Exploiting Web Images for Moth Species Classification

Julia Böhlke [1], Dimitri Korsch [2], Paul Bodesheim [3], Joachim Denzler [4 5 6]

Abstract: Due to shrinking habitats, moth populations are declining rapidly. An automated moth population monitoring tool is needed to support conservationists in making informed decisions for counteracting this trend. A non-invasive tool would involve the automatic classification of images of moths, a fine-grained recognition problem. Currently, the lack of images annotated by experts is the main hindrance to such a classification model. To understand how to achieve acceptable predictive accuracies, we investigate the effect of differently sized datasets and data acquired from the Internet. We find the use of web data immensely beneficial and observe that few images from the evaluation domain are enough to mitigate the domain shift in web data. Our experiments show that counteracting the domain shift may yield a relative reduction of the error rate of over 60%. Lastly, the effect of label noise in web data and proposed filtering techniques are analyzed and evaluated.

Keywords: Web Images; Webly Supervised Learning; Label Noise; Noise Filtering; Species Classification; Fine-grained Recognition; Moth Scanner

1 Introduction

As our environment and climate changes, habitats are destroyed, leading to a decline in plant and animal biodiversity. To effectively combat this loss of species on our planet, monitoring these trends and understanding the reasons is paramount. Moths seem to be especially sensitive to the changes in the environment, as they are affected by an extremely rapid decline in population sizes. While the overall declining trend is undeniable, evidence suggests the situation is complex and heterogeneous for differing species [Wa21, Ha19]. In this work, we analyze different aspects of visual moth species classification to support the continuous monitoring of populations with minimally invasive methods. A clear picture of population changes for specific moth species could help biologists and conservationists make informed decisions to counteract this problem. A moth species classification system is developed as part of the biodiversity monitoring stations in the AMMOD project[7], where it is referred to as the moth scanner. The hardware setup consists of a light trap that attracts moths using ultraviolet light to illuminate a white screen and a camera regularly taking

[1] Computer Vision Group, Friedrich Schiller University Jena, 07737 Jena, Germany; julia.boehlke@uni-jena.de
[2] Computer Vision Group, Friedrich Schiller University Jena, 07737 Jena, Germany; dimitri.korsch@uni-jena.de
[3] Computer Vision Group, Friedrich Schiller University Jena, 07737 Jena, Germany; paul.bodesheim@uni-jena.de
[4] Computer Vision Group, Friedrich Schiller University Jena, 07737 Jena, Germany; joachim.denzler@uni-jena.de
[5] German Aerospace Center (DLR), Institute for Data Science, Mälzerstraße 3, 07745 Jena, Germany
[6] Michael Stifel Center Jena for Data-Driven and Simulation Science, Ernst-Abbe-Platz 2, 07743 Jena, Germany
[7] `https://ammod.de`

images of this screen during the night. To automatically analyze the vast amount of recorded images, especially when multiple stations are in operation across the country, the software components of the monitoring system require a classification model for identifying different moth species. Although there are most likely many insects in a single image that need to be localized to classify each individual separately, we only focus on the moth species classification in this paper.

The main challenge for training a good species classifier is the availability of a large training dataset with many example images for each individual species. This especially holds when considering modern deep neural networks with their large number of parameters. In addition, not only the plain images of moths are important but also the appropriate species labels, which can often only be provided by few experts with a corresponding background in taxonomy and systematics. This leads to different annotation conditions as for other visual object recognition tasks, such as the distinction of cars from motorcycles, for which most humans could provide class labels. Note that the previous challenges are relevant for most species classification problems if very similar animals need to be distinguished by small details to obtain the correct label, but they are especially important when considering a certain niche such as moths.

While, in general, citizen science applications or other crowd-sourcing activities are useful to collect additional training data, their benefit is limited when expert knowledge is required for image annotations. An active learning approach is more suited in this situation, where an expert labels images selected by a classification model. With this approach, only a small relevant subset of the raw data is labeled such that the expensive manual effort by experts is reduced. However, an initial small training dataset is needed to build an acceptable classification model that selects the images for labeling [Kä16].

To improve our understanding of the required size of a dataset for acceptable performance, we analyze how the number of images used for training affects the performance of a classification model. We use image search engines to acquire a large dataset from the Internet and analyze different aspects of the use of web data, such as the *domain shift*. A domain shift refers to the differing data distributions in the training and evaluation data, which is expected when acquiring training data from the Internet. We compare a webly-supervised approach, where only web data is used for training, with a semi-supervised approach, where cleanly labeled data and web data together form a merged training dataset.

Because the images retrieved from the Internet using image search engines are weakly labeled through tags and text content surrounding the images, the downloaded data contains label noise. As identified by Krause et al. [Kr16], who showed the potential of noisy data from the Internet for fine-grained recognition, there are two different types of label noise that can be differentiated with the definition of the dataset *domain*. All categories in a fine-grained dataset are sub-categories of a larger umbrella category, which is called the domain of the dataset. In our study, we focus on species identification for moth images, and hence, the domain of the dataset we are using is *moths*. Another example is the CUB200-

2011-dataset [Wa11] containing images of 200 different bird species, for which the domain would be *birds*. Using this definition, we can differentiate between within-domain label noise, called *cross-class noise*, and out-of-domain label noise, also called *cross-domain noise*. Usually, when using images from the Internet, the overlap between the evaluation dataset and web data needs to be eliminated to ensure a fair and robust evaluation. Because the evaluation dataset used in this paper is not publicly available, this aspect of using web data is not relevant here. However, the interested reader is referred to our previous work [Bö21] that addresses this issue.

Lastly, we want to analyze how filtering label noise in a preprocessing step affects classification performance. To handle cross-class noise, Krause et al. [Kr16] removed exact duplicates between different classes in the web data because their true label is ambiguous. We expand on this approach by considering near duplicates with our duplicate detection method. Also, we evaluate our previously proposed methods [Bö21] for cross-domain noise filtering using manually acquired annotations of label noise.

2 Related Work

In this section, we review three topics related to our work: fine-grained classification (Sect. 2.1), using images from the Internet for training a classification model (Sect. 2.2) and identifying duplicate images (Sect. 2.3).

2.1 Fine-grained Classification

The research field of fine-grained classification tackles the problem of distinguishing very similar subcategories with small inter-class variance, such as different moth species, for example. The part-based approaches [GLY19, KBD19] tackle this problem by identifying local regions in the images containing the distinct features such that the classification model can focus on the right regions. On the other hand, global approaches use the entire image for classification and thereby avoid the complex problem of extracting parts from the images. Global approaches may rely on a specific feature pooling method [LRM15, Si17] or a clever pre-training strategy [Cu18], to name a few examples. As this paper focuses on analyzing the impact of web images and dataset size, we employ the straightforward global approach in our experiments to compare different scenarios and filtering methods.

2.2 Images from the Internet and Label Noise Handling

Utilizing the vast visual information of the Internet accessible through image search engines is a data-driven approach to classification. Unfortunately, due to the weak labeling of images on the Internet, this information is tainted by label noise.

To handle label noise, one set of methods applies the concept of multiple instance learning, either learning an additional de-noising network for a fine-grained setup [Ya20] or using a grouping strategy together with attention mechanisms that set the training focus to the non-noisy samples [Zh17, Pe20]. Another approach was proposed by [Kr16] where a web dataset with more categories than the target dataset was used to pre-train a network for fine-grained classification. We expand their approach to handle cross-class noise (Sect. 3.3).

Several approaches rely on a small training set with high-quality labels to build a clean dataset from noisy data. They train an initial model, and iteratively update it by using non-noisy images selected by the model itself [Zh20b, Xu15, CSG13]. These approaches require an initial representative dataset to detect hard-to-classify instances, which increases the visual diversity for a specific class. In contrast, other methods start with a noisy dataset and filter the noise during training or dynamically re-weight noisy samples such that they have less impact during the model update [Re18]. Noisy instances can be either identified using a threshold for the cosine similarity to representations of class centers [Zh20b] or using the loss, which is typically higher for noisy instances, especially at early stages of the training [Zh20a]. Assuming that the predictions of noisy samples change more rapidly for consecutive training epochs compared to clean samples, a large cross-entropy of corresponding class probability vectors can also be an indicator for noise [Li21].

A metric learning-based approach for fine-grained recognition has been proposed by [Cu16], which requires feedback of domain experts during training via a human-in-the-loop concept. In situations where an initial training set with part annotations exists, these can be exploited in a transfer learning approach to detect noisy samples [Xu18] since outliers likely obtain low detection scores when applying both a whole-object detector as well as individual part detectors. In contrast, we focus on automatic noise filtering methods that neither require domain experts in the training loop nor training data with part annotations.

Prominent work on sample selection as a pre-processing step, proposed in frameworks such as co-teaching [Ha18], de-coupling [MSS17], and MentorNet [Ji18] train two networks and, for example, use the disagreement between them to identify the noisy instances. Unfortunately, this is a computationally expensive approach. Further data cleansing approaches have been introduced, which utilize class-wise auto-encoders [ZT19], or a variety of different ensemble methods [Ga16]. In contrast, an unsupervised approach was proposed by [Ni15] who used k-means clustering, an algorithm entirely independent of labels and, therefore, also independent of label noise. They repeatedly cluster the noisy data and use cluster statistics for identifying and relabeling noisy instances. We use a simplified version of this clustering-based idea to filter cross-domain noise as described in Section 3.2.

2.3 Duplicate Detection

To tackle the issue of cross-class noise, Krause et al. [Kr16] removed duplicates among different classes in the web data because their true category is ambiguous. We expand on

this approach by considering *near-duplicates* as well, i.e., image pairs that originate from the same source but differ on a pixel level due to small transformations. Further, near-duplicate detection is important when using publicly available datasets for the evaluation, where overlap between training and testing subsets might occur. The task of near-duplicate detection is non-trivial, especially when efficiency plays a crucial role since the number of comparison operations grows with the number of samples in a dataset.

Originally, efficient duplicate detection was done by comparing hand-crafted features extracted from the images [Ke04, Wa06]. Feature representations of images learned by a convolutional neural network (CNN) offer an alternative to the hand-crafted features, as shown by [BD20]. They used L^2-normalized feature representations extracted for all images from a CNN pre-trained on the respective training set to compare images. In metric learning approaches, such as the deep ranking method proposed by [Wa14], an embedding of the images in a lower-dimensional space is learned using a task-specific dataset. In the embedded space, similar images are located close to each other, and dissimilar ones are far apart. We employ a variant of the idea proposed by [BD20] utilizing a CNN pre-trained on ImageNet for feature extraction and also use the structural similarity measure (SSIM) [Wa04] discussed further in Section 3.3.

3 Methods

We describe our method of web data acquisition and the two typically occurring variants of label noise in Section 3.1. Two methods for filtering label noise are explained in Sections 3.2 and 3.3, respectively.

3.1 Acquisition of Web Data

Given a small labeled training dataset, also called *seed dataset* in the following, we want to download images for the same classes. The class names, i.e., species names, for this seed dataset are used as keywords in Google Image Search, and the results are downloaded for each class. This process is automated using the Google Cloud Platform services. The dataset of downloaded images will also be referred to as the *augment dataset*.

As mentioned previously, two different types of label noise can be identified in web data. An image of a map, caterpillar, or habitat would be an example of cross-domain noise for the moth species dataset, while an image depicting *Acronicta Aceris* downloaded for the class of *Acronicta Leporina* would be an example of cross-class noise. In the following section, the method for handling cross-domain noise is described, while details on our method for mitigating cross-class noise are given in Section 3.3. The methods described in both sections are used to decide which of the images are instances of label noise and should be discarded.

3.2 Cross-domain Noise Filtering

For cross-domain noise filtering, we want to identify images that do not depict the domain of the seed dataset. The method described in this section was first proposed in our previous work [Bö21]. In theory, features extracted from a CNN of images in the augment set that belong to the same domain as the seed dataset will have smaller distances to features of images from the seed dataset than out-of-domain images. We utilize this observation using a clustering-based approach. We estimate clusters jointly for feature representations of images in the augment and the seed training set. The clusters that contain a certain amount of the seed dataset indicate clusters of images belonging to the domain (*positive clusters*). Images from the augment dataset in clusters with a high proportion of seed data are likely to belong to the same domain as the seed dataset. Augment images in clusters with few to none of the original seed data, on the other hand, are more likely to depict out-of-domain objects, i.e., cross-domain noise. Therefore, we retain images in the positive clusters for training while the remaining images are filtered out.

We distinguish between two different types of positive clusters. A *strong positive cluster* is defined by containing more than $\frac{N}{k}$ samples of the seed dataset, when k clusters are estimated, and N is the total number of images in the seed dataset. The adaptive threshold ensures that at least some strong positive clusters are identified, even when clustering with a large value for k. Preliminary experiments showed that with a visually homogeneous dataset, the features of the seed dataset are too similar and are clustered together in few clusters. In this situation, using images from strong positive clusters only is not enough to identify within-domain images. Therefore, we further define *weak positive clusters* as clusters that are closer to a strong positive cluster than the average pairwise distance between all cluster centers. These clusters are likely to contain images not exactly within the narrow domain of the homogeneous seed dataset but still depict objects from the wider domain the seed dataset defines. These images are valuable in a training set, as they increase the diversity of representations a model can learn for a given class.

Our filtering method does not rely on relatively low levels of label noise to iteratively build an initial model with which non-noisy images are selected. In contrast, the clustering aspect is entirely independent of labels and, therefore, of the level of label noise. Furthermore, with higher levels of label noise, features from noisy images are more likely to be identified as negative clusters. Our previous work showed that, indeed our method is robust to very high levels of label noise [Bö21].

3.3 Cross-class Noise Filtering and Duplicate Detection

Cross-class label noise is arguably the most problematic in a fine-grained domain where only experts can identify wrongly labeled instances. This problem was handled in [Kr16] by filtering out all images that had an exact duplicate in a different class of the augment set.

This approach does not precisely filter cross-class noise but instead mitigates the issue by removing ambiguous images. We expand on this idea by also taking near-duplicates among different classes into account.

We utilize two similarity measures for our near-duplicate detection method. First, the structural similarity index (SSIM) [Wa04] is used, a pixel-based similarity measure taking luminance, structure, and contrast distortions into account. Second, we use the dot product (Dot) of L^2-normalized feature vectors extracted for the images, which is equivalent to the cosine similarity of the features, which was used by Barz et al. [BD20] for their duplicate detection method. The idea behind using these two similarity measures is that the first (SSIM) will be able to detect small transformations applied to an original image, while the second (Dot) will catch broader transformations, which the SSIM does not pick up.

The similarity measures are used to rank the images in the augment dataset, such that images at the top of the ranking are more likely to have a duplicate in a different class of the augment dataset. Further details on how this ranking is extracted using the similarity measures are found in our previous work [Bö21]. The computation of the similarity measures is omitted for images with exact duplicates identified by comparing MD5 hashes[8] of the images. The parameter *portion* (p) indirectly specifies how many images are discarded from the top of the ranking in proportion to the number of exact duplicates. When $p = 0.0$, only exact duplicates are discarded in cross-class noise filtering. This parameter portion defines some prior knowledge about the level of near duplicates in comparison to exact duplicates.

4 Experiments and Results

The effect of differently sized training datasets and different aspects of the utilization of web images are examined in our experiments. All of them are performed using the EU-Moths dataset described in Section 4.1. In Section 4.2, we analyze the effectiveness of web images and the impact of the domain shift. In Section 4.3, we use manually acquired annotations of cross-domain noise in the downloaded data to evaluate the effect of noise on classification accuracies. Finally, we discuss the effectiveness of filtering strategies in Section 4.4.

4.1 Datasets

The **EU-Moths Dataset**[9] contains manually taken images of 200 common moth species found in Central Europe collected by the Zoological Research Museum Alexander König, Bonn. The moths were photographed on a rather homogeneous, mostly white background, sometimes together with multiple other insects. The entire dataset consists of approximately 11 images per class. We cropped out the ground truth bounding boxes such that a moth fills

[8] https://tools.ietf.org/html/rfc1321

[9] http://www.inf-cv.uni-jena.de/eu_moths_dataset.html

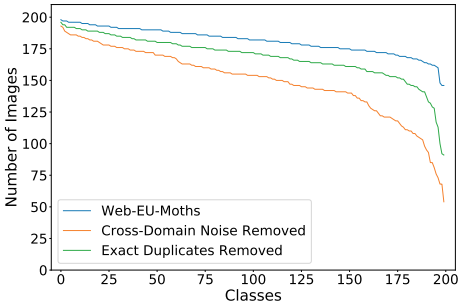
Fig. 1: Example images from the EU-Moths dataset from different classes.

an entire image. Example images from the original dataset are found in Fig. 1. From this dataset, we generated four data splits. Two consist of roughly three images per class in the training and eight in the testing dataset. The other two data splits encompass eight images in the training and three in the testing dataset. As further described in the following, we used these splits to verify the results in two scenarios: (1) an extremely small dataset, with three training images per class only (*three-seed* scenario), and (2) a moderate dataset with eight images per class (*eight-seed* scenario). The four training subsets in these data splits are also referred to as *seed* datasets in the following.

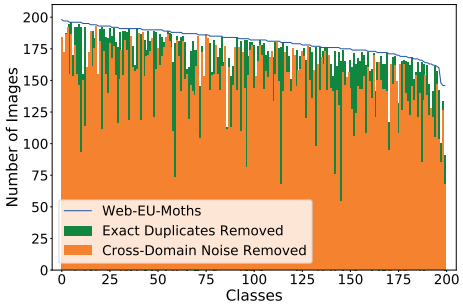
The **Web-EU-Moths Dataset** was compiled by using species names of the 200 moths to query Google Image Search. The links to the images can be found on the dataset website[9]. We downloaded up to 200 images for each class as described in Section 3.1. We call this dataset the *augment dataset* and it consists of 36,274 images in total. The blue line in Fig. 2a shows the sorted distribution of images over the classes in the Web-EU-Moths dataset. This dataset is not balanced because the number of available images downloaded from the Internet varies for different categories. The images in this Web-EU-Moths dataset contain the two discussed types of label noise. Furthermore, the moth images in the Web-EU-Moths dataset cover a larger domain than the narrow domain defined by the EU-Moths dataset of close-up crops of moths photographed on white backgrounds. The images of moths in the Web-EU-Moths dataset were photographed at different angles, on a wider variety of backgrounds, and at different scales. This is the so-called domain shift, and its impact is discussed in the following section.

4.2 The Effect of Webly Annotated Images

Different Learning Approaches Using the seed EU-Moths dataset for training is an example of *supervised* learning, where the training process is supervised by reliable labels. Using web data such as the augment Web-EU-Moths dataset by itself for training is often



(a) Sorted distributions of the Web-EU-Moths dataset, the cleaned version called *Web-EU-Moths dataset where cross-domain noise was identified manually, and the Web-EU-Moths dataset without exact duplicates (cross-class filtering with p=0.0).



(b) The number of images per class in the *Web-EU-Moths dataset (Cross-Domain Noise Removed) and dataset where exact duplicates are removed (cross-class nose filtering with p=0.0). Some classes are more affected by label noise resulting in class imbalance.

Fig. 2: The data distribution over classes in the Web-EU-Moths dataset and cleaned versions of it, where label noise has been removed

referred to as *webly-supervised* learning, where label noise is an issue. When using a reliably labeled dataset together with web data for training, this is arguably a semi-supervised approach to learning, where some labels are correct and others are not. Therefore, when we augment the EU-Moths seed datasets using data from the Web-EU-Moths dataset for training, we call this a semi-supervised learning approach. In this section, we compare semi-supervised learning with supervised and webly-supervised learning.

All following experiments using these different learning approaches were conducted with an InceptionV3 [Sz16] CNN architecture pre-trained on ImageNet [Ru15]. We repeated each experimental setup four times, each classifier is trained for 30 epochs with an initial learning rate of 1×10^{-4} which was lowered by a factor of 10 after 15 and 25 epochs. We also compare the effect of differently sized seed datasets as described in Section 4.1 for the supervised (training with seed data only) and semi-supervised (training with seed and web data) experiments. The reliably labeled EU-Moths seed datasets used in these experiments either have three or eight images per class in the training dataset. The results in Table 1 are the averaged accuracies for the two splits for each of the two scenarios (Sect. 4.1).

Comparing accuracies for the three-seed with the eight-seed scenario of the supervised learning setting (EU-Moths), we see that the difference of 1,101 training images causes a significant increase from 65.69 % to 91.01 %. This shows that even relatively few images (compared to modern benchmark datasets) have an immense impact when the initial situation involves an extremely small dataset, such as in the three-seed scenario.

These accuracies are exceeded when using the Web-EU-Moths dataset for training in a webly-supervised learning approach (second row in Table 1). The relative improvement of the webly-supervised learning compared to supervised learning is less pronounced in the

	three-seed			eight-seed		
Training Dataset	Acc ± std	#img	%RRE	Acc ± std	#img	%RRE
EU-Moths	65.69% ± 0.81	552		91.01% ± 0.40	1,653	
Web-EU-Moths	94.82% ± 0.27	36,274	84.90%	94.80% ± 0.58	36,274	42.16%
EU-Moths+Web-EU-Moths	96.72% ± 0.27	36,826	36.68%	97.94% ± 0.18	37,927	60.38%
EU-Moths+*Web-EU-Moths	96.88% ± 0.32	30,491	4.88%	97.98% ± 0.14	31,592	1.94%

Tab. 1: This table shows the average accuracies and standard deviations, as well as the number of training images and relative reduction of the error (RRE), compared to the learning approach in the row above. Training with the EU-Moths data is a supervised learning strategy, training on the Web-EU-Moths dataset is considered a webly-supervised approach, and training on the union of both is a semi-supervised approach. The *Web-EU-Moths dataset does not contain any cross-domain noise.

eight-seed scenario than in the three-seed scenario. However, in both cases, training on the large web dataset allows for better performance even though this web data has noisy labels.

In the semi-supervised scenario, the noisy Web-EU-Moths dataset is merged with three and eight images per class from the EU-Moths seed datasets, respectively. The increase of data amounts to 1.52 % in the three-seed scenario and 4.56 % in the eight-seed scenario compared to the Web-EU-Moths dataset. This rather small increase in training data yields an impressive improvement of the classification accuracy compared to the webly-supervised learning of roughly 2 % in the three-seed and 3 % in the eight-seed scenario.

Domain Shift in Web Data As expected, our results show that the amount of data used for training impacts the classification accuracy, but the *type* of data used matters as well, which is especially evident when comparing the results for the different learning strategies in the eight-seed scenario. We find that reasonable accuracies (>90 %) can already be achieved in a supervised learning approach when using a dataset of moderate size (eight-seed scenario). With a substantially larger dataset acquired from the Internet for training, the increase in accuracy is comparably low considering that over twenty times more training data is used. This is surprising, especially when considering the roughly 5 % increase in data used for training in the semi-supervised approach yields an equally large gain in performance. All this suggests that not only the total amount of data used for training but also the type of data plays an important role for the achievable classification accuracy.

These observations are explained by the fact that the training data used in the supervised strategy originates from the same source as the test data (EU-Moths dataset), while the web images have a wide variety of sources. As mentioned earlier, the Web-EU-Moths dataset has a shifted domain compared to the validation subsets of the EU-Moths dataset. The fact that the semi-supervised strategy is superior to the webly-supervised training using relatively little additional data suggests that few images from the same domain as the validation dataset can already counteract this domain shift in the Web-EU-Moths dataset.

	Three Seed			Eight Seed		
	Acc ± std	images	%RRE	Acc ± std	images	%RRE
Web-EU-Moths Subset (WEMS)	94.53% ± 0.25	35,178		95.17% ± 0.73	35,178	
WEMS + EU-Moths	96.66% ± 0.24	35,730	38.94%	98.46% ± 0.16	36,831	68.12%
WEMS + Mock Seed Dataset	94.67% ± 0.19	35,730	2.56%	95.53% ± 0.34	36,831	7.45%
EU-Moths	65.81% ± 0.48	552		90.52% ± 0.51	1,653	
Mock Seed Dataset	20.87% ± 0.88	552		53.78% ± 1.62	1,653	

Tab. 2: Comparing the webly-supervised approach using a subset of the Web-EU-Moths dataset with the two semi-supervised approaches were correctly labeled web images in the mock seed dataset (different domain as the validation data or images from the EU-Moths dataset (the same domain as the validation data) are used as seed datasets for training. The relative reduction of error rate compares the error rates of the semi-supervised approach with the webly-supervised approach in the first row.

To verify this and to make sure the boost in performance does not simply result from the extra data, further experiments were conducted. We compare the effect of additional data that originates from the validation dataset domain with additional data from the web. In these experiments, the exact number of images used for training is controlled. For each of the four EU-Moths seed datasets, we create a *mock* seed dataset with images from the Web-EU-Moths dataset and the same number of images per class to replace the EU-Moths seed datasets in the semi-supervised learning approach. The mock seed datasets were drawn from the pool of images identified manually as belonging to the domain (Sect. 4.3). The subset of images of the Web-EU-Moths dataset not used for the mock datasets is used to train a model, which acts as a baseline for this experiment. Then, the accuracies achieved when using the union of a mock dataset with the Web-EU-Moths subset for training are compared with using the original EU-Moths dataset with the same subset and evaluated on the same test data.

The results in Table 2 show that adding the mock seed data to the Web-EU-Moths Subset yields a smaller gain in performance than adding the EU-Moths seed data for training even though the merged datasets consist of exactly the same number of training images. Especially, the substantial difference in the relative reduction of error rate (RRE) compared to the error rate when using the Web-EU-Moths dataset in the webly-supervised approach underlines the superiority of the semi-supervised approach using data from the evaluation domain. The only explanation for this drastically reduced error compared to the mock datasets is that the few images in the EU-Moths dataset are enough to counteract the domain shift in the web images. This domain shift is the reason why using the large Web-EU-Moths dataset in the webly-supervised approach does not yield higher accuracies. However, our results show that this domain shift can be counteracted with very few images from the source domain resulting in higher predictive accuracies.

In summary, web data can be used to boost accuracies, especially when the existing data is extremely limited, and using a semi-supervised approach with few images from the domain of the validation data is advisable.

4.3 The Effect of Cross-domain Noise

To further understand the impact of web data in a training set on the classifier performance, we need to understand the impact of label noise. We want to compare training using noisy web data with training on clean web data. Since only experts could reliably identify cross-class noise, we limit our focus to cross-domain noise. We build a simple annotation tool to obtain *out-of-domain* vs. *within-domain* binary labels for the images in the Web-EU-Moths dataset. The user clicked only on the out-of-domain samples when looking at a small set of sixteen images in a single panel. Thus, images of maps, habitats, or caterpillars, as well as sketches or close-ups of moths' *heads*, can quickly be labeled as cross-domain noise. Using the binary labels to remove cross-domain noise resulted in the clean **Web-EU-Moths* dataset. The distribution of this cleaned dataset is more imbalanced than the original Web-EU-Moths dataset, which is visualized in Fig. 2a. With the manual labeling process, we identified 6,335 images (17.5 %) in the augment Web-EU-Moths dataset as outside of the moth domain. This level is relatively low, which might be explained, by the fact, that we only downloaded up to two hundred images per class. Also, the species in our EU-Moths dataset are pretty common and well studied, resulting in many images being available across the Internet.

We also use this cleaned version of the Web-EU-Moths dataset in a semi-supervised approach as described in Section 4.2. The results for using this cleaned *Web-EU-Moths dataset together with the EU-Moths training datasets for the two scenarios are found in Table 1. The manually cleaned web images do not yield significant performance gains compared to the simple merge of seed and noisy data. This observation holds independently of the number of images in the seed dataset.

Several studies [Ro17, FP17, Zh16] have shown how deep neural networks are surprisingly robust against moderate levels of label noise in coarse-grained classification problems. Our results indicate that models trained for fine-grained classification problems are also robust against label noise. An explanation was offered by [Zh16] for the robustness of deep neural networks. They claim that highly parameterized models such as deep neural networks (DNNs) have the capacity to learn and generalize from the non-noisy images and use brute force to memorize the noisy samples. This means that the quasi memorized noisy samples in the training dataset do not affect the classification accuracy of the test data, especially when the test set is free of out-of-domain images. Thus, the memorized noise does not influence the prediction of noise-free samples during the application.

The increased class imbalance in the clean *Web-EU-Moths dataset might also explain why the accuracies do not differ more when comparing training with and without label noise. Furthermore, label noise might have a positive regularization effect, which would explain why the predictive accuracy is equivalently high when training with noisy data.

	THREE-SEED			EIGHT-SEED		
	PRECISION	RECALL	F1-SCORE	PRECISION	RECALL	F1-SCORE
CROSS-DOMAIN FILTERS						
K=50	0.87446	0.96894	0.919	87.990	96.633	0.921
K=75	0.87751	0.95731	0.916	88.088	95.932	0.918
K=100	0.86520	0.97431	0.917	88.673	96.167	0.923
K=125	0.87696	0.97901	0.925	89.118	96.682	0.927
CROSS-CLASS FILTERS						
P=0.0	0.82919	0.93397	0.878	82.919	93.397	0.878
P=0.01	0.82965	0.93377	0.879	82.965	93.377	0.879
P=0.05	0.83184	0.93336	0.880	83.184	93.336	0.88

Tab. 3: This table shows the precision, recall, and F1-score of different filter setups. Since the cross-class noise filters are entirely independent of the seed dataset, these values are the same for both seed data scenarios.

4.4 The Effect of Filters

In this section, our proposed filtering methods are analyzed. Intuitively, the semi-supervised approach using the *Web-EU-Moths dataset should act as an upper bound for the following setups, where automatically filtered subsets of the augment data are merged with the seed datasets for training. We will discuss the effectiveness of our filtering methods, even though the expected effect on accuracies in our particular application of moth species classification is negligible. Our experiments show, using filtered datasets results in comparable accuracy with far less training data, which reduces training time.

We test the clustering-based cross-domain noise filtering technique with four seed datasets of the EU-Moths dataset, which guide the selection of positive clusters. In addition, the parameter for the number of clusters k in the cross-domain noise filtering method has been set to one of the following four values: 50, 75, 100, or 125. An overview of precision, recall, and F1-score for the different filter settings averaged over two dataset splits for the three-seed and the eight-seed scenario are given in Table 3. The precision refers to the percentage of images retained by the filter that are non-noisy ones. Recall describes the percentage of all non-noisy images, i.e., images depicting moths, retained by a given filter. Since many studies [Ro17, FP17, Zh16] point towards the minimal adverse effect of label noise, it is arguably more critical to retain a large portion of the non-noisy data than to ensure the retained data contains little noise. Therefore, high recall is more important when comparing different hyperparameter settings for the filters.

Overall, we found a high overlap of retained images between different filters using different values of k and differing seed datasets. This indicates the robustness of the proposed filtering method, given different hyperparameter settings and seed datasets. The average intersection-over-union ratio of retained images between all pairs of filters is 94.52 % with

	three-seed			eight-seed		
	Acc ± std	#img	%ret	Acc ± std	#img	%ret
EU-Moths+ Web-EU-Moths	96.72% ± 0.27	36826	100.00%	97.94% ± 0.18	37927	100.00%
EU-Moths+filtered Web-Eu-Moths						
Cross-Domain Filters						
k=50	96.81% ±0.18	33,729	91.46%	98.05% ±0.36	34,565	90.74%
k=75	96.75% ±0.14	33,214	90.04%	97.62% ±0.36	34,258	89.88%
k=100	96.84% ±0.16	34,267	92.94%	97.67% ±0.56	34,123	89.52%
k=125	96.90% ±0.31	33,975	92.14%	97.96% ±0.11	34,136	89.55%
Cross-Class Filters						
p=0.0	96.74% ±0.32	34,274	92.96%	97.92% ±0.46	35,375	92.96%
p=0.01	96.85% ±0.22	34,248	92.89%	97.76% ±0.56	35,349	92.89%
p=0.05	96.92% ±0.29	34,145	92.61%	97.84% ±0.37	35,246	92.61%
Combined Filters						
k=50 p=0.0	96.98% ±0.28	31,438	85.15%	97.86% ±0.48	32,282	84.44%
k=75 p=0.0	96.96% ±0.18	30,958	83.82%	97.85% ±0.38	31,974	83.59%
k=100 p=0.0	96.93% ±0.22	31,980	86.64%	98.01% ±0.32	31,891	83.36%
k=125 p=0.0	96.92% ±0.10	31,697	85.86%	98.05% ±0.37	31,896	83.38%
k=50 p=0.01	97.00% ±0.26	31,432	85.13%	97.98% ±0.34	32,276	84.43%
k=75 p=0.01	96.99% ±0.30	30,952	83.81%	98.05% ±0.22	31,968	83.58%
k=100 p=0.01	96.90% ±0.36	31,976	86.63%	97.72% ±0.50	31,885	83.34%
k=125 p=0.01	96.87% ±0.22	31,692	85.84%	97.85% ±0.44	31,890	83.36%
k=50 p=0.05	97.03% ±0.22	31,416	85.09%	98.10% ±0.26	32,260	84.38%
k=75 p=0.05	97.07% ±0.31	30,936	83.76%	97.94% ±0.20	31,952	83.53%
k=100 p=0.05	97.00% ±0.35	31,960	86.59%	97.96% ±0.22	31,869	83.30%
k=125 p=0.05	96.90% ±0.30	31,676	85.80%	98.10% ±0.36	31,874	83.32%

Tab. 4: Impact of different filter options on the classification accuracy when training a moth species classifier with a semi-supervised approach using both the EU-Moths dataset and the Web-EU-Moths dataset. The retention rate (ret%) refers to the percentage of web images retained by th filter. The first row shows the results when using the unfiltered web data for comparison

a standard deviation of ±1.52 %. This means that on average, two filters *agreed* on more than 94 % images that should be retained. On the other hand, the overlap was lower for the noisy images with much more variation. Here, the intersection-over-union ratio of discarded images of two noise filters was on average 56.75 % ± 9.88 % This lower level of overlap and higher variation is explained by the fact that the total number of filtered images was low in all setups, making the statistics more sensitive to small differences. Overall, the similar F1-scores also show that the cross-domain filtering techniques are robust to different hyperparameters.

Since evaluating cross-class noise filtering methods would require expert knowledge, it is out of the scope of this work. However, we can assess how well the method for cross-class noise filtering is suited for identifying cross-domain noise. The idea is that out-of-domain

images might be downloaded multiple times in different classes, which means the cross-class noise filtering method based on duplicate detection might be effective for filtering these instances. The lower half in Table 3 reveals that even though the cross-class noise filtering method was not designed to do so, it is surprisingly effective at handling cross-domain noise. This confirms the hypothesis that many noisy images are both out-of-distribution and downloaded multiple times for different classes. As our previous work [Bö21] shows, it is advantageous to set the parameter portion (p), which specifies how many images are removed (Section 3.3), to a low value for this domain. Hence, we evaluate the cross-class noise filtering with the values 0.0, 0.01, and 0.05 for the parameter p. With these low values for p, only a few images are discarded by cross-class noise filtering. When filtering out only exact duplicates ($p = 0.0$), we filter out the minimum cross-class noise. Similar to the cleaned Web-EU-Moths dataset without cross-domain noise, the data distribution of the web dataset where the minimum cross-class noise removed is visualized in Figure 2.

We used the filtered subsets of the Web-EU-Moths dataset to train a model in a semi-supervised manner, where seed datasets of different sizes (three-seed and eight-seed) are merged with the retained images (Table 4). We average the results from two splits of the same size for each of the different scenarios. In the experiments in which cross-domain noise and cross-class noise filters are combined, only those images are retained that both filters consistently identify as non-noisy, i.e., images are discarded if at least one filter identifies it as noise. As expected, there is no clear trend towards a performance boost when using only filtered data for training. However, the classification accuracies remain stable for all filter setups, even when less data is used for training the classifier. This shows that the noise filtering techniques succeed in filtering out irrelevant images and are robust with respect to different hyperparameter settings. Therefore, in situations where the level of noise is higher, our methods are likely to be beneficial in terms of predictive accuracy.

5 Conclusion

Our work shows the benefits of additional training datasets acquired through image search engines from the Internet for learning a classification model. The conducted experiments demonstrate that a semi-supervised learning approach, where web images together with images of the evaluation domain are used for training, can counteract the domain shift in web data. This approach yields higher accuracies than a webly-supervised approach, i.e., training using web data only, and a supervised approach, i.e., training on small seed datasets only. We further found evidence that deep neural networks trained for fine-grained classification tasks are robust to label noise, and removing noise does not significantly improve classification performances in our application. Finally, we studied the effect of noise filtering techniques using ground truth annotations of cross-domain label noise and found that the simple to implement methods we proposed are effective in identifying noisy samples.

Acknowledgements

This work has been funded by the German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF, Deutschland) via the project ”Development of an Automated Multisensor Station for Monitoring of Biodiversity (AMMOD) - Subproject 5: Automated Visual Monitoring and Analysis"(FKZ: 01LC1903E).

Bibliography

[BD20] Barz, Björn; Denzler, Joachim: Do we train on test data? Purging CIFAR of near-duplicates. Journal of Imaging, 6(6):41, 2020.

[Bö21] Böhlke, Julia; Korsch, Dimitri; Bodesheim, Paul; Denzler, Joachim: Lightweight Filtering of Noisy Web Data: Augmenting Fine-grained Datasets with Selected Internet Images. In: VISAPP. 2021.

[CSG13] Chen, Xinlei; Shrivastava, Abhinav; Gupta, Abhinav: Neil: Extracting visual knowledge from web data. In: IEEE International Conference on Computer Vision. pp. 1409–1416, 2013.

[Cu16] Cui, Yin; Zhou, Feng; Lin, Yuanqing; Belongie, Serge J.: Fine-grained Categorization and Dataset Bootstrapping using Deep MetricLearning with Humans in the Loop. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1153–1162, 2016.

[Cu18] Cui, Yin; Song, Yang; Sun, Chen; Howard, Andrew; Belongie, Serge: Large scale fine-grained categorization and domain-specific transfer learning. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4109–4118, 2018.

[FP17] Flatow, David; Penner, Daniel: , On the robustness of convnets to training on noisy labels, 2017.

[Ga16] Garcia, Luís PF; Lorena, Ana C; Matwin, Stan; de Carvalho, André CPLF: Ensembles of label noise filters: a ranking approach. Data Mining and Knowledge Discovery, 30(5):1192–1216, 2016.

[GLY19] Ge, Weifeng; Lin, Xiangru; Yu, Yizhou: Weakly supervised complementary parts models for fine-grained image classification from the bottom up. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3034–3043, 2019.

[Ha18] Han, Bo; Yao, Quanming; Yu, Xingrui; Niu, Gang; Xu, Miao; Hu, Weihua; Tsang, Ivor; Sugiyama, Masashi: Co-teaching: Robust training of deep neural networks with extremely noisy labels. In: Advances in Neural Information Processing Systems (NeurIPS). 2018.

[Ha19] Habel, Jan Christian; Trusch, Robert; Schmitt, Thomas; Ochse, Michael; Ulrich, Werner: Long-term large-scale decline in relative abundances of butterfly and burnet moth species across south-western Germany. Scientific reports, 9(1):1–9, 2019.

[Ji18] Jiang, Lu; Zhou, Zhengyuan; Leung, Thomas; Li, Li-Jia; Fei-Fei, Li: MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. In: Proceedings of the International Conference on Machine Learning. 2018.

[Kä16] Käding, Christoph; Rodner, Erik; Freytag, Alexander; Denzler, Joachim: Active and continuous exploration with deep neural networks and expected model output changes. arXiv preprint arXiv:1612.06129, 2016.

[KBD19] Korsch, Dimitri; Bodesheim, Paul; Denzler, Joachim: Classification-Specific Parts for Improving Fine-Grained Visual Categorization. In: German Conference on Pattern Recognition. pp. 62–75, 2019.

[Ke04] Ke, Yan; Sukthankar, Rahul; Huston, Larry; Ke, Yan; Sukthankar, Rahul: Efficient near-duplicate detection and sub-image retrieval. In: ACM Multimedia. volume 4, p. 5, 2004.

[Kr16] Krause, Jonathan; Sapp, Benjamin; Howard, Andrew; Zhou, Howard; Toshev, Alexander; Duerig, Tom; Philbin, James; Fei-Fei, Li: The unreasonable effectiveness of noisy data for fine-grained recognition. In: European Conference on Computer Vision. pp. 301–320, 2016.

[Li21] Liu, Huafeng; Zhang, Chuanyi; Yao, Yazhou; Wei, Xiushen; Shen, Fumin; Zhang, Jian; Tang, Zhenmin: Exploiting Web Images for Fine-Grained Visual Recognition by Eliminating Noisy Samples and Utilizing Hard Ones. arXiv preprint arXiv:2101.09412, 2021.

[LRM15] Lin, Tsung-Yu; RoyChowdhury, Aruni; Maji, Subhransu: Bilinear cnn models for fine-grained visual recognition. In: IEEE International Conference on Computer Vision. pp. 1449–1457, 2015.

[MSS17] Malach, Eran; Shalev-Shwartz, Shai: Decoupling"when to update"from"how to update". In: Advances in Neural Information Processing Systems (NeurIPS). 2017.

[Ni15] Nicholson, Bryce; Zhang, Jing; Sheng, Victor S; Wang, Zhiheng: Label noise correction methods. In: IEEE International Conference on Data Science and Advanced Analytics. pp. 1–9, 2015.

[Pe20] Peng, Xiaojiang; Wang, Kai; Zeng, Zhaoyang; Li, Qing; Yang, Jianfei; Qiao, Yu: Suppressing Mislabeled Data via Grouping and Self-Attention. In: European Conference on Computer Vision. Springer, pp. 786–802, 2020.

[Re18] Ren, Mengye; Zeng, Wenyuan; Yang, Bin; Urtasun, Raquel: Learning to Reweight Examples for Robust Deep Learning. In: Proceedings of the International Conference on Machine Learning. 2018.

[Ro17] Rolnick, David; Veit, Andreas; Belongie, Serge; Shavit, Nir: Deep learning is robust to massive label noise. arXiv preprint arXiv:1705.10694, 2017.

[Ru15] Russakovsky, Olga; Deng, Jia; Su, Hao; Krause, Jonathan; Satheesh, Sanjeev; Ma, Sean; Huang, Zhiheng; Karpathy, Andrej; Khosla, Aditya; Bernstein, Michael; Berg, Alexander C.; Fei-Fei, Li: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115(3):211–252, 2015.

[Si17] Simon, Marcel; Gao, Yang; Darrell, Trevor; Denzler, Joachim; Rodner, Erik: Generalized orderless pooling performs implicit salient matching. In: IEEE International Conference on Computer Vision. pp. 4970–4979, 2017.

[Sz16] Szegedy, Christian; Vanhoucke, Vincent; Ioffe, Sergey; Shlens, Jon; Wojna, Zbigniew: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826, 2016.

[Wa04] Wang, Zhou; Bovik, Alan C; Sheikh, Hamid R; Simoncelli, Eero P: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4):600–612, 2004.

[Wa06] Wang, Bin; Li, Zhiwei; Li, Mingjing; Ma, Wei-Ying: Large-scale duplicate detection for web image search. In: IEEE International Conference on Multimedia and Expo. pp. 353–356, 2006.

[Wa11] Wah, C.; Branson, S.; Welinder, P.; Perona, P.; Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[Wa14] Wang, Jiang; Song, Yang; Leung, Thomas; Rosenberg, Chuck; Wang, Jingbin; Philbin, James; Chen, Bo; Wu, Ying: Learning fine-grained image similarity with deep ranking. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1386–1393, 2014.

[Wa21] Wagner, David L; Fox, Richard; Salcido, Danielle M; Dyer, Lee A: A window to the world of global insect declines: Moth biodiversity trends are complex and heterogeneous. Proceedings of the National Academy of Sciences, 118(2), 2021.

[Xu15] Xu, Zhe; Huang, Shaoli; Zhang, Ya; Tao, Dacheng: Augmenting Strong Supervision Using Web Data for Fine-Grained Categorization. In: IEEE International Conference on Computer Vision. December 2015.

[Xu18] Xu, Zhe; Huang, Shaoli; Zhang, Ya; Tao, Dacheng: Webly-Supervised Fine-Grained Visual Categorization via Deep Domain Adaptation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(5):1100–1113, 2018.

[Ya20] Yao, Yazhou; Hua, Xiansheng; Gao, Guanyu; Sun, Zeren; Li, Zhibin; Zhang, Jian: Bridging the web data and fine-grained visual recognition via alleviating label noise and domain mismatch. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 1735–1744, 2020.

[Zh16] Zhang, Chiyuan; Bengio, Samy; Hardt, Moritz; Recht, Benjamin; Vinyals, Oriol: Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530, 2016.

[Zh17] Zhuang, Bohan; Liu, Lingqiao; Li, Yao; Shen, Chunhua; Reid, Ian: Attend in groups: a weakly-supervised deep learning framework for learning from web data. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1878–1887, 2017.

[Zh20a] Zhang, Chuanyi; Yao, Yazhou; Shu, Xiangbo; Li, Zechao; Tang, Zhenmin; Wu, Qi: Data-driven meta-set based fine-grained visual recognition. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 2372–2381, 2020.

[Zh20b] Zhang, Chuanyi; Yao, Yazhou; Zhang, Jiachao; Chen, Jiaxin; Huang, Pu; Zhang, Jian; Tang, Zhenmin: Web-Supervised Network for Fine-Grained Visual Classification. In: IEEE International Conference on Multimedia and Expo. pp. 1–6, 2020.

[ZT19] Zhang, Weining; Tan, Xiaoyang: Combining Outlier Detection and Reconstruction Error Minimization for Label Noise Reduction. In: IEEE International Conference on Big Data and Smart Computing. pp. 1–4, 2019.