant

In our approach, we assume that as people verify or tag new images new samples appear that can be used to further train the network. We also plan on acquiring new recordings from different hives and using them to extend the network. For this reason, we decided on using an incremental pool-based sampling approach. Due to the nature of convolutional neural networks, based on observations done by Settles [5] and Senera and Savares [10], we decided to use batch-mode Active Learning. The size of the batch and its composition will be explained in detail in the following sections.

3.4. The Knowledge of the Crowd—An Oracle

An important part of Active Learning is the oracle that is able to state whether the given sample is informative or not. In our case, we decided to use the knowledge of the crowd to determine the quality of the given sample. For that purpose, we used our own CenHive system [18–21]. We assume here, that by reaching a wider audience, we are able to annotate more images with smaller costs than by using the work of experts. The drawback here is the initial small pace of the annotation process; however, by embedding the tasks into games or captcha [16] components on commonly used websites, such as laboratory user management, we observed that despite the initial slow start we managed to gather enough users for quick and cheap data annotation.

In our approach, the current state of the network knowledge, represented by network tagged images, is passed to the system in form of tasks for the users. The tasks can be of a different type:

- Simple Yes/No question allowing the user to verify whether the picture contains a bee or not;
- Question about how many bees the user sees on the image and thus allowing verification with the number of objects detected by the network;
- More complex tasks allowing marking of the bees on the images never before seen by the network.

The images, with the appropriate question, are forwarded to the users via dedicated clients in the form of a so-called Games with a Purpose [15]. The CenHive system currently has multiple clients implemented; for this research, we decided to use an Android-based mobile game and captcha component on our computer laboratory user management system.

The users of the systems via the client applications either verify if the network correctly detected bees on the pictures or create new data that is used to train the network. By verification of the network results, we can easily determine whether the given sample is informative or not. For example, if most of the users state that the image does not contain a bee (where the network marked such) or the number of bees is different from the number returned by the network, we can use the image to extend the network quality. The images marked differently by the users and the network are selected for training as informative samples.

4. The Proposed Model Improvement Process

The introduction of the fourth set, as proposed in the previous sections, allowed us to improve the network quality originally trained with a limited set of samples. With a constantly growing number of available samples, both the verification and the training will take more and more time nullifying all benefits gained from this approach. The constant flow of new training data can easily lead to extended training times and overfitting. In a very large training set, the original training data can be underrepresented (e.g., due to excess flow of data from only one source), and thus the network can lose generality; thus, constant network improvement, based on the crowdsourcing verified state of the network, requires a specific approach to the training process. In this paper, we focus on limiting the training time without sacrificing the network quality.

We compared the obtained network quality with a traditionally trained network (related to as a reference model in the following sections), where a full training set was used during each epoch. During the tests, we aimed at maintaining the network accuracy while keeping training time in check. The proposed model allows the limitation of the time needed for network training without sacrificing the final network quality.

4.1. Number of Epochs in Each Iterations

Usually, the training length has an impact on the learning process quality [30,31]. Additional epochs, however, might not contribute to the training time. We trained the network for 1, 5, 10, 15, and 20 epochs in each iteration. In each case, we added 10 photos to the training set in each iteration. During the first iterations, the best result could be observed for the networks trained for the greater number of epochs—15 and 20, respectively. The best final quality was obtained by the network trained for 10, 15, and 20 epochs in each iteration. The lower number of epochs resulted in poor network quality (around 20% lower quality for 1- and 5-epoch networks). Fifteen- and twenty-epoch networks achieved similar quality as the ten-epoch network, but the training time increased by about 30% for every five new epochs. Further training of the 5-, 10-, and 15-epoch networks for another six iterations resulted in a similar increase in network quality for 10- and 15- epoch networks. The 5-epoch network achieved 13% worse quality than the 10-epoch network. The results can be seen in Figure 2.

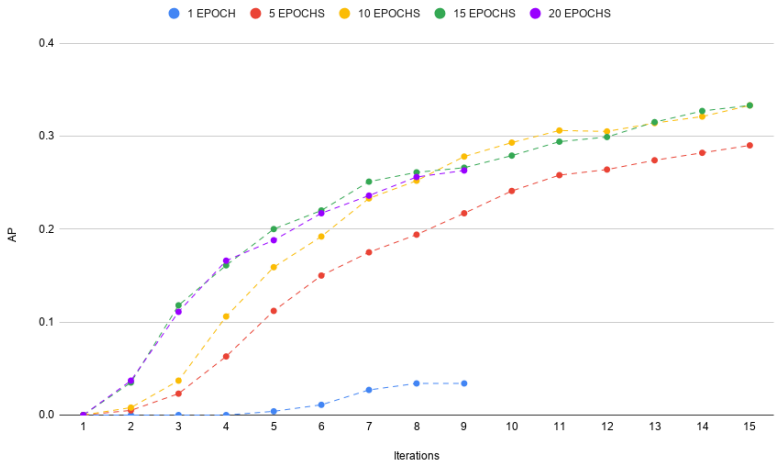

Figure 2. AP to the number of iterations for network trained with batch size of 10.

We repeated the tests for 5-, 10-epoch networks, with batch sizes of 100, 500, and 1000 images. The network quality increased more with the batch size rather than the number of epochs. For batch sizes of 500 and 1000 samples, the five-epoch network reached similar quality as the other networks trained with a higher number of epochs. The obtained results are presented in Figure 3.

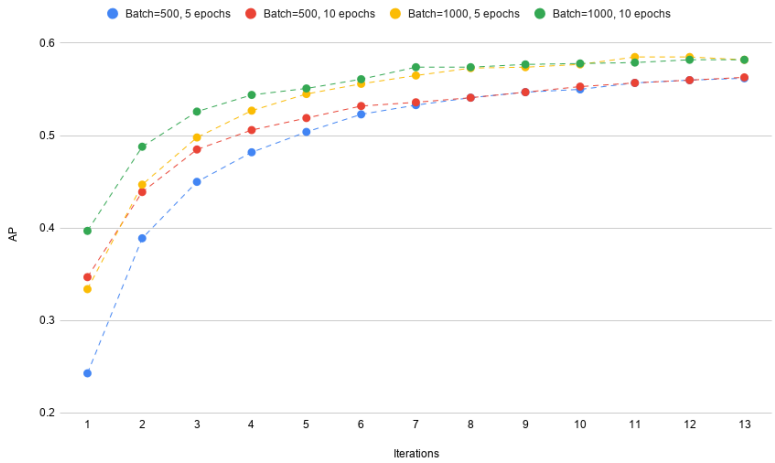

Figure 3. AP to the number of iterations for network trained with batch size of 500 and 1000.

4.2. The Batch Size

In the previous section, we showed that we can limit the number of epochs by increasing the batch size. When crowdsourcing is involved, however, the bigger the batch size, the longer it takes to distribute and verify the samples. In this section, we show how the batch size influences the network quality and thus establishes the correct value of the batch size.

The network was trained for 15 iterations, 10 epochs each, using the batch sizes of 10, 100, 500, and 1000 images that were added to the training set with each iteration. As expected, with the increasing batch size, the Average Precision (AP) also increased, and the 1000-batch network achieved the best results. The 500-batch network had its AP worsen by only 6%, and the difference narrowed with each iteration. The 100-batch network, however, was much worse, with AP only at 75% of the 1000-batch network. This time, the difference in AP did not diminish with further iterations.

When we compare the networks with respect to the ratio of achieved AP to the number of images used, the best results can be achieved for the 100-batch network. This, however, requires training for the whole 15 iterations. The 1000-batch network achieves a similar result, worse only by 6% right after the second iteration.

We also tested the networks with respect to the "understanding" of the images, i.e., the number of the informative samples available to the number of images in the training set. As can be seen in Figure 4, the networks trained using smaller batch sizes reach a similar level of data "understanding" as other networks but require a smaller number of samples. For example for a training set of size 1000 the 1000-batch network "understood" around 50% of the images, whereas the 100-batch network correctly recognized objects on 64% of the images.

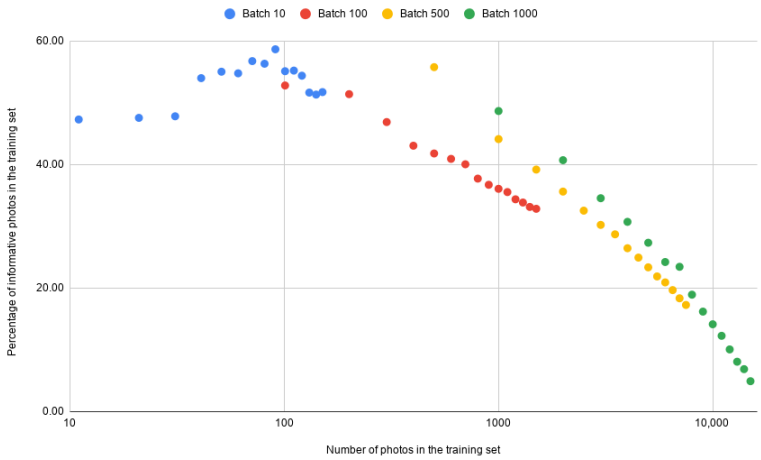

Figure 4. Known to all samples ratio for given batch size.

4.3. Iteration Time Optimization

Te tests presented in Sections 4.1 and 4.2 allowed us to check how different training process parameters impact the quality of the resulting network. The main observation is that using smaller batch sizes allows better sample management, thereby providing similar results; however, this requires a longer learning process, mainly due to the need to use more iterations during network training. To reach an AP of 0.5, our 100-batch network needed 16 iterations, whereas 500- and 1000-batch networks required only 4 and 3 iterations, respectively. We tried to find a way to shorten the iteration time. Our analysis shows that this can be done during the detection phase of our approach and during the training itself.

4.3.1. Detection Optimization

As shown in Section 3.2, the first step of each iteration is an annotation of all available images by the network by locations where bees can be found. In previously described tests, in all cases, each time we performed the detection on all images independently of whether they were considered informative or not in the previous iteration. We observed (Figure 5) that after the second iteration, the number of informative samples drop by 50%. In the following iterations, the number of such samples decreases further. However, care needs to be taken as in cases with small batch sizes the number of informative samples can be in some cases higher.

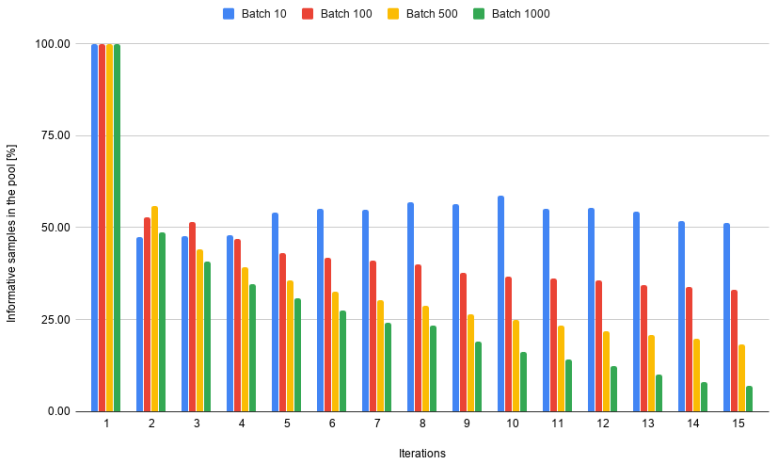

Figure 5. Ratio of informative samples to all samples in the training pool for given batch size.

We decided to modify the process described in Section 3.2. Step 4 of the algorithm was extended with the removal of non-informative samples from the global set. We then performed the training using a batch size of 100, with each iteration lasting 10 epochs. As can be seen in Figures 6 and 7, the removal of non-informative images not only did not impact the AP of the resulting network but also allowed a reduction in the number of detection operations by around 60%.

4.3.2. Training Time

Training time highly depends on the hardware used and the number of other processes running in the system. For this reason, we decided to consider training time as the number of gradient calculations that need to be done during the learning process. The number of calculations for 10-epoch training for a different number of added samples can be seen in Figure 8. During the first iterations, especially for small batch sizes, the training time is relatively short. As we can observe, however, it quickly rises as new samples are added to the training set. We followed observations found in [30] and decided to use a constant number of training samples in the following iterations. For that purpose, we modified our algorithm once again. We introduced a new training set, a global training set, containing all selected training samples. The training set actually used for training contains new randomly

selected images and a constant amount of random images from the global training set. This way, we aim at training the network with new, informative images and a subset of previously used samples, thus limiting the time needed for training and eliminating the overfitting effect that can occur when we would use only new images for a given iteration.

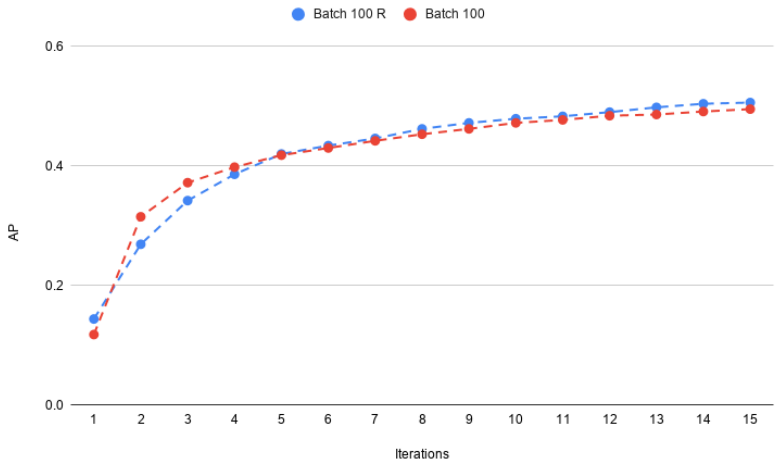

Figure 6. AP for network trained with batch 100 and network trained with modified algorithm with non-informative samples removed.

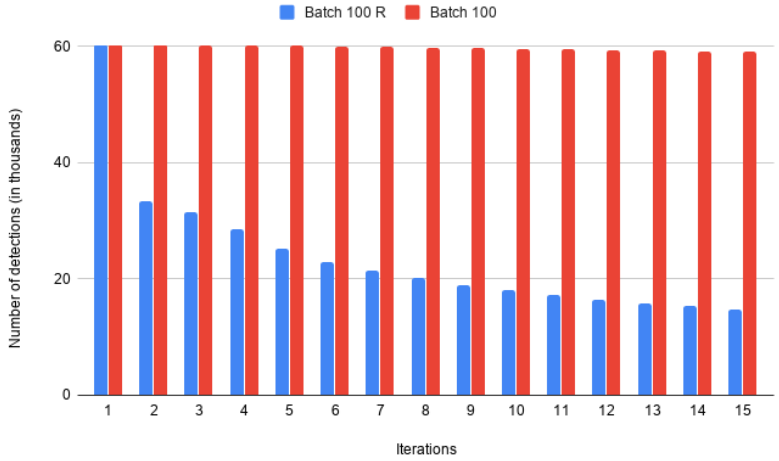

Figure 7. Number of detections done during training of the network with batch 100 and training of the network using modified algorithm with non-informative samples removed.

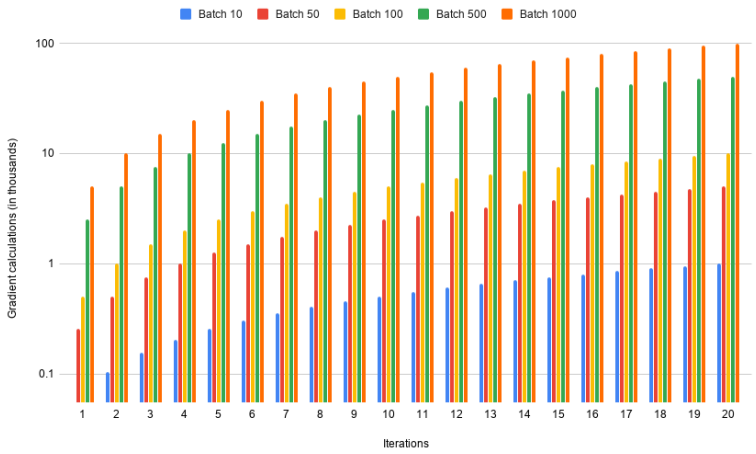

Figure 8. Gradient calculations for given batch size.

The crucial aspect of this modification is the selection of proper ratios between new samples and those from the global training set. We tested the following configurations:

- 50:50—50 known samples and 50 new samples,
- 150:50—150 known samples and 50 new samples,
- 300:50—300 known samples and 50 new samples.

As a reference, we used a network trained using the original approach with a batch size of 50. To eliminate the problem of not enough images in the global training set, we first trained the network for five iterations with a batch size of 100 using the original approach and used the training set created by this approach as the global training set in following iterations, which were done with the aforementioned configurations.

As we can see in Figure 9, not using all previous training samples does not introduce a heavy impact on the training process, and the AP is on similar levels. The best results were obtained during training of the 300:50 network. They were similar to full training of a 50-batch network but required 62% fewer gradient calculations. The remaining 150:50 and 50:50 networks achieved similar results with AP around 98% and 97.6% of the reference network, respectively. During the training of the aforementioned networks, we had to do 78% and 22% fewer gradient calculations, respectively.

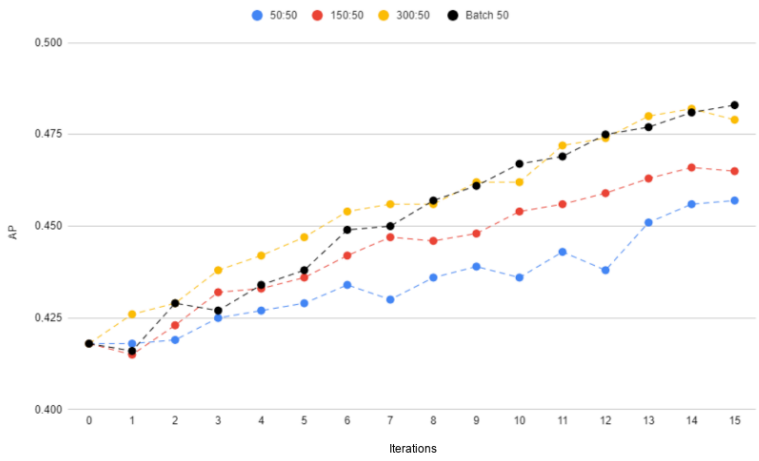

Figure 9. AP for networks trained with only part of the samples used in previous iterations.

The aforementioned changes increased the number of informative samples slightly (Figure 10), especially for 50:50 and 150:50 configurations.

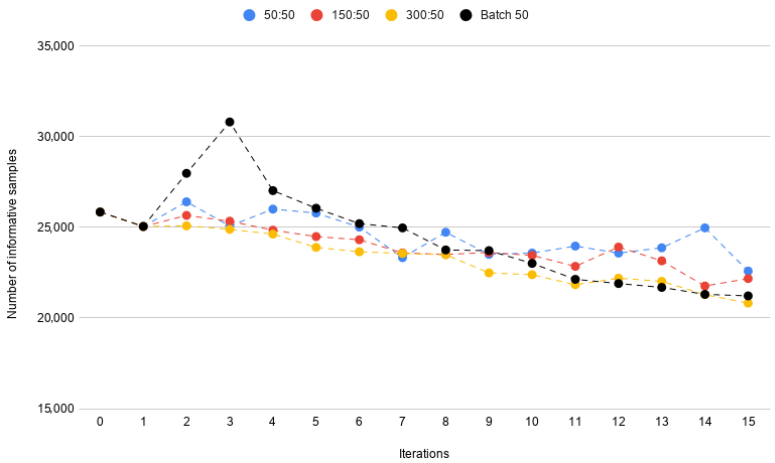

Figure 10. The number of informative samples in the pool for networks trained with only part of the samples used in previous iterations.

To verify the observations we repeated the training adding 100 samples per iteration. The configurations used were 300:100 and 600:100, and the results were compared to a previously trained 100-batch network. The results are presented in Figures 11–13, showing the achieved AP, informative samples count and gradient calculations, respectively. As we can see, the achieved AP was similar to the reference network, reaching 98.6% and 99.2% for 300:100 and 600:100 networks, respectively. The number of gradient calculations is also smaller, by 31% and 53.8%, respectively, of the number required to train the reference network. For the 300:100 network, however, the number of informative samples increased showing that either the sample ratio or the total number of samples was too low.

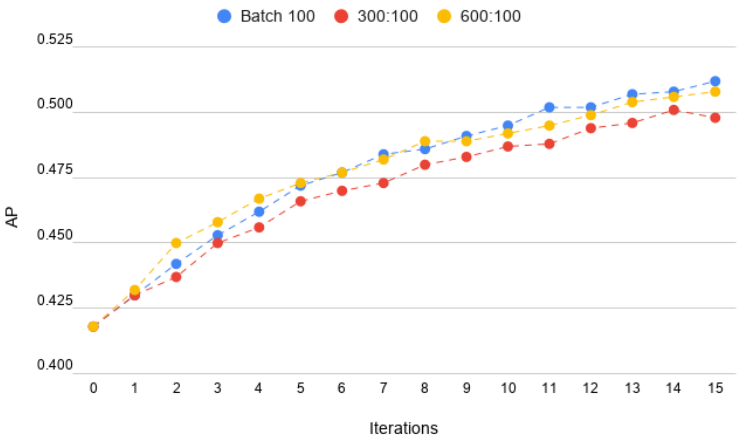

Figure 11. AP for batch 100, 300:100, and 600:100 networks.

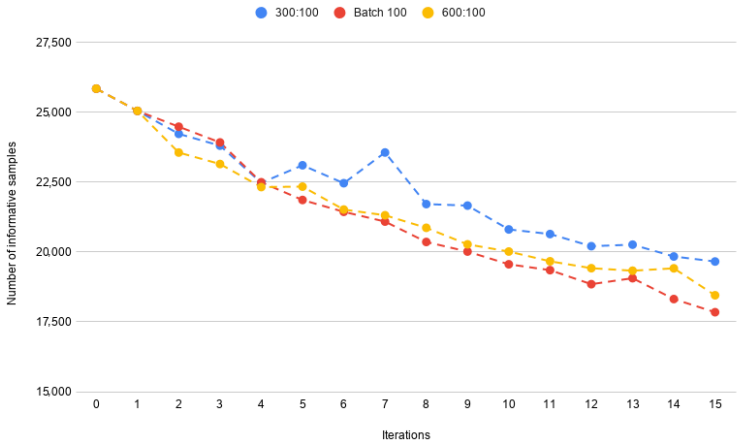

Figure 12. Number of informative samples for batch 100, 300:100, and 600:100 networks.

Figure 13. Number of gradient calculations for batch 100, 300:100, and 600:100 networks.

4.4. Final Model Parameters

Finally, we decided to use a batch size of 100. The model trained with such a batch reached AP equal to around 85% of the reference network AP after 15 iterations and required 30% fewer samples than the 1000-batch trained model. Selecting a smaller batch size allows us a more effective usage of gathered samples, thus lowering the time needed to get them and lowering the total training time. The ability to use smaller batch sizes is crucial in regard to the crowdsourcing approach, as it allows us to update the network quality with fewer responses needed from the users, thus allowing more often network updates. Unfortunately, smaller batch sizes also require more iterations to reach proper values of AP. Considering that gathering responses from users can take some time, usually a slightly increased training time is worth the benefit from a smaller number of samples required.

To mitigate the need for an increased iterations count, we also employed observations presented in Section 4.3. In this case, the removal of noninformative samples lowered the number of detections by up to 60%, and keeping the new to known training samples ratio at 600:100 allowed lowering the number of gradient calculations by 43%, with each iteration lasting 10 epochs.

5. Results

The training process lasted until there were no new informative samples available. The results against the validation set are presented in Figure 14.

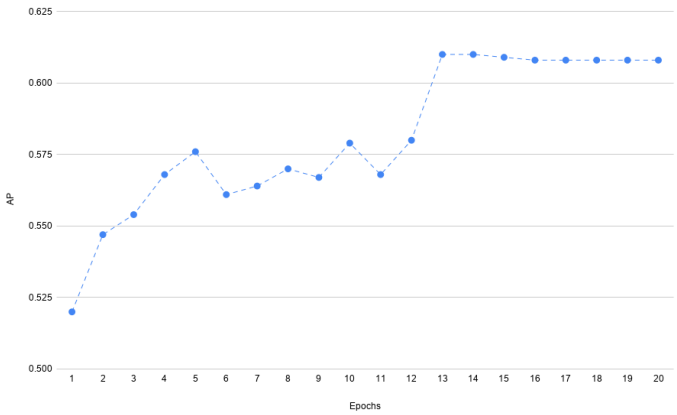

Figure 14. AP in each epoch while training a network with full training set.

After 15 iterations, the model reached a similar AP to the 100-batch model (0.498) with 63% fewer detections and 58% fewer gradient calculations. The whole training lasted 83 iterations, and the model trained during this iteration was the best one. We used 8266 samples during the training (13.7% of all available images). The comparison between the model and the model trained conventionally is presented in Table 3. For both validation and test sets the network reached over 90% of reference network AP. This shows that we can achieve similar network quality with a smaller number of samples. The comparison between detections done by the reference network and the proposed model can be seen in Figures 15 and 16. As we can see, the proposed model in most cases generates similar mappings. It performs worse only in some edge cases such as when a lot of bees are overlapping or only a small part of the insect is visible.

Table 3. Comparison of models trained conventionally and using Active Learning approach.

Approach	AP—Validation Set	AP—Test Set	No of Photos
Conventional	0.61	0.457	60,422
Active Learning	0.572	0.425	8266
Active Learning/Conventional (%)	93.8	93.0	13.7

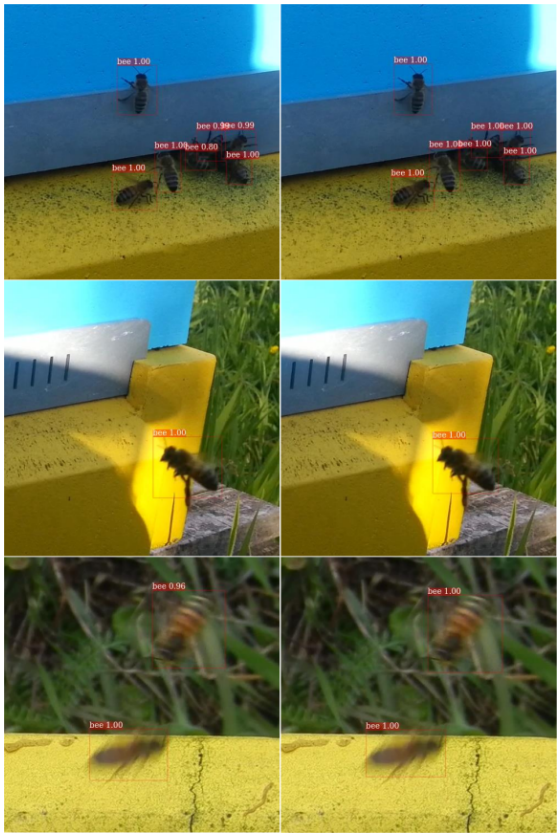

Figure 15. Bees detected by reference network (**left**) and the proposed model (**right**)—image 1.

Figure 16. Bees detected by reference network (**left**) and the proposed model (**right**)—image 2.

6. Conclusions and Future Works

In this paper, we proposed a crowdsourcing-based approach for annotated data acquisition and means to support an Active Learning training approach. The knowledge of the crowd serves here as an oracle able to judge whether the given sample is informative or not. The proposed approach was implemented as a general-purpose framework for data acquisition and verification. The system can be used for data gathering and annotation by both experts and volunteers alike, allowing the generation of high-quality data annotations without the need for special knowledge or expertise. This way, the time and cost needed to prepare training data can be limited in many everyday scenarios and fields of applications. The paper also introduced a fourth set concept to the training process. This set consists of voluntarily provided classification samples obtained during the post-deployment stage when the pre-trained network is actually used in real-life applications and allows a perpetual increase in the quality of the network thanks to new samples provided by the users. The paper also discusses hyper-parameters used during the network training. This way we were able to limit the time needed to train the network, especially in the post-deployment quality update stages. This is done by keeping the size of the new training set while still maintaining the network generality.

The final model obtained proved to be comparable with the model trained traditionally, reaching 93% of AP of the traditional model while using only 13.7% of the available training data.

In further research, we plan on using the approach not only for image-based data but also for text data (for which we already have some promising results [18,19]) and for pure data gathering, where the users are tasked with annotating data with custom tags. We also plan on exploring weakly supervised learning as an addition to the process. For that, we plan on using an approach similar to the Soft Switch proposed by Desai et al. [32]. This would allow further decreasing the number of samples needed for the training.

Author Contributions: Conceptualization, T.M.B. and J.S.; methodology, T.M.B. and A.K.; software, A.K.; validation, T.M.B., J.S. and A.K.; formal analysis, T.M.B. and A.K.; resources, J.S.; data curation, T.M.B.; writing—original draft preparation, T.M.B. and A.K.; writing—review and editing, J.S.; visualization, A.K.; supervision, T.M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All used data is available at the following locations: https://doi.org/10.34808/v23c-qx27, https://doi.org/10.34808/4aeb-1f36, https://doi.org/10.34808/vrnf-3a57 and https://doi.org/10.34808/66c8-j269.

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

References

1. Lanier, J. *Who Owns the Future?* Simon and Schuster: New York, NY, USA, 2014.
2. Mey, A.; Loog, M. Improvability through semi-supervised learning: A survey of theoretical results. *arXiv* **2019**, arXiv:1908.09574.
3. Dziubich, T.; Bialas, P.; Znaniecki, L.; Halman, J.; Brzezinski, J. Abdominal Aortic Aneurysm Segmentation from Contrast-Enhanced Computed Tomography Angiography Using Deep Convolutional Networks. In Proceedings of the ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium—International Workshops: DOING, MADEISD, SKG, BBIGAP, SIMPDA, AIMinScience 2020 and Doctoral Consortium, Lyon, France, 25–27 August 2020; Communications in Computer and Information Science; Bellatreche, L., Bieliková, M., Boussaïd, O., Catania, B., Darmont, J., Demidova, E., Duchateau, F., Hall, M.M., Mercun, T., Novikov, B., et al., Eds.; Springer: Warsaw, Poland, 2020; Volume 1260, pp. 158–168. [CrossRef]
4. Kellenberger, B.; Marcos, D.; Lobry, S.; Tuia, D. Half a percent of labels is enough: Efficient animal detection in UAV imagery using deep CNNs and active learning. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9524–9533. [CrossRef]
5. Settles, B. *Active Learning Literature Survey*; Technical report; University of Wisconsin-Madison Department of Computer Sciences: Wisconsin, WI, USA, 2009.

6. Hasenjäger, M.; Ritter, H. Active learning in neural networks. In *New Learning Paradigms in Soft Computing*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 137–169.
7. Lewis, D.D.; Gale, W.A. *A Sequential Algorithm for Training Text Classifiers*; SIGIR'94; Springer: Berlin/Heidelberg, Germany, 1994; pp. 3–12.
8. Freytag, A.; Rodner, E.; Denzler, J. *Selecting Influential Examples: Active Learning with Expected Model Output Changes*; European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2014; pp. 562–577.
9. Roy, N.; McCallum, A. *Toward Optimal Active Learning through Monte Carlo Estimation of Error Reduction*; ICML: Williamstown, Australia, 2001; pp. 441–448.
10. Sener, O.; Savarese, S. Active learning for convolutional neural networks: A core-set approach. *arXiv* **2017**, arXiv:1708.00489.
11. Makili, L.E.; Sánchez, J.A.V.; Dormido-Canto, S. Active learning using conformal predictors: Application to image classification. *Fusion Sci. Technol.* **2012**, *62*, 347–355. [CrossRef]
12. Konyushkova, K.; Sznitman, R.; Fua, P. Learning Active Learning from Real and Synthetic Data. *arXiv* **2017**, arXiv:1703.03365.
13. Desreumaux, L.; Lemaire, V. Learning active learning at the crossroads? evaluation and discussion. *arXiv* **2020**, arXiv:2012.09631.
14. Konyushkova, K.; Sznitman, R.; Fua, P. Discovering General-Purpose Active Learning Strategies. *arXiv* **2018**, arXiv:1810.04114.
15. Von Ahn, L. Games with a purpose. *Computer* **2006**, *39*, 92–94. [CrossRef]
16. Von Ahn, L.; Maurer, B.; McMillen, C.; Abraham, D.; Blum, M. Recaptcha: Human-based character recognition via web security measures. *Science* **2008**, *321*, 1465–1468. [CrossRef] [PubMed]
17. Curtis, V. Motivation to participate in an online citizen science game: A study of Foldit. *Sci. Commun.* **2015**, *37*, 723–746. [CrossRef]
18. Boiński, T.; Szymański, J. Collaborative Data Acquisition and Learning Support. In *International Journal of Computer Information Systems and Industrial Management Applications*; Springer: Cham, Switzerland, 2020; pp. 220–229. [CrossRef]
19. Boiński, T. Game with a Purpose for mappings verification. In Proceedings of the 2016 IEEE Federated Conference on Computer Science and Information Systems (FedCSIS), Gdańsk, Poland, 11–14 September 2016; pp. 405–409.
20. Szymański, J.; Boiński, T. Crowdsourcing-Based Evaluation of Automatic References Between WordNet and Wikipedia. *Int. J. Softw. Eng. Knowl. Eng.* **2019**, *29*, 317–344. [CrossRef]
21. Jagoda, J.; Boiński, T. Assessing Word Difficulty for Quiz-Like Game. In *Semantic Keyword-Based Search on Structured Data Sources: IKC: International KEYSTONE Conference on Semantic Keyword-Based Search on Structured Data Sources*; Springer: Cham, Switzerland, 2018; Volume 10546, pp. 70–79. [CrossRef]
22. Soo, S. *Object Detection Using Haar-Cascade Classifier*; Institute of Computer Science, University of Tartu: Tartu, Estonia, 2014; Volume 2, pp. 1–12.
23. Cejrowski, T.; Szymanski, J.; Logofătu, D. Buzz-based recognition of the honeybee colony circadian rhythm. *Comput. Electron. Agric.* **2020**, *175*, 105586. [CrossRef]
24. Boiński, T.; Szymański, J. Video Recordings of Bees at Entrance to Hives. 2019. Available online: https://mostwiedzy.pl/en/open-research-data/video-recordings-of-bees-at-entrance-to-hives,102910204810484 62-0 (accessed on 10 July 2020).
25. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
26. Goldman, E.; Herzig, R.; Eisenschtat, A.; Goldberger, J.; Hassner, T. Precise detection in densely packed scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5227–5236.
27. Facebook AI Research. Detectron. 2020. Available online: https://github.com/facebookresearch/detectron (accessed on 15 July 2020).
28. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
29. Deng, J.; Russakovsky, O.; Krause, J.; Bernstein, M.; Berg, A.C.; Li, F.F. Scalable Multi-Label Annotation. In Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI), Toronto, ON, Canada, 26 April–1 May 2014.
30. Käding, C.; Rodner, E.; Freytag, A.; Denzler, J. *Fine-Tuning Deep Neural Networks in Continuous Learning Scenarios*; Asian Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2016; pp. 588–605.
31. Brust, C.A.; Käding, C.; Denzler, J. Active learning for deep object detection. *arXiv* **2018**, arXiv:1809.09875.
32. Desai, S.V.; Chandra, A.L.; Guo, W.; Ninomiya, S.; Balasubramanian, V.N. An adaptive supervision framework for active learning in object detection. *arXiv* **2019**, arXiv:1908.02454.