



Human-centric algebraic machine learning

Image Classification use case Results using AML & Human Interaction - Final

Deliverable D7.2

WP7 - Use cases



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952091.

Consortium



ALGEBRAIC AI



Chantalimaud
Foundation



German
Research Center
for Artificial
Intelligence



TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

uc3m



Project

Title: ALMA: Human Centric Algebraic Machine Learning
 Acronym: ALMA
 Coordinator: Proyectos y Sistemas de Mantenimiento SL (eProsima)
 Grant agreement: 952091
 Call: H2020-EIC-FETPROACT-2019
 Program: Horizon 2020
 Start: 01 September 2020
 Duration: 54 months
 Website: <https://alma-ai.eu>
 E-mail: alma@eprosima.com
 Consortium: **Proyectos y Sistemas de Mantenimiento SL (eProsima) (EPROS)**, Spain
Champalimaud Foundation (CF), Portugal
ALGEBRAIC AI (ALG), Spain
German Research Center for Artificial Intelligence (DFKI), Germany
Inria (Inria), France
TU Kaiserslautern (TUK), Germany
FIWARE Foundation e.V. (FF), Germany
Technical Research Centre of Finland (VTT), Finland
Carlos III University (UC3M), Spain

Deliverable

Number: **D7.2**
 Title: **Image Classification use case Results using AML & Human Interaction - Final**
 Month: 54
 Work Package: WP7 - Use cases
 Work Package leader: Carlos III University (UC3M)
 Deliverable leader: Champalimaud Foundation (CF)
 Deliverable type: Report (R)
 Dissemination level: Public (PU)
 Date of submission: 2025-02-28
 Version: v1.0
 Status: Finished

Version history

Version	Date	Responsible	Author/Reviewer	Comments
v0.1	2024-12	CF	David Méndez	Partial Draft
v0.2	2025-01	CF	Fernando Martin-Maroto	Expanded Draft
v0.3	2025-01	CF	Gonzalo G. de Polavieja	Completed Draft
v0.4	2025-01	ALG	Nabil Abderrahaman	Revision
v0.5	2025-02	ALG	Antonio Ricciardo	Revision
v1.0	2025-02	CF	David Méndez	Final Revision

Executive summary

Algebraic Machine Learning (AML) is a novel machine learning technique based on algebraic representations of data. Unlike statistical learning, AML algorithms are robust regarding the statistical properties of the data and are parameter-free. AML systems can retain what they learn, including sociocultural ethical practices for human–artificial intelligence systems. The aim of the EU-funded ALMA project is to study the mathematical properties of AML and to leverage them to develop a new generation of interactive human-centric machine learning systems. These systems are intended to minimize bias in order to facilitate trust while better enabling distributed collaborative learning.

This report corresponds to *Deliverable D7.2 - Image Classification use case Results using AML & Human Interaction - Final* of the ALMA project. This deliverable covers improvements on the application of AML to image datasets, and specifically datasets of medical images. First, we show how we have extended our models to be able to train a single model for multi-class classification. We also show how we can use logistic regression to weight the importance of atoms in an AML model, increasing classification accuracy. We illustrate these improvements showing accuracies for our models in comparison with machine learning models in MNIST, Fashion-MNIST, several MedMNIST datasets as well as CIFAR-10. We also show how the characteristics of AML lead to higher classification accuracy in situations where only small amounts of data are available, compared to other machine learning models. Finally, we exhibit how some of the unique properties of AML models allow for user interactivity, by letting the user classify images based on a subsection of choice.

Contents

Executive Summary	3
Contents	4
Acronyms	5
1 Introduction	6
2 Multi-class classification	7
3 Atom weights via logistic regression	7
3.1 Methods and results	8
4 AML models for small datasets	10
5 Interactivity	11
References	15

Acronyms

AML	Algebraic Machine Learning.
CNN	Convolutional Neural Network.
DL	Deep Learning.
MLP	Multilayer Perceptron.

Overview

Since D7.1 *Image Classification use case Results using AML & Human Interaction - Initial*, several improvements were made on our image classification pipeline, which we detail in the sequel. In particular:

- Overall, our methods are now more efficient, allowing for analysis of images in higher resolution and with more than one color channel.
- We introduce a pipeline to train models for multi-class classification, Section 2.
- We use logistic regression to assign weights to atoms to improve classification accuracy, Section 3.
- We explore how the unique characteristics of AML provide a significant edge in classification accuracy in regimes with little data availability, compared to some Deep Learning (DL) models, Section 4.
- We show how algebra restrictions can be exploited to classify images based on subsections, and how that can be used in an interactive setting, Section 5.

1 Introduction

Our initial experiments regarding image classification focused on datasets in which the information required for classification was still visible in the discretization of the images to black and white, such as the case of the MNIST handwritten character dataset, [1], as to avoid an explosion in the number of constants and relations required to represent the data. Indeed, if we wanted to directly embed all the information in color or even grayscale images in an atomized semilattice in the naïve way, the amount of constants and relations required would make this computationally very challenging. Even for datasets like MNIST, consisting of 28×28 images with 256 grayscale values, we would require 200,704 constants only to be able to compose the images. To that, we would have to add almost 400,000 relations to ensure that the model can take into account the relations between gray intensity levels.

In the initial version of this deliverable, D7.1, we introduced an extension of the AML engine capable of understanding real values by using infinite chains of continuous values. We then exhibited the initial capabilities of this engine by classifying images in two image-based binary classification datasets: PneumoniaMNIST, [2], and BreastMNIST, [3]. The accurate classification of these images, by their nature, is not compatible with discretization, as the essential information they contain lies extensively in the grayscale levels.

Since then, we have analyzed datasets in higher resolutions, as well as datasets of color images, with great success. In the sequel, we will introduce the improvements we made to the AML pipeline, as outlined in the Overview. The data we obtained in different experiments, backing the discussion of our improvements, can be found in Sections 3 and 4. These data indicate that the classification accuracy of AML models is within the range of what is achievable by reasonable Multilayer Perceptron (MLP) networks. Nonetheless, MLP with such classification accuracies usually require the use of a validation dataset to perform hyperparameter tuning, requiring multiple train runs to find the best possible parameters. This is unlike our AML pipeline where only train data is used, and only one train run is performed.

A more detailed explanation of the methodology used in the experiments reported here can be found in [4], a recently submitted manuscript providing a self-contained introduction to the AML theory and the involved algorithms which also discusses results regarding the image datasets analyzed here as well as some formal problems this deliverable does not cover, with the objective of validating the AML learning paradigm.

2 Multi-class classification

In our initial image classification results, when dealing with multi-class problems such as MNIST, our methods involved training and ensembling one binary classifier per class, its role being to decide whether an example was part of the class it is associated with or not. Since then, we have moved to a learning regime where we use one single model containing one classification constant per class. Our training algorithm now uses Sparse Crossing to enforce that the constant associated to a given class is exclusively below images of that given class, within the order relation in an atomized semilattice. Atoms can have one or several classification constants in them, allowing for a direct compression of the information, as similarities between images of different classes can often lead to similar atoms. But more importantly, training only one model, compared to one model per class as we used to, leads to a great decrease in training time.

The use of this method does not negatively impact classification accuracy. Quite the opposite, we have observed higher accuracy in MNIST compared with our earlier attempts. Indeed, when classifying an image as the class discriminated by the fewest number of atoms associated to that class (henceforth, classification via fewest misses), we observe an increase in test accuracy, from 97.4% to 97.63%. But more importantly, the efficiency gained with this approach is one of the key improvements that allowed us to analyze larger image datasets, as we see in the sequel.

3 Atom weights via logistic regression

Classification via fewest misses, as described above, provides us with a quick and naïve way of using trained AML models to classify unseen images. However, this method implicitly assumes that every atom in an AML model has the same importance, and ignores the potential correlations between atoms, both in the same and in different classes.

For example, it is quite common for atoms of smaller size to be more numerous than larger atoms in a trained model. Generally, smaller atoms find the common rules behind data belonging to a class, whereas the larger atoms are there to aid in the classification of out of distribution examples. As such, the larger atoms could perhaps be given reduced importance, as they will likely generalize poorly to unseen data. Even for the smaller atoms, there will likely be correlations in atom activations that, if exploited, may help improve classification accuracy. These correlations may be quite subtle and hard to detect, and in fact methods we have experimented with using correlations of pairs have not proved successful at improving classification accuracy significantly, compared with using fewest misses.

A great way to detect these subtle group correlations and aid in classification is to weight the atoms via logistic regression. Essentially, we use a linear layer with as many inputs as atoms in the trained model, and as many outputs as data classes. The inputs represent the atoms of the algebra in a fixed order, so given an input image, it is converted to an array in which a 1 signifies that the atom in that position intersects with the image, and a -1 signifies that it does not. The use of -1 instead of 0 allows for the linear layer to use the information of atoms not intersecting the image to move the classification towards any class whose classification constant is not in that atom. And indeed, atoms in a given class can more commonly discriminate images of some other class, compared to all others, providing us with exploitable information.

This linear layer is then trained using Pytorch via logistic regression, thus minimizing cross entropy loss. This ensures that the weights will not be uniform, as even if the linear layer was initially weighted to mimic classification via fewest misses, the softmax of the linear layer outputs would provide similar probabilities for classes discriminated by similar amounts of atoms, which is penalized in this regime. We thus observe large departures from fewest misses classification, generally leading to significant improvements in classification accuracy.

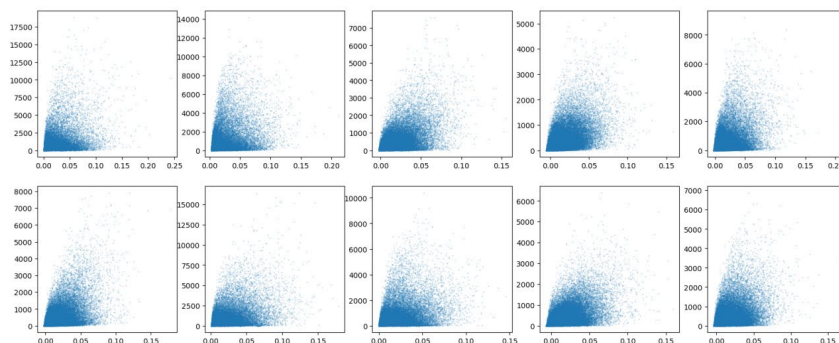


Figure 1: In a linear layer trained via logistic regression to assign weights to atoms in an AML model trained to classify images from the MNIST dataset following the process described in Section 3, and for atoms associated to each of the 10 classes, a plot of the weight assigned to each atom towards classifying the image as belonging to that same class (x-axis), compared to how many images of other classes it does not intersect with (y-axis).

In the particular case of MNIST, accuracy increases from the already mentioned 97.63% to 98.43%, near the best accuracy we have seen achieved by an MLP in this dataset. Crucially, we train the linear layer exclusively on the same training dataset used for the algebra model, as our experiments showed that if the algebra model is mature enough, using additional data to train the linear layer tends to slightly decrease accuracy, around 98.3% in the particular case of MNIST. On the other hand, we can completely overfit this layer to the same training data used to obtain the algebra without losing generalization accuracy. This further supports the already existing evidence that AML models do not overfit to training data, thus finding features in data that are not necessarily frequentist. And indeed, the methodology used in the experiments whose results are reported here has been to train the linear layer until we stop seeing significant improvement in train loss, and at least until the mean loss for the examples in the train dataset is below 10^{-6} . We thus do not use validation data, whereas in order to achieve the best possible accuracies in traditional DL models it is usually necessary to use a validation dataset to tune hyperparameters.

Another observation we can deduce from our experiments is that the linear layer is indeed taking advantage of group correlations that would not be easily detectable in a straightforward manner. Indeed, if we plot the weights assigned to each atom with respect to their discriminatory power we observe quite a spread, for which we have not been able to find a simple explanation, see Figure 1.

3.1 Methods and results

We use this method to classify images in MNIST, Fashion-MNIST, [5], several of the datasets in MedMNIST [6] (PneumoniaMNIST [2], OrganCMNIST [7], BloodMNIST [8], DermaMNIST [9, 10]) and CIFAR-10, [11]. Of these, we analyze BloodMNIST and PneumoniaMNIST both in 28×28 and 64×64 resolution. Importantly, BloodMNIST, DermaMNIST and CIFAR-10 all use color images, which are now within the capabilities of our models. Figure 2 displays example images from each of the analyzed datasets. The results of our experiments are displayed in Table 1. The reflected accuracy values for the MLP and Convolutional Neural Network (CNN) models correspond to the test accuracy of the model that attained the lowest validation loss, out of hundreds of models obtained in a hyperparameter sweep performed using Ray Tune, [12].

In particular, regarding the MLP models, we consider a family of two, three and four hidden layer multilayer perceptrons with ReLU activations. More concretely, we perform a grid search over the

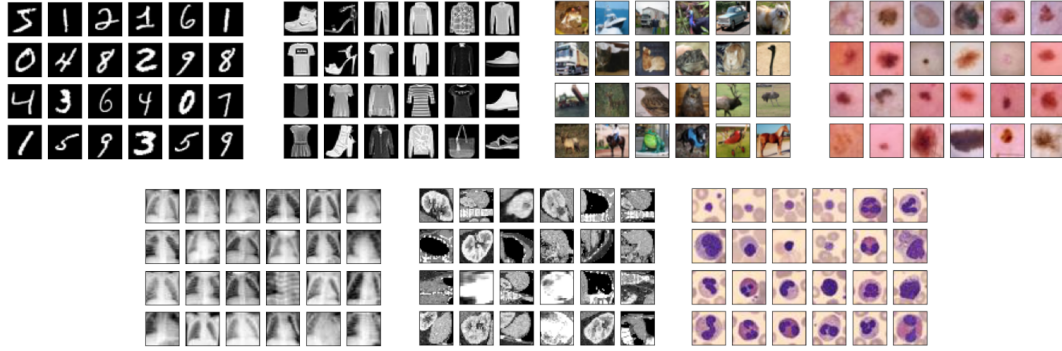


Figure 2: Example images for each of the analyzed datasets: MNIST, Fashion-MNIST, CIFAR-10, DermaMNIST, PneumoniaMNIST, OrganCMNIST and BloodMNIST.

number of neurons in the first hidden layer (512, 2048 or 4096 hidden units) and the second hidden layer (256, 1024 or 2048 hidden units), using Ray Tune, [12], with the goal of minimizing validation loss. The third layer, when it exists, is allowed to have 128, 256 or 512 hidden units, and the fourth layer, when it exists, can have 128 or 256 hidden units. For the third and fourth layers, a random sample is performed for the sizes. We perform 90 runs using two layers, 180 runs using three, and an additional 90 runs with four layers, for a total of 360 train runs. Training runs for 240 iterations, or until the validation loss does not improve for 10 iterations. In each run, we uniformly sample the learning rate ($5 \cdot 10^{-4}$, 10^{-4} , $5 \cdot 10^{-5}$ or 10^{-5}) and the L2-regularization coefficient (10^{-3} , $5 \cdot 10^{-4}$, 10^{-4} or 0). We use the ADAM optimizer to minimize cross-entropy loss. The numbers reported below the accuracy of the MLP models in Table 1 correspond to the number of hidden units in the selected model.

In the case of the CNN models, we perform a hyperparameter sweep over a family of networks with two or three convolutional layers and three linear layers. We use the ReLU activation function between all layers. Convolutional layers have 32 output channels and use a padding and a stride of one, and are followed by a dropout layer with a 20% probability of zeroing out elements. The intermediate convolutional layers are also followed by a 2D-maxpool with a stride of two. The first and second linear layers respectively have 128 and 64 outputs, the third layer having as many outputs as the number of classes in the dataset. We use square kernels for the convolutional and maxpool layers, performing a grid search on their heights/widths for the convolutional layers. The possible values of these widths are independent for the three layers and can be 3, 5 or 7. Note that, in the case of networks with three convolutional layers, not all combinations are feasible given the resolution of the images, particularly in the datasets with 28×28 and 32×32 images. The kernel length of the 2D-maxpools is set to the maximum of 2 and the floor of half of the kernel length for the convolutional layer. For each potential value in the grid we do 10 train runs for a total of 90 runs per dataset for networks with two convolutional layers and up to 270 runs for networks with three convolutional layers. We uniformly sample the learning rate ($5 \cdot 10^{-4}$, 10^{-4} , $5 \cdot 10^{-5}$ or 10^{-5}) and the L2-regularization coefficient (10^{-3} , $5 \cdot 10^{-4}$, 10^{-4} or 0). We use the ADAM optimizer to minimize cross-entropy loss. The numbers below the reported accuracy are the kernel lengths for the convolutional layers in the selected model.

We remark that the results outlined here for MLPs and CNNs are not necessarily the best achievable using these tools. Indeed, larger networks and different regularization techniques could potentially yield better results, and different datasets often require widely different architectures to achieve the best possible classification accuracy on them. The approach used here was to perform a systematic analysis within a fixed set of possible architectures that would work reasonably well for all involved datasets. Nonetheless, this highlights the difficulty of designing architectures for DL models, as every dataset

Dataset	MLP	CNN	AML fewest misses	AML logistic regression
MNIST	98.46%	99.33%		
28 × 28, 10, 50000/10000/10000	(2048, 1024, 128)	(5, 3, 5)	97.63%	98.43%
Fashion-MNIST	89.52%	92.32%		
28 × 28, 10, 50000/10000/10000	(4096, 256)	(3, 3, 7)	87.27%	89.47%
CIFAR-10	54.58%	76.12%		
32 × 32, 10, 50000/5000/5000	(4096, 256)	(5, 5, 5)	48.56%	53.60%
DermaMNIST	65.94%	65.84%		
28 × 28, 7, 7007/1003/2005	(4096, 2048, 128, 256)	(3, 5)	73.47%	74.21%
PneumoniaMNIST	87.66%	88.14%		
28 × 28, 2, 4708/524/624	(2048, 256, 512)	(3, 5, 7)	84.62%	85.90%
PneumoniaMNIST	87.50%	87.82%		
64 × 64, 2, 4708/524/624	(512, 256, 256)	(5, 5, 3)	84.13%	84.93%
OrganCMNIST	76.44%	87.50%		
28 × 28, 11, 12975/2392/8216	(4096, 2048, 128)	(5, 7, 5)	81.28%	86.75%
BloodMNIST	85.30%	93.28%		
28 × 28, 8, 11959/1712/3421	(4096, 1024, 256)	(7, 7, 3)	85.26%	90.93%
BloodMNIST	89.16%	96.38%		
64 × 64, 8, 11959/1712/3421	(2048, 256, 256)	(7, 7, 5)	87.55%	92.90%

Table 1: Accuracy results for AML models, in comparison with baseline MLP and CNN models. Below the dataset name, we display image resolution, number of classes and train/validation/test dataset splits. MLP and CNN models are selected via hyperparameter tuning as explained in Section 3.1. The number below their accuracy results represent the number of units in the hidden layers of the selected model, in the case of the MLPs, and the widths/depths of the square kernels in the convolutional layers, in the case of the CNNs.

will require extensive hyperparameter tuning and tests of different architectures to ensure that the best possible accuracy is achieved. On the other hand, the procedure used for AML models is straightforward, only requires a single train run and does not need the use of a validation dataset.

Overall, we observe that the classification accuracy of AML models, when weighting atoms via logistic regression, is comparable to the accuracy of the best obtained MLP models. They sometimes even have accuracies comparable to those achievable by CNN models, even though the trained AML models have no prior knowledge on the proper biases towards image classification tasks, such as translational or rotational invariance. However, in datasets sensitive to local information that can in principle be located anywhere in the image, such as CIFAR-10, AML models perform unsurprisingly poorly compared with the more appropriately biased convolutional layers. In future work we will explore how to effectively and efficiently incorporate these biases into AML models, hopefully allowing us to overcome this gap in accuracy.

4 AML models for small datasets

In regimes where training data is not abundant, DL models can be subject to overfitting and are rarely the most accurate way of classifying datasets. AML models do not suffer from overfitting and do not require validation data, thus we can expect them to be more accurate in the extreme of having very

little data. To test this hypothesis, we have experimented with training AML and DL models on 1000 examples from MNIST, Fashion-MNIST, and CIFAR-10. We still allowed the use of the full validation dataset in the DL models, thus giving them the greatest possible chance to exhibit the best achievable results in the respective regimes. The results of these experiments can be found in Table 2.

We observe that AML together with logistic regression consistently beats MLP models in classification accuracy by quite a margin, and both in MNIST and CIFAR-10 even fewest misses classification is more accurate than the best obtained MLP network. We observe, however, that CNN models are still more accurate than AML models in these experiments. This again suggests the need for further research regarding adding appropriate biases for image classification in the AML regime.

Dataset	MLP	CNN	AML fewest misses	AML logistic regression
MNIST	88.70%	95.03%		
28 × 28, 10, 1000/10000/10000	(4096, 256, 128)	(3, 7, 5)	90.24%	91.56%
Fashion-MNIST	80.73%	82.18%		
28 × 28, 10, 1000/10000/10000	(2048, 256)	(3, 5)	79.62%	81.89%
CIFAR-10	36.23%	43.05%		
32 × 32, 10, 1000/10000/10000	(4096, 256, 512)	(3, 7, 5)	36.33%	38.49%

Table 2: Classification accuracies for MNIST, Fashion-MNIST, and CIFAR-10, training on 1,000 examples. Below the dataset name, we display the image resolution, number of classes and train/validation/test dataset splits. MLP and CNN models are selected via hyperparameter tuning as explained in Section 3.1. The number below their accuracy results represent the number of units in the hidden layers of the selected model, in the case of the MLPs, and the widths/depths of the square kernels in the convolutional layers, in the case of the CNNs.

5 Interactivity

The AML theory allows for significant analysis and interactivity that is not easily achievable using DL models. In particular, we can use the restriction algebras to reclassify images based on a subsection of interest. To that purpose, remember that given an atomized semilattice model M over a set of constants C , and given $Q \subseteq C$ a subset of the constants, we can obtain $M|Q$, the restriction of M to Q . This happens to be the subalgebra of M generated by Q and can easily be computed via atomizations, since an atomization of $M|Q$ can be found by taking the atoms of M and restricting them to Q , as shown in [13, Section 4].

We have developed a widget letting a user reclassify a single example within an image based dataset by computing the restriction of the trained algebra to a subsection of the image. This lets the user more finely decide the region of interest in a misclassified image, as to see if focusing in a particular region allows for the error to be fixed. The system is also able to offer the user the accuracy of the restriction model to that same subsection, in the entire dataset. The user can get data regarding the image classification for the AML model, both using fewest misses and a pretrained logistic regression classifier, obtained as introduced in Section 3.

We further allow for the logistic regression layer to be fine-tuned by using the restriction of the training set to the region of interest. This is done by changing the input value associated to an atom not present in the restriction area to 0. Since the value associated to a non-activated atom is -1 , this lets the model

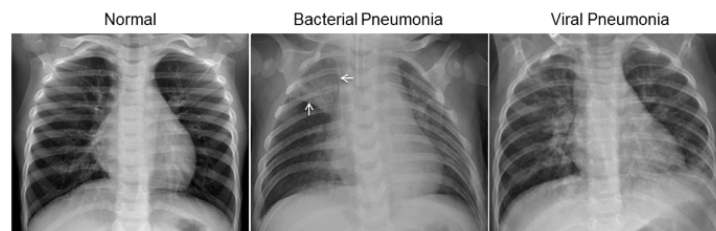


Figure 3: Typical presentation of bacterial and viral pneumonia on an X-ray image, compared to pneumonia-free lungs, [2, Figure S6]. Usually, bacterial pneumonia manifests as a local area of strong opacification, whereas viral pneumonia manifests with a more diffuse pattern in both lungs.

distinguish between the non-activation of an atom that intersects the subsection of interest and the non-presence of an atom. We observe in the examples we have worked with that while fine-tuning can help increase classification accuracy, it is generally not required, as accuracy for the logistic regression model remains higher than using the restriction algebra directly.

The key insight here is that there may be images where the key information needed to correctly classify them resides in a definite area. Furthermore, depending on the dataset, it may be the case that this strong locality is only observed in a handful of images, and the area of interest varies between different images. Thus, when one of these images is misclassified, we could recalculate the classification based on the area where the important information resides. We exemplify this here by using PneumoniaMNIST, [2]. This dataset is composed of lung X-ray images where around 26% of the images (label 0) correspond to healthy lungs, and the remaining 74% of images (label 1) correspond to lungs with pneumonia. Pneumonia is visible in the X-ray images through abnormal opacification, which can be both local or global, c.f. Figure 3.

The main interface of the widget, Figure 4, consists of selectors for the split of the dataset and the image to work with, as well as what form of selection to use to classify based on subsections of the image. It also includes checkboxes to select whether to only use the AML model, the logistic regression on the atoms, or both.

The image is displayed twice, in small and large sizes. The smaller image makes it easier to notice any potential blurriness, whereas the larger image allows for more accurate free-form selections.

Although not very common, a handful of examples from the testing dataset are misclassified as healthy lungs while containing small areas of blurriness, corresponding to bacterial pneumonia. These images are often correctly classified when selecting the appropriate section showing the infection, Figure 5.

Similarly, X-ray images taken from an out-of-distribution angle and distance can also be misclassified, possibly due to the heart and/or bones occupying significantly more or less space than in most other images in the dataset, and these classification errors are often overcome by focusing on subsections of the images, see Figures 6 and 7.

Generally, we observe that the model weighing the atoms via logistic regression maintains slightly higher accuracy than classification via fewest misses even without the need to fine-tune. Fine-tuning does often allow us to improve accuracy when classifying based on the restrictions, but it also often causes the model to perform worse when classifying the full images.

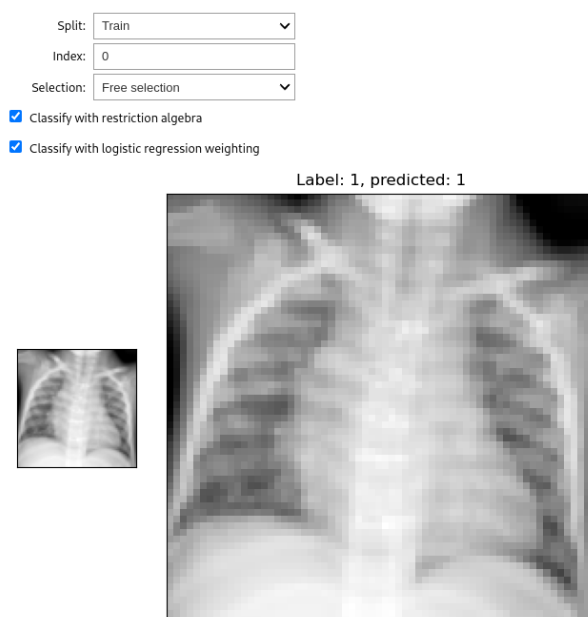


Figure 4: The main interface of the widget. This consists of selectors for the split of the dataset to use (training, validation or testing), the image within the dataset and a selection mode of which section to classify. The available selection modes are classifying the whole image, manually drawing a rectangular area over it, or selecting the left or right half of the image.

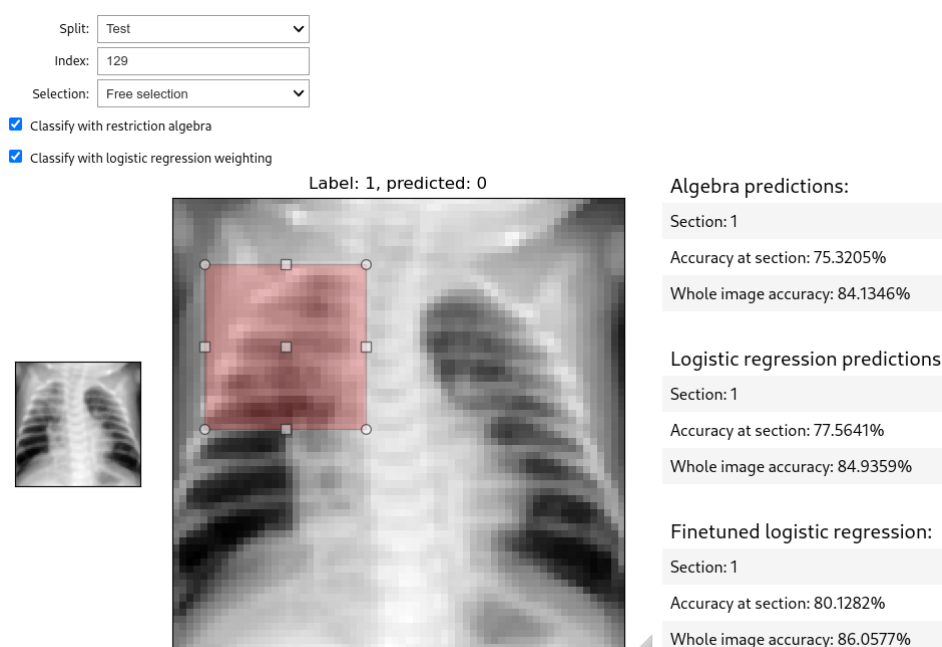


Figure 5: A misclassified image is correctly classified based on a section with visible blurriness, caused by localized bacterial pneumonia.

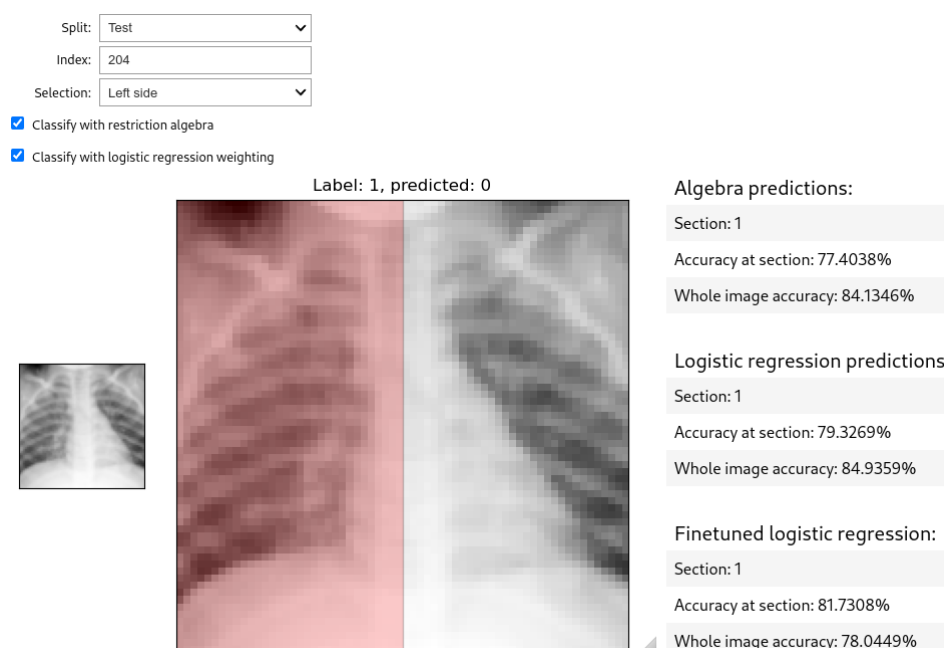


Figure 6: An X-ray image with visible pneumonia gets misclassified as healthy, possibly because the image is taken from an out of distribution angle, making the heart occupy a significant portion of the right side of the image. The image gets correctly classified when only considering its left side.

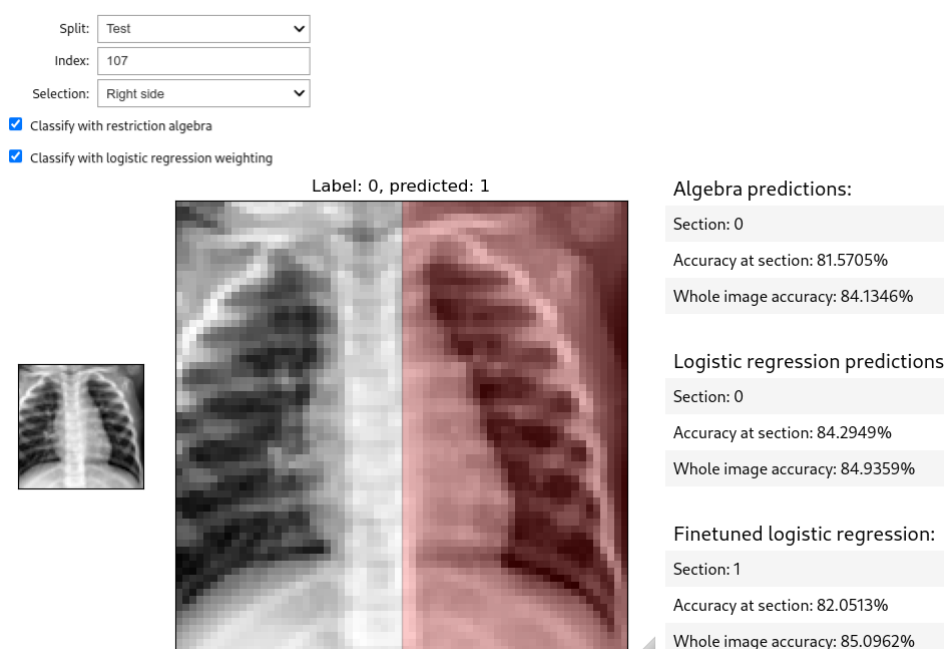


Figure 7: An off-center image of a clear, healthy lung is misclassified as exhibiting pneumonia, but is correctly classified using fewest misses and the weighted atoms. However, in this particular case, the fine-tuned logistic regression classifier fails.

References

- [1] Yann LeCun and Corinna Cortes. *MNIST handwritten digit database*. 2010. URL: <http://yann.lecun.com/exdb/mnist/>.
- [2] Daniel S. Kermany et al. “Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning”. In: *Cell* 172.5 (Feb. 2018), 1122–1131.e9. ISSN: 0092-8674. DOI: 10.1016/j.cell.2018.02.010. URL: <http://dx.doi.org/10.1016/j.cell.2018.02.010>.
- [3] Walid Al-Dhabyani, Mohammed Gomaa, Hussien Khaled, and Aly Fahmy. “Dataset of breast ultrasound images”. In: *Data in Brief* 28 (2020), p. 104863. ISSN: 2352-3409. DOI: <https://doi.org/10.1016/j.dib.2019.104863>. URL: <https://www.sciencedirect.com/science/article/pii/S2352340919312181>.
- [4] Fernando Martin-Maroto, Nabil Abderrahaman, David Mendez, and Gonzalo G. de Polavieja. *Algebraic Machine Learning: Learning as computing an algebraic decomposition of a task*. 2025. arXiv: 2502.19944 [cs.LG]. URL: <https://arxiv.org/abs/2502.19944>.
- [5] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. 2017. arXiv: 1708.07747 [cs.LG]. URL: <https://arxiv.org/abs/1708.07747>.
- [6] Jiancheng Yang et al. “MedMNIST v2 - A large-scale lightweight benchmark for 2D and 3D biomedical image classification”. In: *Scientific Data* 10.1 (Jan. 2023). ISSN: 2052-4463. DOI: 10.1038/s41597-022-01721-8. URL: <http://dx.doi.org/10.1038/s41597-022-01721-8>.
- [7] Xuanang Xu et al. “Efficient Multiple Organ Localization in CT Image Using 3D Region Proposal Network”. In: *IEEE Transactions on Medical Imaging* 38.8 (Aug. 2019), pp. 1885–1898. ISSN: 1558-254X. DOI: 10.1109/tmi.2019.2894854. URL: <http://dx.doi.org/10.1109/TMI.2019.2894854>.
- [8] Andrea Acevedo et al. “A dataset of microscopic peripheral blood cell images for development of automatic recognition systems”. In: *Data in Brief* 30 (2020), p. 105474. ISSN: 2352-3409. DOI: <https://doi.org/10.1016/j.dib.2020.105474>. URL: <https://www.sciencedirect.com/science/article/pii/S2352340920303681>.
- [9] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. “The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions”. In: *Scientific Data* 5.1 (Aug. 2018). ISSN: 2052-4463. DOI: 10.1038/sdata.2018.161. URL: <http://dx.doi.org/10.1038/sdata.2018.161>.
- [10] Noel C. F. Codella et al. “Skin lesion analysis toward melanoma detection: A challenge at the 2017 International symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC)”. In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE, Apr. 2018, pp. 168–172. DOI: 10.1109/isbi.2018.8363547. URL: <http://dx.doi.org/10.1109/ISBI.2018.8363547>.
- [11] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. 2009.
- [12] Richard Liaw et al. *Tune: A Research Platform for Distributed Model Selection and Training*. 2018. arXiv: 1807.05118 [cs.LG]. URL: <https://arxiv.org/abs/1807.05118>.
- [13] Fernando Martin-Maroto and Gonzalo G. de Polavieja. *Finite Atomized Semilattices*. 2021. arXiv: 2102.08050 [math.RA]. URL: <https://arxiv.org/abs/2102.08050>.