<u>Objective:</u> Multiple-instance, Closed-set Butterfly Classification using DiNo Features (d = 384)

<u>Approaches:</u> **I started with whole features.**

<u>Approach 1:</u> Since data is highly imbalanced, for my first approach I combined oversampling and under sampling from imbalanced learning [1] to generate synthetic data. For oversampling I tried SMOTE and for under sampling I tried Random, Tomek links and Edited Nearest Neighbors.

<u>Approach 2:</u> I tried all my basic models with same hyperparameter range as in Task 1. This time Logistic Regression with class_weight = 'balanced' performed best. SMOTE did not give better result compared to this approach. PCA did not help as feature size is 384. I have used minmax normalization for all my experiments. Results are summarized below:

No.	Approaches	Hyper parameter tuning	Best Validation Mean Class Accuracy
1	**Oversampling + Under sampling**	Solver = Liblinear cost_param = [1 10, 20, 30], penalty = 'l2'	0.7809 (SMOTE, Logistic Regression, C=20)
2	**Basic Models**	PCA n = [150, 200, 250, 500, 750] C = [0.1, 1, 10, 100]	**0.7813 (Best!)** (n = 250, Logistic Regression, C= 1)

I then shifted my focus to part features.

<u>Approach 1:</u> I averaged 9-part features with different combination of weights. Then I concatenated the average feature to whole feature to get d=768. Giving middle tiles higher weights gave me the best mean class accuracy on validation data.

<u>Approach 2:</u> I combined all features (d=3840). The rationale is to not lose any information and let linear model do the feature selection. This gave me best result among all my methods.

<u>Approach 3:</u> I trained 9 classifiers based on each part TILE, get prediction on validation data, and used them as part weights. Mean class accuracy was worse than just taking average of part features.

<u>Approach 4:</u> Test data has more images for sample id compared to validation images. To further improve my accuracy from my best model, I averaged features from same sample id. I replaced features with the average feature. The rationale was average feature will be more general compared to individual features and force all same sample ids to have same prediction. This approach decreased validation accuracy by 1.5%. But I still wanted to use this on test data as its distribution was different from validation.

<u>Approach 5:</u> I tried PCA and Logistic Regression with penalty = 'l1' on concatenated features to reduce dimensions. My validation accuracy was lower than just using all combined features.

Results are summarized below:

No.	Approaches	Best Validation Mean Class Accuracy
1	**Averaging parts features and concatenate to whole features**	0.8522 (Logistic Regression, C= 10) Giving middle tiles higher weights

2	**Combining all features**	**0.8571 (Best!)** (Logistic Regression, C=10)
3	**Sample id averaging**	0.8435 (Logistic Regression, C=10, d=3840)
4	**TILES approach**	0.8383 (Logistic Regression, C=10)
5	**Dim Reduction**	0.8433 (Logistic Regression, penalty = ‘l1’, C=15)

<u>Test results:</u>

No.	Model	Validation accuracy	Test accuracy
1	**Averaging parts features and concatenate to whole features (weight = 1/9) + Logistic Regression**	0.8522	0.5517
2	**Combining all features and sample id averaging + Logistic Regression**	0.8435	0.57