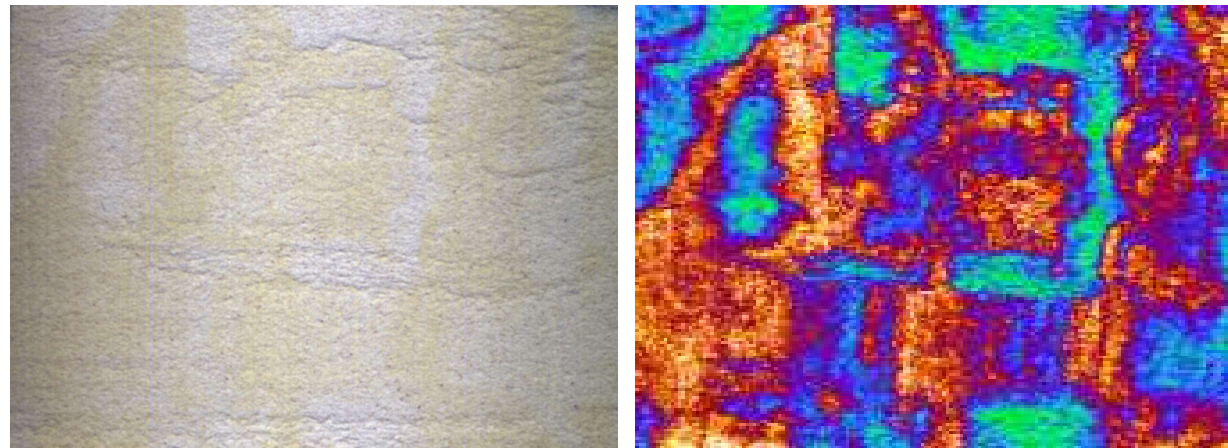


TRANSFER LEARNING OF RGB EMBEDDING MODELS TO HYPERSPECTRAL IMAGE ANALYSIS



A. Deryck^{1,2}, J.A. Fernández Pierna¹, J.R. Quevedo Pérez³, E. Montanés Rocés³, T. Goedemé²

¹ Walloon Agricultural Research Centre, Knowledge and valorization of agricultural products Department, a.deryck@cra.wallonie.be - j.fernandez@cra.wallonie.be

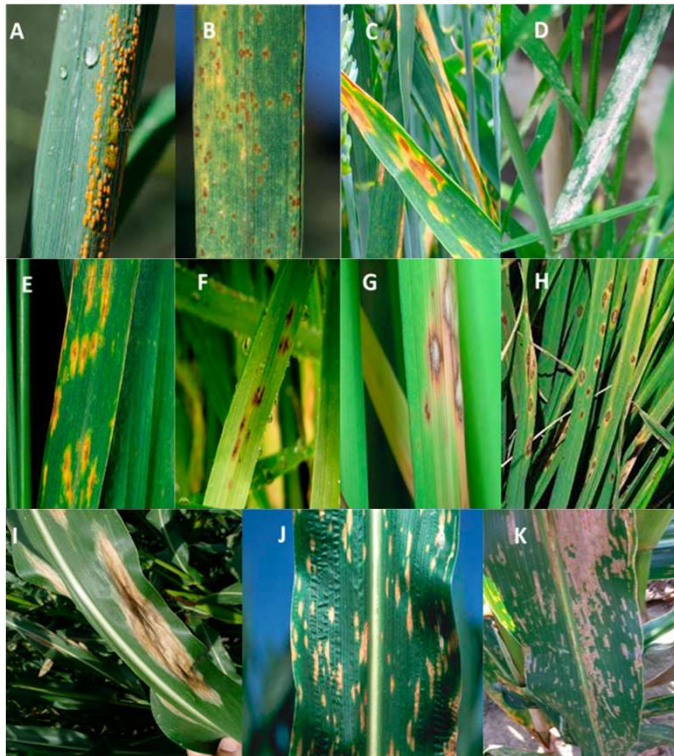
² KU Leuven, Embedded and Artificially intelligent VISION Engineering (EAVISE) research group, toon.goedeme@kuleuven.be

³ University of Oviedo, Artificial Intelligence Center, quevedo@uniovi.es - montaneselena@uniovi.es

Context – Hyperspectral Imaging

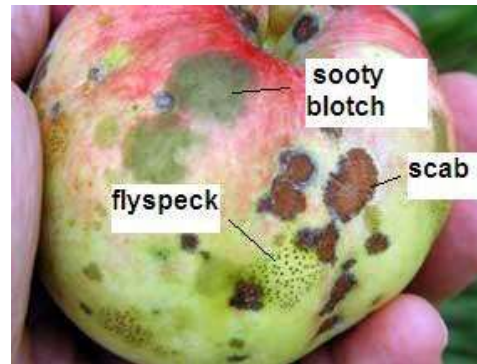
Hyperspectral Imaging = Spectral and spatial information

Plant diseases



Credits: 10.3390/plants12244162

Fruit defects



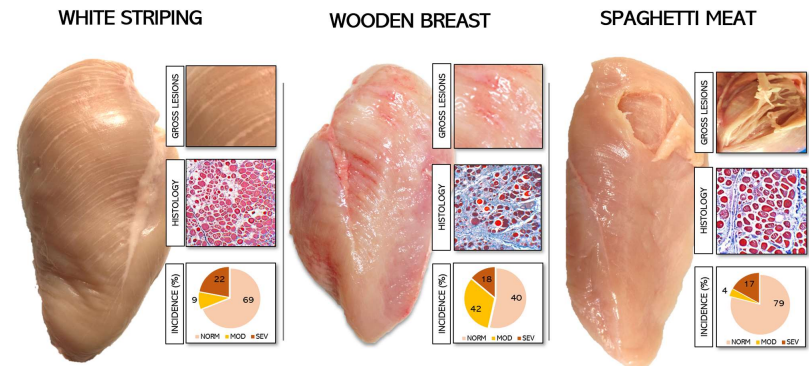
Credits: Michigan State University

...

Impurities detection

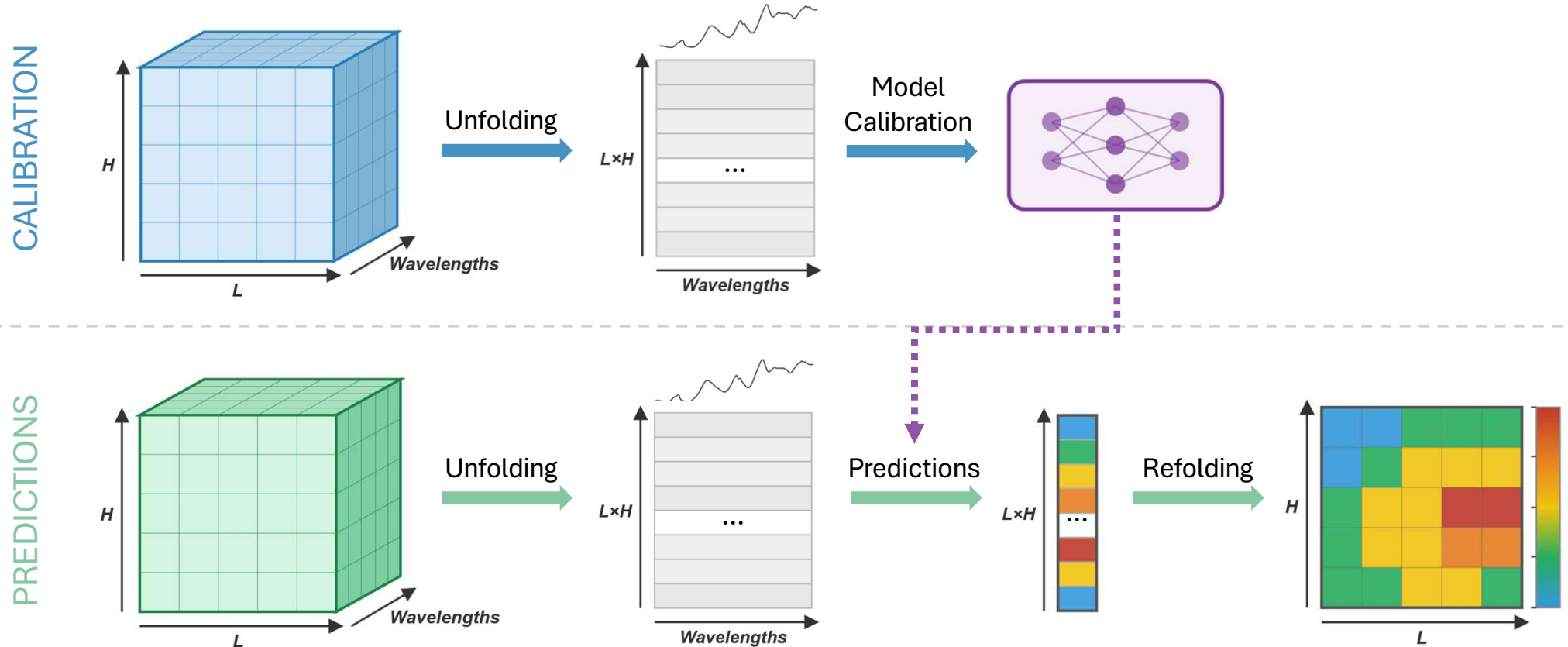


Texture analyses



Credits: University of Bologna (doi:10.22175/mmb.9503)

Context – Unexploited spatial information

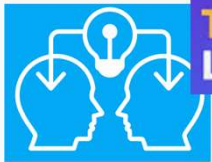


Context – RGB Images and Computer Vision

In **RGB imaging**, computer vision models (made of neural layers) are trained to extract both spectral (RGB) and spatial features that can be used for different prediction tasks.

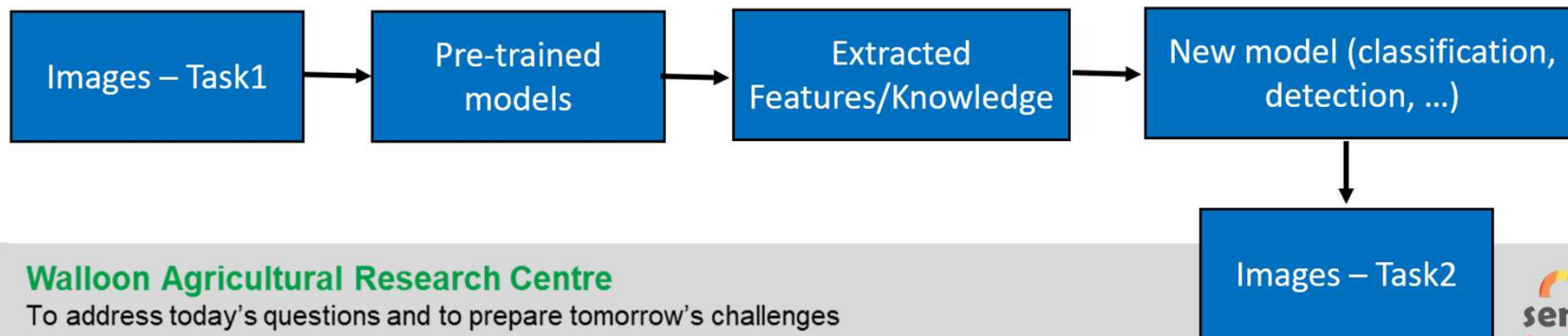
HOW IT WORKS

- Early layers capture basic patterns (edges, colors, textures)
- Deeper layers combine them into shapes, objects, ...
- Models such as YOLO, ResNet and CLIP learn these patterns from millions of images



TRANSFER LEARNING

- Parts/layers of these pretrained models can be used to extract useful features from new RGB images
- The extracted features are used to train a new model for a specific task



Context – From RGB to HSI

Transfer learning works well for RGB images



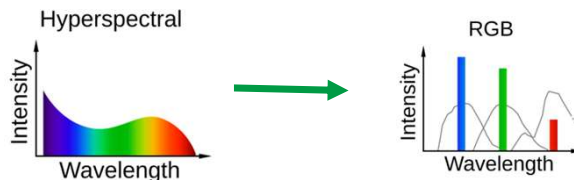
Can pretrained computer vision models extract meaningful spatial-spectral features from HSI?



Models only accept 3 channels (RGB)



Solution: Need dimensionality reduction techniques to go from X wavelengths to 3 variables



Models trained on everyday images may not fit hyperspectral data (more abstract and domain-specific)



Solution: Tests on agronomical HSI datasets, starting with semolina homogeneity

Dataset

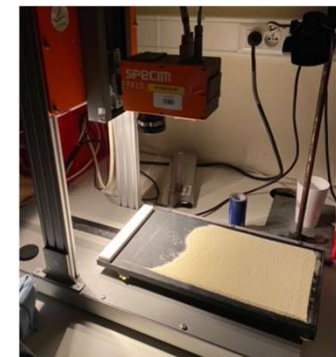
- Mixtures of bread wheat and durum wheat semolinas
- 11 proportions x 5 blending states x 5 biological replicates (different varieties)
- Measured with Specim FX10 (400-1000 nm)
- Hypercubes of dimensions 400x200x224
- Goal = predicting the blending state of samples

Mixture	Bread wheat (%)	Durum wheat (%)	Number of samples
MIX 1	100	0	25
MIX 2	90	10	25
MIX 3	80	20	25
MIX 4	70	30	25
MIX 5	60	40	24
MIX 6	50	50	25
MIX 7	40	60	25
MIX 8	30	70	25
MIX 9	20	80	25
MIX 10	10	90	25
MIX 11	0	100	25

Blending state	Blending time (s)	Number of samples
1	0	55
2	15	55
3	30	54
4	60	55
5	120	55

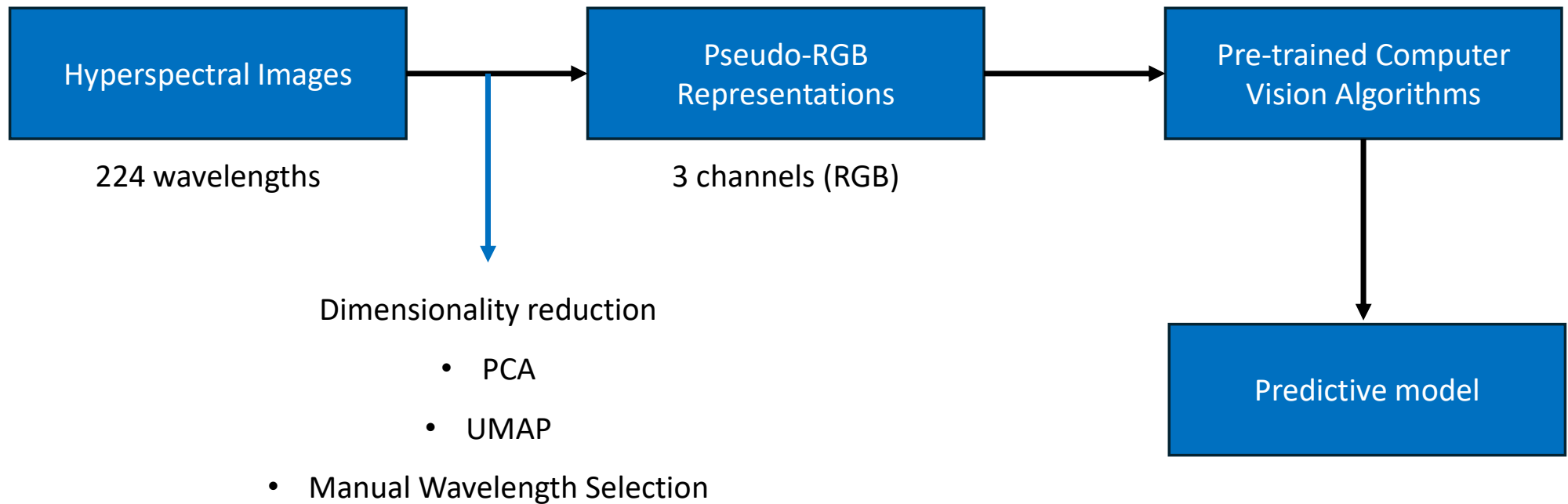


Blender: Heidolph Reax 20 mixer 3 rpm



Camera: specim FX10 (Specim, FI)

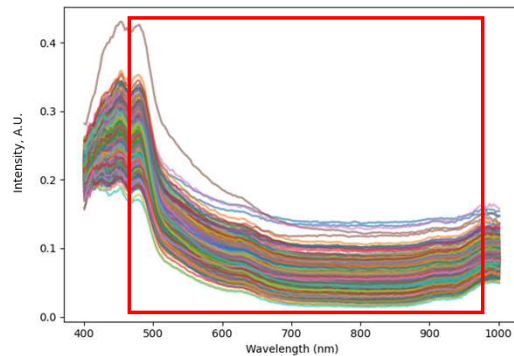
Methodology



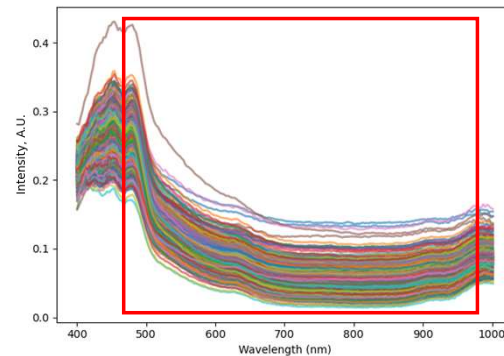
Data Preparation

- 1) Reflectance to Absorbance
- 2) Background Removal (Frame selection Algorithm)
- 3) Preprocessing (6 pipelines) + trimming (475 nm – 962 nm) :

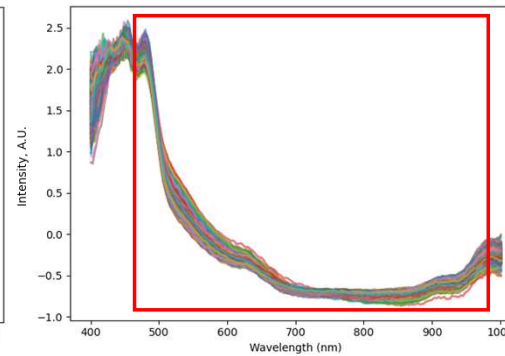
Raw data



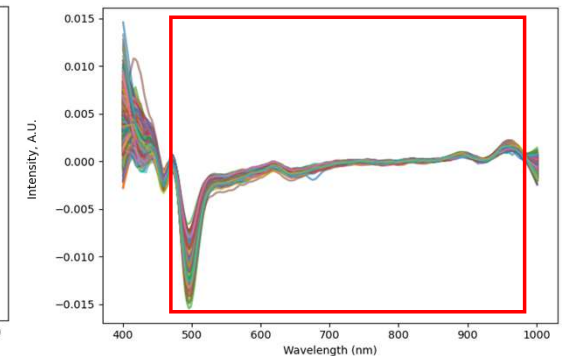
No spikes



No spikes + SNV



No spikes + SG



- 4) Image Resizing (152x170x181)

- 5) Data split:

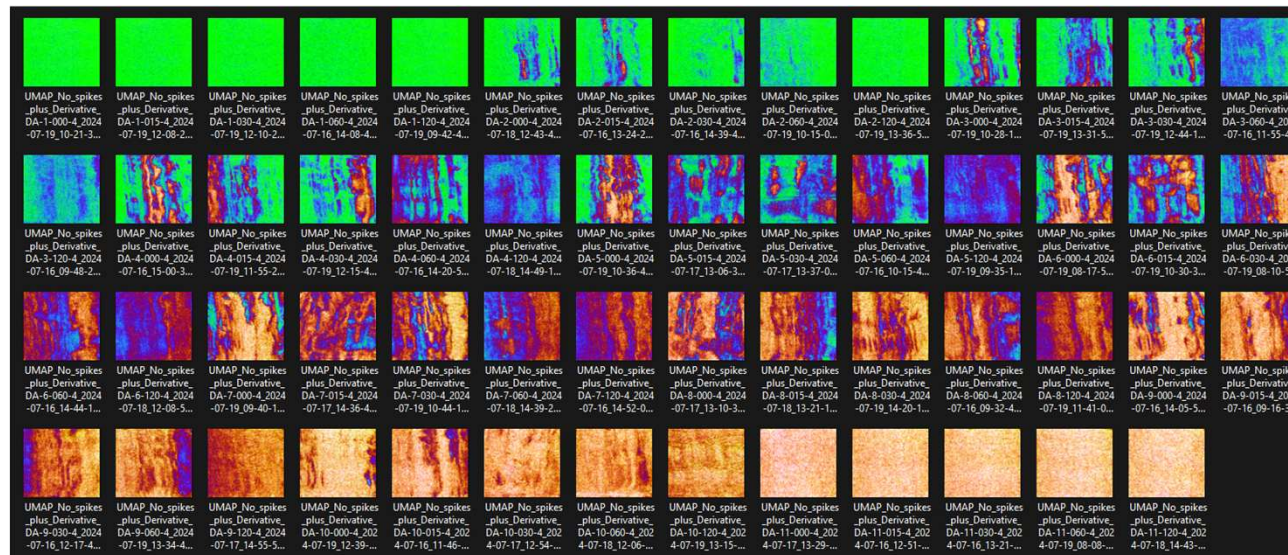
- Training set: Biological replicate 1-2-3-5 (80%)
- Test set: Biological replicate 4 (20%)

Dimensionality Reduction

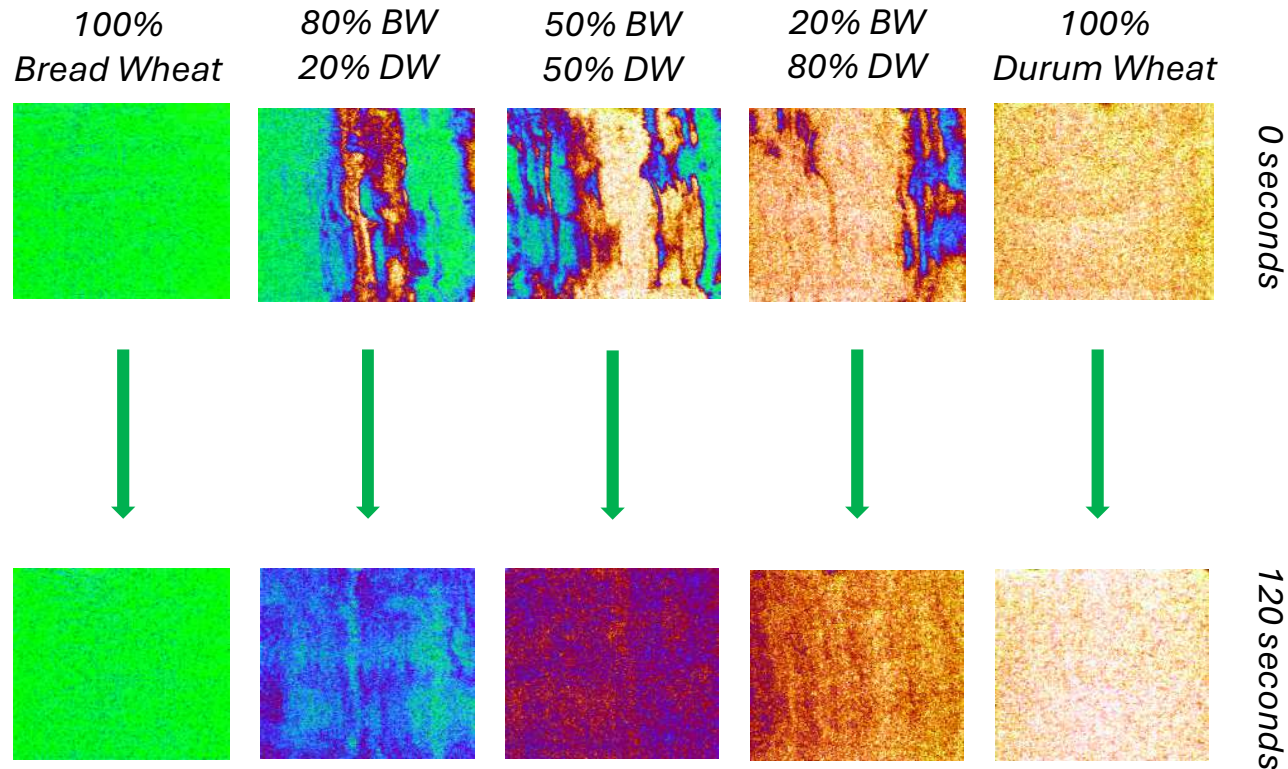
- PCA and UMAP calibrated on 10% of images from training + Manual Wavelength Selection
- Normalization by percentiles to obtain values between 0 and 1

$$x' = \frac{x - P_5}{P_{95} - P_5}$$

- Obtention of pseudo-RGB images



Pseudo-RGB Images



Opens the way to the use of pre-trained computer vision algorithms:

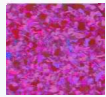
- Classification
- Segmentation
- Object detection
- Embeddings

...

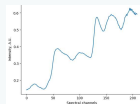
Embeddings

HOW IT WORKS

Object



« UMAP representation
of oregano spices
adulterated with cistus »

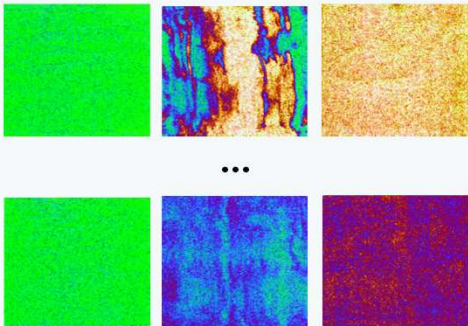


Embedding Model

Vector representation

O_1 O_2 O_3 O_4 ... O_N

WHAT WE DID

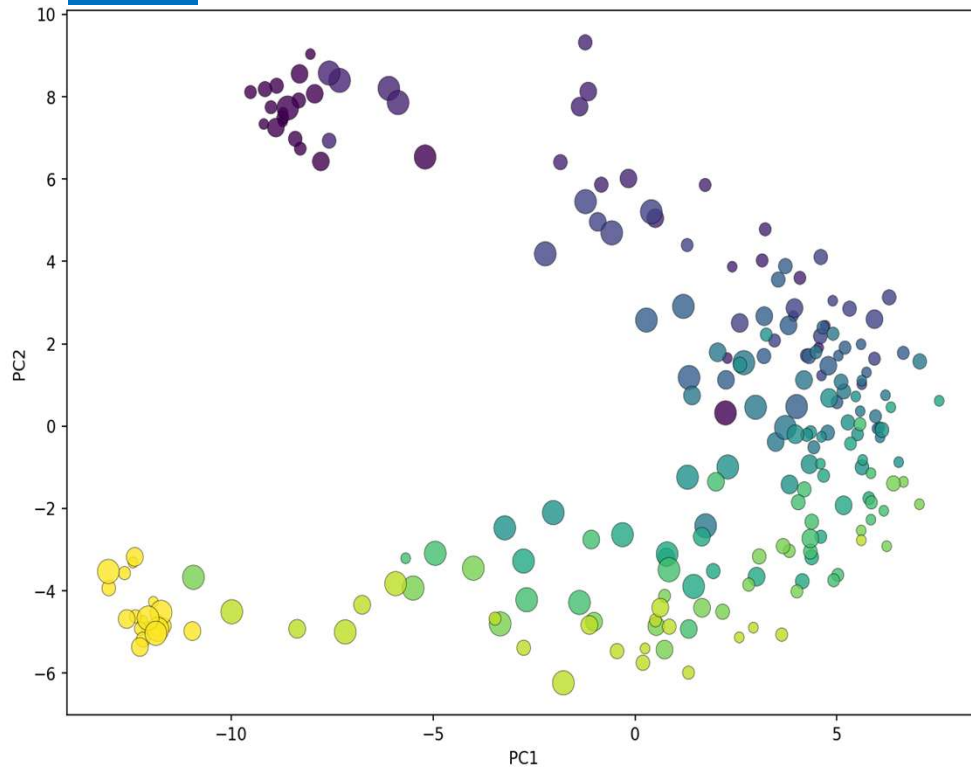


OpenAI CLIP = clip-vit-large-patch14
Google SigLIP = siglip-large-patch16-256
Facebook DINOv3 = dinov3-vitlarge16
Microsoft ResNet = resnet-50

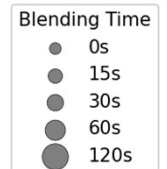
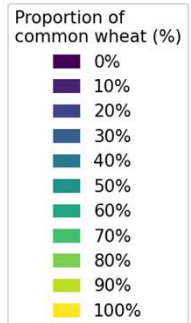
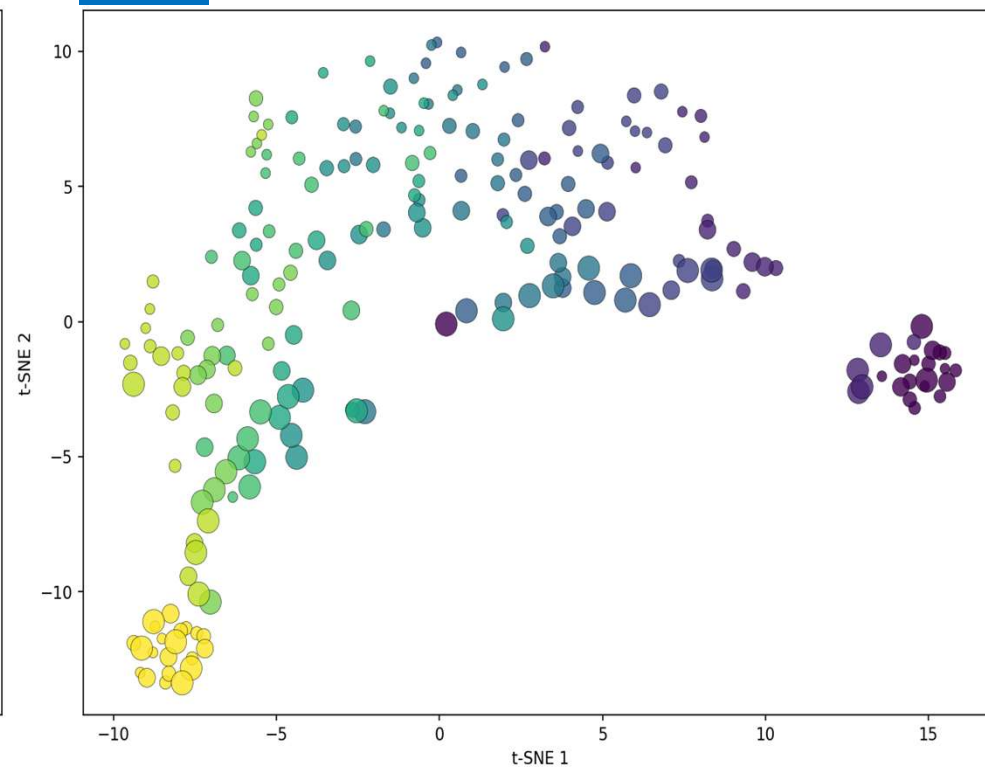
Vectors of dim = 768
Vectors of dim = 1024
Vectors of dim = 1024
Vectors of dim = 2048

Embeddings - Visualisation

PCA



t-SNE



SIGLIP on no_spikes_Derivative data + UMAP

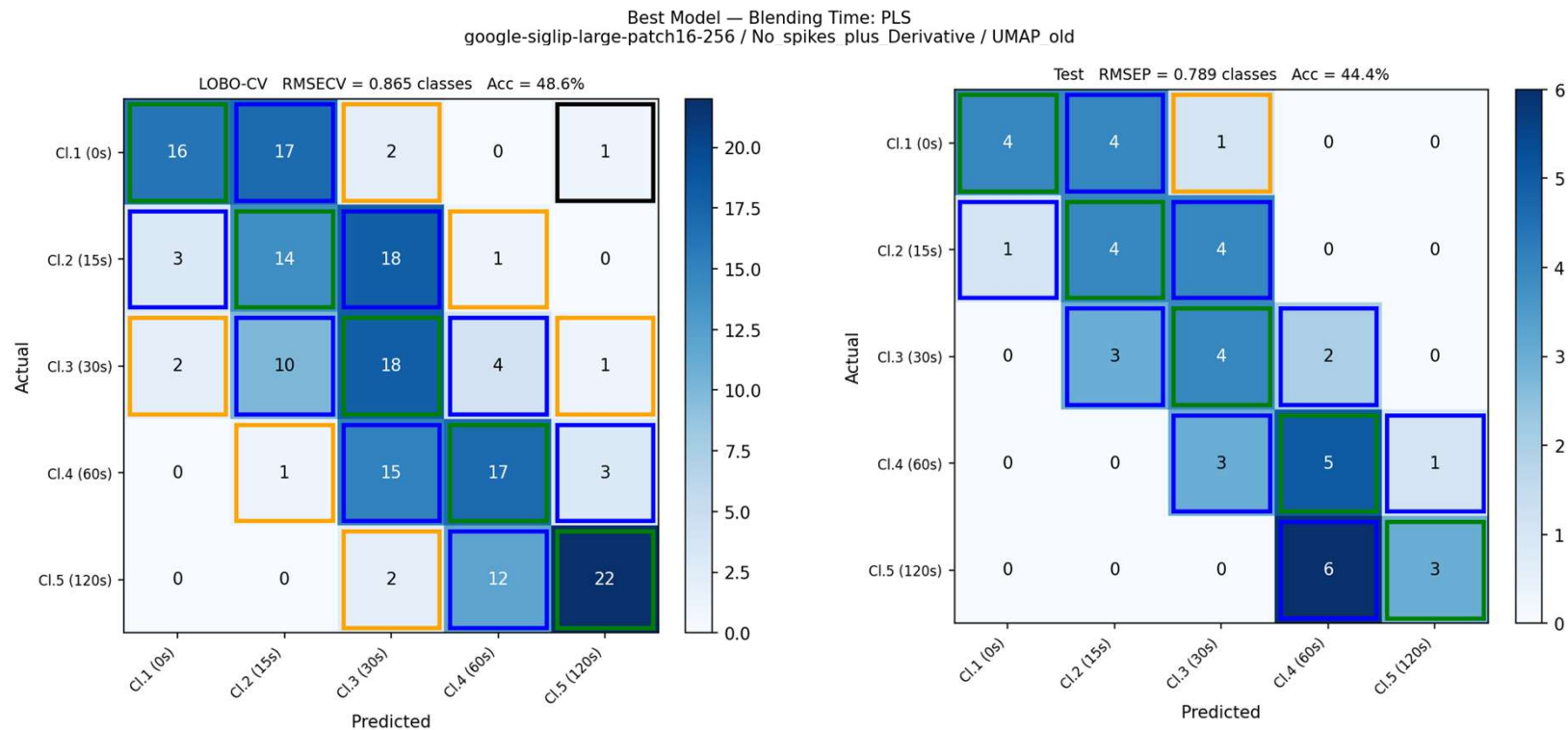
Embeddings - Prediction

- PLS, SVR-RBF, linear SVR, KNN, PCA-KNN
- Model, embeddings, dim reduction, preprocessing selection by leave-one-batch-out CV on the train set (minimization of RMSE)
- Hyperparameters selection by nested leave-one-batch-out CV on the train set (minimization of RMSE)
- Predictions = Continuous values, attribution to the closest class (blending state)

ALGORITHMS	HYPERPARAMETERS
PLS	nLV = [2, 5, 10, 15, 20]
SVR – LIN	C = [0.01, 0.1, 1, 10, 100], Standardisation = [None, StandardScaler, RobustScaler]
SVR - RBF	C = [0.01, 0.1, 1, 10, 100], Gamma: ["scale", "auto"], Standardisation = [StandardScaler, RobustScaler]
KNN	N neighbors = [3, 5, 7], Standardisation = [None, StandardScaler, RobustScaler]
PCA-KNN	N_components = [1, 2, 3, 5, 8, 10, 15, 20, 30, 50, 75, 100, 150, 200], N_neighbors = [3, 5, 7], Standardisation = [None, StandardScaler, RobustScaler]

Embeddings – Prediction Blending Time

- Selected model = PLS (LV = 2) on SigLIP embeddings of No_spikes_plus_Derivative + UMAP
- Evaluation of the model on the test set (Accuracy)

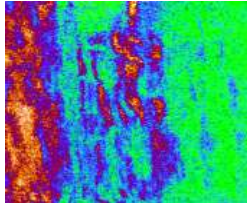


Embeddings – Prediction Blending Time

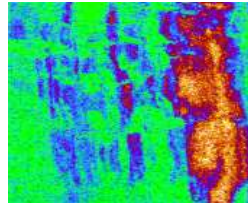
Accuracy of 44.4% ... Bad performances?

Randomness in the blending process

70%/30% - 15 s



70%/30% - 30 s

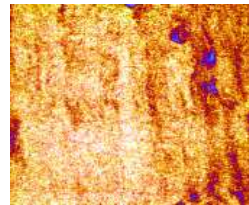


Unbalanced mixtures more homogeneous from the start

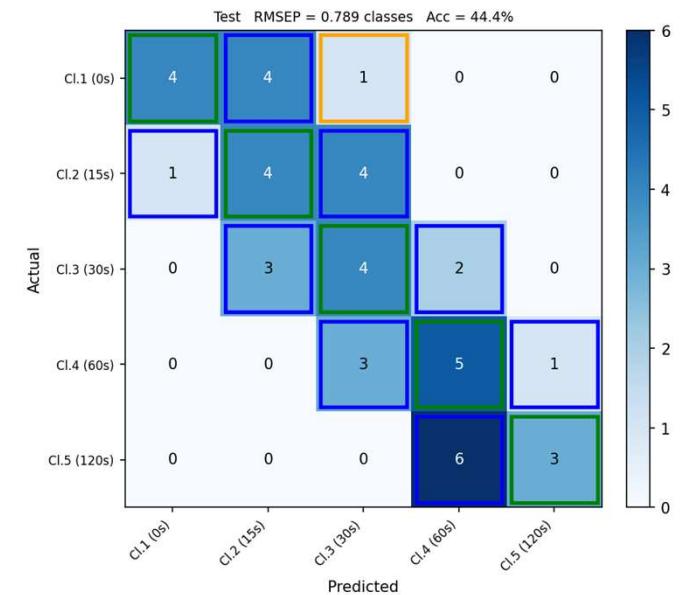
10%/90% - 0 sec



10%/90% - 15 sec



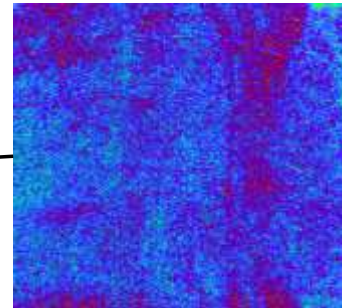
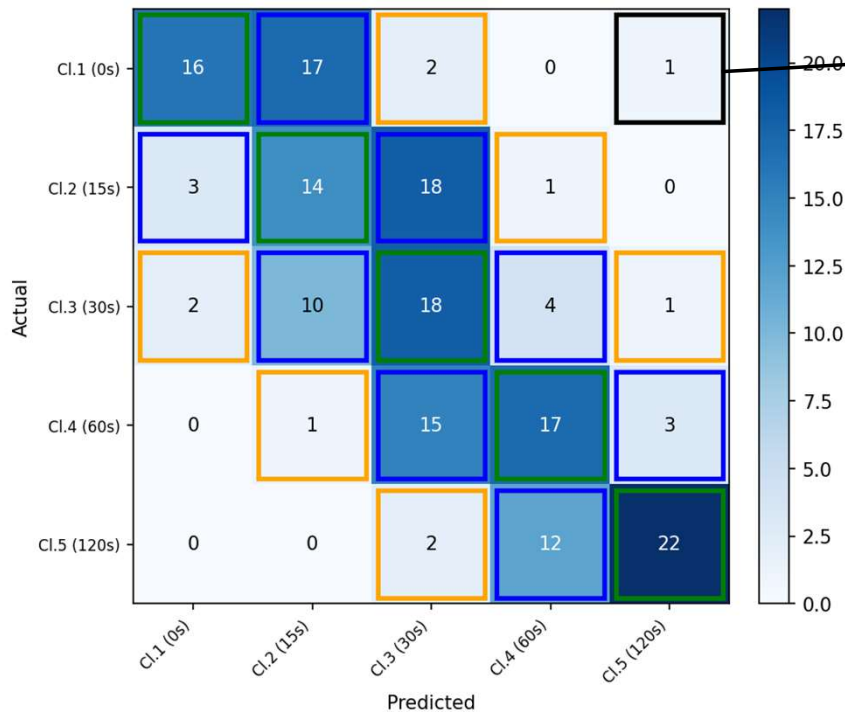
Being one class wrong is acceptable



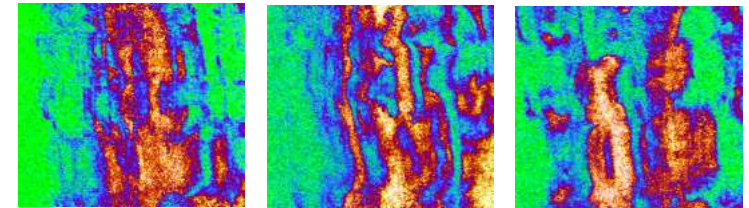
Embeddings – Prediction Blending Time

- Important CV error associated to 1 sample (70%/30% blended at 0 second)

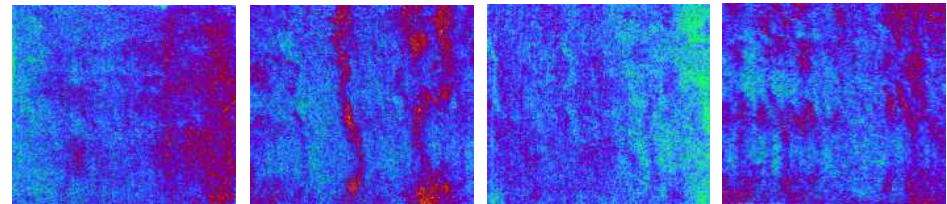
LOBO-CV RMSECV = 0.865 classes Acc = 48.6%



Different from other 70%/30% mixtures blended at 0 second



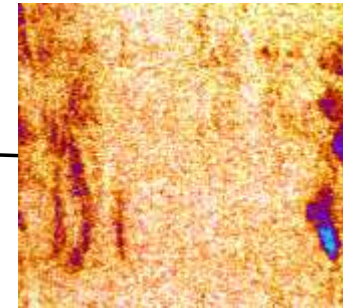
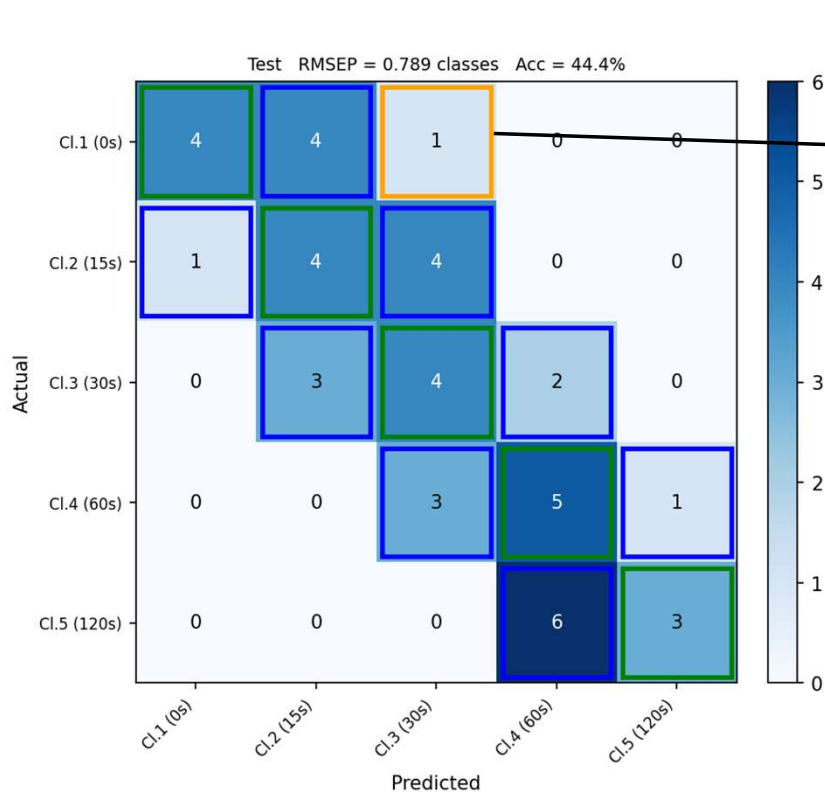
Similar to samples 70%/30% blended for 120 seconds



Metadata: "First pour onto the conveyor belt unsuccessful, the sample was returned to the jar for a second pour"

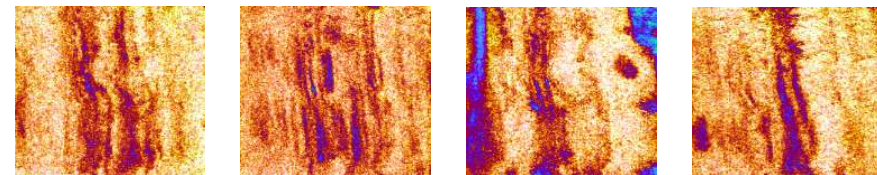
Embeddings – Prediction Blending Time

- Test error of two classes associated to 1 sample (10%/90% blended for 0 second)



10%/90% mixtures, less influence from blending (quite homogeneous from the start)

Comparison with 20%/80% blended at 30 seconds



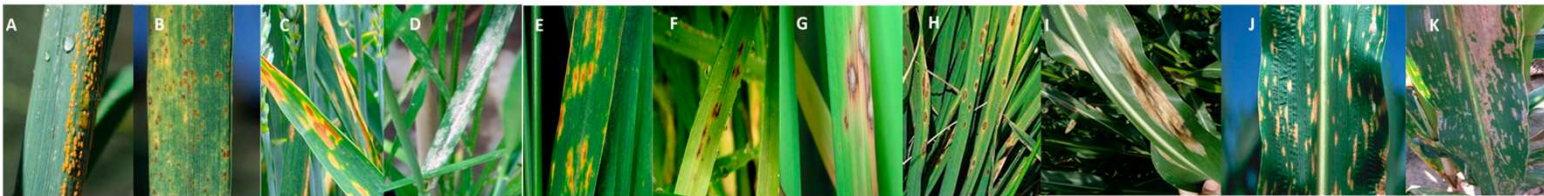
Conclusion and perspectives

- Goal was to investigate whether RGB pretrained models could be applied to pseudo-RGB representation of HSI.
→ Yes but feature extraction/selection technique is critical

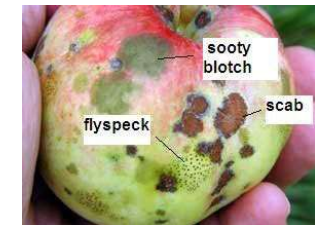
- Generalizable pipeline

- Potential for applications where spatial information matters

For example: Plant diseases, fruit defects, ... (similar spectral response but different spatial patterns)



Credits: 10.3390/plants12244162



Credits: Michigan State University

- Future work: Other use cases



**BRUSSELS
NIR 2029**



Seeing Further,
Understanding Faster

June 3rd-6th 2029



Thank you for your attention

Antoine Deryck

Research Scientist and PhD Candidate
Knowledge and valorization of agricultural products Department
Quality and authentication of agricultural products Unit
a.deryck@cra.wallonie.be | <https://www.cra.wallonie.be/fr/antoine-deryck>

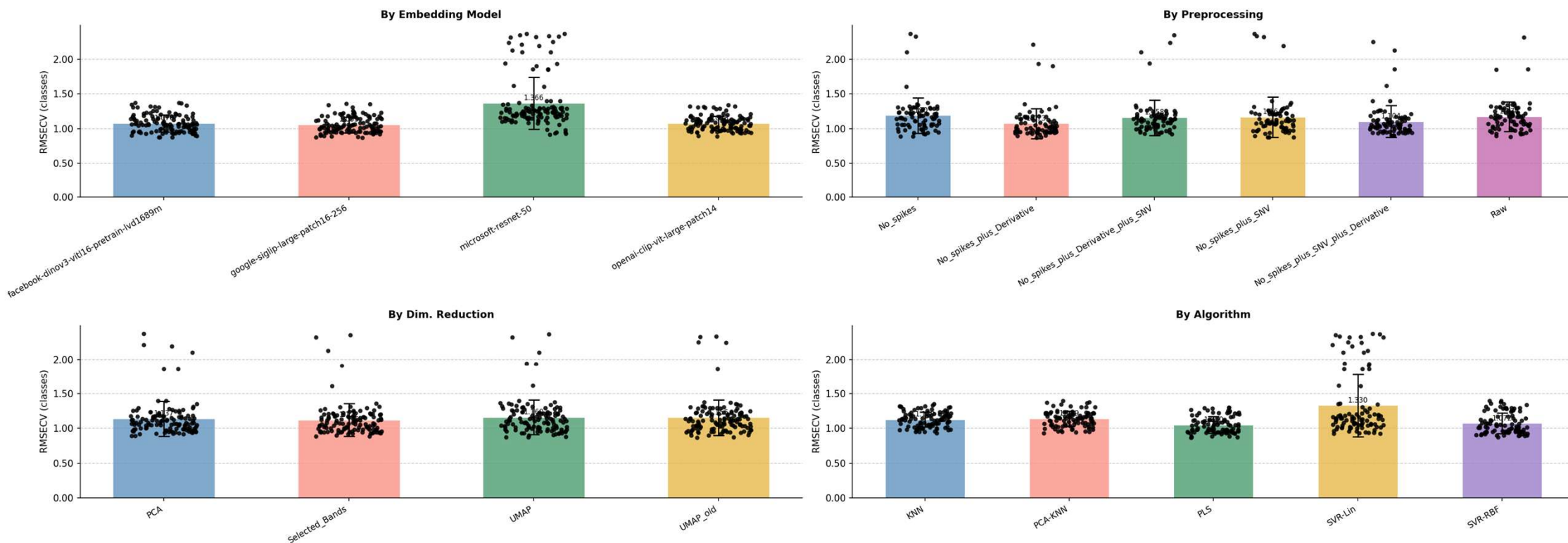


Embedded and Artificially intelligent VISion Engineering (EAVISE)
antoine.deryck@kuleuven.be



Bonus

Influence of preprocessing, dimensionality reduction, embedding models, and algorithms



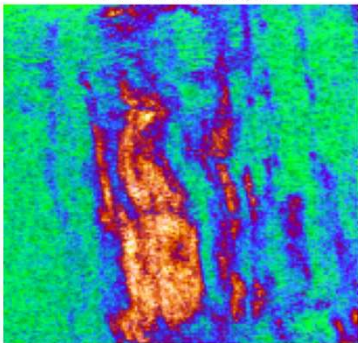
Bonus

Influence of RGB permutations on embeddings

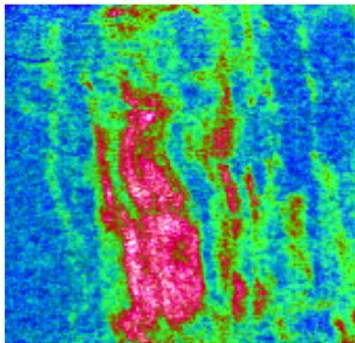
- Human eye more sensitive to the green... What about embeddings?
- Influence of channel order on performance?
- Differences between embedding models?

Example of RGB permutations on 3-000-2 (UMAP_2 on No_spikes_plus_Derivative)

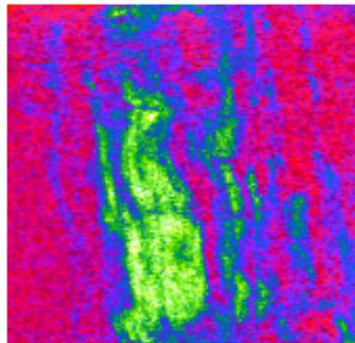
RGB (R→R, G→G, B→B) [original]



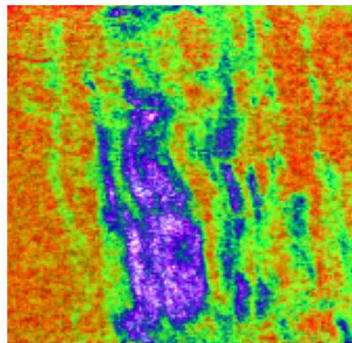
RBG (R→R, B→G, G→B)



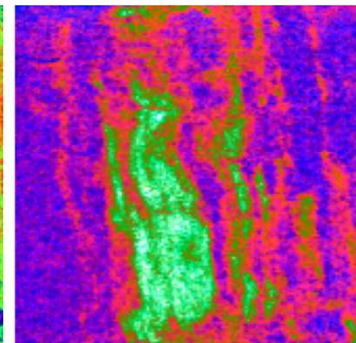
GRB (G→R, R→G, B→B)



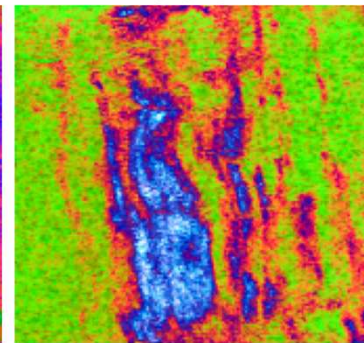
GBR (G→R, B→G, R→B)



BRG (B→R, R→G, G→B)

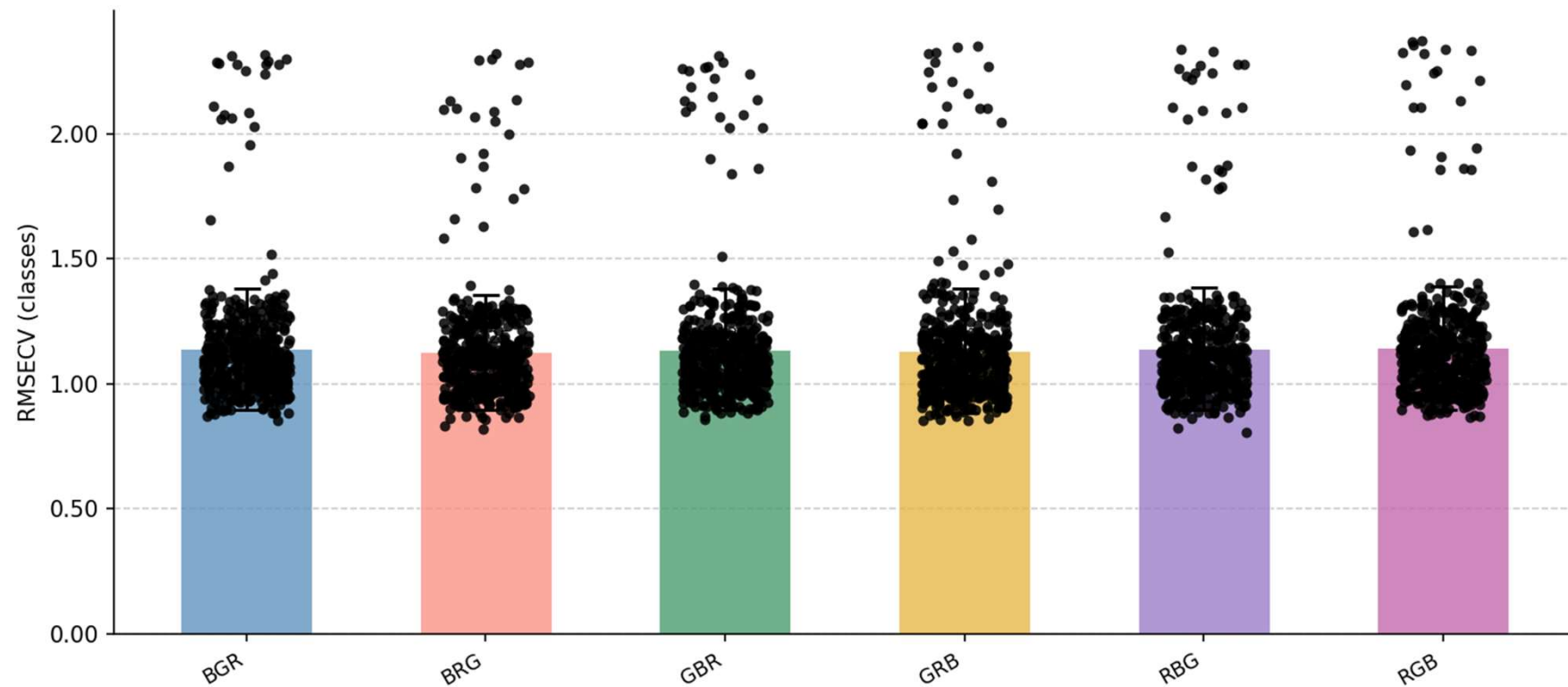


BGR (B→R, G→G, R→B)



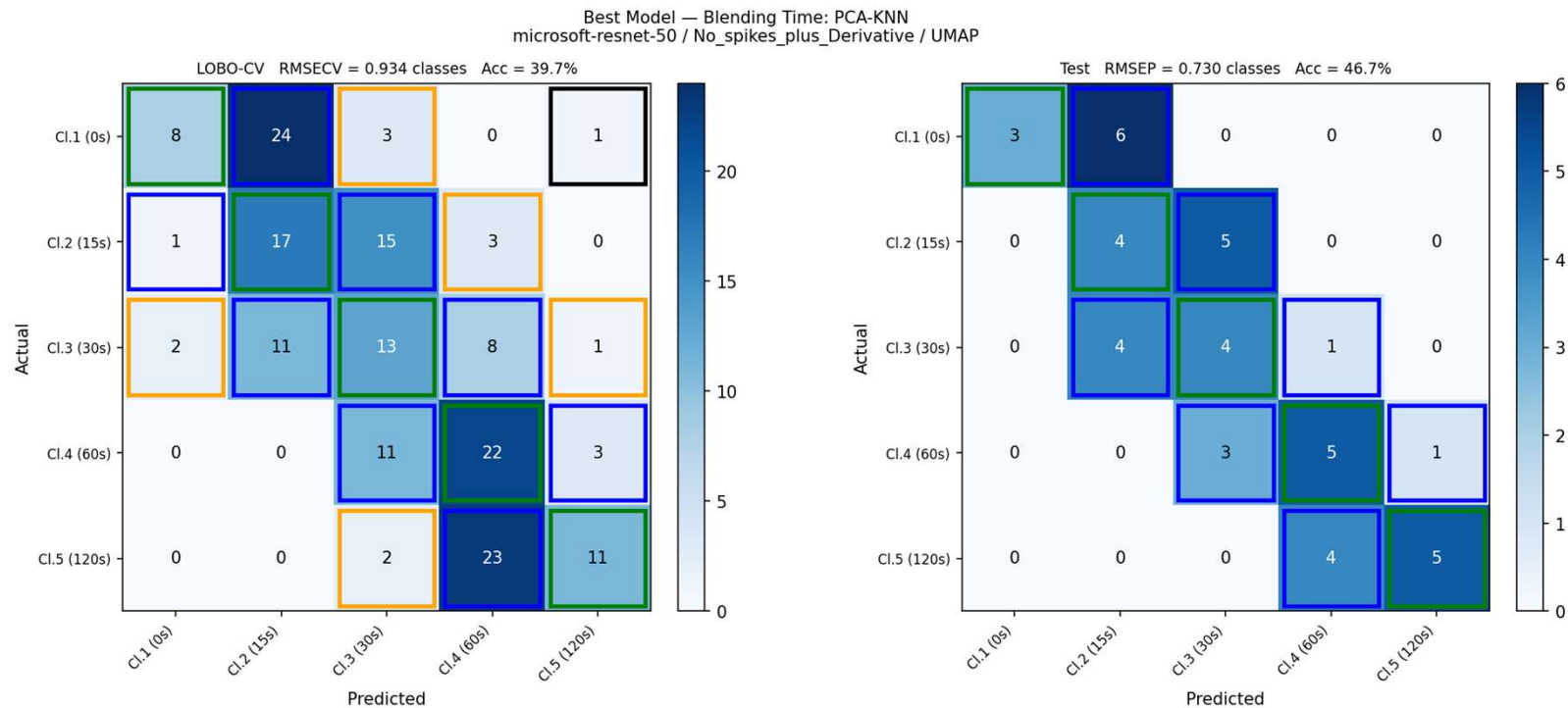
Bonus

Influence of RGB permutations on embeddings



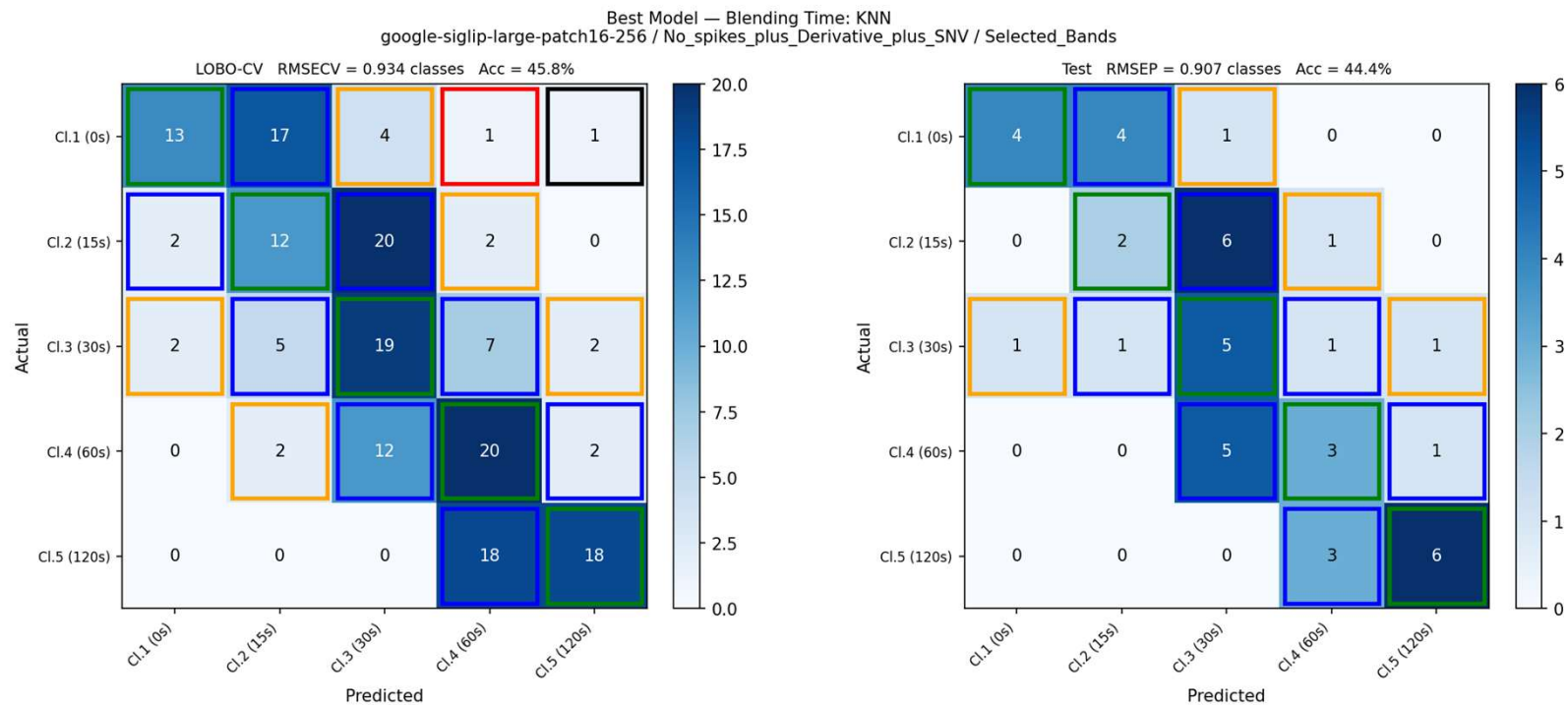
Embeddings – Prediction Blending Time

- PLS fitted on embeddings... Overfitting? What if we use PCA-KNN?
- Slightly better test performance with PCA-KNN (PC = 8, k = 7) on ResNet embeddings of No_spikes_plus_Derivative + UMAP



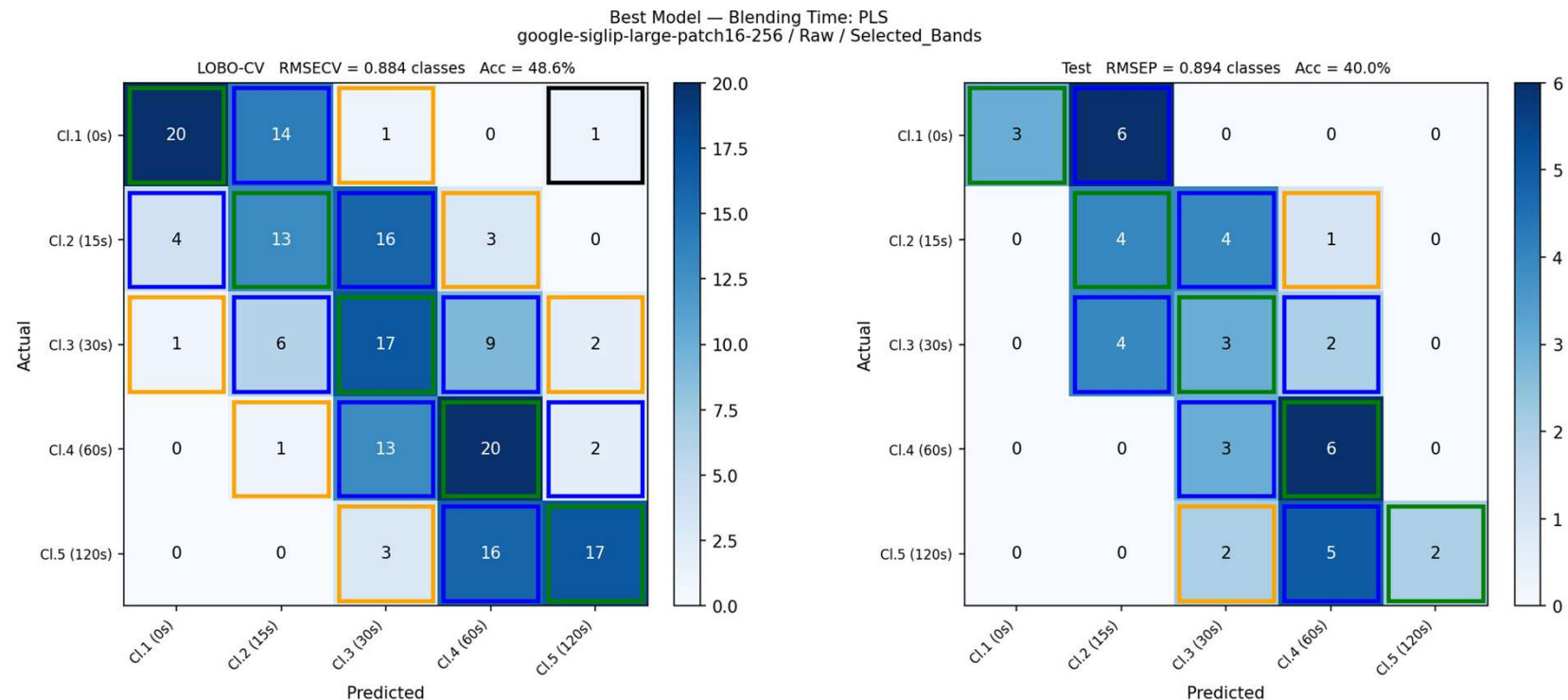
Embeddings – Prediction Blending Time

- PLS fitted on embeddings... Overfitting? What if we use KNN?
- Slightly lower test performance with KNN (k = 5) on SigLIP embeddings of No_spikes_plus_Derivative_plus_SNV + Manual Wavelengths Selection



Embeddings – Prediction Blending Time

- UMAP as dimensionality reduction. Can we get similar results with manual wavelength selection?
- Slightly lower test performance with PLS (LV = 2) on SigLIP embeddings of No_spikes_plus_Derivative

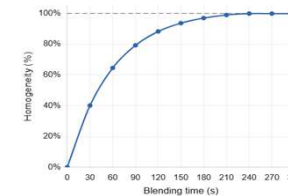
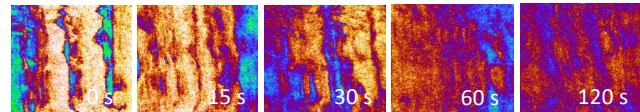


Bonus - Labels

- Embeddings seem to capture the relevant information, but what we want is a prediction model
- A loss function must be defined to select and calibrate best models for proportions and for blending time

Effect of blending decreases with time

0 s, 15 s, 30 s, 60 s, 120 s



Not a linear scale, not a typical regression case

We turn the blending time into ordinal classes (1, 2, 3, 4, 5) and treat it as regression problem



RMSE

Embeddings – Prediction Blending Time

- Selected model = PLS (LV = 2) on SigLIP embeddings of No_spikes_plus_Derivative + UMAP
- Evaluation of the model on the test set (RMSEP + RPD)

