remote sensing

MDPI

Article

Object Identification in Land Parcels Using a Machine Learning Approach

Niels Gundermann [1,2,3,*], Welf Löwe [2], Johan E. S. Fransson [4], Erika Olofsson [4] and Andreas Wehrenpfennig [3]

[1] data experts GmbH, 17033 Neubrandenburg, Germany
[2] Department of Computer Science and Media Technology, Faculty of Technology, Linnaeus University, 35195 Växjö, Sweden
[3] Department of Landscape Sciences and Geomatics, Hochschule Neubrandenburg, University of Applied Science, 17033 Neubrandenburg, Germany
[4] Department of Forestry and Wood Technology, Faculty of Technology, Linnaeus University, 35195 Växjö, Sweden
* Correspondence: niels.gundermann@data-experts.de

Abstract: This paper introduces an AI-based approach to detect human-made objects and changes in these on land parcels. To this end, we used binary image classification performed by a convolutional neural network. Binary classification requires the selection of a decision boundary, and we provided a deterministic method for this selection. Furthermore, we varied different parameters to improve the performance of our approach, leading to a true positive rate of 91.3% and a true negative rate of 63.0%. A specific application of our work supports the administration of agricultural land parcels eligible for subsidiaries. As a result of our findings, authorities could reduce the effort involved in the detection of human made changes by approximately 50%.

Keywords: land cover; object identification; agriculture land administration; change detection; machine learning

Citation: Gundermann, N.; Löwe, W.; Fransson, J.E.S.; Olofsson, E.; Wehrenpfennig, A. Object Identification in Land Parcels Using a Machine Learning Approach. *Remote Sens.* **2024**, *16*, 1143. https://doi.org/10.3390/rs16071143

Academic Editor: Dino Ienco

Received: 16 February 2024
Revised: 17 March 2024
Accepted: 20 March 2024
Published: 25 March 2024

1. Introduction

The analysis of the "physical cover of the Earth's surface" [1]–also called "land cover"–is of importance for authorities in order to manage urbanization as well as natural resources. In this context, remote sensing combined with classification systems offers an effective way to get specific information about territories under the authority's responsibility, which can be used for further decision making.

A number of articles (e.g., [2–7]) have analyzed land cover based on aerial imagery and other data sources, which can be related to both the fields of remote sensing and computer vision. There are some case studies showing the efficient utilization of machine learning (ML) techniques in remote sensing tasks, such as object classification and change detection. However, all these studies used different approaches and datasets. In [3], for example, digital orthophotos (DOPs) in combination with Sentinel-2 images and digital elevation models (DEMs) were used, whereas [8] used Landsat images. According to ML techniques, convolutional neural networks (CNNs) were used in many studies (e.g., [3,9–11]). In [3,9], a CNN was combined with recurrent neural networks (RNNs) [10], whereas [11] combined CNN with support vector machines (SVMs) to perform classification.

However, the ability of a system relying on aerial images to detect objects is constrained by the quality of the images, which can be referred to as the spatial and spectral resolution, and the geometrical correctness (orthorectification) [12]. A higher quality image comes with higher costs. Hence, it makes sense to develop systems for specific tasks, in order to adjust the quality as needed to fit the relevant objects.

One such task is the maintenance of a system for the management of agricultural parcels eligible for subsidiaries. Based on the common agricultural policy (CAP) of the European Union (EU), every member state is forced to maintain such a system for the

administration of all land parcels located in their territory [13]. This system is called the Land Parcel Identification System (LPIS), which utilize "ortho-imagery" (based on aerial or satellite images) [14] and stores the geometries and coordinates of the land parcels in a database. The LPIS database helps the member states to manage agricultural production, as well as to reach the environmental protection targets set by the EU. One common task is the detection of human made changes, e.g., buildings, streets, wind turbines, power lines etc., which has to be performed on a regular basis (i.e., every year). Consequently, this comes with a huge workload for the authorities, since there is no technical support involved.

In this study, we refer to an area covered by human made obstacles as a non-eligible area (NEA), which serves as an indicator of their impact on the subsidy calculations (in general, agricultural subsidies are based on the amount of eligible area within a parcel). Unfortunately, these NEAs are very small and difficult to detect using the human eye, depending on the image quality. The main issue related to the LPIS is the delineation of agricultural parcels. One might think that the detection of NEAs is done by systems focusing on parcel delineation. A number of articles (e.g., [15–22]) have focused on this problem, instituting the detection of a parcel boundary utilizing different ML approaches, especially CNN [15–21]. However, these studies focused on the outer boundaries of the parcels and paid minor attention to the objects located on the parcels (within the outer boundaries). This is due to that a major part of a parcel's boundary is associated with objects and areas located in the neighborhood of the parcel.

According to [22], there is only little research (e.g., [23,24]) that has focused on objects located on a parcel and their contribution to the overall delineation (inner and outer) of the parcels, which was the intended subject of the articles mentioned above. However, these studies focused on specific objects, i.e., field roads [23] or ditches and furrows [24], and did not cover the whole spectrum of NEAs (Figure 1). Additionally, none of these studies utilized neural networks.

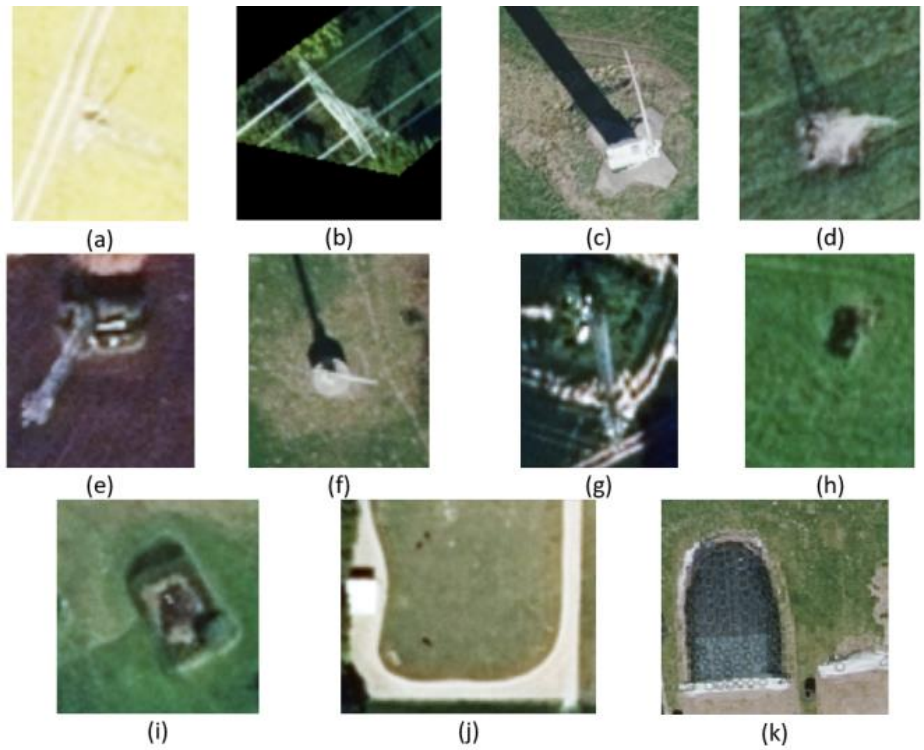


Figure 1. Objects to be detected: (**a**) Small power pole; (**b**) Huge power pole; (**c**) Wind turbine without access way; (**d**) Huge pole; (**e**) Radio pole; (**f**) Wind turbine; (**g**) Technical installation; (**h**) Water container; (**i**) Sewage plant; (**j**) Other constructions; (**k**) Animal food stock.

In this study, the objective was to develop and evaluate a system that can detect new NEAs using a CNN. Therefore, none of the solutions described in the articles mentioned above were evaluated as options for the detection of new NEAs. This evaluation has been left to future works. Additionally, we neglected the detection of removed obstacles, assuming that the farmers had a high interest in reporting these types of changes themselves, because these parcels would result in higher subsidies. Therefore, it is more important to provide the authorities with a system for the detection of new NEAs. Additionally, the evaluation of the system is more straightforward this way, since one has to consider less uncorrelated metrics when focusing on just new NEAs instead of focusing on new as well as removed NEAs. We claim that our approach could detect removed NEAs with some adjustments; however, we have left this evaluation for future work.

For the detection of new NEAs, we applied supervised ML. Since we cannot assume perfect prediction (accuracy = 100%), we must handle and balance two types of errors; Type I errors, i.e., false negatives assessed by the FN-rate (false negatives over false negatives plus true positives), and Type II errors, i.e., false positives assessed by the FP-rate (false positives over false positives plus true negatives). Based on that, we proposed an algorithm to select the best decision boundary, a threshold for predicting positives vs. negatives, according to target values of FP- and FN-rates (or their complements, TN- and TP-rate) set by a user.

According to the specification of the LPIS [25], authorities should use orthoimages to detect NEAs on a parcel. Orthoimages are geometrically corrected ("orthorectified") [26], meaning that the images are represented as if they were captured at a nadir angle instead of an oblique one [12]. However, it could be possible to detect NEAs using aerial images that are non-orthorectified. Since the orthorectification process comes with higher costs, the question arises whether it is necessary to use orthorectified aerial images for this task. Therefore, we evaluated the performance of the developed system using orthorectified and non-orthorectified aerial images. This was done in the context of managing eligible agricultural parcels in a region in the northern part of Germany.

It is hard to compare this study with others, since there are few research studies to be found in the literature that have focused on NEAs on a parcel. However, NEAs examined in this study are quite similar to the relevant objects in the related works focusing on land cover. Therefore, the results are compared to these related studies, although this comparison is still vague (Table A7, Appendix B).

2. Materials and Methods

2.1. Specific Case Study

In Germany, the federal states are responsible for the administration of agricultural subsidies. Therefore, they are also responsible for the maintenance of the LPIS, covering the area which the authorities of the federal states are responsible for.

To keep the LPIS database updated, the federal state officers run a parcel maintenance process (PMP) annually. During this PMP, they try to detect new human-made objects on the parcels and if necessary register them in the LPIS database as NEAs. The PMP is done based on DOPs, which are acquired in the year of the PMP. Since the size of some of the objects (Figure 1) is very small, the officers use DOPs with a ground sampling distance (GSD) of 50 cm. Moreover, the officers use the spectral information in the red, green, and blue bands (RGB), as well as the near infrared band (NIR).

The PMP is a two-step process (Figure 2). In the first step (Assessment), the officer iterates over the parcels ($Parcel_1 \ldots Parcel_n$) in the review step, in order to inspect the parcel for new NEAs. Therefore, each parcel is located in the DOP based on the geometry and coordinates registered in the LPIS. The officer gets an image that combines all relevant information for reviewing the parcel, i.e., the parcel's geometry, the geometries of all NEAs intersecting the parcel's geometry, and the DOP. Based on that, the officer assesses whether there is a NEA depicted in the DOP that matches both of the following two conditions:

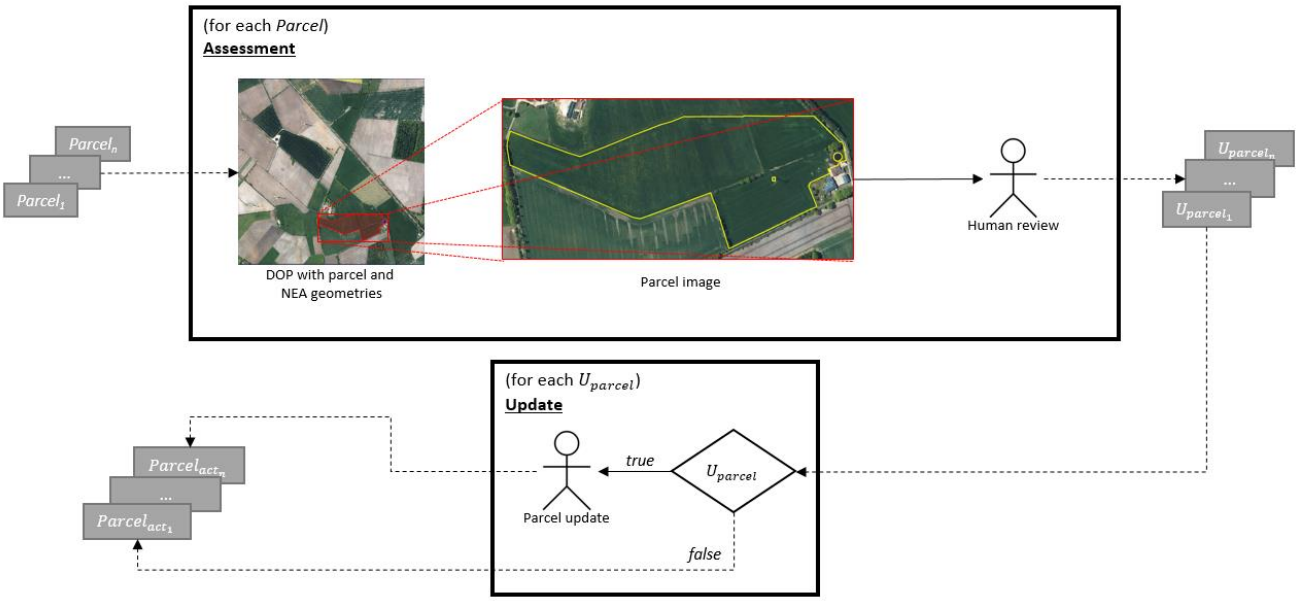


Figure 2. Parcel maintenance process of the LPIS database. First (Assessment), an officer decides for each parcel ($Parcel_1 \dots Parcel_n$) whether it needs an update or not, taking new non-eligible areas (NEA) into account. Second (Update), the officer changes the parcels marked for an update ($U_{parcel} = true$) to match the conditions in the digital orthophoto (DOP).

- The NEA is localized within the parcel geometry.
- The NEA is not yet registered in the LPIS.

If there is such a NEA, the parcel needs an update. After this step, the updated information (U) is formally stored as $U_{parcel_1} \dots U_{parcel_n}$.

In the second step (Update), another officer iterates over the information of each of the parcels to check the need for an update ($U_{parcel_1} \dots U_{parcel_n}$) and to manually verify the updates as required. If the updated information could not be verified, meaning that there is no parcel update necessary, it is dropped. Otherwise, the officer updates the geometry and coordinates of the parcel. In both cases, the parcel information captured in the LPIS database (geometry and coordinates of the NEAs) matches the actual situation, i.e., the information in the given DOPs ($Parcel_{act_1} \dots Parcel_{act_n}$).

Currently, there is no technical support for comparing parcel information with the information in the corresponding DOP; the comparison is solely done manually. Since there are, e.g., around 200,000 parcels registered in the LPIS database of Schleswig-Holstein, the first step of the update process takes a lot of human effort. According to the authorities in Schleswig-Holstein, it takes approximately two to three months of full-time work for at least three employees to complete the first step (i.e., up to nine person months annually).

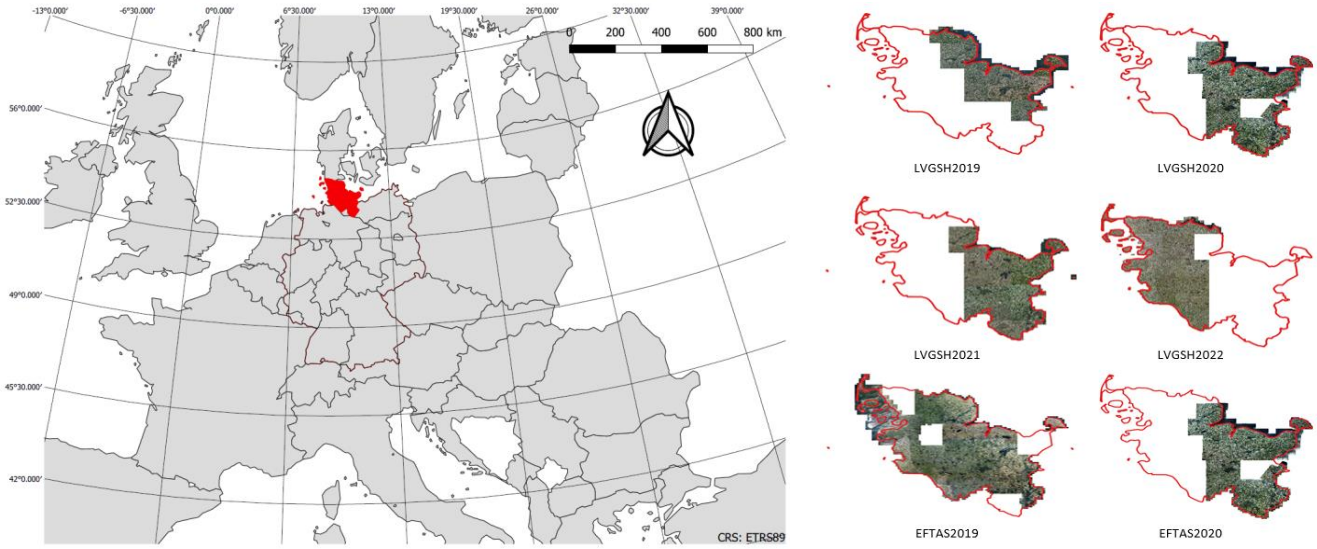
In this study, we propose an automated approach that supports the first part of the update process described above. Implemented in a system, it could automatically decide whether the given parcels need to be updated according to the DOP. To build trust in the system, we keep the humans in the loop, i.e., officers could review all parcels that needed to be updated according to the system's suggestion. Eventually, we aim for a reduction of the workload associated with manually reviewed parcels.

Together with the authorities, we decided that it is more important to suggest necessary updates by the system than it is to reject irrelevant updates. This means that reducing Type I errors is considered more important than reducing Type II errors. Hence, the target values for the classification are a true positive rate (TP-rate, suggested update was necessary, a hit) of a minimum of 90% and a true negative rate (TN-rate, unnecessary update was rejected, a correct rejection) of a minimum of 70%. This leads to a Type I error FN-rate of maximum 10%, and a Type II error FP-rate of maximum 30%.

2.2. Investigated Area and Data

2.2.1. Investigated Area

The investigated area is the federal state of Schleswig-Holstein in northern Germany (Figure 3).



(**a**) (**b**)

Figure 3. Investigated area: (**a**) The location of the investigated area (red) covering the territory of the federal state Schleswig-Holstein in northern Germany; (**b**) Coverage of the provided aerial images in relation to the investigated area. Each dataset of aerial images is associated with a company responsible for the image acquisition (LVGSH or EFTAS) and a year (2019–2022).

2.2.2. Digital Orthophotos

We used six datasets of DOPs derived from aerial photos acquired at different dates in the years 2019–2022 (Table 1). Two institutions created the datasets: Schleswig-Holstein State Office for Surveying and Geoinformation (LVGSH, Landesamt für Vermessung und Geoinformation Schleswig-Holstein), 24106 Kiel, Germany, and EFTAS Remote Sensing Technology Transfer GmbH (EFTAS, EFTAS Fernerkundung Technologietransfer GmbH), 48145 Münster, Germany. The DOPs were obtained with different quality and coverage. Here, the term quality refers to whether they are orthorectified. Coverage refers to the coverage of land area of the federal state Schleswig-Holstein (Figure 3b).

Table 1. Datasets of digital orthophotos used in this study.

Dataset	Flight Date	Spatial Resolution	Channels	Orthorectified
LVGSH2019	31 October 2019	50 cm	R, G, B, NIR	Yes
LVGSH2020	21 April 2020	50 cm	R, G, B, NIR	Yes
LVGSH2021	18 June 2021	50 cm	R, G, B, NIR	Yes
LVGSH2022	20 April 2022	50 cm	R, G, B, NIR	Yes
EFTAS2019	31 October 2019	50 cm	R, G, B, NIR	No
EFTAS2020	1 June 2020	50 cm	R, G, B, NIR	No

2.2.3. Land Parcel Identification System Database

All parcels and NEAs are stored with associated parcel information in the LPIS database. In the database, there are different versions of parcel information, which are harmonized with the DOPs in a preprocessing step, as seen in the first step in Figure 2 (DOP with parcel and NEA geometries). Therefore, we used the information from the review processes the authorities had performed in the past to associate each version of parcel

information with the DOP dataset used in the review process. As a result, we collected a specific number of parcels for each DOP dataset, as shown in Table A3, Appendix B.

2.3. Approach and Workflow

According to the PMP, described in Section 2.1, the goal was to reduce human workload by developing a system that can detect NEAs, which are not yet registered in the LPIS database. Since the current review process iterates over all parcels ($Parcel_1 \ldots Parcel_n$), the solution was integrated into this loop. This is why the whole workflow iterates over each parcel (Figure 4).

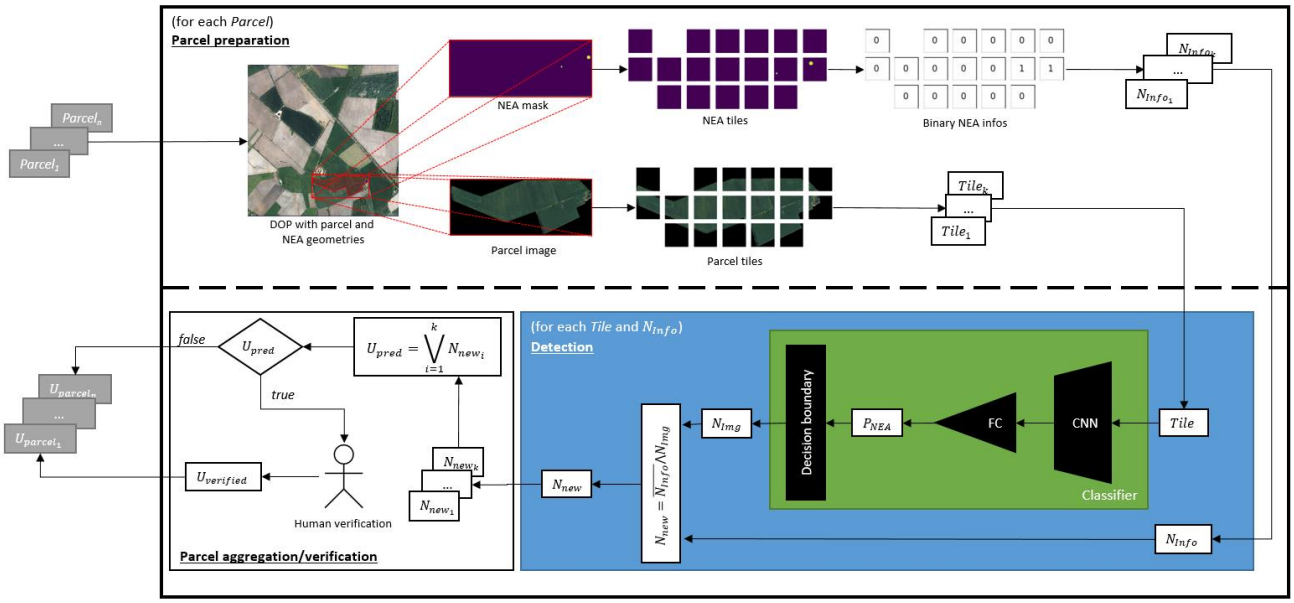


Figure 4. Overview of the workflow of the system. First (Parcel preparation), the parcel image and the location of the non-eligible areas (NEAs) are extracted and then divided in smaller parts (parcel tiles and binary NEA infos). Second (Detection), the system detects new NEAs on each parcel tile. Finally (Parcel aggregation/verification), the system predicts, whether the parcel needs an update or not (U_{pred}), based on the detection of new NEAs. If so, a human must verify this prediction ($U_{verified}$).

2.3.1. Parcel Preparation

In the first part of the system's workflow, an individual parcel was prepared for NEA detection. The parcel preparation started with the localization of the parcel in the DOP, according to the geometry and the coordinates stored in the LPIS database. Based on the localization, the DOP was cut to get an image that focused on the parcels only (parcel image). Additionally, we created a label mask for the registered NEAs (NEA mask) in the same dimensions as the parcel image.

Then, both the parcel image and the NEA mask were divided into tiles (parcel tiles and NEA tiles) of equal dimension. Note that the dimension of the tiles was the same for all images and in all iteration steps. According to the NEA information in the LPIS, each NEA tile was labeled according to the existence of a NEA within it. This resulted in binary NEA info (N_{Info}) for each NEA tile.

2.3.2. Detection of New Non-Eligible Areas

The parcel preparation was followed by the detection of new NEAs. To detect new NEAs, each pair of parcel tiles ($Tile$) and binary NEA info (N_{Info}) was iterated. During one iteration step, the parcel tile was forwarded to a neural network (classifier) consisting of convolutional neural network (CNN) layers and several fully connected (FC) layers, which proposed a probability for the existence of a NEA (P_{NEA}) in the tile. To decide whether there was a NEA depicted in the tile, the output (P_{NEA}) was transformed into binary information,

which resulted in a binary classification (N_{Img}). Here, a specific decision boundary, selected in advance based on the given target values of the TP- and TN-rates, was used. Together with the binary NEA info (N_{Info}), the decision was made as to whether a detected NEA in the tile was a new NEA (N_{new}).

2.3.3. Parcel Aggregation and Verification

After detecting (rough localization by tiling, and classification by the classifier) new NEAs on each parcel tile, this information ($N_{new_1} \ldots N_{new_k}$) was aggregated to find out whether the parcel contained a new NEA and, therefore, needed an update (U_{Pred}). After that, the two possible outcomes of U_{Pred}, i.e., true (new NEA) or false (no new NEA), were directed according to the defined balance of Type I and Type II errors. For this specific case study, the system was optimized to avoid Type I errors by trading them off against Type II errors. Hence, further verification was avoided if U_{Pred} was false (indicating no new NEA), thereby eliminating the need for parcel updates. However, a human verification for U_{Pred} was forced if indicating the opposite, which had the potential to turn out to be a Type II error. Note that if the system had been optimized to avoid Type II errors instead of Type I errors, this step would have been defined the other way around.

2.4. Training and Evaluation

There are a number of parameters that could affect the classifier, and these are described in Sections 2.6 and 2.7 in detail. Consequently, it was necessary to evaluate the different parameter configurations as well as the whole approach. The process described above was, therefore, performed with some minor changes (Figure 5).

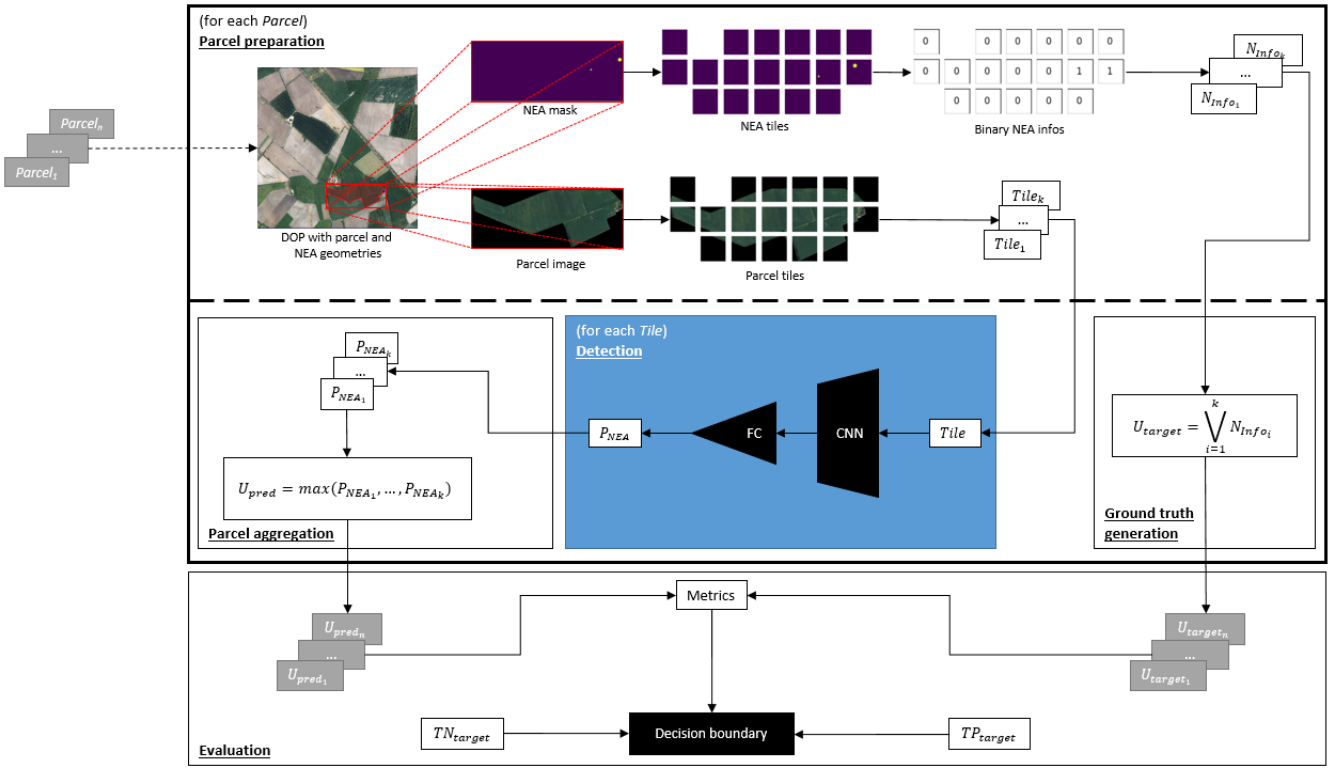


Figure 5. Overview of the evaluation process of the system. First (Parcel preparation), the parcel image and the location of the non-eligible areas (NEAs) are extracted and then divided into smaller parts (Parcel tiles and binary NEA infos). Second (Ground truth generation), the binary NEA infos are aggregated to the parcels ground truth (U_{target}). Third (Detection), the system predicts the probability of containing a NEA for each parcel tile (P_{NEA}). Next, these probabilities were aggregated for each parcel (U_{pred}). Finally (Evaluation), the metrics are calculated based on U_{pred} and U_{target}, followed by the selection of a proper decision boundary taking the target values (TN_{target} and TP_{target}) into account.

First, human verification was not part of the evaluation process, because the process should focus on the evaluation of the different variants of the classifier. Second, some ground truth was needed in order to compare it to the results the classifier produced. Therefore, the classifier was evaluated based on its detection of all NEAs that were already registered. Thus, the information about registered NEAs was retrieved from the binary NEA info ($N_{Info_1} \dots N_{Info_k}$) and aggregated to the parcel level, i.e., true or false (U_{target}), in the ground truth generation step. To create values that could be compared against the ground truth, the extracted tiles ($Tile_1 \dots Tile_k$) were forwarded to the ML model (detection), resulting in a probability (P_{NEA}) for each tile. The set of probabilities was aggregated to one value (parcel aggregation), indicating the probability of the necessity of an update for the parcel, i.e., true or false (U_{pred}), which was then compared against the ground truth, according to some metrics described below. Finally, a proper decision boundary based on the metrics and the given target TP- and TN-rates was selected.

2.4.1. Metrics

The receiver operating characteristic (ROC) curve was used to measure the goodness of the trained ML model, since the TP- and TN-rates (as well as the corresponding FP- and FN-rates) were to be balanced. The ROC curve described the TP-rate and the FP-rate (1–TN-rate) for every possible decision boundary of a classifier. Therefore, the area under the ROC (AUROC) curve was used to aggregate the target metrics as one value. The goal was to maximize this value by varying the parameters of the ML models.

Additionally, the overall accuracy was calculated for each ML model to compare the approach with other studies.

2.4.2. Decision Boundary Selection

Recall that for the case study mentioned in Section 2.1, target values of 90% for the TP-rate and 70% for the TN-rate were defined. In this case, a higher TP-rate was more important than a higher TN-rate. The ratio of differences of the target TP- and TN-rates to their maximum possible values (100%) were interpreted as weights of importance when comparing the two contradicting goals of high TP and TN. In the given case, 10% TP improvement corresponded to 30% TN improvement (100% − 90% = 10% and 100% − 70% = 30%). In other words, TP improvements were three (=30/10) times more appreciated in this study than TN improvements. Thus, the decision boundary that best fit the needs in an application context was selected in a deterministic way.

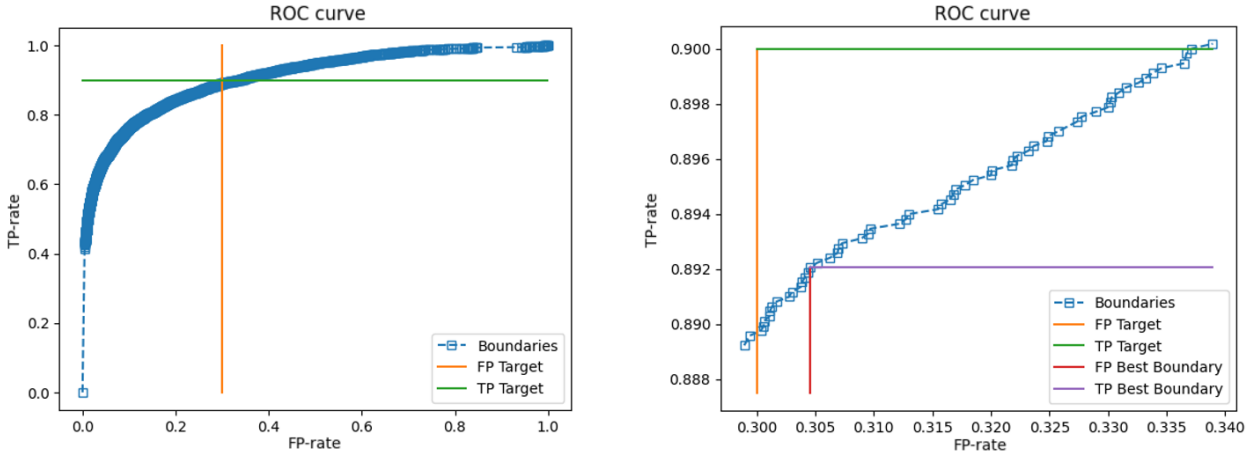
The ROC curve is a discrete function consisting of discrete points, each representing the TP- and FP-rate in the test data for a specific decision boundary. Since the test data is a finite set, the ROC curve steps are discrete decision boundaries, where at least one test data point changes from FN to TP or from TN to FP. In general, there is no decision boundary that exactly matches the target values of TP and FP (Figure 6).

Hence, the decision boundary with the best possible results given the target values as constraints (lower bound), which defines a specific classifier and its actual TP- and TN-rates, had to be found. This led to a constraint optimization problem, defined and solved as follows:

Let B be a set of all discrete decision boundaries. Define the following functions on decision boundaries $b \in B$:

$$TP(b)\text{: = TP-rate of a classifier using decision boundary } b \quad (1)$$

$$FP(b)\text{: = FP-rate of a classifier using decision boundary } b \quad (2)$$



(**a**) (**b**)

Figure 6. Receiver operating characteristic (ROC) curves for a specific classifier: (**a**) Whole ROC curve with target values; (**b**) Relevant part of ROC curve with target true positive (TP)- and false positive (FP)-rate, together with the TP- and FP-rate aimed by the best decision boundary.

The goal of the optimization was to find the best decision boundary $b_{best} \in B$ with the highest TP-rate and the lowest FP-rate (corresponding to the highest TN-rate). Additionally, the (maybe imbalanced) importance of the two target values, e.g., a weight of three for TP-rate and one for TN-rate improvements, was considered. With algebraic transformation, the optimization problem was described using the following formula:

$$b_{best} = ArgMax\left\{ TP(b) + (1 - FP(b)) \frac{100\% - TP_{Target}}{100\% - TN_{Target}} \;\middle|\; b \in B \right\} \quad (3)$$

In the case of two decision boundaries b_1 and b_2 matching the above criteria, we selected $b_{best} = max(TP(b_1), TP(b_2))$ if the TP-rate was more (or equally as) important than the TN-rate, and $b_{best} = min(FP(b_1), FP(b_2))$ if the TN-rate was more important than the TP-rate.

In our data, there were no significant performance problems related to calculating the best decision boundary by checking the optimization goal in a brute force manner, i.e., for all discrete decision boundaries of a model and a test dataset.

Figure 6a shows an example of the ROC curve produced by a specific score-based classification approach (a specific ML model) together with the target values for the TP-rate (90%) and the FP-rate (30%). Each point of the curve corresponds to a specific decision boundary for a specific classifier. Ideally, a solution is found in the top left corner, i.e., above the TP-rate line and left of the FP-rate line. Unfortunately, no decision boundary exists for the model that meets both constraints. To show the selection of b_{best}, focus is placed on the area between the points where the ROC curve crosses the target TP-rate as well as the target FP-rate (Figure 6b).

2.4.3. Cross-Validation

To narrow the confidence interval of the statistical accuracy (TP, TN) estimation, we performed cross-validation for the best model. We used six datasets of DOPs, acquired in different years and by two different institutions. The characteristics of the DOPs differs due to vegetation, weather conditions, and camera equipment used for the photography. Therefore, a six-fold cross-validation was performed, selecting one whole dataset for the test and the other five for the training of the classifier. After that, averaged metrics (i.e., TP-rate, TN-rate, overall accuracy) for the best decision boundary, as well as their standard deviation were calculated to evaluate the selected classifier.

2.5. Machine Learning Model

As mentioned in Section 2.3.2, the ML model consisted of a CNN followed by a set of FC-layers. A logistic sigmoid was used as the final activation function in the model. The FC-layers were part of the parameters and varied through the training iterations, as described in Section 2.6.

The CNN used was ResNet152V2 [27]. To apply the different parameters, the input layer and the FC-layer were changed. Here, we used four input channels, whereas the original CNN consisted of three input channels. Therefore, the network needed to be adapted to our preferences. A fourth channel was added to the input layer of the CNN; the kernel weights of this additional channel were randomly initialized. Except this, no other changes were considered to the original architecture of the ResNet152V2 model.

A training dataset was created based on the datasets described in Sections 2.2.2 and 2.2.3. According to Section 2.3.1, each tile created from the parcel images was collected, together with the binary NEA infos created from the NEA mask. Inspired by the ImageNet challenge [28] (ImageNet), we used a tile size of 224 × 224 pixels. Table 2 describes the resulting number of tiles that contain a NEA (with NEA) and those that do not (without NEA) per dataset.

Table 2. Distribution of tiles in datasets (tiles with NEA vs. tiles without NEA).

Dataset	Tiles with/without NEA
LVGSH2019	2052/430,147
LVGSH2020	6785/1,006,942
LVGSH2021	3509/584,871
LVGSH2022	5245/986,227
EFTAS2019	10,285/1,500,868
EFTAS2020	5660/846,670
Total	33,536/5,355,725

The training was performed in alternating training and validation steps. Therefore, the training dataset was separated into two batches, one for training (90%) and one for validation (10%). The training batch was used in the training step while the model was fitted. The validation batch was used in a separate validation step. In the validation step, the AUROC was calculated to measure any improvement in the model compared to the previous step. The training was considered finished if there was no improvement after four consecutive epochs, providing that at least two epochs had been completed. Moreover, the training was performed with the Adam optimizer [29] with a learning rate of 0.001 and a batch size of 100 in all iterations. As a loss function, we used binary cross-entropy loss.

In the ML models, there were two types of parameters: hyper-parameters and manually selected parameters. Hyper-parameters were used in every possible combination in each iteration, whereas the manually selected parameters were selected with special intention in each iteration. Both types of parameters are described below.

2.6. Manually Selected Parameters

2.6.1. Transfer Learning

To benefit from pretrained computer vision models, we used transfer learning (TL) in our model. Therefore, we used two different approaches. First, we used the feature extractor of a model pretrained from the ImageNet challenge. Second, we used a feature extractor from a model trained in advance using self-supervised learning (SSL) [30], which is used in many studies (e.g., [31–36]) and claims an overall potential to improve the ML models used. In SSL, one differentiates between a pretext task (term for the task performed during pre-training) and a main task that follows (in our case, the detection of NEAs in tiles). Through this, we intended to adapt the chosen model to the DOPs used for the

training of the main task. Eventually, we came up with five different models for transfer learning (Table 3). The parameters considered for the pre-training were:

Table 3. Self-supervised learning (SSL) pretrained models with the relevant parameters used for training these models.

SSL Trained Model	Trained Epochs	MPC	FC-Layer	Dataset
SSLTest_TE100_MPC0_FC-LV1	100	0	V1	SSLTest
SSLTest_TE100_MPC0_FC-LV2	100	0	V2	SSLTest
SSLAll_TE6_MPC0_FC-LV2	6	0	V2	SSLAll
SSLAll_TE11_MPC0_FC-LV2	11	0	V2	SSLAll
SSLAll_TE35_MPC50_FC-LV2	35	50	V2	SSLAll

- Trained epochs–the number of epochs used for training.
- MPC–the minimum parcel coverage in the training dataset.
- FC-layer–the version of the fully connected layer used.
- Dataset–the dataset used for training.

We used the names of the models in Table 3 for the description of the transfer learning parameters in the iterations for the training of the main task.

The details of the training process and the parameters, as well as the results of the pre-training, are described in Appendix A. Self-Supervised Learning Training.

2.6.2. Data Balancing

To balance the training data according to the classes we wanted to discriminate between, two different balancing techniques were investigated. This is particularly relevant as we focused on binary image classification with a major class (without NEA) and a minor class (with NEA). The first technique (referred to as reduce) reduced the major class by a random selection of n samples, where n equals the number of samples in the minor class. The second technique (referred to as augment(n)) duplicated random samples of the minor class n times. In order to vary the duplicated samples of the minor class, we used geometric transformation [37], i.e., a random 90-degree rotation, as an augmentation technique.

2.6.3. Training Data Selection

To find out the impact of the usage of different amounts of training data, we investigated different combinations of the given datasets, as well as different fractions of these datasets, as training data. Assume a balanced dataset named D and a percentage s of the usage of this dataset. Then D(s) denotes a subset containing s percent of random data from D. For example, LVGSH2020(50) indicates that 50% of the dataset LVGSH2020 was used as training data.

2.6.4. Input Channels

We used all four channels provided by the datasets (R, G, B, NIR). Several research studies (e.g., [4,6,8]) have shown that the normalized difference vegetation index (NDVI) comes with an advantage for classification tasks related to vegetative areas. Therefore, we combined the RGB channels with the NDVI (R, G, B, NDVI) instead of the NIR channel. The NDVI was calculated based on the near infrared channel NIR and the red channel R:

$$\mathrm{NDVI} = \frac{\mathrm{NIR} - \mathrm{R}}{\mathrm{NIR} + \mathrm{R}} \tag{4}$$

2.7. Hyper-Parameters

2.7.1. Trainable Layers

One hyper-parameter concerned the layers of the model that were trained. We varied different trainable layers of the CNN. The training determination (yes or no) for the input layer depended on the transfer learning used. If transfer learning started from a CNN

that was initially trained on the ImageNet challenge [28], the first layer needed addition training. This is because the pretrained CNN came with a three-channel input instead of a four-channel input, as used in our approach. Additionally, we varied training for the last layer, as well as for the last two layers of the CNN. Finally, we also tested and trained all CNN layers.

2.7.2. Fully Connected Layers

We used different sets of FC-layers to test whether they had a significant impact on the model's performance. In a network with fully connected layers, each layer consisted of neurons, which were connected to all neurons of the subsequent layer. We considered the following FC-layer variations:

- 3 layers with neurons: 4096, 4096, 1
- 4 layers with neurons: 4096, 4096, 1000, 1

2.8. Training Iterations

An iterative approach for the training of the classifier was used, where each iteration was divided into one main iteration and multiple sub-iterations. The main iteration was determined by the manually selected parameters described in Section 2.6. Manual selection of parameters was performed to avoid a brute force approach, which would try out too many variants and overwhelm the computing resources. The sub-iteration tested all combinations of the hyper-parameters described in Section 2.7. Since there were only a few values considered for each of the hyper-parameters, optimization techniques were not necessary. Hence, the hyper-parameters were tested using a brute force approach. Appendix B, Table A4 shows the set of main iterations, and Table A5 shows the set of sub-iterations, where an iteration is named as main-iteration.sub-iteration, e.g., 3.6 for main iteration 3 and sub-iteration 6.

We ran the training and evaluation process described in Section 2.4 for each main and sub-iteration. After determining the best decision boundary as described in Section 2.4.2, we calculated the resulting TP- and TN-rates and compared the performance of the trained models to the target values (TP- and TN-rate).

3. Results

Since we wanted to focus on the impact of the different manually selected parameters, we first selected the best sub-iteration of each main iteration. This selection was based on a comparison of the AUROC values of the sub-iterations. Figure 7 shows AUROC, TP- and TN-rates, and the overall accuracy of the best sub-iterations of each main iteration, along with the target TP- (90%) and TN-rate (70%) values (red dashed lines). The values for the metrics are outlined in Table A6, Appendix B.

Every time the target value for the TP-rate was hit, the target value for the TN-rate was missed. Although the TP-rate was quite stable, the TN-rate showed a high variability. The overall accuracy was very similar to the TN-rate. This similarity was caused by the huge imbalance of the tiles with and without NEAs, as shown in Table 2 (i.e., a very small proportion of cases were NEA-positive).

Based on the AUROC shown in Figure 7, the best model was found in iteration 15.7. Table 4 shows the results of the six-fold cross-validation of this iteration for the other performance measures. Our proposed method achieved an average TP-rate of 91.3% with a sample standard deviation of 1.0%, an average TN-rate of 63.0% with a sample standard deviation of 3.0%, and an average overall accuracy of 69.4% with a sample standard deviation of 5.5%.

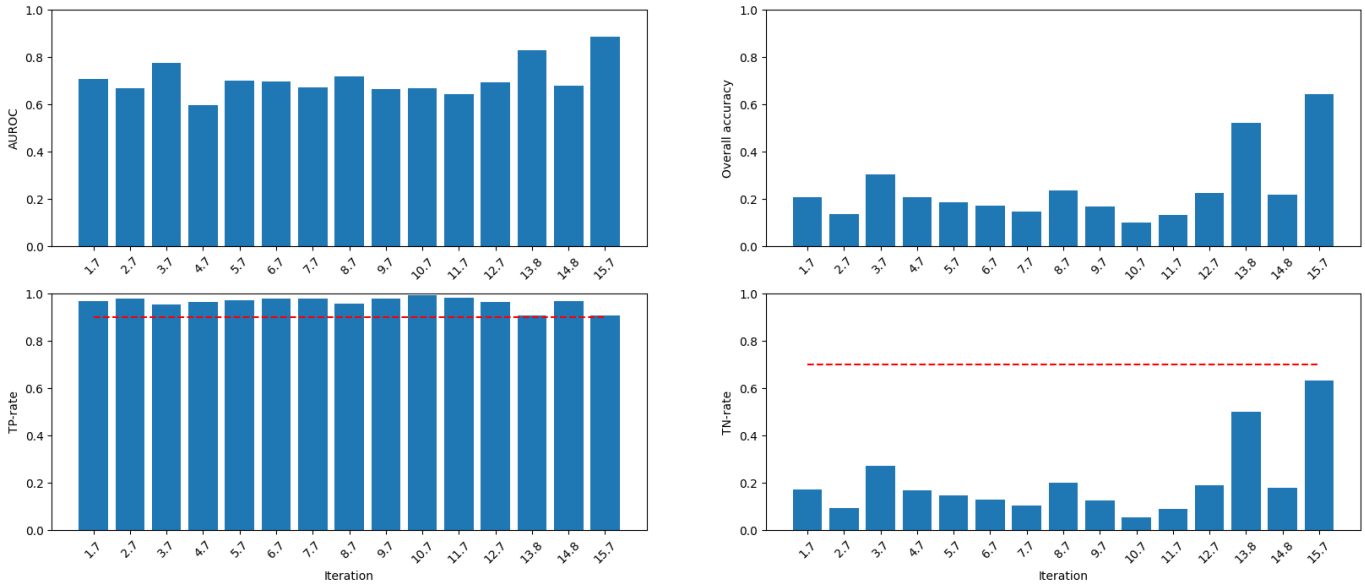


Figure 7. Metrics in terms of area under the receiver operating characteristic (AUROC) curve, true positive rate (TP-rate), true negative rate (TN-rate), and the overall accuracy per iteration. The red dashed lines indicate the target values of the TP- and TN-rate.

Table 4. Metrics in terms of decision boundary, true negative rate (TN-rate), true positive rate (TP-rate), and overall accuracy of the six-fold cross-validation of iteration 15.7.

Test Dataset	Decision Boundary	TN-Rate	TP-Rate	Overall Accuracy
LVGSH2019	0.206801	0.634	0.901	0.641
LVGSH2020	0.161092	0.675	0.919	0.664
LVGSH2021	0.174515	0.618	0.928	0.674
LVGSH2022	0.200660	0.636	0.919	0.738
EFTAS2019	0.116302	0.582	0.905	0.664
EFTAS2020	0.249435	0.631	0.908	0.664
Average	0.184801	0.630	0.913	0.694
Sample standard deviation	0.045328	0.030	0.010	0.055

4. Discussion

Regarding the hyper-parameter selection, training the last two and the first layer of the feature extractor produced the best results in every iteration. Regarding the structure of the FC-layers, the combination of three layers performed best most of the time. Only in iterations 13 and 14 did the more complex structure of FC-layers perform better.

Transfer learning of pretrained feature extractors using SSL did not perform as well as the transfer learning of the feature extractor trained in the ImageNet challenge. This is shown by a comparison of the main iterations where we used the SSL approach (9–12 and 14) with the other iterations (Figure 7).

The augmentation of the data had a positive impact on the performance. A higher augmentation factor n resulted in an improvement in performance. This is shown by comparing iterations two and three, where the augmentation factors were n = 4 and n = 22, respectively.

The training data selection had the most significant impact on performance. This is shown when comparing iterations 4, 5, 7, 13, and 15 in Figure 7, which differ from each other based on the selection of the training data.

The selection of the input channels changed depending on the best models in the main iterations. In iterations 1–6, the NDVI performed better than the NIR channel combined

with the RGB channels. With more training data, the NIR channel combined with the RGB channels performed better than the combination with the NDVI.

Cross-validation showed, on average, a TP- and TN-rate of 91.3% and 63.0%, respectively. With a sample standard deviation for the metrics of 1% (TP-rate) and 3% (TN-rate), the most important metric (TP-rate) is near our target value of 90%, whereas the TN-rate is lower than the target value of 70%. Assuming a normal distribution of all datasets of aerial images used in the update process, we conclude that 95% of all datasets would exhibit a TP-rate of at least 89.3% and a TN-rate of at least 57.0%. With a more conservative view for any other distribution, we consider Chebyshev's inequality and conclude that at least 75% of all datasets of aerial images used in the update process would exhibit the TP- and TN-rates mentioned above (i.e., TP-rate at least 89.3%, and TN-rate at least 57.0%).

Moreover, the decision boundary seems to vary a lot; the averaged decision boundary of 0.184801 exhibits a sample standard deviation of 4.5%. A closer look at the decision boundaries calculated for each test dataset shows that datasets not rectified (EFTAS2019 and EFTAS2020) especially seem to be outliers compared to the other values. Nevertheless, these experiments still show quite good metrics (in terms TP- and TN-rate).

Finally, we needed to suggest a decision boundary for the integration of the model in the PMP mentioned in Section 2.1, where it is most important to detect the majority of the relevant objects (TP). In general, a lower decision boundary comes with a smaller TN-rate and a higher TP-rate. Since the TP-rate is more important than the TN-rate according to the case study, we would rather select a lower than a higher decision boundary. Following this, we suggest the lowest decision boundary based on the average and the standard deviation. In this way, our approach ended up with a decision boundary of 0.094145 (=0.184801 $-$ 2 $\times$ 0.045328).

5. Conclusions

In this study, a system for the detection of new NEAs was proposed. Additionally, it was shown how to integrate such a system in an existing workflow performed by the authorities. The authorities can benefit from our proposed system according to the special application described in Section 2, even though the results did not hit the target values. The idea was to spare the authorities from reviewing parcels that do not require updates. Our system achieves this efficiently. According to the authorities, approximately 15% of all parcels need an update caused by the presence of a new NEA. This implies that even with a TN-rate of 57%, the workload can be reduced by approximately 50%.

Compared to the related work that focused on comparable case studies (Table A7, Appendix B), the overall accuracy of approximately 69.4% in this study was low. However, the studies are not directly comparable with respect to the objects that were to be classified or identified. To get a comparative analysis, it is necessary to evaluate traditional remote sensing techniques as well as the approaches described in the related studies for the special case study and research area used in this study.

This study focuses on particularly small objects that are difficult to identify with low-resolution satellite images. To overcome this issue, aerial images with a GSD of 50 cm were utilized for enhanced precision in object identification.

In order to improve the results achieved here, one could consider using other remote sensing data, such as LiDAR or derivatives of that, e.g., DEMs, as an input channel in addition to the RGB, NIR, and NDVI channels. In theory, the approach presented in this study is not limited to the amount of input channels since it utilized a CNN. The impact of increased spatial resolution could be analyzed as well, e.g., the impact on accuracy when detecting NEAs in aerial imagery with a GSD of 20 cm. Since imagery with higher resolution comes with higher costs, one could also investigate and evaluate this approach with remote sensing data of a lower spatial resolution such as Sentiel-2 images.

The SSL approach did not give an advantage compared to the transfer learning based on the model from the ImageNet challenge. Since the SSL approach includes two training loops and is, thus, associated with higher costs, it is not reasonable to invest in this kind of

pretraining when working on aerial images. Instead, it is more efficient and effective to use a pretrained feature extractor, like the one from the ImageNet challenge.

The results also show that it could be possible to use aerial images that are non-orthorectified. Cross-validations with the non-orthorectified datasets (EFTAS2019 and EFTAS2020) showed competitive results compared to other cross-validations based on orthorectified images (LVGSH2019-2022). However, the suggested decision boundaries for the EFTAS datasets varied more than the ones for the LVGSH datasets. Since there are many properties of the aerial images to consider, e.g., brightness, contrast, color, sharpness, temporal changes, etc., it is not clear that orthorectification has caused the variation of the decision boundary. We also need to consider that the orthorectified and non-orthorectified images were provided by different companies (Table 1).

In this study, the reviewed parcels lead to verified data, i.e. verified by a human expert, which could be used further in training iterations. Hence, we might investigate the adaptation of our approach described in Section 2.3 to collect this verified data in an easy and integrated way. As indicated by the results, this could yield better performance over time. Furthermore, a stronger focus on augmentation techniques could also lead to an improvement in the performance, as it increases the size and the diversity of the datasets. Also, it would be worthwhile to review the dataset described in this paper to find failures in the labelling. Especially when looking at false positive and false negative classified tiles, we found some inconsistent data. Based on that, one could investigate the impact of those failures.

According to the pre- and post-processing of our approach, we should mention some disadvantages that came especially with the creation of tiles. Since we created tiles that were located next to each other, it is possible that NEAs were located right at the borderline between two or more tiles. We do not know if the classifier was able to detect those partially depicted NEAs. Moreover, it was not possible to detect a new NEA on a tile where a NEA was already registered. To reduce the impact of this problem, one could investigate other techniques for the creation of tiles, e.g., overlapping the borders of the tiles, or using other machine learning tasks like semantic segmentation.

The selection of an appropriate decision boundary is crucial but risky, since we do not know how it works with new data. To avoid this critical aspect of the whole case study, a list of probabilities of necessary updates for each parcel could be provided, instead of a definite decision. In this way, the officers can decide how many parcels they want to review, beginning with the ones associated with the highest probability of requiring an update. This could reduce the Type I and Type II errors of the overall process with a human in the loop.

Note that neither the problem nor the solution is specific for a special case study or research area. Instead, the authorities of other federal states or EU countries could directly apply the proposed solution. In general, the accurate detection of human-made changes to nature has many applications, including but not limited to the detection of new buildings without permission, illegal changes to nature reserves, and the need for updates in all types of maps and plans. However, these applications would most likely require a retraining of the ML models based on remote sensing images specific to the application.

Author Contributions: Conceptualization, N.G.; methodology, N.G.; software, N.G.; validation, N.G.; formal analysis, N.G.; investigation, N.G.; resources, N.G.; data curation, N.G.; writing—original draft preparation, N.G.; writing—review and editing, W.L., J.E.S.F., E.O. and A.W.; visualization, N.G.; supervision, W.L., J.E.S.F., E.O. and A.W.; project administration, N.G.; funding acquisition, N.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by data experts GmbH, 17033 Neubrandenburg, Germany.

Data Availability Statement: The data are not publicly available due to privacy.

Acknowledgments: The data used in this article was provided by the Ministry of Agriculture, Rural Areas, European Affairs and Consumer Protection (MLLEV), 24103 Kiel, Germany.

Conflicts of Interest: Author Niels Gundermann was employed by the company data experts GmbH, 17033 Neubrandenburg, Germany. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Self-Supervised Learning Training

To adapt the feature extractor (FE) of the classifier to the data of aerial images, we performed pre-training on generated data. After completing this, we could use the FE from the pre-trained network for the training of the main task. Since we needed to generate data for the task of pre-training, also called a pretext task, we needed to use another preprocessing method for this.

This section is structured as follows. First, the pretext task that we selected is introduced. The second subsection describes the generation of the training and evaluation data. In the third and fourth section, the details of the machine learning model and the various parameters varied to train different networks, respectively, are presented. Finally, the results of each of the trained models are outlined.

Appendix A.1. Pretext Task

Similar to [35], we employed the estimation of rotation angles as a pretext task. Consequently, two images were used as inputs: the original image and a rotated version thereof. The pretext task was then to estimate the rotation angle, as shown in Figure A1 for a 180-degree rotation.

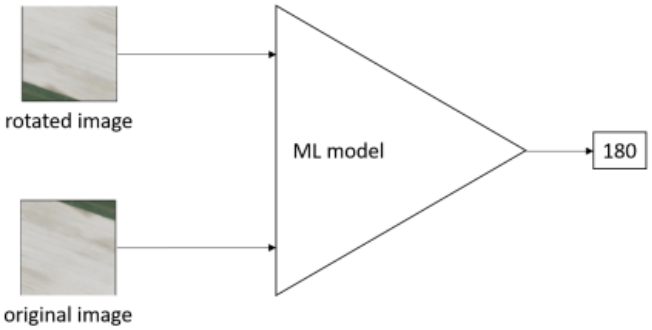


Figure A1. Pretext task of the self-supervised learning approach. Here the machine learning (ML) model predicts the rotation angle of the rotated image in relation to the original image.

We focused on multiple 90-degree rotation angles. Therefore, we only needed to distinguish between four possible outcomes in our pretext task. Hence, we were able to design a multiclass classification task as a pretext task, where the predicted class relates to the factor of 90-degree rotation for the generation of the rotated image.

Appendix A.2. Generation of Training and Evaluation Data

The generation of data for training and evaluation was quite simple, since we only rotated the data. The raw data we used were the digital orthophotos (DOPs) from the datasets described in Section 2.2.2. Here, it was important to keep the input resolution equal to the input resolution of the main task. Hence, we needed to create smaller images with a resolution equal to the resolution of the tiles described in Section 2.5. As we performed this for the tiles of a parcel, we created tiles of a whole DOP to get data for the first input (original image). Based on that, we could easily generate a rotated version of the original image. Figure A2 shows the process of data generation for one tile of a DOP.

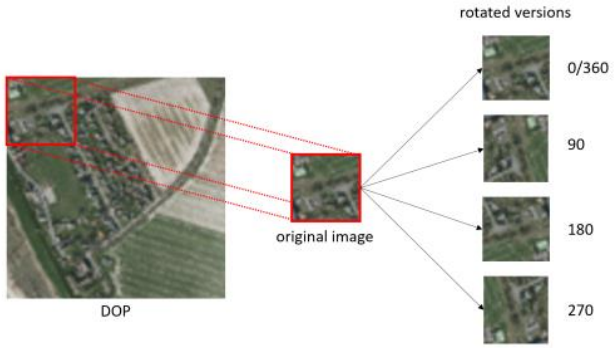


Figure A2. Data generation process for pretext task. First, a suitable tile is cut out of the digital orthophotos (DOP), used as the original image. Second, four rotated versions of the original image are created.

Appendix A.3. Machine Learning Model

Similar to the model trained with the main task, the model trained with the pretext task was also based on ResNet152 [27], which we enhanced with an additional input dimension for the NIR channel, as described in Section 2.5. During the pretext task, we first applied the FE on both images, i.e., the original image and the rotated version thereof. After we extracted the features of both images, we passed them to a projector layer, as was performed in [32]. After that, we applied a fully connected layer, which came with an output layer with four neurons and a softmax activation function. Figure A3 shows the architecture of the ML model, where we used the same FE and the same projector for both images.

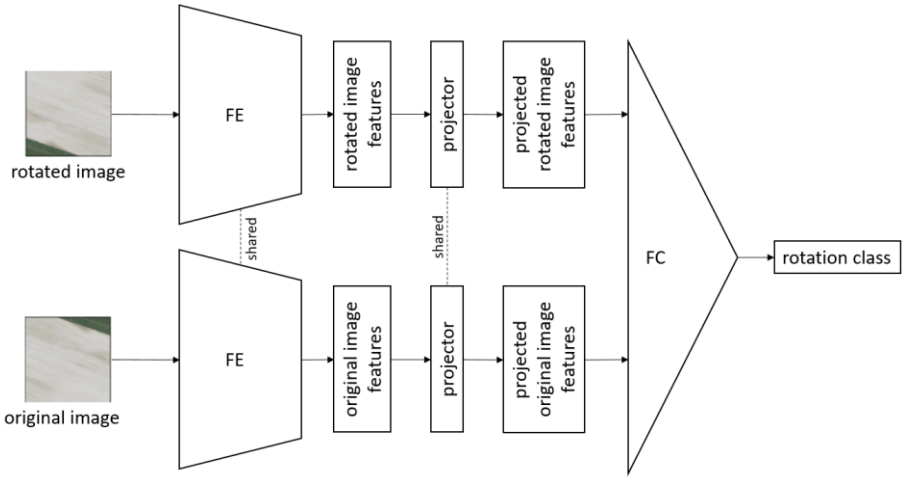


Figure A3. The machine learning model for the self-supervised learning approach. Both images (original and rotated) are forwarded to a shared feature extractor (FE), extracting feature sets for both images (original image features and rotated image features). Then, both feature sets are forwarded to a shared projector, creating a projection for both feature sets (projected original image features and projected rotated image features). Finally, the projections are forwarded to a neural network with a full connected layer (FC), resulting in the predicted rotation class.

We initialized FE using the ResNet152 model, which we pretrained from the ImageNet challenge dataset. As a loss function, we used the cross-entropy loss. Furthermore, we set up a learning rate of 0.001.

Appendix A.4. Parameters

All trained models were associated with different parameters that were carefully considered for the training process. Table 3 shows the set of trained models with the parameters described in the following subsections.

Appendix A.4.1. Self-Supervised Learning Datasets

First, we used different datasets for the generation of training and test (evaluation) data. The datasets used for the training of the pretext task (SSL dataset) consisted of the DOPs from the datasets described in Section 2.2.2. We differentiated between a training dataset, which we used for the training of the pretext task, and a test dataset, which we used to evaluate the trained model in Appendix A.3. Here, we randomly selected only a proportion of the DOPs contained by the dataset, described by the percentage values after the name of the datasets outlined in Table A1.

Table A1. The datasets (training and test) for the self-supervised learning (SSL) approach and the proportion of digital orthophotos used during training and test (evaluation) in percentage values within parenthesis.

SSL Dataset	Training Dataset	Test Dataset
SSLTest	LVGSH2022 (2.45%) LVGSH2021 (2.89%) LVGSH2019 (2.87%) LVGSH2018 (2.09%)	LVGSH2020 (2.49%)
SSLAll	LVGSH2022 (100%) LVGSH2021 (100%) LVGSH2019 (100%) LVGSH2018 (100%)	LVGSH2020 (100%)

Appendix A.4.2. Minimum Parcel Coverage

Based on the first parameter, we considered a value *MPC* (minimum parcel coverage), which described the minimal percentage coverage of parcels shown in the DOP. Therefore, let $parcel\_cov(DOP)$ describe the percentage coverage of parcels in a DOP based on the information in the LPIS database. Then, data from a DOP were generated if $MPC \leq parcel\_cov(DOP)$.

Appendix A.4.3. Fully Connected Layers

We ran the training of the pretext task with two different variations of fully connected layers. To discriminate between four possible rotation classes, the last layer contained four neurons. The structures of both versions V1 and V2 were as follows:

- V1: 4096, 6144, 2048, 1024, 4
- V2: 4096, 2048, 1000, 4

Appendix A.4.4. Trained Epochs

Considering the significant time and resources required for training the pretext task, we used a fixed number of training epochs for each trained model (Table 3).

Appendix A.5. Results

Table A2 shows the overall accuracy of the different models we trained for the pretext task.

Table A2. Self-supervised learning (SSL) pretrained models with the results (overall accuracy) according to the related test dataset.

SSL Trained Model	Overall Accuracy (%)
SSLTest_TE100_MPC0_FC-LV1	52
SSLTest_TE100_MPC0_FC-LV2	98
SSLAll_TE6_MPC0_FC-LV2	98
SSLAll_TE11_MPC0_FC-LV2	99
SSLAll_TE35_MPC50_FC-LV2	74

Appendix B. Additional Tables

Table A3. Number of parcels per digital orthophoto dataset that could be retrieved from the LPIS database.

Dataset	Number of Parcels
LVGSH2019	41,354
LVGSH2020	128,512
LVGSH2021	70,903
LVGSH2022	128,574
EFTAS2019	148,211
EFTAS2020	68,784
Total	586,338

Table A4. Main iterations with manually selected parameters.

Main-Iteration	Transfer Learning	Balancing	Training Data	Input
1	ImageNet	reduce	LVGSH2021(90)	R, G, B, NDVI
2	ImageNet	augment(4) + reduce	LVGSH2021(90)	R, G, B, NDVI
3	ImageNet	augment(22) + reduce	LVGSH2021(90)	R, G, B, NDVI
4	ImageNet	reduce	LVGSH2022(90)	R, G, B, NDVI
5	ImageNet	reduce	LVGSH2021(90) LVGSH2022(90)	R, G, B, NDVI
6	ImageNet	augment(4) + reduce	LVGSH2021(90) LVGSH2022(90)	R, G, B, NDVI
7	ImageNet	reduce	LVGSH2019(90) LVGSH2021(90) LVGSH2022(90)	R, G, B, NDVI
8	ImageNet	reduce	LVGSH2019(90) LVGSH2021(90) LVGSH2022(90)	R, G, B, NIR
9	SSL_TestV1	reduce	LVGSH2019(90) LVGSH2021(90) LVGSH2022(90)	R, G, B, NIR
10	SSL_TestV2	reduce	LVGSH2019(90) LVGSH2021(90) LVGSH2022(90)	R, G, B, NIR
11	SSL_MPC0E6	reduce	LVGSH2019(90) LVGSH2021(90) LVGSH2022(90)	R, G, B, NIR
12	SSL_MPC0E11	reduce	LVGSH2019(90) LVGSH2021(90) LVGSH2022(90)	R, G, B, NIR
13	ImageNet	reduce	LVGSH2019(90) LVGSH2020(90) LVGSH2021(90) LVGSH2022(90) EFTAS2019(90)	R, G, B, NIR
14	SSL_MPC50	reduce	LVGSH2019(90) LVGSH2020(90) LVGSH2021(90) LVGSH2022(90) EFTAS2019(90)	R, G, B, NIR
15	ImageNet	reduce	LVGSH2019(100) LVGSH2020(100) LVGSH2021(100) LVGSH2022(100) EFTAS2019(100)	R, G, B, NIR

Table A5. Sub-iterations with corresponding hyper-parameters.

Sub-Iteration	Trainable Layers	FC-Layers
1	First	4096, 4096, 1
2	First	4096, 4096, 1000, 1
3	All	4096, 4096, 1
4	All	4096, 4096, 1000, 1
5	First and last	4096, 4096, 1
6	First and last	4096, 4096, 1000, 1
7	First and two last	4096, 4096, 1
8	First and two last	4096, 4096, 1000, 1

Table A6. Metrics in terms of area under the receiver operating characteristic (AUROC) curve, true negative rate (TN-rate), true positive rate (TP-rate), and overall accuracy of the best models trained in the main iterations.

Main-Iteration.Sub-Iteration	AUROC	TN-Rate	TP-Rate	Overall Accuracy
1.7	0.708	0.170	0.967	0.209
2.7	0.669	0.094	0.979	0.137
3.7	0.775	0.271	0.953	0.304
4.7	0.595	0.168	0.963	0.207
5.7	0.701	0.146	0.971	0.186
6.7	0.698	0.130	0.978	0.171
7.7	0.673	0.105	0.978	0.147
8.7	0.716	0.198	0.959	0.235
9.7	0.663	0.125	0.979	0.167
10.7	0.668	0.055	0.993	0.101
11.7	0.641	0.089	0.981	0.133
12.7	0.693	0.188	0.963	0.226
13.8	0.828	0.501	0.908	0.521
14.8	0.679	0.178	0.969	0.217
15.7	0.886	0.631	0.908	0.664

Table A7 lists the differences between the studies (i.e., [2–4,6,8,9,11]) with respect to the following aspects of each article:

1. The main task of the case study according to image processing, which is either change detection or classification of the named objects.
2. The data sources used (if DOPs are used, the GSD as well as the angle of the sensor, i.e., nadir or oblique, is mentioned).
3. The research area the study was focused on.
4. The method used (keywords related to ML methods, or "No ML" if no ML approach was considered).
5. The overall accuracy achieved in the study.

Table A7. Related studies using remote sensing and computer vision to detect changes or to classify objects.

Reference	Case Study	Sources of Data	Research Area	Method	Overall Accuracy
[4]	Classification (Trees, Grassland, Barren land, Roads, Building, Water)	Satellite images SAT-4 and SAT-6	California (USA)	CNN	99.4%
[3]	Change detection + Classification (Urban, Vegetation, Water)	Satellite images Sentinel-2 DOPs (nadir, GSD: 10 cm) DEMs (digital elevation models)	North Rhine Westphalia (Germany)	CNN + RNN	87.8%

Table A7. *Cont.*

Reference	Case Study	Sources of Data	Research Area	Method	Overall Accuracy
[2]	Classification (Impervious surface, Pervious surface)	DOPs (nadir, GSD: 90 cm) DEMs	Wuppertal (Germany)	Random Forest	92.3%
[8]	Classification (Crop field, No crop field)	Satellite images Landsat (WELD)	Texas, California, South Dakota (USA)	No ML	90.1%
[6]	Classification (Background, Construction, Cultivated land, Woodland, Grassland, Water)	Satellite images	China, Japan, Vietnam, France	CNN	88.4%
[9]	Classification (Damaged building, Intact building)	Satellite images DOPs (nadir and oblique, GSD: 12–18 cm and 2–10 cm)	Ecuador, Haiti, Taiwan, Katmandu	CNN + RNN	94.4%
[11]	Classification (Residential, River, Highlands, Permanent Crop, Industrial, Posture)	Satellite images	India	CNN + SVM	98.0%
Ours	Change detection + Classification (with NEA, without NEA)	DOPs (nadir and oblique, GSD: 50 cm)	Schleswig-Holstein (Germany)	CNN + FC	69.4%

References

1. European Comission. *Manual of Concepts on Land Cover and Land Use Information Systems*; Office for Publications of the European Communities: Luxembourg, 2001. Available online: https://ec.europa.eu/eurostat/web/products-manuals-and-guidelines/-/ks-34-00-407 (accessed on 22 February 2024).
2. Langenkamp, J.-P.; Rienow, A. Exploring the Use of Orthophotos in Google Earth Engine for Very High-Resolution Mapping of Impervious Surfaces: A Data Fusion Approach in Wuppertal, Germany. *Remote. Sens.* **2023**, *15*, 1818. [CrossRef]
3. Sandmann, S.; Hochgürtel, G.; Piroska, R.; Steffens, C. Cop4ALL NRW—Ableitung der Landbedeckung in Nordrhein-Westfalen mit Fernerkundung und künstlicher Intelligenz. *ZfV* **2022**, *5*, 299–310.
4. Sasidhar, T.; Sreelakshmi, K.; Mt, V.; Vishvanathan, S.; Kp, S. Land Cover Satellite Image Classification Using NDVI and SimpleCNN. In Proceedings of the 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019.
5. Shin, N.; Saitoh, T.M.; Takeuchi, Y.; Miura, T.; Aiba, M.; Kurokawa, H.; Onoda, Y.; Ichii, K.; Nasahara, K.N.; Suzuki, R.; et al. Review: Monitoring of land cover changes and plant phenology by remote-sensing in East Asia. *Ecol. Res.* **2023**, *38*, 111–133. [CrossRef]
6. Yang, C.; Hou, J.; Wang, Y. Extraction of land covers from remote sensing images based on a deep learning model of NDVI-RSU-Net. *Arab. J. Geosci.* **2021**, *14*, 2073. [CrossRef]
7. Yang, Z.; Zhang, W.; Wang, W.; Xu, Q. Change detection based on iterative invariant area histogram matching. In Proceedings of the 9th International Conference on Geoinformatics, Geoinformatics, Shanghai, China, 6 January 2011.
8. Yan, L.; Roy, D.P. Automated crop field extraction from multi-temporal Web Enabled Landsat Data. *Remote Sens. Environ.* **2014**, *144*, 42–64. [CrossRef]
9. Chavhan, G.K.; Singh, S.; Doke, A.S.; Pilli, D.; Kailasavalli, S.; Ghamande, M.V. Identification of building damages using satellite images based on CNN-recurrent neural network approach. In Proceedings of the 8th International Conference on Communication and Electronics Systems, Coimbatore, India, 1–3 June 2023.
10. Medsker, L.R.; Jain, L. Recurrent neural networks. *Des. Appl.* **2001**, *5*, 2.
11. Veerraju, V.; Sirisha, M.; Kumar, K.S.; Priya, J.G.; Lekha, K.S.; Kumar, K.Y.N. Data Classifying Techniques to classifiy Satellite Images. *Ind. Eng. J.* **2023**, *52*, 2589–2597.
12. Lillesand, T.; Kiefer, R.W.; Chipman, J. *Remote Sensing and Image Interpretation*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
13. Počivavšek, G.; Ljuša, M. Characteristics of the Land Parcel Identification System (LPIS) as the main subcomponent of the Agriculture Information System. In Proceedings of the 23rd International Scientific-Experts Congress on Agriculture and Food, Izmir, Turkey, 27 September 2013.
14. European Court of Auditors. The Land Parcel Identification System: A useful tool to determine the eligibility of agricultural land—but its management could be further improved. *Off. J. Eur. Union* **2016**, *C 396*, 5.
15. Garcia-Pedrero, A.; Lillo-Saavedra, M.; Rodriguez-Esparragon, D.; Gonzalo-Martin, C. Deep Learning for Automatic Outlining Agricultural Parcels: Exploiting the Land Parcel Identification System. *IEEE Access* **2019**, *7*, 158223–158236. [CrossRef]
16. Liu, S.; Liu, L.; Xu, F.; Chen, J.; Yuan, Y.; Chen, X. A deep learning method for individual arable field (IAF) extraction with cross-domain adversarial capability. *Comput. Electron. Agric.* **2022**, *203*, 107473. [CrossRef]
17. Xu, L.; Ming, D.; Du, T.; Chen, Y.; Dong, D.; Zhou, C. Delineation of cultivated land parcels based on deep convolutional networks and geographical thematic scene division of remotely sensed images. *Comput. Electron. Agric.* **2022**, *192*, 106611. [CrossRef]

18. Zhu, Y.; Pan, Y.; Hu, T.; Zhang, D.; Zhao, C.; Gao, Y. A generalized framework for agricultural field delineation from high-resolution satellite imageries. *Int. J. Digit. Earth* **2024**, *17*, 2297947. [CrossRef]
19. Luo, W.; Zhang, C.; Li, Y.; Yan, Y. MLGNet: Multi-Task Learning Network with Attention-Guided Mechanism for Segmenting Agricultural Fields. *Remote Sens.* **2023**, *15*, 3934. [CrossRef]
20. Gui, S.; Song, S.; Qin, R.; Tang, Y. Remote Sensing Object Detection in the Deep Learning Era—A Review. *Remote Sens.* **2024**, *16*, 327. [CrossRef]
21. Shen, Q.; Deng, H.; Wen, X.; Chen, Z.; Xu, H. Statistical Texture Learning Method for Monitoring Abandoned Suburban Cropland Based on High-Resolution Remote Sensing and Deep Learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 3060–3069. [CrossRef]
22. Wang, X.; Shu, L.; Han, R.; Yang, F.; Gordon, T.; Wang, X.; Xu, H. A Survey of Farmland Boundary Extraction Technology Based on Remote Sensing Images. *Electronics* **2023**, *12*, 1156. [CrossRef]
23. Ren, Y.; Pan, Y.; Liu, Y.; Tang, X.; Gao, B.; Gao, Y. Evaluation Method of Field Road Accessibility Based on GF-2 Satellite Imagines. In Proceedings of the 7th International Conference on Agro-geoinformatics (Agro-geoinformatics), Hangzhou, China, 6–9 August 2018.
24. Ayana, E.; Fisher, J.; Hamel, P.; Boucher, T. Identification of ditches and furrows using remote sensing: Application to sediment modelling in the Tana watershed, Kenya. *Int. J. Remote Sens.* **2017**, *38*, 4611–4630. [CrossRef]
25. Valentina, S.; Wim, D. *LPIS Core Conceptual Model: Methodology for Feature Catalogue and Application Schema*; JRC49818; OPOCE: Luxembourg, 2008.
26. Orthoimagery. Available online: https://inspire.ec.europa.eu/theme/oi:3 (accessed on 21 December 2023).
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
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. Diederik, P.K.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 22 December 2014.
30. Balestriero, R.; Ibrahim, M.; Sobal, V.; Morcos, A.; Shekhar, S.; Goldstein, T.; Bordes, F.; Bardes, A.; Mialon, G.; Tian, Y.; et al. A Cookbook of Self-Supervised Learning. *arXiv* **2023**, arXiv:2304.12210.
31. Ali, M.; Hashim, S. Survey on Self-supervised Representation Learning Using Image Transformations. *arXiv* **2022**, arXiv:2202.08514.
32. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations. In Proceedings of the ICML'20: International Conference on Machine Learning, Virtual, 13–18 July 2020.
33. Falcon, W.; Cho, K. A Framework for Contrastive Self-Supervised Learning And Designing A New Approach. *arXiv* **2020**, arXiv:2009.00104.
34. Kolesnikov, A.; Zhai, X.; Beyer, L. Revisiting Self-Supervised Visual Representation Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 25 January 2019.
35. Xu, H.-M.; Liu, L.; Gong, D. Semi-supervised Learning via Conditional Rotation Angle Estimation. In Proceedings of the Digital Image Computing: Techniques and Applications (DICTA), Adelaide, Australia, 29 November–1 December 2021.
36. Zhai, X.; Oliver, A.; Kolesnikov, A.; Beyer, L. S4L: Self-Supervised Semi-Supervised Learning. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 9 May 2019.
37. Xu, M.; Yoon, S.; Fuentes, A.; Park, D.S. A Comprehensive Survey of Image Augmentation Techniques for Deep Learning. *Pattern Recognit.* **2023**, *137*, 109347. [CrossRef]