

Earth and Space Science



RESEARCH ARTICLE

10.1029/2024EA004112

Key Points:

- ResNets can accurately and reliably classify clouds into 30 World Meteorological Organization cloud classes from ground-based RGB pictures
- Class-specific data augmentation substantially reduces prediction biases and enables successful model training but introduces overfitting
- Ensemble mean predictions outperform single members in all classes

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:

M. Rosenberger,
markus.rosenberger@univie.ac.at

Citation:

Rosenberger, M., Dorninger, M., & Weissmann, M. (2025). Deriving WMO cloud classes from ground-based RGB pictures with a residual neural network ensemble. *Earth and Space Science*, 12, e2024EA004112. <https://doi.org/10.1029/2024EA004112>

Received 25 NOV 2024

Accepted 27 MAR 2025

Author Contributions:

Conceptualization: Manfred Dorninger, Martin Weissmann

Formal analysis: Markus Rosenberger

Investigation: Markus Rosenberger

Software: Markus Rosenberger

Supervision: Manfred Dorninger, Martin Weissmann

Validation: Markus Rosenberger

Visualization: Markus Rosenberger

Writing – original draft:

Markus Rosenberger

Deriving WMO Cloud Classes From Ground-Based RGB Pictures With a Residual Neural Network Ensemble

Markus Rosenberger¹ , Manfred Dorninger¹, and Martin Weissmann¹

¹Institut für Meteorologie und Geophysik, Universität Wien, Vienna, Austria

Abstract Clouds of various kinds play a substantial role in a wide variety of atmospheric processes. They are directly linked to the formation of precipitation, and significantly affect the atmospheric energy budget via radiative effects and latent heat. Moreover, knowledge of currently occurring cloud types allows the observer to draw conclusions about the short-term evolution of the state of the atmosphere and the weather. Therefore, a consistent cloud classification scheme has already been introduced almost 100 years ago. In this work, we train an ensemble of identically initialized multi-label residual neural network architectures from scratch with ground-based RGB pictures. Operational human observations, consisting of up to three out of 30 cloud classes per instance, are used as ground truth. To the best of our knowledge, we are the first to classify clouds with this methodology into 30 different classes. Class-specific resampling is used to reduce prediction biases due to a highly imbalanced ground truth class distribution. Results indicate that the ensemble mean outperforms the best single member in each cloud class. Still, each single member clearly outperforms both random and climatological predictions. Attributes diagrams indicate underconfidence in heavily augmented classes and very good calibration in all other classes. Autonomy and output consistency are the main advantages of such a trained classifier, hence we consider operational cloud monitoring as main application. Either for consistent cloud class observations or to observe the current state of the weather and its short time evolution with high temporal resolution, for example, in proximity of solar power plants.

Plain Language Summary Monitoring clouds in the sky can give experts important information on the current and upcoming weather, as well as other processes in the atmosphere. Machine learning models, more specifically so-called Convolutional Neural Networks, can be trained to constantly and automatically retrieve this information. This is done by repeatedly showing the model pictures of the sky in combination with cloud classes, which are visible on these pictures and have been determined by human experts in the past. During this process, the model learns distinctive visual properties of each class and can at some point correctly predict cloud classes from previously unseen pictures. In this work, we trained such models to find up to three out of 30 different cloud classes for each picture, in order to mimic human operational observations. Results show, that our model has overall a good performance but suffers from data shortage in specific classes, which may reduce applicability. However, our work can be considered a decent starting point, since in the future a larger data set or an even more sophisticated method may resolve this issue.

1. Introduction

Although there have already been a number of studies investigating the possibility to automatically classify clouds with Convolutional Neural Networks (CNNs) from pictures, either ground- or satellite-based, none of them considered more than 11 different classes. Some of them (Jiang et al., 2022; Lai et al., 2019; Wohlfarth et al., 2018; Zhang et al., 2018) define classes in analogy, or at least very similar, to the 10 cloud genera defined by the World Meteorological Organization (WMO) in the International Cloud Atlas (WMO, 2017), while others compress them to a smaller number of categories (Phung & Rhee, 2018; Wang et al., 2020; Xia et al., 2015). Contrary to these approaches, the model trained in our work also discriminates between cloud species and varieties. To account for random variations during the training process, due to for example, random mini-batch shuffling, we train 10 identical residual neural network architectures from scratch, and call it a classifier ensemble. We use a supervised learning approach, to classify ground-based cloud pictures into 30 classes, which are defined by the WMO for operational synoptic observations.

Clouds of different kinds can exhibit substantial influence on processes in the Earth's lower atmosphere as well as on its surface, for example, precipitation, or the atmospheric energy budget. The WMO defined 10 cloud genera as

© 2025. The Author(s).

This is an open access article under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

Table 1
Names and Descriptions of the Ten Cloud Genera (WMO, 2017)

Cloud genus	Short description
Cumulus (Cu)	Vertically developing, sharp outlines, nearly horizontal base
Cumulonimbus (Cb)	Considerable vertical extent, flattened top (anvil), often with precipitation
Stratocumulus (Sc)	Grey and/or whitish rounded masses, non-fibrous, apparent width mostly $>5^\circ$
Stratus (St)	Grey layer, uniform base, accompanied by drizzle or snow
Altostratus (As)	Greyish or bluish layer, covering the sky, partly thin enough to reveal Sun
Nimbostratus (Ns)	Grey, often dark layer; often accompanied by rain/snow, thick enough to blot out sun
Alto cumulus (Ac)	Grey and/or white patches or layers, apparent width mostly between 1° and 5°
Cirrus (Ci)	White filaments or patches or narrow bands, fibrous appearance and/or silky sheen
Cirrostratus (Cs)	Transparent and whitish veil, fibrous or smooth appearance
Cirrocumulus (Cc)	Thin and white patch or layer, elements in the form of for example, ripples, apparent width $<1^\circ$

Note. Genera Cu, Cb, Sc, and St are assigned to the low level, genera As, Ns, and Ac are assigned to the middle level and genera Ci, Cs, and Cc are assigned to the high level.

well as 15 species and nine varieties to differ between (WMO, 2017). The 10 cloud genera emerge from considering the most typical cloud forms ranging from grey layers close to the surface (*Stratus*) to patches with very small apparent widths in the highest levels of the troposphere (*Cirrocumulus*). Table 1 provides short descriptions for each cloud genus grouped by altitude level. To further describe the internal structure or the shape of clouds, cloud species were defined. For instance, Cirrostratus clouds can belong to the species *fibratus* or *nebulosus*, indicating that their appearance is either hair-like or smooth and without distinct details, respectively. Due to their mutually exclusive definitions, each cloud can only be assigned with one genus and one species. Transparency and arrangements of macroscopic elements of clouds are described by cloud varieties, which are in general not mutually exclusive and therefore one cloud can show properties of different varieties. For instance, varieties *translucidus* and *opacus* describe whether the majority of a cloud is translucent/opaque enough to reveal/mask the Sun or Moon. And although all of these definitions are primarily based on the visual appearance of clouds as seen from the Earth's surface, various categories influence atmospheric processes in different ways. For example, Cumulonimbus clouds can be responsible for severe rainfall and thunderstorm events, whereas an Altostratus opacus layer at night effectively traps infrared radiation emitted by the Earth's surface and thus reduces cooling of lower atmospheric layers. Based on this categorization scheme, operational synoptic observations (SYNOPS) differentiate between 30 well-defined cloud classes to report. These 30 classes are grouped into three height levels: low, middle, and high (coded as C_L , C_M , and C_H , respectively), with 10 classes each, of which nine are cloud classes and the tenth is used if no cloud is observed in the respective level. Operational observations consist of one class per level. Table 2 lists the cloud types, that is, combinations of cloud genus, species, and sometimes variety, that define each class, together with comics showing representative appearances of each category (WMO, 2017). Moreover, the Supporting Information S1 document contains pictures of each class from our data set.

The first attempts to classify clouds based on their visual appearance have been made at the beginning of the 19th century. In the following decades these primitive schemes have become more and more elaborate until almost 100 years ago the WMO defined global standards for cloud classifications (WMO, 2017). Thus, time series of cloud observations covering several decades can in principle exist for a number of long-living weather stations. And although the WMO created a sophisticated scheme, visual cloud classification is not at all straightforward. Sometimes clouds are in a transition phase between two defined classes, which makes it difficult to assign them to a single category. Also if clouds at different height levels overlap, properties of clouds in higher altitudes cannot

Table 2

Schematic Images and Short Descriptions (Both Taken From WMO, 2017) of Cloud Classes Used in This Work

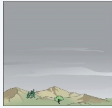




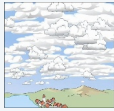
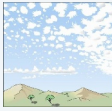



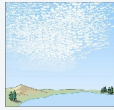
	C _L	C _M	C _H
1	Cu, small vertical extent 	As translucidus 	Ci fibratus/uncinus 
2	Cu, moderate to strong vertical extent 	(a) As opacus or (b) Ns (a)  (b) 	Ci spissatus/castellanus/floccus 
3	Cb calvus 	Ac translucidus, single level 	Ci spissatus cumulonimbogenitus 
4	Sc cumulogenitus 	Ac translucidus, continually changing 	Ci fibratus/uncinus or both, invading the sky 
5	Sc non-cumulogenitus 	Ac translucidus/opacus, invading the sky 	Cs invading the sky, <45° above horizon 
6	St of dry weather 	Ac cumulogenitus/cumulonimbogenitus 	Cs invading but not covering whole sky, >45° above horizon 
7	St fractus/Cu fractus of wet weather 	(a) Ac duplicatus or (b) Ac opacus in a single level or c) Ac + As/Ns (a) 	Cs covering whole sky 

Table 2
Continued

	C_L	C_M	C_H
		(b) 	
		(c) 	
8	Cu + Sc, different base heights 	Ac castellanus/floccus 	Cs not invading and not completely covering the sky 
9	Cb capillatus 	Ac of chaotic sky 	Cc 

Note. Columns show low, middle, and high altitude levels (C_L , C_M , and C_H , respectively) and rows represent SYNOP code for each cloud class.

be observed properly. Moreover, as can be seen from the images in Table 2, clouds of different classes and even in different height levels can have similar appearances, for example, *Stratocumulus non-cumulogenitus* ($C_L = 5$) and *Altostratus translucidus in a single level* ($C_M = 3$), which mainly differ in cloud base height and thus also apparent width of single elements. Another example are *Cirrostratus invading the sky*, which are coded as $C_H = 5$ if the cloud veil is less than 45° above the horizon and $C_H = 6$ if it is above 45° . Clearly, in such cases different observers potentially report different cloud types and thus cloud classification is always to some extent subjective. In addition, the number of operational cloud class observations stagnates around the world due to high monetary and personnel expenses for observers as well as the continuous automation of meteorological observations, but in general without automated cloud observations. An automatized method for such cloud classifications could solve both problems: It works independently and provides consistent results.

The first studies showing the feasibility of automatic cloud classification have been published more than 20 years ago. Tian et al. (1999) used feature extraction in combination with early neural networks to classify clouds on satellite images and Calbó and Sabburg (2008) used only feature extraction to differentiate between eight heuristic cloud classes on all sky images. Taravat et al. (2015) and Xia et al. (2015) also used ground based images to classify clouds into four categories with early machine learning methods, namely Support Vector Machines and Neural Networks, and k-nearest-neighbors, respectively. The rapid evolution of available computational power in recent years led to a broad range of new methods for image classification models. A large number of studies have proven that among those methods, CNNs (Fukushima, 1980; LeCun et al., 1998) show excellent accuracy in a broad field of image classification tasks in different fields, for example, face recognition (Schroff et al., 2015), detection of plant diseases (Lu et al., 2021), and detection of diseases of human lungs (Q. Li et al., 2014). CNNs are well suited for such tasks because they can learn spatial structures of input images and adapt well to the image's translation invariance (Nielsen, 2018). Therefore, also several studies have been published which utilized CNNs to classify clouds from different image types: Cai and Wang (2017) and Wohlfarth et al. (2018) discriminated between five and nine cloud classes, respectively, on satellite images. On the other hand, there have also been several studies, which classified clouds on ground based pictures into 5–10 categories (e.g., Lai et al., 2019; Phung & Rhee, 2018; Wang et al., 2020; Zhang et al., 2018). Zhao et al. (2020) combined an

algorithm to extract cloud-like objects from sky pictures with a CNN to classify these objects into three classes. However, it has to be mentioned that cloud classes used in these studies only consider cloud genera, or similar heuristic class definitions, without going much into detail. In our work, we want to show that CNNs can also be trained on a much more sophisticated classification scheme, which has been applied for almost 100 years, and can therefore be used to extend already existing time series.

Krizhevsky et al. (2012) trained a deep CNN on around 1.2 million images to discriminate between 1,000 categories, ranging from dog breeds over everyday objects to food items, and clearly outperformed previous state-of-the-art classifier. Since then, in order to further improve the model's accuracy, not only deeper CNNs, that is, with more layers, have been trained but also new methods have been introduced. The motivation was not only to improve performance but to do so with little or no additional model complexity to keep training times within reasonable borders. These methods also base on classical convolutional layers but differ for example, in the style of their output, for example, a U-Net returns an image instead of a vector of probabilities and can therefore assign classes pixel-wise (Ronneberger et al., 2015). While Sommer et al. (2024) use this approach to mark each pixel in All Sky Images either as cloud or clear sky, Fabel et al. (2022) additionally differentiate between low, middle, and high level clouds and Jiang et al. (2022) assign each pixel of satellite images one out of 10 different classes (clear sky + nine cloud types). He et al. (2015) did not alter the CNN architecture itself, but slightly changed the way information is processed within the model. Instead of using just the output of one layer as input for the subsequent layer, *identity shortcut connections* were introduced. This is a mapping technique where the input of a layer is added to the layer's output before it is processed to subsequent model layers, to counteract the possibly decreasing accuracy of models with increasing depth. This kind of model architecture is called Residual Neural Network (ResNet; He et al., 2015) and allows models to consist of tens of convolutional layers without suffering from degrading performance. Later, Zhu and Newsam (2017) introduced the *DenseNet*, where shortcut connections not only skip single convolutional layers but range from each layer to each subsequent layer until the top of the model, allowing information to be transported from very coarse to the finest resolutions. S. Li et al. (2023) use this DenseNet architecture as backbone to train a classifier via transfer learning for up to 11 cloud classes from ground-based pictures. Similarly, Guzel et al. (2024) fine-tuned different pre-trained image classification models for cloud classification into 11 classes and reached an overall accuracy above 97%.

The above mentioned cloud classification models are based on different methods and some of them even achieved accuracy scores above 90%. However, they suffer from limited usability in an operational setup because they only consider a single cloud class per image (except for the U-Net based model of Jiang et al. (2022)) and most of them only consider a small subset of defined cloud classes. Therefore, in this work we want to show that CNNs can also handle the more sophisticated multi-label cloud classification approach and can in principle be used to automatize this task. A major problem of this approach is the highly imbalanced number of observations for each class. However, we perform class-specific data augmentation as well as random sub-image shuffling in each training period to reduce the model's prediction bias. Moreover, an ensemble consisting of 10 model runs with identical initialization was trained to get more stable results. Evaluation scores are therefore calculated for single ensemble members as well as the ensemble mean and indicate that the latter outperforms in each class the respectively best single member.

After describing the data set in Section 2, details of the used model architecture and evaluation metrics are introduced in Section 3. Section 4 presents results of our work and final conclusions are drawn in Section 5.

2. Data

2.1. Pictures

We train our model on ground-based RGB pictures, which are taken in Vienna, Austria (48°13'48"N, 16°21'36"E, altitude: 198 m). Four 1.3 Megapixel Canon VB-M600VE cameras are aligned in the main cardinal directions and each of them takes one picture every 30 s. The angle of view of each camera is 101.2° and thus large enough to allow some overlap between two neighboring pictures. In the vertical direction the cameras cover the sky up to an elevation of 75°. Therefore, combining the four pictures covers almost 97% of the visible sky. Raw pictures are curved towards the edges and therefore have to be pre-processed to retain the natural cloud shape. Pictures are available for two different periods, namely 05.10.2016–19.02.2019 and 04.05.2022–19.02.2023. Picture resolution after pre-processing depends on period and direction, ranging from 567 × 967 pixels to 581 × 1092 pixels. To achieve conformity and keep the model size feasible, but still approximately preserve the picture shape

to avoid distortions, all pictures are re-scaled to 64×100 pixels. Although pictures are available at a temporal resolution of 30 s, we only include those taken at full hours into our data set, because operational cloud observations are also available on an hourly basis. Moreover, only those pictures are considered, in which enough sunlight is available to see all visible features in the sky, dependent on time-of-day and day-of-year, based on the author's subjective perception.

Since operational cloud observations by humans are made based on the whole visible sky, in our data set we only consider time steps where pictures in all four directions are available. In cases where at least one picture at full hour is missing, which happened at less than 5% of the instances in the raw data set, pictures taken in the same direction shortly before or after the initial time step are used as replacement. We set the temporal radius to 15 min before/after the initial time of observation since we estimate that within this time span observable cloud classes usually do not change significantly, so that the ground truth observation is still applicable. Examples of pictures in our data set are shown in Figure 7. Moreover, representative pictures of each cloud class are provided in the Supporting Information S1.

2.2. Ground Truth

Ground truth cloud class observations are taken from human operational SYNOP observations, which are performed at the station Vienna Hohe Warte (WMO station number 11035) according to criteria defined by the WMO. The distance between this station and the location of our cameras is only approximately 2 km, which ensures that observers at Vienna Hohe Warte report cloud classifications according to the same part of the sky that is also covered by the cameras. Meteorological observations at Vienna Hohe Warte are reported every hour between 05UTC and 21UTC, as well as at 00UTC. Observations are freely available from the data hub of the operating weather service GeoSphere Austria (GeoSphereAustria, 2020). Each observation instance used in this work contains one out of 10 categories per height level. Category 0 is “no cloud in this level” and categories 1–9 are the cloud classes depicted in Table 2. For cases, where more than one cloud class is present in a level, there is a well-defined decision tree for each level to find out which cloud category has to be reported (WMO, 2017). The flow-chart for low level clouds, for example, starts with the question if Cb clouds are present. If they are, the next question is about the existence of fibrous structures, an anvil, or a plume to discriminate between the classes $C_L = 3$ and $C_L = 9$. If no Cb clouds are present the next question is if the observed cloud is formed by spreading out of Cu which would result in reporting $C_L = 4$. Similar yes-no-questions guide an observer through the remaining low level cloud classes as well as through the middle and high level. Observations of every instance are converted to a single ground truth vector with 30 entries, where the first 10 entries correspond to the low level cloud observation, the following 10 entries to the middle level and the last 10 to the high level. The ground truth vector is one-hot encoded, that is, entries which correspond to observed classes are set to one and all other entries are set to zero. If in one layer clouds were not observable either because of phenomena like dust and fog, or because lower clouds formed a continuous layer, $C_M = /$ and/or $C_H = /$ are reported. In this case, all entries of the corresponding part of the ground truth vector are set to zero. Thus, each instance in the ground truth consists of either one (only the lowest level was always observable), two (the upper level was not observable), or three (all levels were observable) cloud classes, which makes this problem a multi-label classification task.

Although the cloud classification scheme was well-defined by the WMO, ambiguities can still arise. Often, clouds in transitional stages can be observed, which cannot be clearly assigned to a single class. The same is true if not all properties of a cloud are observable due to an underlying cloud layer. Hence, visual cloud classification is always subjective to some extent and the result can depend among others on the observer's experience and the additionally used information, like satellite or radar data. Since all ground truth observations are created by experts from the same weather service, consistency for all labels can be assumed. However, mistakes which are made during the creation of the operational SYNOP report can lead to erroneous reports. In this work, we did not account for potential errors in the ground truth, neither of systematic nor of random nature. Thus, each cloud observation is treated as perfectly accurate, although our investigations showed that the data set contains incorrectly classified instances, which could lead to reduced model performance.

2.3. Data Augmentation

In total, almost 12,000 instances consisting of four pictures and an assigned ground truth observation are available. However, by nature cloud classes have a highly imbalanced probability of occurrence, which is shown

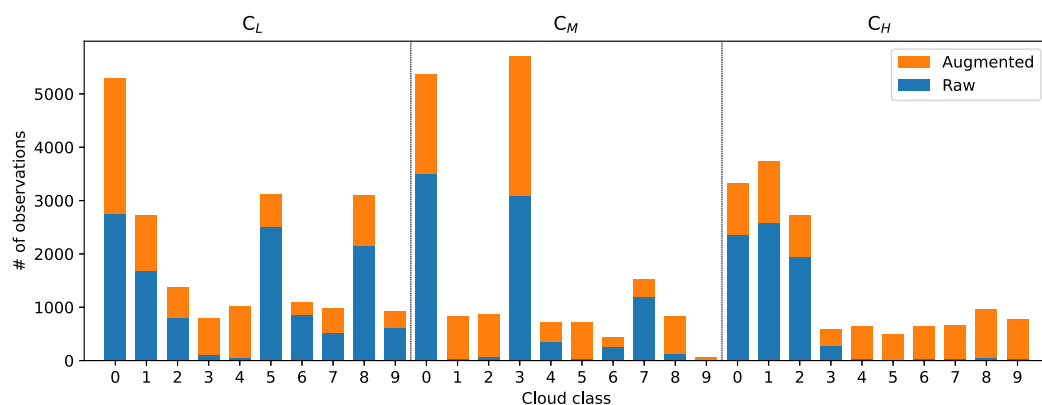


Figure 1. Number of observations per class in each height level in the raw data set (blue) and after class specific resampling (orange) of less abundant categories.

by the blue bars in Figure 1. Experiments, in which models were trained using this raw data set, showed that the imbalance results in severe model forecast biases or no model convergence at all. Hence, we performed *class specific data augmentation*. Similarly to what was described previously, we added pictures to the data set which have been taken within up to 15 min before or after the observation of a rare cloud class. With this data augmentation strategy we can increase the abundance of specific classes by a factor of up to 60 and therefore ensure that each category is represented by a sufficient number of instances in the data set. However, a drawback of this method is that some cloud classes are still represented by only a few instances and a large fraction of available pictures for the same class looks rather similar, which increases the risk of overfitting in affected classes (Kaur et al., 2019). Moreover, since more than two thirds of all instances contain at least two cloud classes, by adding instances to one class, we automatically also increase the number of instances in all categories that have been observed at the same time. Hence, this method only slightly reduces the relative imbalance between all categories. After data augmentation, our data set consists of more than 20,000 pictures. Orange bars in Figure 1 show the number of instances added to each class in the raw data set by augmentation. We also tried other data augmentation methods like horizontal shift and rotation but experiments showed no improvement in model performance. Training, validation, and test data sets make up 68%, 17%, and 15% of the total data set, respectively, while keeping the characteristics of the observation frequencies of the total data set in each subset.

3. Method

3.1. Model

In this study we utilize a residual neural network architecture, since this type of CNN is less prone to possibly degrading accuracy in deeper network architectures compared to shallower counterparts (He et al., 2015). The characteristic ingredient of residual neural networks are identity shortcut connections, where the input of a convolutional layer is added element-wise to the output of the same or a subsequent convolutional operation. This identity mapping adds no parameters to the model and therefore also computational complexity remains unchanged. Our model is trained from scratch and starts with a single convolutional layer with 7×7 filters and a stride of 2, which increases the number of feature maps from the three color channels RGB to 8. Afterwards, there are four blocks of *stacked-residual-residual layers* (S-R-R). Each one halving the feature map size and doubling the feature map number. Figure 2 shows a schematic overview of the model architecture, the *stacked layer*, and the *residual layer* used in this work. Grey boxes represent outputs of convolutional operations and numbers along each axis indicate the size of the respective dimension.

A *stacked layer* consists of a convolutional layer with 5×5 filters and a stride of 3, which also doubles the number of feature maps, followed by a batch normalization layer and a leaky ReLU activation function with a negative slope of 0.3. Adding a batch normalization layer before the application of an activation function smoothes the loss surface, which allows higher learning rates and thus leads to shorter training times (Ioffe & Szegedy, 2015; Prince, 2023). Furthermore it regularizes the model, which makes additional dropout layers redundant. In our work, a *residual layer* consists of two Convolution-LayerNorm-LeakyReLU blocks and

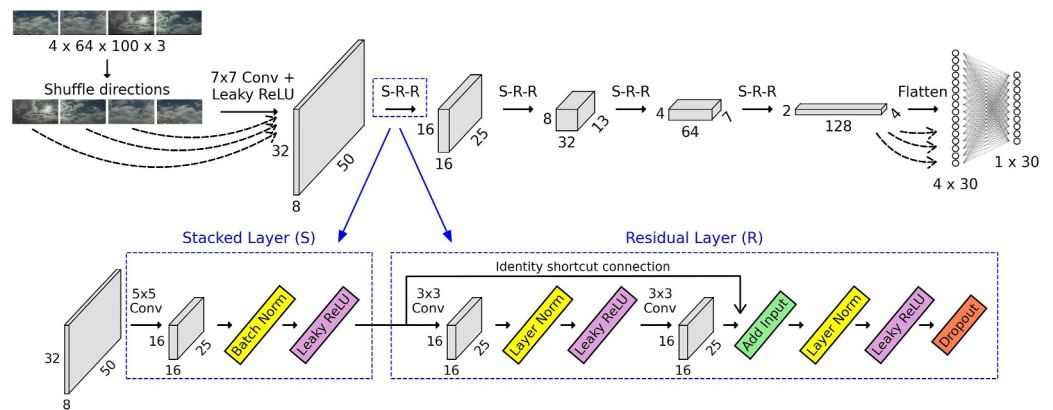


Figure 2. Schematic representation of the residual neural network architecture we utilize in this work. At first, the arrangement of sub-pictures of the same instance is randomly changed at each epoch. Then, as indicated by the dashed arrows, each sub-picture is separately processed by a combination of stacked layers (S) and residual layers (R) and results in a 30-node output array. These arrays are then combined into a final 30-node dense layer, which generates the predicted probabilities for each class to be observed in any of the sub-pictures. Grey boxes represent outputs of convolutional layers with numbers along the axes denoting the size of the respective dimension.

changes neither the number nor the size of feature maps. Convolutional layers have 3×3 filters and a stride of 1 so that the model can also account for small-scale patterns in input pictures. Ba et al. (2016) state that using batch normalization introduces limits to the smallest possible batch size. Thus, in residual layers we use layer normalization after the convolution, to weaken possible side effects of using small batch sizes. Also, models trained with the same architecture but other combinations of layer and batch normalization, or those using only one of them throughout the whole model, perform worse or do not converge at all. The subsequent leaky ReLU activation is identical to the one in the stacked layer. Following He et al. (2015), in each residual layer shortcut connections are placed in a way that the input of the first convolution is added element-wise to the output of the second one, before normalization and activation are applied. At the very end of each residual layer, a dropout layer with ratio 0.2 is put for regularization. The output of the fourth S-R-R block is flattened and then fed into a dense layer with sigmoid activation and 30 output nodes. The most notable differences to the ResNet architecture of He et al. (2015) are the utilization of non-residual layers with 5×5 convolutions, the combination of layer normalization and batch normalization throughout our model, as well as not using any pooling layers. Also we reduced the number of parameter layers approximately by a factor of 2 compared to ResNet-34.

For each instance, we use the pictures of all four directions as model input, because human cloud observers also consider the whole visible sky for their report. However, rather than stacking the pictures to generate a single panorama image, each picture is processed separately by the model and converted into a 30-node output vector as described above, which is indicated by the dashed arrows at the beginning and the end of the model in Figure 2. The four resulting output arrays are then combined by another dense layer to the final output vector, again with 30 nodes and sigmoid activation, to allow multi-label classification. Since filter weights are shared within the picture dimension, all weights in the model, except for those in dense layers, are trained on the four sub-images simultaneously and the number of parameters in this part of the model remains unchanged compared to a model trained on panorama images. Moreover, using separate sub-images allows us to carry out sub-image shuffling as additional data augmentation method. By randomly switching positions of sub-images within the same instance at each training epoch, the first set of dense layers is also trained on pictures of each direction throughout the training process, which, as we assume, reduces overfitting on direction-specific picture properties, for example, visible landscape and buildings.

We use the Adam optimizer (Kingma & Ba, 2014) for optimization with a mini-batch size of 16, which is large enough to guarantee a stable training process but still small enough to get a large number of model updates per training epoch. Furthermore, we use an initial learning rate of 1×10^{-4} , which is reduced stepwise by 15%, as soon as a plateau in validation loss is persistent for more than seven epochs. Our model is trained for a maximum of 200 epochs and training is stopped if validation loss does not improve for eight consecutive epochs. Each model has less than 200 million FLOPs, which is a considerable reduction compared to ResNet-34 (3.6 billion FLOPs) and training needs approximately four hours on 16 CPUs, while inference is done within a few ms per instance.

3.2. Loss Function

When training a neural network, choosing a proper loss function is very important, since it is not only used to calculate the difference between predicted and observed labels but it can also put emphasis on specific properties of the output. For example, Ben-Baruch et al. (2021) introduce the Asymmetric Loss function *ASL* for multi-label classifications, which is based on Focal Loss (Lin et al., 2018) and therefore also on binary cross-entropy, the state-of-the-art loss function for multi-label classification tasks. *ASL* accounts for the fact that the number of observed classes is mostly smaller than the number of not observed classes in a single instance and therefore puts different weights on the positive and the negative component of the loss function. Though this can be an improvement compared to for example, binary cross-entropy, it also introduces three additional hyperparameters to tune. In this work, we use the Brier Score (*BS*; Wilks, 2019) as loss function \mathcal{L} , though this choice may be uncommon, especially in classification tasks. However, experiments with a variety of combinations of hyperparameters and model architectures, showed that it leads to very good performance scores:

$$\mathcal{L} = BS = \frac{1}{N} \sum_{k=1}^N (\mathbf{y}_k - \mathbf{o}_k)^2, \quad (1)$$

where \mathbf{y}_k and \mathbf{o}_k denote vectors of predicted probabilities and observations, respectively, for instance k . Values in \mathbf{y}_k for class c are in the interval $[0,1]$, while $o_c = 1$ if the class c has been observed and $o_c = 0$ otherwise. N is the batch size. Although this definition is slightly different than the original Brier Score (Brier, 1950), for the sake of convenience and consistency with other studies, and because it is more common nowadays, in this work we will also call this definition Brier Score. Murphy (1973) showed, that *BS* can be decomposed into three terms, namely uncertainty, reliability, and resolution (cf. Equation 5). Calculated for a single cloud class, the values of reliability and resolution indicate how well a classifier is calibrated, that is, if predicted probabilities represent the frequency of observation. Therefore, by using batch-wise *BS* as loss function, we expect not only to improve the model's accuracy but also properties such as resolution and reliability.

3.3. Evaluation Metrics

3.3.1. Accuracy

In meteorology, non-probabilistic dichotomous forecasts are usually evaluated using a 2×2 contingency table. Rows of this table contain instances where a specific event was observed or not and columns contain at which instances the same event was predicted or not. Thus, the four entries of the table count how often the event was both predicted and observed (*hit*), predicted but not observed (*false alarm*), observed but not predicted (*miss*), or neither predicted nor observed (*correct rejection*). These four values can then be used to calculate a wide variety of statistics (Wilks, 2019). The same framework is also used for machine learning tasks like single-label image classification, though terminology is slightly different. The contingency table is called confusion matrix, a hit is called *True Positive (TP)*, a false alarm is the same as a *False Positive (FP)*, a miss becomes a *False Negative (FN)*, and a correct rejection is called *True Negative (TN)*. In this work, we will stick to machine learning terminology but mention standard names in the meteorological framework for statistics, if applicable. Note, that it is common for machine learning classifiers to return probabilities for each class to occur. Therefore, to convert probabilistic to dichotomous forecasts, a probability threshold p_t has to be introduced above which a class is said to be predicted. In this work $p_t = 0.50$.

To evaluate a multi-label classifier, often the one-vs-rest approach is used, where TP, FP, FN, and TN are calculated for each class separately versus all other classes. However, Heydarian et al. (2022) claim that this leads to erroneous FP and FN values and introduced the Multi-Label Confusion Matrix (MLCM), which correctly calculates these scores. The MLCM consists of one row and one column per class as well as one additional row *No True Label (NTL)* and one additional column *No Predicted Label (NPL)*. Entries in NTL are instances for which no ground truth label was assigned. Since this does not apply to any instance in the data set we use, the last row of the MLCM is always empty. The column NPL consists of instances where one or more observed classes were not predicted by the model and at the same time no false positive prediction occurs.

In the resulting MLCM, the number of TP predictions for a single cloud class is located in the main diagonal entry of the respective row and the amount of TN predictions results from the sum of all other main diagonal entries.

The number of FN and FP predictions can be calculated by row-wise and column-wise summation, respectively, of associated off-diagonal values. From these values, statistics can be computed to assess the classifier's performance. In this work we focus on some of the most common and, if considered together, also highly informative multi-label measures (Chicco & Jurman, 2020; Pereira et al., 2018), that is, Precision (P , also called *post agreement*, which is calculated as $1 - \text{false alarm ratio}$), Recall (R , also known as *hit rate*), and Matthew's Correlation Coefficient MCC , which are defined as:

$$P = \frac{TP}{TP + FP} \quad (2)$$

$$R = \frac{TP}{TP + FN} \quad (3)$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \quad (4)$$

In terms of the MLCM, P of each category can be calculated by dividing the value on the main diagonal by the sum of all values in the respective column. Therefore, P gives how often a specific class was actually observed at instances where it was predicted by the model. Similarly, R is calculated as the main diagonal entry divided by the row-wise sum for each row of the MLCM and can be interpreted as the proportion of instances for which a specific class has been predicted correctly by the model divided by the total number of observations of this class. Both metrics range in the interval $[0,1]$ and higher scores indicate better model performance. MCC can be considered as the correlation between true classes and predicted labels and can also be calculated as their covariance divided by the product of their respective standard deviations (Chicco & Jurman, 2020). MCC is especially useful for classification tasks with imbalanced classes because it considers all elements of the confusion matrix and therefore a classifier has to perform well on the majority of both positive and negative cases in order to achieve a high score (Chicco & Jurman, 2020). MCC ranges in the interval $[-1, +1]$, where $MCC = -1$ indicates perfect misclassification, $MCC = 0$ is the result of random classifications, and $MCC = +1$ is achieved by a perfect classifier.

3.3.2. Calibration

Guo et al. (2017) state that modern neural networks, among others also ResNets which are utilized in this work, tend to show high accuracy but lack confidence, or reliability. A classifier is said to have calibrated confidence, if predicted probabilities of a class represent the actual probability that this class is observed in the associated instance, that is, a class should be observed in $p\%$ of the instances for which it is assigned a probability of $p\%$. Murphy (1973) introduced a way to assess this property by partitioning the Brier Score BS into three terms, namely uncertainty (UNC), reliability (REL), and resolution (RES). Computing BS for a single cloud class and all instances and following Murphy (1973) gives:

$$BS = \frac{1}{n} \sum_{k=1}^n (y_k - o_k)^2 = \underbrace{\bar{o}(1 - \bar{o})}_{\text{UNC}} + \underbrace{\frac{1}{n} \sum_{i=1}^I n_i (y_i - \bar{o}_i)^2}_{\text{REL}} - \underbrace{\frac{1}{n} \sum_{i=1}^I n_i (\bar{o}_i - \bar{o})^2}_{\text{RES}}, \quad (5)$$

where n is the total number of instances. y_k is the predicted probability of the cloud class under investigation at instance k . Again, $o_k = 1$ if this class was observed at instance k and $o_k = 0$ otherwise. In order to decompose BS , Murphy (1973) divided the range of predictable probabilities into I mutually exclusive and collectively exhausted states with n_i instances in the $i - th$ subsample. y_i is a representative probability value for each probability bin and \bar{o}_i is the relative frequency of occurrence of the specific cloud class for all instances in the $i - th$ subsample. \bar{o} is the overall relative frequency of occurrence (Hsu & Murphy, 1986) of this cloud class.

While the value of UNC does not depend on the forecast system in any way and is determined purely by the climatological frequency of occurrence of a cloud class, REL and RES describe the trustworthiness of predicted probabilities. More precisely, REL quantifies how close predicted probabilities match the frequency of observation. Perfect reliability or calibration, that is, $REL = 0$, is achieved if an event is observed in $p\%$ of occasions when the predicted probability was $p\%$ (Wilks, 2019). On the other hand, the resolution measures how well a

forecast system can discriminate events with a high from those with a low frequency of occurrence. $RES = 0$ indicates that the overall frequency of occurrence of an event is identical to the frequency of occurrence in each probability interval. Hence, high values of RES are desirable. Both measures can be illustrated together in an attributes diagram, an extended version of the classical reliability diagram (Hsu & Murphy, 1986). In order to compare a forecast system under investigation to a reference forecast the Brier Skill Score (BSS) can be calculated. Wilks (2019) showed that using climatological predictions as reference forecast gives:

$$BSS = \frac{RES - REL}{UNC}, \quad (6)$$

where $BSS > 0$ indicates skill when the forecast system is compared to climatology, which is the case whenever $RES > REL$.

3.3.3. Averaging Methods

There are two methods to combine scores of single classes to overall scores in multi-class or multi-label classification tasks: *micro-averaging* and *macro-averaging*. In the context of P , R , and MCC , micro-averaged scores are computed using sums of class-wise calculated TP, FP, FN & TN values. On the other hand, macro-averaged scores are computed by calculating the respective score for each class separately and then calculating the (weighted) average (Giraldo Forero et al., 2015).

$$M_{\text{micro}} = M\left(\sum_{q=1}^Q TP_q, \sum_{q=1}^Q FP_q, \sum_{q=1}^Q FN_q, \sum_{q=1}^Q TN_q\right) \quad (7)$$

$$M_{\text{macro}} = \frac{1}{Q} \sum_{q=1}^Q M(TP_q, FP_q, FN_q, TN_q) \quad (8)$$

$$M_{\text{macro, weighted}} = \frac{1}{W} \sum_{q=1}^Q w_q \cdot M(TP_q, FP_q, FN_q, TN_q), \quad (9)$$

where $M(\cdot)$ is the evaluation measure, Q is the number of classes, w_q is the weight of class q and $W = \sum_q w_q$. Unweighted macro-averaging of a score assigns each label with the same weight and gives therefore a hint on the overall performance of a classifier, given that the result for each class is equally important. Micro-averaging, however, may be more suitable for problems with imbalanced classes since it weighs all instances equally (Yang, 1999). This has two implications: Firstly, if the micro-average is higher than the macro-average of a score, higher abundant classes tend to perform better. Secondly, a weighted macro-average score is equal to the micro-average of the same score, if observation frequencies are used as weights. However, for multi-label classification, Heydarian et al. (2022) suggest to use row-wise sums of the MLCM as weights. By construction of the MLCM, this value in general exceeds the number of actual observations of a class and hence weighted macro-averages can differ from micro-averages, although this is not true for the Recall. Since $R = \frac{TP}{TP + FN}$ and the weights are also given by $TP + FN$, micro-averaged and weighted macro-averaged recall scores are always identical. To show this similarity, we report results of micro-averaged scores together with (weighted) macro-averages in Table 3 but we do not include them in the discussion.

The fact that we trained an ensemble of classifiers, allows three further ways of evaluation. One can either calculate measures from the ensemble mean of predicted probabilities (*average-based, AB*) or compute the average of measures calculated for each ensemble member separately (*member-based, MB*). Though we expect the ensemble mean to outperform the single-member approach, the span width of MB evaluated scores indicates the degree of variability within the ensemble. Large span widths would indicate that results of a single trained model may not be trustworthy because of random variations during the training process, which may lead to the model performing well in some classes just by chance. The third approach is to derive a single prediction by majority voting (*MV*) of all ensemble members. Given an ensemble which predicts a range of different classes for the same instance, majority voting can still filter commonalities of several members to get a meaningful forecast. Therefore, the final MV prediction of a given instance consists only of classes which have been predicted by at

Table 3

Summary of Scores to Evaluate Performance of the Cloud Classifier Ensemble Trained in Our Work

	AB micro	AB macro	AB macro (w)	MB micro	MB macro	MB macro (w)	MV micro	MV macro	MV macro (w)
Precision	0.72	0.83	0.72	0.61	0.70	0.61	0.60	0.72	0.61
Recall	0.58	0.62	0.58	0.53	0.58	0.53	0.59	0.66	0.59
MCC	0.63	0.68	0.60	0.54	0.60	0.52	0.57	0.65	0.54

Note. Macro- and micro-averages of each score are calculated on ensemble average and single member basis (AB and MB, respectively) as well as for predictions created by majority voting of ensemble predictions (MV). Macro-averaged scores are computed without weights as well as using row-wise sums of the MLCM as weights (w). Boldface font highlights best values for each score.

least two ensemble members. Therefore, for each metric nine different values can be computed: micro-averages, unweighted macro-averages, and weighted macro-averages for majority voting, average-based, and member-based evaluation. In the rest of this paper, for each statistic both the averaging and the evaluation method will be indicated by subscripts.

4. Results

4.1. Multi-Label Confusion Matrix (MLCM)

Figure 3 shows the average-based MLCM. High values along the main diagonal indicate that the majority of instances was classified correctly by our model. P_{AB} , achieved by column-wise normalization of the matrix, gives values between 60% and 100%, while on the other hand R_{AB} , that is, row-wise normalization, tends to show smaller values. For some classes R_{AB} is even below 40% but also reaches 90% in others. Both metrics show highest values in classes which have been augmented most aggressively, for example, $C_H = 5$ and $C_H = 6$, and lower, but still good, scores in classes which appear more frequently in the raw data set, for example, $C_L = 5$ and $C_H = 2$. Since some cloud categories have initially only been observed in a very small number of instances, for example, $C_M = 9$ with only one single observation in more than 3 years of data, all available pictures that contain clouds of these specific classes are rather similar. Therefore, the model can easily recognize them in the test data set, which leads to almost 100% correct predictions. It is highly probable that the model overfits in these categories and will not perform as good on out-of-sample pictures. This is a problem that can possibly occur when sampling methods are used to reduce class imbalances (Kaur et al., 2019).

It is also evident that there are several reasons for false model predictions: On the one hand, the model is still biased towards most abundant classes, for example, $C_L = 5$, $C_M = 3$, and $C_H = 1$, which are therefore predicted more frequently than others. However, on top of this bias also visual similarities of two or more categories lead to false predictions. For example, classes $C_L = 5$ and $C_L = 8$ both contain Stratocumulus clouds, which did not evolve from Cumulus or Cumulonimbus clouds. If only this kind of Stratocumulus clouds is visible in the lowest level, or if it is accompanied by Cumulus clouds which have their bases at the same height as the Stratocumulus clouds, $C_L = 5$ has to be reported. However, if also Cumulus clouds with bases at different levels than the Stratocumulus are visible, $C_L = 8$ has to be chosen. Without additional information on cloud base height or a proper 3-dimensional view of the sky, it can be very difficult to distinguish between both classes, even for trained cloud observers. Our model also seems to struggle with discriminating between these two categories since 39 instances were classified as $C_L = 5$ when $C_L = 8$ was present and 58 times the opposite happened. Both numbers are several times larger than FP instances for other observed classes in the same columns, indicating that this is not just noise but actually indicates a systematic pattern.

Instances, where not every true class but at the same time not a single false one was predicted by the model, that is, instances that can be found in the *NPL* column of the MLCM, make up the third major error source. The ratio of such cases reaches up to around 50% for some categories, examples are $C_L = 3$ (Cumulonimbus calvus), $C_M = 6$ (Altostratus originating from Cumulus/Cumulonimbus), and $C_M = 7$ (Altostratus translucidus in layers or Altostratus opacus or Altostratus together with Altostratus/Nimbostratus). For the prior class, numbers of FP and, apart from the *NPL* column, also FN are close to zero, which indicates that the model did not confuse this class with any other. It was just not able to recognize all observable classes in the associated pictures or the predicted probability was too small even if the correct class was found. On the other hand, the latter two categories suffer from a substantial number of FN predictions compared to the respective category sizes. This indicates that

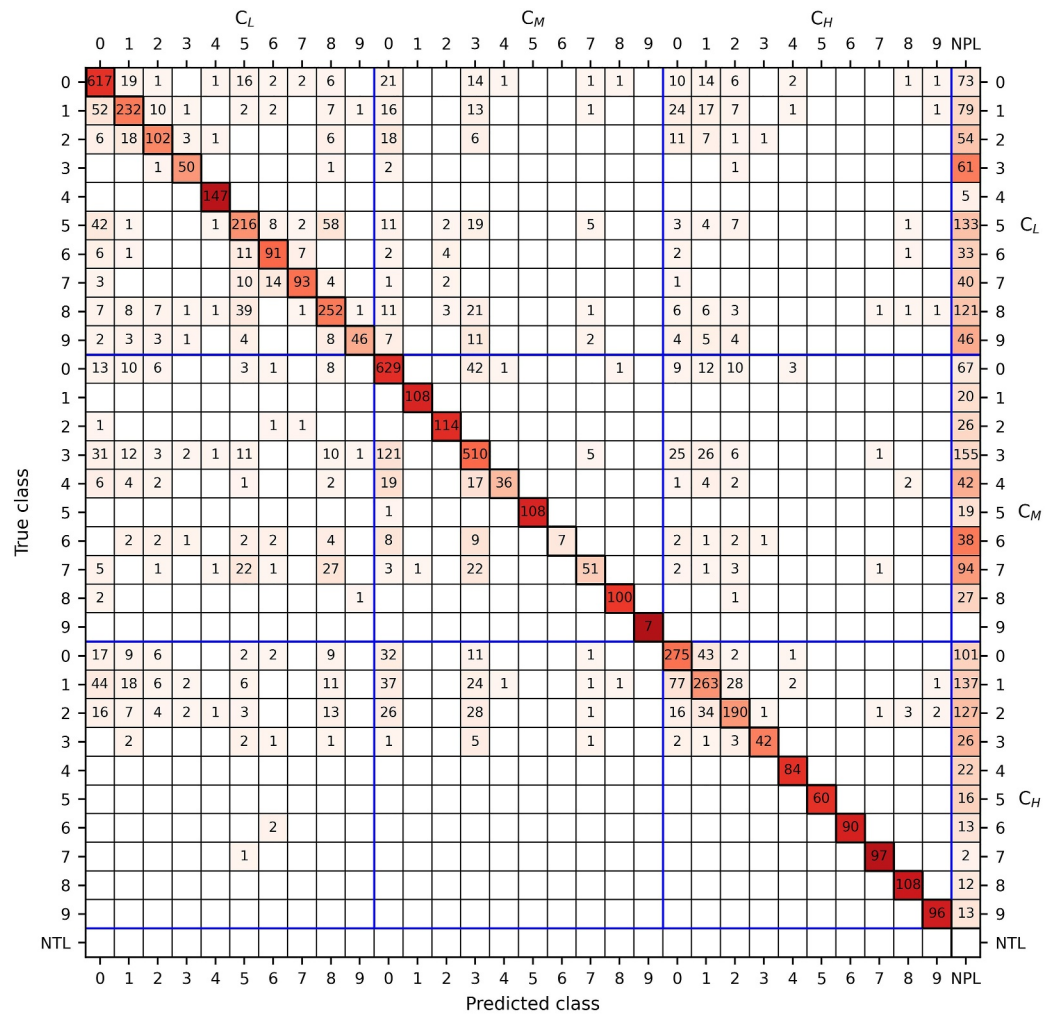


Figure 3. Average-based Multi-Label Confusion Matrix. Each row and each column correspond to one cloud class in one of the three height levels low, middle, and high. Row *No True Label* is empty because no unlabeled pictures are present in our ground truth data set and column *No Predicted Label* contains instances at which at least one ground truth label was not predicted and at the same time no false positive predictions were made.

the model often cannot discriminate the true class from similar ones like *Altostratus translucidus* ($C_M = 3$) or both *Stratocumulus* categories ($C_L = 5$ and $C_L = 8$). In several other categories, for example, $C_L = 9$ (*Cumulonimbus capillatus*) and $C_H = 3$ (*Cirrus spissatus cumulonimbogenitus*), up to 30% of instances were assigned to the *NPL* column. Reconsidering Equation 3, one can see that high numbers of FN predictions lead to reduced recall values in these classes. Evaluating single forecast instances also showed that often the model recognized the correct class c from the pictures but the predicted probability was too small to be considered as positive prediction, that is, $p_c < 0.50$. Therefore, the recall scores can be improved by increasing the model's confidence to predict potentially correct classes with higher probabilities.

4.2. Class-Wise Metrics

Figure 4 summarizes distributions of P , R , and MCC in panels a, b, and c, respectively, for each class and the whole classifier ensemble. Patches at the right edge of each plot indicate weighted averages of the full ensemble span width, that is, the range between the class-wise average of the worst and the best classification performance of our ensemble. Black dashes indicate weighted MB macro-averages, which are above 0.50 for R and MCC , and even >0.60 for P , with averaged maxima being between 0.04 and 0.08 higher. Unweighted averaging led to slightly higher scores (0.70, 0.58, 0.60 for P , R , and MCC , respectively) and also larger average span widths for each metric. This means, that in general higher abundant classes show worse scores than sparsely represented

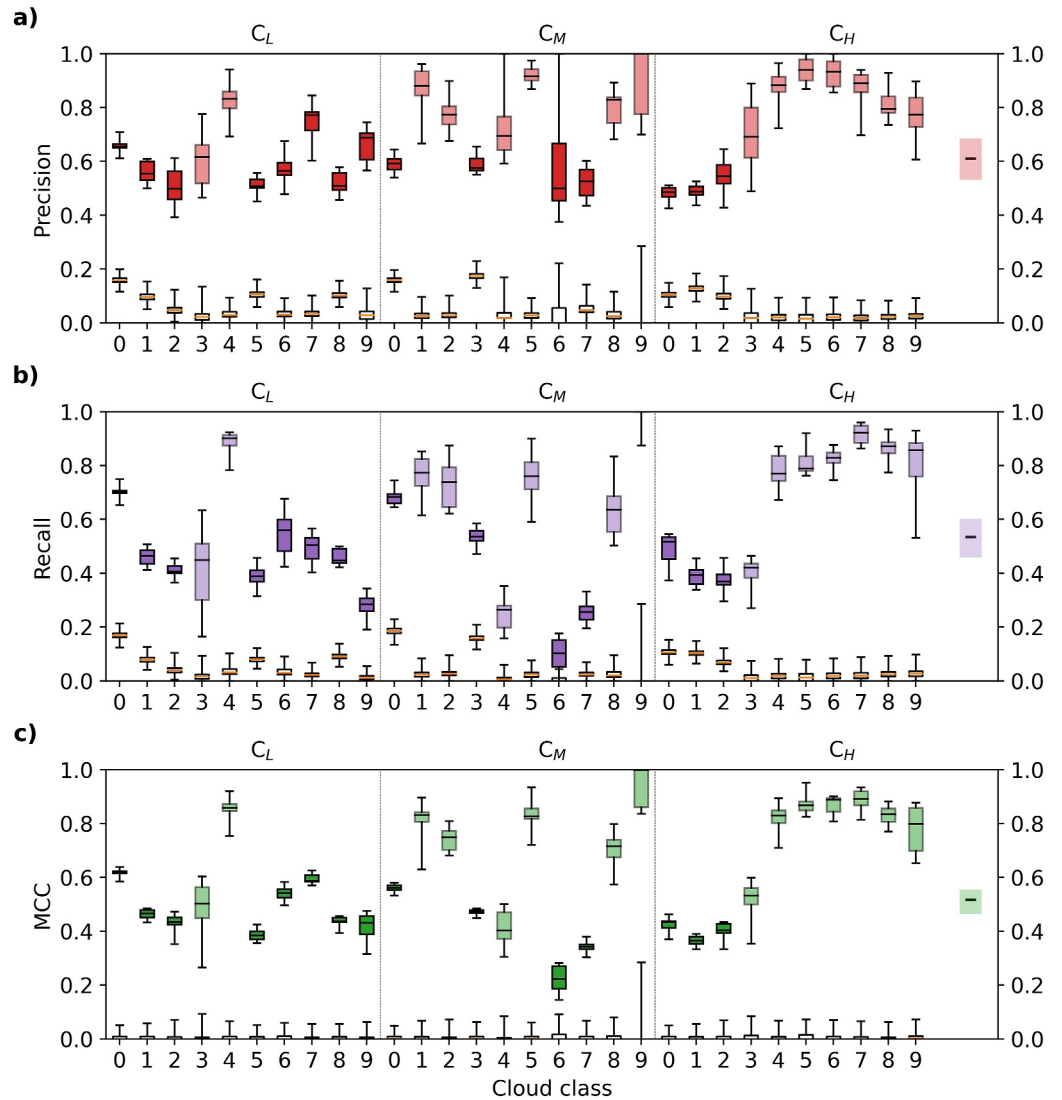


Figure 4. P , R , and MCC for each class in panels (a–c), respectively. Distributions of scores of all ensemble members are shown as filled boxplots. Boxes with lighter color filling indicate classes, where data augmentation at least doubled the number of available instances. Patches at the right edge of each plot represent weighted macro-averages of ensemble minima, mean values, and maxima for each statistic. Empty boxplots with orange median lines represent distributions of 10,000 random re-allocations of the averaged MLCM. Whiskers indicate respective maximum and minimum values for both measured and randomly generated distributions.

categories. However, the variation of scores across all ensemble members is smaller in higher abundant classes. The reason for this behavior is twofold: On the one hand, the performance in the least abundant classes is best because they are initially made up of a small number of instances that has been aggressively upsampled with similar images. Thus the model successfully learns, and probably also overfits on the training pictures (see further discussion on overfitting in Section 4.7), to detect these classes during inference. On the other hand, the number of instances in these categories in the test set is still rather small after resampling, so that single falsely classified instances can make up a substantial difference in performance metrics. For example, the test sample of cloud class $C_M = 9$ (chaotic sky in the middle level) consists of only seven instances. If one model run misclassifies one of these instances as any other class, P or R would drop from 100% to less than 86%. Similar statements can be formulated for several other categories with less than around 100 instances in the test data set. Contrary to this, pictures of the highest abundant classes exhibit substantial differences between two instances and therefore the classifier has more problems to learn to identify representative patterns for these categories. However, once the

training was successful, single misclassified instances only lead to minor variations in any score, that is, 0.25% per instance for P and R for a class with 400 instances.

Note that MCC per definition ranges in the interval $[-1, +1]$ but only the interval $[0, 1]$ is shown in Figure 4c as MCC is positive in all classes and for all ensemble members. As already mentioned, $MCC = 0$ is the expected result for a coin tossing classifier. In order to get similar baselines for P and R we calculate TP, FP, FN, and TN for each class, that is, we reduce the MLCM to one binary confusion matrix per cloud class. Afterwards, we randomly re-allocate instances within each binary confusion matrix 10,000 times and calculate P , R , and MCC for each draw separately. For the random re-allocations we followed the rules by Fowlkes and Mallows (1983), who state that values in a confidence matrix have a generalized hypergeometric distribution (Lancaster, 1969), given that column-wise and row-wise sums are fixed. Boxplots of these 10,000 re-allocations are shown as empty boxes with orange median lines in Figure 4. As for measured scores, whiskers indicate the full span of the distribution. First of all, Figure 4c confirms that the average MCC score of random allocations of the confidence matrix entries reproduces the expected value of $MCC_{\text{random}} = 0$. Moreover, maximum MCC scores of the random distribution are rather independent of the cloud class, except for class $C_M = 9$ which shows the highest maximum. Since this is the least abundant class in the test data set, probability is higher to achieve better MCC scores by mere chance. On the other hand, average P and R of the random classifications are very similar to one another and show larger variations for different classes than average MCC . Still, the span width is rather uniform throughout all classes. Classes $C_M = 6$ for the precision and $C_M = 9$ for both the precision and the recall exhibit the largest positive outliers of the random distribution.

The MLCM in Figure 3 already indicated that the mean of our classifier ensemble overall performs worst on the class $C_M = 6$. The metrics in Figure 4 confirm this statement. The worst recall score that can be found in the model ensemble for this class is even as bad as random guessing, indicated by overlapping boxplot whiskers in Figure 4b. However, neither P nor MCC show an overlap in this category, which means that in this context our model still outperforms random classifications. This is also true for all other classes and all considered metrics, that is, each single run of our model ensemble leads to superior results compared to random classifications, independent of the used metric. Furthermore, our model shows excellent performance in many aggressively augmented classes with average scores >0.80 , for example, in classes describing Cirrostratus clouds. However, also in classes which are already highly abundant in the raw data set, ensemble-averaged performance scores tend to be around 0.50 and best performing ensemble members even reach statistics up to and above 0.70. MCC scores tend to be smaller than P and R , on the one hand because of its higher range and on the other hand because it is a more elaborate statistic which considers all parts of the confusion matrix, that is, TP, FP, FN, and TN. In summary, the obtained results can be considered very good given the data situation and the task at hand.

4.3. Reliability and Resolution

As already mentioned in Section 3, attributes diagrams can be used to easily assess a model's reliability and resolution (Hsu & Murphy, 1986). Figure 5 shows attributes diagrams of AB predictions evaluated for three different cloud classes, namely $C_L = 1$, $C_M = 3$, and $C_H = 9$. Following the decomposition of BS in Equation 5, in addition to the diagonal line that indicates perfect reliability, that is, $REL = 0$, the horizontal dashed line at $\bar{o}_i = \bar{o}$ represents the *no-resolution line*. Resolution of a forecast system is optimal if $\bar{o}_i = 0$ for $y_i < \bar{o}$ and $\bar{o}_i = 1$ otherwise, which is represented by the vertical dashed line at $y_i = \bar{o}$. Following the implications of Equation 6, the attributes diagram also contains a *no-skill line* midway between the no-resolution and the diagonal line, which is represented by the skewed dashed line in Figure 5. The shaded area between no-skill and perfect-resolution lines indicates skillful regions of a forecast system compared to climatological predictions.

Attributes diagrams in Figures 5a and 5b are representative for cloud classes with high abundance in the raw data set. In such classes, our classifier achieves almost perfect calibration, that is, both very good reliability and resolution, since empirical values follow the diagonal line very closely. Contrary to this, Figure 5c represents a heavily augmented cloud class, where reliability of our model is less pronounced. Instead, the attributes diagram indicates what is called underconfidence, that is, small observed relative frequencies are overestimated by our classifier and high frequencies are underestimated. Still, resolution is also very good in these classes since the empirical curve shows large deviations from the no-resolution line. Even less abundant classes than $C_H = 9$, for instance $C_M = 9$ and $C_H = 7$ show qualitatively similar attributes diagrams. However, central probability bins in these classes are often only sparsely populated and thus values there have to be taken with caution. Panels in the second row in

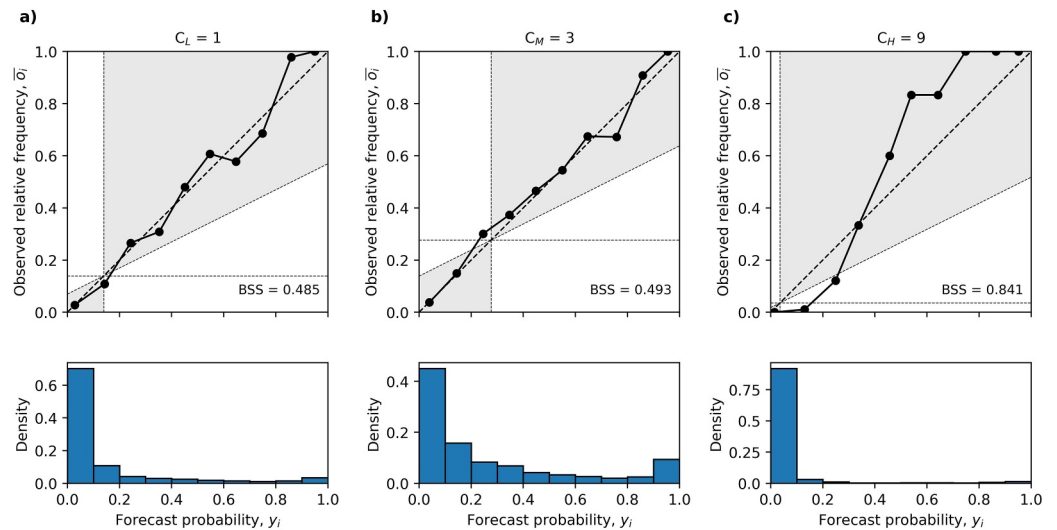


Figure 5. Attributes diagrams of micro-averaged predictions for three cloud classes, (a) $C_L = 1$, (b) $C_M = 3$, and (c) $C_H = 9$. Points along the diagonal line indicate perfect forecast reliability in the respective probability bin. The horizontal dashed lines indicate no forecast resolution. The shaded areas indicate where our model exhibits skill compared to climatological predictions and are bounded by the vertical perfect-resolution line and the skewed no-skill line. The lower panels show distributions of predicted probabilities for the respective cloud class and indicate sharpness if probabilities often deviate from the climatological observation frequency.

Figure 5 show the ratio of forecasts that fall into each probability bin. It is not surprising that a large fraction of predicted probabilities is close to 0, due to the fact that for each cloud class the number of not-observations is much higher than the number of observations. Also the u-shape of the distribution that is more or less apparent in the majority of classes is desirable, because it indicates high sharpness of our classifier, that is, predicted probabilities often deviate from the climatological frequency of occurrence. Following Equation 6, we compute the Brier Skill Score compared to climatological predictions. In general, BSS reaches values >0.80 for aggressively upsampled cloud classes (cf. Figure 5c), while higher abundant classes tend to have $BSS \approx 0.50$ (cf. Figures 5a and 5b). However, it has to be mentioned that BSS of classes with different observation frequencies cannot easily be compared. Since UNC in the denominator of Equation 6 depends exclusively on the climatological frequency of occurrence of a class, BSS can easily be increased in rare classes. Notable exceptions are categories $C_M = 4$ and $C_M = 6$ where our classifier struggles to provide both accurate and reliable predictions. Still, we obtained $BSS > 0$ in each class, indicating that our classifier ensemble outperforms climatological predictions.

4.4. Class-Averaged Statistics

Table 3 summarizes micro-averaged and macro-averaged scores for AB, MB, and MV evaluation. Macro-averages are computed unweighted and with row-wise MLCM sums as weights (w). It is evident, that for each metric and all evaluation options, that is, AB, MB, and MV, unweighted macro-average scores show the highest values. In particular, unweighted averages perform better than weighted averages, which confirms that our classifier performs better on less abundant classes. Partly, this has to be accounted to overfitting in some of the aggressively augmented categories.

Moreover, as expected $R_{\text{micro}} = R_{\text{macro},w}$, while there are small deviations between P_{micro} and $P_{\text{macro},w}$. The number of cases in the NPL column is quite large for some classes, and thus the precision scores are in all cases higher than recall scores. However, the difference is very small for MV predictions because it is very probable that at least two out of 10 members predict the same cloud class in the same instance. This decreases the number of NPL entries, counted as FN predictions, while at the same time the number of FP predictions increases, leading to an improved recall and a reduced precision compared to AB evaluation. On the other hand, AB metrics perform best in their respective averaging method over MB and MV evaluation for the precision and MCC . While $P_{\text{AB},\text{macro},w} = 0.72$ and $P_{\text{AB},\text{macro}} = 0.83$, respective P_{MB} and P_{MV} scores are by ≈ 0.10 worse. The best recall score, $R_{\text{MV},\text{macro}} \approx 0.66$, is only around 0.05 higher than all other recall scores. MCC shows values between 0.52

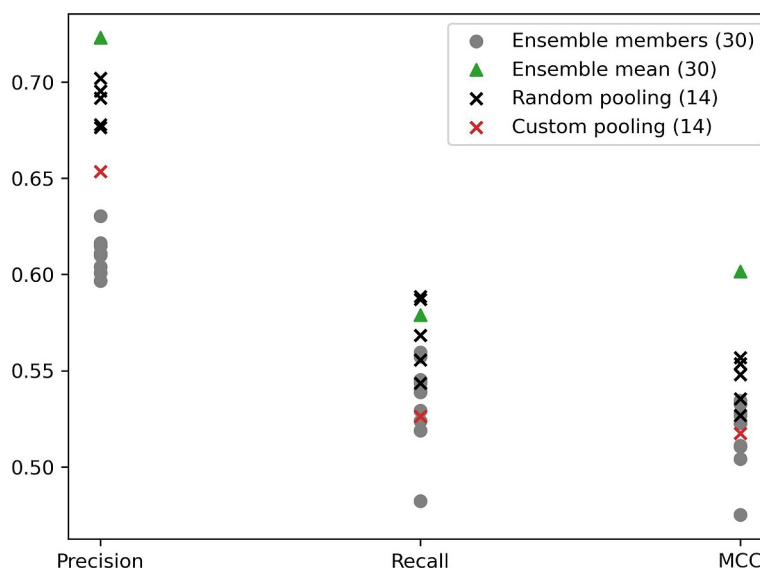


Figure 6. Comparison of Precision, Recall, and MCC for different pooling strategies with single ensemble member runs and AB weighted macro averaged ensemble mean. Red and black crosses represent custom and random pooling schemes, respectively, where the number of classes was reduced to 14. Grey circles indicate single ensemble members and green triangles show performance of the ensemble mean.

and 0.68 and again AB evaluation leads to the best results. Still, *MCC* scores are very good considering that $MCC = 0$ indicates random guessing and our classifier ensemble performs far better than that. Summarized, results indicate that there is a notable difference between the considered evaluation methods and that using ensemble mean probabilities leads to the best performance of the classifier ensemble.

4.5. Ablation Study

We also performed an ablation study, where we trained and tested our model from scratch on 14 instead of 30 classes in order to investigate whether reduced complexity in the classification scheme influences model performance. For the first experiment, we pooled classes based on physical and visual properties. For example, all high non-cirrus clouds, that is, $C_H = 5$, $C_H = 6$, $C_H = 7$, $C_H = 8$, and $C_H = 9$ were combined to a single class. In addition, we considered class abundances, for example, we did not combine $C_L = 5$ and $C_L = 8$ although they have very similar properties, because both are already highly abundant classes in the augmented data set. Apart from this custom scheme we pooled classes randomly five times. To ensure comparability, the number of pooled classes was also fixed to 14 and each pooled class had to consist of one to five SYNOP classes. In both random and custom schemes, there was a distinct class *clear sky*, consisting of instances where $C_L = 0$ and $C_M = 0$ and $C_H = 0$.

Figure 6 shows results of this ablation study compared to the performance of single ensemble members as well as the ensemble mean when trained and evaluated on the 30 SYNOP cloud classes. As expected, the model's classification performance in single runs is better when the number of classes is reduced. This is especially valid for the Precision and less so for the Recall and the MCC. Interestingly, performance on the custom classification scheme is worse than for each random run for all metrics. This suggests that visual similarities of different classes matter less than the actual label information. We consider this, at least partly, as another consequence of overfitting during the training process. On the other hand, the ensemble mean even outperforms the runs of the ablation study with Precision and MCC scores being above and Recall being almost on par with the best performing model run. Hence, it can be said that reducing the number of classes leads in general to improved classification performance of single runs but leveraging the advantages of a classifier ensemble still outperforms these results.

4.6. Investigation of Single Instances

In addition to summary scores, we also evaluated the accuracy of our classifier ensemble on single instances. Figure 7 contains representative examples of three different cases of results: In Figure 7a the model was able to

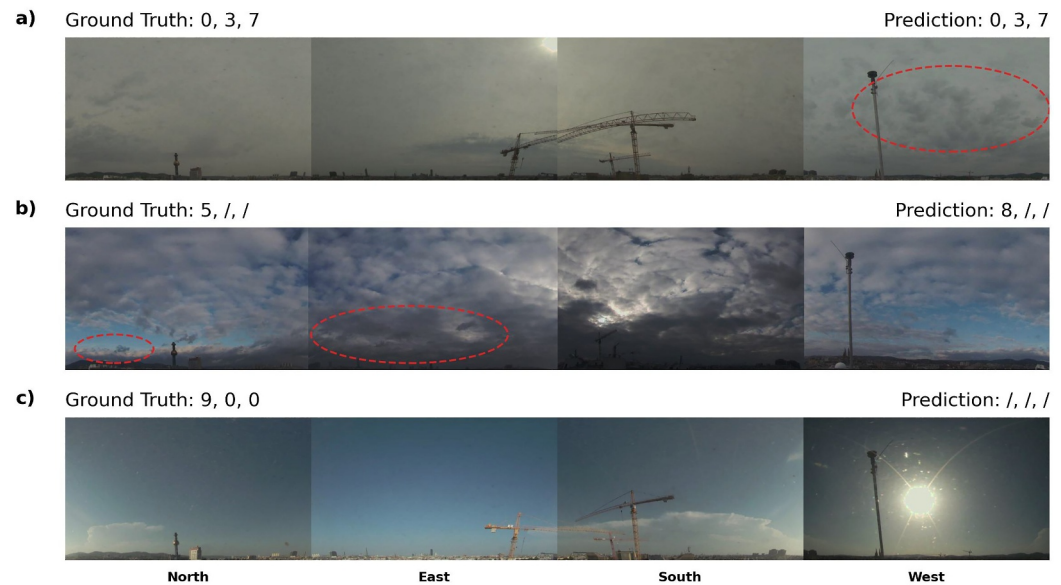


Figure 7. Comparison of ground truth and ensemble mean predictions for three different instances. In panel (a) the model perfectly reproduces the ground truth. In panel (b) the model prediction differs from the ground truth but both are reasonable classifications for the given picture. In panel (c) the model misses clearly visible Cb clouds in two different sub-images.

perfectly reproduce the ground truth containing Ac translucidus in the middle level and Cs covering the whole sky in the high level, that is, $C_M = 3$ and $C_H = 7$, respectively. Ac patches are indicated by the red ellipse but can also be seen in the sub-images facing north- and eastward. The dominating cloud veil in all sub-images is correctly classified as Cs. Although model prediction and ground truth differ in Figure 7b, it is not straightforward to tell which of them is correct. The ground truth observation is $C_L = 5$, representing Sc clouds, which are not formed by spreading out of Cu clouds. On the other hand, the model predicts $C_L = 8$, that is, a combination of Cu and Sc clouds with different base heights. Clearly, Sc clouds dominate the picture, yet there are some small patches in several sub-images, which look like Cu fractus clouds in a lower altitude level. In north- and eastward facing sub-images these patches are again circled. In this instance the ground truth seems to contain an incorrect class, while the model's prediction is correct. Still, this counts as FP prediction for class $C_L = 8$ and FN prediction for class $C_L = 5$. Eliminating similar instances with potentially flawed ground truth observations from the data set could therefore further increase the model's accuracy.

Contrary to this, in Figure 7c the model was not able to recognize any of the Cb clouds that are visible. In fact the two clouds even represent different Cb classes. While the one on the left edge of the picture does not seem to show clear fibrous parts and thus is still in an evolving stage and belongs to class $C_L = 3$, the cloud in the southward facing sub-image seems to have completed its evolution to a Cb capillatus, that is, $C_L = 9$. This is indicated by the clear presence of a fibrous anvil at its top. Since $C_L = 9$ is prioritized in the classification flowchart and there are no middle or high level clouds visible, the ground truth report is accurate. Investigation showed that 7 out of 10 model runs even predicted this class with a probability higher than $p_i = 0.50$. Still the ensemble mean probability is only approximately 0.48, which leads to a FN prediction in this instance. Hence, increasing the model's confidence in such instances is another way to reduce the number of false predictions.

4.7. Out-Of-Sample Data

We also investigated the classification performance of our model ensemble on out-of-sample pictures of our camera system. These pictures have been taken from 20.02.2023 onwards and have not been considered for the training data set. Especially pictures containing rare classes are very similar throughout the training and test data set and therefore out-of-sample data is needed to give insights into the generalization ability as well as the degree of overfitting of our model due to augmentation methods. Although this out-of-sample data set contains 2 280 instances, 16 of the 30 classes appear less than 40 times, while some others have been observed more than 500 times during the same period.

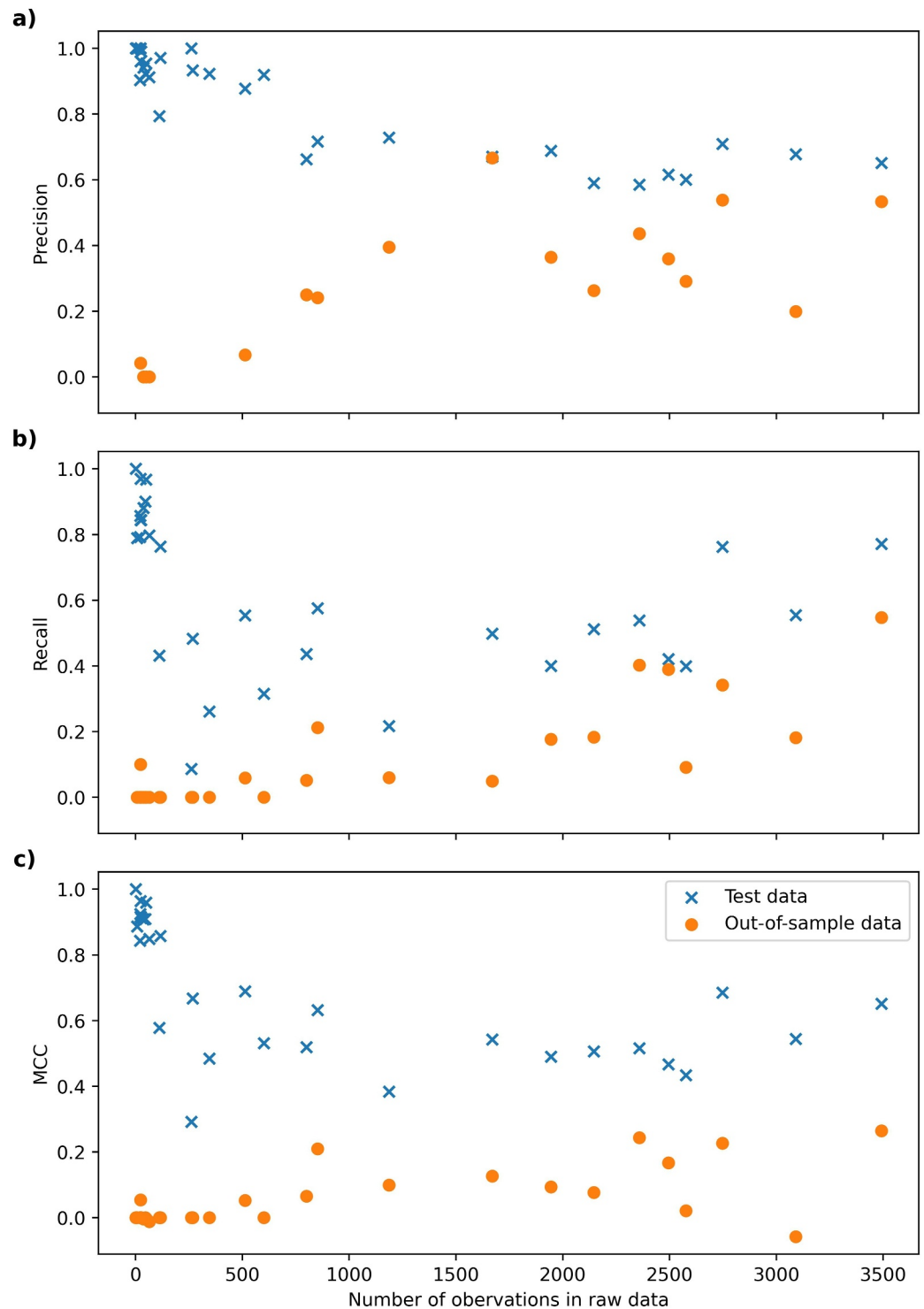


Figure 8. P_{AB} , R_{AB} , and MCC_{AB} versus number of observations in the raw data set in panels (a–c), respectively. Blue crosses and orange dots represent performance of the model ensemble on test data and out-of-sample data, respectively.

Figure 8 shows the AB performance evaluated using the test and the out-of-sample data set as crosses and dots, respectively, versus the number of observations of a class in the raw data set. As expected, statistics based on the test data are better than those on out-of-sample data for each class. Most prominent is the patch of crosses in the upper left corner of each plot, which consists of initially small and aggressively augmented classes, which are well

classified by our ensemble. However, Figure 8 also indicates that the performance difference between the two data sets is dependent on the abundance of a class in the raw data set, with out-of-sample performance being in general better for higher abundant classes. This is not surprising, since upsampling methods increase the risk of overfitting (Kaur et al., 2019). And since rare cloud classes are often observed simultaneously with more common ones, all classes have been upsampled to some degree, which could explain that these overfitting effects are also visible for classes with several thousand observations. Of the three metrics we calculated here, the precision exhibits the largest difference between low and high abundant classes. While P for the test data is close to 1 and still around 0.60 for small and large classes, respectively, the precision of AB predictions is close to 0 for the smallest classes when evaluated on the out-of-sample data. Moreover, for almost half of all classes the precision score could not be calculated at all because $TP = FP = 0$, which means that these classes have not been predicted by the model in a single instance. However, towards larger classes, P on out-of-sample data comes closer to P on test data, which indicates reduced overfitting issues in these categories. The difference between performance on out-of-sample data in small and large classes is less pronounced for the recall and MCC , since maximum values of these statistics are only around 0.40 and 0.20, respectively. However, the same is true for test data statistics, and thus the absolute difference in precision and recall values between both data sets is in general rather similar and slightly larger for MCC . Since classification performance of our model ensemble on out-of-sample data approaches that on test data for highly abundant classes, we can expect the generalization ability of such a model to increase with data set size and with more balanced category sizes.

5. Conclusions

In this work, we showed that an ensemble of 10 residual neural networks trained from scratch can accurately classify clouds in a multi-input multi-label framework, based on the operationally used WMO cloud classification scheme, where for each observation up to three out of 30 cloud classes are reported. Moreover, our classifier also shows very good reliability and resolution.

As model input we use four ground-based RGB pictures per instance, which cover close to 97% of the visible sky at the observation site in Vienna. Observation frequencies of different cloud classes within this scheme are highly imbalanced, with the least abundant class being only observed once in 3 years. Thus we developed a class-dependent upsampling strategy, where we added pictures to the data set, which have been taken up to 15 min before/after the actual time of observation of such a rare cloud class. This way no picture is used twice in the data set but similarities are still evident. We trained 10 identically initialized residual neural networks from scratch and evaluated single member (MB), ensemble mean (AB), and majority voting (MV) performance. Results show that classifier performance is good in initially highly abundant classes and exceptionally well in aggressively augmented classes. However, the latter is at least partly due to overfitting in these categories. On the other hand, the reliability and the resolution are best in initially highly abundant classes, while others tend to show under-confidence in attributes diagrams.

Overall, statistics indicate that using the ensemble mean of predicted probabilities to generate the classifier output leads to the best performance with $P_{\text{macro}} = 0.83$ and both R_{macro} and $MCC_{\text{macro}} > 0.60$. Member based evaluation and majority voting both result in lower scores, except for $R_{\text{MV, macro}}$, which is slightly above the AB score due to the fact that majority voting leads to less FN predictions. Moreover, all AB metrics are higher than the respective score of the best performing single ensemble member. Therefore, in each class the ensemble mean provides better predictions than the respectively best single member. Also, fluctuations between single members are larger in augmented classes, because their cardinality is still by far smaller and single false predictions can therefore substantially change evaluation scores.

Though these scores are below state-of-the-art results of different single-label cloud classification studies (e.g., S. Li et al., 2023), one has to consider the additional complexity of multi-label classification with 30 different categories as it is done in this work. Overlapping clouds in different height levels, visual similarities of various cloud classes, as well as the occurrence of clouds in transitional states complicate the classification process substantially and lead to both subjective and potentially erroneous classification reports. On the one hand, flawed ground truth observations introduce errors in the trained model and on the other hand the model experiences the same ambiguities during the classification process, which leads to worse forecast performance. Moreover, often there is more than one cloud class present in a height level and the WMO flow chart decides which of them is reported. Since this flow chart is not implemented in our model, a predicted class can be visible in the picture but

is still not part of the ground truth and hence considered as false prediction. Note also, that class imbalances are still present after the augmentation process, leading to a slight prediction bias towards highest abundant classes and neither weighted loss functions nor standard data augmentation methods like rotating and horizontally flipping images led to improved results.

Therefore, the results of this work with $P_{\text{macro}} = 0.83$ can be considered as very good, given the task at hand, though there is still room for improvements in further research. We have shown that impacts of highly imbalanced class cardinalities can only be reduced to some extent by upsampling. However, evaluation on out-of-sample data indicates that our model still shows the best generalization ability in initially large classes and that overfitting is for sure an issue in less abundant categories. Thus, a much larger raw data set with a sufficient number of observations of each class would be necessary to obtain unbiased predictions. Still, the reliability and the resolution of model predictions are very good and we have also shown that our model outperforms both random guessing and climatological predictions, which pleads for a well-chosen model architecture.

In future work, the model trained in this work can be extended by fine-tuning of the model weights and applying the resulting model to pictures taken at different locations to test independent and global applicability. Another approach is to add temporal information during model training by processing pictures of not only one but several subsequent time steps, so that the model learns to discriminate clouds with different origins, for example, Stratocumulus clouds which evolved from spreading of cumulus clouds versus those that did not. Adding for example, ceilometer measurements could help the model to discriminate similar cloud classes that differ by their vertical position in the sky, for example, Stratocumulus and Altopumulus clouds. We will also investigate the mentioned ambiguity in cloud classifications by comparing reports of different experts for the same time and location. If the error rate or probability could be quantified, for example, dependent on the cloud class, the CNN training process could be adapted accordingly to reduce the effect of erroneous ground truth observations, which manifest themselves in wrong model forecasts.

Data Availability Statement

Raw ground truth cloud classifications are freely available from the data hub of GeoSphere Austria (GeoSphereAustria, 2020). Ground truth observations and images, as used in this work, are publicly available in an open repository (Rosenberger, 2024a). Python code used to train and evaluate the model, and to create output plots is publicly available in another open repository (Rosenberger, 2024b) as well as on GitHub (<https://github.com/MarkusRosenberger/EnsembleCloudClassifier.git>).

Acknowledgments

We thank Claudia Plant and Lukas Miklautz from the research group Data Mining and Machine Learning at the University of Vienna as well as Irene Schicker from GeoSphere Austria and Tobias Necker from the research department at ECMWF for fruitful discussions on model architectures. We also thank two anonymous reviewers for their valuable and constructive feedback.

References

- Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. <https://doi.org/10.48550/arXiv.1607.06450>
- Ben-Baruch, E., Ridnik, T., Zamir, N., Noy, A., Friedman, I., Protter, M., & Zelnik-Manor, L. (2021). Asymmetric loss for multi-label classification. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 82–91. <https://doi.org/10.1109/iccv48922.2021.00015>
- Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1), 1–3. [https://doi.org/10.1175/1520-0493\(1950\)078<0001:VOFEIT>2.0.CO;2](https://doi.org/10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2)
- Cai, K., & Wang, H. (2017). Cloud classification of satellite image based on convolutional neural networks. In *2017 8th IEEE international conference on software engineering and service science (ICSESS)* (pp. 874–877). <https://doi.org/10.1109/ICSESS.2017.8343049>
- Calbó, J., & Sabburg, J. (2008). Feature extraction from whole-sky ground-based images for cloud-type recognition. *Journal of Atmospheric and Oceanic Technology*, 25(1), 3–14. <https://doi.org/10.1175/2007JTECHA959.1>
- Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1), 6. <https://doi.org/10.1186/s12864-019-6413-7>
- Fabel, Y., Nouri, B., Wilbert, S., Blum, N., Triebel, R., Hasenbalg, M., & Pitz-Paal, R. (2022). Applying self-supervised learning for semantic cloud segmentation of all-sky images. *Atmospheric Measurement Techniques*, 3, 797–809. <https://doi.org/10.5194/amt-15-797-2022>
- Fowlkes, E. B., & Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383), 553–569. <https://doi.org/10.2307/2288117>
- Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4), 193–202. <https://doi.org/10.1007/bf00344251>
- GeoSphereAustria. (2020). Synopdaten. Retrieved from <https://data.hub.geosphere.at/dataset/synop-v1-1h>
- Giraldo Forero, A. F., Jaramillo-Garzón, J., & Castellanos-Domínguez, G. (2015). Evaluation of example-based measures for multi-label classification performance. In *Bioinformatics and biomedical engineering* (Vol. 9043, pp. 557–564). Springer. https://doi.org/10.1007/978-3-319-16483-0_54
- Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. Retrieved from <https://arxiv.org/abs/1706.04599>
- Guzel, M., Kalkan, M., Bostanci, E., Acici, K., & Asuroglu, T. (2024). Cloud type classification using deep learning with cloud images. *PeerJ. Computer science*, 10, e1779. <https://doi.org/10.7717/peerj-cs.1779>

- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).
- Heydarian, M., Doyle, T. E., & Samavi, R. (2022). MLCM: Multi-label confusion matrix. *IEEE Access*, 10, 19083–19095. <https://doi.org/10.1109/ACCESS.2022.3151048>
- Hsu, W.-R., & Murphy, A. H. (1986). THE attributes diagram A geometrical framework for assessing the quality of probability forecasts. *International Journal of Forecasting*, 2(3), 285–293. [https://doi.org/10.1016/0169-2070\(86\)90048-8](https://doi.org/10.1016/0169-2070(86)90048-8)
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448–456). pmlr.
- Jiang, Y., Cheng, W., Gao, F., Zhang, S., Wang, S., Liu, C., & Liu, J. (2022). A cloud classification method based on a convolutional neural network for FY-4A satellites. *Remote Sensing*, 14(10), 2314. <https://doi.org/10.3390/rs14102314>
- Kaur, H., Pannu, H., & Malhi, A. (2019). A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Computing Surveys*, 52(4), 1–36. <https://doi.org/10.1145/3343440>
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25.
- Lai, C., Liu, T., Mei, R., Wang, H., & Hu, S. (2019). The cloud images classification based on convolutional neural network. In *2019 international conference on meteorology observations (icmo)* (pp. 1–4). <https://doi.org/10.1109/ICMO49322.2019.9026121>
- Lancaster, H. O. (1969). *The chi-squared distribution*. Wiley.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324. <https://doi.org/10.1109/5.726791>
- Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D. D., & Chen, M. (2014). Medical image classification with convolutional neural network. In *2014 13th international conference on control automation robotics & vision (ICARCV)* (pp. 844–848). <https://doi.org/10.1109/ICARCV.2014.7064414>
- Li, S., Wang, M., Sun, S., Wu, J., & Zhuang, Z. (2023). CloudDenseNet: Lightweight ground-based cloud classification method for large-scale datasets based on reconstructed DenseNet. *Sensors*, 23(18), 7957. <https://doi.org/10.3390/s23187957>
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018). Focal loss for dense object detection. Retrieved from <https://arxiv.org/abs/1708.02002>
- Lu, J., Tan, L., & Jiang, H. (2021). Review on convolutional neural network (CNN) applied to plant leaf disease classification. *Agriculture*, 11(8), 707. <https://doi.org/10.3390/agriculture11080707>
- Murphy, A. H. (1973). A new vector partition of the probability score. *Journal of Applied Meteorology*, 12(4), 595–600. [https://doi.org/10.1175/1520-0450\(1973\)012<0595:ANVPOT>2.0.CO;2](https://doi.org/10.1175/1520-0450(1973)012<0595:ANVPOT>2.0.CO;2)
- Nielsen, M. A. (2018). *Neural networks and deep learning*. Determination Press. Retrieved from <http://neuralnetworksanddeeplearning.com/>
- Pereira, R., Plastino, A., Zadrozny, B., & Merschmann, L. (2018). Correlation analysis of performance measures for multi-label classification. *Information Processing & Management*, 54(3), 359–369. <https://doi.org/10.1016/j.ipm.2018.01.002>
- Phung, V., & Rhee, E. (2018). A deep learning approach for classification of cloud image patches on small datasets. *Journal of Information and Communication Convergence Engineering*, 16, 173–178. <https://doi.org/10.6109/jicce.2018.16.3.173>
- Prince, S. J. (2023). *Understanding deep learning*. The MIT Press. Retrieved from <http://udlbook.com>
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention (MICCAI)* (Vol. 9351, pp. 234–241). Springer. https://doi.org/10.1007/978-3-319-24574-4_28
- Rosenberger, M. (2024a). Data for Deriving WMO cloud classes from ground-based RGB pictures with a residual neural network ensemble [Dataset]. *Zenodo*. <https://doi.org/10.5281/zenodo.14185063>
- Rosenberger, M. (2024b). Python Code for Deriving WMO cloud classes from ground-based RGB pictures with a residual neural network ensemble [Software]. *Zenodo*. <https://doi.org/10.5281/zenodo.14185529>
- Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. *Proc. CVPR*, 815–823. <https://doi.org/10.1109/cvpr.2015.7298682>
- Sommer, K., Kaban, W., & Brunet, R. (2024). Infrared radiometric image classification and segmentation of cloud structure using deep-learning framework for ground-based infrared thermal camera observations. *egusphere*. <https://doi.org/10.5194/egusphere-2024-101>
- Taravat, A., Del Frate, F., Cornaro, C., & Vergari, S. (2015). Neural networks and Support vector machine algorithms for automatic cloud classification of whole-sky ground-based images. *IEEE Geoscience and Remote Sensing Letters*, 12(3), 666–670. <https://doi.org/10.1109/LGRS.2014.2356616>
- Tian, B., Shaikh, M., Azimi-Sadjadi, M., Haar, T., & Reinke, D. (1999). A study of cloud classification with neural networks using spectral and textural features. *IEEE Transactions on Neural Networks*, 10(1), 138–151. <https://doi.org/10.1109/72.737500>
- Wang, M., Zhou, S., Yang, Z., & Liu, Z. (2020). CloudA: A ground-based cloud classification method with a convolutional neural network. *Journal of Atmospheric and Oceanic Technology*, 37(9), 1661–1668. <https://doi.org/10.1175/JTECH-D-19-0189.1>
- Wilks, D. S. (2019). *Statistical methods in the atmospheric sciences* (4th ed.). Elsevier Academic Press.
- WMO. (2017). Manual on the observation of clouds and other meteors – International cloud Atlas. Retrieved from <https://cloudatlas.wmo.int>
- Wohlfarth, K., Schröder, C., Klaub, M., Hakenes, S., Venhaus, M., Kauffmann, S., et al. (2018). Dense cloud classification on multispectral satellite imagery. In *2018 10th iapr workshop on pattern recognition in remote sensing (prrs)* (pp. 1–6). <https://doi.org/10.1109/PRRS.2018.8486379>
- Xia, M., Lyu, W., Yang, J., Ma, Y., Yao, W., & Zheng, Z. (2015). A hybrid method based on extreme learning machine and k-nearest neighbor for cloud classification of ground-based visible cloud image. *Neurocomputing*, 160, 238–249. <https://doi.org/10.1016/j.neucom.2015.02.022>
- Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1/2), 69–90. <https://doi.org/10.1023/a:1009982220290>
- Zhang, J., Pu, L., Zhang, F., & Song, Q. (2018). CloudNet: Ground-Based cloud classification with deep convolutional neural network. *Geophysical Research Letters*, 45(16), 8665–8672. <https://doi.org/10.1029/2018GL077787>
- Zhao, M., Chang, C. H., Xie, W., Xie, Z., & Hu, J. (2020). Cloud shape classification system based on multi-channel CNN and improved FDM. *IEEE Access*, 8, 44111–44124. <https://doi.org/10.1109/ACCESS.2020.2978090>
- Zhu, Y., & Newsam, S. (2017). DenseNet for dense flow. Retrieved from <https://arxiv.org/abs/1707.06316>