Alzheimer's Disease Onset Recognition by Handwriting:

A Deep and Machine Learning Approach

Cartaya A.
Gonzalez J.
Hartmann F.
Rivas M.
Machine and Deep Learning 2022-2023

Introduction

Alzheimer's disease is a neurodegenerative disorder characterized by cognitive impairments that progressively affect motor skills and cognitive abilities. Early diagnosis is crucial for slowing down brain damage and improving the quality of life for affected individuals.

This study employs machine learning and deep learning approaches to analyze data from Alzheimer's patients, focusing on pen gesture dynamics and task-related images.

Dataset and Feature Selection

Dataset

Original dataset includes 25 tasks distributed among three different types: tasks including memory stimulation, tasks that draw a graphic figure and tasks that requires copying text. Data consists of 166 samples (images and offline handwriting) divided into 88 patients and 78 healthy controls.

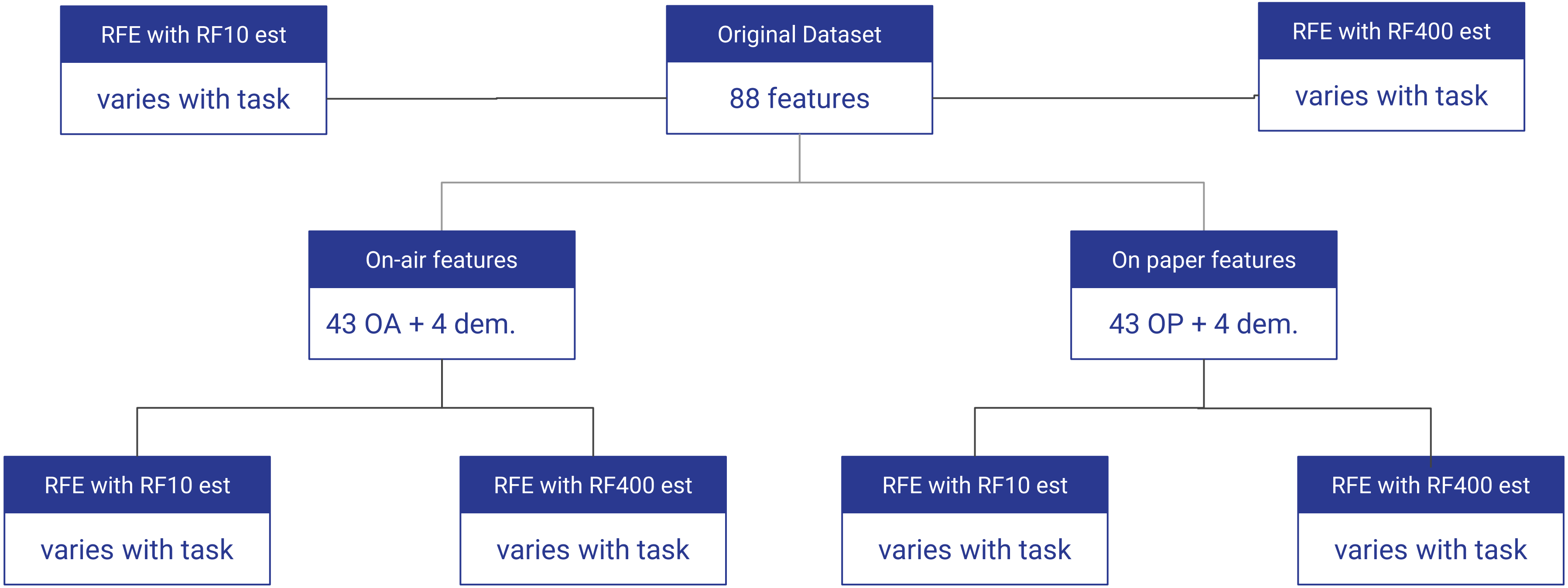
Task#	Task Description	Task Type
1	Signature drawing	MT
2	Join two points with a horizontal line continuously for four times	GT
3	Join two points with a vertical line continuously for four times	GT
4	Retrace a 6cm-diameter circle continuously for four times	GT
5	Retrace a 3cm-diameter circle continuously for four times	GT
6	Copy the letters 'l', 'm' and 'p'	CT
7	Copy the letters 'n', 'l', 'o' and 'g' in adjacent rows	CT
8	write the letter 'l' 4 times, continuously in cursive format	CT
9	write the bigram 'le' 4 times, continuously in cursive format	CT
10	Copy the word "foglio"[1]	CT
11	Copy the word "foglio" above a line	CT
12	Copy the word "mamma"	CT
13	Copy the word "mamma" above a line	CT
14	Memorize the words "telefono", "cane" and "negozio"[2] and rewrite them	MT
15	Copy in reverse the word "bottiglia"[3]	CT
16	Copy in reverse the word "casa"[4]	CT
17	Copy six words	CT
18	Write of an object shown in a picture	MT
19	Copy the fields of a postal order	CT
20	Write a simple sentence under dictation	MT
21	Retrace a complex form	GT
22	Copy a telephone number	CT
23	Write a telephone number under dictation	MT
24	Draw a needle clock pointing 11:05	GT
25	Copy a paragraph	CT

Table 1: Tasks description and type [5, 6]

Feature Selection

Recursive Feature elimination was used. This uses a backward stepwise approach. The “importance” of the features is given by using a Random Forest Classifier with 10 or 400 estimators.

Feature Selection



Machine Learning Approach

Things to Consider

A pipeline was considered in order to do data preprocessing, selection of best classifier and hyperparameter tuning. All handled by a reliable subset selection method: k-fold cross validation.

ML considerations

Preprocessing

Raw data, in most cases, is not enough to guarantee the success of a classifier. Instead a tidy version of each dataset improve the results. This includes encoding categorical features, scaling and handling outliers.

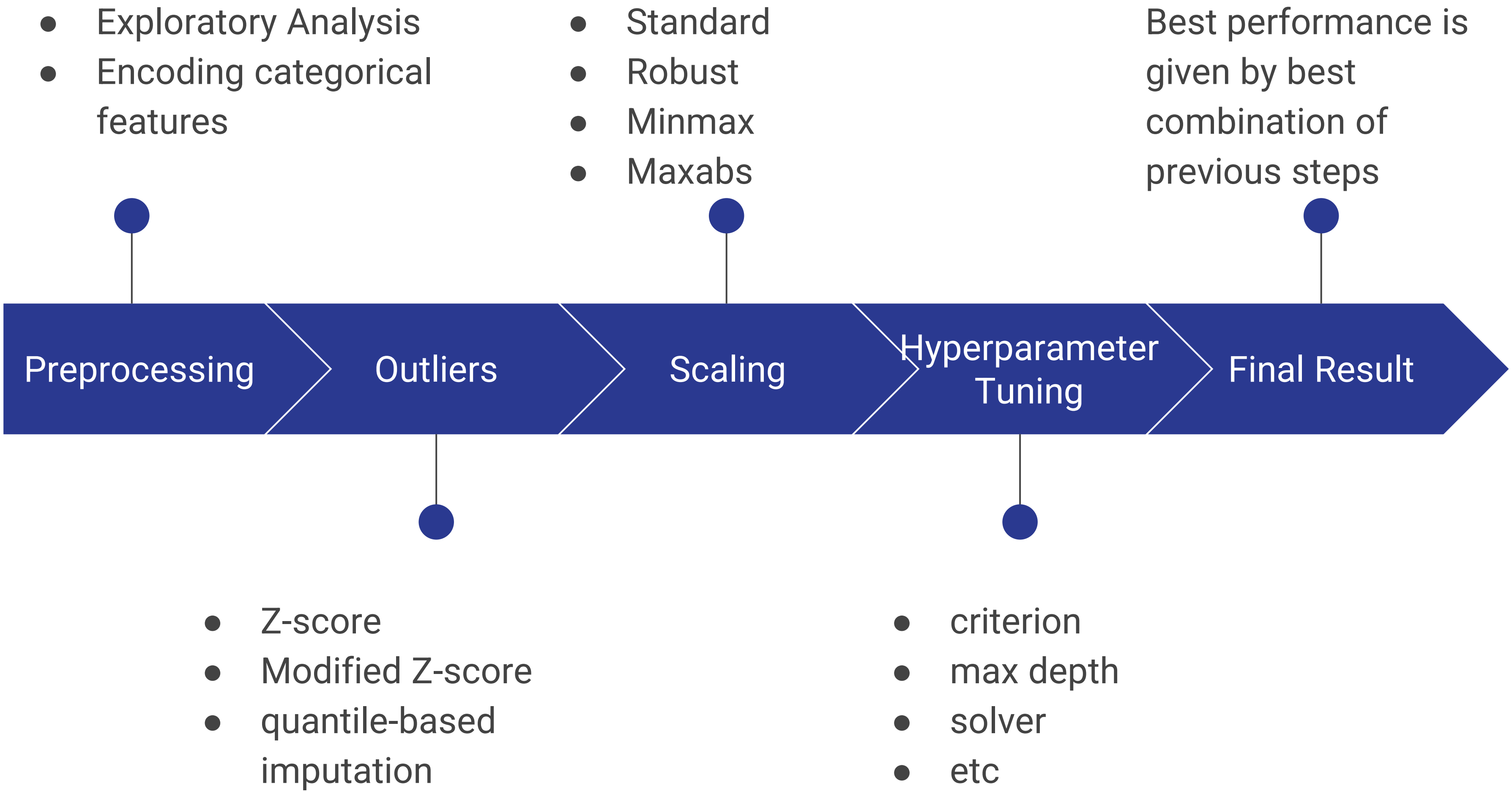
Classifier Selection

For the scope of this project, we selected:

- Decision Tree
- Gradient Boosting
- LDA
- Random Forest
- SVC
- Extreme Gradient boosting

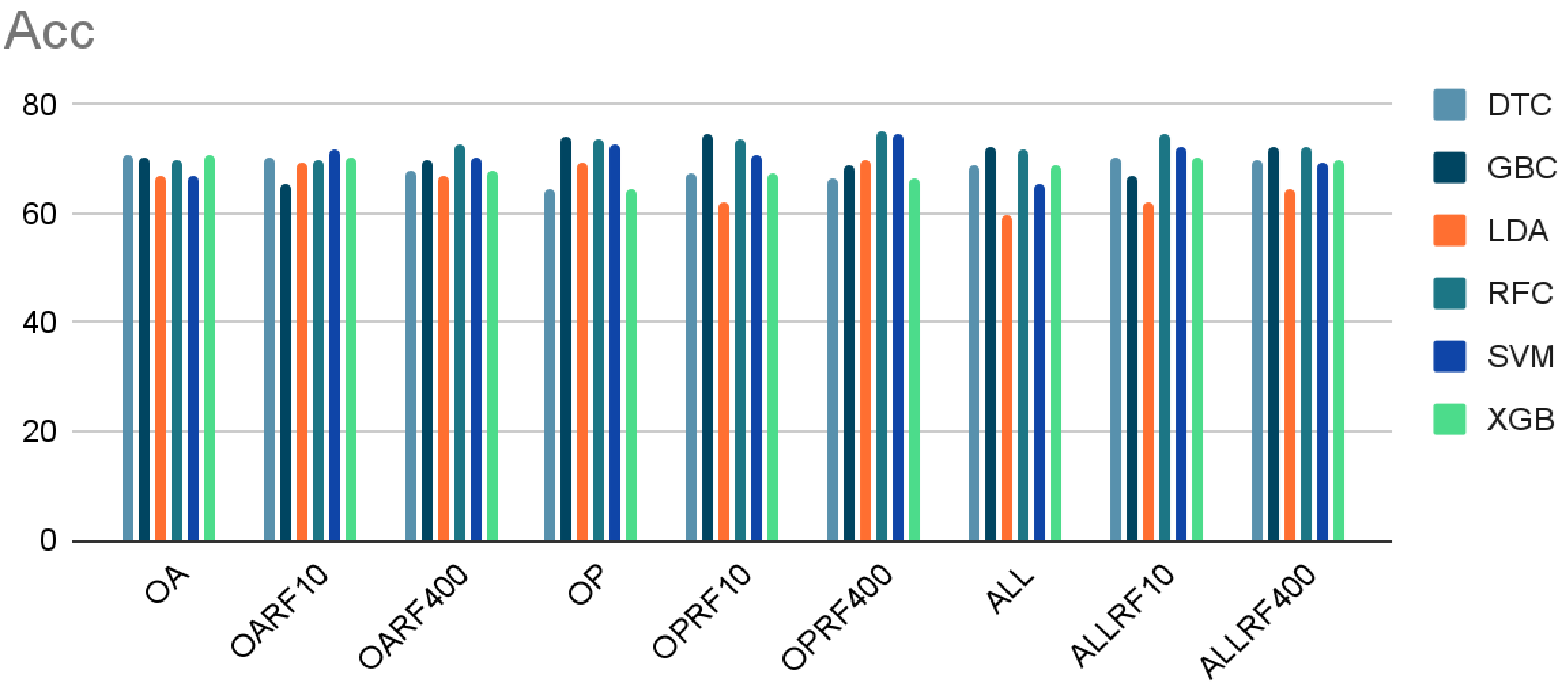
Hyperparameters

Use of Gridsearch to find the best parameters for each classifier. All within a frame of cross validation.

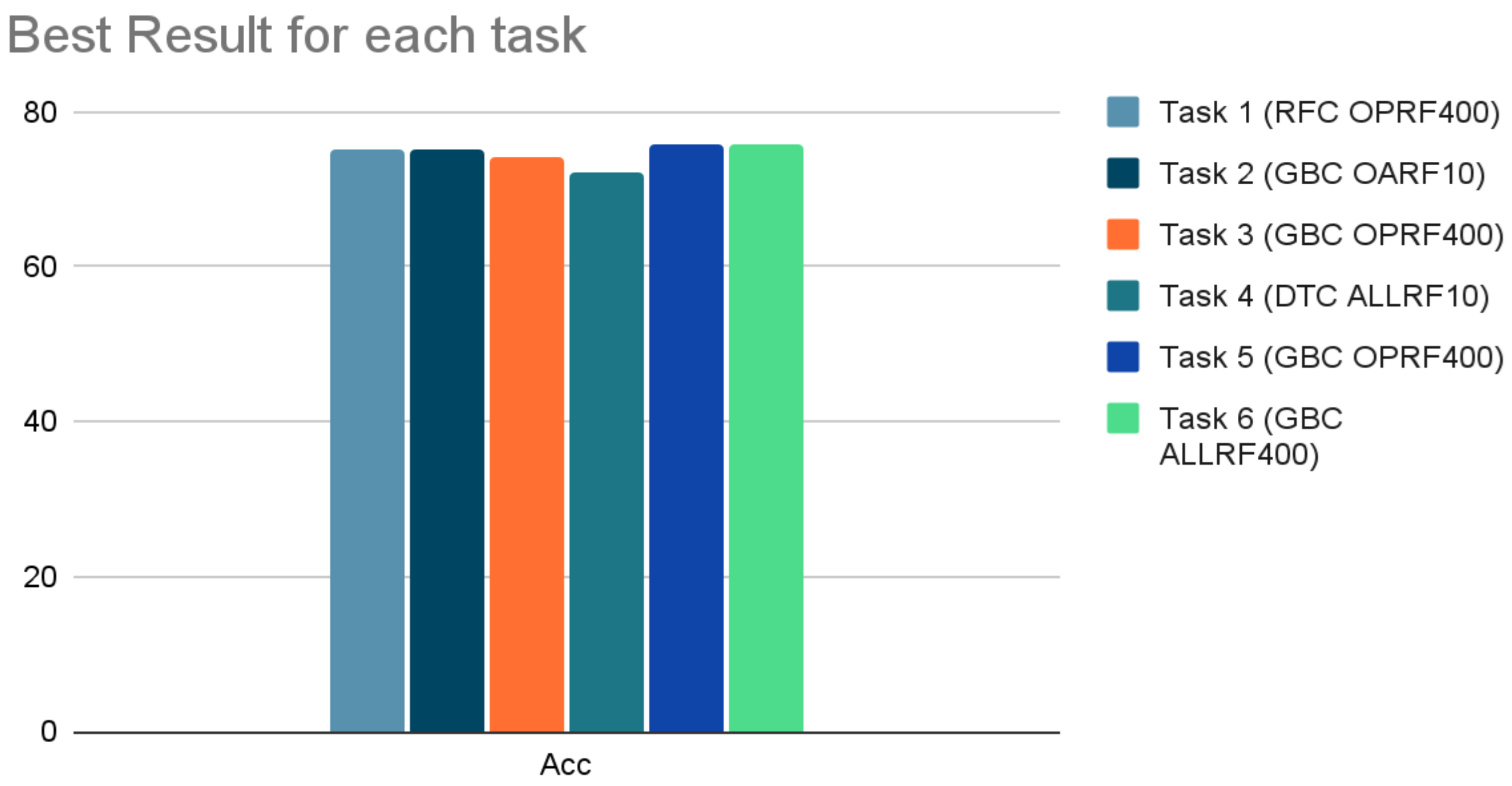


Exploratory Analysis
Encoding categorical features
Standard
Robust
Minmax
Maxabs
Best performance is given by best combination of previous steps
Preprocessing
Outliers
Scaling
Hyperparameter Tuning
Final Result
Z-score
Modified Z-score
quantile-based imputation
criterion
max depth
solver
etc

Pipeline Results - Task 1



Pipeline Results - Best Result for each task



Deep Learning Approach

Things to Consider

Training a neural network requires the design of a structure and a lot of computational power. Even if used an existing architecture. A transfer learning approach was used considering our computational resources and the small dataset.

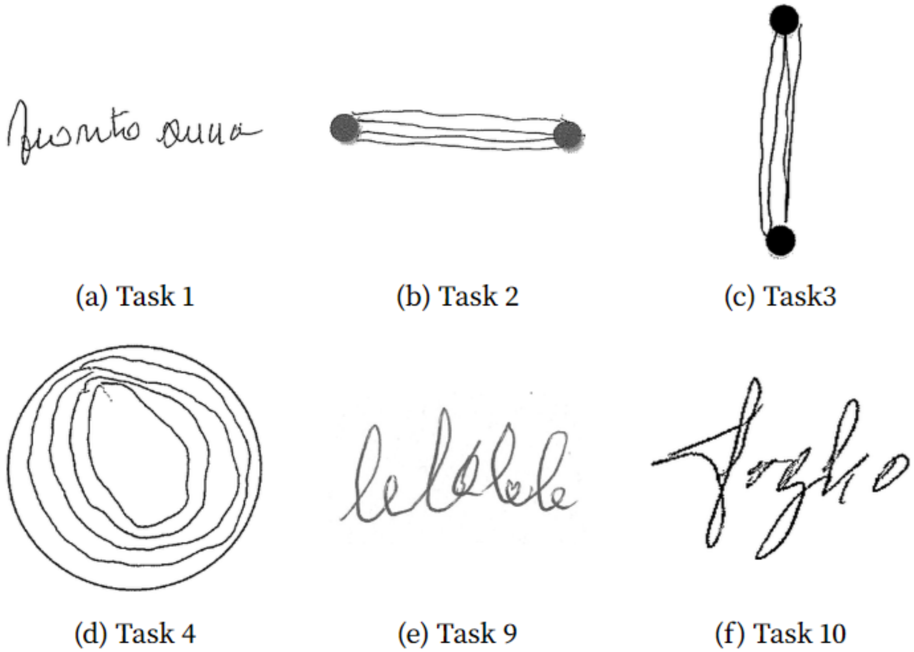


Figure 1: Sample of one image per task

DL considerations

Backbone base model

Among all possible architectures the following ones were selected:

- VGG19
- Inceptionv3
- Resnet50
- InceptionResnetv2

Optimizer

Stochastic Gradient Descent tend to perform better when transfer learning, but **Adam** converge faster-

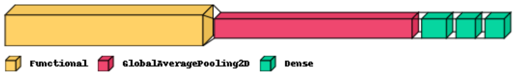
Constraints

Dataset images are 299x299 pixels and not all backbone base models have the same input. Consider fine tuning since our problem has few data and is different from the OG data: ImageNet.

DL considerations

Custom classifier

Different architectures would be tested, on top of the base model.

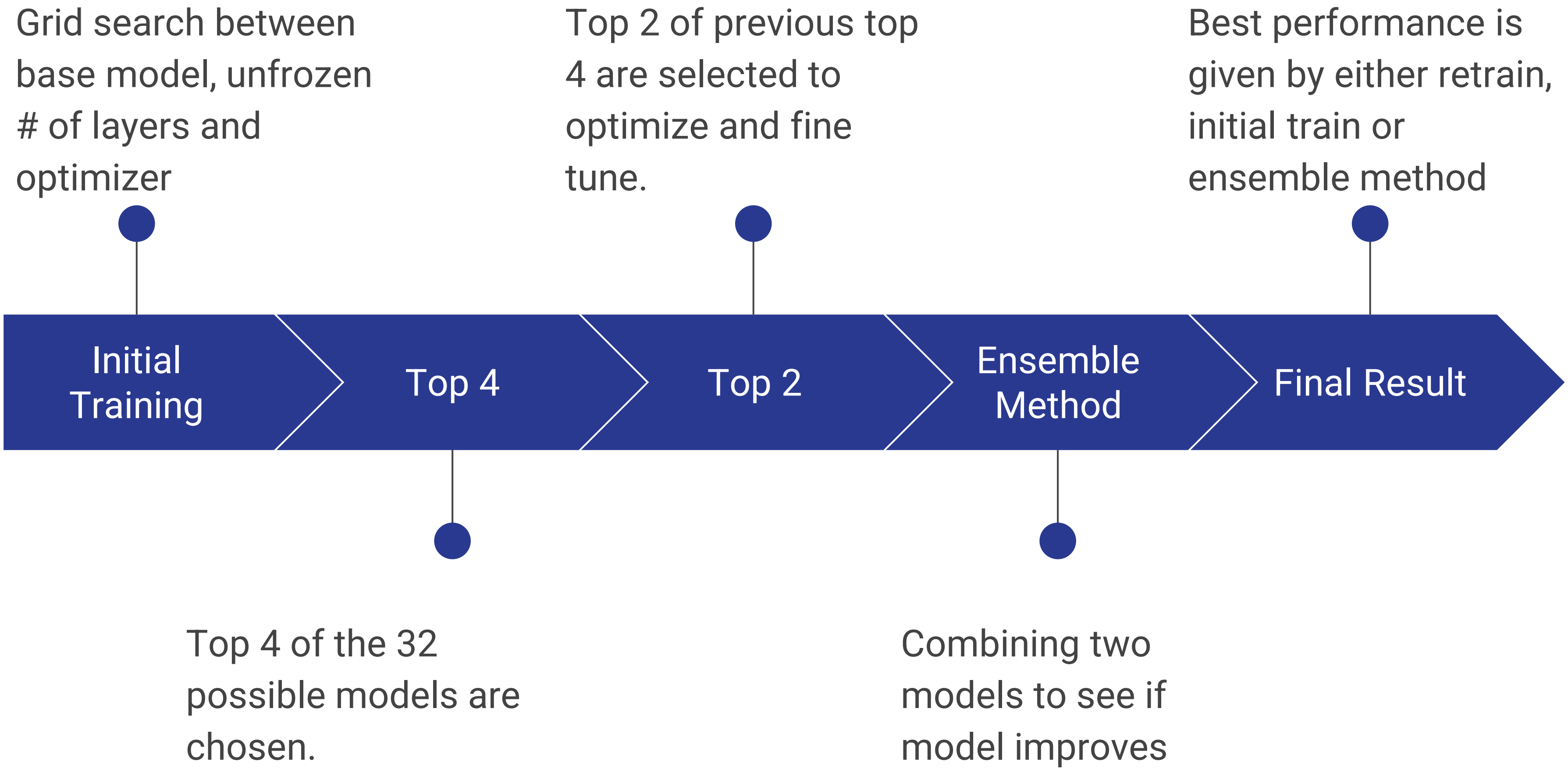


Frozen Layers

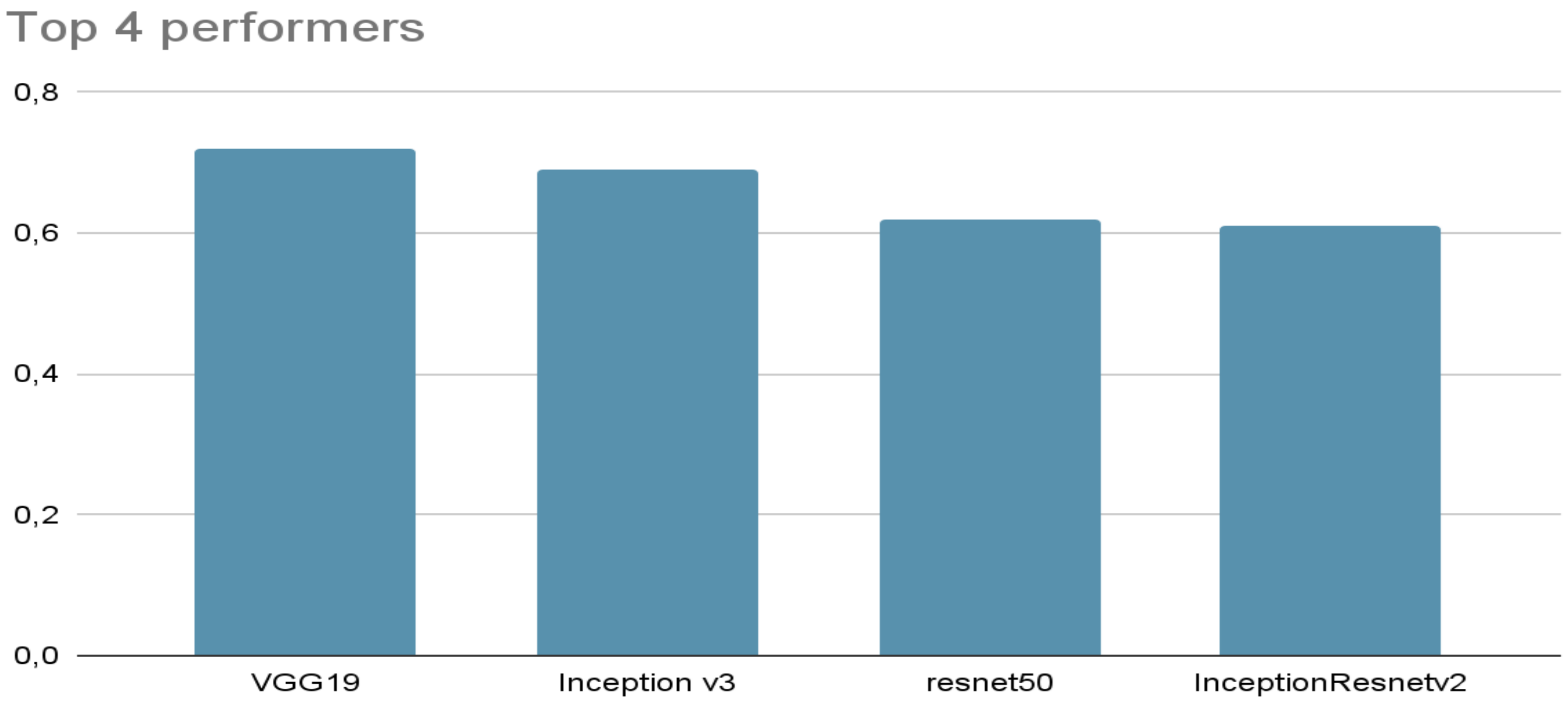
How many layers would be unfrozen to trade-off between learning from the new data and the already achieved generalization capacity.

Selecting the Best

The best Model would be retrained, changing parameters inherent to ANN like the epochs. After that, having an ensemble of models may improve performance.

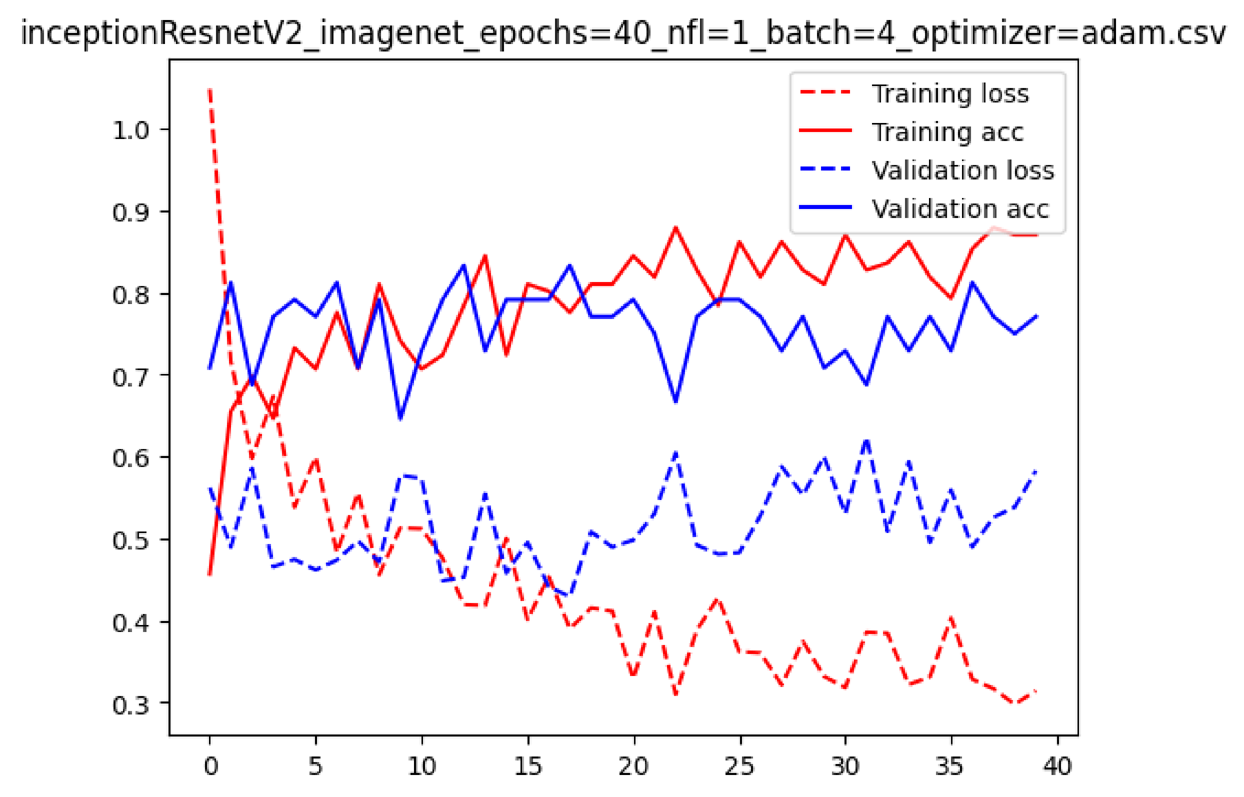

Grid search between base model, unfrozen # of layers and optimizer
Top 2 of previous top 4 are selected to optimize and fine tune.
Best performance is given by either retrain, initial train or ensemble method
Initial Training
Top 4
Top 2
Ensemble Method
Final Result
Top 4 of the 32 possible models are chosen.
Combining two models to see if model improves

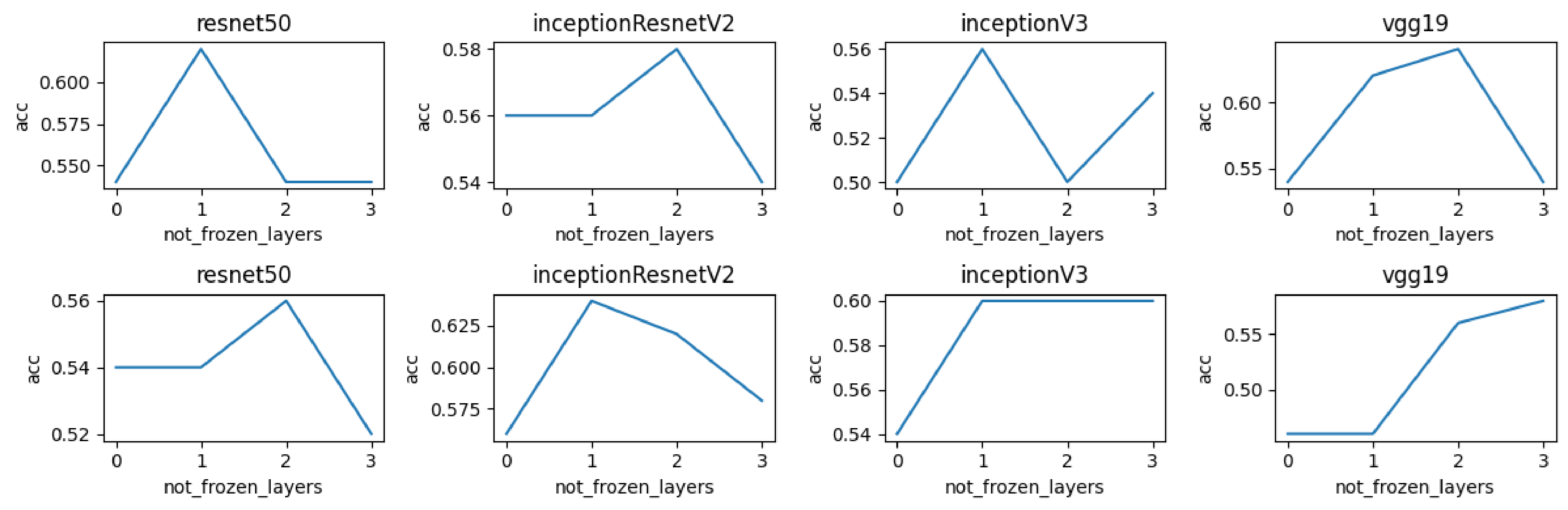
Pipeline Results - Best Result task 1



	Model	Optimizer	Unfrozen layers	Acc	Sen	Spe	Pre	F1	AUC	MCC	BA
29	VGG19	Adam	2	64.0	59.2	69.5	69.5	64.0	0.72	0.28	64.4
22	inceptionv3	SGD	3	60.0	77.7	39.1	60.0	67.7	0.69	0.18	58.4
3	resnet50	Adam	1	62.0	59.2	65.2	66.6	62.7	0.62	0.24	62.2
14	iR2	SGD	3	58.0	66.6	47.8	60.0	63.1	0.61	0.14	57.2
17	inceptionv3	Adam	0	68.0	92.5	39.1	64.1	75.7	0.84	0.38	65.8
10	iR2	SGD	1	72.0	85.1	56.5	69.7	76.6	0.84	0.43	70.8
1	resnet50	Adam	0	72.0	96.2	43.4	66.6	78.7	0.82	0.47	69.8
30	VGG19	SGD	3	68.0	62.9	73.9	73.9	68.0	0.81	0.36	68.4
29	VGG19	Adam	2	84.0	77.7	91.3	91.3	84.0	0.89	0.69	84.5
7	resnet50	Adam	3	68.0	44.4	95.6	92.3	60.0	0.84	0.45	70.0
14	ir2	SGD	3	74.0	77.7	69.5	75.0	76.3	0.82	0.47	73.6
16	inceptionv3	SGD	0	74.0	92.5	52.1	69.4	79.3	0.81	0.49	72.3

Table 14: CNN results for tasks 1, 2 and 3


inceptionResnetV2_imagenet_epochs=40_nfl=1_batch=4_optimizer=adam.csv
Training loss
Training acc
Validation loss
Validation acc
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
5
10
15
20
25
30
35
40


resnet50
inceptionResnetV2
inceptionV3
vgg19
acc
not_frozen_layers
0.600
0.575
0.550
0.58
0.56
0.54
0.52
0.50
0.60
0.55
0.625
0.600
0.575
0.50
0 1 2 3

Final results

- Task 1: Phase 3 Ensemble of VGG19 + inceptionv3 for an AUC of 0.73.
- Task 2: Phase 3 Ensemble of inceptionV3 + inceptionResnetV2 with an AUC of 0.85.
- Task 3: Phase 2 VGG19 with an AUC of 0.90.
- Task 4: Phase 1 resnet50 with an AUC of 0.90.
- Task 9: Phase 1.5 inceptionResnetV2 with an AUC of 0.89.
- Task 10: Phase 1 inceptionResnetV2.

Deep Learning Features Approach

Things to Consider

Transfer learning not only serves as a learning model for new data, but also as a feature extraction tool. Using the convolutional layers, we extracted features from the images of the task and then used the same ML approach explained before.

DL features considerations

Number of features

100 features were extracted, taking advantage of feature extraction layers of each of the base models used before.

Analysis Approach

In this phase of the project, the same ML approach was used before in order to curate data, and select the best classifier with its own best parameters.

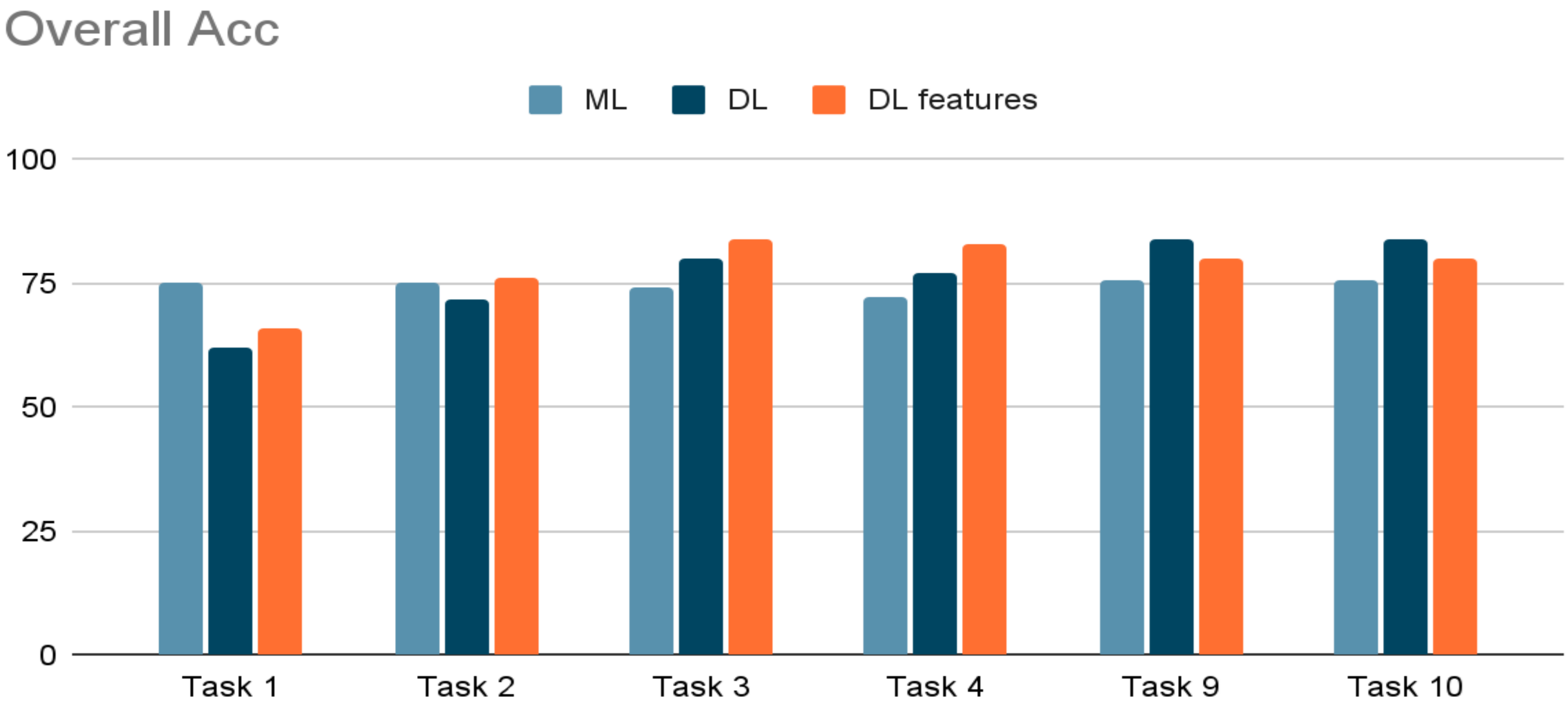
Classifiers

As we are working with ANNs, we used a Multi Layer Perceptron classifier, tuning hyperparameters like hidden layers, optimizer, solver and activation function.

Task			1		
Classifier	**RFC**	**DTC**	**SVM**	**XGB**	**MLP**
Accuracy	0.58	**0.66**	0.62	0.62	0.54
F1-Score	0.618	0.666	0.612	0.641	0.701
Precision	0.607	0.708	0.681	0.653	0.54
Recall	0.629	0.629	0.555	0.629	1.0
Task			**2**		
Classifier	**RFC**	**DTC**	**SVM**	**XGB**	**MLP**
Accuracy	0.72	0.64	**0.76**	0.7	0.46
F1-Score	0.758	0.639	0.777	0.716	0.0
Precision	0.709	0.695	0.777	0.730	0.0
Recall	0.814	0.592	0.777	0.703	0.0
Task			**3**		
Classifier	**RFT**	**DTC**	**SVM**	**XGB**	**MLP**
Accuracy	0.8	0.8	**0.84**	0.82	0.76
F1-Score	0.814	0.799	0.84	0.823	0.799
Precision	0.814	0.869	0.913	0.875	0.727
Recall	0.814	0.740	0.777	0.777	0.888

Table 16: Performance Achieved using Image Features. Tasks 1, 2, 3

Overall Results



Key takeaways: ML

- 4 out of 6 tasks performed better using RF400 feature elimination method.
- 4 out of 6 tasks performed better with GBC classifier.
- Just one task performed better with just on-air features. Meaning on-paper features are really descriptive when discriminating between classes.
- Is the best approach when dealing with such irregular task as #1

Key takeaways: DL

- Tasks 9 and 10 top performance.
- Except from task 1, copy tasks appear to have better performance using this approach.
- There are more “image” features than in graphic tasks.
- Data augmentation was not done. After tests, general performance improved without it.

Key takeaways: DL features

- Task 3 and 4 top performance.
- Task 2. If a simpler model were required, then ML is better.
- Graphic tasks benefit more of the features extracted by a DL approach.
- WRT tasks 9 and 10, there may be almost the same dynamics but less shapes.