Reply to Referee #2

We thank you Referee#2 for her/his constructive comments on our manuscript. Please find below our replies in blue.

1. A ResNet-18 is selected as the main feature extractor; why is that the case? Are smaller nets better than larger, are e.g. Residual Blocks superior to Dense Blocks (DenseNet)? What made the authors chose this architecture?

Both ResNet and DenseNet are standard tools and very alike, hence both would apply here. The choice of ResNet is based on its comparatively lower memory consumption and greater speed [1], especially since the paper is mainly a proof of concept. We anyhow tried to increase the complexity of the ResNet design (i.e. ResNet-34 and ResNeXt versions) but no significant performance increase was observed - at the same time the memory consumption did increase considerably. The network design was therefore a result of the tradeoff between complexity and performance.

[1]https://openaccess.thecvf.com/content/WACV2021/papers/Zhang_ResNet_or_DenseNet_Introducing_Dense_Shortcuts_to_ResNet_WACV_2021_paper.pdf

2. Are the authors dealing with an object classification or detection problem? There is a distinct difference between the two: The first aims at classifying images of isolated particles, while the latter aims at detecting (counting) and classifying a number of particles in an image. I assume it is the first (also due to the images shown in Appendix C), then how are the segmented/isolated particle images generated? By-hand, by the FlowCam software, or a different method?

It is an image classification problem. Every image portraits a single particle. Such single particle images are generated by the FlowCam software, by analyzing the whole camera view and segmenting out the pixels whose luminosity exceeds some threshold (see Sect. 2.1 and Appendix A). We slightly reworded Sect. 2.1 to better clarify the image creation pipeline.

3. The authors mention in line 138 “false positives”. This hints at a typical metric from object detection tasks, typically depicted via a confusion matrix. However, this also includes more important metrics than accuracy (such as F1, precision, etc.). This is confusing (see also point no. 2).

The purpose of the 7th class (Contamination/Blurry) is to identify those particles that do not carry climate significance. If this class were not included and these particles entered the detector, they would be classified as any of the other classes by the model, thus effectively creating a “fake” climate signal, which is highly non optimal especially for rare particles such as tephra. This effect can be best seen from the test-time confusion matrix in Fig. 2: about 1% of contamination particles are erroneously classified as tephra. Obviously, measuring clean samples in the first place is the optimal solution, but adding a contamination class to the model limits this effect to about 1%.

We deleted here the word “false positives” and reworded the sentence:

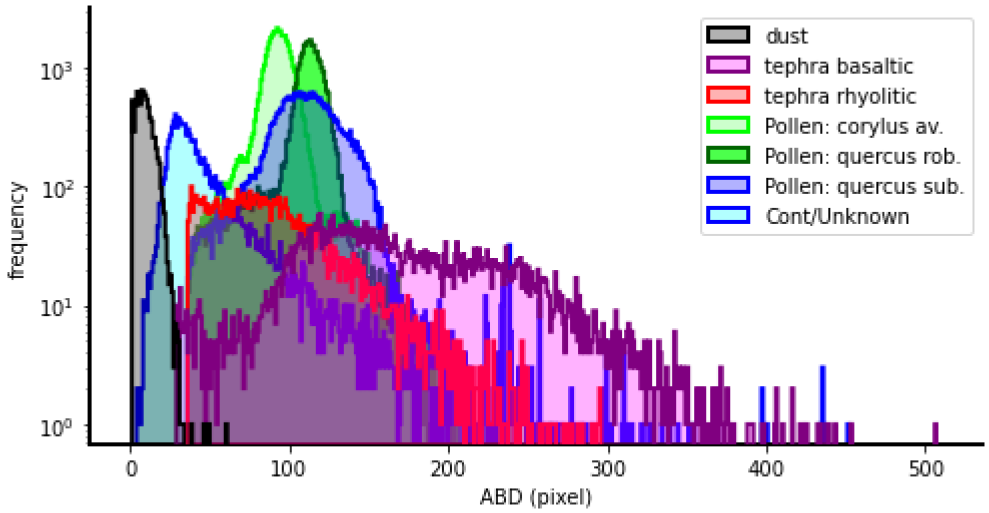
"While blurry images is an intrinsic limitation of this methodology, the Contamination/Blurry class serves the purpose of an important controlled channel for the model to be able to identify particles that do not carry climate significance."

4. Why are the images downscaled to 128x128 pixels and from what original size? Pretrained nets (including ResNet-18) on ImageNet usually take inputs of the size 224x224.

The images are scaled to 128x128 on the basis of their original sizes, described in the table and figure below, both of which refer to the training dataset. The vast majority of the particles are small(er), therefore we decided to limit (up)scaling to 128x128 pixels. Scaling to a bigger size seems not appropriate in the current setup (20X magnification scaling of 0.2752 µm/px).

The network is anyhow robust with respect to the input size (convolutional layers followed by global average pooling), which can therefore be changed if needed. If, for example, much bigger particles are expected to be contained in the samples (provided that they can enter the flow cell), training the model with images scaled to 224x224 or 512x512 pixels may be more appropriate.

Particle	**Mean µm (pixel)**	**Median µm (pixel)**	**2S µm (pixel)**	**Median + 2σ µm (pixel)**
Dust	2.5 (9.2)	2.4 (8.6)	2.92 (10.6)	5.3 (19.2)
Tehra F.	24.4 (89)	22.5 (81.8)	21.7 (78.9)	44.2 (161)
Tehra B.	46.9 (171)	43.8 (159)	36.1 (131)	79.9 (290)
Pollen *Corylus*	25.6 (93)	25.6 (93)	8.2 (30)	33.8 (123)
Pollen *Q. robur*	30.3 (110)	30.9 (113)	9.4 (34)	40.4 (147)
Pollen *Q. suber*	30.4 (110)	30.4 (110)	12.9 (47)	43.3 (157)
Cont/Blurry	13.9 (51)	10.5 (38)	22.2 (81)	32.7 (119)



5. The numerical features are based on geometrical properties and the product of a black-box by the FlowCam software. Are they necessary? Deep Learning methods operate extracting intrinsic features (if enough data is given) “by themselves”, feeding the network additional “hand-crafted” features is contrary to the very idea of DL. Is there an advantage in using the numerical features? (e.g. is there proof that they increase the accuracy?)

Yes, they are needed. If the MLP is removed, the accuracy drops by a few % (at test time). Similarly, the accuracy decreases (by ~10%), if the CNN is removed and a MLP model fed with the numerical features is used.

The reason for which the numerical features help is not obvious, as the Reviewer points out. Our explanation is that, by scaling all images to a fixed input size (in our case 128x128) as input for the CNN, the size information is “lost”, or less obvious for the model to pick up. By explicitly keeping such information (which is very important for some classes), the network is facilitated in correctly classifying the images.

We don’t really agree with the “black-box” software. We have created Appendix B so that the geometrical properties calculated by the software can be reproduced. For example, the ABD metrics can be obtained from the image by summing the pixels exceeding the luminosity threshold (set by the user, in our case to 18), multiplying by the calibration factor (=0.2752 μm/pixel in our setup), which result in the equivalent circle area A=π*(ABD/2)**2, and then obtaining ABD.

6. The work lacks some reference to practical computer vision applications in related fields. I would recommend including at least “Pattern recognition methodologies for pollen grain image classification: a survey” (Viertel, König, MVA Journal, 2022). This would also bring more insight into the problem mentioned in point no. 5.

Thank you. We have now added some references at the end of Sect. 1.