CART451_F2022_Ex1_Deliverable_illiez

PART A:

1. The initial data set consisted of photographs of 11 objects. The objects, selected based on aesthetic characteristics as well as portability, included a mug, a lighter, an apple, a pen, a film projector, a toy cone, a deck of cards, a statuette, a measuring tape, a vial of nail polish and a tin of tobacco. Each object was captured in 50-100 photographs, varying in angle, zoom and lighting. For the purposes of the exercise, two of the objects – the apple and the lighter – were selected as the “distinct objects”, and the photographs of the 9 remaining objects were culled and joined into a single group of other objects (”noise”). The apple was selected for portability and post-exercise reuse, while – presuming the nature of the exercise – the lighter was chosen to probe the limits of the ML classification with a rather small, uniform and nondescript object.

CART451_F2022_Ex1_Deliverable_illiez

2. The purpose of the exercise can putatively be stated at a few levels of analysis
   - Practical: train an ML model to categorize images of objects
   - Methodological: introduce students to embedded machine learning, specifically harnessing a user-friendly online platform
   - Epistemological: push students to reflect critically on ML applications
   - Pedagogical: provide alternative learning structures to copy-pasting
   - Administrative?

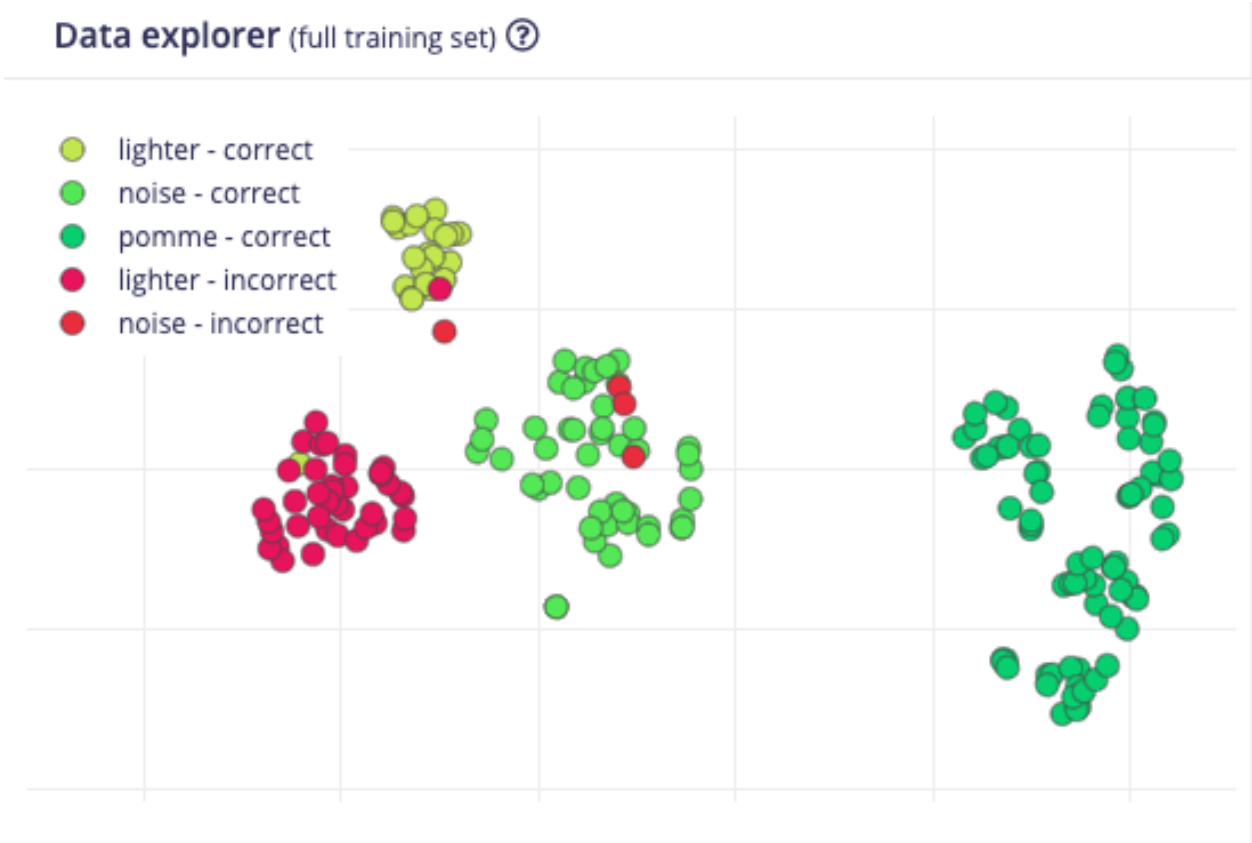
3. Task steps (*italics* → location on Edge Impulse UI)
   - Preprocess data:
     - sort JPEG files
     - file in 3 named directories (2 “objects” & 1 “other”)
   - Import data (*Data acquisition* > *Upload data*): upload data in separate sets (1 per directory), entering appropriate label for sets
   - Explore data (*Data acquisition* > *Data explorer*): generate preliminary data explorer using pretrained visual model & t-SNE dimensionality reduction technique
   - Create impulse (*Impulse Design* > *Create Impulse*): establish impulse blocks:
     - Input: image dimensions (96x96) default for best transfer learning; resize fit to longest axis to avoid truncation
     - Processing: Image processing, with single input axis (photo)
     - Learning: Image Transfer Learning (verify input and output features)
     - Output: verify correspondence between output features and object sets
   - Generate impulse features (*Impulse Design* > *Image*):
     - Select color depth parameters of images, then *Save parameters*
     - *Generate features*, then review feature explorer, processing time & ram usage
   - Train impulse (*Impulse Design* > *Transfer Learning*):
     - Select training settings (most default, *Auto-balance dataset* & *Data augmentation* for training correction/optimization then *Start training*
     - Review model, inferencing time & ram/flash usage
   - Test model (*Model testing* > *Classify all*) then review results
   - Test live (*Deployment* > *Computer*)

4. Accuracy, Precision & Recall:
On the validation set, the model performed with 74.4% accuracy. However, this does not adequately translate the results, in that all the incorrect assessments (10/39) occurred when the model was presented with images of the lighter. All images of the apple and noise were correctly classified. Whether this speaks to the size and uniformity of the correctly identified photographs, or responds to one of the initial hypotheses regarding the nondescript appearance of the lighter – or both – the model's performance at this stage is clearly questionable.

The values for precision and sensitivity in this 3x3 matrix are as follows:

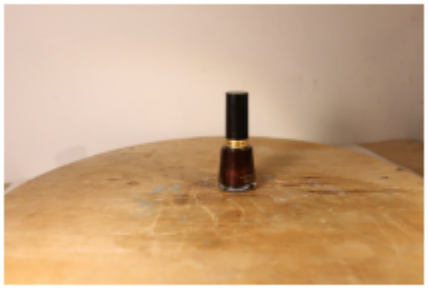
	Precision - TP/(TP+FP)	Sensitivity - TP/(TP+FN)	F-score → 2rp/(r+p)
Lighter	6/(6+0) = 1	6/(6+10) = 0.375	2*1*0.375/(1+0.375) = 0.545
Noise	14/(14+10) = 0.583	14/(14+0) = 1	2*0.583*1/(0.583+1) = 0.737
Pomme	9/(9+0) = 1	9/(9+0) = 1	2*1*1/(1+1) = 1

5.



The training data explorer graph generally comforts the data breakdown, the images being clustered according to the labels, the apple in dark green on the right, the noise in lighter green in the centre, and the lighter images on the left. However, this last element further provides insight into the aforementioned low accuracy of the model in terms of lighter predictions, which is roughly split between two clusters: correctly labeled lighter images in the lightest green on the top of the graph and incorrectly labeled lighter images at the far left. Perplexing, though, is the incorrectly labeled lighter image in the correctly labeled cluster. Can't explain that one..
Even more confusing with respect to the confusion matrix are four incorrectly labeled noise data-points. The images themselves are understandably ambiguous (e.g. IMG_3932, a vial of nail polish not too dissimilar from a lighter – which it was mislabeled as – that had even me fooled for a few seconds..) , but I am unsure why these data points are not reflected in the confusion matrix as decreased sensitivity for noise (or decreased precision for apple and lighter)

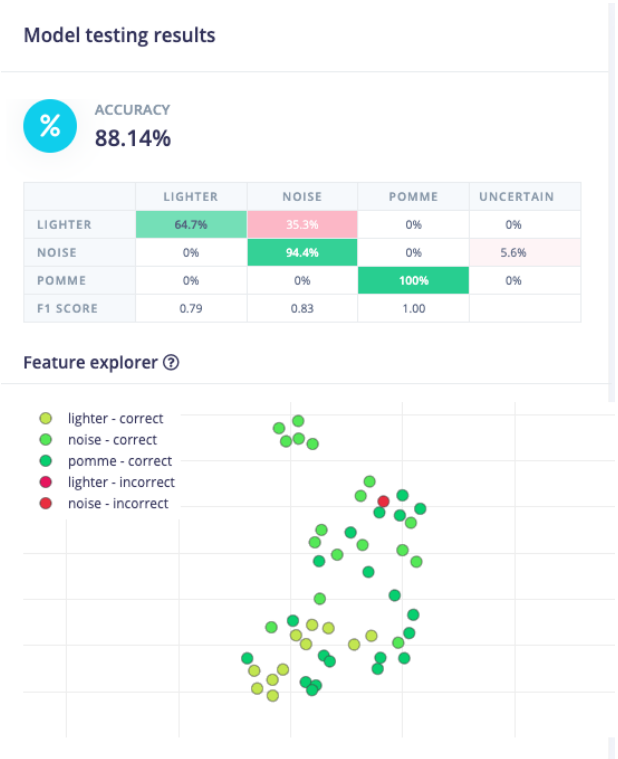
IMG_3932
Label: noise
Predicted: lighter
View sample



CART451_F2022_Ex1_Deliverable_illiez

Testing the model on the algorithmically determined 20% of sample photos initially led to a suspiciously high 97.96% accuracy, following which I decided to add some webcam photos of the lighter, to see whether it had any chance of recognizing it when not in an identical context as the training set. All this did was to add an “unknown” category, leave the confusion matrix unchanged, but increase the accuracy by 0.15%. Confusing. However, when the new samples were labeled as lighter, the accuracy dropped to 88.14%, (possibly) echoing the decrease in model sensitivity for lighter images.

However, interpreting the feature explorer graph, and especially the drastic differences between each iteration with first the addition of unlabeled data then just the modification of image expected outcome, is beyond me.



6. Better performance, in the context of this specific model, might first and foremost refer to a more reasonable spread of predictive values. This could be achieved by a broader aesthetic representation of the objects selected (different backgrounds, more lighting changes, etc..), without which the model appears to be overfitting. It is for example incapable of detecting an object like the lighter, even alone in the scene, in a different setting. The uniformity of the background could also contribute to mislabeling between the objects.
With regards to the nondescriptivity of the lighter, it might also be constructive to add a greater range of objects that are more similar to it (e.g. the nail polish vial) to provide the model with more fine-grained distinction in lighter detection.

PART C : My first thought in terms of Object Detection integration hovered around an automated Waldo finder, until a Google search revealed that this Has Already Been Done™ (elegantly so)
The second idea, likely engendered by the state of my stomach and fridge, coalesced around a pantry content-based recipe crawler. The general idea would be an Object detection ML model embedded in a microcontroller with a camera. When prompted, the system would gather photographs of the fridge contents, assess which food items are present and output a list of ingredients and quantities. This would then be matched with entries in a database of recipes (or even live crawling? If the sky is the limit..), to algorithmically determine which dishes could be made with the present ingredients.
I would anticipate difficulties in detecting similar vegetables or standard liquid/broth/sauce containers that would require fine-tuned training, but seems like a fundable project to me..
Ideally, there would also be some fashion of appearance/colour-based sorting, to classify perishable fridge contents according to age/time spent untouched.

Behold, the storyboard: