

IMPROVING POLLEN SPECIES CLASSIFICATION USING DEEP LEARNING AND SIZE FEATURES

NANCY TRUONG

Bachelor's thesis
2026:K25



LUND UNIVERSITY

Faculty of Science
Centre for Mathematical Sciences
Mathematical Statistics

Contents

Abstract	3
1 Introduction	4
1.1 Research Question	5
1.2 Use of generative AI	5
2 Data	7
2.1 Dataset, Variables, and Split Design	7
2.2 Preprocessing and Missing Size Measurements	8
3 Theory and Method	12
3.1 Problem formulation and notation	12
3.2 Data handling and feature construction	12
3.2.1 Incorporating a ratio-based shape descriptor	13
3.2.2 Data imputation	13
3.2.3 Data augmentation	14
3.3 Model architectures, loss functions, and fitting	17
3.3.1 CNN-based multimodal model	17
3.3.2 ResNet18 baseline model	18
3.3.3 256-dimensional image representation	18
3.3.4 ConvNeXt-Tiny architecture	21
3.3.5 Trainable parameter counts	23
3.3.6 Loss functions	23
3.3.7 Model fitting	24
3.4 Evaluation	25
3.4.1 Evaluation metrics	25
3.4.2 Class difficulty and feature-space structure	26

3.4.3	Sensitivity analysis with alternative test flowers	27
4	Analysis	29
4.1	Baseline 512-dimensional ResNet18 models	31
4.2	ResNet18 models with reduced 256-dimensional representation	35
4.3	ConvNeXt-Tiny models	38
4.4	Sensitivity analysis	40
5	Discussion of results	42
5.1	Overall accuracy and recall	42
5.2	Effect of augmentation	48
5.3	Effect of imputation and size-related information	49
5.4	Effect of focal loss	51
5.5	Persistent class confusions	52
5.6	Remaining classification errors in the best ConvNeXt-Tiny model	53
5.7	Robustness and sensitivity to the flower-level split	60
6	Conclusion	64
	Code and data availability	66
A	Flower IDs Used in Train Dataset	69
B	Sensitivity-analysis	71
B.1	Flower-level test cases	71
B.2	Confusion Matrices	71

Abstract

Automated pollen classification is challenging because visually similar species may differ only by subtle morphological characteristics. This thesis investigates deep-learning models for classifying microscopic pollen-grain images from ten plant species using RGB image crops together with two segmentation-derived size measurements: the major and minor axes of the pollen grain. The study builds on work by Jurdell (2025) and uses the same ResNet18-based image-and-size model in Python/PyTorch to establish a comparable baseline.

This thesis evaluates the effects of missing-size handling, validation-based model selection, class-balancing augmentation, image-feature representation, focal loss, and backbone architecture. The reproduced baseline, denoted R512-Base, achieved 76.8% test accuracy and 76.2% macro recall. Species-wise mean imputation of missing size features and the change in image representation improved performance. The best overall model replaced Resnet with ConvNeXt-Tiny, achieved 85.2% test accuracy and 85.3% macro recall.

The improvement was not uniform across species. The most persistent difficulty was the separation of *Crepis capillaris* and *Hypochaeris radicata*. In the baseline model, *Crepis capillaris* had zero recall; the final ConvNeXt-Tiny model substantially increased the recall of *Crepis capillaris* to 38.5%. Thus, the main limitation was not poor performance across all species, but concentrated uncertainty around a small number of visually similar and flower-sensitive classes.

A sensitivity analysis using 16 alternative flower-level splits for *Crepis capillaris* and *Hypochaeris radicata* showed that the final model remained strong on average, with mean accuracy 84.2% and mean macro recall 85.2%. However, the recalls of these two species varied substantially depending on which flowers were held out for testing, whereas the remaining species were comparatively stable. These findings support the use of flower-level evaluation for pollen classifiers and show that aggregate accuracy should be reported together with class-wise recall and error-structure analysis.

1 Introduction

Understanding which plant species pollinating bumble bees/insects have visited is essential for studying plant-pollinator interactions and ecosystem functions. Artificial hives and pollen traps make it possible to collect pollen grains transported by insects, but manual microscopic identification is labor-intensive, time-consuming, and requires expert knowledge (Beil, Horn, and Schwabe 2008). Automated pollen classification therefore represents an important methodological advancement for ecological research (Olsson et al. 2021).

Recent work has shown that deep learning, and especially convolutional neural networks (CNNs), can perform well in image-based pollen classification (LeCun, Bengio, and Hinton 2015; Olsson et al. 2021). CNNs are well suited to this task because they can learn hierarchical features from microscopic images (LeCun, Bengio, and Hinton 2015). In particular, earlier work by Olsson et al. (2021) demonstrated that deep learning can provide efficient, automated, and robust pollen analysis, establishing a good baseline for pollen classification. Jurdell (2025) extends that line of research; he investigated if classification performance could be improved by combining images with pollen grain size as an additional tabular input. A relevant motivation for including size is that visual recognition models are often designed to be relatively insensitive to scale, whereas pollen size can itself contain useful discriminatory information between species (Noord and Postma 2017). In the dataset used here, the original high-resolution microscopic images were preprocessed through cropping and rescaling so that each grain was represented within a common 224×224 pixel frame. This means that the original size information was not directly available in the image data, true differences in both size and aspect ratio may become less visually apparent after preprocessing. While this standardization is useful for image-based modelling, it also reduces the extent to which the apparent size and shape of the grain in the image reflects its true biological morphology. By explicitly including size as a separate input, the model can learn to use this information in combination with visual features, which may be especially helpful for distinguishing species that are visually similar but differ in size. In pollen analysis, this makes size a biologically meaningful predictor rather than a nuisance variable. Using a modified ResNet18 architecture trained on 10 pollen species, he reported that adding size information increased overall accuracy from 73.6% to 77.4%, while also improving recall for visually similar species (Jurdell 2025).

The present study uses the same dataset, species selection, and flower-level train-test split as Jurdell (2025). This makes the studies directly comparable and allows the effect of

methodological changes to be evaluated without confounding changes in the data.

Although Jurdell’s results showed that size-aware classification can improve performance, important questions remain. Classification performance still differs substantially across species: some classes achieve very high recall, whereas others remain difficult to classify and are repeatedly confused with a small number of other species. Since the class frequencies in the training data are reasonably similar, these errors are more naturally interpreted as reflecting intrinsic class difficulty, such as visual similarity between species or high within-class variability, rather than severe class imbalance. In addition, some training images lack size measurements. Whereas Jurdell (2025) excluded these images, the present study examines whether retaining them through imputation, including species-wise mean imputation and bootstrap-based imputation, can make better use of the available data and improve classification performance.

1.1 Research Question

Against this background, the aim of this thesis is not to redefine the experimental setting, but to investigate whether classification performance can be further improved within the same setting. To do so, Jurdell’s MATLAB-based model is first reimplemented in Python to establish a comparable baseline. The study then evaluates a series of methodological extensions, including imputation of missing size values, an explicit validation split, image augmentation, alternative loss formulations, a reduced ResNet18 image representation, and comparison with the more modern ConvNeXt-Tiny architecture (Liu et al. 2022). In addition, particular attention is given to species with low recall in order to better understand class difficulty and the types of confusion that persist across models.

The main research questions are:

- To what extent can pollen-species classification performance be improved through alternative training strategies and architectural modifications?
- Which methodological choices are most effective for improving performance on intrinsically difficult species?
- Can we understand why certain species are difficult to classify?

1.2 Use of generative AI

Generative AI tools were used as support during the thesis process, primarily for language editing, restructuring draft text, and improving clarity of explanations. They were also

used to discuss possible ways of formulating methodological descriptions and to support debugging and organization of code. The statistical analysis, model implementation, experimental design, interpretation of results, and final academic decisions were made by the author. All generated or AI-assisted text was reviewed, edited, and verified by the author before inclusion in the thesis. No generative AI tool was used to produce, alter, or fabricate the empirical results reported in this thesis.

2 Data

This chapter describes the dataset, the flower-level train-test split, the available size measurements, and the handling of missing size values. The aim is to make the data structure and preprocessing choices clear before the model specifications are introduced in the following chapter.

2.1 Dataset, Variables, and Split Design

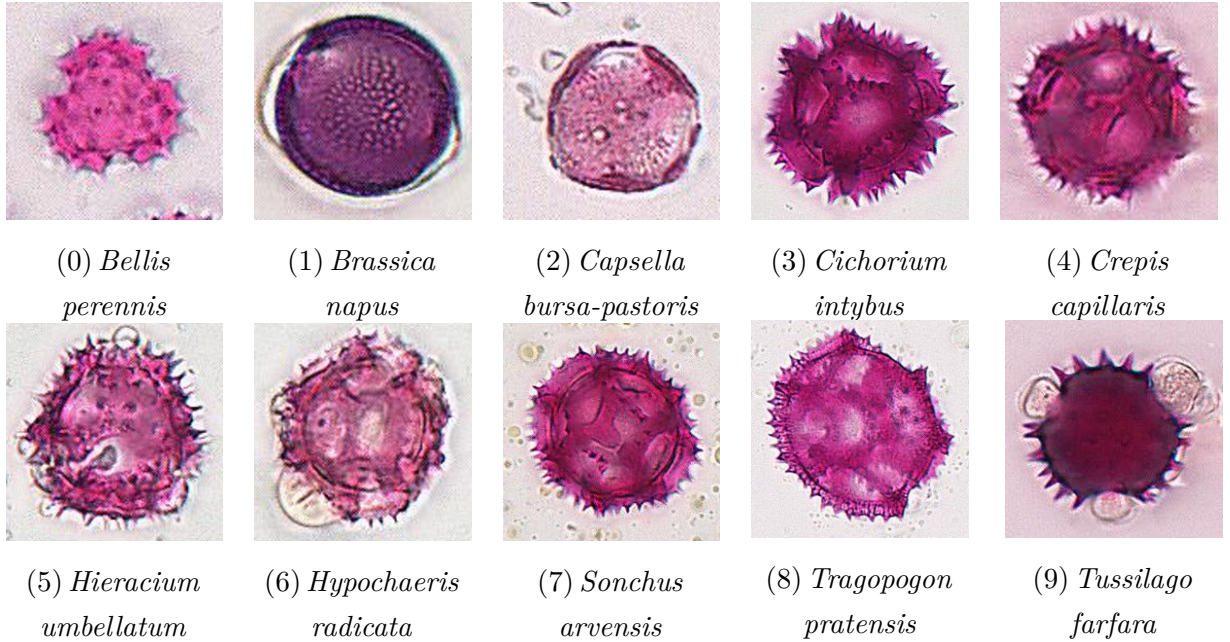


Figure 2.1: Representative pollen grain images from the ten species included in the dataset, one example per species. Each image is a 224×224 RGB microscopy crop.

The present study uses the same dataset, species selection, and flower-level train-test split as Jurdell (2025). The dataset consists of 10,000 microscopic RGB pollen-grain images from 10 plant species, each cropped to 224×224 pixels. The split contains 8078 training images and 1922 test images. Each image is associated with two numerical size measurements: the major and minor axes of the pollen grain. The major and minor axes are segmentation-derived measurements obtained from the image segmentation pipeline. More specifically, they correspond to the `regionprops` output in MATLAB, where the major and minor axis lengths describe the principal axes of an ellipse fitted to the segmented pollen-grain object (Olsson et al. 2021). The major axis is the longer principal axis and the minor axis is the shorter principal axis. Representative examples are shown

in Figure 2.1.

The pollen-grain images were prepared as described in Olsson et al. (2021). Following Jurdell (2025), the data were split at the flower level so that no flower contributed images to both training and test sets. More specifically, one flower per species was reserved exclusively for the test set, except for *Brassica napus*, for which two flowers were used because a single flower did not provide enough pollen-grain images. This design is motivated by the fact that several images come from the same flower. Pollen grains from the same flower may share biological, preparation-related, or imaging-related characteristics, and may therefore be more similar to each other than to grains from another flower of the same species. If images from the same flower appeared in both training and test sets, the model could partly learn flower-specific patterns, leading to overly optimistic performance estimates. This concern is supported by Olsson et al. (2021), who found much higher accuracy when pollen grains were pooled and randomly split than when the model was evaluated on pollen-grain images of held-out flowers. This does not assume that pollen grains from different flowers are always visually different. It treats possible flower-to-flower variation as part of the generalization problem. The flower-level split evaluates whether the classifier generalizes to unseen flowers, which is closer to the intended ecological use case.

The resulting split is summarized in Table 2.1. Since the split is defined by flower identity, each observation can be traced back to a specific flower ID. For transparency, the table reports the test flower ID or IDs for each species, while the complete lists of training and test flower IDs are provided in Appendix A; these identifiers are used again in the sensitivity analysis, where alternative test flowers are considered for selected species.

After complete-case filtering, the training image counts ranged from 666 to 870 observations across species; see Table 2.1. The test set was less even, as expected under a flower-level split, because each test class consists of all images from one or two reserved flowers. The training imbalance was therefore moderate, and the analysis focuses primarily on species-specific difficulty and error structure rather than on severe class imbalance.

2.2 Preprocessing and Missing Size Measurements

All images were transformed into PyTorch tensors with the three RGB colour channels stacked along the channel dimension and normalized using the standard ImageNet channel means and standard deviations before being passed to the pretrained image models (Paszke et al. 2019; Deng et al. 2009). In addition to the image input, each observation was associated with two size features: the major axis and the minor axis of the pollen grain. Figure 2.2 shows these measurements by species for the training and test data.

Table 2.1: Numbers of unique flowers, test flower identities, and image counts for each species.

Species	Unique flowers			Images			
	Train	Test	Test flower ID(s)	Complete-case	Imputed	Change	Test
<i>Bellis perennis</i>	6	1	1011316	856	858	2	142
<i>Brassica napus</i>	8	2	1005374, 1005376	795	800	5	200
<i>Capsella bursa-pastoris</i>	5	1	1005390	817	817	0	183
<i>Cichorium intybus</i>	8	1	1005458	870	870	0	130
<i>Crepis capillaris</i>	4	1	1010538	799	800	1	200
<i>Hieracium umbellatum</i>	4	1	1012176	735	800	65	200
<i>Hypochaeris radicata</i>	4	1	1012069	717	800	83	200
<i>Sonchus arvensis</i>	2	1	1011968	666	666	0	334
<i>Tragopogon pratensis</i>	3	1	1017595	834	834	0	166
<i>Tussilago farfara</i>	5	1	1005388	833	833	0	167
Total	49	11	–	7922	8078	156	1922

Note. Under *Unique flowers*, “Train” and “Test” denote the number of distinct flower individuals in each split, following the protocol of Jurdell (2025). “Test flower ID(s)” identifies the individual(s) assigned to the test set.

Under *Images*, “Complete-case” reports the number of training images after excluding observations with missing size measurements, “Imputed” gives the training-set size after imputation, and “Change” denotes the increase relative to the complete-case training set. “Test” reports the number of test images.

For most species, the training and test observations lie in broadly the same region of the major-minor axis plane. In some species, however, one split is more dispersed than the other. In addition, a few species show observations lying clearly away from the main cluster, most noticeably *Bellis perennis* and *Sonchus arvensis*, and to some extent also *Tragopogon pratensis* and *Hypochaeris radicata*.

A subset of the training images, 156 observations, lacks recorded size measurements. In Jurdell (2025), such observations were removed before modelling, which naturally motivates a missing-data discussion in the present study, since excluding incomplete cases reduces the amount of available training information and may also change the empirical feature distribution (Jurdell 2025).

Three approaches were considered: (i) complete-case training, where observations with missing size measurements were excluded; (ii) species-wise mean imputation, where missing values were replaced by the corresponding species mean; and (iii) species- and flower-level bootstrap imputation, where missing size vectors were sampled from observed major-minor axis pairs within the same species and, when available, the same flower; see Section 3.2.2. Complete-case training reduced the effective training set from 8078 to 7922 observations, whereas both imputation-based approaches retained all 8078 training images.

As shown in Table 2.1, the restored observations were concentrated mainly in *Hypochaeris radicata* and *Hieracium umbellatum*. The effect on the remaining affected species was

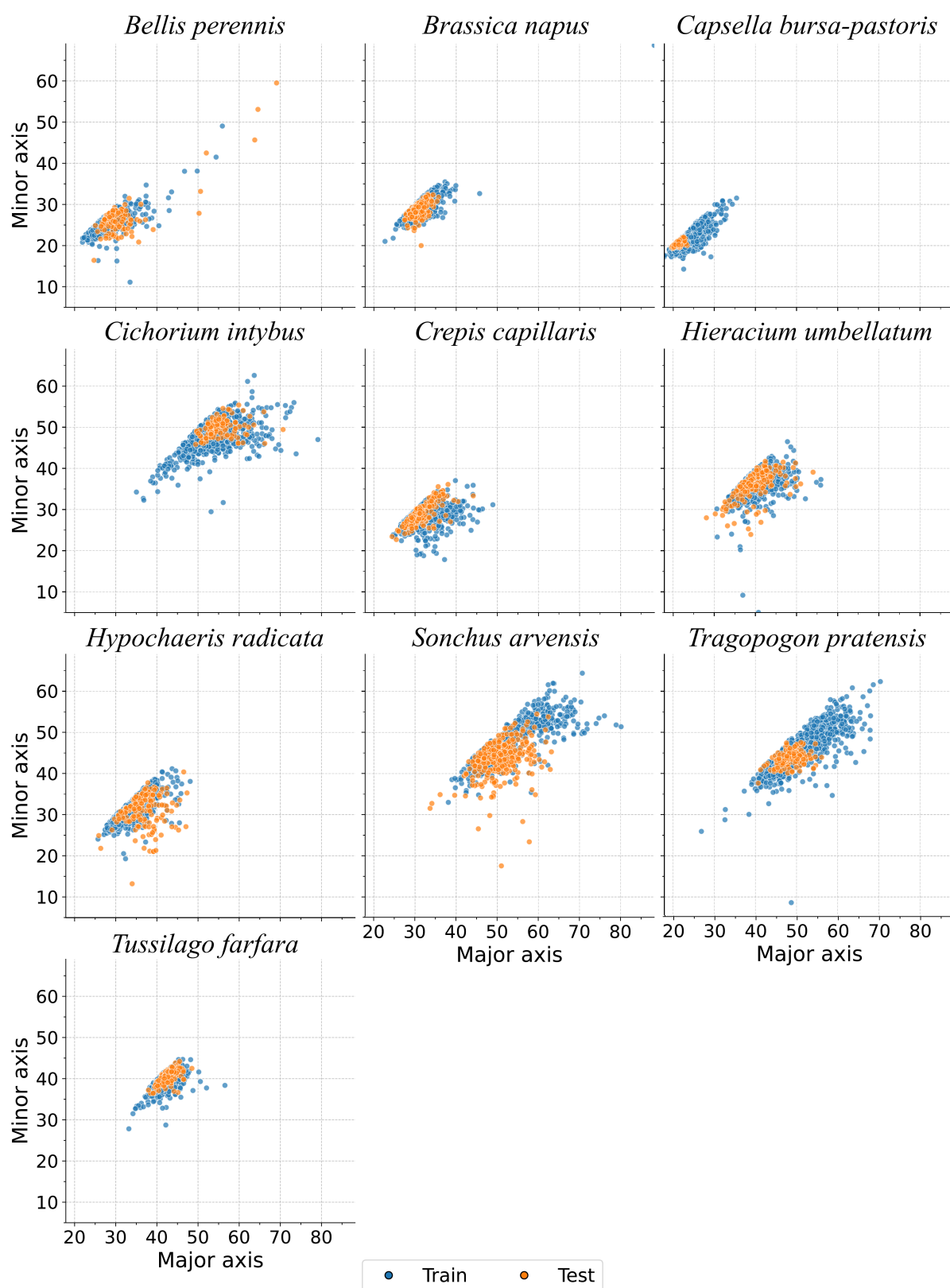


Figure 2.2: Major-axis versus minor-axis measurements by species in the train and test sets. All panels share identical axis scales to enable direct comparison across species.

small. Thus, the impact of imputation was species-specific rather than evenly distributed across the training set.

3 Theory and Method

3.1 Problem formulation and notation

Let $X \in \mathcal{X} \subset \mathbb{R}^{224 \times 224 \times 3}$ denote a microscopic RGB pollen image, let $S \in \mathbb{R}^m$ denote the associated size-feature vector, and let $Y \in \{1, \dots, 10\}$ denote the species label. In the baseline setting, $m = 2$, corresponding to the major and minor axes of the pollen grain. In the ratio-based setting, $m = 3$, where an additional shape descriptor is included; see Section 3.2.1. The task is therefore a supervised multiclass classification problem with mixed input modalities: image data and low-dimensional tabular size features. This formulation follows earlier image-based work on pollen classification (Olsson et al. 2021) and the later extension using size information (Jurdell 2025).

For observation i , write (x_i, s_i, y_i) for the observed image, size-feature vector, and class label. A classifier with parameters θ maps the image and size features to a vector of class probabilities,

$$p_\theta(x_i, s_i) = (p_{\theta,1}(x_i, s_i), \dots, p_{\theta,10}(x_i, s_i)) \in [0, 1]^{10},$$

where

$$\sum_{k=1}^{10} p_{\theta,k}(x_i, s_i) = 1.$$

The predicted class is therefore

$$\hat{y}_i = \operatorname{argmax}_{k \in \{1, \dots, 10\}} p_{\theta,k}(x_i, s_i).$$

The estimation of θ from the training data is described in Section 3.3.

3.2 Data handling and feature construction

For observation i , let M_i and m_i denote the recorded major and minor axes of the pollen grain, respectively. In the baseline setting, the tabular input is therefore

$$s_i = (M_i, m_i) \in \mathbb{R}^2.$$

In some experiments, this input was augmented by a derived ratio feature.

3.2.1 Incorporating a ratio-based shape descriptor

For each observation i with recorded size measurements, the ratio

$$\frac{m_i}{M_i}$$

was computed before any species-level averaging was performed. This ratio provides a simple summary of grain shape: values closer to 1 correspond to more circular grains, whereas smaller values indicate more elongated grains. In the ratio-based setting, the tabular input becomes

$$s_i = \left(M_i, m_i, \frac{m_i}{M_i} \right) \in \mathbb{R}^3.$$

This feature was introduced to examine whether explicit shape information could complement the absolute size variables, particularly since cropping and rescaling may weaken the visual expression of true aspect differences in the image data.

3.2.2 Data imputation

Missing size measurements create a missing-data problem in the predictor space. A complete-case strategy is simple and avoids introducing assumptions about the missing values, but it reduces the amount of available information and may increase estimation variability. Imputation offers an alternative way to retain incomplete observations.

In this thesis, the main imputation strategy was species-wise mean imputation. For species $k \in \{1, \dots, 10\}$, let

$$\bar{s}_k = \begin{cases} (\bar{M}_k, \bar{m}_k), & \text{in the baseline setting,} \\ (\bar{M}_k, \bar{m}_k, \overline{(m/M)}_k), & \text{in the ratio-based setting,} \end{cases}$$

where $\bar{m}_k$ and $\bar{M}_k$ denote the species-specific sample means of the major and minor axes, respectively. In the ratio-based setting, $\overline{(m/M)}_k$ denotes the species-specific mean of the ratios after m_i/M_i has first been computed for each complete training observation. Thus, $\overline{(m/M)}_k$ is not computed as $\bar{m}_k/\bar{M}_k$. For an incomplete observation i with species label $y_i = k$, the missing tabular input was imputed by

$$\tilde{s}_i = \bar{s}_{y_i}.$$

All imputation quantities were estimated from the training data only.

Mean imputation preserves the nominal sample size and allows incomplete observations to remain in the analysis. However, it does not preserve the within-species variability

of the imputed observations and may distort the empirical relationship between size measurements. It may also introduce bias if the missingness mechanism is not close to random. Its role in this thesis is therefore pragmatic: it is evaluated as a trade-off between retaining incomplete observations and distorting the predictor distribution (Pigott 2001).

A possible alternative to species-wise mean imputation is bootstrap-based imputation within the same species and flower. For an observation with missing size measurements, a complete size vector (M_i, m_i) was sampled with replacement from the observed training observations belonging to the same species and the same flower. The major and minor axes were therefore sampled jointly, rather than independently. In the ratio-based experiments, the ratio feature was then computed from the sampled axes.

This procedure was motivated by the flower-level size patterns in Figure 3.1. Within several species, observations from the same flower form local clusters in the major-minor axis plane, suggesting that size measurements may vary systematically between flowers. Sampling within the same species and flower therefore gives imputed values from the local empirical distribution of that flower, rather than shrinking all missing observations toward a species-level mean.

Compared with species-wise mean imputation, this approach retains more within-flower variation and preserves the observed relationship between the major and minor axes more directly. The bootstrap-sampled sizes were therefore used to create an alternative completed version of the training data. By training otherwise comparable models on the mean-imputed and bootstrap-imputed datasets, it was possible to examine whether the results depended on how missing size values were represented. A substantial performance difference would suggest that replacing missing values by species means may hide relevant within-species or flower-to-flower size variation. Similar performance, on the other hand, would indicate that the main conclusions were not strongly driven by the particular imputation strategy. Bootstrap-based imputation was therefore used as a sensitivity analysis, not as a new source of information, but as a way to assess whether a more local, flower-aware treatment of missing size values affected model performance.

3.2.3 Data augmentation

Data augmentation was applied only to the training set and used as a regularization technique to reduce overfitting. The transformations included random horizontal and vertical flips, rotations, colour jitter, and Gaussian blur. These transformations created modified versions of the original pollen-grain images while keeping the species label unchanged. This made it harder for the model to memorize image-specific details, such as exact orientation, staining intensity, focus, or small artefacts, and encouraged it to rely on visual patterns that remained useful under moderate image changes. The transforma-

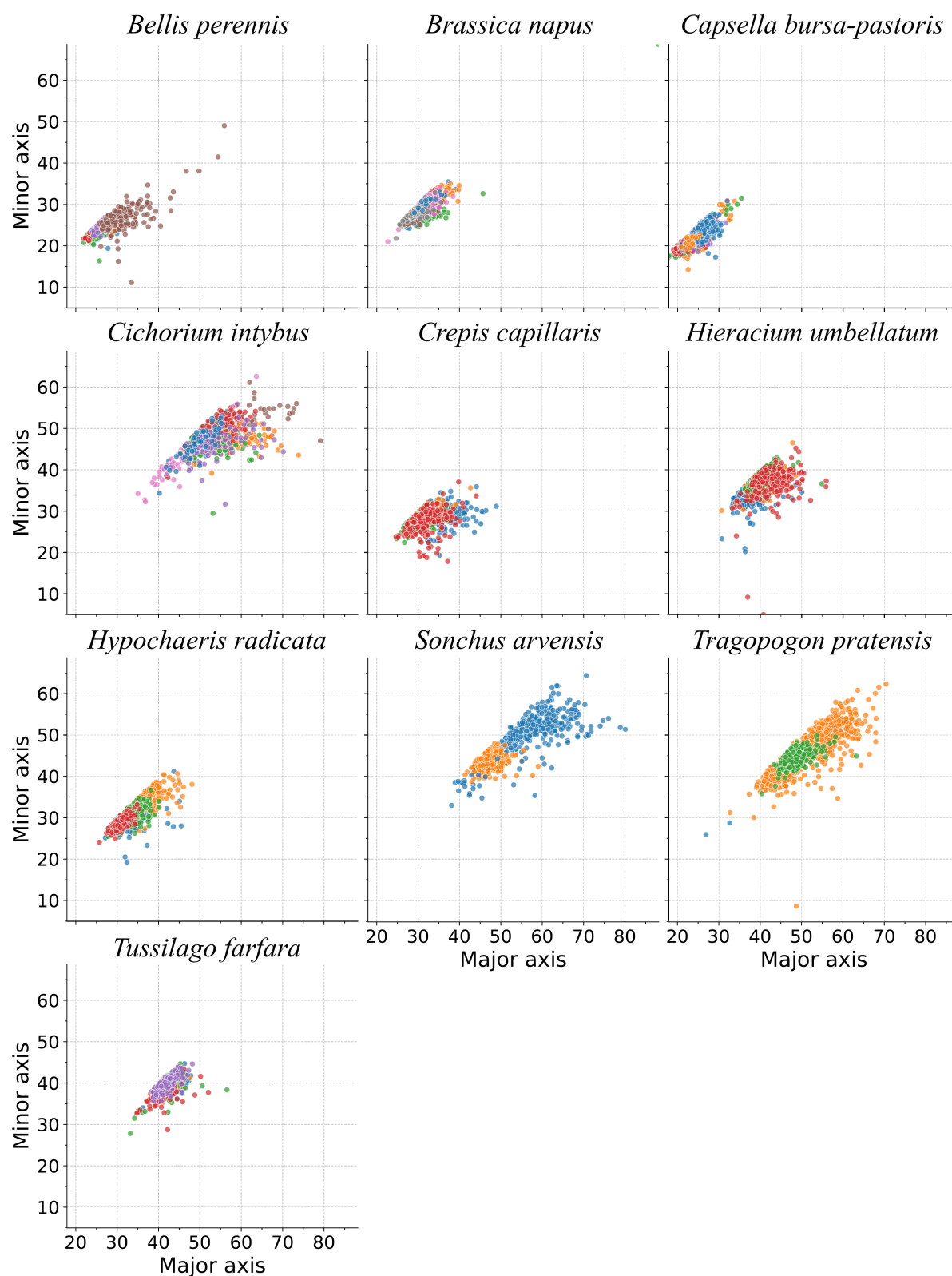


Figure 3.1: Training-set size measurements by species, shown as major axis versus minor axis and coloured by flower. All panels share identical axis scales.

tions were therefore chosen to increase training variability without intentionally changing species-defining morphology (Wong et al. 2016).

Augmentation was used in two different ways. First, transformed versions of existing images were sampled during training without changing the nominal class composition. Second, augmentation was used to expand the training set explicitly by generating additional samples for the smaller classes. Two target sizes were considered for the expansion. In the first scheme, all species were augmented up to the size of the largest class in the complete-case training set, 870 observations. Since one species already had 870 training observations, this scheme augmented the remaining nine species and required 778 generated images in total, corresponding to approximately 9.6% of the imputed training set of 8078 images. In the second scheme, all species were expanded to 1000 observations, requiring 2078 generated images, or approximately 25.7% of the imputed training set. Comparing these two schemes, summarized in Table 3.1, the 1000-observation scheme relies substantially more on synthetic augmentation than the 870-observation balancing scheme.

Table 3.1: Number of augmented images required per species to reach target training-set sizes of 870 and 1000 observations from the complete-case data.

Species	To 870	To 1000
<i>Bellis perennis</i>	14	144
<i>Brassica napus</i>	75	205
<i>Capsella bursa-pastoris</i>	53	183
<i>Cichorium intybus</i>	0	130
<i>Crepis capillaris</i>	71	201
<i>Hieracium umbellatum</i>	135	265
<i>Hypochaeris radicata</i>	153	283
<i>Sonchus arvensis</i>	204	334
<i>Tragopogon pratensis</i>	36	166
<i>Tussilago farfara</i>	37	167
Total	778	2078

Because the original training distribution was already relatively balanced, these augmentation-based expansions are interpreted primarily as robustness studies rather than as corrections for severe class imbalance. The comparison between the 870- and 1000-observation schemes also indicates how sensitive the model is to additional synthetically augmented images.

3.3 Model architectures, loss functions, and fitting

3.3.1 CNN-based multimodal model

CNNs were used in this thesis as nonlinear feature extractors for microscopic pollen images (LeCun, Bengio, and Hinton 2015). A CNN backbone is written as a feature map

$$\phi_{\theta_\phi} : \mathcal{X} \rightarrow \mathbb{R}^d,$$

where θ_ϕ denotes the parameters of the image branch and $\phi_{\theta_\phi}(x_i)$ is the image representation for image x_i . Depending on the model, the image branch consists of the pretrained backbone alone or the pretrained backbone followed by an additional projection layer.

The image representation is fused with the tabular input through concatenation,

$$z_i = (\phi_{\theta_\phi}(x_i), s_i).$$

Thus $z_i \in \mathbb{R}^{d+2}$ in the baseline setting and $z_i \in \mathbb{R}^{d+3}$ in the ratio-based setting.

Let

$$h_{\theta_h} : \mathbb{R}^{\dim(z_i)} \rightarrow \mathbb{R}^{10}$$

denote the classifier head, with parameters θ_h . The model outputs logits

$$\eta_i = h_{\theta_h}(z_i),$$

which are converted to class probabilities by a softmax transformation,

$$p_\theta(x_i, s_i) = \text{softmax}(\eta_i).$$

The full trainable parameter vector is denoted by $\theta = (\theta_\phi, \theta_h)$.

All architecture variants follow the same multimodal principle: a pretrained convolutional backbone extracts an image representation, this representation is concatenated with the size features, and the fused vector is passed to a classifier head. The implementation diagrams below report the feature dimensions, fusion point, and classifier head for each architecture. These details are reported to make the experimental setup more reproducible and to support reuse of the models; see Figures 3.2 to 3.4.

In the diagrams, B denotes the mini-batch size and tensor shapes follow the PyTorch channel-first convention. Thus, although an RGB image is conceptually represented as $224 \times 224 \times 3$, it is passed to the network as a tensor of shape $[B, 3, 224, 224]$. Unless otherwise stated, pretrained backbones were fine-tuned end-to-end together with the

classifier head.

3.3.2 ResNet18 baseline model

The baseline model was a Python reimplementation of the combined image-and-size MATLAB-based model introduced by Jurdell (2025). A pretrained ResNet18 backbone was used to extract image features from the pollen images (He et al. 2016). ResNet is based on residual learning, in which a block with input h produces output

$$h + F(h),$$

where F is a learned nonlinear transformation. These skip connections improve optimization in deep networks by allowing information and gradients to pass more directly through the architecture (He et al. 2016).

In the baseline implementation, the image branch produced a 512-dimensional feature vector,

$$\phi_{\theta_\phi}(x_i) \in \mathbb{R}^{512}.$$

With the two size features, the fused representation had dimension

$$z_i \in \mathbb{R}^{514}.$$

In the ratio-based variants, the additional ratio feature increased the fused dimension to 515. The classifier head was a single linear layer applied to the fused representation. This ResNet18-based multimodal model is outlined in Figure 3.2 and served as the reference point for all subsequent experiments.

3.3.3 256-dimensional image representation

A second ResNet18-based variant projected the image representation from 512 to 256 dimensions before concatenation with the size variables, to reduce the risk of the 512-dimensional embedding dominating the low-dimensional tabular input. Reducing the image representation made it possible to examine if a more compact image branch improved the balance between image and size information in the fused predictor.

The resulting implementation is summarized in Figure 3.3, the only architectural change is the projection from 512 to 256 image features before fusion.

The classifier head remains a single linear classification layer, with 258 inputs when using two size features, and 259 inputs for the ratio-based variants.

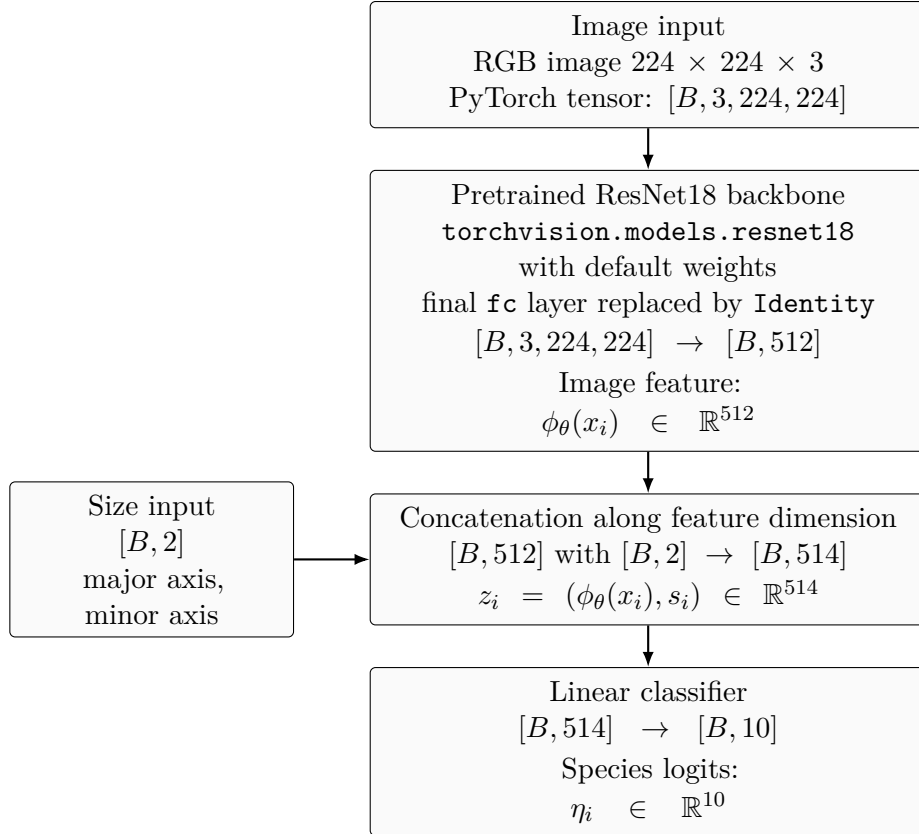


Figure 3.2: PyTorch implementation diagram of the ResNet18 baseline with 512-dimensional image features and size-feature fusion. The diagram shows the two-feature size-input version; in the ratio-based variants, the size vector has dimension three and the classifier input dimension increases from 514 to 515.

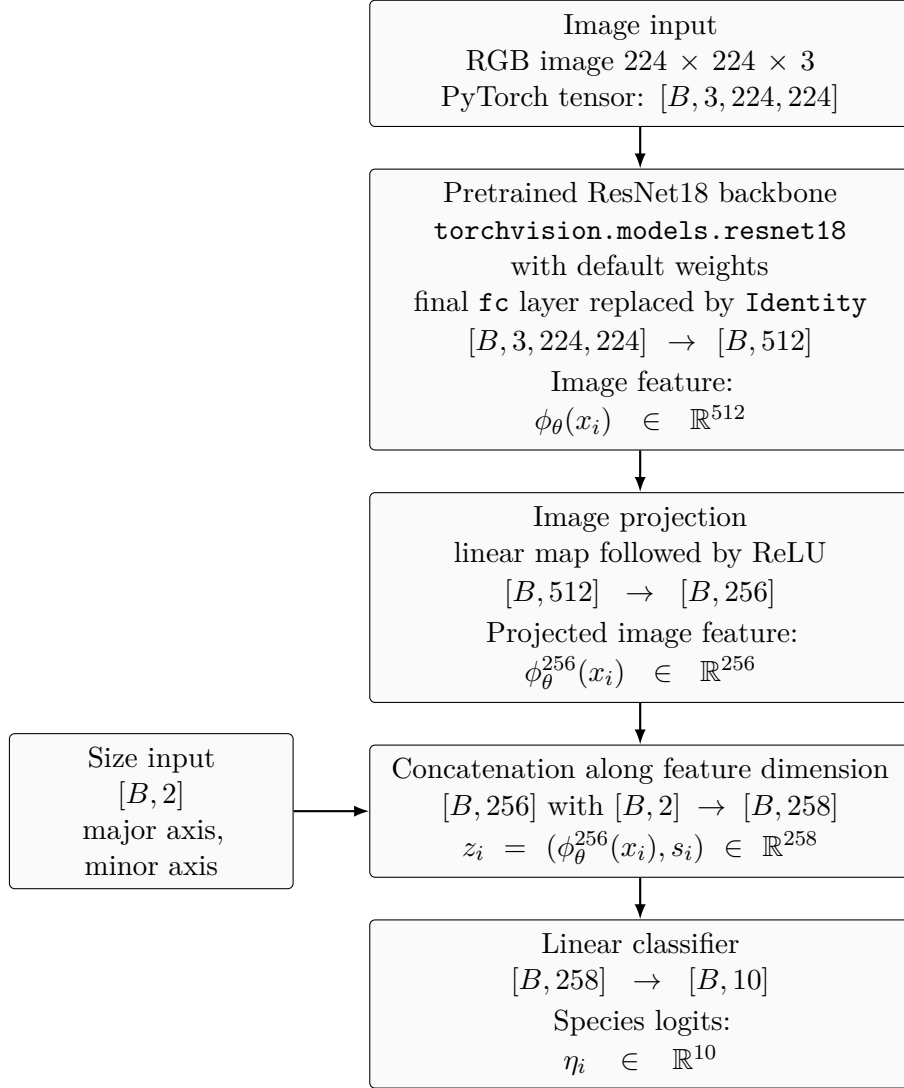


Figure 3.3: PyTorch implementation diagram of the projected ResNet18 model with 256-dimensional image features and size-feature fusion.

3.3.4 ConvNeXt-Tiny architecture

In addition to the ResNet-based models, a ConvNeXt-Tiny backbone was evaluated as a higher-capacity image feature extractor. The motivation was to investigate to what extent classification errors were due to limitations in the image representation or the handling of size variables. ConvNeXt was introduced as a “modernized” convolutional network obtained by gradually adapting a standard ResNet design towards more recent architectural choices, while retaining the convolutional assumption that local image patterns are informative across spatial locations (Liu et al. 2022; He et al. 2016).

Compared with ResNet18, the main architectural difference is in the design of the feature blocks. ResNet18 is built from relatively small residual blocks with 3×3 convolutions and identity skip connections, introduced to ease optimization of deeper networks through residual mappings (He et al. 2016). ConvNeXt uses a more modern block design with depthwise convolutions, larger effective kernels, LayerNorm, and an inverted bottleneck-style expansion, while still remaining a pure convolutional architecture (Liu et al. 2022).

The implemented model used a pretrained ConvNeXt-Tiny network from `torchvision`, for feature extraction. After global average pooling, the image branch produced an embedding

$$\phi_{\theta_\phi}(x_i) \in \mathbb{R}^{768}.$$

With the two size features, the fused representation had dimension

$$z_i \in \mathbb{R}^{770}.$$

The single head layer used in the ResNet18-based models was replaced by a shallow multilayer perceptron consisting of a linear layer from 770 to 256 hidden units, a ReLU activation, dropout with probability 0.2 for regularization (Srivastava et al. 2014), and a final linear layer mapping to the 10 output classes. The ConvNeXt-Tiny model therefore differed from the ResNet18-based models both in the image backbone and in the classifier head. The PyTorch implementation of this architecture is shown in Figure 3.4.

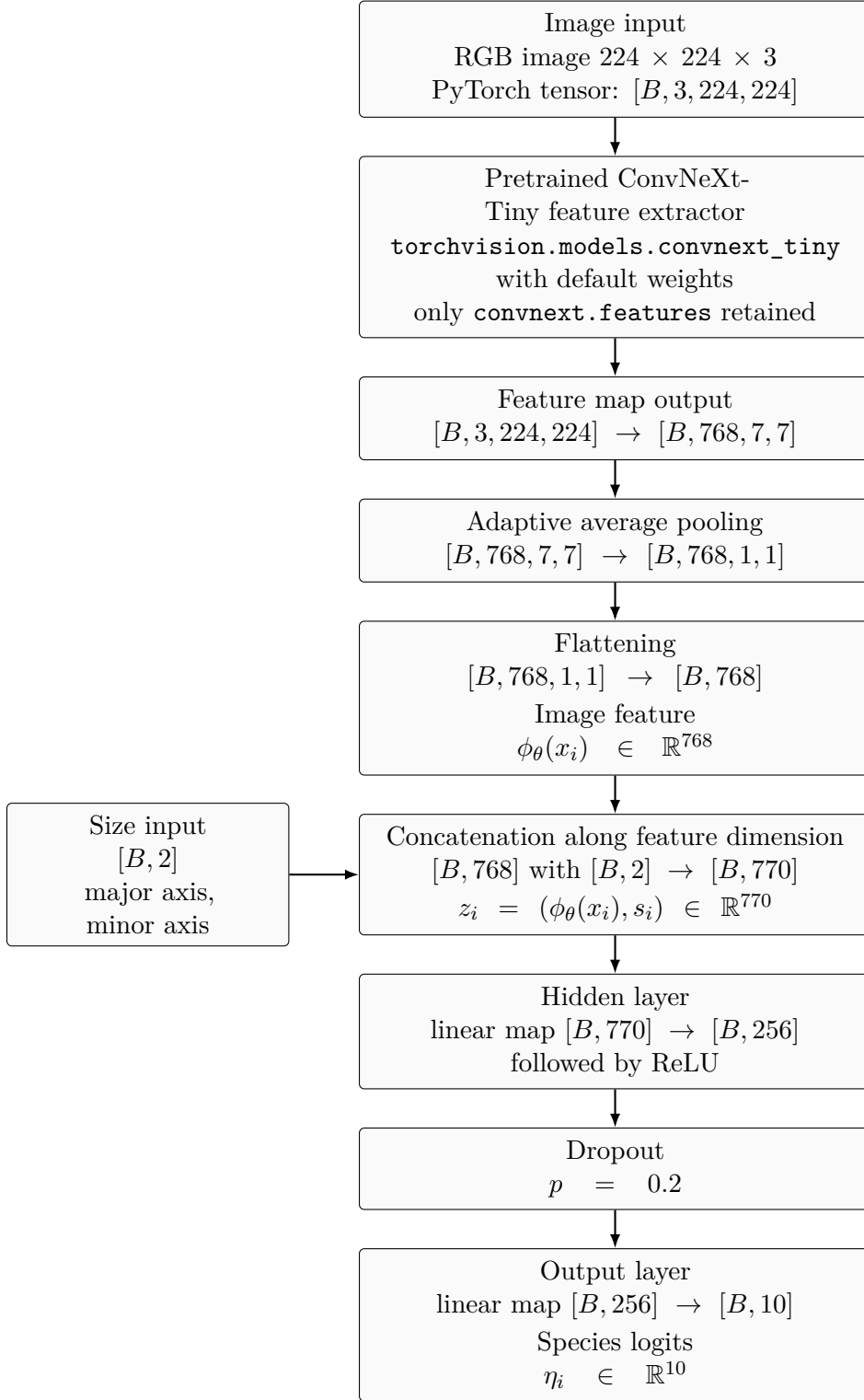


Figure 3.4: PyTorch implementation diagram of the ConvNeXt-Tiny model with 768-dimensional image features, size-feature fusion, and a $770 \rightarrow 256 \rightarrow 10$ classifier head. The diagram shows the two-feature size-input version; in ratio-based variants, the fused representation has dimension 771.

3.3.5 Trainable parameter counts

Since the pretrained backbones were fine-tuned end-to-end, the number of trainable parameters includes both the convolutional feature extractor and the added size-fusion classifier head. The backbone implementations were taken from `torchvision` (TorchVision Contributors 2026b; TorchVision Contributors 2026a), while the final classifier layers were modified for the present ten-class pollen classification task. The counts in Table 3.2 were computed from the PyTorch implementations used in this thesis using the command `sum(p.numel() for p in model.parameters() if p.requires_grad)`.

Here, `numel()` returns the number of scalar elements in a parameter tensor, and `requires_grad` indicates whether gradients are computed for that tensor during training (PyTorch Contributors 2026).

Table 3.2: Trainable parameter counts by model.

Component	R512	R256	C768
Image feature extractor	11,176,512	11,176,512	27,818,592
Projection layer		131,328	
Classifier head	5,150	2,590	199,946
Total	11,181,662	11,310,430	28,018,538

The R512 and R256 models are similar in size because both use the same ResNet18 feature extractor; the R256 model only adds a relatively small $512 \rightarrow 256$ projection layer. The C768 model is substantially larger because the ConvNeXt-Tiny feature extractor contains more parameters than ResNet18. The number of trainable parameters is large relative to the 8,078 training images, which motivates the use of pretrained ImageNet weights, data augmentation, validation-based checkpoint selection, and early stopping.

3.3.6 Loss functions

The primary training objective was the multinomial cross-entropy loss. For an observation with true class y_i , this loss is

$$\ell(y_i, p_\theta(x_i, s_i)) = - \sum_{k=1}^{10} \mathbf{1}_{\{y_i=k\}} \log p_{\theta,k}(x_i, s_i),$$

where $p_{\theta,y_i}(x_i, s_i)$ is the predicted probability assigned to the true class. Minimizing this loss corresponds to maximum likelihood estimation under a conditional multinomial model and is natural when the network outputs a class-probability vector (LeCun, Bengio, and Hinton 2015).

Focal loss was also evaluated to examine if increased emphasis on difficult observations

could improve recall for species hard to classify. Focal loss modifies cross-entropy by down-weighting observations that are already classified with high confidence and assigning greater weight to observations that are misclassified or uncertain (Lin et al. 2017). The focal loss is

$$\ell_{\text{FL}}(y_i, p_{\theta}(x_i, s_i)) = - \sum_{k=1}^{10} \mathbf{1}_{\{y_i=k\}} \alpha_k \left(1 - p_{\theta,k}(x_i, s_i)\right)^{\gamma} \log p_{\theta,k}(x_i, s_i),$$

where $\gamma \geq 0$ is a tunable focusing parameter and α_k is an optional weighting factor for species k .

When $\gamma = 0$ and $\alpha_k = 1$, focal loss reduces to ordinary cross-entropy. With class weights and $\gamma = 0$, it reduces to weighted cross-entropy. As γ increases, the loss places relatively more emphasis on hard-to-classify observations, that is, observations for which the predicted probability of the true class, $p_{\theta,y_i}(x_i, s_i)$, is small. This makes it a natural candidate for improving performance on difficult classes, although its effect depends on the data and model.

The implementation also allowed for class-specific weighting through α_k . If provided, these class weights scaled the loss contribution of the true class for each observation. In this thesis, focal loss was mainly used to test whether shifting the optimization emphasis toward hard examples could improve recall for species with low baseline recall. The default focusing parameter was set to $\gamma = 2$, as commonly used in the original focal-loss formulation, while other values were explored in sensitivity analyses (Lin et al. 2017).

3.3.7 Model fitting

Model fitting refers to estimating the network parameters θ from the training data by empirical risk minimization. Given training observations $\{(x_i, s_i, y_i)\}_{i=1}^n$, the fitted parameter vector is

$$\hat{\theta} = \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^n \ell(y_i, p_{\theta}(x_i, s_i)),$$

where ℓ is either cross-entropy or focal loss, depending on the experiment.

All models were implemented and trained in Python using PyTorch. Training was performed by mini-batch optimization with a batch size of 32. Optimization used the Adam optimizer (Kingma and Ba 2015), with learning rate 10^{-3} .

Most models were trained for up to 40 epochs with early stopping patience of 8 epochs (Prechelt 1998), so training stopped when validation accuracy no longer improved. The baseline reimplementations of Jurdell (2025) were instead trained for 20 epochs. Since the number of epochs in Jurdell’s original MATLAB implementation was not explicitly

reported, 20 epochs was chosen because it gave test performance closest to his reported accuracy and recall.

Model checkpoints were saved during training and later reloaded for evaluation and comparison. All experiments used the fixed random seed 42¹ for random operations such as the training-validation split, model initialization, mini-batch ordering, and stochastic data augmentation. This was done to improve reproducibility and to ensure that differences between model variants were not driven by uncontrolled changes in the random seed.

An explicit validation subset was introduced to monitor performance during training and reduce the risk of overfitting. In most experiments, the available training data were divided into training and validation subsets using an 80/20 split with a fixed random seed. This split was performed only within the original training set; the flower-level test set was not used during model fitting or checkpoint selection. The validation split was therefore used for model selection and early stopping, not as a final estimate of flower-level generalization. Final performance was evaluated only on the held-out flower-level test set.

3.4 Evaluation

3.4.1 Evaluation metrics

Model performance was evaluated using accuracy, class-specific recall, macro recall, and confusion matrices. Let n denote the number of observations in the evaluation set, and let $\hat{y}_i$ be the predicted class label for observation i .

Accuracy is defined as

$$\text{Accuracy} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{\hat{y}_i = y_i\},$$

that is, the proportion of correctly classified observations. For species k , recall is defined as

$$\text{Recall}_k = \frac{\text{TP}_k}{N_k},$$

where TP_k is the number of correctly classified observations from species k , and N_k is the total number of observations from that species. Recall therefore measures the probability of correct classification conditional on the true species.

¹The value 42 has no statistical significance; it was used only as a fixed seed to make random operations reproducible.

Macro recall is the unweighted mean of the class-specific recalls,

$$\text{Macro recall} = \frac{1}{10} \sum_{k=1}^{10} \text{Recall}_k.$$

This metric is useful here because it gives equal weight to each species, regardless of class frequency, and therefore better reflects class-wise predictive balance.

Confusion matrices were used to examine the structure of the classification errors and to identify persistent misclassification patterns between specific species (Powers 2011; Olsson et al. 2021).

Precision and F1-score were also considered but were not used as primary metrics. Precision measures how often predictions of a given species are correct, whereas recall measures how often grains from a true species are correctly classified. F1-score combines precision and recall into a single number and is useful when both types of error are equally central (Powers 2011). In this thesis, the main focus was instead on class-wise identification performance and on identifying which true species remained difficult to classify. Class-specific recall and macro recall were therefore more directly aligned with the research questions.

3.4.2 Class difficulty and feature-space structure

Even when class frequencies are reasonably similar, classification performance may still differ substantially between species. In this application, some species achieve consistently high recall, whereas others are repeatedly confused with one or two specific species. This suggests that the main difficulty is not severe class imbalance, but class difficulty (Jurdell 2025).

Class difficulty may arise from overlapping feature distributions, high within-species variation, or limited discriminatory information in the learned representation. Persistent confusion between two species can therefore be interpreted as weak separation in the representation used by the classifier. In particular, if observations from confusing species remain close in

$$z_i = (\phi_\theta(x_i), s_i),$$

the final classifier is likely to assign similar class probabilities and produce unstable decisions near the corresponding class boundary.

To investigate this structure, the later analysis visualizes selected low-recall species together with their most common confusion partners. The visualizations are based on t-distributed stochastic neighbor embedding (t-SNE), a nonlinear method that emphasizes local neighborhood structure in a low-dimensional map (Maaten and Hinton 2008).

The method first defines pairwise similarities between observations in the original high-dimensional representation and then constructs a two-dimensional configuration that preserves these local similarities as well as possible. In this study, t-SNE was used as an exploratory tool to examine whether frequently confused species formed overlapping regions, local subclusters, or partially separated groups in the learned feature space. Since t-SNE mainly preserves local neighborhoods, distances between well-separated groups should be interpreted with caution.

3.4.3 Sensitivity analysis with alternative test flowers

To assess whether the model comparisons depended on the particular flower-level train-test split inherited from Jurdell (2025), an additional sensitivity analysis was conducted for two difficult species: *Crepis capillaris* and *Hypochaeris radicata*. These species were selected because they formed the most persistent confusion pair in the earlier study. In Jurdell’s combined ResNet18 model with size features, *Crepis capillaris* had zero recall and was mainly classified as *Hypochaeris radicata*, while *Hypochaeris radicata* itself had recall 45.5%, low relative to most other species. This indicates that the boundary between these two species was especially difficult under the original flower-level split.

As shown in Table 2.1 on page 9, each of these species had five flowers in the split design: four assigned to training and one assigned to testing. For each species, one flower contained missing size measurements and was therefore excluded as a candidate test flower in the sensitivity analysis. For *Crepis capillaris*, this affected one image from one flower; for *Hypochaeris radicata*, it affected 83 images from one flower. This left four complete flowers per species as eligible test-flower candidates.

The sensitivity analysis considered all $4 \times 4 = 16$ combinations of eligible test flowers across the two species. For each combination, one eligible flower from *Crepis capillaris* and one eligible flower from *Hypochaeris radicata* were assigned to the test set, while the train-test assignments for the other eight species were kept fixed. The purpose was to isolate the effect of changing the unseen flowers for the two difficult species, without also changing the test composition of the remaining species, and to examine whether the observed performance patterns were robust across different training-test splits for *Crepis capillaris* and *Hypochaeris radicata*.

All 16 split configurations were trained using the same model structure as C768-Aug870 (see Table 4.1 on page 30), with the same training protocol and fixed random seed 42. For each split configuration, the resulting model was evaluated using the same metrics as in the main analysis. The random seed was kept fixed because the aim was to study sensitivity to the choice of test flowers, rather than sensitivity to random initialization, mini-batch ordering, or stochastic augmentation. One of the 16 configurations, denoted

Case 11, corresponds to the original flower-level split used in the main experiment. This configuration is reported as Case 11 and is the original-split C768-Aug870 model.

4 Analysis

The study began with a Python reimplementation of the combined image-and-size model in Jurdell (2025). It then examined whether predictive performance could be improved through modified preprocessing, alternative loss functions, a reduced image representation, and a stronger image backbone. For each evaluated model, the analysis focuses on three aspects: how the model differs from earlier variants, its aggregate predictive performance in terms of test accuracy and macro recall, and the main class-specific error patterns. An overview can be found in Table 4.1.

The main model comparisons are based on the fixed-seed training protocol described in Section 3.3. Consequently, small numerical differences between models should be interpreted cautiously. The interpretation below therefore emphasizes larger performance changes and recurring class-specific patterns rather than minor differences in decimal values. Across the experiments, *Crepis capillaris* remains the most difficult species, although later models reduce this problem substantially.

The experiments in this thesis should be interpreted as a controlled computational model-comparison study rather than as a randomized controlled trial. The experimental units are trained model configurations evaluated on the same held-out flower-level test set, and the response variables are predictive performance measures such as accuracy, macro recall, class-wise recall, and confusion-matrix structure. Several modelling factors were varied, including imputation strategy, validation-based model selection, augmentation level, image-feature dimension, loss function, and backbone architecture. However, the design is not a full factorial design, since not all combinations of factors were evaluated, and some changes are bundled together. For example, the ConvNeXt-Tiny model differs from the ResNet18 models both in the image backbone and in the classifier head. The results are therefore interpreted as evidence from structured ablation and model comparison, not as causal estimates from a randomized controlled trial.

Table 4.1: Overview of the main design choices in the evaluated models.

Model alias	Backbone	Imputation	Val. split	Aug.	Ratio	Loss
ResNet18 (512-dim)						
R512-Base	R18-512	–	N	–	N	CE
R512-Std	R18-512	Mean	Y	–	N	CE
R512-Aug870	R18-512	Mean	Y	870	N	CE
R512-Aug1000	R18-512	Mean	Y	1000	N	CE
R512-FL1	R18-512	Mean	Y	–	N	FL1
R512-FL2	R18-512	Mean	Y	–	N	FL2
R512-Boot-Aug870	R18-512	Bootstrap	Y	870	N	CE
R512-Boot-Aug870-Rat	R18-512	Bootstrap	Y	870	Y	CE
ResNet18 (256-dim)						
R256-Std	R18-256	Mean	Y	–	N	CE
R256-Boot	R18-256	Bootstrap	Y	–	N	CE
R256-Aug870	R18-256	Mean	Y	870	N	CE
R256-Aug870-FL1	R18-256	Mean	Y	870	N	FL1
R256-Aug870-FL2	R18-256	Mean	Y	870	N	FL2
ConvNeXt-Tiny (768-dim)						
C768-Std	CNeXt-768	Mean	Y	–	N	CE
C768-Aug870	CNeXt-768	Mean	Y	870	N	CE

Note. Columns indicate the backbone architecture and feature dimension (*Backbone*); the method used to handle missing size values (*Imputation*); whether a validation split was used during training (*Val. split*); the target number of observations per class under class-balancing augmentation (*Aug.*); whether the ratio feature was included (*Ratio*); and the training loss function (*Loss*).

Model aliases are constructed to be self-explanatory. The prefix denotes the backbone family and feature dimension (**R** = ResNet18, **C** = ConvNeXt-Tiny; 256, 512, 768 = feature dimension). The suffix indicates the main modification relative to the standard pipeline: **Std** = standard pipeline; **Aug870/Aug1000** = augmentation to a fixed number of observations per class; **Boot** = bootstrap imputation; **Rat** = inclusion of the ratio feature; **FL1/FL2** = focal loss with $\gamma = 1$ or $\gamma = 2$. Abbreviations: **Y** = yes; **N** = no; **CE** = cross-entropy; **FL1** = focal loss ($\gamma = 1$); **FL2** = focal loss ($\gamma = 2$); – = not included.

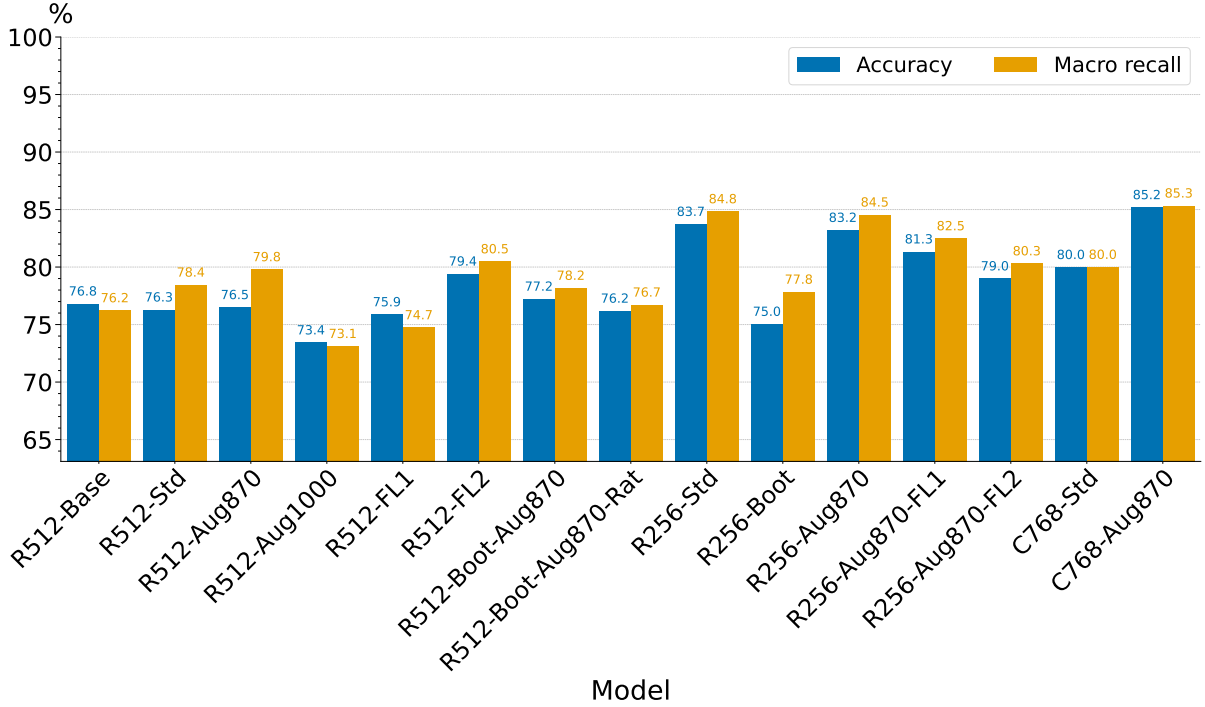


Figure 4.1: Test accuracy and macro recall for the 15 evaluated model variants, higher is better.

To complement the design overview in Table 4.1, Figure 4.1 summarizes the aggregate predictive performance of all evaluated models in terms of test accuracy and macro recall. This figure provides a global point of reference for the more detailed family-wise analysis that follows. While several of the early 512-dimensional ResNet18 variants produced moderate results, the reduced 256-dimensional ResNet18 models, especially R256-Std and R256-Aug870, yielded a clear improvement. The highest recorded aggregate performance was obtained by the ConvNeXt-Tiny model with augmentation, C768-Aug870, which achieved the highest values for both test accuracy and macro recall.

4.1 Baseline 512-dimensional ResNet18 models

This section analyzes the first family of evaluated models, all of which are built on the original ResNet18-based architecture with a 512-dimensional image representation before concatenation with the size variables, as described in Section 3.3.2. These models serve as the starting point of the empirical study and examine whether performance can be improved through comparatively modest modifications while keeping the basic representation unchanged. The investigated changes include species-wise mean imputation or bootstrap-based imputation of missing size values, the introduction of validation-based model selection, class-balancing augmentation at different target levels, focal loss with

different values of the focusing parameter γ , and the addition of the derived ratio feature.

R512-Base

R512-Base is the direct baseline reproduction and serves as the reference point for all later experiments. It uses the original 512-dimensional image representation together with the two size variables, without any other additions. It represents the simplest version of the modelling pipeline.

In overall terms, R512-Base achieved a test accuracy of 76.8% and a macro recall of 76.2%. This is close to Jurdell (2025), who reported 77.4% accuracy and a macro recall of approximately 73.0%, computed from the reported class-wise recalls. These values indicate moderate predictive performance, but they also conceal substantial heterogeneity across species. Several species achieved very high recall, showing that the model could separate some classes very well. This good performance was not evenly distributed.

The main weakness of R512-Base is the highly uneven species-level behaviour. In particular, *Crepis capillaris* was never correctly classified. *Bellis perennis* and *Hypochaeris radicata* were also noticeably weaker than the best species. Thus, the baseline error structure is not diffuse, but concentrated in a few difficult class boundaries that remain problematic throughout much of the study.

R512-Std

Relative to R512-Base, R512-Std introduces two changes: species-wise mean imputation for missing size values and an explicit validation split. These changes are modest structurally, but they are important methodologically, since imputation allows more observations to remain in the training set and the validation split provides a more principled basis for model selection.

The model achieved a test accuracy of 76.3% and a macro recall of 78.4%. Hence, overall accuracy decreased slightly relative to the baseline, but macro recall increased. This suggests that the modification did not primarily increase the total number of correct predictions, but instead redistributed performance more evenly across classes.

The most interesting class-specific change is that some weaker species improved despite the lack of a gain in overall accuracy. *Bellis perennis* improved clearly, and *Hypochaeris radicata* also became somewhat stronger. However, *Crepis capillaris* still had zero recall. Thus, R512-Std improved class balance without resolving the principal failure mode of the baseline architecture.

R512-Aug870

R512-Aug870 extends R512-Std by adding augmentation, with the smaller classes expanded up to 870 images. Structurally, this model keeps the same backbone and mean-imputed size data, but attempts to improve generalization by increasing within-class variation during training.

In aggregate, R512-Aug870 achieved a test accuracy of 76.5% and a macro recall of 79.8%. Compared with R512-Std, the gain in accuracy is very small, but the increase in macro recall is more noticeable.

The most notable improvements were for *Bellis perennis* and *Hypochaeris radicata*, both of which benefited from the augmented training setup. However, *Crepis capillaris* still remained at zero recall, which shows that simple augmentation alone was not enough to solve the most difficult class boundary. Thus, R512-Aug870 may be seen as a mild improvement on R512-Std, but not a qualitative change in the behaviour of the classifier.

R512-Aug1000

R512-Aug1000 keeps the same overall structure as R512-Aug870, but uses a more aggressive augmentation target, increasing class sizes to 1000 images instead of 870. The purpose is to examine whether more extensive balancing produces a further gain.

This setting was not beneficial. R512-Aug1000 achieved a test accuracy of 73.4% and a macro recall of 73.1%, both clearly below the corresponding values for both R512-Aug870 and the R512-Base. This indicates that the more aggressive augmentation harmed the classifier overall, likely by introducing too much artificial variation relative to the original data structure.

At the class level, the result is especially interesting because not all species deteriorated equally. In some cases the model showed a small improvement for *Crepis capillaris*, but this came at the cost of substantial deterioration for previously stronger classes, especially *Bellis perennis*. Thus, the 1000-setting did not improve the baseline family overall and suggests that stronger oversampling is not automatically beneficial.

R512-FL1

R512-FL1 is the direct focal-loss counterpart of R512-Std. It keeps the same 512-dimensional ResNet18 representation, species-wise mean imputation, validation split, and non-augmented training setup, but replaces cross-entropy with focal loss using $\gamma = 1$. This makes the comparison with R512-Std a direct test of whether placing more emphasis on difficult

examples can improve the weak species in the baseline 512-dimensional setting.

The model achieved a test accuracy of 75.9% and a macro recall of 74.7%. These values are weaker than those of R512-Std and R512-Aug870, suggesting that focal loss with $\gamma = 1$ did not improve the baseline 512-dimensional architecture in aggregate terms.

The class-specific recalls show why. Although many easy species remained almost perfectly classified, the difficult classes did not improve in a useful way. *Crepis capillaris* remained essentially unclassified, with recall close to zero, and *Bellis perennis* also became very weak. *Hypochaeris radicata* remained moderate but far from strong. Hence, focal loss with $\gamma = 1$ did not provide a useful improvement for the difficult classes.

R512-FL2

R512-FL2 is the same as R512-FL1, but with a stronger focal-loss parameter, $\gamma = 2$. This further increases the emphasis on hard or misclassified examples.

In aggregate, R512-FL2 achieved a test accuracy of 79.4% and a macro recall of 80.5%. These are stronger than the values for R512-FL1, indicating that the larger focal parameter was more effective than $\gamma = 1$ in this baseline family. In fact, the model became competitive with the stronger baseline-style variants.

The detailed recalls show a mixed but informative pattern. *Bellis perennis* improved strongly relative to R512-FL1, and most of the easier species remained close to perfect recall. This may at first seem unintuitive, since focal loss is intended to emphasize hard-to-classify observations. However, emphasizing hard observations does not guarantee that the hardest species will improve. If the learned representation does not separate *Crepis capillaris* sufficiently from its main confusion partner, the loss function alone cannot resolve that class boundary. In this case, R512-FL2 improved the balance of the baseline family mainly by improving some moderately difficult species, but *Crepis capillaris* still had zero recall. The result therefore suggests that the main limitation for this species was not only the loss weighting, but also the feature representation available to the classifier.

R512-Boot-Aug870

R512-Boot-Aug870 is the bootstrap-imputation counterpart of R512-Aug870. It keeps the same 512-dimensional ResNet18 backbone, validation split, augmentation to 870 images per class, and cross-entropy loss, but replaces species-wise mean imputation with bootstrap-based imputation of the missing size values. This makes the comparison with R512-Aug870 a direct test of whether a more variable imputation strategy improves the baseline 512-dimensional setting.

The model achieved a test accuracy of 77.2% and a macro recall of 78.2%. Relative to R512-Aug870, this corresponds to a small increase in accuracy but a lower macro recall. Thus, bootstrap imputation did not yield a clear overall improvement over the simpler mean-imputation approach.

The species-level results suggest a redistribution rather than a reduction of errors. *Bellis perennis* improved markedly, but *Hypochaeris radicata* weakened substantially, and *Crepis capillaris* still had zero recall. Hence, bootstrap imputation changed which species benefited from the model, but it did not resolve the main class-specific weakness of the baseline architecture.

R512-Boot-Aug870-Rat

R512-Boot-Aug870-Rat extends R512-Boot-Aug870 by adding the minor-to-major axis ratio m_i/M_i as an extra derived size-related feature. Structurally, this is a small feature-engineering step intended to make shape information more explicit.

The resulting model achieved a test accuracy of 76.2% and a macro recall of 76.7%. Both values are slightly lower than those of R512-Boot-Aug870, indicating that the additional ratio feature did not improve overall performance.

At the class level, the ratio feature shifted performance in a different way compared to R512-Boot-Aug870 rather than improved it. *Hypochaeris radicata* benefited noticeably instead of weakening, suggesting that the added shape information may have helped for that species. However, *Bellis perennis* weakened instead of improving, and *Crepis capillaris* remained essentially unresolved. Hence, R512-Boot-Aug870-Rat did not improve on the better baseline-family models overall.

4.2 ResNet18 models with reduced 256-dimensional representation

This section analyzes the models in which the original 512-dimensional ResNet18 image representation was replaced by a projected 256-dimensional representation before concatenation with the size features, as described in Section 3.3.3. Unlike the earlier pre-processing and training modifications, this changes the learned feature space itself and therefore represents a more substantial architectural modification.

The aggregate comparison in Figure 4.1 indicates that mean imputation, validation splitting, and augmentation could improve performance within the original baseline architecture, although the gains were limited. The validation split was retained to monitor

overfitting and support model selection, while imputation was retained to preserve incomplete observations in the relatively small training set. Augmentation was kept only in the 870-setting, where classes were expanded up to the size of the largest original class. The more aggressive 1000-setting reduced both test accuracy and macro recall, suggesting that excessive oversampling introduced more harm than benefit.

Within the 256-dimensional representation setting, the analysis therefore compares mean and bootstrap imputation, augmentation to 870 observations per class, and focal loss with $\gamma = 1$ and $\gamma = 2$. The aim is to assess both whether the compact representation improves predictive performance and which training choices remain beneficial after changing the feature representation.

R256-Std

R256-Std is the direct 256-dimensional counterpart of R512-Std . It uses the same overall training setup: species-wise mean imputation, a validation split, no augmentation, no ratio feature, and cross-entropy loss; but replaces the original 512-dimensional image representation with the reduced 256-dimensional version. This makes it possible to assess more directly whether the change in representation alone improves performance relative to the corresponding model in the baseline family.

This model achieved a test accuracy of 83.7% and a macro recall of 84.8%. These values represent a clear improvement over all earlier baseline-style models, showing that the reduced representation substantially strengthened the classifier. The result therefore indicates that the main improvement in this part of the study comes from the changed representation itself rather than from additional preprocessing or alternative training choices.

The species-level recalls support the same conclusion. Relative to the earlier baseline-family models, *Bellis perennis* and *Hypochaeris radicata* improved markedly, indicating that the reduced representation handled several previously unstable class boundaries more effectively. At the same time, *Crepis capillaris* still had zero recall. Thus, R256-Std represents a substantial overall step forward, but it does not yet resolve the most difficult classification problem in the study.

R256-Boot

R256-Boot keeps the same 256-dimensional architecture as R256-Std, but replaces species-wise mean imputation with bootstrap-based imputation. Its main purpose is therefore to assess whether the stronger representation also benefits from a more variable imputation

strategy.

This change did not improve performance. R256-Boot achieved a test accuracy of 75.0% and a macro recall of 77.8%, both clearly below the corresponding values for R256-Std.

The class-specific pattern is consistent with this interpretation. Several of the gains observed for R256-Std were weakened, and the more balanced behaviour of that model was not maintained. In particular, *Crepis capillaris* remained very weak, while the broader improvement across difficult species was reduced.

R256-Aug870

R256-Aug870 extends R256-Std by adding class-balancing augmentation up to 870 observations per class, while keeping the same 256-dimensional ResNet18 representation, species-wise mean imputation, validation split, and cross-entropy loss. In this sense, it is the direct augmented counterpart of R256-Std and reduced representation counterpart of R512-Aug870. The comparison with R256-Std therefore isolates the effect of augmentation within the 256-dimensional setting, while the comparison with R512-Aug870 shows whether the stronger representation remains beneficial when the training setup is otherwise unchanged.

The model achieved a test accuracy of 83.2% and a macro recall of 84.5%. These values are only slightly below those of R256-Std, showing that augmentation did not materially improve the already strong 256-dimensional model in aggregate terms. However, the difference is small enough that the two models may be regarded as equal.

The class-specific pattern is more interesting than the aggregate numbers. *Bellis perennis* improved further, and *Crepis capillaris* reached a small but nonzero recall. This is notable because it is one of the first signs that the hardest species may benefit from a stronger representation combined with augmentation. At the same time, *Hypochaeris radicata* deteriorated relative to R256-Std. Thus, augmentation again redistributed errors rather than removing them.

R256-Aug870-FL1

R256-Aug870-FL1 is the focal-loss counterpart of R256-Aug870. It replaces cross-entropy with focal loss with $\gamma = 1$. This tests whether the strong representation can be improved further by emphasizing hard observations during optimization.

R256-Aug870-FL1 achieved a test accuracy of 81.3% and a macro recall of 82.5%. Both values are below those of R256-Aug870, which achieved 83.2% accuracy and 84.5% macro recall. Thus, replacing cross-entropy with focal loss at $\gamma = 1$ weakened overall perfor-

mance rather than improving it.

At the species level, the effect was limited and uneven. *Crepis capillaris* improved slightly relative to the zero-recall models, but the gain remained small in absolute terms, while *Hypochaeris radicata* deteriorated.

R256-Aug870-FL2

R256-Aug870-FL2 is the same as R256-Aug870-FL1 , but with the stronger focal-loss parameter $\gamma = 2$. The model achieved a test accuracy of 79.0% and a macro recall of 80.3%. These values are substantially below those of the best 256-dimensional cross-entropy models and also below R256-Aug870-FL1 . Therefore, increasing the focal parameter further weakened the model rather than strengthening it.

The class-specific behaviour shows that the additional emphasis on difficult examples did not resolve the principal failure modes. *Crepis capillaris* again remained extremely weak, while several other classes lost stability. Thus, the main limitation in this family appears not to be the loss function itself, but the difficulty of the class boundaries. The best-performing 256-dimensional models are therefore the cross-entropy variants rather than the focal-loss variants.

4.3 ConvNeXt-Tiny models

This section analyzes the final model family, in which the ResNet18 backbone was replaced by ConvNeXt-Tiny, as described in Section 3.3.4. This is the most substantial architectural change in the study, since it replaces the image feature extractor itself rather than modifying the representation dimension, preprocessing, or loss function within the same backbone. Two ConvNeXt-Tiny variants were evaluated: a non-augmented model with mean imputation and validation-based model selection, and a second variant that adds class-balancing augmentation to 870 observations per class. The purpose of this section is therefore to assess whether a stronger backbone improves overall predictive performance and, in particular, whether it reduces the persistent class confusions that remained under the ResNet18-based models.

C768-Std

C768-Std is the ConvNeXt-Tiny counterpart to the non-augmented mean-imputed ResNet-based models R512-Std and R256-Std. Relative to R512-Std, it combines the same training setup with a substantially stronger feature extractor, and relative to R256-Std, it

replaces the reduced ResNet18 representation with ConvNeXt-Tiny while preserving the surrounding training pipeline. This makes C768-Std a useful test of whether a stronger backbone alone is sufficient to improve performance beyond the best ResNet-based variants.

C768-Std achieved a test accuracy of 80.0% and a macro recall of 80.0%. These results are clearly stronger than those of the corresponding 512-dimensional baseline-style model R512-Std, but they remain below those of the best 256-dimensional model R256-Std. This is an important result, since it shows that a more modern architecture does not automatically dominate a more effective representation within the earlier model family.

At the class level, the model shows both progress and limitations. Most notably, *Crepis capillaris* achieved a small but nonzero recall, which is an improvement over several earlier models in which this species was never correctly classified. This suggests that the ConvNeXt-Tiny backbone captured some additional discriminative structure for the most difficult class. However, the improvement was still modest, and *Hypochaeris radicata* remained clearly weaker than under R256-Std. Thus, C768-Std should not be interpreted as a clear breakthrough, but rather as evidence that backbone improvements alone can partly reduce the most persistent class difficulty, while still falling short of the strongest overall ResNet-based configuration.

C768-Aug870

C768-Aug870 extends C768-Std by adding class-balancing augmentation to 870 observations per class, with the remaining settings unchanged. It is therefore the ConvNeXt-Tiny counterpart to R256-Aug870. This makes C768-Aug870 a direct test of whether the augmentation strategy that was previously most successful can produce a further gain when combined with the strongest backbone in the study.

C768-Aug870 achieved a test accuracy of 85.2% and a macro recall of 85.3%, which are the highest values among all evaluated models. Relative to C768-Std, this shows that augmentation was beneficial within the ConvNeXt-Tiny setting, unlike in some earlier comparisons where its effect was smaller or more mixed. Relative to the strongest ResNet-based models, C768-Aug870 also establishes a new best overall result. The main conclusion is therefore that the highest recorded performance in this study was obtained neither from a stronger backbone alone, nor from augmentation alone; but from their combination.

The species-level recalls clarify why C768-Aug870 is the strongest model overall. Most importantly, *Crepis capillaris* improved substantially to a clearly nonzero recall, representing the first substantial improvement for the most persistent failure class in the study.

Hypochaeris radicata also improved relative to C768-Std, although it still did not match the exceptionally strong recall achieved in the best 256-dimensional ResNet variant. At the same time, *Bellis perennis* was somewhat weaker than in some earlier models. Thus, C768-Aug870 is best interpreted not as uniformly best for every species, but as the model that delivers the strongest overall balance between aggregate performance and reduction of the most important class-specific weaknesses.

4.4 Sensitivity analysis

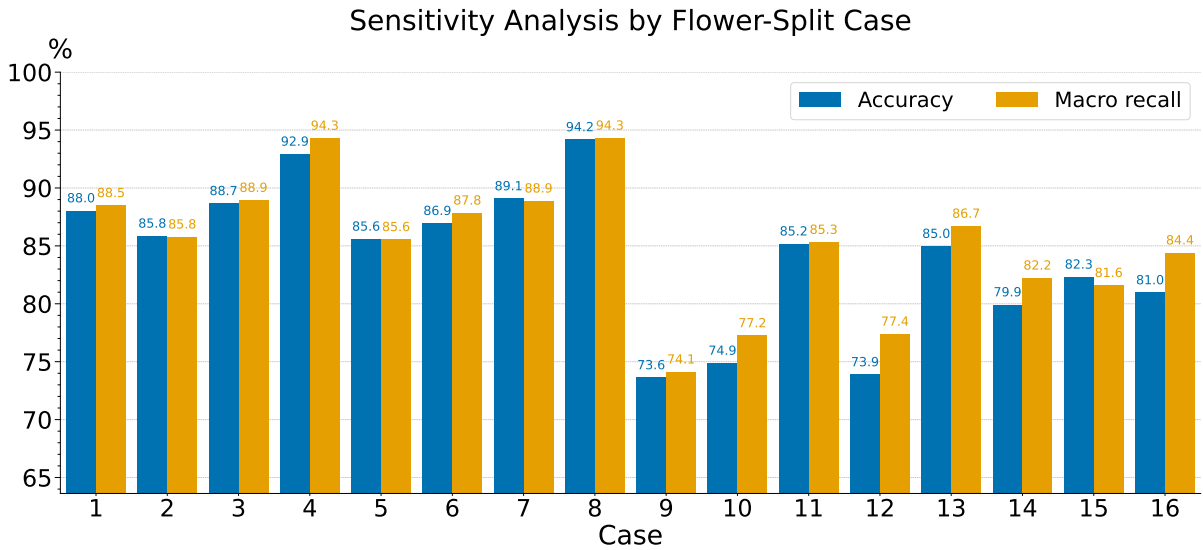


Figure 4.2: Test accuracy and macro recall across the 16 sensitivity-analysis cases; Case 11 is the original-split C768-Aug870 result.

To assess how strongly the main conclusions depended on the particular flower-level test split, an additional sensitivity analysis was carried out using the same model structure as the strongest model from the main experiments, C768-Aug870. The purpose was to evaluate the robustness of the best-performing configuration when the held-out test flowers for the two most problematic species, *Crepis capillaris* and *Hypochaeris radicata*, were varied. The test accuracy and macro recall values are shown in Figure 4.2.

Table 4.2: Summary statistics across the 16 sensitivity-analysis cases.

Metric	Range (%)	Mean (%)	Standard deviation (%)
Accuracy	(73.6–94.2)	84.2	6.2
Macro recall	(74.1–94.3)	85.2	5.7

Overall, the sensitivity analysis shows that the final model remained strong on average, but that its performance was not invariant to the composition of the test set. Across

the 16 fixed-seed flower-split configurations, the mean accuracy was 84.2% with a standard deviation of 6.2 percentage points, and the mean macro recall was 85.2% with a standard deviation of 5.7 percentage points. Thus, C768-Aug870 retained strong average performance, but the results also show non-negligible split sensitivity.

The strongest split-dependence remained concentrated in the two species that dominated the error structure throughout the study, *Crepis capillaris* and *Hypochaeris radicata*.

The variation in aggregate performance across the 16 split configurations was driven primarily by the class-wise recalls of *Crepis capillaris* and *Hypochaeris radicata*. This is consistent with the main experiments, where these two species formed the dominant error pattern. Their recalls varied substantially, from 11.0% to 100.0% for *Crepis capillaris* and from 0.0% to 96.0% for *Hypochaeris radicata*. The remaining species were comparatively stable. The sensitivity analysis therefore refines the main conclusion of the thesis: the final model has strong overall predictive ability, but its remaining weakness is concentrated in an unstable separation between a small subset of difficult species, rather than in a general lack of performance across all classes.

5 Discussion of results

This chapter synthesises the empirical findings reported in Chapter 4. Whereas the previous chapter presented the results model by model, the purpose here is to discuss the main patterns across the full sequence of experiments. The emphasis is therefore not on repeating the numerical results in detail, but on identifying which design choices contributed most to predictive performance, which changes mainly redistributed errors across species, and which classification difficulties remained unresolved throughout the study.

5.1 Overall accuracy and recall

The results show that the strongest improvements did not come from alternative loss functions or small feature-engineering changes, but from changes in representation and backbone. The baseline model, R512-Base, achieved a test accuracy of 76.8% and a macro recall of 76.2%, a reasonable but clearly imperfect starting point. Several modifications of the original 512-dimensional ResNet18 family, led to modest improvements, but did not address the dominant failure modes. In particular, *Crepis capillaris* remained unresolved for most of this model family.

A clearer step forward was obtained when the original 512-dimensional ResNet18 representation was replaced by the reduced 256-dimensional version. The 256-dimensional representation improved both test accuracy and macro recall under otherwise comparable training settings (R512-Std versus R256-Std, and R512-Aug870 versus R256-Aug870). Especially *Bellis perennis* and, in some settings, *Hypochaeris radicata* show improvements. However, even the stronger 256-dimensional models did not fully resolve the persistent difficulty of *Crepis capillaris*.

The focal-loss experiments did not provide a consistent improvement. In general, focal loss changed the class-wise distribution of errors, but it did not yield a robust overall gain and did not remove the principal class boundary problems.

The final architectural change, replacing ResNet18 with ConvNeXt-Tiny, produced the strongest overall result when combined with the more stable preprocessing choices identified earlier. The comparison between C768-Std and C768-Aug870 shows that augmentation remained beneficial in this setting, and C768-Aug870 achieved the highest aggregate performance, with a test accuracy of 85.17% and a macro recall of 85.31%. Most importantly, it was the first model to produce a substantial improvement for *Crepis capillaris*,

indicating that the most persistent errors were linked to limitations in the models' representational capacity.

From a statistical perspective, the overall pattern suggests that the dominant limitation of the earlier models was not simply suboptimal weighting of hard observations or insufficient balancing of class counts. Instead, the main gains arose when the feature representation itself became more informative. This indicates that the central challenge in the pollen-classification problem is representational: the harder species are not difficult merely because they are underweighted during training, but because their class boundaries are genuinely hard to separate using the available feature space.

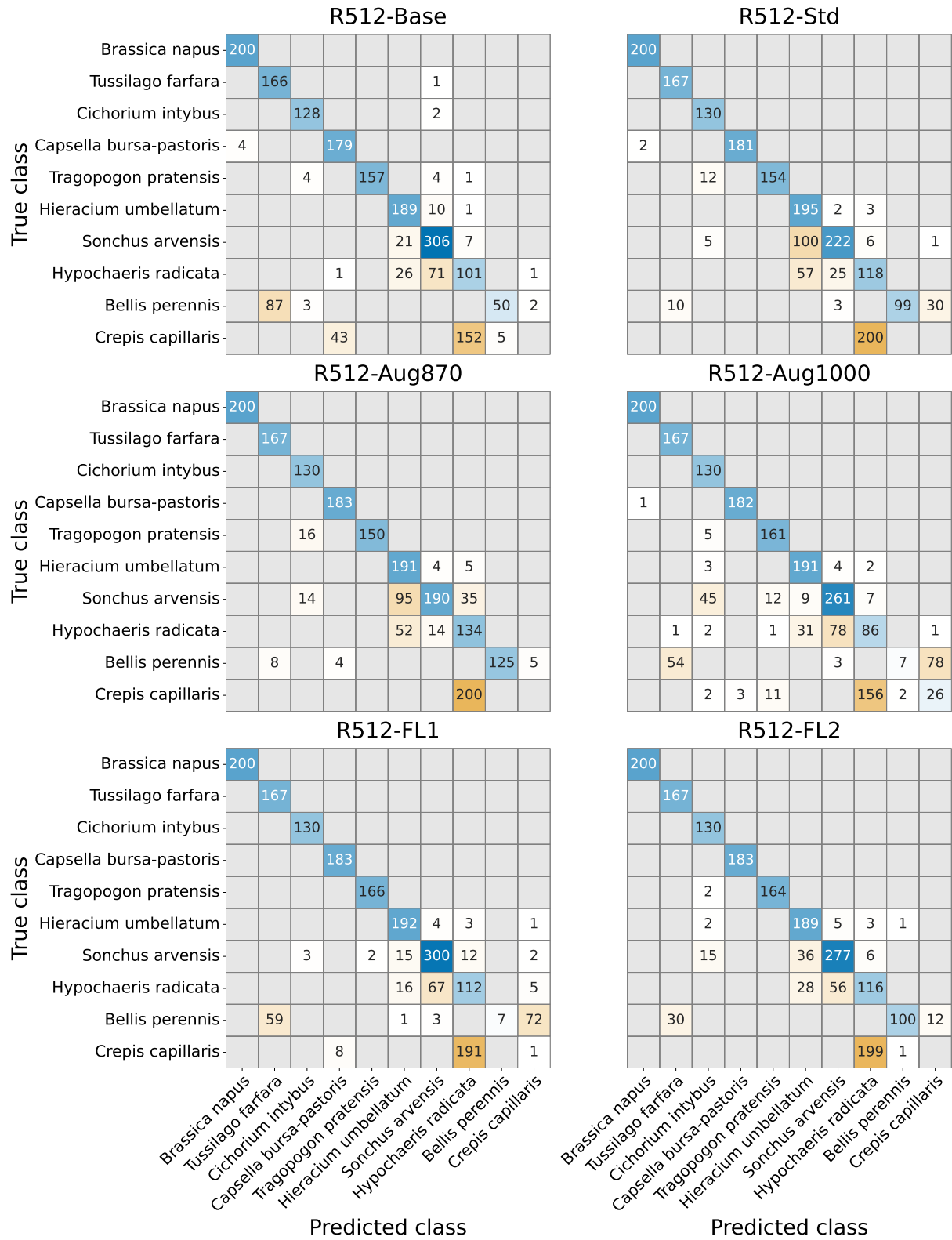


Figure 5.1: Confusion matrices for all models. The class order is fixed to the same baseline ordering as in R512-Base, sorted by descending class-wise recall, to facilitate direct comparison of changes in the error structure across models.

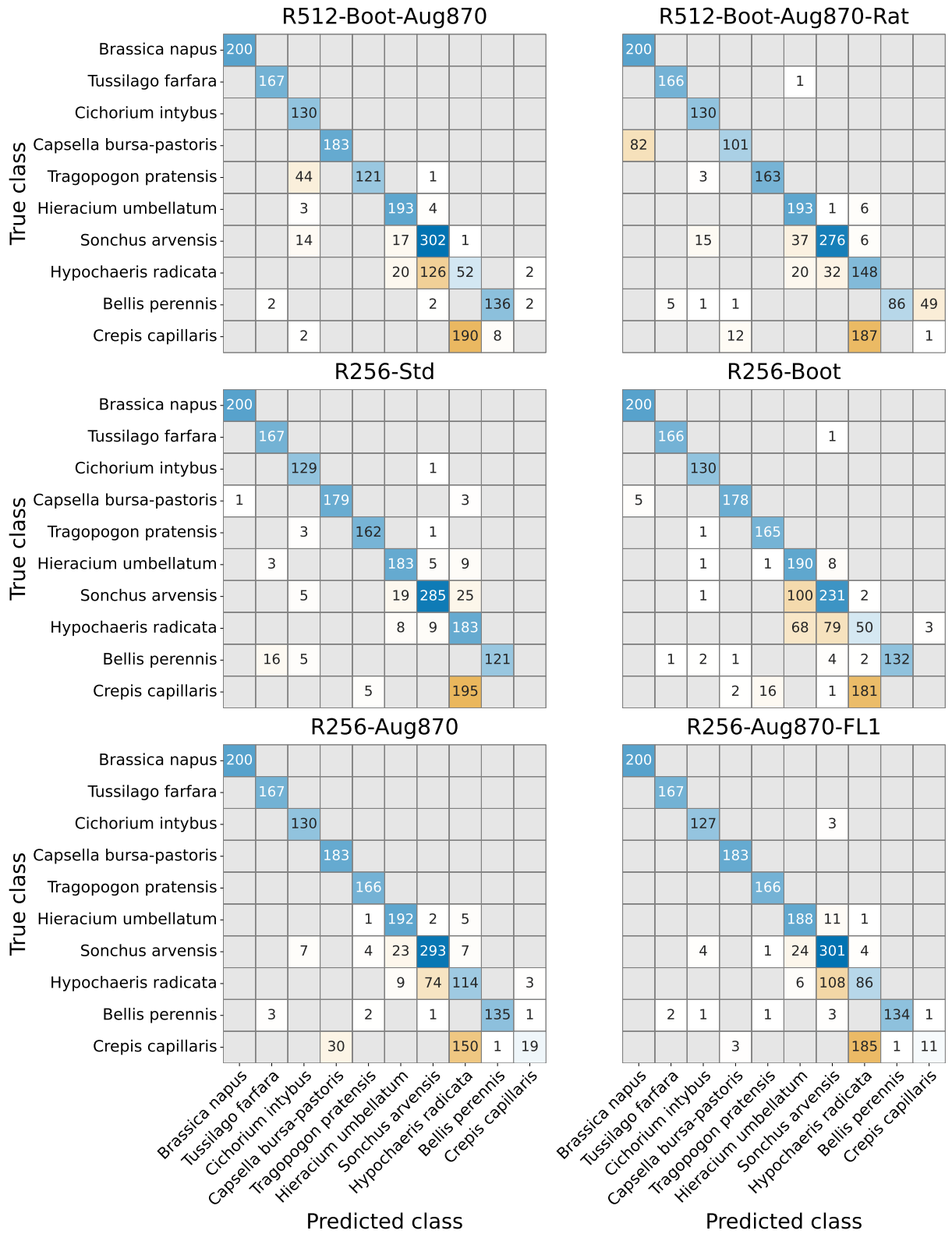


Figure 5.1: Confusion matrices for all models, continued. The class order is fixed to the same baseline ordering as in R512-Base, sorted by descending class-wise recall, to facilitate direct comparison of changes in the error structure across models.

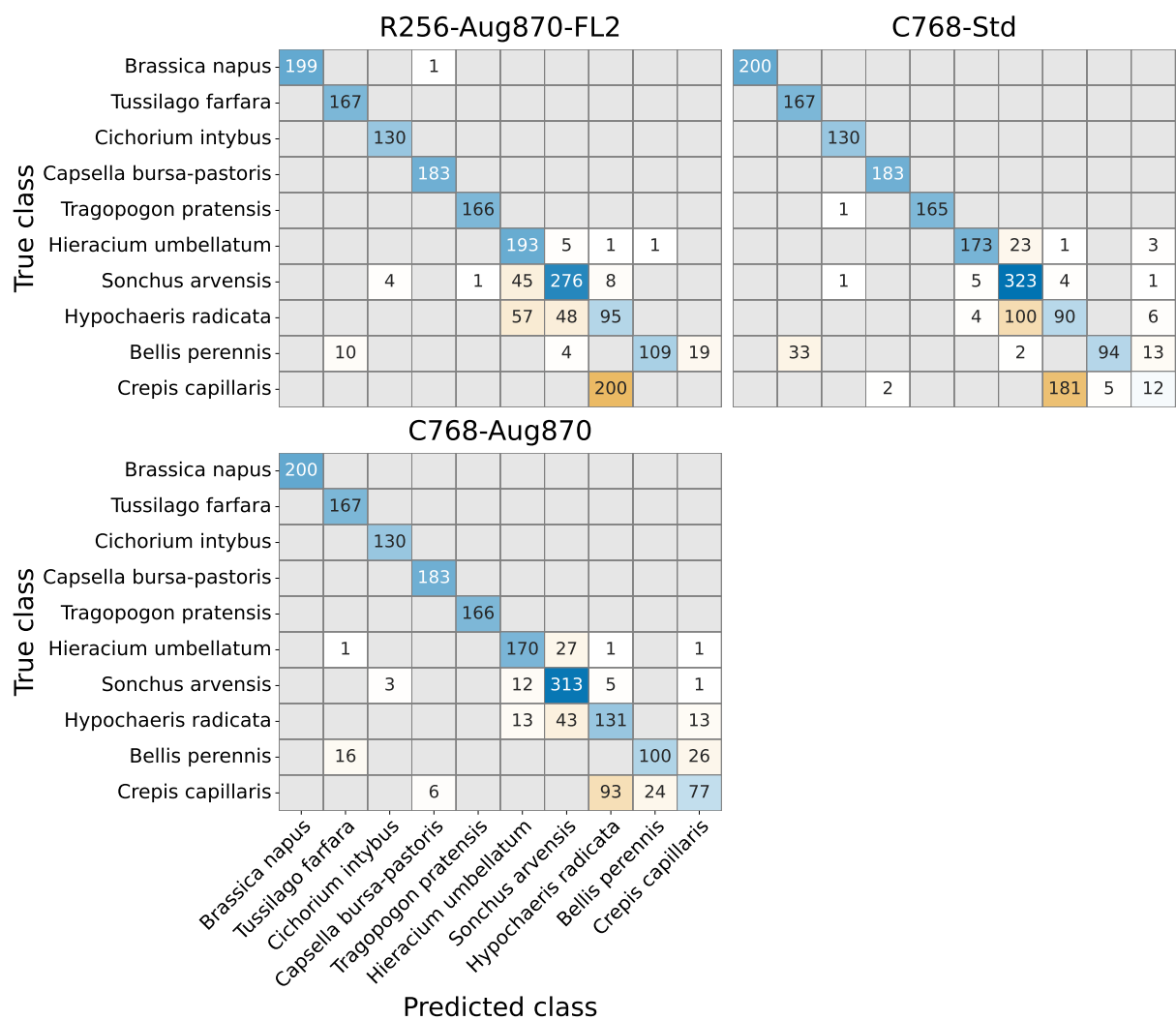


Figure 5.1: Confusion matrices for all models, continued. The class order is fixed to the same baseline ordering as in R512-Base, sorted by descending class-wise recall, to facilitate direct comparison of changes in the error structure across models.

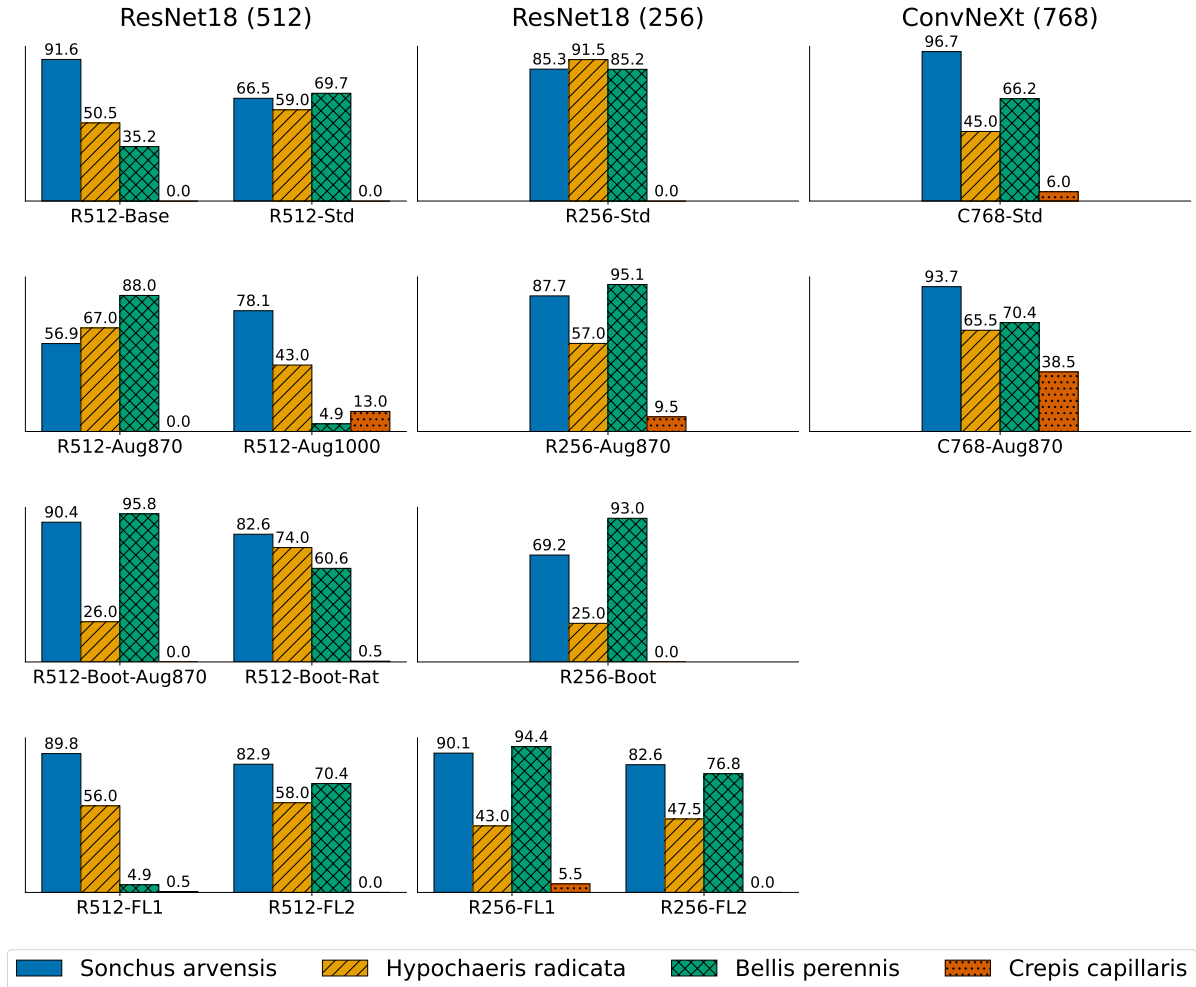


Figure 5.2: Recall across all models on the most difficult focus species.

5.2 Effect of augmentation

The augmentation experiments were introduced to assess whether a richer training distribution could improve generalization, particularly for species with weaker recall. Across the model sequence, augmentation did not produce a uniform gain across all classes. Instead, its main effect was to reshape *which* species were classified correctly. This pattern is visible in both the full per-species recall overview in Figure 5.2 and the confusion-matrix comparisons in Figure 5.1, where changes in the diagonal entries show which classes gained or lost correctly classified observations under augmentation.

In the 512-dimensional ResNet18 family, augmentation to 870 observations per class produced only a selective improvement. Relative to R512-Std, R512-Aug870 increased *Bellis perennis* from 99 correctly classified observations (69.7%) to 125 (88.0%), and *Hypochaeris radicata* from 118 (59.0%) to 134 (67.0%). These gains help explain why macro recall improved even though overall accuracy changed little. At the same time, *Sonchus arvensis* fell from 222 correctly classified observations (66.5%) to 190 (56.9%), and *Crepis capillaris* remained completely unresolved, with all 200 *Crepis capillaris* test observations assigned to *Hypochaeris radicata*. Thus, in this family, moderate augmentation improved some weaker classes, but it did not remove the central class boundary problem.

The more aggressive balancing strategy in R512-Aug1000 makes this limitation even clearer. Relative to R512-Aug870, *Crepis capillaris* increased to 26 correctly classified observations (13.0%), and *Sonchus arvensis* recovered to 261 (78.1%). However, *Bellis perennis* fell sharply to only 7 correct classifications (4.9%), and *Hypochaeris radicata* decreased to 86 (43.0%). This deterioration is also reflected in the aggregate metrics, with both test accuracy and macro recall dropping relative to the 870-image setting.

The 256-dimensional ResNet18 family produced a more favourable, but still selective, augmentation effect. Comparing R256-Std with R256-Aug870, *Bellis perennis* increased from 121 correctly classified observations (85.2%) to 135 (95.1%), and *Crepis capillaris* improved from 0 (0.0%) to 19 (9.5%). *Sonchus arvensis* changed only modestly, from 285 (85.3%) to 293 (87.7%). However, *Hypochaeris radicata* deteriorated sharply, falling from 183 correctly classified observations (91.5%) to 114 (57.0%). The confusion matrices show that this loss corresponded mainly to renewed confusion with *Sonchus arvensis*: the number of *Hypochaeris radicata* observations assigned to *Sonchus arvensis* increased from 9 to 74. In parallel, *Crepis capillaris* showed its first nonzero recall in this family, but most observations were still assigned to *Hypochaeris radicata*. Augmentation therefore improved some weaker classes, especially *Bellis perennis*, and partially recovered the hardest class, but it did so at the cost of weakening a class that had previously been classified well.

The clearest positive effect of augmentation appears in the ConvNeXt-Tiny family. From C768-Std to C768-Aug870, *Hypochaeris radicata* increased from 90 correctly classified observations (45.0%) to 131 (65.5%), *Bellis perennis* from 94 (66.2%) to 100 (70.4%), and *Crepis capillaris* from 12 (6.0%) to 77 (38.5%). At the same time, *Sonchus arvensis* remained strong, decreasing only slightly from 323 correctly classified observations (96.7%) to 313 (93.7%). Unlike the ResNet-based families, augmentation here improved not only weak classes but also the overall error structure. The confusion matrices show that the most important change occurred for *Crepis capillaris*: the number of observations misclassified as *Hypochaeris radicata* fell from 181 to 93. Although this boundary remained difficult, the class was no longer close to complete collapse into a neighbouring species. In the same comparison, *Hypochaeris radicata* also strengthened substantially, with confusion into *Sonchus arvensis* falling from 100 to 43. This is the strongest evidence in the chapter that augmentation could help the hardest species when paired with a stronger backbone.

Taken together, the augmentation results show that a broader training distribution could be beneficial, but the effect depended strongly on model family. Its main value was as a targeted adjustment of class-wise decision boundaries, and its success depended on whether the underlying representation was strong enough to convert the added within-class variation into improved separation of the most difficult species.

5.3 Effect of imputation and size-related information

This part of the study examines whether performance could be improved through alternative handling of the size variables. The main questions were whether missing size measurements should be imputed, and whether the derived ratio feature added useful information beyond the original two size variables. Overall, the results show that the treatment of missing size values mattered more than the ratio feature itself. Across the models considered here, the size-related modifications mainly changed *which* species were classified correctly, rather than producing a broad and stable improvement across the dataset. This pattern is visible in the full per-species recall overview and in the full set of confusion matrices; see Figure 5.2 and Figure 5.1. The most informative comparison is R256-Std versus R256-Boot, because it gives the clearest view of the difference between mean imputation and bootstrap imputation in the stronger ResNet18 family. The comparison R512-Aug870 versus R512-Boot-Aug870 supports the same conclusion in the 512-dimensional family, while R512-Boot-Aug870 versus R512-Boot-Aug870-Rat shows that the ratio feature likewise altered the class-wise balance without resolving the main difficulty.

A first indication of the effect of *mean imputation* appears in the comparison between the baseline model R512-Base and R512-Std. This comparison does not isolate mean imputation alone, since R512-Std also introduced a validation split, so the change must be interpreted cautiously. Even so, it is informative. Relative to R512-Base, R512-Std gave slightly lower overall accuracy but higher macro recall, suggesting a somewhat more balanced classifier, possibly due to reduced overfitting under validation-based model selection. Among the more difficult species, the most notable gain was for *Bellis perennis*, which increased from 50 correctly classified observations (35.2%) to 99 (69.7%). *Hypochaeris radicata* also improved, from 101 (50.5%) to 118 (59.0%). At the same time, *Sonchus arvensis* weakened substantially, from 306 correct classifications (91.6%) to 222 (66.5%), and *Crepis capillaris* remained unresolved with 0 correct classifications in both models.

The clearest comparison between *mean imputation* and *bootstrap imputation* is R256-Std versus R256-Boot. Here, replacing mean imputation with bootstrap imputation weakened a previously stronger model. The most striking change was for *Hypochaeris radicata*, which fell from 183 correctly classified observations (91.5%) to 50 (25.0%). *Sonchus arvensis* also deteriorated, from 285 (85.3%) to 231 (69.2%). In contrast, *Bellis perennis* improved from 121 (85.2%) to 132 (93.0%), showing that bootstrap imputation did not uniformly reduce performance. The main difficulty nevertheless remained unresolved: *Crepis capillaris* had 0 correct classifications under R256-Std and only 3 (1.5%) under R256-Boot. In this family, mean imputation was therefore clearly more reliable than bootstrap imputation.

The comparison between *mean imputation* and *bootstrap imputation* in the 512-dimensional family, R512-Aug870 versus R512-Boot-Aug870, supports the same conclusion for augmented models. Here the most interesting point is the instability of the trade-off. Under bootstrap imputation, *Sonchus arvensis* improved markedly from 190 correctly classified observations (56.9%) to 302 (90.4%), and *Bellis perennis* also improved from 125 (88.0%) to 136 (95.8%). However, this came at the cost of a sharp deterioration for *Hypochaeris radicata*, which fell from 134 (67.0%) to 52 (26.0%), while *Crepis capillaris* remained unresolved with 0 correct classifications in both models.

The comparison between *bootstrap imputation without* and *with the ratio feature*, R512-Boot-Aug870 versus R512-Boot-Aug870-Rat, points in the same direction. The most notable change is that *Hypochaeris radicata* recovered strongly, increasing from 52 correctly classified observations (26.0%) to 148 (74.0%). However, this was offset by a clear weakening of *Bellis perennis*, which fell from 136 (95.8%) to 86 (60.6%). At the same time, *Crepis capillaris* remained essentially unresolved, improving only from 0 to 1 correctly classified observation (0.5%). The ratio feature therefore changed the class-wise balance, but it did not produce a convincing net gain or resolve the dominant class boundary.

Taken together, these comparisons show that stable handling of the original size variables mattered more than further feature engineering. Mean imputation was the most reliable strategy, whereas bootstrap imputation introduced class-wise volatility without resolving the dominant *Crepis capillaris*–*Hypochaeris radicata* confusion. From a modelling perspective, the main advances in this thesis came from the use of simple mean imputation.

5.4 Effect of focal loss

This part of the study examines whether focal loss could improve performance by placing greater emphasis on difficult observations during training. The most informative comparisons are R512-Std versus R512-FL2 in the 512-dimensional ResNet18 family, and R256-Aug870 versus R256-Aug870-FL1 in the 256-dimensional family. The remaining focal-loss variants support the same conclusion and are visible in the full set of confusion matrices and per-species recall figures, but are not discussed further here.

Across both model families, focal loss does not provide a robust improvement over cross-entropy. Instead, its effect is at best understood as a redistribution of errors across species. This is evident in Figure 5.2 and Figure 5.1, where gains for some classes are offset by losses for others, while the most difficult class boundaries remain largely unchanged.

A first comparison is R512-Std versus R512-FL2. Here, focal loss does not change the overall structure of the classifier in a meaningful way. For example, *Sonchus arvensis* strengthens from 222 to 277 correctly classified observations, while *Bellis perennis* remains essentially unchanged (99 to 100). At the same time, *Hypochaeris radicata* does not improve (118 to 116), and the most difficult class, *Crepis capillaris*, has no correct classifications in either model.

The same conclusion is even clearer in the comparison between R256-Aug870 and R256-Aug870-FL1. This comparison is particularly informative because it evaluates focal loss in a setting where class balance is already relatively strong. Here, focal loss again shifts performance between species rather than improving overall scores on the hardest cases. For instance, *Sonchus arvensis* increases slightly (293 to 301 correctly classified observations), and *Bellis perennis* remains high (135 to 134), but *Hypochaeris radicata* decreases markedly from 114 to 86. Most importantly, *Crepis capillaris* remains very weak, with only 19 correct classifications under cross-entropy and under focal loss.

Taken together, these results indicate that focal loss does not address the main limitation of the classifier. The evidence suggests that although some species remain difficult under cross-entropy, both training and validation performance are already very high, indicating that the models largely separate the in-distribution training data. The remaining errors

are therefore more likely due to limited generalization to unseen flowers and incomplete class separation, rather than insufficient emphasis on hard training examples. Under this interpretation, focal loss targets the wrong bottleneck: it reweights difficult training observations, but does not resolve the distribution shift between training/validation and test data.

From a statistical perspective, the findings are consistent with the broader pattern of the study. Reweighting the loss may shift the class-wise recall profile, but it does not remove the underlying representational difficulty. The dominant confusions, especially the persistent *Crepis capillaris* boundary, remain.

5.5 Persistent class confusions

Across almost all model families, the same difficult class boundaries reappeared. The most persistent problem was the classification of *Crepis capillaris*. In many of the earlier models, this species had zero recall, and its observations were assigned mainly to neighbouring classes, especially *Hypochaeris radicata*. Across all evaluated models, recall for *Crepis capillaris* ranged from 0.0% to 38.5%. This confirms that the baseline problem was not diffuse error across all species, but a highly structured confusion concentrated on a small number of hard boundaries.

The stronger models improved some of these boundaries, but not uniformly. The 256-dimensional ResNet18 models substantially improved *Bellis perennis* and often also *Hypochaeris radicata*, but they did not fully resolve *Crepis capillaris*. The ConvNeXt-Tiny models changed this pattern. Most importantly, C768-Aug870 achieved the first clearly meaningful improvement for *Crepis capillaris*, raising recall to 38.5%. Even so, the remaining confusion with *Hypochaeris radicata* shows that the boundary was improved rather than solved.

A second persistent confusion involved *Bellis perennis*, which in weaker models was often confused with *Tussilago farfara*. This error was reduced considerably in the stronger ResNet18 variants, especially after the move to the 256-dimensional representation. Across all models, recall for *Bellis perennis* ranged from 4.9% to 95.8%, with substantial variability. This wide range reflects that the class is recoverable but highly sensitive to modelling choices. Although the strongest models achieved near-ceiling performance, the confusion did not disappear entirely and in some later variants re-emerged in modified form.

For *Hypochaeris radicata*, the situation was more complex. Recall varied widely across models, from approximately 25.0% to 74.0%, with a standard deviation of about 15–20 percentage points. In some models, especially those with stronger representations, performance improved substantially, while in others it weakened again, particularly when

improvements were obtained for *Crepis capillaris*. Under the strongest ConvNeXt-Tiny model, the dominant confusion shifted away from *Crepis capillaris* and more often involved *Sonchus arvensis*. This indicates that stronger models did not remove the broader difficulty of *Hypochaeris radicata*, but rather changed the neighbouring boundary on which that difficulty appeared.

Finally, *Sonchus arvensis* remained one of the most stable classes across the study. Its recall typically ranged from about 56.9% to 96.7%, with relatively lower variability compared with the more difficult species. However, even for this class, performance fluctuations were linked to trade-offs with neighbouring species, particularly *Hypochaeris radicata*, reinforcing that the main challenges arise from overlapping class boundaries rather than isolated class difficulty.

Taken together, the class-specific results show that the empirical problem is driven by a small number of biologically and visually similar species pairs. The strongest models reduce these confusions, but do not eliminate them entirely. The observed ranges and variability in recall further support that the remaining errors reflect intrinsic overlap in the feature space rather than isolated model deficiencies.

5.6 Remaining classification errors in the best ConvNeXt-Tiny model

To better understand the remaining errors under the best-performing ConvNeXt-Tiny model, C768-Aug870, the predicted class probabilities obtained from the softmax layer were examined for selected true classes together with t-SNE visualizations of the learned fused feature representations. The t-SNE input was the vector

$$z_i = (\phi_\theta(X_i), S_i),$$

where $(\phi_\theta(X_i) \in \mathbb{R}^{768})$ is the ConvNeXt-Tiny image embedding after global average pooling and flattening, and $(S_i = (M_i, m_i))$ contains the two size features. Thus, each point in the t-SNE plot represents the 770-dimensional fused image-and-size representation before the MLP classifier head and before the final softmax transformation. The softmax probabilities were used separately to inspect the model’s prediction uncertainty and confusion patterns.

The analysis focuses on species for which the model still showed reduced recall, especially *Crepis capillaris*, *Hypochaeris radicata*, and *Bellis perennis*. Since the test set consists of held-out flowers, the aim is not only to identify which species were confused, but also to assess whether the remaining errors are associated with specific flower-level clusters.

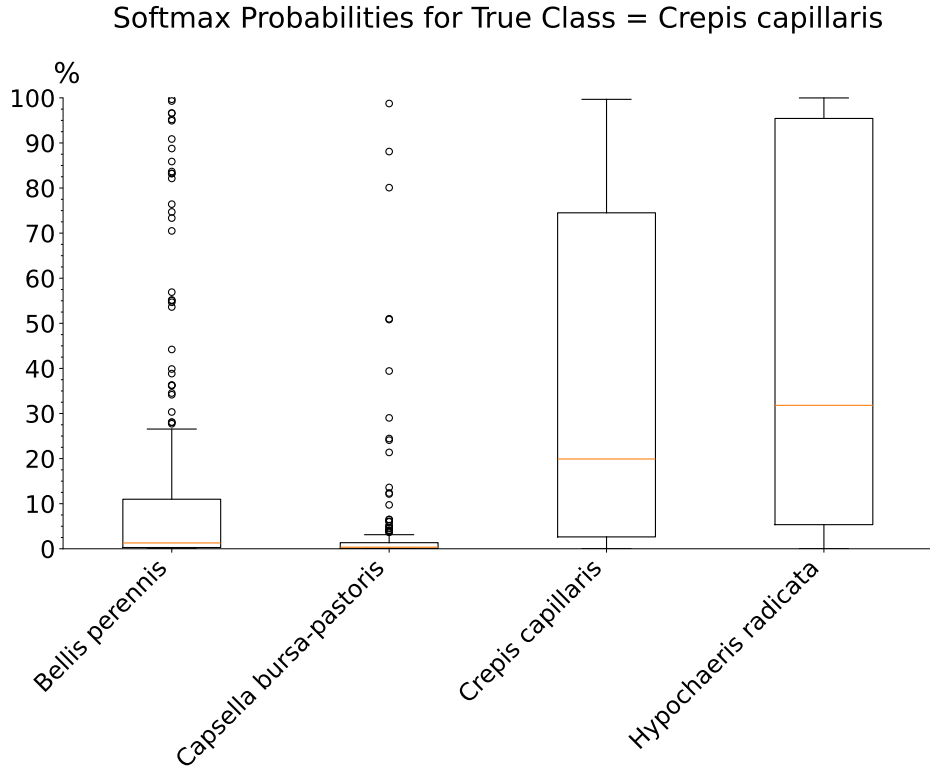


Figure 5.3: Boxplot of predicted softmax probabilities under C768-Aug870 for observations whose true class is *Crepis capillaris*. Only classes with at least one predicted probability above 5% are shown to emphasize the main competing classes.

The most difficult class remained *Crepis capillaris*. Under C768-Aug870, its recall was 38.5%, corresponding to 77 correct classifications out of 200 test observations, while 93 observations were assigned to *Hypochaeris radicata* in the confusion matrix, shown in figure 5.2 and figure 5.1 respectively. The probability boxplot for true *Crepis capillaris* in Figure 5.3 supports this interpretation. The predicted probabilities are concentrated mainly on *Crepis capillaris* itself and *Hypochaeris radicata*, and the two distributions are both broad. For the true class *Crepis capillaris*, the median appears to lie around 0.25, with the first quartile close to 0.05 and the third quartile around 0.7. For *Hypochaeris radicata*, the distribution is similarly wide and even slightly higher in the centre, with a median around 0.35 and an upper quartile close to 0.9. Thus, for a substantial fraction of *Crepis capillaris* observations, the model assigns comparable or even greater probability to *Hypochaeris radicata* than to the true class. By contrast, *Bellis perennis* and *Capsella bursa-pastoris* have much lower central probability mass, although both show a few large outliers. Hence, the residual uncertainty for *Crepis capillaris* is not spread across many species, but concentrated on a small set of nearby alternatives, above all *Hypochaeris radicata*.

For *Bellis perennis*, the probability structure is more favourable, but still shows a clear competing class. Under C768-Aug870, recall for this species was 70.4%, corresponding to

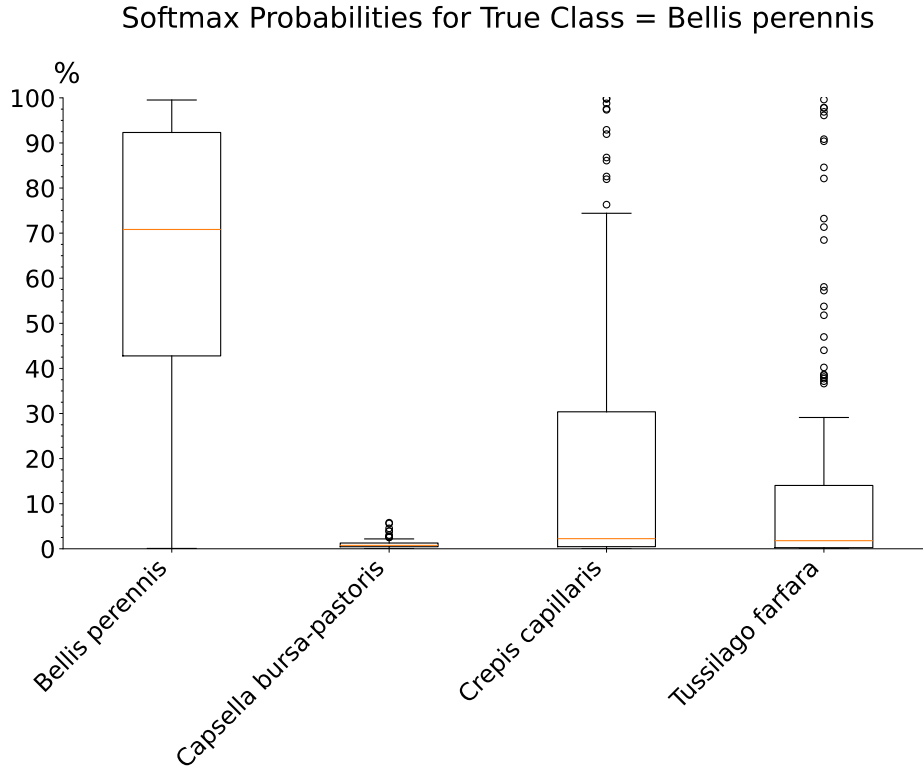


Figure 5.4: Boxplot of predicted softmax probabilities under C768-Aug870 for observations whose true class is *Bellis perennis*. Only classes with at least one predicted probability above 5% are shown to emphasize the main competing classes.

100 correct classifications out of 142 test observations. The confusion matrix shows that the main remaining errors were 26 assignments to *Crepis capillaris* and 16 to *Tussilago farfara*. In the corresponding probability boxplot in Figure 5.4, the true class *Bellis perennis* has a clearly higher centre than the competing classes, with a median around 0.75 and an upper range extending close to 1.0. However, *Tussilago farfara* shows a cluster of high outliers, some approaching probability one, indicating that a subset of *Bellis perennis* observations is still judged very similarly to that species. The distribution for *Crepis capillaris* is concentrated much closer to zero for most observations, but with a moderate upper tail. This suggests that the remaining classification problem for *Bellis perennis* is best described as partial overlap with a specific neighbour rather than a general lack of separation.

For *Hypochaeris radicata*, the probability structure lies between these two cases. Its recall under C768-Aug870 was 65.5%, corresponding to 131 correct classifications out of 200 test observations, with the main confusion being 43 assignments to *Sonchus arvensis*. The boxplot for true *Hypochaeris radicata* in Figure 5.5 shows that the probability assigned to the correct class is on average clearly higher than for most competing classes, with a median around 0.6 and a broad interquartile range that extends well above 0.8. However, *Sonchus arvensis* also receives non-negligible probability mass, with a long upper tail and

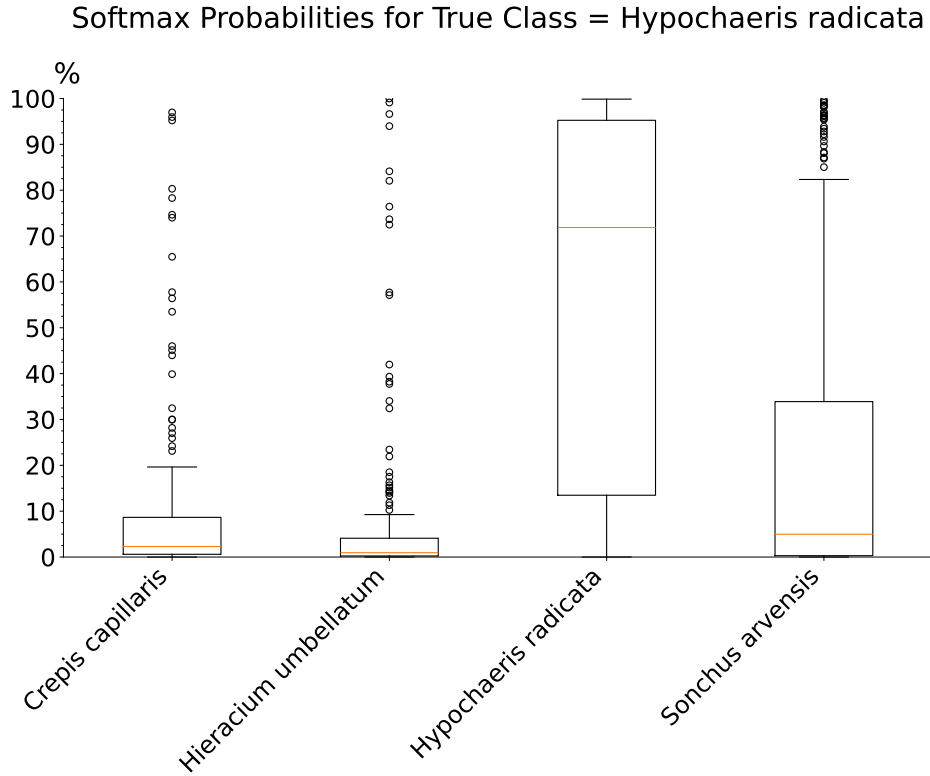


Figure 5.5: Boxplot of predicted softmax probabilities under C768-Aug870 for observations whose true class is *Hypochaeris radicata*. Only classes with at least one predicted probability above 5% are shown to emphasize the main competing classes.

many moderate-to-large values. By contrast, the distributions for *Crepis capillaris* and *Hieracium umbellatum* are much more concentrated near zero, with only scattered high outliers. This is consistent with the confusion matrix: the main unresolved boundary for *Hypochaeris radicata* is not with *Crepis capillaris*, but with *Sonchus arvensis*.

Because the data split was performed at flower level, the test observations represent flowers that were not seen during training. The embedding visualizations were therefore interpreted with this structure in mind. In the t-SNE plots, marker shape indicates species and colour indicates flower ID. The plots are used as exploratory visualizations of local neighbourhood structure, not as formal evidence of separability. Accordingly, the main question is whether held-out test flowers form compact clusters that remain close to their own species, or whether they are embedded near flowers from competing species.

For *Crepis capillaris* and *Hypochaeris radicata*, Figure 5.6 shows a clear flower-level structure. In the training projection, the two species form well-separated clusters, and the colouring by flower ID shows that observations from the same flower occupy compact and distinct local subclusters within each species. This indicates that the model learns a representation that separates not only species-level structure, but also within-species flower-level variation.

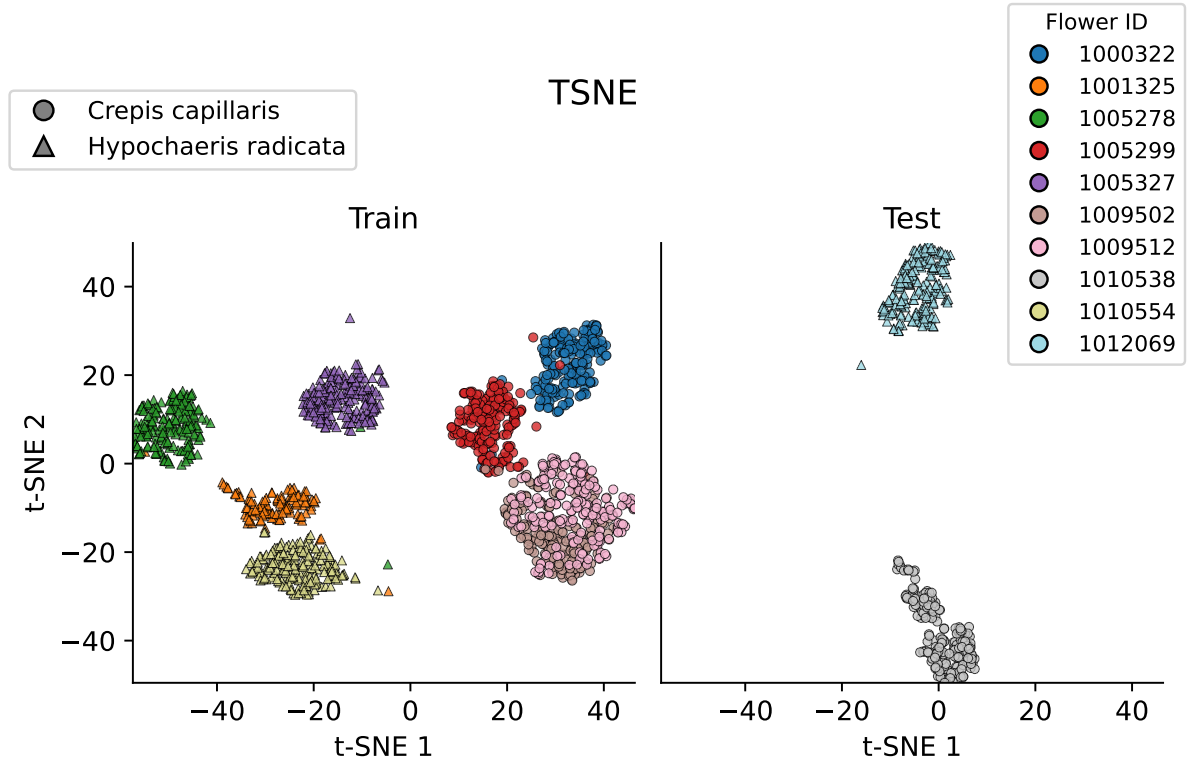


Figure 5.6: t-SNE projection for classes *Crepis capillaris* and *Hypochaeris radicata* under C768-Aug870.

The test projection shows the same within-flower coherence: the held-out flowers form compact clusters rather than being diffusely scattered. However, their relative positions in the embedding are not as clearly aligned with the training separation. This suggests that the learned representation generalizes less consistently at the flower level. In other words, flowers remain internally coherent, but some held-out flowers are embedded in regions that are closer to competing species. A separation that appears reliable among the training flowers does not necessarily transfer to unseen flowers. This motivates examining the same species together with additional neighbouring classes from the confusion matrix.

When *Sonchus arvensis* is included, the structure becomes more nuanced; see Figure 5.7. In the training set, all three species form clearly separated clusters, again with distinct flower-level subclusters. In the test set, *Crepis capillaris* remains relatively distinct, while *Hypochaeris radicata* lies between *Crepis capillaris* and *Sonchus arvensis*. The flower-level coloring shows that this effect is driven by the positions of test flowers: some *Hypochaeris radicata* flowers lie closer to *Sonchus arvensis*, leading to consistent misclassification for those specific biological instances. This aligns with the confusion matrix and probability analysis, where the dominant remaining confusion for *Hypochaeris radicata* is with *Sonchus arvensis*, while *Crepis capillaris* remains mainly challenged by a narrower set of neighbouring classes.

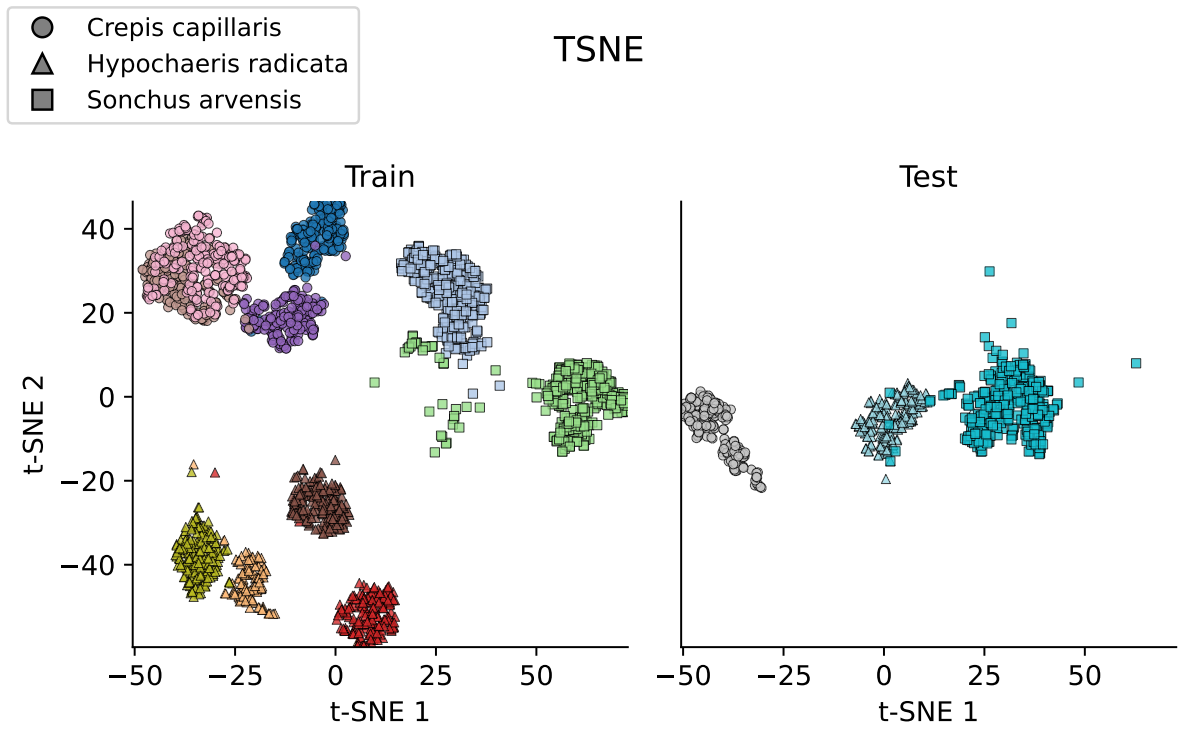


Figure 5.7: t-SNE projection for *Crepis capillaris*, *Hypochaeris radicata*, and *Sonchus arvensis* under C768-Aug870, with points colored by flower ID.

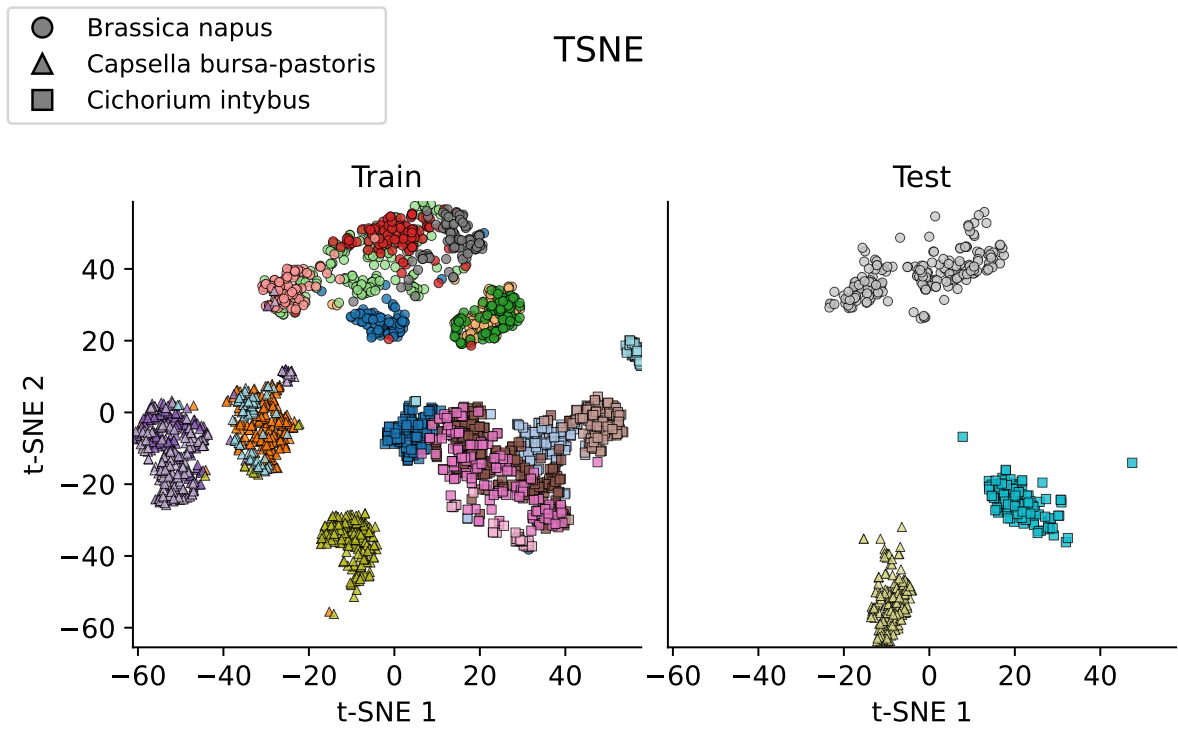


Figure 5.8: t-SNE projection for *Brassica napus*, *Capsella bursa-pastoris*, and *Cichorium intybus* under C768-Aug870, with points colored by flower ID.

Sensitivity Analysis by Flower-Split Case: recall heatmap

Brassica napus	99.0	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Tussilago farfara	100	100	100	100	100	100	100	100	100	100	100	100	99.4	100	100
Cichorium intybus	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Capsella bursa-pastoris	100	99.5	97.8	100	100	100	98.9	100	100	95.6	100	100	100	93.4	84.7
Tragopogon pratensis	99.4	98.8	99.4	100	90.4	98.8	98.2	98.8	99.4	99.4	100	95.2	100	98.8	99.4
Hieracium umbellatum	98.5	96.5	91.5	92.5	93.0	94.0	95.5	94.0	93.0	92.5	85.0	78.0	94.5	97.5	92.0
Sonchus arvensis	96.4	94.0	92.5	80.2	95.8	87.4	95.2	91.6	86.5	57.8	93.7	40.1	81.4	74.6	94.9
Hypochaeris radicata	0.0	3.5	35.0	78.0	1.0	13.0	32.0	81.5	0.5	55.0	65.5	96.0	11.0	21.0	74.0
Bellis perennis	91.5	65.5	78.2	92.3	75.4	85.2	69.0	78.2	50.7	52.1	70.4	50.0	97.2	96.5	44.4
Crepis capillaris	100	100	95.0	100	100	100	100	99.0	11.0	20.0	38.5	14.5	83.5	40.5	26.5
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Case														

Figure 5.9: Species-wise recall across the 16 sensitivity-analysis cases; Case 11 is the original-split C768-Aug870 result.

By contrast, well-classified species such as *Brassica napus*, *Capsella bursa-pastoris*, and *Cichorium intybus* show the same separation in both training and test sets; see Figure 5.8. Here, flowers are more overlapping, creating larger species-level clusters that are well separated from other species. This provides a useful benchmark for the difficult-species analysis: for these classes, the learned representation is sufficiently stable that a separation visible in the training set remains valid in the test set.

Taken together, the embedding visualizations show that the remaining errors of C768-Aug870 are not caused by diffuse overlap across all classes. They are concentrated in local regions where flower-level variation creates unstable train-test transfer. The stronger the separation between flowers within a species, the larger this risk becomes: a held-out flower may be shifted away from the training flowers of the same species and closer to a competing class. This is most visible for *Crepis capillaris*, where the flowers are well separated. By contrast, *Brassica napus* also shows flower-level clustering, but without the same clear separation between flower clusters, reducing the risk that the held-out flower falls outside the training distribution.

5.7 Robustness and sensitivity to the flower-level split

The species-wise recall patterns across the 16 cases in Figure 5.9 show where split-dependent instability in C768-Aug870 is concentrated. This makes it possible to compare the effect of alternative flower-level splits on recall.

Several species remained highly stable across the flower-level test combinations. In particular, *Brassica napus*, *Tussilago farfara*, *Cichorium intybus*, *Capsella bursa-pastoris*, and *Tragopogon pratensis* were all very stable, with mean recalls above 98% and negligible or modest variation across cases.

A second group of species exhibited moderate but still limited sensitivity to the test-flower choice. *Hieracium umbellatum* retained a high mean recall of 93.4%, although with some variability. *Sonchus arvensis* showed more variability, with mean recall 82.4% and a standard deviation of 16.1 percentage points. *Bellis perennis* displayed a similar pattern, with mean recall 76.1% and a standard deviation of 18.2 percentage points.

The strongest split-dependence remained concentrated in the two species that dominated the error structure throughout the study. *Crepis capillaris* had mean recall 65.91%, but with substantial variability, ranging from 11.0% to 100.0% and a standard deviation of 36.81 percentage points. *Hypochaeris radicata* showed similarly pronounced sensitivity, with mean recall 40.38%, a range from 0.0% to 96.0%, and a standard deviation of 33.38 percentage points. Thus, the performance for these species depends strongly on the choice of test set.

The favorable and unfavorable confusion matrices in Figure 5.10 provide a concrete illustration of how the error structure changes under different flower-level splits. The favorable case is Case 8, while the unfavorable case is Case 9.

In the favorable case, the final model achieved near-complete separation even for the most difficult species. In particular, *Crepis capillaris* reaches a recall close to perfect, and *Hypochaeris radicata* forms a much stronger diagonal with substantially reduced confusion into neighbouring classes. The remaining species, already well separated in earlier analyses, remain close to perfectly classified. This indicates that the modelling pipeline has sufficient representational capacity to separate even the most challenging class boundaries under favorable training/test split.

In the unfavorable case, the main deterioration is again concentrated in a few difficult species rather than spread across all species. *Crepis capillaris* is frequently misclassified as *Hypochaeris radicata*, at the same time, *Hypochaeris radicata* shows confusion not only into *Crepis capillaris* but also into other neighbouring species.

This difference is clarified by the corresponding t-SNE projections in Figures 5.11 and 5.12.

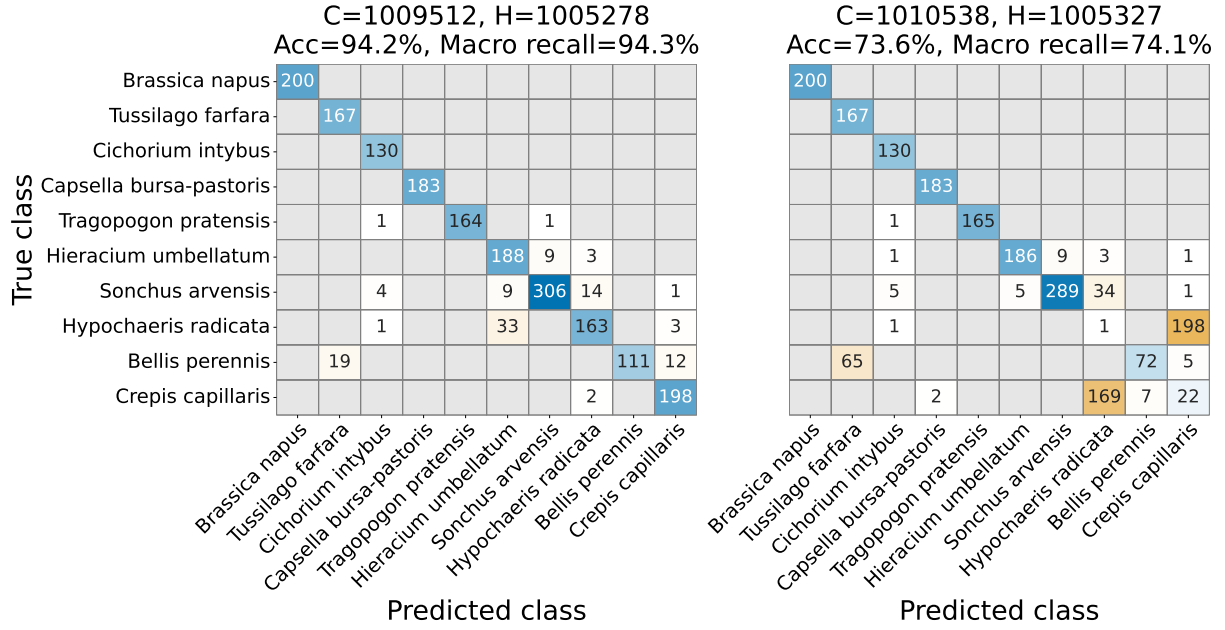


Figure 5.10: Confusion matrices for the most favorable sensitivity-analysis case, Case 8 (left), and the least favorable case, Case 9 (right). C denotes the held-out *Crepis capillaris* flower and H denotes the held-out *Hypochaeris radicata* flower.

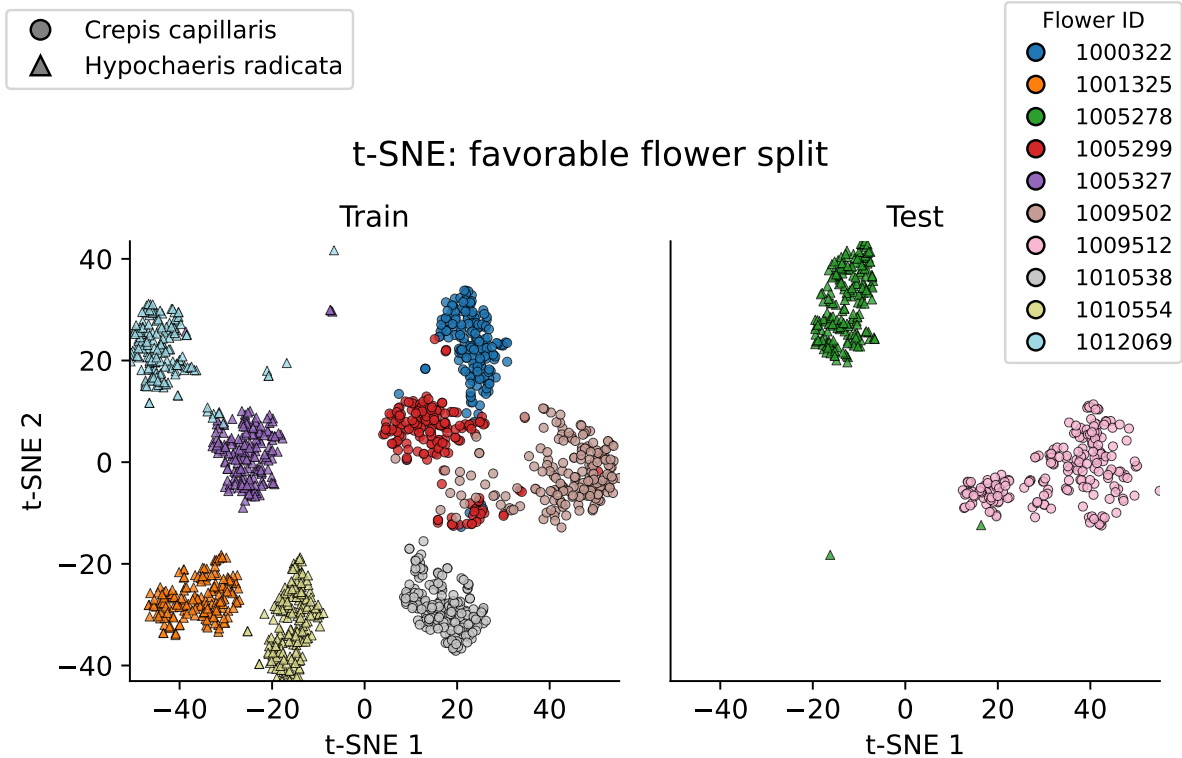


Figure 5.11: t-SNE projection of learned features for *Crepis capillaris* and *Hypochaeris radicata* under a favorable sensitivity-analysis case, Case 8 ($C = 1009512$, $H = 1005278$) using C768-Aug870. Points are colored by flower ID and marker shape indicates species.

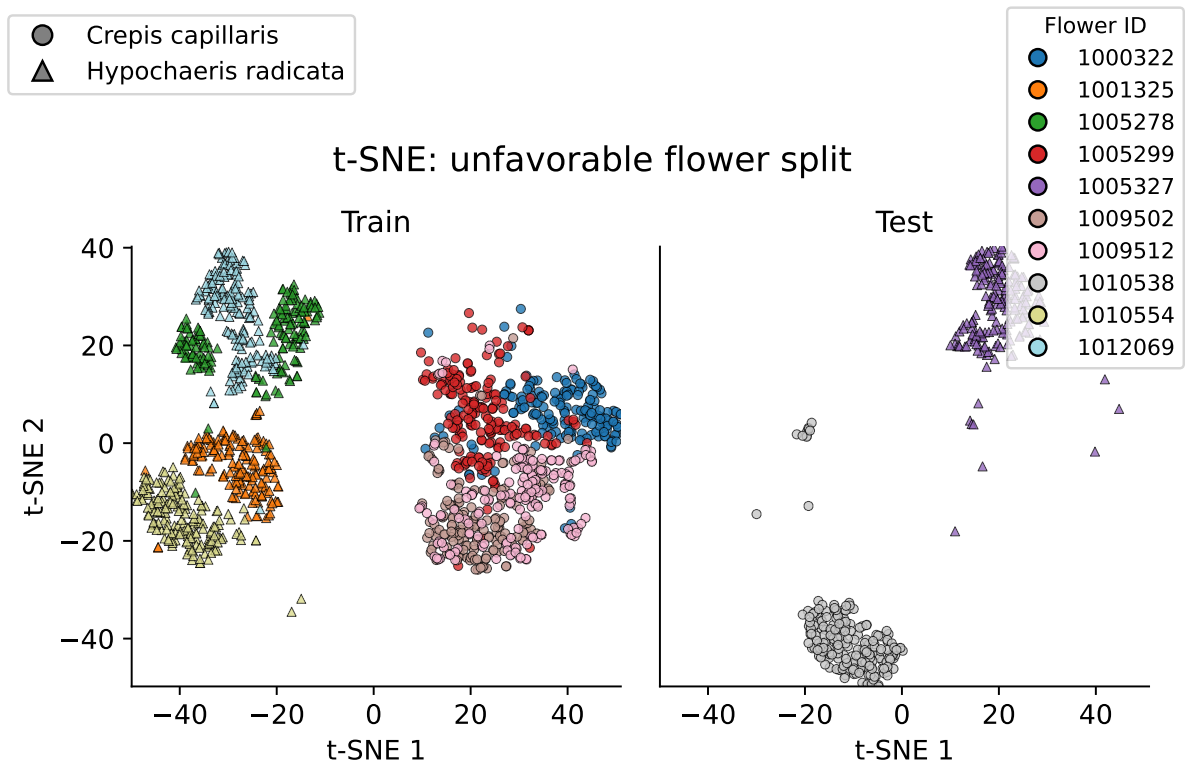


Figure 5.12: t-SNE projection of learned features for *Crepis capillaris* and *Hypochaeris radicata* under an unfavorable sensitivity-analysis case, Case 9 ($C = 1010538$, $H = 1005327$) using C768-Aug870. Points are colored by flower ID and marker shape indicates species.

In the favorable case, the test samples form compact clusters that are well separated between species and consistent with the structure and separation observed in the training set. However, we still see the clear separation into distinct flowers, especially for *Crepis capillaris*.

In the unfavorable case, the structure changes qualitatively. While both training and test samples still form compact, flower-specific clusters, the test clusters are displaced relative to the training clusters and are no longer aligned with the training distribution. In particular, the test clusters for *Crepis capillaris* and *Hypochaeris radicata* are shifted into regions of the feature space associated with the other species, placing them near or across the decision boundaries. This mismatch between training and test distributions explains the increased confusion observed in the confusion matrix.

The observed change is not driven by increased dispersion within flowers, but by shifts in the relative positioning of the flower-level clusters. Each flower remains internally coherent in the learned representation, yet its location in feature space varies across splits.

Thus, the sensitivity analysis suggests that the strongest model in the thesis has sufficient capacity to resolve even the hardest class boundaries under favorable conditions, but that these boundaries remain highly sensitive to variability between flowers. The remaining errors are thus best understood as reflecting incomplete robustness of a few neighbouring class boundaries.

All remaining confusion matrices for the 16 sensitivity-analysis cases are provided in Section B.2, so that the reader can inspect the full range of split-specific error structures without interrupting the main narrative.

6 Conclusion

This thesis investigated whether pollen-species classification could be improved by extending the image-and-size based deep-learning approach of Jurdell (2025) under the same dataset, species selection, and flower-level train-test split. The answer to the main research question is yes: the best model, C768-Aug870, improved the reproduced ResNet18 baseline from 76.8% to 85.2% test accuracy and from 76.2% to 85.3% macro recall. The improvement shows that classification performance can be improved within the same experimental setting, but also that aggregate accuracy alone is not sufficient for evaluating pollen classifiers.

The experiments further showed which modelling choices were most important. More complex missing-data handling, the additional ratio feature (m_i/M_i), and focal loss did not provide consistent improvements. In contrast, the main gains came from changing the learned image representation and using moderate class-balancing augmentation. The reduced ResNet18 representation improved performance compared with the original 512-dimensional representation, suggesting that the 256-dimensional projection may have provided a better balance between the image representation and the two size features before classification. The ConvNeXt-Tiny model gave the strongest overall result. The trainable-parameter counts help interpret these results: both R256 and C768 contain more trainable parameters than R512, so the improvement may partly reflect increased model capacity. However, the gains cannot be attributed to parameter count alone, since R256 also changes the representation through a projection bottleneck, while C768 changes the backbone, pretrained feature representation, and classifier head. The results therefore suggest that representation quality, model capacity, the relative balance between image and size information, and controlled augmentation all contributed to improved performance.

The study also answered which species remained difficult. Several species were classified with consistently high recall, but *Crepis capillaris*, *Bellis perennis*, and *Hypochaeris radicata* formed the main model-specific failure cases. This does not show that these species would necessarily be equally difficult for a human pollen expert, since no blinded expert comparison was performed under the same test conditions. It shows that they were difficult for the models in this dataset. The best model reduced the most severe failure case by increasing the recall of *Crepis capillaris* from 0% to 38.5%, but it did not eliminate the confusion between visually similar species.

The cluster analysis in Section 5.6, based on the learned representations from the best-performing C768-Aug870 architecture, supported this interpretation. Species that were

classified reliably tended to show clearer separation in the learned feature representation, whereas the difficult *Crepis capillaris*–*Hypochaeris radicata* cases remained close or overlapping. The sensitivity analysis used the same C768-Aug870 model structure across 16 alternative flower-level splits for these two species. Across these splits, the recalls of *Crepis capillaris* and *Hypochaeris radicata* varied substantially, while the other species were comparatively stable. This suggests that the remaining uncertainty comes both from limited separation between difficult species and from flower-level variation within species.

The practical implication is that pollen classifiers should be used as decision-support tools rather than complete replacements for expert identification. High-confidence predictions for well-separated species may reduce the amount of manual inspection, while low-confidence predictions and predictions involving repeatedly confused species should be prioritized for expert review.

Future work should focus on the species pairs that remain difficult, especially *Crepis capillaris* and *Hypochaeris radicata*. Possible directions include collecting more flowers for these species, using repeated flower-level cross-validation rather than a single fixed split, selecting validation data at the flower level, and evaluating representations that are explicitly designed to separate visually similar pollen types. More detailed modelling of flower-level dependence may also be useful, since the sensitivity analysis showed that the choice of held-out flower can strongly affect the apparent difficulty of selected species. These extensions would provide a more robust assessment of model generalization and help determine if errors are primarily due to limited data, biological overlap, or the learned image representation.

Code and data availability

The code and experiment material used in this thesis are archived on Zenodo and are available through the DOI `10.5281/zenodo.20187141` (Truong 2026). The corresponding GitHub repository is available **here**.

The archived release corresponds to version `v1.0.0` of the repository. The repository contains the Python/PyTorch implementation, model-training notebooks, evaluation scripts, plotting utilities, experiment summaries, and selected generated figures used in the thesis.

The pollen image dataset itself is not redistributed in the repository, since it may be subject to separate access restrictions. The repository is therefore intended to support reproducibility of the code, model structure, preprocessing pipeline, and analysis workflow, while access to the original dataset must be handled separately. Unless otherwise stated, the source code is released under the MIT License.

Bibliography

- Beil, Marion, Helmut Horn, and Angelika Schwabe (2008). “Analysis of pollen loads in a wild bee community (Hymenoptera: Apidae) — a method for elucidating habitat use and foraging distances”. In: *Apidologie* 39.4, pp. 456–467. DOI: 10.1051/apido:2008021.
- Deng, Jia et al. (2009). “ImageNet: A Large-Scale Hierarchical Image Database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.
- He, Kaiming et al. (2016). “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. DOI: 10.1109/CVPR.2016.90.
- Jurdell, Erik (2025). “Can the accuracy of deep learning pollen analysis be improved by implementing both image data and pollen size”. Bachelor’s thesis. Lund University.
- Kingma, Diederik P. and Jimmy Ba (2015). “Adam: A Method for Stochastic Optimization”. In: *International Conference on Learning Representations (ICLR)*. arXiv: 1412.6980 [cs.LG].
- LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). “Deep learning”. In: *Nature* 521.7553, pp. 436–444. DOI: 10.1038/nature14539.
- Lin, Tsung-Yi et al. (2017). “Focal Loss for Dense Object Detection”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988. DOI: 10.1109/ICCV.2017.324.
- Liu, Zhuang et al. (2022). “A ConvNet for the 2020s”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11976–11986. DOI: 10.1109/CVPR52688.2022.01167.
- Maaten, Laurens van der and Geoffrey Hinton (2008). “Visualizing Data using t-SNE”. In: *Journal of Machine Learning Research* 9, pp. 2579–2605.
- Noord, Nanne van and Eric Postma (2017). “Learning scale-variant and scale-invariant features for deep image classification”. In: *Pattern Recognition* 61, pp. 583–592. DOI: 10.1016/j.patcog.2016.06.005.
- Olsson, Ola et al. (2021). “Efficient, automated and robust pollen analysis using deep learning”. In: *Methods in Ecology and Evolution* 12.5, pp. 850–862. DOI: 10.1111/2041-210X.13575.
- Paszke, Adam et al. (2019). “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems*. Vol. 32. DOI: 10.48550/arXiv.1912.01703.

- Pigott, Therese D. (2001). “A Review of Methods for Missing Data”. In: *Educational Research and Evaluation* 7.4, pp. 353–383. DOI: 10.1076/edre.7.4.353.8937.
- Powers, David M. W. (2011). “Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation”. In: *Journal of Machine Learning Technologies* 2.1, pp. 37–63.
- Prechelt, Lutz (1998). “Early Stopping – But When?” In: *Neural Networks: Tricks of the Trade*. Ed. by Genevieve B. Orr and Klaus-Robert Müller. Berlin, Heidelberg: Springer, pp. 55–69. DOI: 10.1007/3-540-49430-8_3.
- PyTorch Contributors (2026). *PyTorch tensor documentation: numel and requires_grad*. <https://docs.pytorch.org/docs/stable/>. Accessed 2026-06-13.
- Srivastava, Nitish et al. (2014). “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56, pp. 1929–1958. URL: <https://jmlr.org/papers/v15/srivastava14a.html>.
- TorchVision Contributors (2026a). *ConvNeXt Tiny – TorchVision model documentation*. https://docs.pytorch.org/vision/main/models/generated/torchvision.models.convnext_tiny.html. Accessed 2026-06-13.
- (2026b). *ResNet18 – TorchVision model documentation*. <https://docs.pytorch.org/vision/main/models/generated/torchvision.models.resnet18.html>. Accessed 2026-06-13.
- Truong, Nancy (2026). *Improving Pollen Species Classification Using Deep Learning and Size Features: Code and Experiment Material*. Version v1.0.0. Code repository archived on Zenodo. DOI: 10.5281/zenodo.xxxxxxx. URL: https://github.com/myngoc-trg/Bachelor_Project_Statistics.
- Wong, Sebastien C. et al. (2016). “Understanding Data Augmentation for Classification: When to Warp?” In: *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–6. DOI: 10.1109/DICTA.2016.7797091.

A Flower IDs Used in Train Dataset

Table A.1: Training-set flower-level image counts and missing size measurements.

Species	Flower ID	Images	Missing size count
<i>Bellis perennis</i>	1001572	143	0
<i>Bellis perennis</i>	1001608	143	2
<i>Bellis perennis</i>	1005368	143	0
<i>Bellis perennis</i>	1005371	143	0
<i>Bellis perennis</i>	1005375	143	0
<i>Bellis perennis</i>	1010522	143	0
<i>Brassica napus</i>	1000406	100	4
<i>Brassica napus</i>	1001635	100	1
<i>Brassica napus</i>	1001647	100	0
<i>Brassica napus</i>	1004452	100	0
<i>Brassica napus</i>	1004453	100	0
<i>Brassica napus</i>	1004454	100	0
<i>Brassica napus</i>	1004455	100	0
<i>Brassica napus</i>	1005373	100	0
<i>Capsella bursa-pastoris</i>	1001617	184	0
<i>Capsella bursa-pastoris</i>	1004520	184	0
<i>Capsella bursa-pastoris</i>	1004521	184	0
<i>Capsella bursa-pastoris</i>	1005389	184	0
<i>Capsella bursa-pastoris</i>	1010525	81	0
<i>Cichorium intybus</i>	1000318	130	0
<i>Cichorium intybus</i>	1001372	130	0
<i>Cichorium intybus</i>	1005059	131	0
<i>Cichorium intybus</i>	1005097	130	0
<i>Cichorium intybus</i>	1005200	130	0
<i>Cichorium intybus</i>	1005235	131	0
<i>Cichorium intybus</i>	1005321	35	0
<i>Cichorium intybus</i>	1005509	53	0
<i>Crepis capillaris</i>	1000322	200	1
<i>Crepis capillaris</i>	1005299	200	0
<i>Crepis capillaris</i>	1009502	200	0
<i>Crepis capillaris</i>	1009512	200	0
<i>Hieracium umbellatum</i>	1001326	200	65
<i>Hieracium umbellatum</i>	1005270	200	0

Continued on next page

Table A.1: Training-set flower-level image counts and missing size measurements, continued.

Species	Flower ID	Images	Missing size count
<i>Hieracium umbellatum</i>	1005289	200	0
<i>Hieracium umbellatum</i>	1012102	200	0
<i>Hypochaeris radicata</i>	1001325	200	83
<i>Hypochaeris radicata</i>	1005278	200	0
<i>Hypochaeris radicata</i>	1005327	200	0
<i>Hypochaeris radicata</i>	1010554	200	0
<i>Sonchus arvensis</i>	1000335	333	0
<i>Sonchus arvensis</i>	1005268	333	0
<i>Tragopogon pratensis</i>	1004991	632	0
<i>Tragopogon pratensis</i>	1010351	2	0
<i>Tragopogon pratensis</i>	1017334	200	0
<i>Tussilago farfara</i>	1000328	167	0
<i>Tussilago farfara</i>	1001625	166	0
<i>Tussilago farfara</i>	1004811	166	0
<i>Tussilago farfara</i>	1004814	167	0
<i>Tussilago farfara</i>	1005387	167	0

B Sensitivity-analysis

B.1 Flower-level test cases

Table B.1: Flower-level test combinations used in the sensitivity analysis for *Crepis capillaris* and *Hypochaeris radicata*.

Case	<i>Crepis capillaris</i> 's flower	<i>Hypochaeris radicata</i> 's flower
1	1009502	1005327
2	1009502	1010554
3	1009502	1012069
4	1009502	1005278
5	1009512	1005327
6	1009512	1010554
7	1009512	1012069
8	1009512	1005278
9	1010538	1005327
10	1010538	1010554
11	1010538	1012069
12	1010538	1005278
13	1005299	1005327
14	1005299	1010554
15	1005299	1012069
16	1005299	1005278

B.2 Confusion Matrices

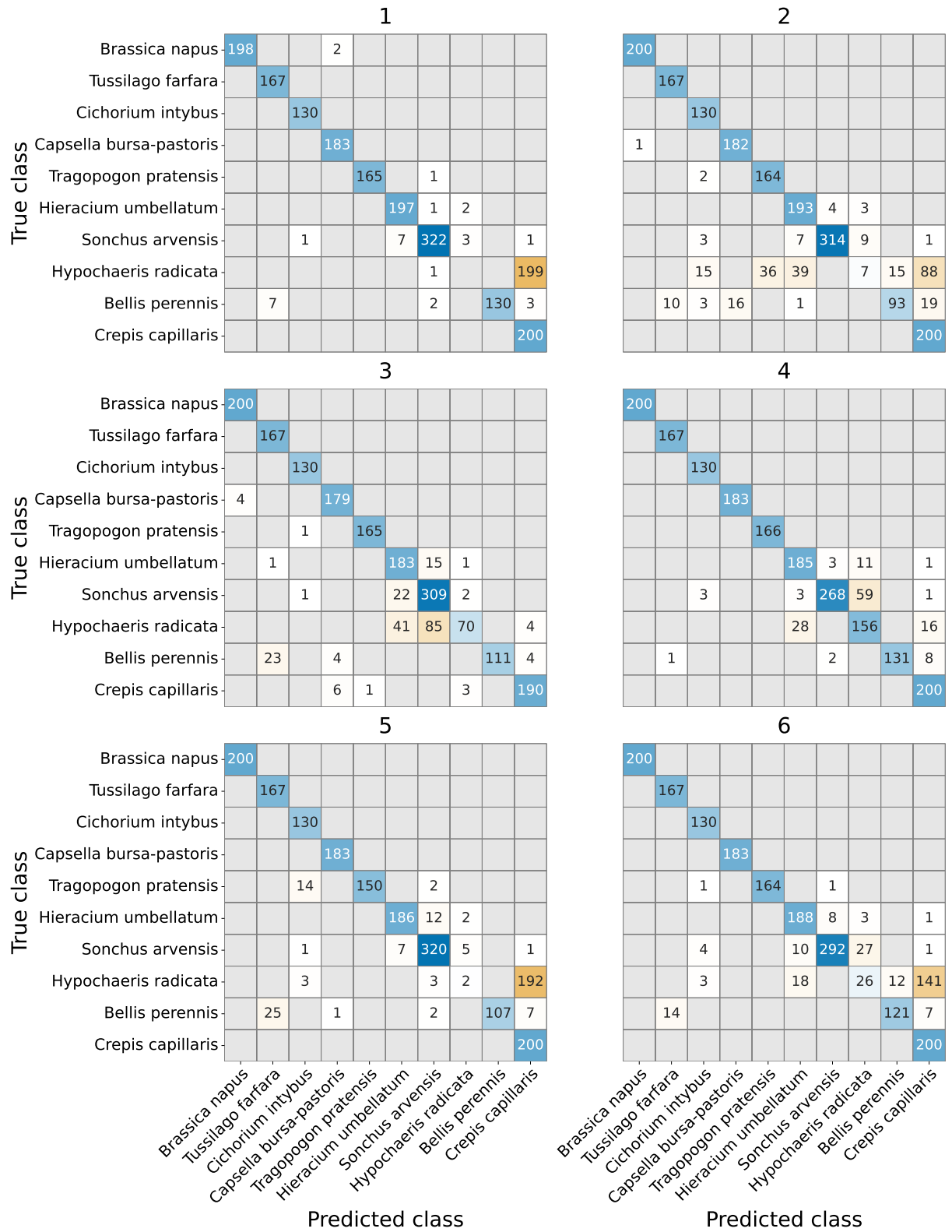


Figure B.1: Confusion matrices for sensitivity-analysis cases 1–6. The class order is fixed across all panels according to the class-wise recall order of R512-Base, from highest to lowest recall.

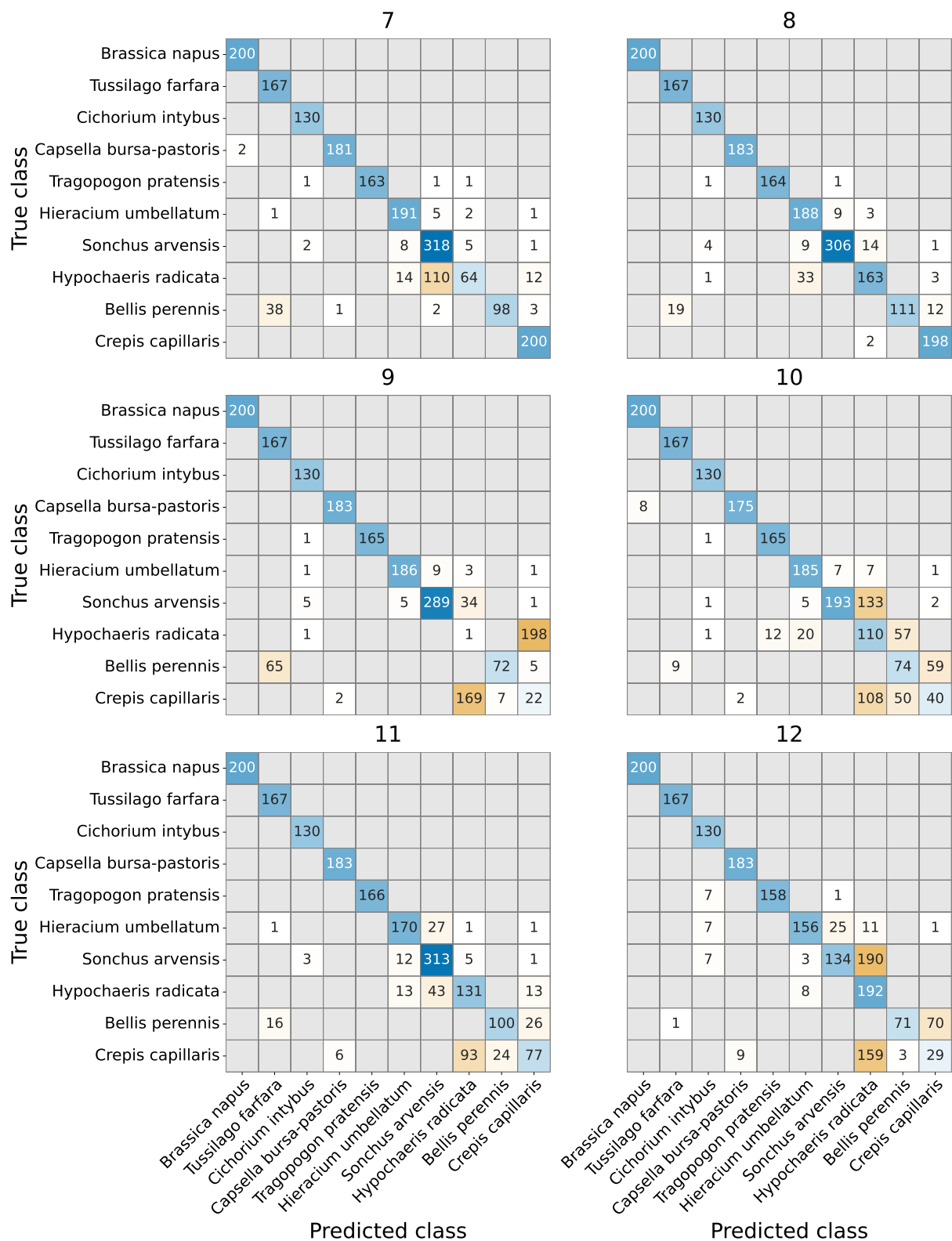


Figure B.1: Confusion matrices for sensitivity-analysis cases 7–12, continued from Figure B.1. The same fixed class order is used in all panels.

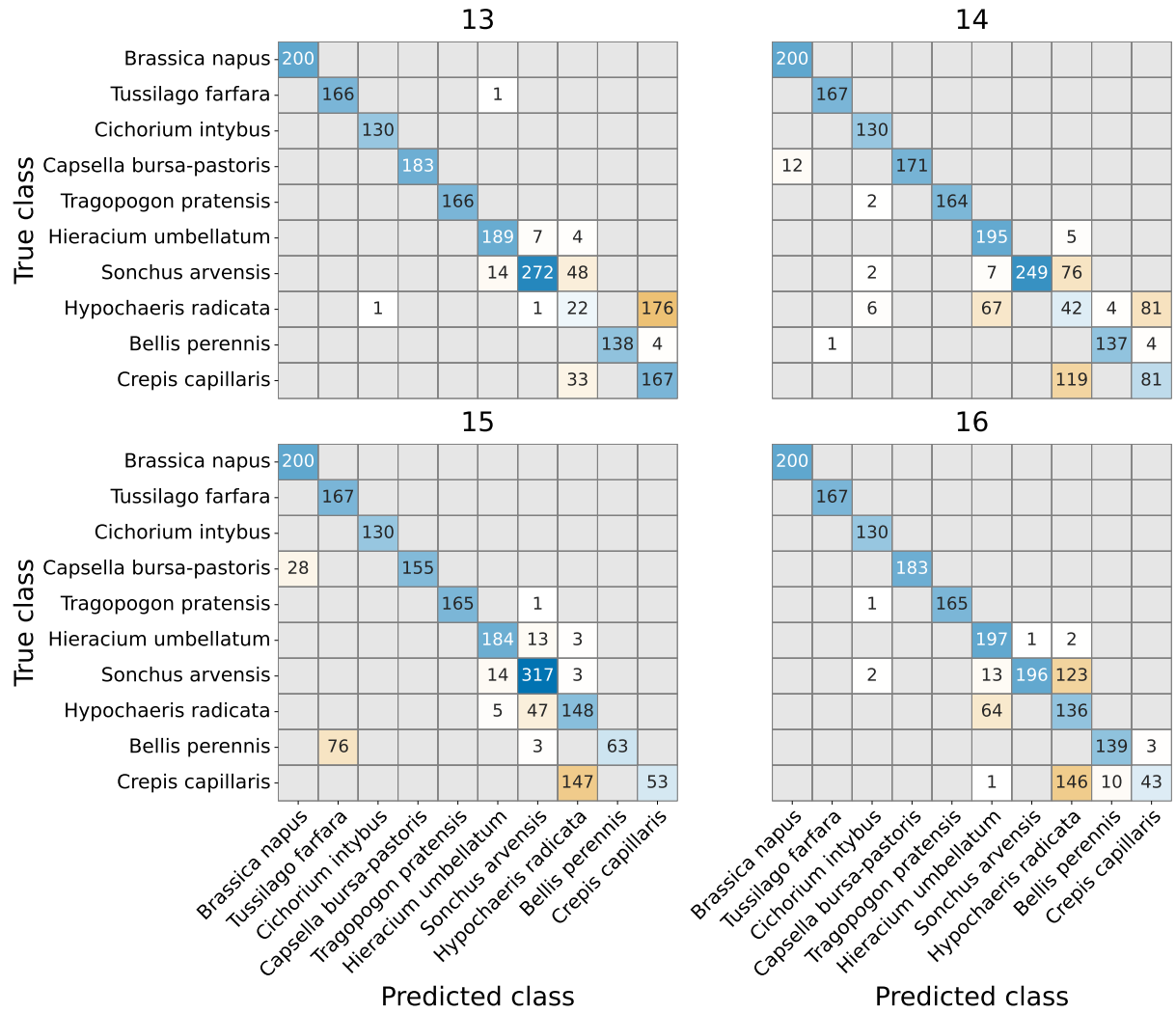


Figure B.1: Confusion matrices for sensitivity-analysis cases 13–16, continued from Figure B.1. The same fixed class order is used in all panels.

Bachelor's Theses in Mathematical Sciences 2026:K25
ISSN 1654-6229
LUNFMS-4090-2026
Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden
<http://www.maths.lu.se/>