

Linear Discriminant Analysis

Machine learning under physical constraints

Ines BESBES

Contents

1	Introduction	2
2	Loading data	2
3	Boundary conditions	3
4	Linear Discriminant Analysis (LDA) with the empirical mean	4
4.1	Training and test data means	4
4.2	Classification using LDA	5
5	Linear Discriminant Analysis (LDA) with the empirical autocovariance matrix	5
5.1	Understanding the autocovariance definition	5
5.2	Estimation of autocovariance matrix from single images	6
5.3	Classification using LDA	6
6	Linear Discriminant Analysis (LDA) with the power spectrum (periodogram)	6
6.1	Relationship between power spectrum and autocovariance	6
6.2	Analysis of full spectrum vs truncated spectrum for different <i>dom</i> values	7
6.3	Classification using LDA	9
7	Performance analysis	9
7.1	Results with the empirical mean	9
7.2	Results with the empirical autocovariance matrix	10
7.3	Results with the power spectrum	12
7.4	Exploring invariance in feature representation	13
8	Image classification using scattering features	13
8.1	Performance analysis	14
9	Conclusion	15

1 Introduction

In this report, we aim to classify simulations of the interstellar medium using Linear Discriminant Analysis with different statistical estimators. We will explore three methods:

- The empirical mean
- The empirical autocovariance matrix
- The power spectrum

We will compare the performance of these methods to determine which estimator provides the most accurate and reliable classification of interstellar medium simulations.

2 Loading data

In this study, we utilize simulated maps of the interstellar medium, each with dimensions of **256x256 pixels**. These simulations vary based on two physical parameters: the magnetic field and temperature, resulting in **nine distinct classes**.

Moreover, the data is organized into two categories: training and testing. The training set, used for model training, contains **20 images** per class, while the testing set, reserved for validation, includes **15 images** per class.

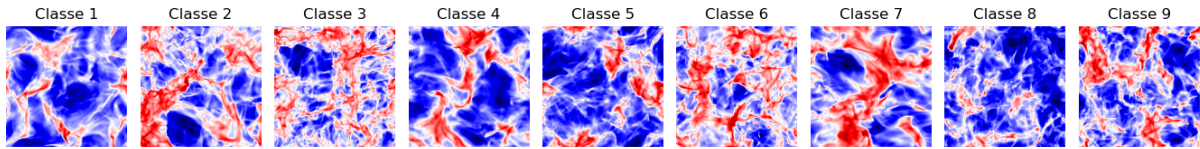


Figure 1: Example of image from each class

When we look at the images, it is tough to tell which class each image belongs to just by eye because they all look quite similar.

3 Boundary conditions

We used the numpy *roll* function to shift the image pixels either vertically or horizontally by a set number. The shift parameter determines how many positions the elements will move. In our example, we set the shift at **60**. The images obtained will help us understand the boundary conditions.

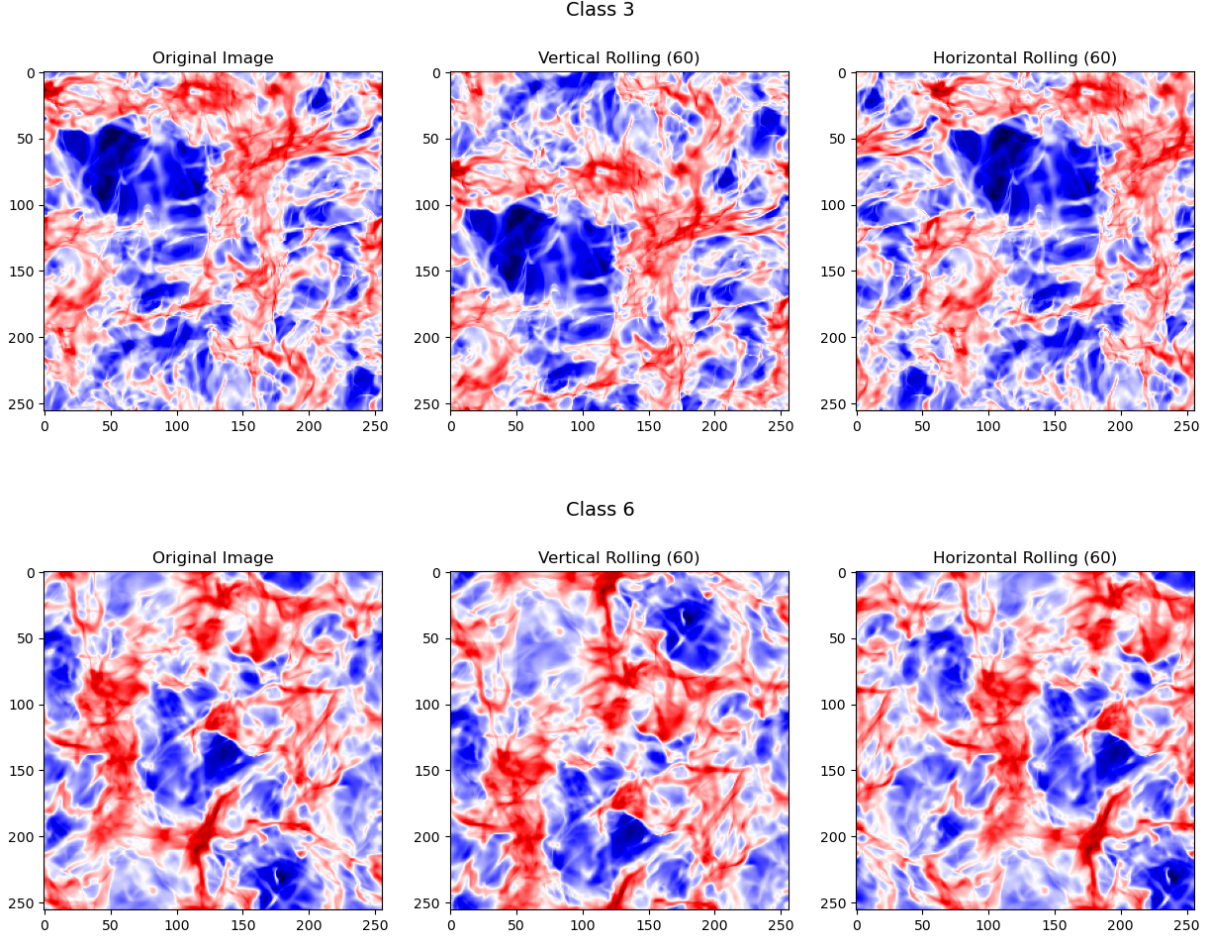


Figure 2: Applying vertical and horizontal rollings to images from classes 3 and 6

We observe that the images use **periodic boundary conditions**. This means that when we roll the image either vertically or horizontally, the parts that move beyond one edge of the image reappear on the opposite edge. This boundary condition help maintain the **continuity** of patterns in the images, which is crucial for accurately simulating and analyzing the interstellar medium.

4 Linear Discriminant Analysis (LDA) with the empirical mean

4.1 Training and test data means

We used the function *compute_features_mean* to calculate the mean value of each image in a class.

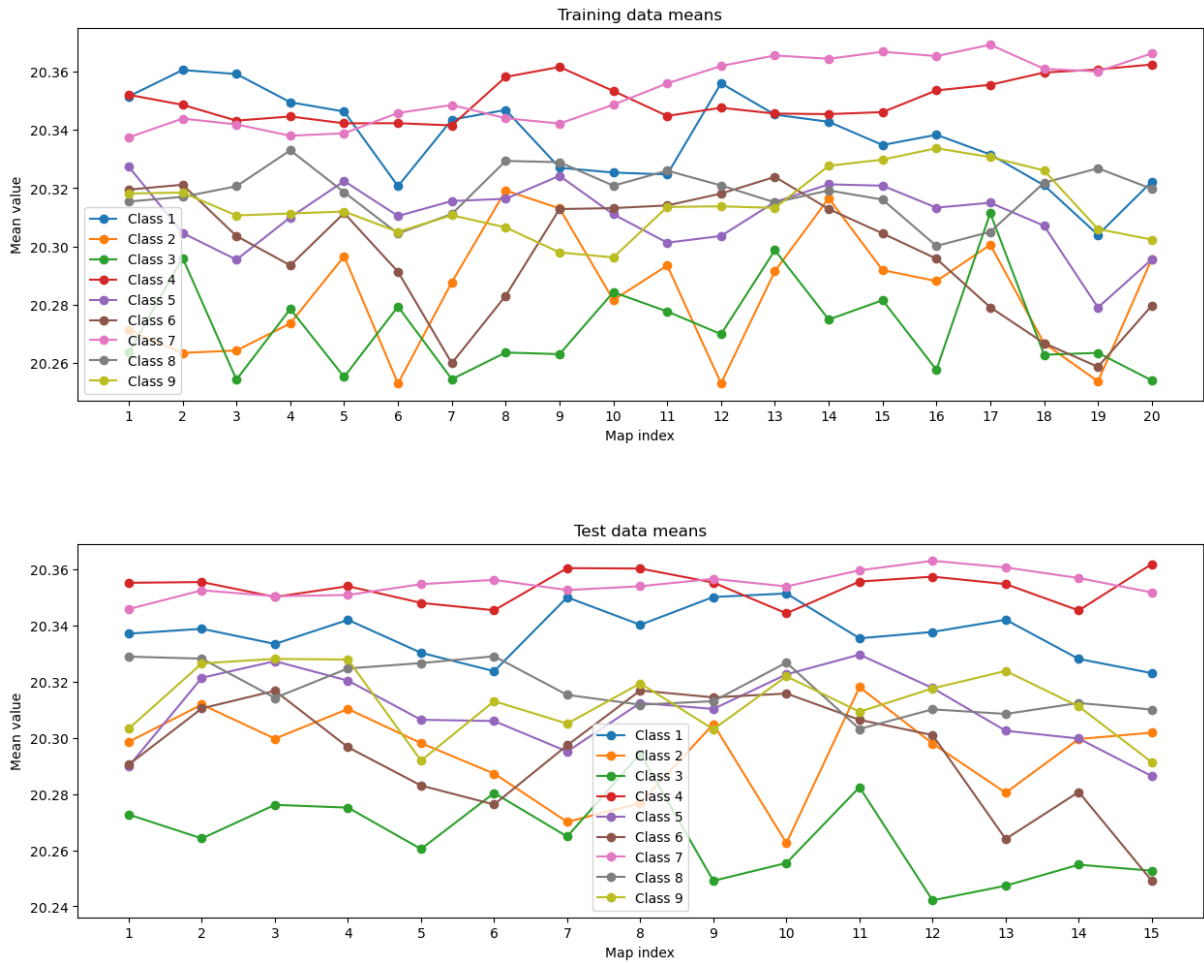


Figure 3: Training and test data means for each class

When we look at the plots of the mean values for both the training and test data, we notice that the mean might not be the best marker for classifying these images. In fact, the mean values for the different classes often **overlap and don't show clear separation**, which makes it hard to tell the classes apart using just the mean.

4.2 Classification using LDA

We implemented an LDA classifier to evaluate how well the mean values could classify the images. The LDA classifier was trained on the mean features of the training data and then tested on the test data.

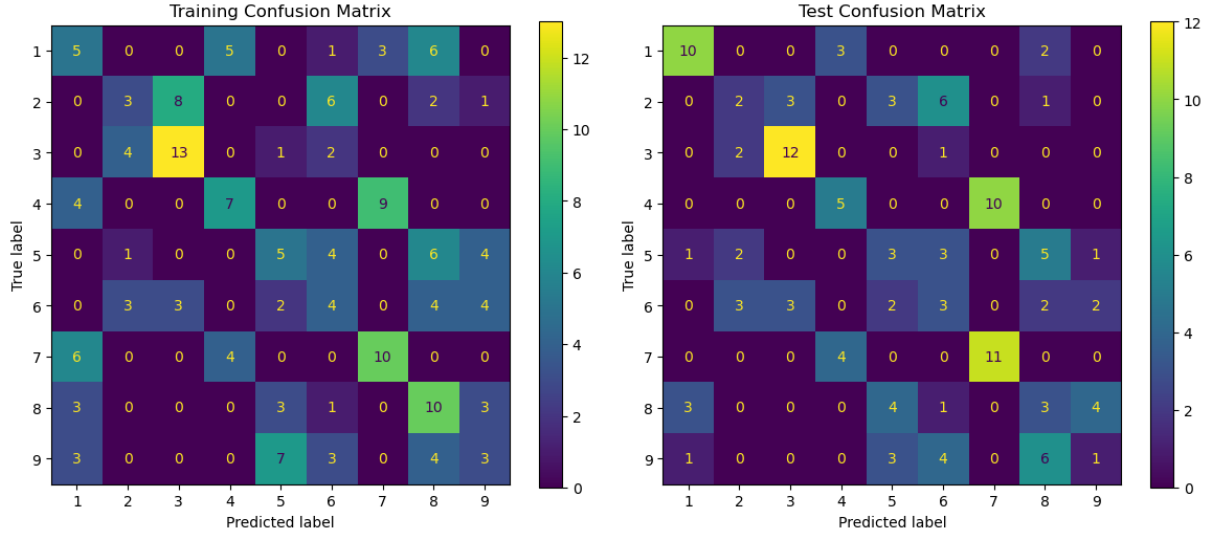


Figure 4: Training and test confusion matrices

The results from using the mean as a feature for classification with Linear Discriminant Analysis show **low accuracy**, with training accuracy at 0.33 and test accuracy at 0.37. This suggests that the mean alone is not a good marker for this data.

On the other hand, the confusion matrices show that there is a lot of overlap and many mistakes in classifying the different groups. This means that using only the mean value does not give enough unique information to correctly classify the images. Thus, we might need to use different methods, which is what we will do in the next part.

5 Linear Discriminant Analysis (LDA) with the empirical autocovariance matrix

Let $\tau = (\tau_1, \tau_2)$ be the translation vector with $\tau_1 = 0, 1, \dots, dn$ and $\tau_2 = -dn, \dots, -1, 0, 1, \dots, dn$. Let $X(u)$ denote the value of the image X at pixel u . The autocovariance of the process X (assumed to be real-valued) is defined by:

$$\phi(u, \tau) = \mathbb{E}[(X(u) - \mu_u)(X(u - \tau) - \mu_{u-\tau})]$$

5.1 Understanding the autocovariance definition

Here, \mathbb{E} represents the expected value or the mean of a random variable. It is used to denote the average value that a random process would take over many trials.

μ_u and $\mu_{u-\tau}$ represent the mean values of the image X at pixel u and $u - \tau$, respectively. These means are calculated as $\mu_u = \mathbb{E}[X(u)]$ and $\mu_{u-\tau} = \mathbb{E}[X(u - \tau)]$.

The autocovariance depends only on τ if the process X is *stationary*, which means that its statistical properties, such as mean, variance, and autocovariance, do not change over time or space. In the context of images, this means that the statistical properties are **invariant under translations**.

In our case, if the image process X is stationary, the autocovariance $\phi(u, \tau)$ will depend only on τ and not on u . This justifies the term **"autocovariance matrix"**, as it can be represented as a matrix where each entry corresponds to a specific translation τ .

Given the choice of dn , the dimensions of the autocovariance matrix will be $(2dn + 1) \times (2dn + 1)$. This is because τ_1 ranges from 0 to dn and τ_2 ranges from $-dn$ to dn , resulting in a total of $2dn + 1$ possible values for each dimension.

5.2 Estimation of autocovariance matrix from single images

In this section, we propose an estimator for the autocovariance matrix calculated from a single image. The goal is to compute the autocovariance matrix for each image in a given class, using a specified displacement parameter dn . The autocovariance matrix captures the statistical relationship between the pixel values of an image and its shifted versions.

We implemented the function `compute_features_cov` to calculate these autocovariance matrices. For each image, the function shifts the image by various translations (τ_1, τ_2) and computes the covariance of the differences from the mean.

5.3 Classification using LDA

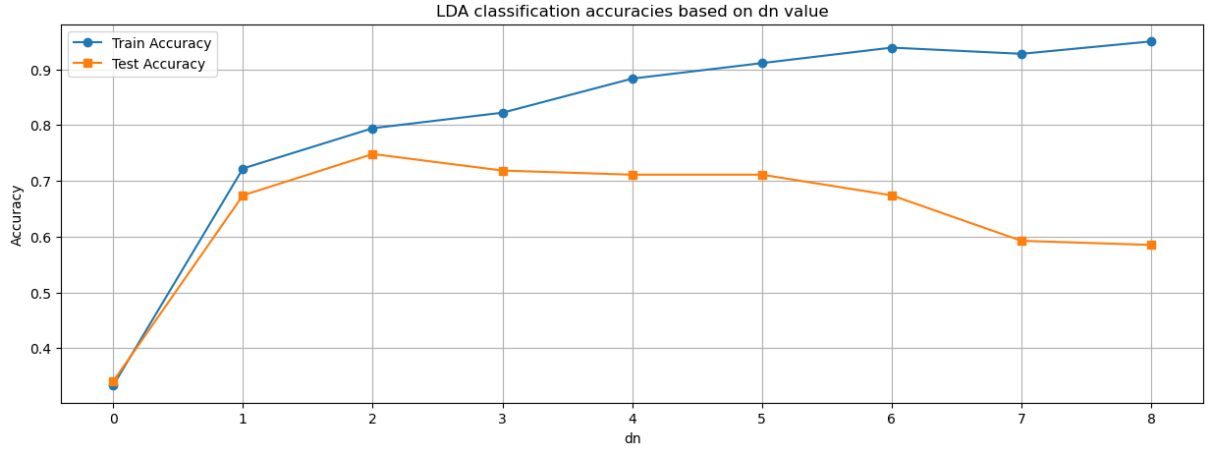


Figure 5: Training and test accuracies depending on dn

This plot shows the classification accuracies of the Linear Discriminant Analysis based on different values of dn . We observe that when dn increases, the training accuracy keeps improving and reaches about 0.95, but the test accuracy peaks around $dn = 2$ at 0.75 and then starts declining. This suggests that using too much spatial information makes the model memorize the training data too well, reducing its ability to generalize to new images. Therefore, the optimal dn value appears to be around 2, where test performance is highest.

Moreover, using the autocovariance matrix as a feature provides better classification performance compared to using just the mean values. In fact, the mean values alone resulted in much lower accuracies. This improvement suggests that the autocovariance matrix is a more effective feature for classification.

6 Linear Discriminant Analysis (LDA) with the power spectrum (periodogram)

6.1 Relationship between power spectrum and autocovariance

The power spectrum provides a way to analyze the frequency components of a signal, while the autocovariance describes how the signal correlates with itself over different time lags or spatial shifts.

Moreover, the power spectrum $S(\omega)$ is the Fourier transform of the autocovariance function $\phi(\tau)$. This relationship can be expressed as:

$$S(\omega) = \mathcal{F}\{\phi(\tau)\} = \int_{-\infty}^{\infty} \phi(\tau) e^{-i\omega\tau} d\tau$$

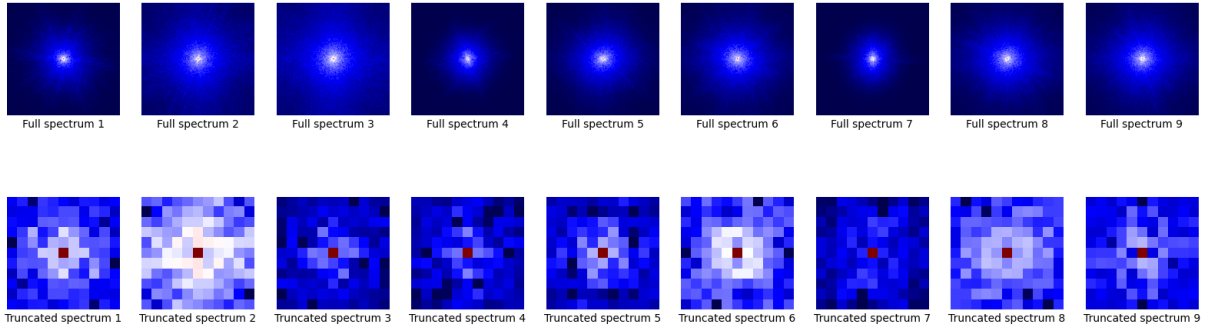
where:

- $S(\omega)$ is the power spectrum, representing the distribution of power over different frequency components.
- $\phi(\tau)$ is the autocovariance function, which measures the correlation of the signal with itself at different lags τ .
- ω represents the angular frequency.
- \mathcal{F} denotes the Fourier transform.

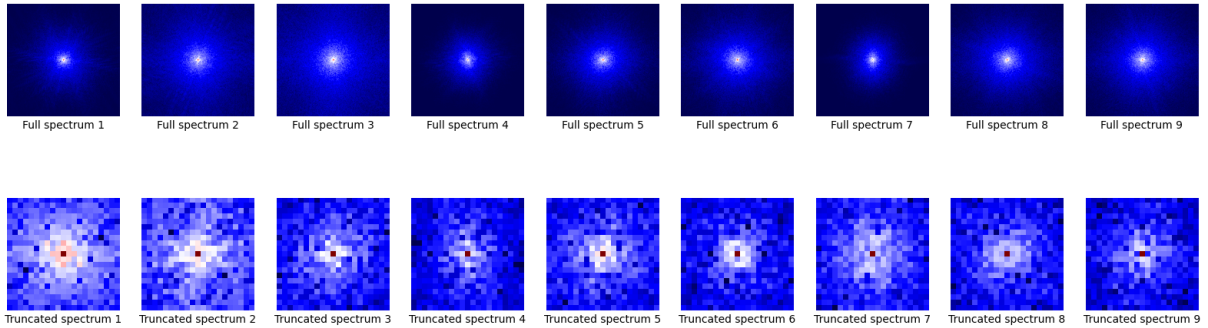
6.2 Analysis of full spectrum vs truncated spectrum for different *dom* values

We will now analyze different spectra for different *dom* values of each class.

Full spectrum vs truncated spectrum with dom=5



Full spectrum vs truncated spectrum with dom=10



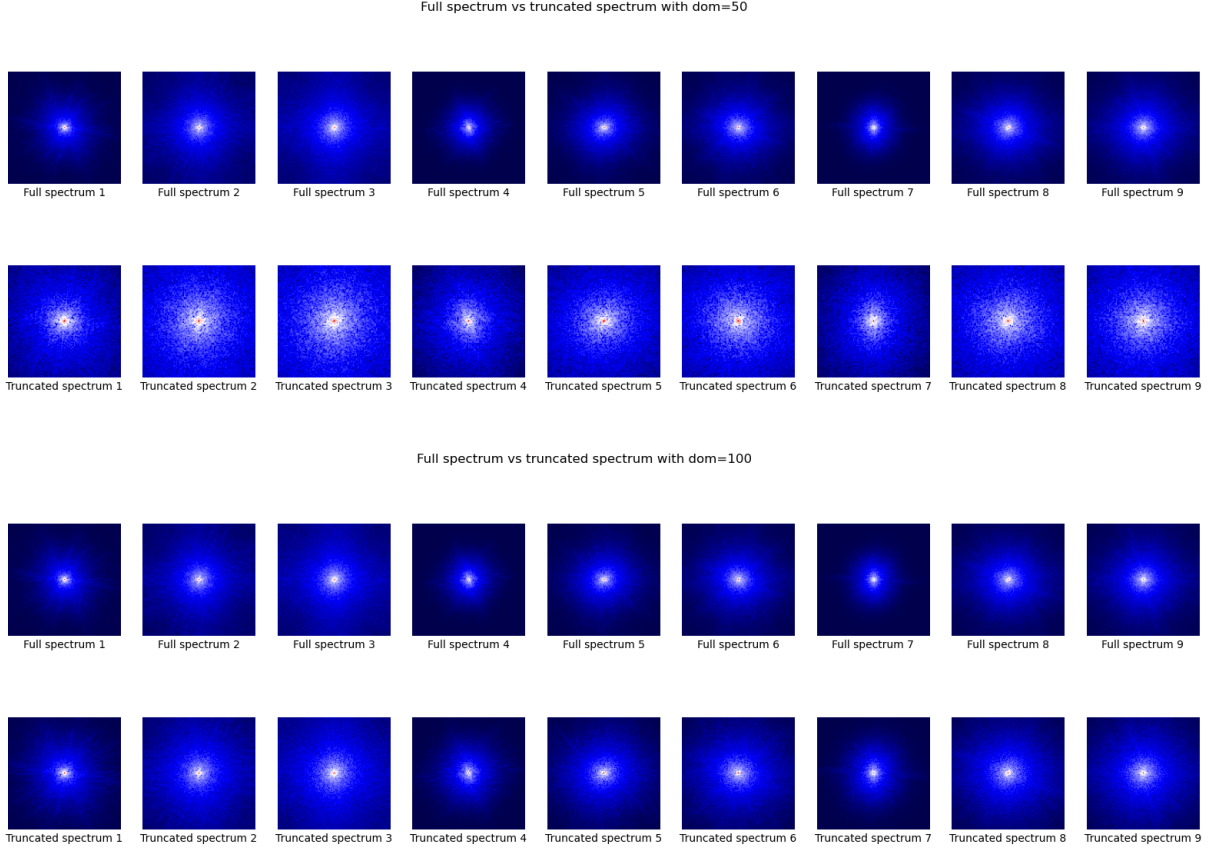


Figure 6: Full spectrum vs truncated spectrum for different dom values of each class

We observe that the truncated spectrum changes as we increase the value of dom . For $dom = 5$, the truncated spectrum is quite small and doesn't capture much detail from the full spectrum. However, when we increase dom to 10, 50, and 100, the truncated spectrum starts to include more details and looks more like the full spectrum.

We suggest studying a range of dom values from **20 to 100**. By studying this range, we can find the best dom value that gives us good classification results without requiring too much computational power.

6.3 Classification using LDA

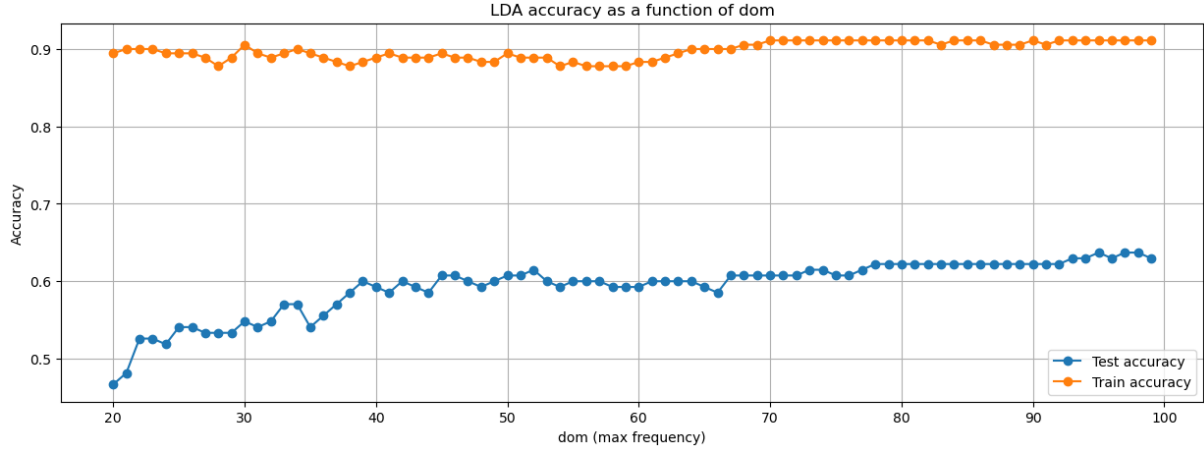


Figure 7: Train and test accuracies as a function of *dom*

We observe that the training accuracy stabilizes and remains high, indicating that the model is consistently performing well on the training data. However, the test accuracy shows a more gradual increase and eventually stabilizes at a lower level than the training accuracy. This discrepancy between training and test accuracies suggests that while the model fits the training data well, it may not generalize as effectively to unseen data. The optimal *dom* value appears to be around 80.

7 Performance analysis

7.1 Results with the empirical mean

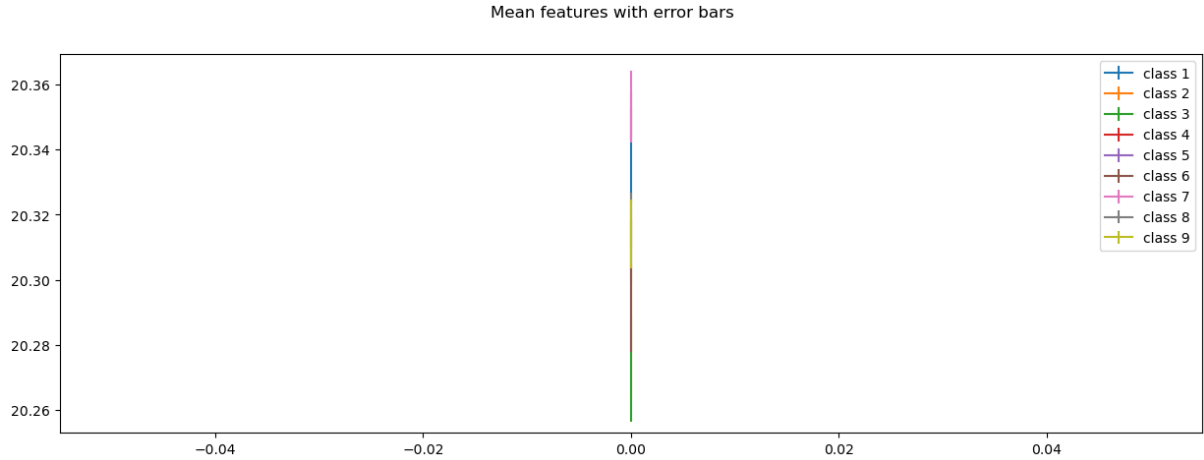


Figure 8: Mean features with error bars obtained with the empirical mean

This plot shows the mean feature vectors for each class with error bars representing the standard deviation. Each class is represented by a different color.

We can observe that the mean values for all classes are very close to each other, centered around approximately 20.30. We can also notice that the error bars are almost indistinguishable. This indicates very low variability within each class. Thus, the mean feature alone is not sufficient to distinguish between the different classes.

7.2 Results with the empirical autocovariance matrix

The figures presented below allow us to evaluate the relevance of the feature vectors extracted from the autocovariance matrix with the optimal displacement parameter $dn = 2$ for classification.

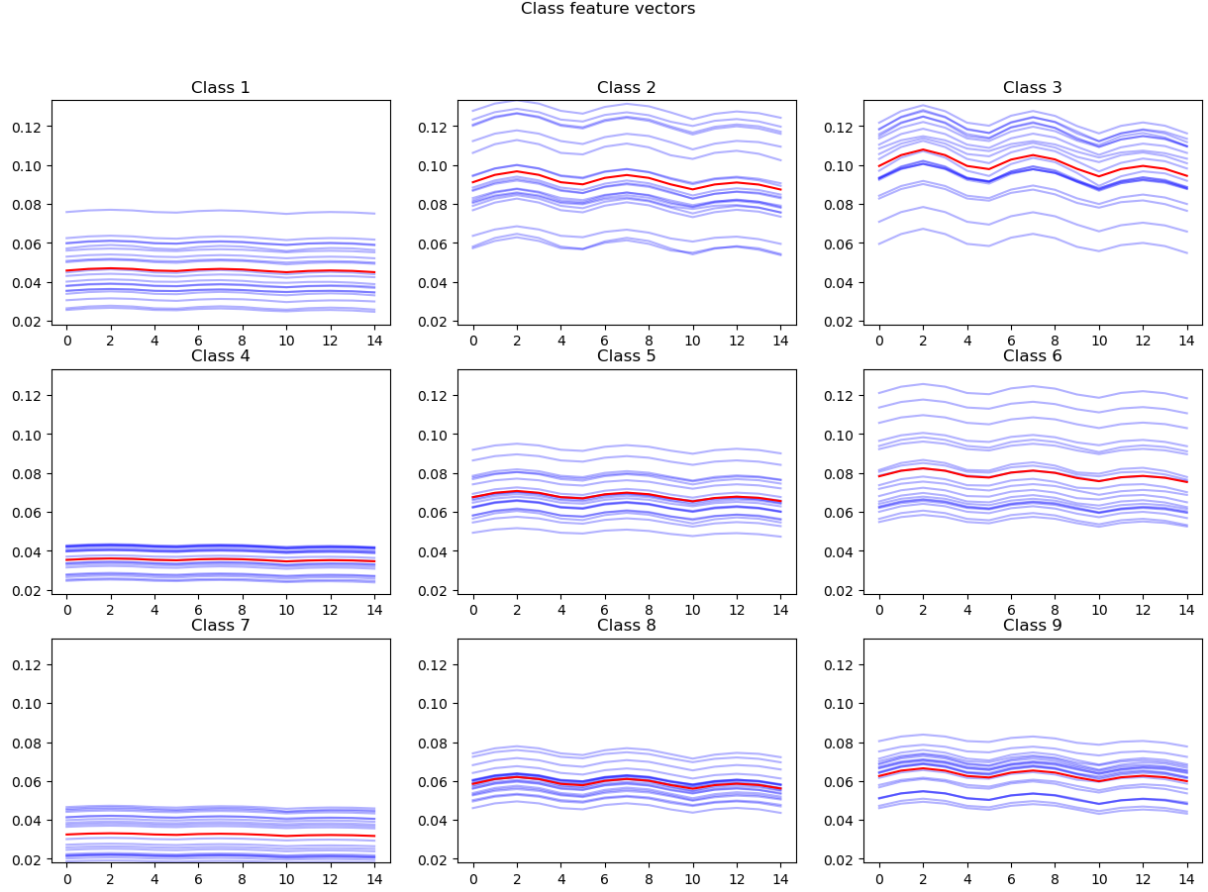


Figure 9: Class features vectors extracted from the autocovariance matrix

The figure above displays, for each class, all the projected feature vectors (in blue), as well as their mean (in red). It displays more variability among the individual samples within each class, as seen by the spread of blue lines around the red mean line. This variability suggests that the autocovariance captures more detailed spatial relationships within the images, providing better separation between classes.

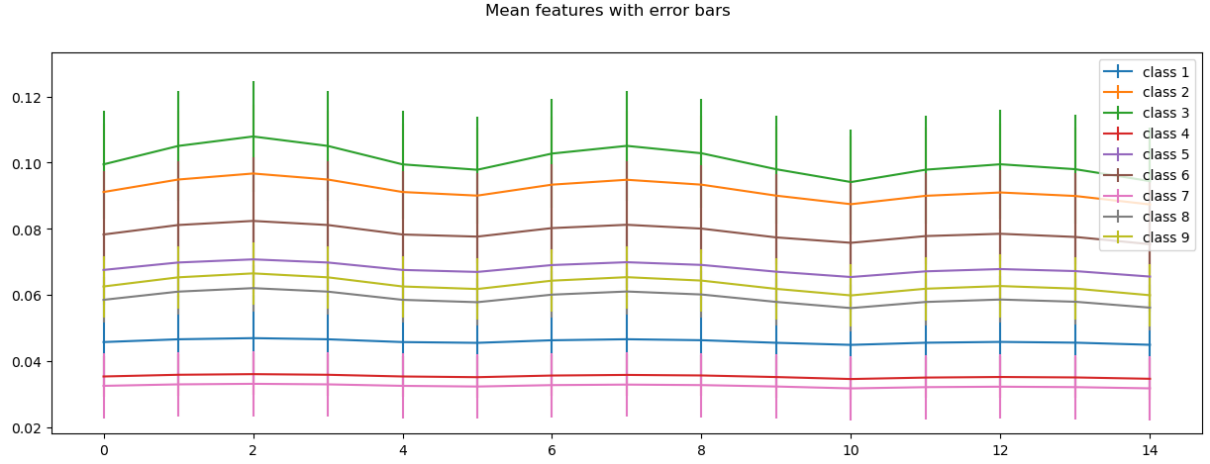


Figure 10: Mean features with error bars obtained with the autocovariance matrix

Here, we observe distinct patterns for each class, indicating that the autocovariance features are more effective in distinguishing between different classes. The error bars show some overlap but are more spread out compared to the mean-only features. In fact, some classes, such as class 7, exhibit very low variability, while others, such as class 2, show more significant dispersion. The mean curves are well separated on several coefficients, indicating that some of them are particularly discriminative.

These results show that the features derived from the autocovariance matrix capture relevant structural properties and provide a solid basis for classification.

7.3 Results with the power spectrum

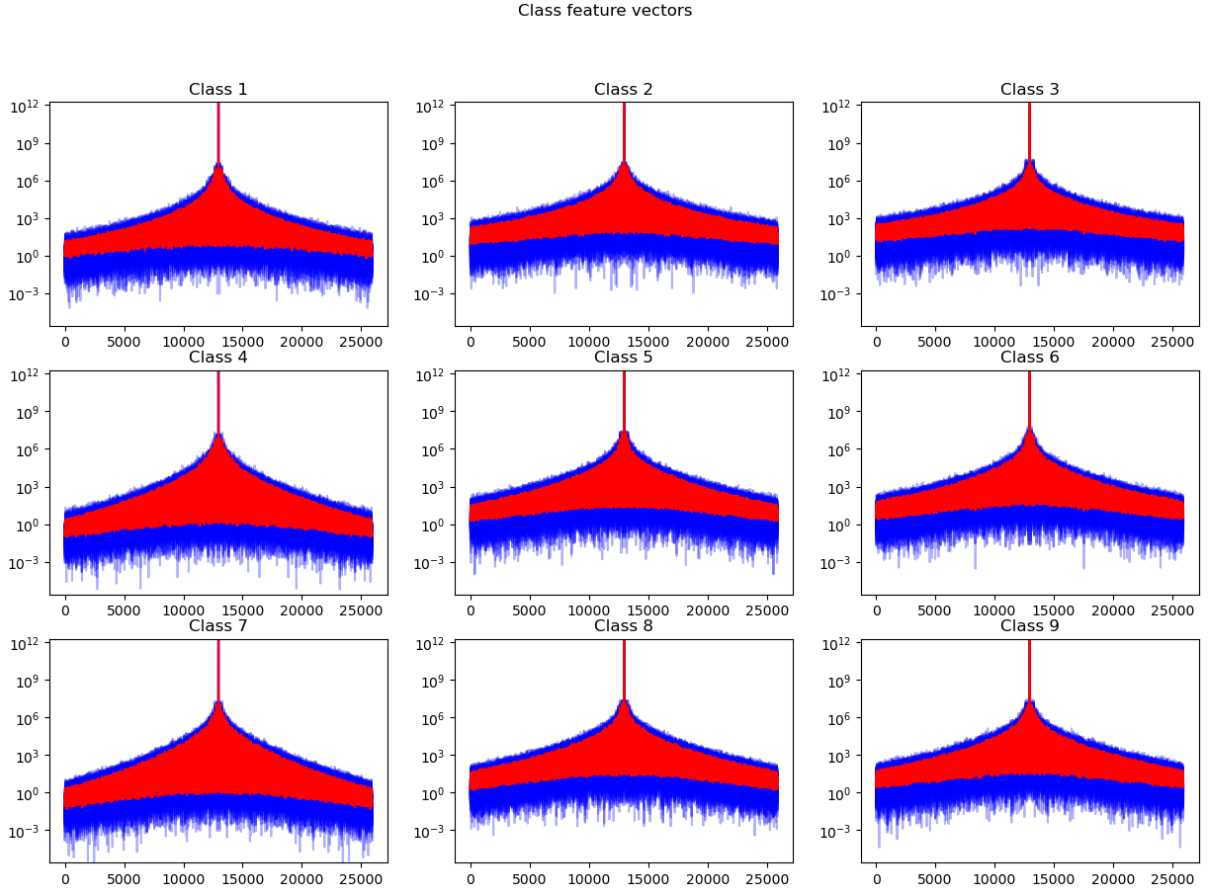


Figure 11: Class features vectors extracted from the power spectrum

The results using the power spectrum features with the optimal truncation parameter $dom = 80$ show distinct patterns for each class. For each plot, we observe a high peak at the center, which is characteristic of power spectrum representations. In fact, it reflects the concentration of power at lower frequencies. The spread of the blue lines around the red mean line in each class plot suggests variability among individual samples, but the overall shape remains consistent, indicating that the power spectrum features are effective distinguishing between different classes.

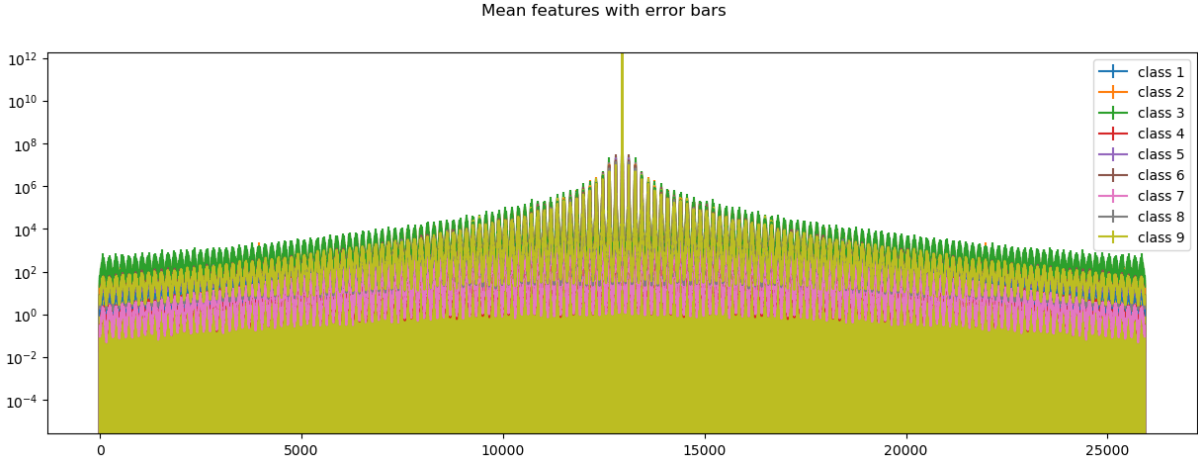


Figure 12: Mean features with error bars obtained with the power spectrum

In the figure above, the power spectrum features for each class show clear separation, especially in the central region where the power is most concentrated. The error bars are more pronounced and show distinct patterns for each class. Thus, using power spectrum features enhances the classification performance by capturing essential frequency domain characteristics of the images, leading to better differentiation between classes.

7.4 Exploring invariance in feature representation

In the construction of the autocovariance and power spectrum features, one group invariance that has not been explicitly exploited is **rotational invariance**.

Rotational invariance refers to the property of a feature that remains unchanged under rotations. In our context, this means that the features extracted from an image should ideally be the same regardless of how the image is rotated.

While the power spectrum is invariant to translations due to the use of the Fourier transform, and the autocovariance captures spatial relationships, neither of these methods inherently considers the rotational aspects of the image. One method that we studied in class to include rotational invariance is to use the concept of **rotational invariant scattering coefficients**. This approach leverages the properties of *waveletscatteringtransforms* to create features that are invariant to rotations.

8 Image classification using scattering features

In this part, we used scattering features derived from a wavelet scattering transform to classify the images. We also evaluated the performance using Linear Discriminant Analysis.

The function *compute_scattering_features* computes scattering coefficients for each image and averages them over orientations to achieve rotational invariance, which is crucial for tasks where the orientation of objects can vary. Then, we used the resulting features to train and test an LDA classifier.

The training accuracy achieved was approximately 0.89, indicating that the model fits the training data well. However, the test accuracy was lower at around 0.58, suggesting a potential overfitting to the training data or a need for more diverse training samples.

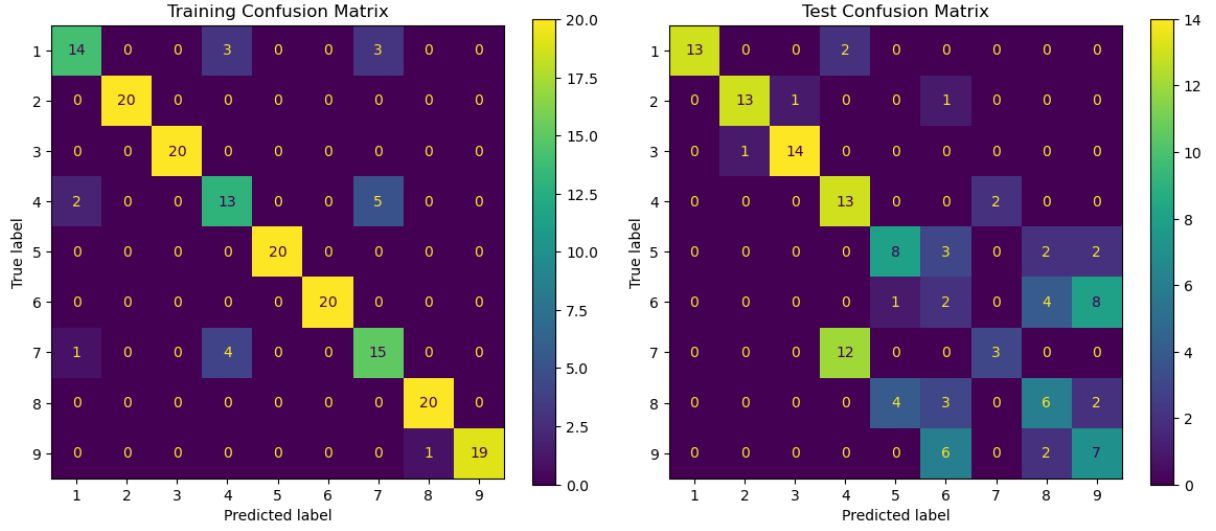


Figure 13: Training and test confusion matrices

The confusion matrices for both the training and test sets show that the model can distinguish some classes well, but it struggles with others, especially in the test set. This difference in performance between the training and test sets shows that the model has difficulty applying what it learned to unseen data.

8.1 Performance analysis

In the figure below, we observe that all classes share a similar general structure, but with distinguishable amplitudes in specific regions, particularly in the early coefficients. This indicates that the initial scattering coefficients contain the most discriminative information.

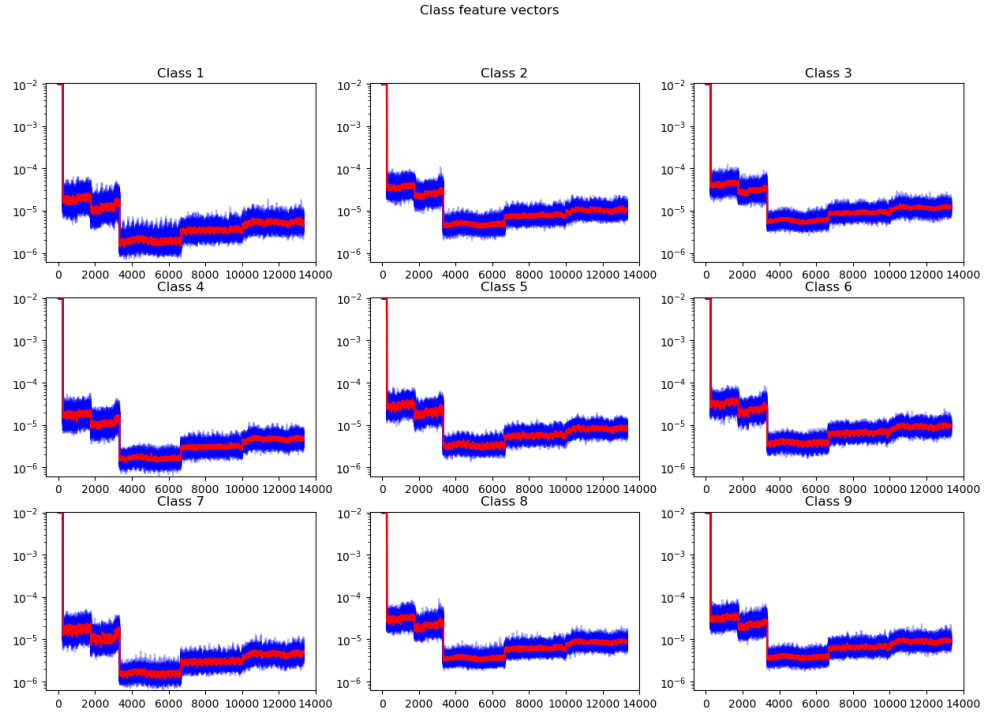


Figure 14: Class features vectors extracted from the scattering coefficients

Furthermore, we observe that the error bars remain relatively small for most classes, which shows that the intra-class variability is low and the extracted features are stable. In fact, certain classes, such as class 3 and class 9, display low dispersion, especially in the mid and high-frequency coefficients, which could contribute to some classification confusion. Overall, these results demonstrate that the wavelet scattering transform produces coherent features across classes, which are well-suited for LDA classification.

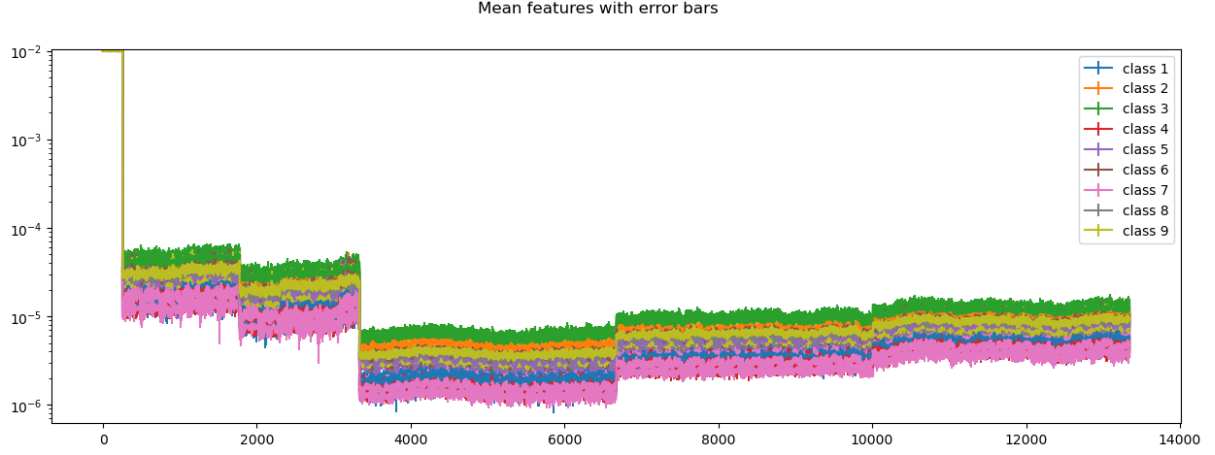


Figure 15: Mean features with error bars obtained with the scattering coefficients

9 Conclusion

To conclude, this study allowed us to explore various methods for extracting features from images to enhance classification performance. Among the methods tested, autocovariance features demonstrated the most promising results, achieving the highest accuracy on both training and test datasets. This indicates that autocovariance features are particularly effective in capturing the essential characteristics of interstellar medium simulations, thereby enhancing classification performance when using Linear Discriminant Analysis.