Universiteit Leiden

Master Computer Science

Assessing the Effect of Background Content on Pollen Classification Performance

Name:	Ymkje Elisabeth Hordijk
Student ID:	2361035
Date:	[23/09/2025]
Specialisation:	Bioinformatics
1st supervisor:	Lu Cao
2nd supervisor:	Fons Verbeek

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Contents

Abstract

Automated, accurate, and fast pollen classification is essential for monitoring and managing allergic rhinitis. However, for imaging methods, preprocessing steps need to be performed in order to optimise the prediction performance. These steps can be computationally expensive and labor-intensive, especially if the pollen objects need to be precisely segmented as opposed to using bounding boxes. In order to optimise the preprocessing stage, acquiring insights on the effects of using bounding boxes as opposed to segmentation is essential. In this thesis, we examined the effect of background information from bounding boxes on prediction performance, in comparison with segmentation methods. Our analysis considers various backgrounds, models, colour spaces, and feature selection methods to explain their impact on classification performance. As clean ground truth data we used pollen from *B. pendula*, *C. lawsoniana*, and *C. nootkatensis* collected directly from trees. We found that classical feature-based machine learning model SVM is sensitive to background information, where the performance is negatively correlated with the size of the bounding box. Deep learning model ResNet50 remains unaffected by different types of background and generally outperforms classical machine learning. Deep learning on RGB images consistently outperforms greyscale classification, whereas for feature-based learning methods feature redundancy can in some cases negatively affect the performance. Overall, our results show that bounding boxes, even with noisy backgrounds, are sufficient for achieving high prediction accuracy, making precise segmentation largely unnecessary.

1 Introduction

This thesis explores the influence of background information on image classification of pollen species. In this section, we provide the context and motivation for this research, present the research questions, and outline the structure of this thesis.

1.1 Background & Context

Pollen released into the air by trees and grasses is a major trigger for allergic reactions. An estimated 40% of the European population suffers from allergic rhinitis caused by airborne pollen (commonly known as *hay fever*) [DVS+17]. Symptoms include itchy eyes, runny noses, and headaches, and in severe cases affect quality of life. For immunocompromised individuals, or those with chronic respiratory conditions like asthma, increased airborne pollen levels can present a serious health risk [Paw14]. According to Akesaka et al. (2023) the allergic reactions caused by hay fever can even be associated with accidents and injuries [AS23]. Vuurman et al. (2014) estimate that allergic rhinitis has a similar effect on driving performance as having a blood alcohol content of 0.05%, which many countries have set as a limit of what is still safe [VVLK14].

Climate change is causing airborne pollen quantities to progressively increase because the warming temperatures have caused trees and grasses to bloom earlier and longer, extending pollen seasons [Sch16, PTD25, dCNMO+20]. Longer seasons also increase the chances of overlapping flowering seasons for species, and the resulting cooccurring airborne pollen could potentially make the allergic reactions more severe. Moreover, due to the temperature changes, non-native allergenic plant species are now able to thrive in regions that were previously unsuitable for these species. This results in new sources of allergens to which a large part of the population is not sensitised [ABL+18, BBE+23, PTL+22, KHB+24].

To manage their symptoms, people with pollen allergies typically take medication when complaints occur or adjust their outside routine. However, the intake of medication can be optimised and also unnecessary exposure to pollen can be avoided if reliable and timely pollen monitoring is available. Accurate quantification of airborne pollen also aids biodiversity monitoring and climate impact assessments [TGB+17]. Many pollen measurements are still performed manually by trained analysts using a brightfield microscope. This method is effective in term of accuracy but unfortunately also time-consuming and the quality of the annotation is dependent on the skill of the analyst. Automated image-based pollen classification offers a faster and more scalable alternative. However, achieving reliable performance depends heavily on how the image data is preprocessed—particularly the treatment of background information. Manual annotation also introduces the risk of classification bias, especially for morphologically similar species. [DMB+24] reported that many hand-labeled datasets were annotated by individuals lacking specific expertise in palynology, highlighting the need for consistent, automated classification approaches.

There are mainly two approaches to automate image classification: traditional feature based machine learning and deep learning. Traditional classifiers (e.g., Support Vector Machines, Random Forests) rely on features that are extracted from the images beforehand. Deep learning models, particularly convolutional neural networks (CNNs), use the raw image as input. This avoids the need for extracting features in advance, but comes at the cost of greater computational demands and a requirement for larger datasets. Nevertheless, deep learning has shown excellent performance in pollen classification [KKKPWS21, SZA+20]. Li et al. (2023)

compared several machine learning and deep learning models [LPC+23]. For the classical machine learning methods, Support Vector Machines (SVMs) were found to perform best with an accuracy of 94.5%, while for the Convolutional Neural Networks, ResNet50 achieved the highest accuracy of 99.4%. These two methods are also used in this thesis to compare differences in background influence between classical machine learning and deep-learning methods.

If the background is not controlled, it can significantly influence the feature extraction quality. For example, when computing colour-, texture-, or shape features, background pixels contribute to the calculated descriptors unless they are explicitly excluded. As a result, models may inadvertently learn to rely on background cues rather than object features. For instance, Kamal et al. (2021) showed that models trained on images with homogeneous backgrounds outperformed those trained on cluttered backgrounds by up to 12% in binary classification tasks, and also converged faster [KYLW21]. When background is not controlled, it may introduce noise into feature calculations and model predictions. In addition to random background noise, the background information could also include a bias that is tied to the species. In pollen classification, usually there is no specific background naturally tied to a species. However, if single-species samples are captured under different conditions (e.g. different microscopic glass, mounting solution, or scanning with dissimilar settings like illumination) the background could hold species specific information. Xiao et al. (2020) demonstrated that CNNs trained on ImageNet could be misled when objects were placed against unexpected backgrounds [XEIM20]. Similarly, Moayeri et al. (2022) emphasised that background manipulation as a data augmentation method improves generalisation, especially when training data is limited [MPBF22]. One data augmentation method that has been proposed is CutMix [YHO+19]. In this technique, part of an image is replaced with a patch from another image to reduce a model's reliance on background information and thereby improve robustness. The CutMix method served as an inspiration for the techniques used in this thesis to study background influence.

Different types of classifiers are expected to respond to the background contamination differently. ResNet50 values local texture features more than global shapes and colours [GRM+18], therefore it should inherently be more robust to background influences. SVM, or other methods that use feature vectors, cannot separate the background contribution from the object of interest and is therefore expected to be more sensitive to irregular background information.

To prevent background interference with classification performance, objects can be segmented precisely along their edges so that no background pixels remain. This background removal can be performed manually or automatically. Simple automated methods include intensity-based thresholding, which may struggle with background elements with similar intensity values as the object of interest. More sophisticated segmentation methods use deep learning, but these still require ground-truth masks created manually in a labor-intensive process.

Moreover, accurately segmenting pollen along their edges is a particularly challenging task, as pollen grains often cluster together, making it difficult to distinguish individual objects. Additionally, due to the way 3D microscopy image stacks are projected into 2D (for example, using maximum- or standard deviation projection), pollen grains may be surrounded by faint halos. These halos complicate the thresholding process, causing pollen to cluster or allow the inclusion of background pixels. As a result, segmentation errors can lead to irregular feature extraction and ultimately reduce classification performance. A compromise is to use bounding boxes, which require less precise annotation while still focusing on relevant regions. However, they also allow background information to enter the classification. The background influence depends on box size and the level of background noise.

1.2 Motivation

There is a growing demand for automated and standardised pipelines in pollen classification to improve speed and precision, both for medical forecasting and ecological monitoring. However, to standardise such pipelines requires clear insights into the influence and necessity of each processing step, including background removal.

In this thesis, we used pollen sampled directly from trees as a clean ground truth. However, the final goal is to apply classifiers to aerosol-sampled data, where pollen grains are collected directly from the atmosphere and are mixed with dust, debris, and other contaminants. This underlines the importance of understanding the impact of background noise on classification performance.

Furthermore, understanding which parts of an image contribute the most to model decisions can help to improve the model's interpretability. We would be able to detect to what degree the model takes background into consideration and how different parts of the pollen grains contribute to the classification. Feature sensitivity analysis can reveal how background contamination affects the extracted features which can help to improve feature selection or preprocessing strategies.

1.3 Research Question

Thus, this research aims to quantify the effect of background pollution and gain insight into how features and models respond to such noise. This knowledge is essential for building efficient, interpretable, and generalizable pollen classification pipelines.

The general research question for this thesis is as follows:

"How does background information influence the prediction performance in pollen classification using three types of pollen comparing machine learning and deep learning?"

In order to be able to answer the general research question we have composed the following four subquestions:

1. *Does grayscale vs RGB input images make a difference in the performance?*
2. *How does the boxsize influence the prediction performance?*
3. *How do variations in background characteristics impact prediction performance and the contribution of specific features?*
4. *How does the influence of background information on prediction performance differ between classical machine learning models and deep learning models?*

These questions aim to identify how much preprocessing (segmentation quality, background control) matters in a pollen classification pipeline.

1.4 Thesis Structure

This thesis is structured as follows. We will first describe the dataset, preprocessing steps and experimental setup, including the segmentation process and classification algorithms used, in section 2. Section 3 presents the results of the experiments, including comparisons between segmentation approaches and background influences. In Section 4, we discuss the implications of the findings, reflect on limitations, and make suggestions for future work.

2 Methods and Materials

This section outlines the methods and materials employed to design the experiment, with a focus on enabling the reproducibility of the experiments. We will first give an overview of the used data, sample acquisition, and computational environment and software. Then we will discuss the preprocessing methods after which the classification methods are explained.

2.1 Sample Acquisition

We used microscopic pollen grain images of three species: *B. pendula* (2 slides), *C. nootkatensis* (1 slide), *C. lawsoniana* (2 slides). The trees that were used to create the samples can be found in the area of Leiden, The Netherlands. The samples were taken by holding and shaking the catkins above the microscope slide causing the pollen grains to fall onto the glass. Catkins from three different trees were used in order to ensure to capture variation within the species in the sample.

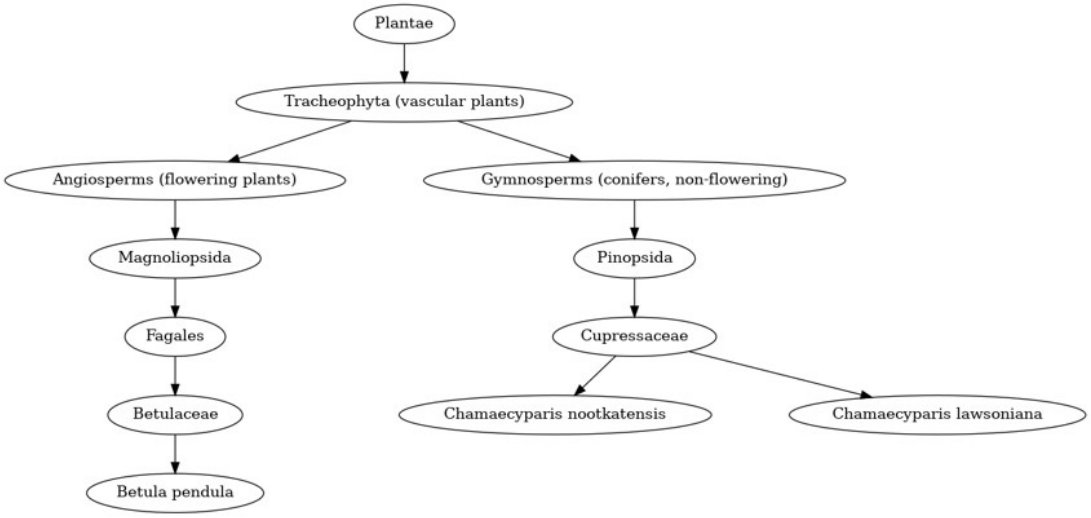
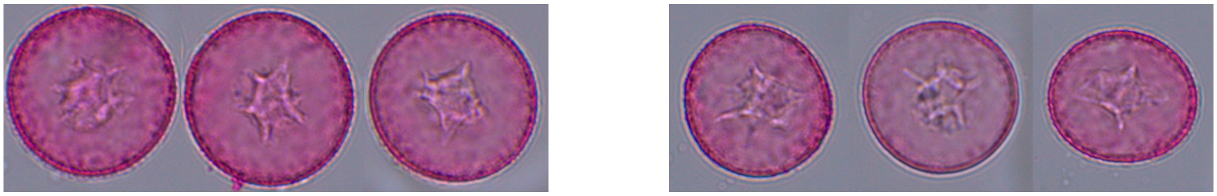

Plantae
Tracheophyta (vascular plants)
Angiosperms (flowering plants)
Gymnosperms (conifers, non-flowering)
Magnoliopsida
Pinopsida
Fagales
Cupressaceae
Betulaceae
Chamaecyparis nootkatensis
Chamaecyparis lawsoniana
Betula pendula

Figure 1: Family tree showing the relation between the pollen species *B. pendula*, *C. nootkatensis*, and *C. lawsoniana* illustrating taxonomic similarity, which influences classification difficulty.

The relation between the three species is shown in the family tree in Figure 1. Since *C. nootkatensis* and *C. lawsoniana* are in the same genus, we can expect them to look more similar compared to *B. pendula*.



(a) *C. Lawsoniana*

(b) *C. nootkatensis*

Figure 2: Three images of both *C. lawsoniana* and *C. nootkatensis* made by a brightfield microscope.

Figure 2 shows three samples of both *C. lawsoniana* and *C. nootkatensis* taken by a bright-field microscope. These images demonstrate that it is very difficult, if not impossible to distinguish these types by eye, even for trained experts. This would make them a challenging pair for the classification problem. In comparison with *B. pendula*, which is visually much easier to distinguish from the two *Cupressaceae* species, we would expect to see more misclassifications within the same genus than between different genera. Figure 3 shows three types of z-projections: minimal, maximum, and standard deviation. It is clear that the two *Cupressaceae* species are rather easy to distinguish by eye after they have been preprocessed. This makes the classification problem less complex. In addition to the pollen grain samples, a sample was taken from a sniffer, an apparatus that draws in air and the debris that it contains sticks to the vaseline that is on the slides. The sniffer was positioned on the roof of the hospital LUMC in Leiden. A slide from the sniffer was chosen that contained a lot of pollution but not pollen grains. This scan was used for background. No pollen were extracted from this scan.

2.2 Data Acquisition by Microscope

The data was produced by the confocal microscope, type: ZEISS Axioscan 7. The pollen grains could be found at different heights in the microscope slide, therefore getting all pollen grains in focus would require much manual attention. By scanning the pollen grains at 20 different heights (z-slices) we created 3D images of which the images would be in focus in a few of the planes. The stepsize between the z-slices is $1.80\ \mu\text{m}$, therefore the total z-range is $34.20\ \mu\text{m}$. The objective used was an Apochromat 40x ($0.086\ \mu\text{m}$/pixel) with an Axiocam 705 colour CMOS camera. The regions of interest were found manually. The images were delivered as 10% overlapping tiles in `tiff` format with a constant size of 2056 x 2464 pixels. The tiles were left unstitched for further processing.

2.3 Computational Environment

All data preprocessing, analysis, and model training were performed on a desktop workstation with the following specifications:

- **Processor**: Intel(R) Core(TM) i7-6950X CPU @ 3.00GHz
- **Memory**: 46 GiB RAM
- **Graphics**: 2 × NVIDIA GeForce RTX 2070, 8192 MiB VRAM each
- **Operating System**: Ubuntu 22.04.4 LTS
- **CUDA**: Version 12.4, NVIDIA driver 550.54.15

2.4 Software and Libraries

All scripts and analyses were written in Python (version 3.10.12), with custom scripts for all parts of the pipeline such as the preprocessing steps, the data augmentation, feature extraction, background preparation, and building machine learning and deep learning classifiers. For these scripts the following key packages were used;

- `torch` (2.7.1)
- `opencv-python` (4.12.0.88)

- `scikit-image` (0.25.2)
- `scikit-learn` (1.7.1)
- `diplib` (3.5.2)
- `pillow` (11.3.0)

All scripts were executed using the standard CPython interpreter. For reproducibility, random seeds were set fixed in the code.
The implementation and list of packages is available at GitHub.

2.5 Preprocessing

In order to prepare the input data for classification, several preprocessing steps were performed.

2.5.1 Projecting 3D Images to 2D

The input data was presented as 3D images produced by the confocal microscope each image consisting of 20 stacks. For our pipeline we have decided to work with 2D images so therefore the first step was to project the 3D images into 2D images. We initially used three different methods to make these projections [Col07] minimal, maximum, and standard deviation. The projection calculations calculate the minimum, maximum or standard deviation of all pixels in the z-dimension, for each pixel in the x, y plane. After the projections were made, we decided what projections to use for further steps.

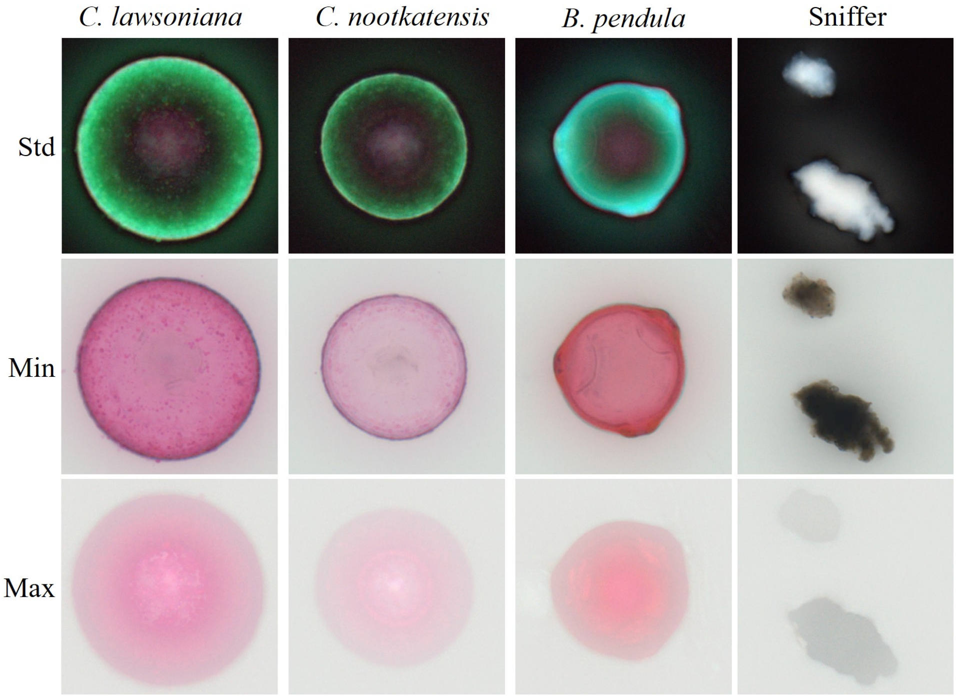
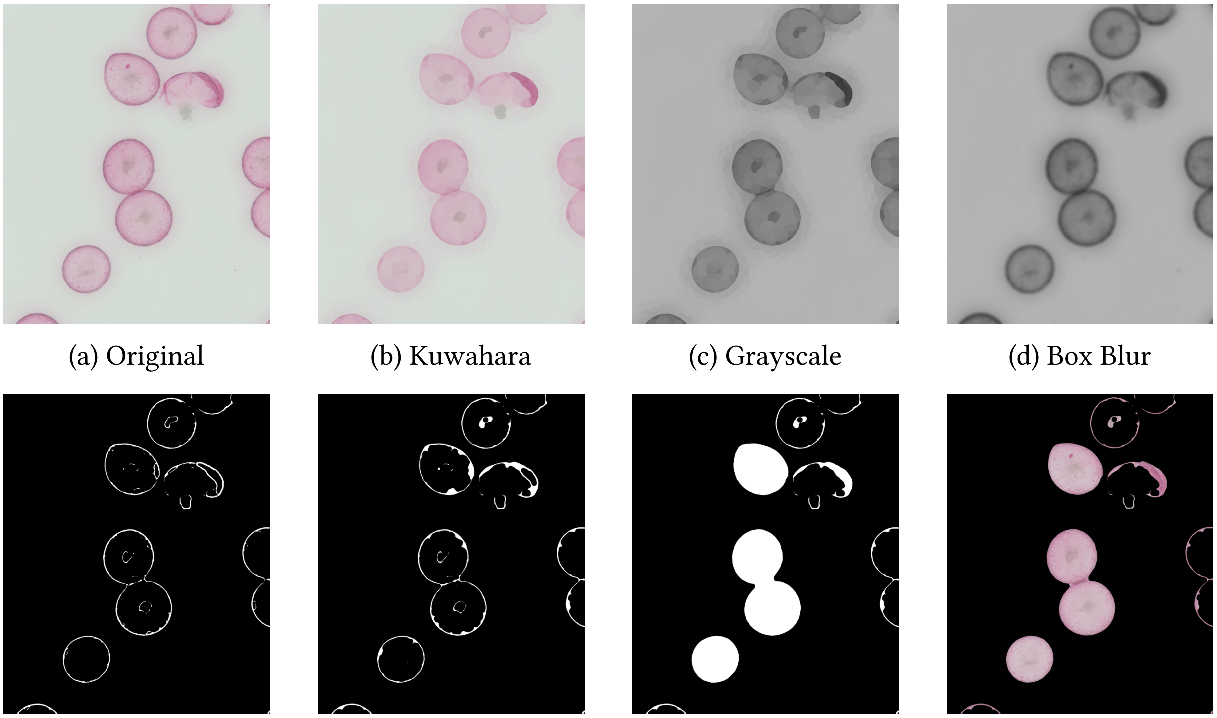


Figure 3: 2D projections from 20 z-stacks 3D pollen and sniffer images. The standard deviation projection has been normalised for better visibility.

Figure 3 shows the projections of all three used species and sniffer images. To reduce redundancy we only used minimal projections for further processing. This projection type was selected due to the well-defined edges.

2.5.2 Segmentation

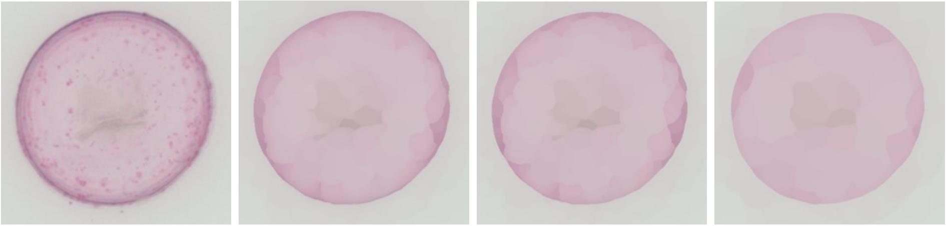
From the projected tiles we extracted the individual pollen grains so that we could either compute the feature vectors or use the segmented pollen directly as input to the deep learning model.In order to obtain the most accurate feature measurements we needed to create binary masks that would fit the single pollen as precisely as possible. In many other studies that require object segmentation, researchers manually annotate object edges to create a ground truth for the entire dataset [PBB+21]. We chose to use a more automated approach for which we applied multiple morphological operations. An example of the intermediate results of the segmentation process are shown in Figure 4.


(a) Original (b) Kuwahara (c) Grayscale (d) Box Blur

(e) Thresholding (f) Closed (g) Filled (h) Masked

Figure 4: Image processing pipeline showing each intermediate step.

We start with the minimal projection of a tile (Figure 4a). The first step is to smoothen the image and suppress small texture details while preserving the object boundaries (Figure 4b). The Kuwahara filter [KHEK76] was chosen for this step because of its edge-reserving property. We applied three Kuwahara filters sequentially with elliptic kernels of radii 30, 10, and 40 pixels. The kernels for the Kuwahara filters have been tuned experimentally to give visually clean and sharp edges around the pollen grains. An example of applying the Kuwahara filters is shown in Figure 5.



(a) Original (b) kernel 30x30 (c) kernel 10x10 (d) kernel 40x40

Figure 5: Effect of the three consecutive Kuwahara filters applied with elliptic kernels of different sizes. The image is zoomed in on a single pollen grain.

The smoothed image was then converted to a greyscale image (Figure 4c). Afterwards, further details were removed by applying a box blur filter (implemented via Pillow's `ImageFilter.BoxBlur`) which is a local averaging filter (Figure 4d). The box blur filter computes the local mean given a window size of 31 pixels (radius of 15 pixels).

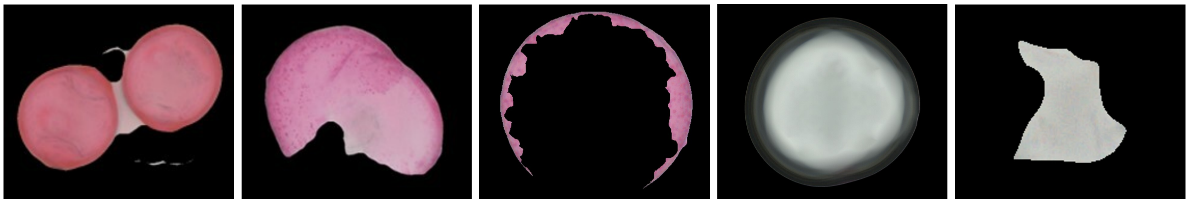
Then we apply local mean thresholding, as proposed by Bradley et al. (2007) [BR07]. Each pixel is compared to the local mean calculated by the box blur filter (Figure 4d), minus an offset of 10. By slightly lowering the threshold more pixels are included in the foreground. All pixels that have a higher intensity than `mean-10` are considered foreground. Because the image had previously been processed by the Kuwahara filters, the inside and outside (background) of the pollen grains are already very smooth and do not change much by applying the local averaging filter. However, the edges are being changed because they take intensity information from both in- and outside the grain into consideration. Therefore, by pixelwise subtracting the box blurred image from the Kuwahara filtered image, the differences will be higher at the edges.

We have now obtained a binary mask that captures the edges of the objects (Figure 4e). However, small gaps can be found in the edge-mask and therefore the object is not fully enclosed. To remove these small gaps we apply a morphological closing operation with an elliptic structuring element of an empirically tuned size of 40x40 pixels (Figure 4f). We now have a solid ring and in order to get the full mask of the pollen grain we apply a hole-filling operation (Figure 4g) [Gon09]. The final binary mask is then used to crop the original (coloured) image by multiplying the binary mask with the original, producing the segmented object image (Figure 4h). For each separate object the bounding-box coordinates were also stored in a JSON file.

To save computation time, before processing a tile, a quick check is performed to skip empty tiles that do not contain any pollen grains. This function converts the image to Lab (CIELAB) colourspace [Gon09] and uses the a- and b-channels to detect the presence of pollen-like colour information. This has been empirically defined as foreground if the product of the average a- and b-channel values is greater than -5. Tiles that do not pass this test are omitted from further processing. All connected components in the tiles are then stored as separate images. Objects that touched the border were often incomplete and therefore not stored.

2.5.3 Object Filtering

As shown in Figure 4h not all segmented objects are complete single pollen grains. A few examples from the different slides are demonstrated in Figure 6.



(a) Semi cluster (b) Burst (c) Incorrect (d) Droplet (e) Background

Figure 6: Examples of objects produced during the segmentation process (not to scale).

Some pollen-objects form clusters (Figure 4a) either because they are touching each other or because there is a strong overlapping halo between the objects which complicates the segmentation process and often leads to clusters being captured as one object (reffig:couple). In addition, some pollen grains have burst (Figure 6b). This could happen because of the concentration of the solution and the amount of burst pollen can differ greatly between microscope slides, possibly due to the manner in which the slide is prepared. When pollen burst, the insides of the pollen grain can end up outside the grain and will often be detected as an isolated object as well. Then, there are cases where the segmentation process was unsuccessful (Figure 6c). This could be either because of complex colour patterns in the background, debris, or because the colours of the pollen grain are not as bright as other specimen. Furthermore, there are a few water droplets (Figure 6d) and other unidentified artifacts found in the tiles. Lastly, sometimes background is identified as an object because of slight stains in the background. To be able to run sensible classification experiments the segmented images need to be filtered so that only images containing single, correctly cropped pollen grains remain. In order to do this efficiently first up to 20% with a minimum of 200 objects per slide were classified manually using an interface tool that allows for fast manual annotation and the results were stored in `.csv` files. Multiple labels were assigned to the objects if the pollen grains were cropped well the number of pollen would be assigned, so for a single cropped pollen grain that would be 1, but for clusters the numbers are higher. Other labels are “burst” , “Incorrect crop”, or “other”. Only objects with the label 1 assigned are considered valid samples for classification. The remaining 80% was classified by a Support Vector Machine with a linear kernel that was trained on the manually assessed samples. The features we used three classes of features to train on usability that are listed below:

- **Shape and size features**: Area-size, Perimeter, Aspect-Ratio-Feret, Bending-Energy, Convex-Area, White-Pixels, Average-Edge-Colour
- **Intensity and colour features**: Mean- and Standard deviation, Minimal-, and Maximum-intensity
- **Texture features**: HOG

The Convex-Area represents the amount of pixels in the convex hull. The White-Pixels feature contains the amount of pixels in the grayscaled image that are above the threshold of 70. And lastly, the Average-Edge-Colour captures the average values for the red-, gree-, and blue-channel along the boundary of an object. We conciderd and edge band of two pixels wide on the inside of the edge. The other features will be explained in section 2.9.1. Before the SVM was used to classify the remaining unassessed samples the performance of the classifier was first tested on the manually assessed dataset with a 80-20 train-test distribution. The performance results are shown in Table 1. The distribution of the labels that were assigned both manually and automatically are demonstrated in Figure 7.

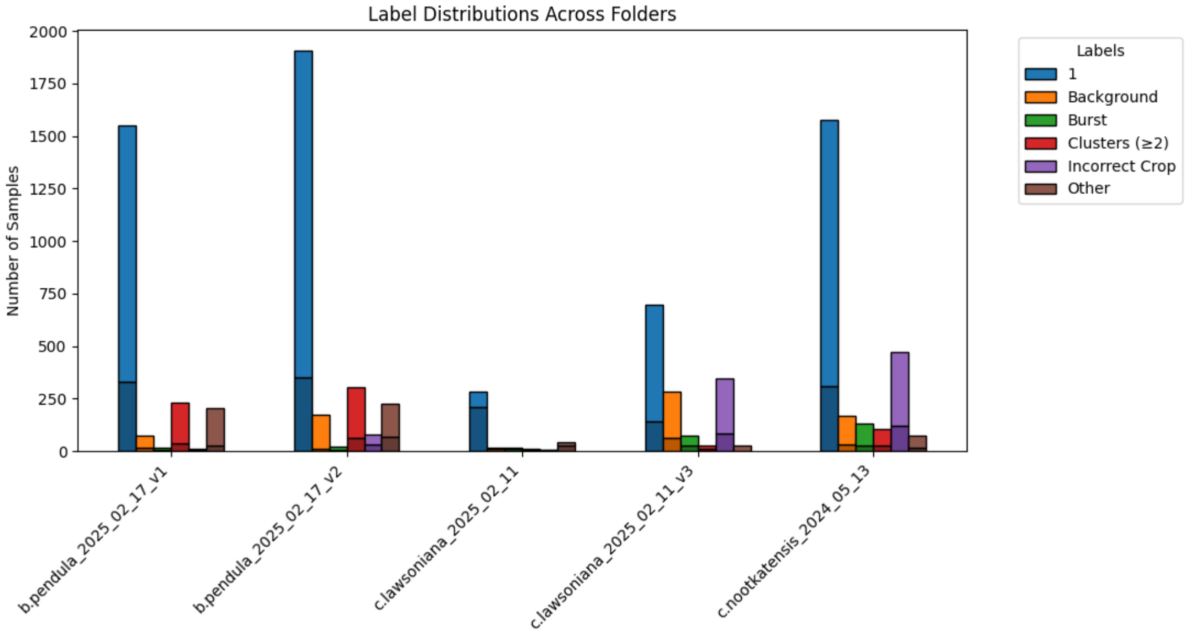


Figure 7: Barplot showing the distribution of the classification of the segmentation results. The dark colour represents manually assessed, whereas light colour is classified by SVM. The dates denote the scanning date of the sample, and the number following 'v' the version number.

The plot shows the distribution for each of the separate sets. The segmentation process produced more single pollen grains in some scans than in others, either due to colour differences or simply higher pollen counts. For example, the second *C. lawsoniana* scan (v3) and the *C. nootkatensis* scan contained brighter coloured grains (demonstrated in Figure 29 in Appendix Chapter B), leading to incorrect crops that included background. In the *B. pendula* scans, grains were more abundant and often overlapped, resulting in more clusters. The combined counts of usable pollen per species, used for further processing, are shown in the dark bars of Figure 8.

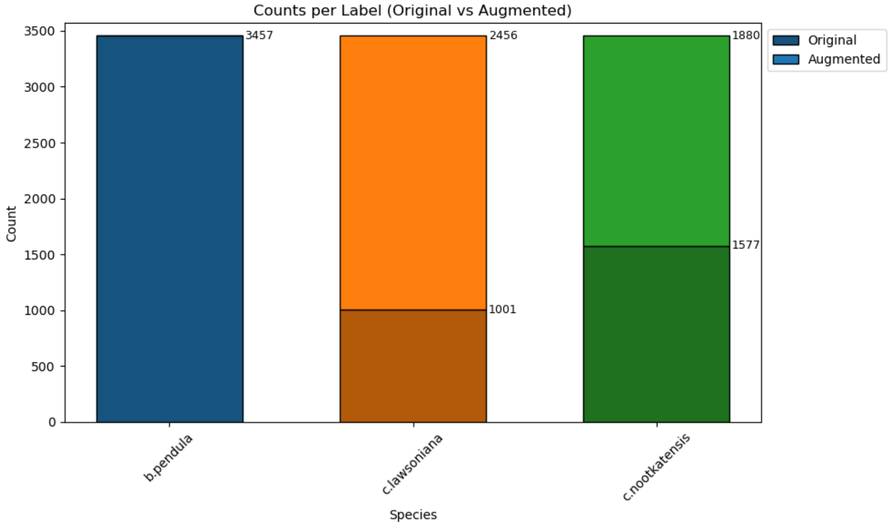


Figure 8: Barplot showing the distribution of usable pollengrain samples from the species *B. pendula*, *C. nootkatensis*, and *C. lawsoniana*. The dark colours denote the original data, the light colours denote the augmented data.

2.6 Data Augmentation

Figure 8 shows the distribution of single usable pollen grains after projecting, segmentation, and selection. The dark coloured parts show the original amounts of the acquired images, revealing the dataset is highly imbalanced, with more than twice as many samples for the *B. pendula* species compared to the other two species. Even though the classical machine learning methods like SVM do not require large amounts of training samples, deep learning models do. If the dataset is unbalanced, the model could develop a strong bias toward the overrepresented species, leading to unreliable performance. In order to amplify and balance the dataset we create synthetic data by performing classical data augmentation techniques on the remaining two species. While augmentation improves balance and provides sufficient training data for deep learning, it does not fully substitute the biological variation. The adjustments that were performed on the existing data consisted of a random selection from the following options:

- Flip-rotate
- Scale adjustment
- Brightness adjustment
- Hue adjustment

Pollen grains are not completely symmetrical, especially for *B. pendula*. Therefore, flip-rotate operations can generate new, realistic variations. For the *Cupressaceae* species, it is mostly the patterns inside the grains that are responsible for the asymmetry. Furthermore, within species, the samples differ in scale, brightness and hue (Appendix B, and C Figure 32a). The flip-rotate adjustments yield exactly 7 unique variations of the original, more combinations would result in an equivalent of these 7 or the original. The image can be flipped horizontally, vertically and over the two diagonal axis and are rotated a discrete $90°$, $180°$, or $270°$.
For the scale-, brightness- and hue-adjustments continuous parameters need to be selected that determine the amount of deviation from the original. In order to create a realistic synthetic dataset we measured the sample-specific average and standard deviation for each of these values. Since the values can differ greatly between species and perhaps between microscopic slides, the measurements were done for each of the samples separately. The results are summarised in Table 2. These statistics highlight inter-species differences in size and colour properties, which motivated the species-specific parameter selection for realistic data augmentation. For the brightness, hue, and size adjustments factors were randomly sampled within calculated ranges. The following formulas demonstrate how the factor-range was calculated based on the mean and standard deviation of these features on the original data:

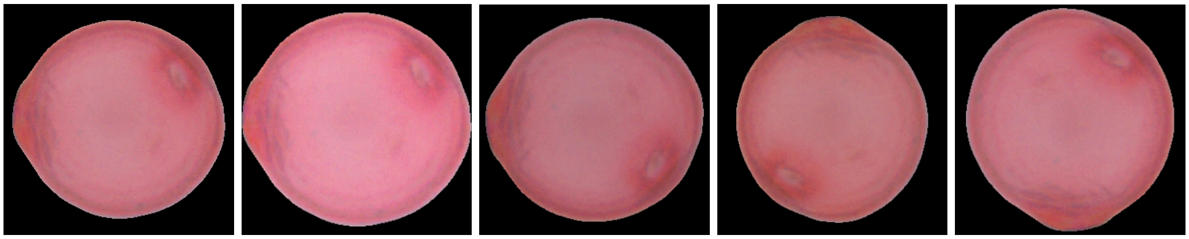
$$\text{Factor-range}_{\text{brightness}} = \left[1 - \left(\frac{\mu_{\text{brightness}} + \sigma_{\text{brightness}}}{\mu_{\text{brightness}}} - 1\right),\ \frac{\mu_{\text{brightness}} + \sigma_{\text{brightness}}}{\mu_{\text{brightness}}}\right] \tag{1}$$

$$\text{Factor-range}_{\text{hue}} = [-\sigma_{\text{hue}},\ +\sigma_{\text{hue}}] \tag{2}$$

$$\text{Factor-range}_{\text{scale}} = \left[1 - \left(\frac{\mu_{\text{size}} + \sigma_{\text{size}}}{\mu_{\text{size}}} - 1\right),\ \frac{\mu_{\text{size}} + \sigma_{\text{size}}}{\mu_{\text{size}}}\right] \tag{3}$$

Formulas 1, 2, and 3 have been formed empirically by manually assessing the visual realism of the artificial data. Note that hue is a circular range between $0°$ and $360°$, therefore range-values below $0°$ or above $360°$ wrap around back into the range.

At least one and at most 4 operations can be applied on a base image with a flip-rotate combination seen as one operation. The pollen grains that are used as a base for the adjustments are randomly selected with equal probability and duplicates in the resulting augmented data are actively avoided. The amount of augmented data is shown in the light coloured parts of Figure 8.



(a) **Original**

(b) **Scale**: 1.08
Hue: -3.35
Brightness: 1.125

(c) **Scale**: 1.03
Flip: horizontal
Brightness: 0.911

(d) **Scale**: 0.97
Hue: 4.38
Brightness: 0.937
Flip: main-diagonal

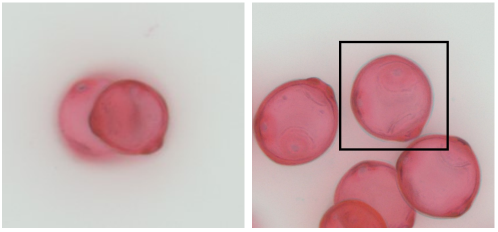
(e) **Scale**: 1.05
Brightness: 0.974
Flip: anti-diagonal

Figure 9: Examples of data augmentation techniques applied to pollen images. Each subfigure indicates the transformation parameters used. Padding has been added for visual alignment to illustrate size differences, no padding was applied to the images used for actual processing.

Figure 9 shows four examples of different combinations of the augmentation techniques being applied on a single *B. pendula* grain (Figure 9a). The created images look natural (Figures 9b–e). Though the pollen from *C. lawsoniana* or *C. nootkatensis* are more symmetrical than the ones from *B. pendula* and therefore flip-rotate operations will be less impactful. The augmented images will be used exclusively for training deep learning models.

2.7 Preparing the Backgrounds

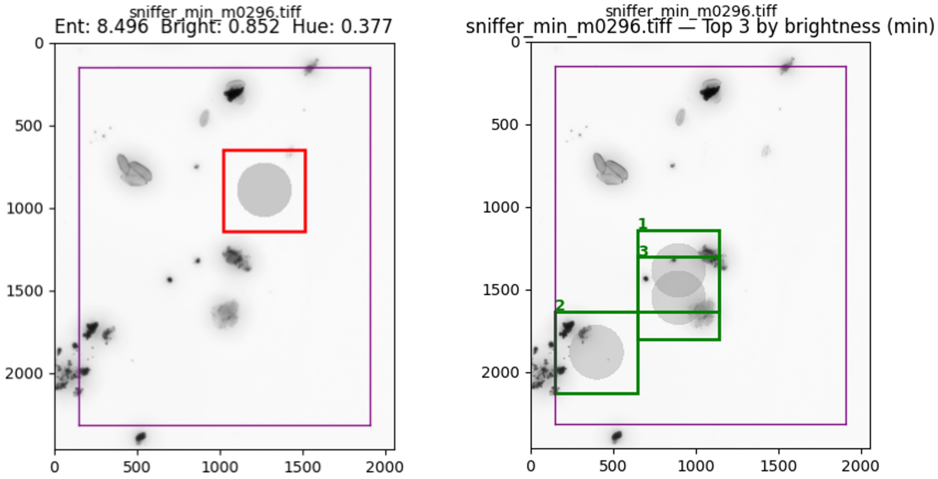
We now have a complete dataset of real and artificial pollen grains that have been precisely cropped around the edges. To measure the influence of the background, a logical solution would be to use the bounding-box coordinates to recover the background from the original images. The produced masks could then be used to calculate features with and without precise segmentation. However, due to the manner in which the data has been produced (described in section 2.1), slide scans contain very little background noise (Figure 4a). As a result, we artificially change the background by pasting the objects onto images provided by the airsniffer. The images produced by the airsniffer contain all sorts of dust, plastic debris, and biological materials, but have been selected to contain few pollen grains. Two examples of these sniffer tiles are shown in Appendix C Figure 32b and 32c. We only need a small window from the sniffer tiles to serve as background, however, not all parts of the sniffer tiles would make for an interesting background since some parts are almost completely white. We thus scan the sniffer tiles to search for areas of interest. A similar process is executed on the pollen tiles with the three species. As shown in Figure 10a, pollen grains can be scanned on top of each other. Figure 10b shows that if the bounding box is not super tight around the object, neighbouring pollen could appear in the bounding box especially if pollen are used that are parts of clusters. Pollen grains can also serve as backgrounds for other pollen grains.



(a) Overlapping (b) Neighbouring

Figure 10: Real examples of pollen occuring within the bounding box of an object of interest.

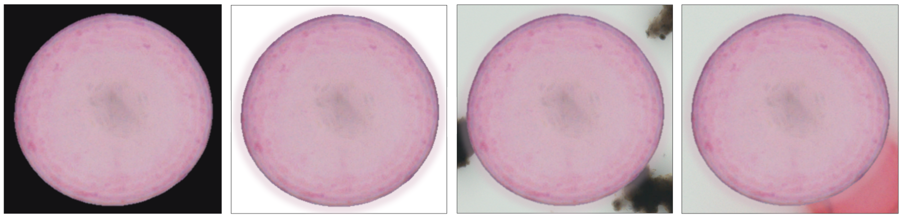
Firstly, a global scan was conducted that calculated the amount of shannon-entropy [Sha48] of the entire greyscale tiles, if the entropy was lower than 9.3 it would be unlikely to find a region of interest in that specific tile and so it would be excluded.
Tiles passing the global entropy threshold were scanned by a sliding window to identify three regions per tile that are suitable for background placement. The dimensions of the square window were equal to the maximum of the heights and widths of the found pollen grains (496 pixels). The boundaries of the tile are excluded, creating a buffer; a fixed 150 pixels from the edge. In case the background window is touching the inner buffer-edge, this buffer provides enough room for increasing the boundingbox size for future experiments.



(a) Searching (b) Found

Figure 11: The sliding window searching for regions of interest. The green boxes are the top three most interesting regions. The purple lines denote the boundarybuffer of 150 pixels.

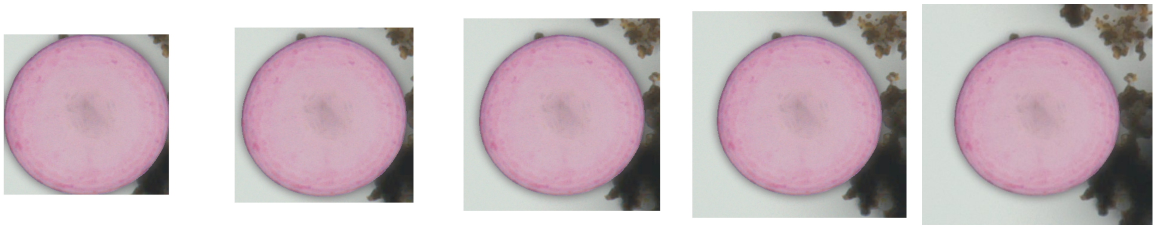
Within the square sliding window, a circle was placed to mark the future position of a pollen grain (Figure 11a), and only the pixels outside this circle were considered. To decide what regions are interesting, we measure the brightness, entropy, and hue, but only the brightness was used. The three regions with the lowest brightness are stored as potential background candidates. For each dataset 700 regions were selected. Later only background regions with a brightness lower than 0.7 (scaled between 0 and 1) were selected. Images would range from having spots in the background to completely covering the background.



(a) Segmented (b) White (c) Sniffer (d) Combination

Figure 12: Examples of a pollen grains with different type of backgrounds. d has a background containing *B. pendula.*

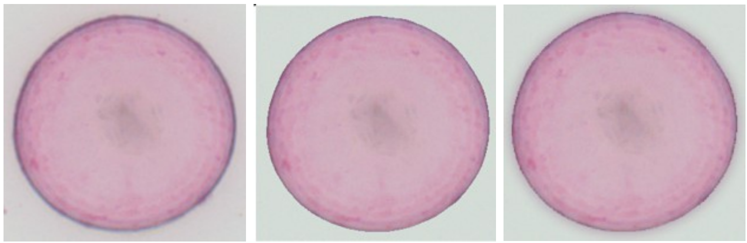
The earlier precisely segmented pollen grains are then pasted on top of the selected backgrounds. We randomly select a background for each pollen object with the object name $+repeatnumber$ as randomseed for reproducibility. The pollen grains are placed exactly in the center of the background. By default the background would exactly fit the dimensions of the pollen grain, but parameters can be set either to make all backgrounds the same size (the largest known dimensions), or to increase the background size either by scaling (×1.1, ×1.2, ×1.3, ×1.4) or by adding absolute pixel margins $\{+10, +20, +30, +40\}$ to all four edges while keeping the object size exactly the same (Figure 13). The box sizes were scaled according to the object size. This reflects the realistic scenario that bounding box precision decreases for larger objects compared to smaller ones.



(a) 1.0 (b) 1.1 (c) 1.2 (d) 1.3 (e) 1.4

Figure 13: Different background sizes.

The implementation also allows disabling the halo or using augmented data. The same image preparation methods were also applied on a completely white background. The white background could be used as a substitute for the precisely segmented images without background in the deep learning models since they do not accept transparent input and need set dimensions. In addition, the white background might be better for the background influence comparison since only the content of the background changes but other factors are completely identical to the images with sniffer background. We can also test if the constant white background gives different results from the precise segmentation. We have obtained four different types of object-background combinations: Precise segmentation (Figure 12a), White (Figure 12b), Sniffer (Figure 12c), and Combination (sniffer or one of all types of pollen) (Figure 12d).



(a) Original (b) Without halo (c) With halo

Figure 14: Pollen grain pasted onto a white microscopic background. (a) shows the original image, while (b) and (c) are artificial: the former without a halo, and the latter with one.

To make the pasted pollen grains appear more natural on different backgrounds, a soft halo was generated around each object. The natural halos are visible in Figures 4a and 10. The halo colour was computed as a weighted mix of the pollen edge, center, and underlying background colour, then slightly darkened to better mimic subtle shadowing. The transparency of the halo would smoothly decrease the further it gets from the object. The halo was blurred with a Gaussian filter, ensuring smooth transitions between object and background. As illustrated in Figure 14, the effect reduces sharp cutout edges and makes it visually more plausible.

2.8 Contrast Calculations

After the backgrounds were selected, we calculated three contrast values that describe how different the background is from the object: brightness contrast, colour distance, and histogram overlap. These measures allow us to quantify how different the background is from the object and to investigate how background contrast influences prediction performance. For example, if a *B. pendula* grain is placed on a *C. nootkatensis* background, we expect high contrast, and the extracted features may be biased toward *C. nootkatensis.* It is therefore interesting to examine how prediction performance changes as a function of these contrast values. The contrast values were calculated using the following formulas.

- **Brightness Contrast**:
$$C = \left|\mu_{\text{obj}} - \mu_{\text{bg}}\right| \tag{4}$$
where μ_{obj} and μ_{bg} are the mean greyscale intensities of the object and background, respectively.

- **Colour Distance (Lab)**:
$$d_{\text{Lab}} = \left\|\mu_{\text{Lab}}^{\text{obj}} - \mu_{\text{Lab}}^{\text{bg}}\right\|_2 \tag{5}$$
where $\mu_{\text{Lab}}^{\text{obj}}$ and $\mu_{\text{Lab}}^{\text{bg}}$ are the mean CIELab colour vectors of the object and background regions [SWD05].

- **Histogram Overlap**:
$$H_{\text{overlap}} = \sum_{i=1}^{N} \min\left(h_i^{\text{obj}},\ h_i^{\text{bg}}\right) \tag{6}$$
where h_i^{obj} and h_i^{bg} are the normalised greyscale histogram bin values for the object and background, and N is the number of bins [SB91].

2.9 Classification

In this subsection we will discuss the chosen features, parameters and algorithms that were used for training the classification models.

2.9.1 Features

For classical machine learning methods like SVM a vector of features needs to be calculated for each image that will consequently be used as input for the model. We can subdivide these features into roughly three categories:

- **Shape and size features**: Area-size, Perimeter, Convexity, Roundness, Aspect-Ratio-Feret, Bending-Energy
- **Intensity and colour features**: Mean- and Standard deviation, Minimal-, and Maximum-intensity
- **Texture features**: HOG, LBP, and GLCM

We shall henceforth briefly explain the features that are less intuitive.

HOG (Histogram of Oriented Gradients) captures the edge directions and gradient structures in an image. It works by dividing the image into small cells, computing the gradient orientations and magnitudes in each cell, and creating histograms of these orientations with 8 orientation bins where the magnitudes function as weights. In our implementation, images are first resized to a fixed size of 128×128 pixels to ensure that all resulting vectors will have the same size, and then divided into small cells of 8×8 pixels. The histograms are then normalised within each cell (Eq. 7) to improve invariance to illumination differences.

$$b = \frac{b}{\sqrt{||b||^2 + \epsilon}} \tag{7}$$

where **b** is the feature block and ϵ is a small positive constant to prevent zero division [Tom12]. We used one cell per block. The normalised blocks are then concatenated in a single vector. Because the vector is of considerable length, we apply Principal Component Analysis to reduce the length of the vector in to 50 elements as suggested by [WCS+19]. HOG is particularly effective for capturing shape and contour information and is widely used in object detection [DT05].

LBP (Local Binary Patterns) encodes local texture in a greyscale image by comparing each pixel to its surrounding neighbors. For each neighbor, a binary value is assigned:

$$s(p_i - p_c) = \begin{cases} 1 & \text{if } p_i \geq p_c, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

where p_c is the center pixel intensity and p_i are the neighboring pixel intensities and $s(p_i - p_c)$ represents the comparison between p_c and p_i. In our implementation we considered 8 neighbouring pixels and used a radius of 1. We used the uniform method which scans each of the LBPs to check if it has at most two transitions between 0 and 1 in the circular binary string it is considered uniform. Such uniform patterns present small textures like edges or spots. There are nine possibilities for rotations of the circular uniform LBPs those form nine bins in a histogram, one extra bin is added for the nonuniform patterns. The histogram is then normalised so that it becomes insensitive to differences in size. This results in a vector size of $\texttt{number of neighbors}+2 = 10$ elements. LBP is simple yet powerful for detecting fine-grained textures and patterns, and is robust to illumination changes [OPM02].

GLCM (Gray-Level Co-occurrence Matrix) captures image texture by quantifying how often pairs of pixel intensities occur at a given spatial relationship, defined by a distance d and angle θ. The GLCM is a square matrix of size $L \times L$, where L is the number of gray levels (256 for 8-bit images) in which each entry (i, j) represents how often a pixel with gray level i occurs with a neighbor of gray level j for a given (d, θ). In our implementation, we computed GLCMs for distances $d = 1$ and angles $\theta = [0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}]$ on 8-bit greyscale images with symmetry allowing both directions per angle. From each matrix, we extracted five standard texture features: contrast, correlation, energy, homogeneity, and dissimilarity, averaging them across all GLCMs. For instance, contrast is computed as

$$\text{Contrast} = \sum_{i,j} (i - j)^2 P(i, j) \tag{9}$$

Where P is the normalised GLCM that holds the probability for every entry $P(i, j)$. The other formulas are listed in Appendix chapter D. This procedure produces a 5-element feature vector, summarizing the statistical relationships of gray-level intensities in the image. GLCM features are particularly useful for surface analysis [HSD07].

Convexity measures how closely the shape of an object approaches a convex shape. It is defined as:

$$\text{Convexity} = \frac{\text{Area of Object}}{\text{Area of its Convex Hull}} \tag{10}$$

A completely convex object has a convexity of 1, whereas objects with concavities have values less than 1 [DIP23]. This feature is particularly useful for highlighting the small surface bumps observed in *B. pendula* grains.

Roundness assesses how circular an object is. It is often defined as:

$$\text{Roundness} = \frac{4\pi \times \text{Area}}{(\text{Perimeter})^2} \tag{11}$$

A perfect circle has a roundness of 1, and less circular shapes have lower values. This feature is useful for distinguishing nearly spherical pollen grains from more elongated or irregular ones.

Aspect-Ratio-Feret Measures the ratio of the minimum perpendicular Feret diameter to the minimum Feret diameter:

$$\text{Aspect-Ratio-Feret} = \frac{\text{Feret}_{\perp\text{min}}}{\text{Feret}_{\text{min}}} \tag{12}$$

This feature quantifies elongation, values close to 1 indicate nearly circular objects, and higher values indicate elongation.

Bending-Energy Quantifies curvature variation along the object boundary:

$$\text{Bending-Energy} = \sum_{i=1}^{N} (\theta_i - \bar{\theta})^2 \tag{13}$$

where θ_i is the angle between consecutive boundary segments and $\bar{\theta}$ is the mean angle and N the number of pixels on the edge [YWB74, BY77]. High values indicate more irregularities or bumps along the contour.

For the Size & Shape and Colour & Intensity features we used the DIPlib library [DIP23]. For the texture features, the `scikit-image` package was used [VdWSNI+14].

Most shape features are only meaningful when a precise object mask is used. For example, the feature `roundness` does not have any value when computed from bounding boxes. Therefore, we evaluate shape features only once to assess their performance. The feature `area-size`, however, remains relevant for bounding boxes, since larger pollen grains naturally have larger bounding boxes. Because many other size-related features are derived from or strongly correlated with `area-size` in the case of bounding boxes, we restrict our experiments to `area-size` alone. A similar reasoning applies to intensity and colour features. We use only the `mean-intensity`, as other intensity and colour descriptors are closely correlated with it. Therefore only five features are used for the background comparisons: area-size, mean intensity, HOG, LBP, and GLCM.

2.9.2 Performance Metrics

To effectively evaluate the prediction performance of the models we use the following metrics

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{15}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{16}$$

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{17}$$

For the classical machine learning methods we take the average accuracy over different folds. To get more robust results that are not dependent on a specific selection of backgrounds, we train and test the models 10 times and then report the average and standard deviation over those 10 prediction scores.For ResNet50, hard voting was used to obtain a single score from the 5 folds instead of averaging scores.But the model was trained three times and the average score over these three models was used as the final performance score. For averaging the precision, recall and F1 score, macro-averaging was applied to get one score for all classes.

2.9.3 Classification Algorithms

We applied both classical machine learning and deep learning methods to classify pollen images. We first tested three different classical machine learning classifiers: Support Vector Machine, Multi Layer Perceptron, and Gradient Boosting. For further steps, we limited the research to only two algorithms for the experiments and selected SVM and ResNet50.

Comparing classical machine learning with deep learning is valuable because they process information fundamentally differently. Feature vectors extracted from images may be more influenced by background noise, whereas deep learning models take pixel values directly as input and might better handle background variability. However, the features importance of SVM might be more interpretable.

Classical Machine Learning For the classical machine learning approach, we used an SVM. Feature vectors extracted from the images were first normalised using `scikit-learn`'s `StandardScaler()` to ensure that all features contribute equally, and to improve convergence. We specifically used an SVM with a linear kernel, as this allows for inherent feature selection by inspecting the learned weight vector, which is particularly useful for interpreting which features drive the classification. The model was evaluated using 5-fold stratified cross-validation to preserve class distribution. The data was shuffled and a fixed random seed was set to ensure reproducibility. For all initially tested classical machine learning methods, the hyperparameters were optimised using Bayesian search to create a baseline for later comparisons. Our goal was not to create a classifier with the highest prediction accuracy, but to investigate how different input types and preprocessing steps affect the resulting scores. Therefore, for comparing input types and feature representations, we deliberately avoided extensive parameter tuning and the hyperparameters were left at their default values as implemented in `scikit-learn`. In addition, if the model's performance is too high, differences in performance are less pronounced.

Convolutional Neural Network (CNN) For the deep learning approach, we employed a ResNet50 convolutional neural network ([HZRS16]), with up to 50 trainable layers. The architecture introduces residual connections that skip layers, which reduce the vanishing gradient problem and improve stable learning.
Previous research has demonstrated its strong performance in similar pollen classification tasks [LPC+23, GAS+24]. The network was both pretrained on ImageNet and then fine-tuned on our dataset and additionally trained on our dataset from scratch. We performed a stratified split of the data into training, validation, and test sets to preserve the class distribution, with a fixed random seed for reproducibility. Input images were resized to 224×224 pixels, which corresponds to the expected input size of the pretrained ResNet50. Although using the original image size would have preserved more details, our goal was to compare input types rather than maximise classification performance. We used an Adam optimiser with a learning rate of 1×10^{-4} and trained the network using the CrossEntropyLoss function. The architecture followed the standard ResNet50 design, where for training the pretrained network the first three residual stages (layers 1–3) were frozen, while the final residual stage (layer4) and the fully connected classification layer were left trainable. For training the model from scratch all layers were trainable. The activation function used throughout the network was ReLU. Training was performed with 5-fold cross-validation to provide robust performance estimates. The images were normalised using z-score normalisation before they were used as input. This makes the input more compatible with the pretrained network because ResNet50 is trained on ImageNet images that have been normalised, and it helps prevent gradients to explode or vanish.
An overview of the ResNet50 architecture is shown in Figure 15.

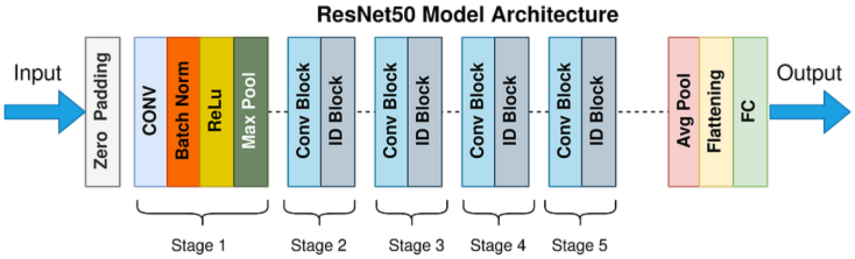


Figure 15: The ResNet50 architecture (image from Medium: The Annotated ResNet-50 [Ibr22]).

2.9.4 Feature Importance

Support Vector Machine For a SVM with a linear kernel, the classifying function is defined as:

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

Where the weight vector $\mathbf{w}$ indicates the contribution of each feature to the prediction and b is the bias. We use the way the model is built to extract the features directly from the decision formula by extracting the weights w_i. Features with a larger absolute weight ($|w_i|$) have greater influence on the final decision.

ResNet50 In order to investigate the relative importance of the feature channels in a given convolutional layer, we implemented iterative channel ablation. This procedure was implemented following these steps:

1. We first compute a baseline confidence score s_0 for target class y by passing an image through the network
$$s_0 = \text{softmax}(f(x))_y,$$
2. For each channel c that has not yet been deactivated, together with the channels that were already turned off in previous steps, replace feature map c with zeros ($A_c = 0$). Then pass the image through the network again and compute the new confidence score.
$$s_c = \text{softmax}\big(f(x; A_c = 0)\big)_y,$$
3. We compute the channel c^* that produces the largest drop in confidence relative to the previous step.
$$c^* = \arg\max_c \left(s_{t-1} - s_c\right).$$
4. Update the ablation set to include c^* and continue the procedure.

This process is repeated recursively until either all channels have been ablated. The outcome is a sequence of confidence values

$$(s_0, s_1, s_2, \dots),$$

that shows how the model's prediction degrades as progressively more channels are removed. The order in which channels are eliminated provides a ranking of their importance for the classification decision.

We applied this channel ablation procedure to the first convolutional layer (`conv1`) in stage 1 of the ResNet50 architecture (Figure 15) since this layer captures the most superficial and understandable features.
In addition to the first convolutional layer (`conv1`), we also visualised two deeper layers; the output of the third block in stage three (`layer3.2.conv3`) and the output of the third and final block in stage 4 (`layer4.2.conv3`), which is also the final layer of the entire network (Figure 15). On these layers we performed Gradient-weighted Class Activation Mapping (Grad-CAM) [SCD+17] and used this produce heatmaps that visualise the most important regions for the classification. This procedure was implemented following these steps:

1. Firstly, the input image is forwarded through the network. Activations from the selected convolutional layer are recorded using a forward hook creating the feature maps. There are C channels, each with dimensions $H \times W$.

2. We then back propagate to calculate the gradient of the target class logit y_c with respect to the activations of channel A_k. The gradients indicate the sensitivity of the class confidence score to a specific pixel in a given feature map.

$$\frac{\partial y^c}{\partial A_k^{ij}}$$

3. Now that we have the gradients for each pixel, we average them to decide the channel weight for each channel k:

$$\alpha_k = \frac{1}{H \cdot W} \sum_i \sum_j \frac{\partial y^c}{\partial A_k^{ij}}$$

4. Now that we have one weight α_k per channel, we take the weighted sum of all the activations in all channels and pass it through the ReLU function to filter out the negative values and obtain the Grad-CAM heatmap:

$$\text{Grad-CAM}_c = \text{ReLU}\left(\sum_k \alpha_k A_k\right)$$

   By passing the heatmap through the ReLU function we only highlight features that positively contribute to the class.

5. We then normalise the image between 0 and 1. Because the image resolution is now very low ($H \times W$), we upsample the image using bilinear interpolation to regain the original resolution so that the heatmap can be overlaid on the original image.

This visualisation demonstrated which regions of the image are most influential for the model's prediction.

3 Results

This section first presents intermediate results obtained during data preparation, followed by the outcomes of the classification experiments where we focus on explaining the results.

3.1 SVM Performance on Predicting Usability

When we selected the usable segmentation results, we used an SVM classifier that was trained and tested on manually assessed samples. Both the per-label accuracy and the accuracy for when all labels except 1 are grouped together are shown in Table 1.

Table 1: Usability classification scores per scan. Precision and recall are calculated for class 1.

Dataset	**Normal Accuracy**	**1 vs All Accuracy**	**1 vs All Precision**	**1 vs All Recall**
b.pendula_2025_02_17_v1	0.940	0.988	0.986	1.000
b.pendula_2025_02_17_v2	0.944	0.981	0.987	0.987
c.lawsoniana_2025_02_11	0.946	1.000	1.000	1.000
c.lawsoniana_2025_02_11_v3	0.954	0.969	1.000	0.933
c.nootkatensis_2024_05_13	0.914	0.981	1.000	0.967

It shows that the models are able to make highly accurate predictions regarding which found objects are usable. The higher the precision score, the fewer items are falsely classified as usable segments. With a minimum precision of 0.986 and a “1 vs All” accuracy of at least 0.969, the model demonstrates reliable decision-making. When manually checking the samples misclassified as usable segments, we observe that these contain only minor deviations that would not significantly impact the classification process. These misclassified objects consist of grains that are slightly out of focus or, in the case of *B. pendula*, consist of two grains overlapping for a large part. As shown in Figure 7, the *B. pendula* scans contain abundant grains and a relatively high number of clusters. The confusion matrices for all labels can be found in Appendix Chapter A.

3.2 Data Statistics for Data Augmentation

The statistics of the features size, brightness, and hue of the data are presented in Table 2.

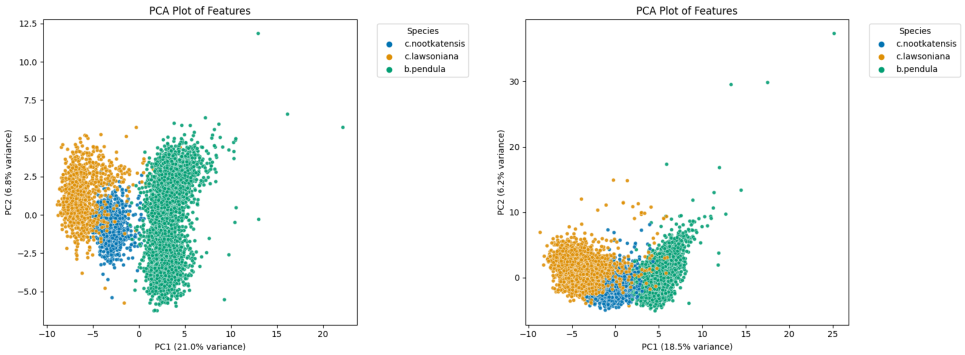
Table 2: Descriptive statistics (mean ± std) for brightness, hue, and size across datasets.

Dataset	**Brightness**	**Hue**	**Size**
b.pendula_2025_02_17_v1	198.02 ± 6.05	237.50 ± 10.13	237.82 ± 18.88
b.pendula_2025_02_17_v2	196.32 ± 5.91	232.62 ± 16.56	236.76 ± 22.69
c.lawsoniana_2025_02_11	198.36 ± 4.07	233.79 ± 0.65	374.92 ± 41.85
c.lawsoniana_2025_02_11_v3	202.41 ± 1.57	234.25 ± 0.80	394.09 ± 24.76
c.nootkatensis_2024_05_13	202.39 ± 1.27	234.13 ± 0.48	304.72 ± 20.61

The most evident difference shown is that the species are clearly separable by size. Furthermore, the differences in average brightness are rather minor, although the standard deviation is much higher for *B.pendula* and the first version of *C.lawsoniana*. The hue is also higher, specifically for the first *B.pendula* scan, and it has a larger standard deviation. The plots demonstrating the distribution of brightness, hue and size for each species separate and all species combined can be found in Appendix Chapter B.

3.3 Data Augmentation

In order to assess the quality of the artificial data produced by the data augmentation techniques described in Section 2.6, we plotted two PCA projections, shown in Figure 16. Figure 16a contains only the original samples, whereas Figure 16b includes both original and augmented samples.



(a) original data

(b) Augmented data

Figure 16: PCA plots based on all features extracted from precisely cropped pollen.

Figure 16 shows that the augmented data fills in the original data distribution and thus provides a good representation of the original data. The principal component values are somewhat smaller, indicating a decrease in variance after augmentation. This occurs because augmented data is derived from existing data points. Therefore, these new points will also be added in the space between outliers and the main cluster effectively smoothing the dataset. Since the variance is strongly influenced by outliers, the varience is reduced when the artificial datapoints are added.

3.4 Prediction Performances & Learning Curves

3.4.1 Machine Learning

Classifier Comparisons Before we perform experiments with several different parameters setting backgrounds, colourspaces and feature selections, we want to select a suitable classifier type. We did an initial experiment with Support Vector Machine, Multilayer Perceptron, and Gradient Boosting. The prediction performances for each of these algorithms for different colourspaces and with and without optimisation are shown in Table 3. We used all feature types (size, shape, colour, and intensity) as defined in Section 2.9.1, and no pre-selection of features was performed. For every experiment the same images were used. These results can be used as a baseline for further comparisons.

Table 3: Classification performance of SVM, MLP, and Gradient Boosting (GB). F1-score, precision, and recall are macro-averaged across 3 classes. Training time is measured in seconds.

Optimised	Model	Colour space	Accuracy	Precision	Recall	F1 Score	Training Time (s)
No	svm	RGB	0.977	0.977	0.977	0.977	3.23
No	svm	Gray	0.967	0.967	0.967	0.966	3.33
No	mlp	RGB	0.960	0.960	0.960	0.960	6.14
No	mlp	Gray	0.977	0.977	0.977	0.977	6.27
No	gb	RGB	0.990	0.990	0.990	0.990	19.65
No	gb	Gray	0.983	0.983	0.983	0.983	18.30
Yes	svm	RGB	0.987	0.987	0.987	0.987	25.72
Yes	svm	Gray	0.990	0.990	0.990	0.990	26.60
Yes	mlp	RGB	0.977	0.977	0.977	0.977	103.07
Yes	mlp	Gray	0.983	0.984	0.983	0.983	118.59
Yes	gb	RGB	0.987	0.987	0.987	0.987	395.85
Yes	gb	Gray	0.983	0.983	0.983	0.983	419.40

The table indicates that optimisation significantly improves the performance, but has to trade off in training time which in the cases of gradient boosting results in relatively long training time. However for gradient boosting, the scores did not improve compared to the default hyperparameters, this could be due to overfitting or because the optimiser simply could not find the optimal settings. In most cases the performance is better when the model is trained and tested on RGB images, but occasionally greyscale outperforms RGB images, like for MLP.
By removing colour information, greyscale forces the classifier to rely on size, shape and brightness-related features, which appear to be more beneficial for the MLP classifier. The optimal hyperparameters that were found by the Bayesian Search algorithm are presented in Table 4.

Table 4: Optimised hyperparameters for SVM, Gradient Boosting, and MLP on gray and RGB images. `hidden_layer_sizes` is abbreviated as `HLS`.

SVM Param	SVM-RGB	SVM-Gray	GB Param	GB-RGB	GB-Gray	MLP Param	MLP-RGB	MLP-Gray
C	3932.25	2674.86	**LR**	0.110	0.123	**alpha**	0.066	0.065
gamma	0.001	0.006	**MaxD**	2	6	**HLS**	225	151
kernel	rbf	rbf	**NE**	150	80	**LR_init**	0.041	0.028

We will continue further experiments with the five features mean intensity, area size, HOG, LBP, and GLCM. We select SVM as the classifier to use for further testing because it performs relatively well and fast. It is not the best performing classifier (Gradient Boosting achieved higher scores), but we do not necessarily need the best performance since our research focuses on differences in performances depending on input rather than optimizing the performance. For this same reason we will not optimise the performance any more by searching for the optimal hyperparameters, but use the default settings instead.

Table 5: Classification performance of SVM using only five features for different backgrounds (standard size). The precision, recall, and F1-score are macro-averaged across 3 classes.

Background	Colour	Accuracy	Precision	Recall	F1 Score
segmentation	RGB	0.974 ± 0.000	0.973 ± 0.000	0.973 ± 0.000	0.973 ± 0.000
	Gray	0.973 ± 0.000	0.973 ± 0.000	0.973 ± 0.000	0.973 ± 0.000
white	RGB	0.973 ± 0.000	0.973 ± 0.000	0.973 ± 0.000	0.973 ± 0.000
	Gray	0.970 ± 0.000	0.970 ± 0.000	0.970 ± 0.000	0.970 ± 0.000
sniffer	RGB	0.957 ± 0.000	0.957 ± 0.000	0.957 ± 0.000	0.957 ± 0.000
	Gray	0.958 ± 0.011	0.957 ± 0.011	0.957 ± 0.011	0.957 ± 0.011
combination	RGB	0.951 ± 0.010	0.951 ± 0.010	0.951 ± 0.010	0.951 ± 0.010
	Gray	0.948 ± 0.010	0.947 ± 0.011	0.947 ± 0.011	0.947 ± 0.011

Table 6: Classification performance of SVM using only five features for different backgrounds (size = 1.0). The precision, recall, and F1-score are macro-averaged across 3 classes.

Background	Colour	Accuracy	Precision	Recall	F1 Score
white	RGB	0.980 ± 0.000	0.980 ± 0.000	0.980 ± 0.000	0.980 ± 0.000
	Gray	0.983 ± 0.000	0.983 ± 0.000	0.983 ± 0.000	0.983 ± 0.000
sniffer	RGB	0.975 ± 0.009	0.975 ± 0.009	0.975 ± 0.009	0.975 ± 0.009
	Gray	0.977 ± 0.006	0.977 ± 0.006	0.977 ± 0.006	0.977 ± 0.006
combination	RGB	0.972 ± 0.007	0.972 ± 0.007	0.972 ± 0.007	0.972 ± 0.007
	Gray	0.972 ± 0.005	0.972 ± 0.005	0.972 ± 0.005	0.972 ± 0.005

Table 7: Classification performance of SVM using only five features for different backgrounds (size = 1.1). The precision, recall, and F1-score are macro-averaged across 3 classes.

Background	Colour	Accuracy	Precision	Recall	F1 Score
white	RGB	0.987 ± 0.000	0.987 ± 0.000	0.987 ± 0.000	0.987 ± 0.000
	Gray	0.984 ± 0.000	0.983 ± 0.000	0.983 ± 0.000	0.983 ± 0.000
sniffer	RGB	0.971 ± 0.007	0.971 ± 0.007	0.971 ± 0.007	0.971 ± 0.007
	Gray	0.976 ± 0.009	0.976 ± 0.010	0.976 ± 0.010	0.976 ± 0.010
combination	RGB	0.966 ± 0.005	0.966 ± 0.005	0.966 ± 0.005	0.966 ± 0.005
	Gray	0.968 ± 0.008	0.968 ± 0.008	0.968 ± 0.008	0.968 ± 0.008

Background Comparisons Tables 5–7 present the classification results of the SVM model for different background types, separated by colourspace and background size.

Table 5 shows results for images with a standard background size, set to the maximum observed object dimensions (except for precisely segmented images). Tables 6 and 7 correspond to backgrounds scaled to 1.0 and 1.1 times the object size, respectively.

Comparing precise segmentation with its white background representation, we observe that both yield similar scores, consistently outperforming sniffer and combination backgrounds across all three tables. This effect is most pronounced for the standard background size (Table 5), where accuracy decreases by 0.2 from white to combination background. In this case, the background area is relatively large compared to the object size for most pollen grains, since the background dimensions are set to the maximum observed object dimensions.

A similar decrease in accuracy is observed for the 1.1 times box size. However, overall scores are generally higher than those for the standard background, where the white backgrounds have an accuracy of 0.987 for standard box sizes compared to 0.973 for the standard box sizes. The higher accuracy for the white background with box size 1.1 mostly occurs because these experiments were not repeated using different backgrounds, since all white backgrounds are identical. Minor variations are expected when the image size changes, as this affects the division of cells for the HOG feature, as described in section 2.9.1.

For bounding boxes that precisely fit the object (Table 6), the accuracy drop from white to combination background is only 0.08. The smaller bounding box decreases the influence of the background information. The overall accuracy scores are slightly higher compared to the 1.1 bounding boxes.

We measured background sizes up to a scalar of 1.4; results for the remaining sizes are presented in Appendix E. Across all experiments, combination backgrounds consistently yield lower accuracy than sniffer backgrounds. The same pattern is observed for both RGB and greyscale images, although greyscale scores tend to be slightly lower.

These findings suggest that both background uniformity and background size influence classification accuracy.

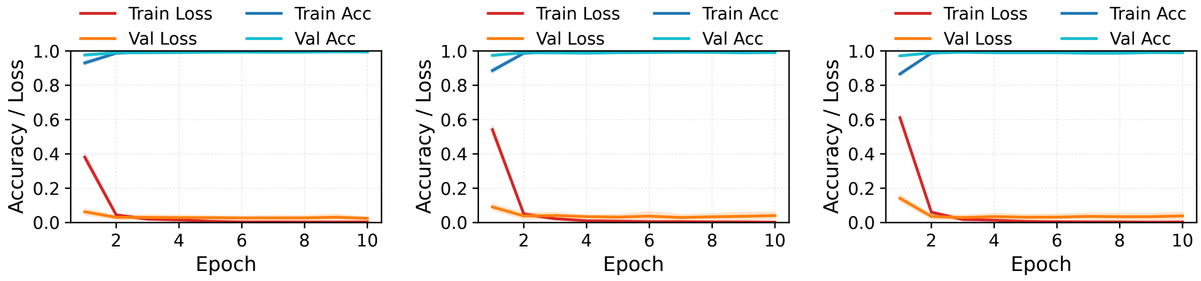
3.4.2 Deep Learning

We did a similar study comparing the prediction performance with different backgrounds for the deep learning model ResNet50. Table 8 presents the averages over three repeats of the training process. In order to test whether or not the colour of the segmented background is of any influence to the prediction performance we tested, in addition to white, the background colour black as well. Table 8 shows no significant difference between black and white. We can see that the model performs best on the combination background giving an average accuracy of 0.996. This score also outperforms all classical machine learning algorithms. We can see that the accuracy is about 0.01 lower for greyscale images compared to RGB, but between different background types no significant difference is noticeable. We do therefore not recognise the same relation between background types and prediction performance as for the SVM (section 3.4.1).

Table 8: Classification results for various backgrounds and colour spaces on a pretrained Resnet50 model. Values are averages over 3 models, each with 5 folds and 10 epochs. The precision, recall, and F1-score are macro-averaged across 3 classes.

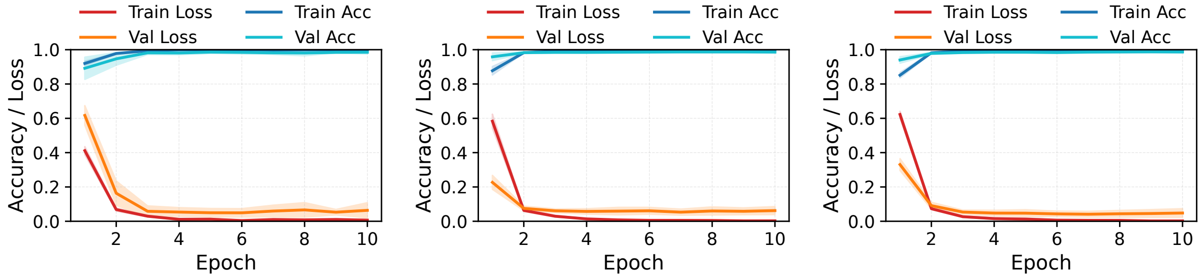
Background	Colour space	Accuracy	Precision	Recall	F1 Score
White	RGB	0.9944 (±0.002)	0.9944 (±0.002)	0.9945 (±0.002)	0.9944 (±0.002)
	Gray	0.9889 (±0.004)	0.9889 (±0.004)	0.9881 (±0.004)	0.9889 (±0.004)
Black	RGB	0.9947 (±0.003)	0.9947 (±0.003)	0.9948 (±0.003)	0.9947 (±0.003)
	Gray	0.9878 (±0.002)	0.9878 (±0.002)	0.9880 (±0.002)	0.9878 (±0.002)
Sniffer	RGB	0.9913 (±0.002)	0.9913 (±0.002)	0.9914 (±0.002)	0.9913 (±0.002)
	Gray	0.9878 (±0.003)	0.9878 (±0.003)	0.9880 (±0.003)	0.9878 (±0.003)
Combination	RGB	0.9960 (±0.003)	0.9960 (±0.003)	0.9960 (±0.003)	0.9960 (±0.003)
	Gray	0.9878 (±0.002)	0.9878 (±0.002)	0.9880 (±0.002)	0.9878 (±0.002)

For training and testing ResNet50 we split the dataset into a train-validation and test set. Not only the end score on the test set is interesting, but also the process of how fast the model learns to recognise the classes. This information is presented in the learning curves in Figures 17–20.



(a) White (b) Sniffer (c) Combination

Figure 17: Pretrained ResNet50 learning curves for RGB images with different backgrounds.



(a) White (b) Sniffer (c) Combination

Figure 18: Pretrained ResNet50 learning curves for grayscale images with different backgrounds.

We first look at the pretrained network that is trained on our data set for 10 epochs. The model was evaluated during training on both the validation data after every epoch and on the training data itself. For both the average values over three repeats are shown with the standard deviation in Figures 17 and 18 for RGM and greyscale images respectively. For all pretrained learning curves, the model starts with a training accuracy of about 0.9 and reaches nearly 1.0 after only two epochs, after which it stabilises. The validation accuracy is already close to 1.0 for RGB images in the first epoch and converges with the training accuracy after two epochs. In addition, for the RGB images, we can see that the curves are extremely stable showing no bumps and the curve has a very small standard deviation. However, for the greyscale images the initial validation accuracies are lower, but approach an accuracy of 1.0 after 2 epochs nonetheless. This suggests the model finds it slightly harder to learn the classes for greyscale images.

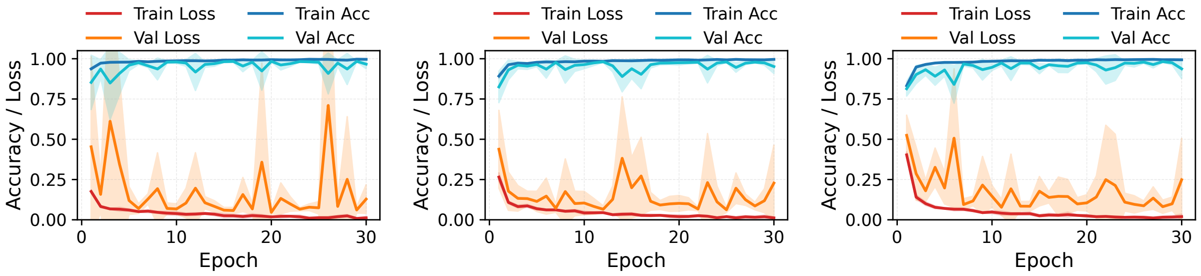
When we compare the different background types we can see very little difference for the RGB images, except for the training loss being higher before the second epoch and being less stable for sniffer and combination background.

A similar trend is exhibited for the greyscale images (18). For the white background however the standard deviation is much higher and the initial validation loss starts at a higher value compared to the sniffer and combination background. The high standard deviation might be explained by overfitting on the white background.

Table 9: Classification results for various backgrounds and colour spaces on a Resnet50 model trained from scratch. Values are averages over 3 models, each with 5 folds and 10 epochs; F1-score, precision, and recall are macro-averaged across 3 classes.

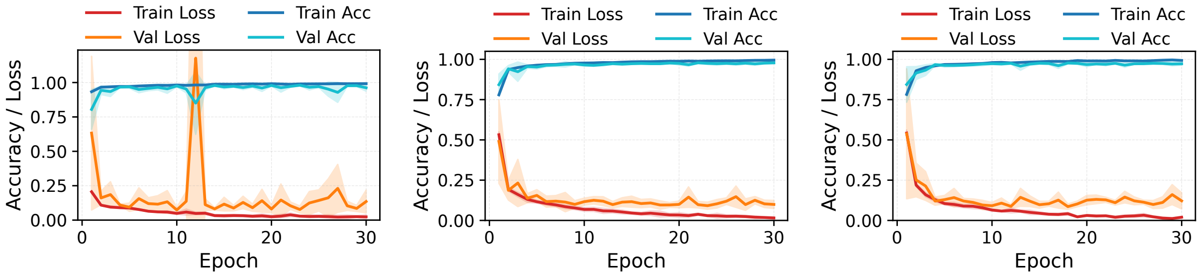
Background	Colour space	Accuracy	F1 Score	Precision	Recall
White	RGB	0.9908 (±0.004)	0.9908 (±0.004)	0.9909 (±0.004)	0.9908 (±0.004)
	Gray	0.9819 (±0.004)	0.9819 (±0.004)	0.9822 (±0.003)	0.9819 (±0.004)
Sniffer	RGB	0.9864 (±0.003)	0.9864 (±0.003)	0.9864 (±0.003)	0.9864 (±0.003)
	Gray	0.9833 (±0.003)	0.9833 (±0.003)	0.9835 (±0.003)	0.9833 (±0.003)
Combination	RGB	0.9897 (±0.002)	0.9897 (±0.002)	0.9898 (±0.002)	0.9897 (±0.002)
	Gray	0.9831 (±0.002)	0.9830 (±0.002)	0.9832 (±0.002)	0.9831 (±0.002)

We consequently trained the ResNet50 model from scratch to see if differences in the learning curve would be more visible across background types if the network had not been pretrained. The final average test results are demonstrated in Table 9. If we compare it to Table 8 we can see that the scores are unsurprisingly lower for the sniffer and combination backgrounds. Amongst background types however, no substantial differences are exhibited, similar to Table 8.



(a) White (b) Sniffer (c) Combination

Figure 19: From scratch ResNet50 learning curves for RGB images with different backgrounds.



(a) White (b) Sniffer (c) Combination

Figure 20: From scratch ResNet50 learning curves for Gray images with different backgrounds.

The training curves are shown in Figures 19 and 20 for RGB and greyscale respectively. We now train for 30 epochs since training from scratch usually takes longer to converge. The first aspect that draws the attention is that the learning curve is far less stable compared to the pretrained network and standard deviation is substantially higher. For greyscale images the learning curve is more stable than for the RGB images, with the exception for white background where there is a peak in validation loss after 10 epochs. This is likely due to overfitting. Furthermore, no noticeable differences between the different background types are visible.

3.5 Contrast and Prediction Performance Correlation

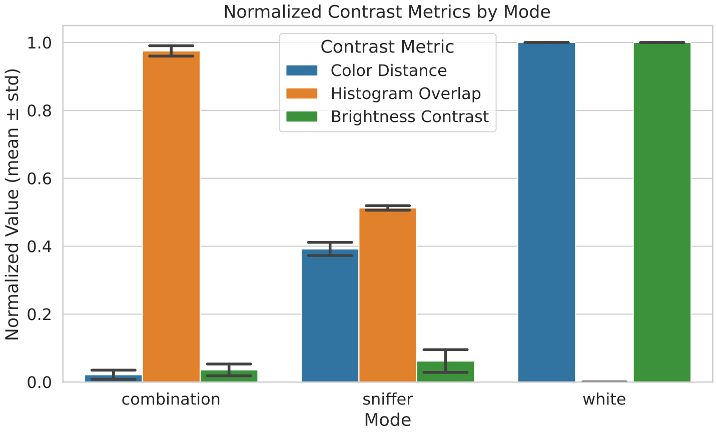


Figure 21: Normalised contrast values per background type

The normalised contrast metrics per background type as were explained in section 2.8 are shown in Figure 21. The results indicate that the three background types have clearly distinct contrast profiles.

The white background stands out as an outlier: brightness contrast and colour distance are significantly higher compared to the sniffer and combination background and the histogram overlap is almost zero. The sniffer and combination backgrounds display lower contrast values, with the contrast being generally higher for combination backgrounds compared to sniffer. The most prominent difference between sniffer and combination background is that sniffer backgrounds almost completely consist of black-gray-brown coloured objects, whereas for the combination background set this only makes up only 25% of the background tiles. The other background tiles consist of pollen and therefore share the red-purple colours that the object has resulting in a lower colour distance and higher overlap.

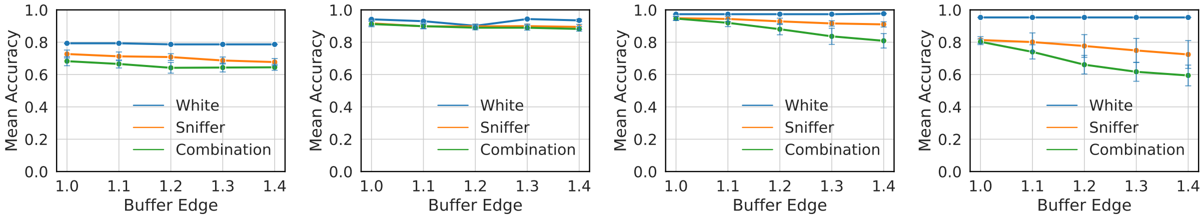
Table 10: Spearman rank correlation between classification accuracy and contrast values per feature on RGB images. P-values show the statistical significance of the correlations. CD = colour distance, HO = histogram overlap, BC = brightness contrast.

Feature	Corr. CD	p-value CD	Corr. BC	p-value BC	Corr. HO	p-value HO
HOG	0.319	3.55e^{-46}	0.316	4.22e^{-45}	-0.254	2.55e^{-29}
LBP	0.483	4.98e^{-111}	0.581	1.13e^{-171}	-0.288	2.07e^{-37}
GLCM	0.369	5.81e^{-62}	0.458	1.62e^{-98}	-0.208	9.03e^{-20}
Mean	0.369	5.17e^{-62}	0.446	1.49e^{-92}	-0.216	2.18e^{-21}

We examined whether contrast-related metrics are correlated with model accuracy for one feature at the time. To assess this, we computed the Spearman rank correlation [Wis05] between the accuracy and the three contrast metrics: colour distance, histogram overlap, and brightness contrast. We also report the corresponding p-values, which indicate the statistical significance of the relations. The results shown in Table 10 reveal clear patterns across feature sets. Both colour distance and brightness contrast are positively correlated with accuracy. This indicates that larger colour differences and greater brightness differences between pollen and background generally improve the classification performance. In contrast, histogram overlap shows consistently negative correlations, suggesting that when the colour distributions of pollen and background are more similar, the discriminative information available to the classifier is reduced and the accuracy decreases. The LBP feature is most sensitive to the contrast values, whereas HOG a a relatively low sensitivity, likely because it mainly looks at edge texture. All found correlations have very low p-values (< 0.001), emphasising that these relationships are highly robust. The results highlight the importance of contrast between the object and the background for accurate pollen classification by SVM.

3.6 Background Size Experiments

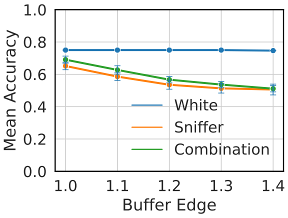
We experimented with different background sizes for each background type. Our starting point is `1.0` times the object size where the box precisely fits the object. We increase the box size up to `1.4` times the object size. We measured the effect for each feature separately and the results are shown in Figure 22. The experiment was repeated 10 times with different background selections for sniffer and combination backgrounds and the average and standard deviations are reported in the plots. For each change in box size, all other experiment factors stayed the same, meaning that the exact same images were presented for every different box size.



(a) Mean (b) Texture PCA HOG (c) Texture LBP (d) Texture GLCM

Figure 22: Comparison of feature performance across different background sizes. Mean intensity is computed on RGB images, while HOG, LBP, and GLCM are extracted from grayscale images.

Mean For the mean intensity, the white background is almost a constant that is added to each channel, with very slight fluctuations caused by differences in area size of the object that become more influential as the background size increases. We can notice a slight drop in accuracy for images with sniffer and combination background (figure 22a). As the background size increases, more noise is added to the average intensities of each channel, which confuses the classifier. The combination background exhibits this effect even more strongly than the sniffer data, which is expected since the combination backgrounds are more diverse. However, the performance for the two background types seems to converge at boxsize 1.4.



(a) Mean

Figure 23: Comparison of different feature types on grayscale images with different background sizes

We have repeated this experiment using greyscale images, the result is shown in Figure 23. For the mean intensity feature, we observe a significant difference compared to the RGB images. Unlike the RGB images, changing the box size has a substantial impact on prediction performance for the greyscale version. The difference between the sniffer and combination backgrounds appears smaller and converges at box size 1.4. This is likely because for greyscale images the difference in colour distance and histogram overlap between sniffer and combination backgrounds is not as predominant compared to RGB images. When converting to greyscale this colour difference between sniffer and combination backgrounds (Figure 21) is reduced and this is likely the cause of the plots being more similar in Figure 23.

HOG The stable, slightly declining trend for the HOG features (22b) suggests that HOG is hardly sensitive to background size. This can be explained by the fact that HOG primarily represents strong edges. Since the pollen objects are artificially pasted onto the backgrounds, the object boundaries, despite the addition of an artificial halo, produce much stronger gradients than the natural edges present in the dusty background. The small drop for the white background at 1.2 is most likely the result of minor fluctuations in performance, which become more noticeable here because the white-background experiments were conducted only once, in contrast to the dusty and combined backgrounds. Moreover, since HOG divides the image into cells and computes gradients within orientation bins, changes in the relative position of the pollen object when enlarging the background can slightly alter the distribution of gradients. These small differences in the resulting HOG feature vectors can lead to subtle fluctuations in prediction accuracy.

LBP For the Local Binary Pattern feature, a clear difference in sensitivity to background size can be observed for the three background types (Figure 22c). The white background appears completely unaffected to increasing background size. This can be explained by how LBP works. The patterns describing the edges between the pollen grain and the white background remain the same regardless of the box size. As the size increases a constant amount of completely white (only one's) LBP's will be added to one bin which will be added to all images and therefore not influence the performance.

For the sniffer and combination backgrounds, however, show a significant decrease in prediction performance, especially the combination background, which drops by almost 0.2 in accuracy from box size 1.0 to 1.4. The sniffer and combination backgrounds contain both objects and slight colour changes that change the distribution of the bins. The combination dataset disrupts the distribution less consistently due to its heterogeneous nature which complicates the classification process.

GLCM For the GLCM features (contrast, dissimilarity, homogeneity, energy, and correlation) a distinct trend can be observed over the different background types (Figure 22d). The white background remains completely stable, maintaining an accuracy of about 0.953 for all box sizes. In contrast, both the sniffer and combination backgrounds show a strong sensitivity to the increasing background size, with performance dropping from roughly 0.8 at box size 1.0 to about 0.7 and 0.6, respectively, at box size 1.4. The decrease is particularly steep between box sizes 1.0 and 1.2, the decrease becomes more moderate between 1.2 and 1.3, and then slows considerably down towards 1.4, suggesting that the effect of the background noise reaches a plateau.

This behaviour can be interpreted by the way the Gray-Level Co-occurrence Matrix represents image textures, as was explained in section 2.9.1. For increasing a uniform white background, only the values in the row/column for greylevel 255 would increase, especially the 255–255 entry. All object-related gray-level co-occurrences remain unchanged. This would be done very consistently for all images in the dataset and therefore the GLCM features are largely unaffected.

For noisy backgrounds, however, additional gray-level pairs are introduced throughout the matrix, disrupting the statistics that capture texture patterns of the pollen object. The combination background, containing more variety than the sniffer background, causes the least consistent disturbances, leading to the strongest decline in accuracy.

3.7 Feature Importance

To better understand the way the prediction models assess the images and decide on their classification we extract feature importance from the models. We first discuss the SVM and then ResNet50. Figure 24 shows the feature importance of all the features we initially used for SVM, only forty most important features are shown. The feature importance was decided using the method for SVM explained in section 2.9.4 and the absolute values were taken for the plot. The first thing to notice is that there are no shape features in the plot, therefore these were deemed insignificant to the classification. In contrast, colour features are the most important, followed by the first principle component of the Histogram of Oriented Gradient and size features. In this combined feature extraction HOG, LBP and GLCM features seem to be of similar importance. We can compare the features size, mean intensity, HOG, GLCM, and LBP by examining the initial values in the plots shown in Figure 22. When training and testing using only the mean intensity, the accuracy scores are generally lower than for the other features (HOG, LBP, GLCM), reflecting its limited ability to capture object-specific information. The white background achieves the highest performance at around 0.8 accuracy, while sniffer and combination backgrounds are lower. LBP achieves the highest accuracy amongst all three background types for a precisely fitting box size, likely because it encodes local texture patterns that are highly specific for the pollen species. GLCM performs well only for the white background, whereas its performance drops substantially for the sniffer and combination backgrounds, starting at about 0.8 accuracy, due to the disruption of texture statistics by heterogeneous background noise. HOG exhibits robust performance across all background types, reflecting its insensitivity to background variations, although its initial accuracy is slightly lower than LBP for small box sizes.

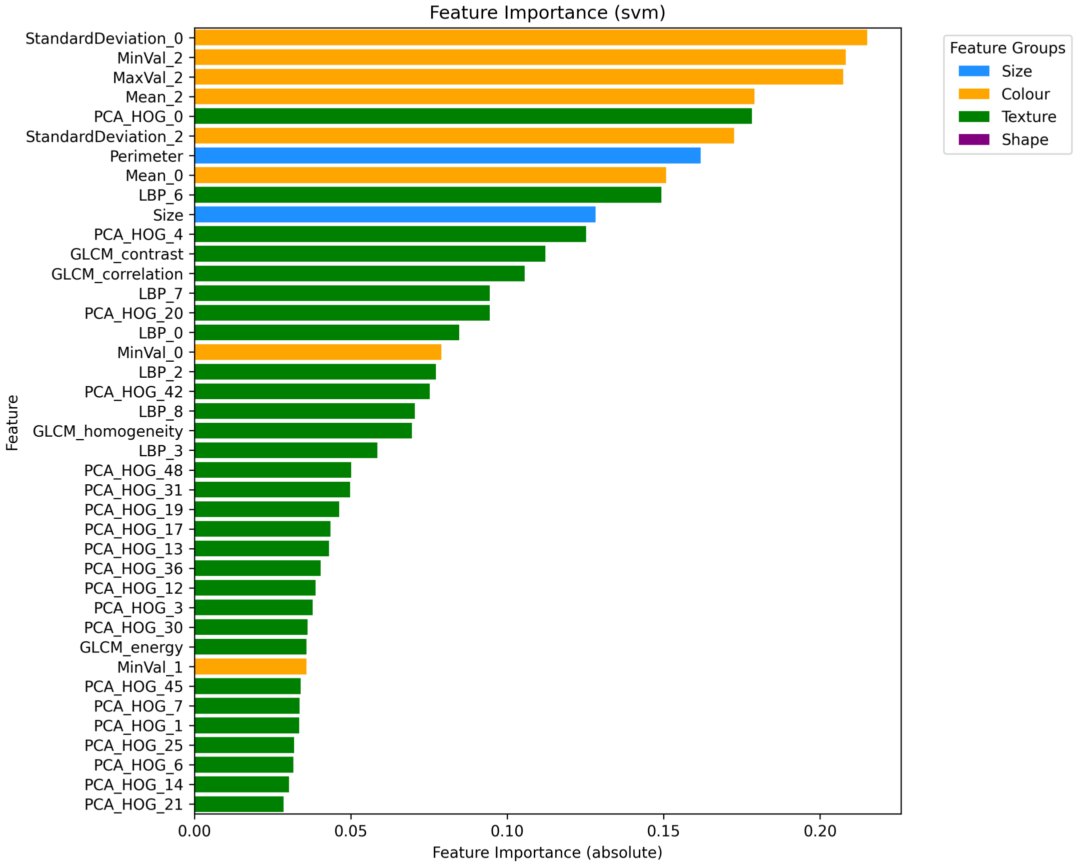
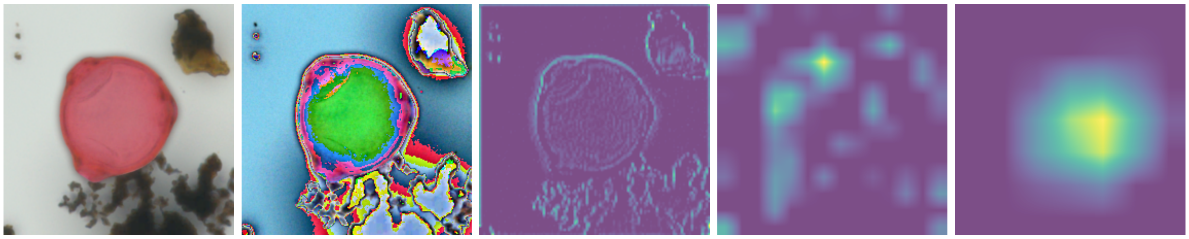


Figure 24: The forty most important features for the SVM model using precise segmentation on RGB images. The numbers for the colour features denote R(0), G(1) and B(2) channels. For the HOG they denote the principal components.

Figure 25a shows an example of *B. pendula* on a randomly selected background. Before the image is passed through the network the image is standardised resulting in the image shown in Figure 25b. In order to highlight the features we created Grad-CAM images of three layers. The Grad-CAM of the first convolutional layer `conv1` is shown in Figure 25c where we can see that mostly texture features are highlighted both of the pollen object and debris in the background. Especially pixels in the edges are activated, but also on the inner surface of the pollen grain a lot of activated dots are found.

In layer `layer3.2.conv3` (Figure 25d) the areas of interest become more precise, although still spread over the image. The two brightest parts of the heatmap coincide with the intersection of edges describing the characteristic external bump on the *B. pendula* grain.

Lastly, Figure 25e shows the Grad-Cam of the last image where we can see a clearly highlighted area which overlaps with the right side of the pollen grain. Note that this is also the side on which the most background dirt is positioned suggesting that the model does slightly take background noise into consideration for the final classification.



(a) Original (b) Standardised (c) conv1 (d) layer3.2.conv3 (e) layer4.2.conv3

Figure 25: Visualization of the ResNet input and feature attention. a shows the raw input image, b the channel-wise standardized version used for training. c–e display Grad-CAM heatmaps at increasingly deeper layers of the network, illustrating how the model progressively focuses on the pollen grain and suppresses background information.

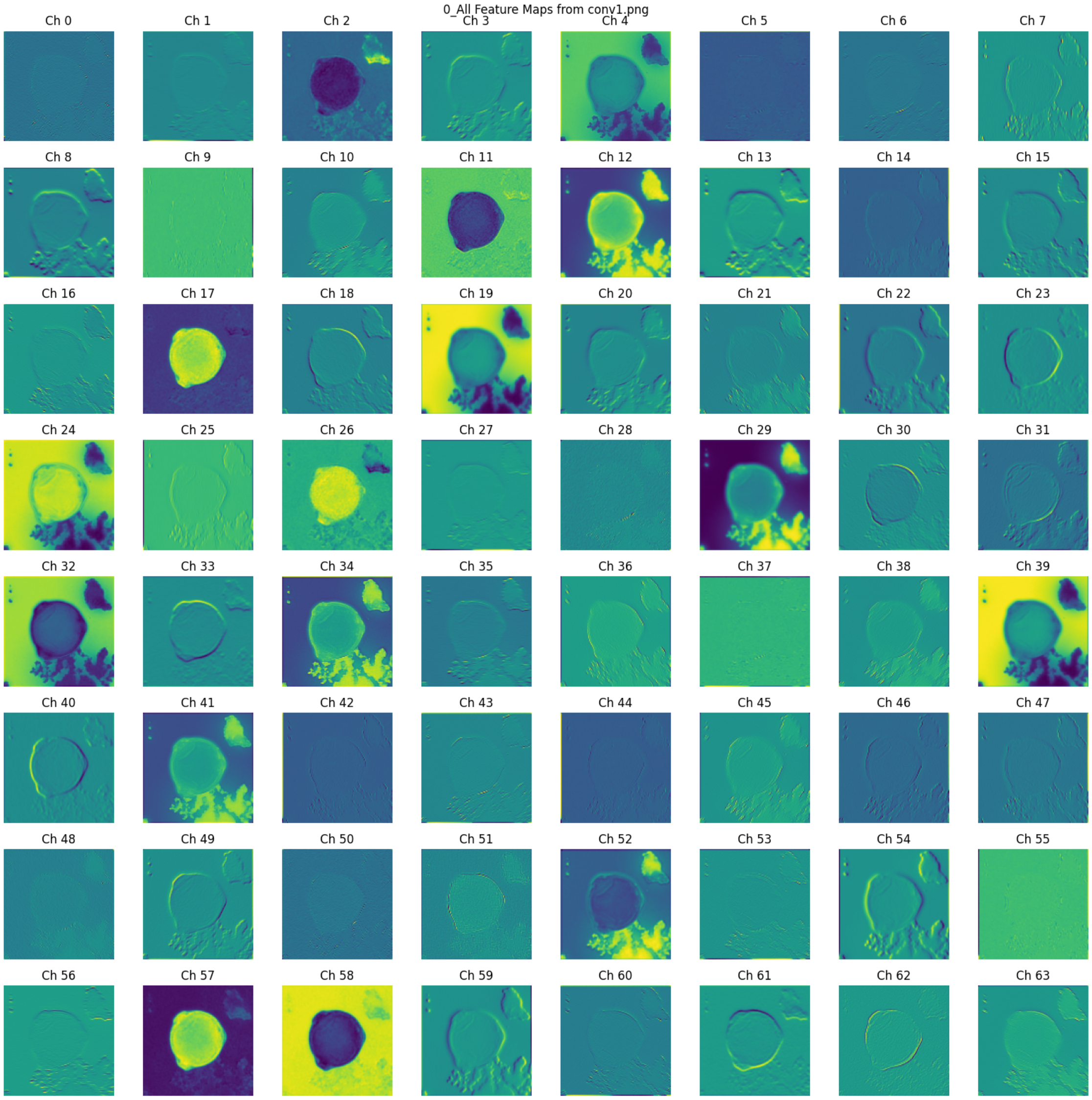


Figure 26: Featuremap showing the channel activations of the first convolutional layer in ResNet.

Table 11: Channel ablation results for the first convolutional layer sorted on ablation order. The baseline confidence corresponds to the unmodified input. Cells with orange confidence indicate an increase compared to the previous entry. The baseline confidence with all channels activated is 0.9998.

Ch.	Conf.	Ch.	Conf.	Ch.	Conf.	Ch.	Conf.
17	0.8386	13	0.1319	54	0.1133	20	0.1431
57	0.5337	59	0.1256	6	0.1093	49	0.189
29	0.3648	52	0.1311	9	0.0984	7	0.2078
0	0.2533	53	0.1354	33	0.0947	16	0.1511
30	0.2161	1	0.1329	63	0.1176	61	0.1759
5	0.2135	45	0.1288	3	0.1251	10	0.2165
42	0.208	41	0.1346	56	0.1291	40	0.269
22	0.1997	37	0.1349	2	0.0661	50	0.3222
51	0.1849	47	0.1262	15	0.0901	39	0.3201
21	0.1755	46	0.1357	55	0.1161	11	0.316
44	0.1584	8	0.1229	60	0.1099	62	0.3186
48	0.1377	27	0.1077	28	0.1103	26	0.3262
38	0.1247	12	0.1241	43	0.1038	18	0.3392
4	0.1239	19	0.11	35	0.1184	31	0.3143
24	0.1239	25	0.0976	23	0.122	36	0.34
32	0.1239	14	0.0992	34	0.1228	58	0.3451

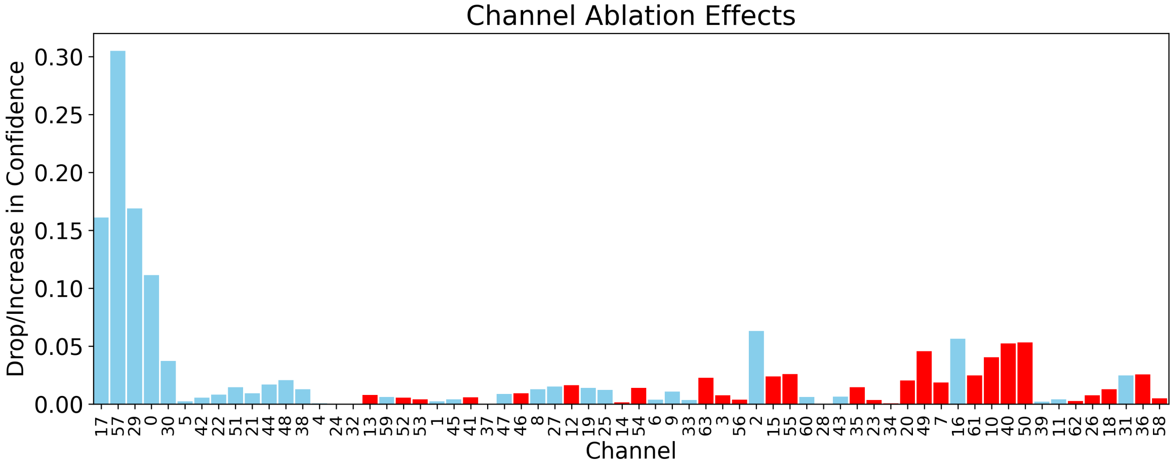


Figure 27: Drop in confidence score after channel ablation in the first convolutional layer of ResNet50.

In Figure 26, the channel outputs of ResNet's first convolutional layer are shown using the same input image as in Figure 25a. Bright yellow regions indicate activation, whereas dark blue regions denote deactivation. Interestingly, in the top two channels (57 and 17), the pollen grain is almost fully activated while both the white and dirty background are deactivated, effectively mimicking a segmentation mask. To assess the contribution of each channel to the prediction confidence score of the correct label (*B. pendula* in this case), we conducted a channel ablation study. The results are presented in Table 11. Channels that rank higher exert a stronger influence on the confidence score when they are deactivated. In addition, Figure 27 illustrates the confidence drop for each ablation, where confidence increases are shown in red. Notably, the two channels that precisely activate the pollen grain also rank highest in the ablation study, and when both are removed, the confidence score decreases significantly. Channel 29, which specifically highlights the background, also ranks high in the study. Detecting the background may aid later layers in segmenting foreground from background. Channels such as 0, 5, and 48 mainly activate object edges; in particular, channels 5 and 48 highlight the boundary between the pollen grain and background debris. Channels that simultaneously activate features in both the grain and the background debris (e.g., 12, 13, and 41) actually improve the confidence score when removed. In contrast, channels where these regions are specifically deactivated (e.g., 19 and 39) rank relatively low in the ablation study. The lowest-ranking channels (11, 62, 26, 18, 31, 36, 58) strongly activate both the grain and either the background dirt or the entire background. This indicates that channels responding indiscriminately to both foreground and background tend to be unhelpful for the model's classification task. Overall, the distribution of channel weights in the first convolutional layer gives the model an inherent segmentation property, making it relatively robust to background noise. Also because ResNet50 has multiple channels it is not dependent on one, if a single channel activates on background information, others can restore the segmentation. Combined with texture-sensitive channels and higher-level features extracted in subsequent layers, ResNet50 proves to be a highly effective classifier with resilience to unpredictable backgrounds.

4 Conclusion & Discussion

In this section we will summarise our findings and answer the research questions stated in section 1.3. We will also discuss the limitations of this thesis and propose suggestions for future work.

4.1 Discussion

Our main research question was: *"How does background information influence the prediction performance in pollen classification using three types of pollen comparing machine learning and deep learning?"*. We have conducted several experiments to come to a conclusion. To summarise the results and answer the main research question we address the subsections:

"Does greyscale vs RGB input images make a difference in the performance?" We observed that for classical machine learning algorithms, greyscale images can sometimes outperform RGB images, in particular when model hyperparameters are optimised. This is likely due to feature redundancy in RGB images, to which SVM and MLP are more sensitive compared to Gradient Boosting. When combined with feature pre-selection, predictions on RGB images are expected to achieve higher performance. Texture features such as HOG, LBP, and GLCM are automatically computed on the greyscale version of the images. However, for the mean intensity feature, we found that greyscale images were substantially more sensitive to background information as the size of the background increased (Figure 23).
For the deep learning model ResNet50, RGB images consistently outperformed predictions based on greyscale images. This difference is likely because RGB channels provide additional discriminative information that helps the model separate the stained pollen grain from background noise.

"How does the boxsize influence the prediction performance?" We found that increasing the bounding box size influences prediction performance when using a specific subset of features: LBP, GLCM, and mean intensity (for greyscale images). For these features, an increasing bounding box progressively introduces background information that perturbs the description of the object. In contrast, the HOG feature is largely insensitive to this effect, as it primarily captures edge textures. When all features are combined, the overall performance remains relatively stable, since feature pre-selection can reduce the impact of redundant or noisy features. For deep learning, we used a fixed bounding box size based on the largest pollen grain. Due to the large variability in pollen grain sizes across species, this meant that the relative background proportion ranged roughly between 1.0 and 1.5 times the object size. Nonetheless, prediction performance remained very high, with a minimum accuracy of 0.9913 for RGB images with sniffer background.

"How do variations in background characteristics impact prediction performance and the contribution of specific features?" When we tested four types of features on segmented images we found that shape features do not significantly contribute to the classification performance. Colour and texture features appear to be the most influential. Individual feature testing showed that overall, LBP appears to be the most effective feature for achieving a maximum accuracy for a precisely fitting box size, while HOG may be a more reliable choice as the box size increases. For segmented images on a white background LBP is a stable and best performing feature reaching an almost perfect accuracy. The contrast correlation study has shown that images with a lower contrast between object and background generally lead to lower prediction accuracy. The features LBP, GLCM and mean intensity are more sensitive to this compared to the relatively insensitive HOG feature.

"How does the influence of background information on prediction performance differ between classical machine learning models and deep learning models?" We have seen consistent evidence that classical machine learning models are sensitive to the different types of background, performing best when the background is completely white, decreasing the accuracy for sniffer background, and reaching the lowest accuracy for combination backgrounds. Such a trend was not visible for the deep learning model ResNet50, appearing completely insensitive to background changes. The feature visualisation has shown that ResNet50 has the capability to function as a segmentation mask and finding the region containing the pollen object effortlessly, even with relatively large bounding boxes. Therefore deep learning models are less sensitive to background pollution than classical feature-based machine learning methods.

With the insights gained by answering the sub-questions we can answer our main research question. The sub-questions have proven that the sensitivity of classification to background information is model and feature dependent. The deep learning model outperforms classical feature-based models and shows little sensitivity to background type or size. The classical feature-based models show sensitivity especially for larger bounding boxes, where for some features the sensitivity correlates with bounding box size. Since shape features appear to be of little significance compared to colour and texture features, and texture features perform very well for a precisely fitting bounding box, the precise segmentation steps appear to be redundant.

4.2 Limitations & Future Work

When assessing the results we can conclude that the extracted features and selected models were highly effective. However, we should also consider the difficulty of the classification problem. As discussed in section 2.1 two of the pollen families are very distant and after preprocessing they are visually easily separable which is mostly responsible for the high prediction scores. The shapes of the pollen are also very similar, with the *Cupressaceae* species having an almost perfectly circular form, and the *Betula* species deviating little from that. This also addresses the importance of shape features which would probably be of more importance if the shapes were more interesting. In addition, in the preprocessing steps the pollen have been automatically segmented, but manually selected. In these steps the decisions may subconsciously have already inserted properties into the dataset that made classification easier.

The segmentation process did gain a fair amount of training samples, but since there is no ground truth we cannot say how effective it was in terms of quantity. The process was still highly manually intensive since the parameters were empirically found by iteratively running the process and manually assessing the results. Therefore, finding that bounding boxes give a fine score is valuable information that could help future researchers motivate why they omit precise segmentation steps from their pipeline.

In our study we used stained pollen grains that are easily separable as RGB images. In most cases, the predictions based on RGB images outperform those using greyscale images. If the pollen grains would not be stained, the results could be different. However, the results we gained suggest that staining the pollen grains and using the higher dimensional RGB images do give the highest possible score.

The data augmentation steps that we used in our study, while helpful for balancing the trainingset, was likely unnecessary due to the simplicity of the classification problem. In addition, due to the uniformity of the data within each class and the simple augmentation methods, using the augmented data is more likely to harm the model's performance than improve it. Using generative methods like GAN with added noise would have created a more diverse dataset.

The contrast computations, to describe the relationship between the object and the background, were performed on groups of images and their corresponding accuracy. A better approach would have been to perform the correlation calculations on individual images and their corresponding confidence score. This would minimise the possible confounding effects introduced when data are grouped. Finally, we used three background types where one was completely white, and two were selected in such a way that there would always be a given amount of noise in the background. It would have been better to train and test on combinations of clean and dirty backgrounds to prevent the model from seeing the background noise as an added constant during training. Future research should therefore explore more complex classification problems with more realistic data to better generalise the findings. Our CutMix method of pasting objects on background created clear edges, despite our efforts to produce an artificial halo. The HOG feature was found least affected by background and benefits from these sharp edges, especially since the objects in the background do not have them. To gain more robust results, the study should be repeated with natural data with more diverse backgrounds to see if HOG retains its stability. In addition, the manual part of the process should be reduced to a minimum to avoid bias. To test the limits of the theory, an exaggeration of the classification problem could also be attempted, such as classifying mixed objects on abstract paintings, to better understand the boundaries of both classical- and deep-learning models.

4.3 Conclusion

To summarise this thesis, we have shown that the influence of backgrounds varies with the size of the bounding boxes and the contrast between the object and the background. Moreover, background sensitivity depends on the selected features in classical machine learning. In contrast, deep learning models demonstrate almost no sensitivity to background noise.

However, the approach used in this thesis has certain limitations, most notably the relative simplicity of the classification problem and the preprocessing methodology. The effects of these factors on the results remain to be explored in future research in order to better generalise the findings and make them applicable to a wider range of classification problems.

The results of this study suggest that using bounding boxes combined with deep learning methods is not only sufficient, but also outperforms precise segmentation methods in classical machine learning. Therefore, time-consuming preprocessing steps involving segmentation can be avoided in future research.

5 Acknowledgments

I would like to express my gratitude to my supervisors, Lu Cao and Fons Verbeek, for their guidance and feedback throughout the creation of this thesis. I also thank Nemi Dorst (Naturalis) and Martijn van de Velde (HHS) for the sample preparation and data acquisition. Their contributions have been essential for the completion of this thesis.

References

[ABL$^+$18] Maureen Agnew, Ivana Banic, Iain R Lake, Clare Goodess, Carlota M Grossi, Natalia R Jones, Davor Plavec, Michelle Epstein, and Mirjana Turkalj. Modifiable risk factors for common ragweed (ambrosia artemisiifolia) allergy and disease in children: a case-control study. *International journal of environmental research and public health*, 15(7):1339, 2018.

[AS23] Mika Akesaka and Hitoshi Shigeoka. “invisible killer”: Seasonal allergies and accidents. Technical report, National Bureau of Economic Research, 2023.

[BBE$^+$23] Karl-Christian Bergmann, Randolf Brehler, Christina Endler, Conny Höflich, Sabine Kespohl, Maria Plaza, Monika Raulf, Marie Standl, Roma Thamm, Claudia Traidl-Hoffmann, et al. Impact of climate change on allergic diseases in germany. *Journal of Health Monitoring*, 8(Suppl 4):76, 2023.

[BR07] Derek Bradley and Gerhard Roth. Adaptive thresholding using the integral image. *Journal of Graphics Tools*, 12(2):13–21, 2007.

[BY77] Jaok E Bowie and IT Young. An analysis technique for biological shape-ii. *Acta Cytol*, 21(3):455–464, 1977.

[Col07] Tony J Collins. Imagej for microscopy. *Biotechniques*, 43(sup1):S25–S30, 2007.

[dCNMO$^+$20] Gennaro d’Amato, Herberto Jose Chong-Neto, Olga Patricia Monge Ortega, Carolina Vitale, Ignacio Ansotegui, Nelson Rosario, Tari Haahtela, Carmen Galan, Ruby Pawankar, Margarita Murrieta-Aguttes, et al. The effects of climate change on respiratory allergy and asthma induced by pollen and mold allergens. *Allergy*, 75(9):2219–2228, 2020.

[DIP23] DIPlib Developers. Diplib. https://github.com/DIPlib/diplib, 2023. Computer software.

[DMB$^+$24] Divya Dwarakanath, Andelija Milic, Paul J Beggs, Darren Wraith, and Janet M Davies. A global survey addressing sustainability of pollen monitoring. *World Allergy Organization Journal*, 17(12):100997, 2024.

[DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, volume 1, pages 886–893. Ieee, 2005.

[DVS$^+$17] Gennaro D’Amato, Carolina Vitale, Alessandro Sanduzzi, Antonio Molino, Alessandro Vatrella, and Maria D’Amato. Allergenic pollen and pollen allergy in europe. *Allergy and allergen immunotherapy*, pages 287–306, 2017.

[GAS$^+$24] Benjamin Garga, Hamadjam Abboubakar, Rodrigue Saoungoumi Sourpele, David Libouga Li Gwet, and Laurent Bitjoka. Pollen grain classification using some convolutional neural network architectures. *Journal of Imaging*, 10(7):158, 2024.

[Gon09] Rafael C Gonzalez. *Digital image processing*. Pearson education india, 2009.

[GRM$^+$18] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International conference on learning representations*, 2018.

[HSD07] Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 2007.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Ibr22] Mostafa Ibrahim. The annotated resnet-50, Nov 2022. Accessed: 2025-08-30.

[KHB$^+$24] Nadja Kabisch, Thomas Hornick, Jan Bumberger, Roland Krämer, Rupert Legg, Oskar Masztalerz, Maximilian Bastl, Jan C Simon, Regina Treudler, and Susanne Dunker. Monitoring and perception of allergenic pollen in urban park environments. *Landscape and urban planning*, 250:105133, 2024.

[KHEK76] M. Kuwahara, K. Hachimura, S. Eiho, and M. Kinoshita. Processing of riangiocardiographic images. In K. Jr. Preston and M. Onoe, editors, *Digital Processing of Biomedical Images*, pages 187–202. Plenum Press, New York, 1976.

[KKKPWS21] Elżbieta Kubera, Agnieszka Kubik-Komar, Krystyna Piotrowska-Weryszko, and Magdalena Skrzypiec. Deep learning methods for improving pollen monitoring. *Sensors*, 21(10):3526, 2021.

[KYLW21] KC Kamal, Zhendong Yin, Dasen Li, and Zhilu Wu. Impacts of background removal on convolutional neural networks for plant disease classification insitu. *Agriculture*, 11(9):1–16, 2021.

[LPC$^+$23] Chen Li, Marcel Polling, Lu Cao, Barbara Gravendeel, and Fons J Verbeek. Analysis of automatic image classification methods for urticaceae pollen classification. *Neurocomputing*, 522:181–193, 2023.

[MPBF22] Mazda Moayeri, Phillip Pope, Yogesh Balaji, and Soheil Feizi. A comprehensive study of image classification model sensitivity to foregrounds, backgrounds, and visual attributes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19087–19097, 2022.

[OPM02] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

[Paw14] Ruby Pawankar. Allergic diseases and asthma: a global public health concern and a call to action. *World Allergy Organization Journal*, 7(1):1–3, 2014.

[PBB$^+$21] Maxim Privalov, Nils Beisemann, Jan El Barbari, Eric Mandelka, Michael Müller, Hannah Syrek, Paul Alfred Grützner, and Sven Yves Vetter. Software-based method for automated segmentation and measurement of wounds on photographs using mask r-cnn: a validation study. *Journal of Digital Imaging*, 34(4):788–797, 2021.

[PTD25] Nhan Pham-Thi and Pascal Demoly. Pollen allergy and climate change: perceptions by physicians and patients. *Exploration of Asthma & Allergy*, 3:100986, 2025.

[PTL$^+$22] Freerk Prenzel, Regina Treudler, Tobias Lipek, Maike Vom Hove, Paula Kage, Simone Kuhs, Thorsten Kaiser, Maximilian Bastl, Jan Bumberger, Jon Genuneit, et al. Invasive growth of ailanthus altissima trees is associated with a high rate of sensitization in atopic patients. *Journal of Asthma and Allergy*, pages 1217–1226, 2022.

[SB91] Michael J Swain and Dana H Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[SCD+17] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[Sch16] Charles W Schmidt. Pollen overload: seasonal allergies in a changing climate, 2016.

[Sha48] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[SWD05] Gaurav Sharma, Wencheng Wu, and Edul N. Dalal. The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application*, 30(1):21–30, 2005.

[SZA+20] Eric Sauvageat, Yanick Zeder, Kevin Auderset, Bertrand Calpini, Bernard Clot, Benoît Crouzy, Thomas Konzelmann, Gian Lieberherr, Fiona Tummon, and Konstantina Vasilatou. Real-time pollen monitoring using digital holography. *Atmospheric Measurement Techniques*, 13(3):1539–1550, 2020.

[TGB+17] M Thibaudon, C Galán, M Bonini, S Röseler, and D Fernándezgonzález. Ambient air-sampling and analysis of airborne pollen grains and fungal spores for networks related to allergy-volumetric hirst method (cen/ts 16868: 2015). *Aerobiologia*, 33(4):581–592, 2017.

[Tom12] Carlo Tomasi. Histograms of oriented gradients. *Computer Vision Sampler*, 1:1–6, 2012.

[VdWSNI+14] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014.

[VVLK14] EFPM Vuurman, LL Vuurman, I Lutgens, and B Kremer. Allergic rhinitis is a risk factor for traffic safety. *Allergy*, 69(7):906–912, 2014.

[WCS+19] Yuchen Wu, Zhi Chen, Dounan Sun, Lichang Zhao, Chuan Zhou, and Wenjing Yue. Human ear recognition using hog with pca dimension reduction and lbp. In *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pages 72–75. IEEE, 2019.

[Wis05] Clark Wissler. The spearman correlation formula. *Science*, 22(558):309–311, 1905.

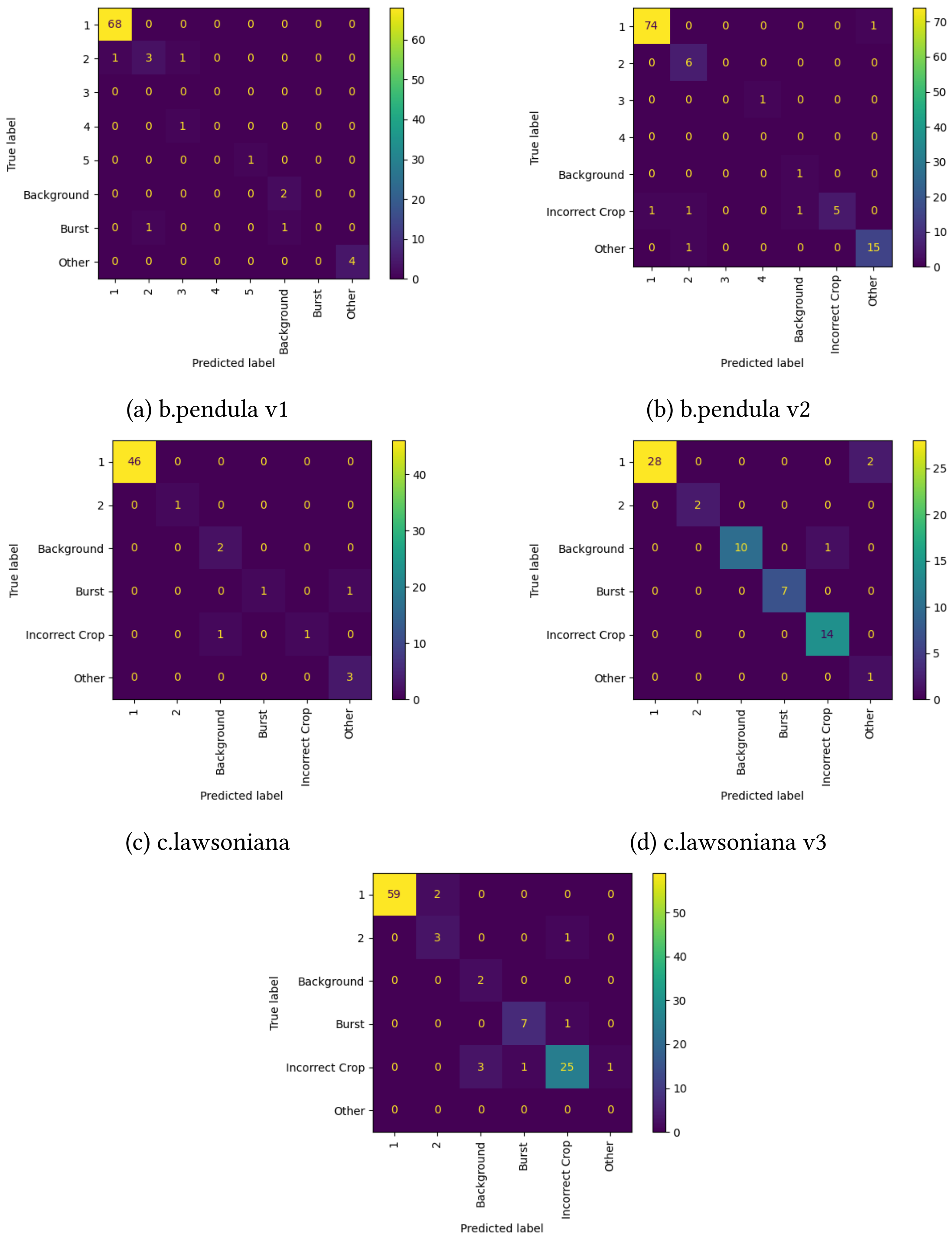
[XEIM20] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*, 2020.

[YHO+19] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[YWB74] Ian T Young, Joseph E Walker, and Jack E Bowie. An analysis technique for biological shape. i. *Information and control*, 25(4):357–370, 1974.

Appendices

A Automated Data Filtering with SVM Confusionmatrices



(a) b.pendula v1

(b) b.pendula v2

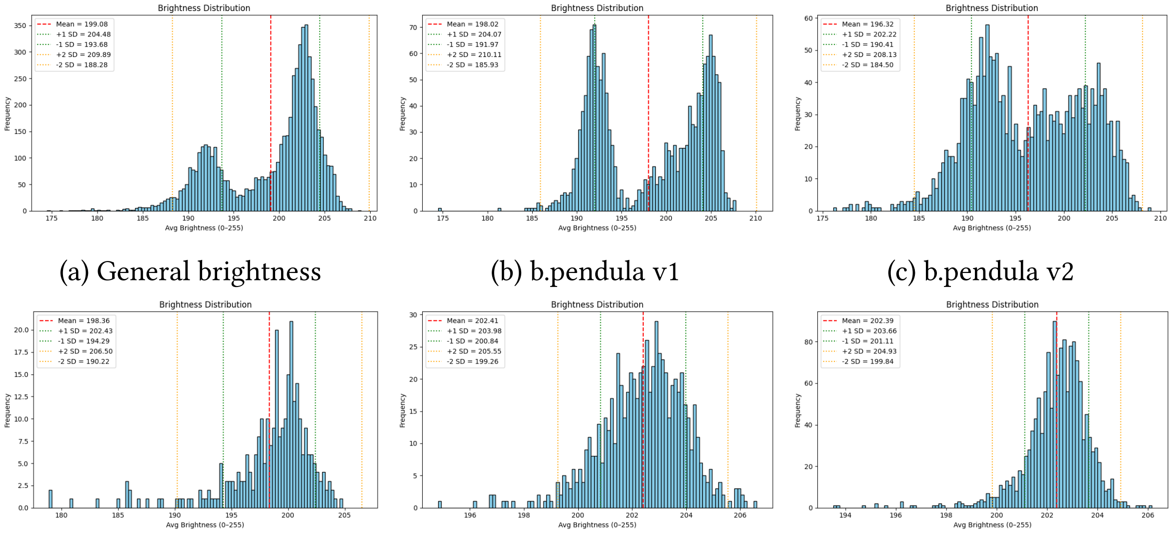
(c) c.lawsoniana

(d) c.lawsoniana v3

(e) c.nootkatensis

Figure 28: Confusion matrices for different pollen datasets.

B Brightness, Hue, Size Distributions per Slide



(a) General brightness

(b) b.pendula v1

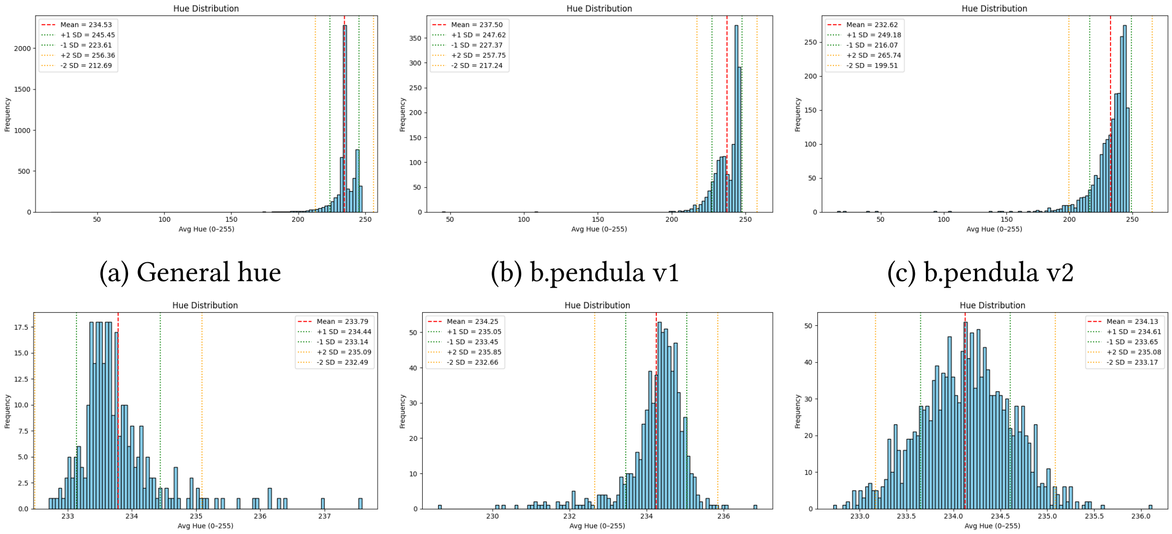
(c) b.pendula v2

(d) c.lawsoniana

(e) c.lawsoniana v3

(f) c.nootkatensis

Figure 29: Brightness distributions for different pollen datasets.



(a) General hue

(b) b.pendula v1

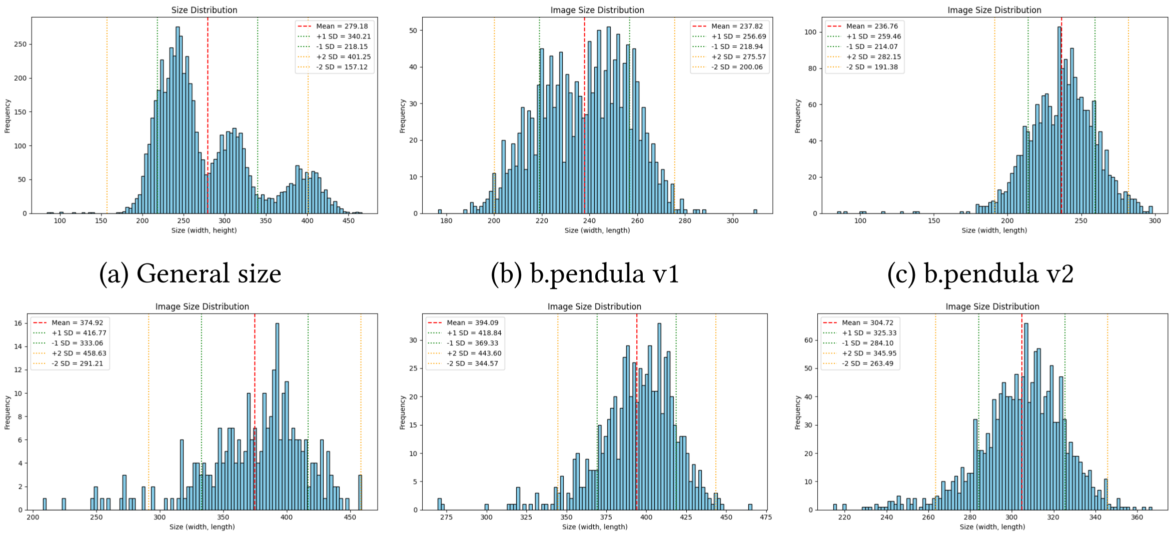
(c) b.pendula v2

(d) c.lawsoniana

(e) c.lawsoniana v3

(f) c.nootkatensis

Figure 30: Hue distributions for different pollen datasets.



(a) General size

(b) b.pendula v1

(c) b.pendula v2

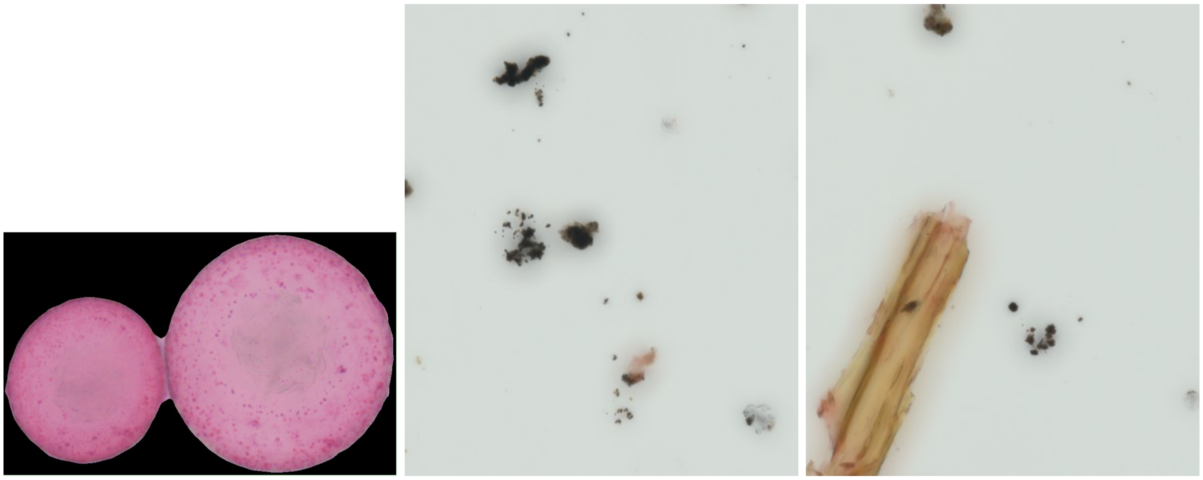
(d) c.lawsoniana

(e) c.lawsoniana v3

(f) c.nootkatensis

Figure 31: Size distributions for different pollen datasets.

C Image Examples



(a) Example scale variance within the same species

(b) Sniffer tile1

(c) Sniffer tile 2

Figure 32: Examples of data

D GLCM Feature Definitions

Given a normalized GLCM $P(i, j)$ of size $L \times L$, the standard texture features are defined as follows:

$$\text{Contrast} = \sum_{i,j} (i - j)^2 \, P(i, j) \tag{18}$$

$$\text{Correlation} = \frac{\sum_{i,j} (i - \mu_i)(j - \mu_j) P(i, j)}{\sigma_i \sigma_j} \tag{19}$$

$$\text{Energy} = \sum_{i,j} P(i, j)^2 \tag{20}$$

$$\text{Homogeneity} = \sum_{i,j} \frac{P(i, j)}{1 + |i - j|} \tag{21}$$

$$\text{Dissimilarity} = \sum_{i,j} |i - j| \, P(i, j) \tag{22}$$

where

$$\mu_i = \sum_i i \sum_j P(i, j), \qquad \mu_j = \sum_j j \sum_i P(i, j),$$

$$\sigma_i = \sqrt{\sum_i (i - \mu_i)^2 \sum_j P(i, j)}, \qquad \sigma_j = \sqrt{\sum_j (j - \mu_j)^2 \sum_i P(i, j)}.$$

E Prediction Performance

Table 12: Classification performance of SVM for different backgrounds (size = 1.2). F1-score, precision, and recall are macro-averaged across 3 classes.

Background	Colour	Accuracy	Precision	Recall	F1 Score
white	RGB	1.2	0.967 ± 0.000	0.967 ± 0.000	0.967 ± 0.000
	Gray	1.2	0.973 ± 0.000	0.973 ± 0.000	0.973 ± 0.000
sniffer	RGB	1.2	0.968 ± 0.008	0.968 ± 0.008	0.968 ± 0.008
	Gray	1.2	0.971 ± 0.007	0.970 ± 0.007	0.970 ± 0.007
combination	RGB	1.2	0.965 ± 0.010	0.965 ± 0.010	0.965 ± 0.010
	Gray	1.2	0.964 ± 0.008	0.964 ± 0.008	0.964 ± 0.008

Table 13: Classification performance of SVM for different backgrounds (size = 1.3). F1-score, precision, and recall are macro-averaged across 3 classes.

Background	Colour	Accuracy	Precision	Recall	F1 Score
white	RGB	1.3	0.977 ± 0.000	0.977 ± 0.000	0.977 ± 0.000
	Gray	1.3	0.980 ± 0.000	0.980 ± 0.000	0.980 ± 0.000
sniffer	Gray	1.3	0.966 ± 0.007	0.966 ± 0.007	0.966 ± 0.007
	RGB	1.3	0.970 ± 0.008	0.970 ± 0.008	0.970 ± 0.008
combination	RGB	1.3	0.958 ± 0.010	0.958 ± 0.011	0.958 ± 0.011
	Gray	1.3	0.956 ± 0.013	0.956 ± 0.013	0.956 ± 0.013

Table 14: Classification performance of SVM for different backgrounds (size = 1.4). F1-score, precision, and recall are macro-averaged across 3 classes.

Background	Colour	Accuracy	Precision	Recall	F1 Score
white	RGB	1.4	0.973 ± 0.000	0.973 ± 0.000	0.973 ± 0.000
	Gray	1.4	0.977 ± 0.000	0.977 ± 0.000	0.977 ± 0.000
sniffer	RGB	1.4	0.965 ± 0.006	0.965 ± 0.006	0.965 ± 0.006
	Gray	1.4	0.964 ± 0.008	0.964 ± 0.008	0.964 ± 0.008
combination	RGB	1.4	0.958 ± 0.008	0.958 ± 0.008	0.958 ± 0.008
	Gray	1.4	0.956 ± 0.010	0.956 ± 0.010	0.956 ± 0.010