Hydrometeor Classification and riming degree estimation from Multi-Angle Snowflake Camera images

Marie-Alix GILLYBOEUF, Baptiste HERNETTE, Gaspard VILLA
EPFL, CS-433, Machine Learning

***Abstract*—The purpose of this study is to find a machine learning method to classify hydrometeors and to estimate riming degree, based on images collected with a Multi-Angle Snowflake Camera (MASC). The dataset consist of grayscale images for which there are geometric and texture descriptors to classify hydrometeors among six predefined classes, and estimate the degree of riming among five classes. These two classification problems are solved using different supervised machine learning methods. The best method for our classification tasks achieved accuracy scores of 0.90 and 0.70 for respectively hydrometeors classification and riming degree estimation.**

I. INTRODUCTION

Hydrometeors consist of liquid or solid water particles. When they are deposited on a surface, they are considered as snow or water. Their micro-structures give important quantitative information about the evolution and formation of precipitations, which are useful for radar applications. Today, it is more than necessary to explore individual particle properties to improve quantitative estimation and get better characterization of the microphysics behind snowfall. One way to do so is by classifying particles according to their hydrometeor class and their riming degree. To solve these classification tasks, snowflake images were collected with a Multi-Angle Snowflake Camera (MASC). This method is a ground-based snowflake imager that takes three pictures of a falling snowflake using different angles of three cameras. The high-resolution of these pictures allow to observe particles individually, to manually labeled them, and then to classify them in a supervised approach.
In this project, we want to generate a model that predicts hydrometeor classes and estimates riming degrees based only on geometric and textural features, without taking into consideration speed or environmental conditions. We focus our study on four machine learning methods: *Multinomial Logistic Regression* (MLR), *Support Vector Machine* (SVM), *Random Forest Classifier* (RFC) and *Multilayer Perceptron* (MLP). Each of these method is used to first classify hydrometeors and then estimate riming degree of the particles.

II. METHODS

A. Feature selection

First, we need to have an efficient method for the feature selection. The goal of feature selection is to reduce the number of features in order to decrease the computational cost and to improve performance of models. There are different methods that can be used to select the subset of the most relevant features, such as *Principal Component Analysis* (PCA), *Lasso* or *Recursive Feature Elimination* (RFE) algorithms.

PCA is a feature extraction technique that maps data into a new feature space. This method removes noise by reducing the number of features to a lower number of Principal Components (PC). PC are orthogonal vectors that explain an amount of variance in the original features. Each successive PC explains variance that is left after its preceding component and the first PC explains the greatest amount of variance. Generally, we select the number of components that altogether explain a chosen amount of variance. PCA is easy to compute, however, due to the change in feature space, analysis and interpretation are more difficult.
Thus, another widely use method for feature selection is *Lasso*. It works on a standardized dataset and automatically select useful features and discard useless or redundant features. To select features, a cost function with $\mathcal{L}_1$ penalty is used and α is a hyperparameter that tunes the intensity of this penalty term. We usually use cross-validation approach to find α. This method has drawbacks since it creates a linear model which is not optimal if the relationship between features and target is not linear.
A third way to do feature selection is by using RFE method. The algorithm starts with all features and removes feature until the desired number remains. At each step, a measure of variable importance that ranks predictors by order of importance is computed and the least important predictors are iteratively eliminated. There are two configuration options for this method: choosing the number of features to select and selecting the algorithm used to chose features. These two configurations can be explored by cross-validation approach.

B. Classification

To solve the two classification tasks, we need to implement supervised machine learning algorithms. First of all, we apply a data standardization to convert all data in a common format in order to clarify have a better understanding of the data. Then, by plotting an histogram of the distribution of the data between the different classes (either for hydrometeor or riming degree dataset), we notice that it is significantly highly

imbalanced. This is taken into account in each individual method by rebalancing strategies where values of y are used to adjust weights inversely proportional to class frequencies in the input data. Additionally, to overcome this imbalanced problem, we perform data augmentation to get approximately the same amount of data in each class. To do so, we generate data by synthetic minority oversampling technique (SMOTE) which creates artificial data based on K-nearest neighbors and on the feature space similarities.

Multinomial Logistic Regression (MLR) is a well-known machine learning method used to solve supervised multi-classification problems. It works directly on the original feature spaces which enable a direct interpretation of regression weights. It is an extent of Logistic Regression, a statistical probabilistic model for binary classification.MLR assigns to each observation a probability of belonging to each class of the model according to a specific estimator. To avoid overfitting and to decrease dependency of the final model on the training set, a regularization term is added. This penalty term can be either $\mathcal{L}_1$ or $\mathcal{L}_2$. The strength of the regularization is controlled by the hyperparameter λ, where a small λ means no regularization and a large one means strong regularization. Both penalty and λ are needed to be tuned by cross-validation in order to select the best parameters for the model. Moreover, as we are dealing with imbalanced problem it is needed to rebalance classes in order to avoid biased of the MLR toward majority classes.

Another widely used method for classification objectives is the *Support Vector Machine* (SVM) algorithm. It aims to find hyperplanes in a N-dimensional space, where N is the number of features, to distinctly classify the data points. The hyperplanes are defined such that they maximize the margin (*i.e.*, maximum distance between data points of two different classes). The loss function used to maximize margins is the hinge loss. A regularization term is added to balance the margin maximization and the loss. As in MLR, the strength of this regularization is controlled by λ. The shape of the hyperplane is defined by the kernel that the SVM algorithm uses (linear, polynomial, radial basis function (RBF) or sigmoid). Both kernel and λ are needed to be tuned by cross-validation in order to select the best parameters for the model.

To increase the accuracy score, it is common to try to fit the data with more flexible and non-linear methods. That is why, we introduce *Random Forest Classifier* (RFC), one of the most popular learning algorithms. This method consists of creating a collection of multiple decision trees applied on the training set. Decision tree is a learning technique frequently used in classification problems. It is a tree-structured classifier where each internal nodes identifies features of the dataset, branches describe the decision rules and leaf nodes represent the outcome. For this method, the hyperparameters to tuned with cross-validation are: the number of decision trees in the forest, the maximum depth of each decision tree, the minimum number of samples needed to split an internal node in the decision tree and the minimum number of samples required to be at a leaf node.

Lastly, an alternative approach, that is frequently more efficient, is developping a neural network model. For classification problem, we use a *Multilayer Perceptron* (MLP) classifier, which is a feed-forward artificial neural network. This method consists of input layers for each feature and output layers for each class in the classification problem but, unlike *Logistic Regression*, there are additional intermediate hidden layers. Each of these nodes is a neuron that applies a nonlinear activation function that matches the weighted input to the output. Hyperparameters to be tuned using cross-validation are the activation function for each hidden layer (*tanh* or *ReLU*), the regularization term, α, for $\mathcal{L}_2$ penalty, the solver for the weight optimization problem (*Stochastic Gradient Descent* or *Adam*), the learning rate associated to the solver and its schedule for weight updates (either constant or adaptive).

Furthermore, to find the optimal hyperparameters for each models, we need to apply cross-validation. This technique consists in splitting the dataset into k-folds. Then, a training is performed on $k-1$ folds with some particular hyperparameters and the model is tested on the remaining fold. Iterating this process with each fold taking as the test set, we compute the mean test accuracy and the variance of the test accuracy in order to compared the different models. Hence finding the best hyperparameters of the model with the better mean test accuracy and also with the lower variance.

III. Hydrometeors Classification

The goal of hydrometeors classification is to categorize precipitations into six qualitative classes, in order to define the dominant class in a given volume. These six classes are: *Small particle* (SP), *Columnar crystal* (CC), *Planar crystal* (PC), *Combination of column and plate crystal* (CPC), *Aggregate* (AG) and *Graupel* (GR) which are represented on Fig. 1 using MASC technique.

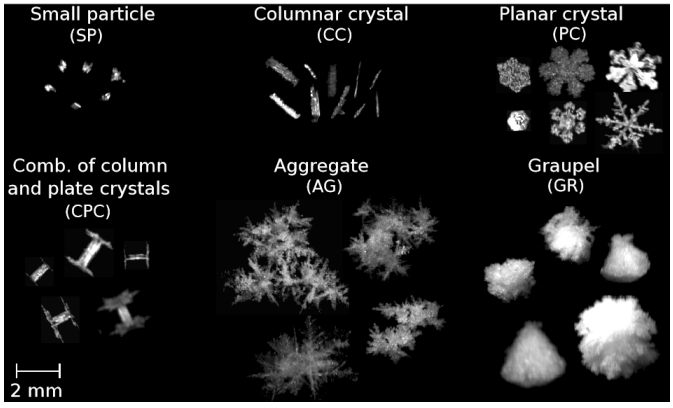

Fig. 1: Images (collected with MASC) of the six different hydrometeor classes [1]

A. Data Preprocessing

Performance of the hydrometeor classification is evaluated on an initial dataset of $\mathcal{N}$=1896 data sample manually labeled by human. Exploring the dataset, we observe that some snowflakes are assigned to two different labels, making their correct classification impossible (Fig. 2).

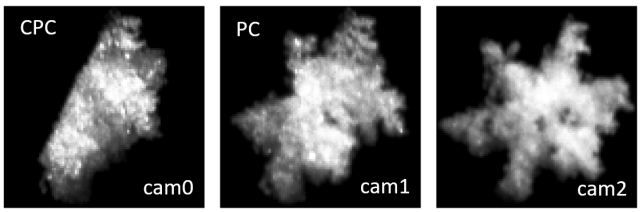

Fig. 2: Miss-classification of a snowflake in two different classes [2]. Due to the the angle of each camera: cam1 and cam2 show that this flake is a PC but cam0 induced the mistake by assigning it to class CPC.

After having these miss-classified snowflakes removed, we standardize and split the labeled dataset into a training set ($\mathcal{N}_{train} = 1608$) which is used to fit and tune the hyperparameters of the model, and a testing set ($\mathcal{N}_{test} = 403$) which is used to evaluate and validate the predicting classification on unseen data. To get rid of redundant and irrelevant features, we reduce the initial set of 72 features to a final set of 30 features using the *Lasso* method. By looking deeper at the distribution of the data into the different classes, we notice that class SP and GR are undersampled compared to other classes. To rebalance these two classes, we apply SMOTE procedure (see Section II) on the training set which leads to and augmented set $\mathcal{N}_{train} = 3006$.

B. Model selection

For each of method presented in Section II, we follow cross-validation procedure on the augmented training set to tune their respective hyperparameters. The classification accuracy is then assessed by fitting the model on the test set with the selected best hyperparameters for the model. The accuracy scores obtained on the training set and the testing set for each models are reported in the following Table I:

Methods	Train accuracy [%]	Test accuracy [%]
Multinomial Logistic Regression	91.33 ± 1.05	88.66 ± 0.90
Supervised Vector Machine	99.25 ± 0.22	89.01 ± 1.39
Supervised Vector Machine polynomial	98.47 ± 0.36	87.16 ± 2.02
Random Forest	100	87.76 ± 1.66
Multi-layer Perceptron	99.98 ± 0.02	87.86 ± 1.66

TABLE I: Average accuracy scores obtained with each method for the hydrometeor classification. Numbers indicate the mean ± the standard deviation calculated over 5-fold cross validation and applied on test (*i.e.*, unknown) data.

We clearly see that the MLR model achieves the best accuracy score (88.66%) without overfitting the training set. The confusion matrix (Fig. 3) assesses that all individual classes are well represented and that none of them are neither overfitted nor underfitted. The MLR model therefore confirmed that the selected features are discriminating well between the classes and are relevant to the hydrometeor classification task.

As MLR is a linear model and works directly in the original feature space without any remapping, each weight can be used to assess the importance of the associated feature. Fig. 7 displays the obtained results. An interesting remark is that that among the 14 features with an importance score higher than 0.5, 3 use a textural information and two of them are actually the most important for the model.

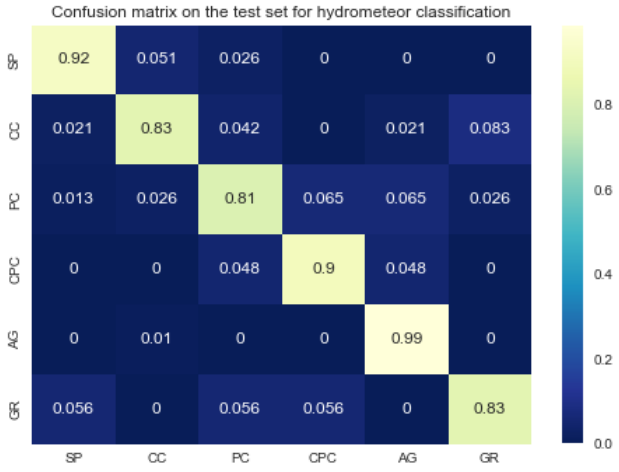

Fig. 3: Confusion matrix of test set obtained for hydrometeor classification with MLR method. True labels are on the horizontal axis and predictions on the vertical axis. Correct classifications are on the diagonal. Entries have been normalized.

IV. Riming Degree Estimation

In this part we aim to quantify the presence of cloud frozen droplets on the surface of the particles. To do so, a continuous riming index $\mathcal{R}_c \in [0, 1]$ is introduced. Then, a riming degree $\mathcal{R}_d$ is assigned to each particle according to its value of $\mathcal{R}_c$. There are five qualitative classes: *none* ($\mathcal{R}_c = 0$), *rimed* ($\mathcal{R}_c = 0.15$), *densily rimed* ($\mathcal{R}_c = 0.5$), *graupel-like* $\mathcal{R}_c = 0.85$) and *graupel* ($\mathcal{R}_c = 1$) which are represented on Fig. 4.

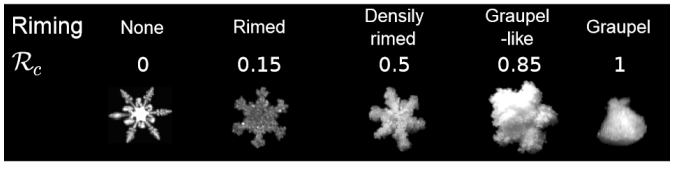

Fig. 4: MASC images of the different riming degrees [1].

A. Data Preprocessing

Performance of the riming degree estimation is evaluated on an initial dataset of $\mathcal{N} = 2011$ data sample manually labeled by human. As in hydrometeor classification dataset, we observe

that some snowflakes have two different labels, making their estimation impossible (Fig. 5).

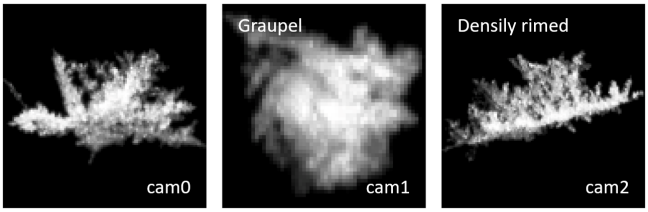

Fig. 5: Miss-classification of a snowflake in two different classes [2]. Due to the the angle of each camera: cam0 and cam2 show that this flake is a *graupel* but cam1 induced the mistake by assigning it to class *densily rimed*.

We remove these miss-classified snowflakes and we standardize and split the labeled dataset into a training set ($\mathcal{N}_{train} = 1516$) and a testing set ($\mathcal{N}_{test} = 380$). As for hydrometeor classification, we use the training set to fit and tune the model parameters and we use the testing set to evaluate and validate the classification on unseen data. The initial set of 72 features is reduced to a final set of 60 features using the *Lasso* method to get rid of irrelevant and redundant features. When we look at the distribution of the data into the different classes, we notice that class *none* and *graupel-like* are undersampled compared to other classes. To rebalance these two classes, we apply SMOTE procedure (see Section II) on the training set which leads to and augmented set $\mathcal{N}_{train} = 2400$.

B. Model selection

By following the same step as the ones followed for hydrometeor classification we obtain the accuracy scores on the training set and the testing set for each models. The results are reported in the following Table II:

Methods	Train accuracy [%]	Test accuracy [%]
Multinomial Logistic Regression	76.20 ± 1.13	72.20 ± 2.13
Supervised Vector Machine	96.98 ± 0.86	74.20 ± 2.70
Supervised Vector Machine polynomial	95.27 ± 1.54	71.72 ± 2.14
Random Forest	100	73.57 ± 1.87
Multi-layer Perceptron	100	73.84 ± 3.08

TABLE II: Average accuracy scores obtained with each method for the riming degree estimation. Numbers indicate the mean ± the standard deviation calculated over 5-fold cross validation and applied on test (*i.e.*, unknown) data.

As for the hydrometeor classification task, the MLR model achieves the best accuracy score (72.20%) and avoids overfitting. However, one can see that these values are significantly worse than for the hydrometeor classification model. On the associated confusion matrix (Fig. 6) we observe that most of the misclassification are located next to the diagonal, meaning that the classifier discriminates well between *non-rimed*, *rimed* and *gtraupel* particles but there is an uncertainty of ± 1 level in the predictions.

As previously, each weight can be used to assess the importance of the associated feature. Fig. 8 displays the obtained results. We remark that that among the 35 most important features kept, 8 use a textural information. As for hydrometeor classification, the two most important features use textural information.

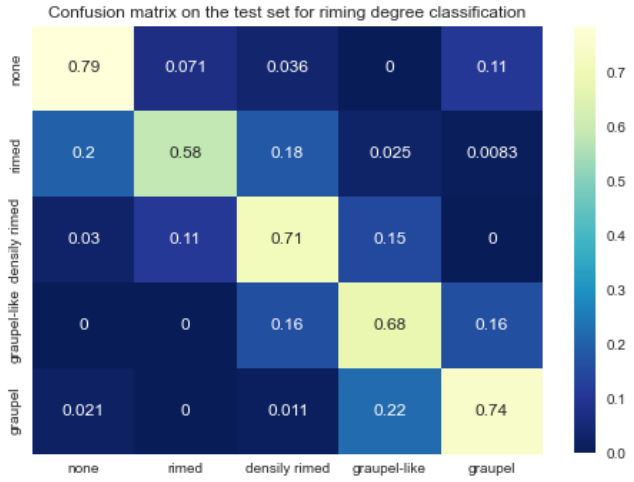

Fig. 6: Confusion matrix of test set obtained for riming degree estimation with MLR method. True labels are on the horizontal axis and predictions on the vertical axis. Correct classifications are on the diagonal. Entries have been normalized to sum up to 100.

V. Conclusion

To conclude, through this study we show that we generate models to classify hydrometeors and estimate riming degree. On both data set we do standardization, remove the miss-classified snowflakes, we balance classes and we split into a training and a testing set. Then, training set is used to fit and tune hyperparameters of each method using cross-validation approach. The testing set is then used to evaluate and validate the model. Doing this, MLR is found to be the best models for both classification problems, which achieves an accuracy score of 88.66% and 72.20% respectively for each task.
By looking at the feature importance for the model of each classification, we notice that textural information is important in estimating the riming degree and in classifying hydrometeor. A way to assess and verify that would be to remove textural features form each model and see whether the accuracy score is significantly decreased or not. Moreover, we can think about trying to do a Convolutional neural network (CNN), on the images directly instead of values, to improve our accuracy score in both classification problems.

VI. References

[1] Praz, C., Roulet, Y.-A., and Berne, A., Solid hydrometeor classification and riming degree estimation from pictures collected with a Multi-Angle Snowflake Camera, (2017). Atmos. Meas. Tech., 10, 1335–1357 https://doi.org/10.5194/amt-10-1335-2017.

[2] Ghiggi, G. and Grazioli, J., Github : MASC_DB API - An API to query MASC data.https://github.com/ltelab/pymascdb

[3] Ghiggi, G. and Grazioli, J., mascdb's documentation (2021). Retrieved from https://pymascdb.readthedocs.io/en/latest/index.html

[4] H. He and E. A. Garcia, Learning from Imbalanced Data, in IEEE Transactions on Knowledge and Data Engineering (2009), vol. 21, no. 9, pp. 1263-1284, http://dx.doi.org/10.1109/TKDE.2008.239

[5] Pedegrosa, F., Varoquaux, G., Gramfort, A., et al., Scikit-learn: Machine Learning in Python, in Journal of Machine Learning Research 12 (2011) 2825-2830.

Appendix

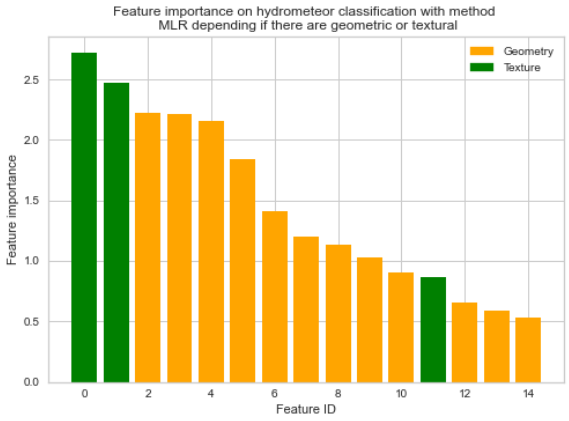

Fig. 7: Importance of each selected feature according to the logistic weights for hydrometeor classification. Feature ID are in Table III

Feature ID	Feature description
0	local_intens
1	intensity_mean
2	ell_in_B
3	sym_mean
4	quality_xhi
5	skel_perim_ratio
6	convexity
7	sym_Pmax_id
8	sym_P2
9	sym_std_mean_ratio
10	skel_area_ratio
11	hist_entropy
12	area_porous
13	frac_dim_theoretical
14	area

TABLE III: List of features with importance score higher than 0.5, provided to hydrometeor classification model. To have a better description of the corresponding features refer to [3]

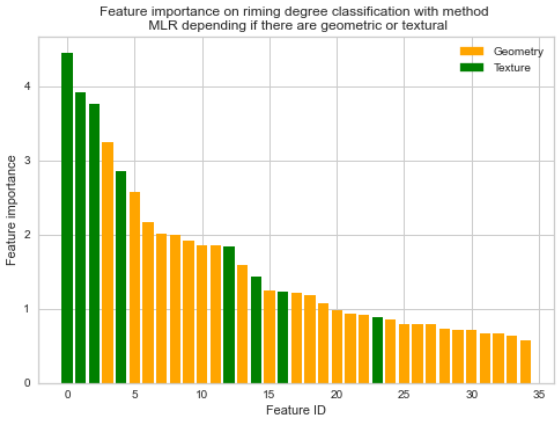

Fig. 8: Importance of each selected feature according to the logistic weights for riming degree estimation. Feature ID are in Table IV

Feature ID	Feature description
0	lap_energy
1	intensity_mean
2	wavs
3	p_circ_out_r
4	intensity_std
5	convexity
6	frac_dim_theoretical
7	area
8	ell_fit_B
9	sym_P6_max_ratio
10	Dmean
11	ell_fit_a_r
12	complexity
13	sym_P6
14	local_std
15	sym_P2
16	local_intens
17	rect_perim_ratio
18	ell_fit_ecc
19	area_porous
20	sym_std_mean_ratio
21	sym_mean
22	ell_fit_area
23	hist_entropy
24	ell_in_area
25	ell_in_A
26	rect_aspect_ratio
27	skel_perim_ratio
28	skel_N_ends
29	rectangularity
30	ell_in_B
31	skel_area_ratio
32	har_hom
33	ell_fit_A
34	compactness

TABLE IV: List of features with importance score higher than 0.5, provided to riming degree estimation model. To have a better description of the corresponding features refer to [3]