ARN - Laboratory 05

Anthony Coke, Mehdi Salhi, Guilain Mbayo

June 16, 2022

Contents

Introduction

Our application is made in the context of the course ARN (Apprentissage par réseau de neurone) At the HEIG-VD. Its goal is to classify different types of nuts. In order to achieve this goal, we decided to take several pictures of cashew nuts, hazelnuts, and pecans with various backgrounds, various specimens and various angles. Depending on the result, we will then eventually add pictures from the web.

We will use MobileNet V2, which is a pre-trained convolutionnal network. We will then add some layers and train them to our specific application.

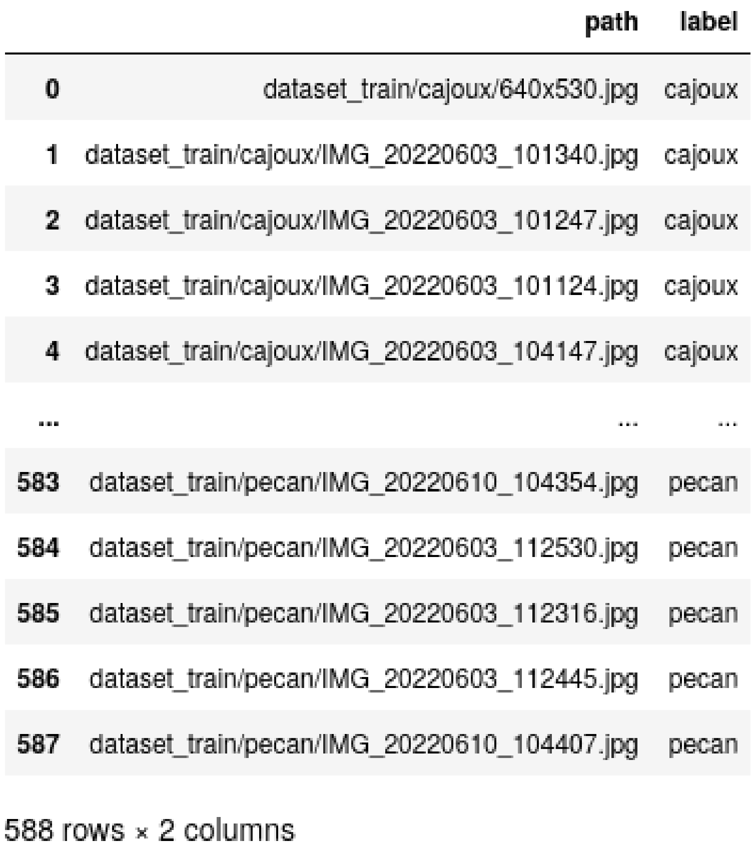
Our application could be used for various purpose. For example, allow users to identify different nuts, for allergy purpose. This could also be used for industrial application when sorting nuts or detecting allergens in food or even better, an app on social media that tells you what sort of nut you are !

The problem

The problem is to be able to identify and differentiate between 3 sorts of nuts: pecan, hazelnuts and cashews, using a camera on a portable device.

Dataset and Data preparation

We took photos of the different nuts on various background (color, texture) with various angles and zoom without our smartphones. Our dataset consist of 587 images of pecans, cashews and hazlenuts. We were careful to take pictures that would represent the final condition best, that is, trying to identify a nut with a mobile device.



	path	label
0	dataset_train/cajoux/640x530.jpg	cajoux
1	dataset_train/cajoux/IMG_20220603_101340.jpg	cajoux
2	dataset_train/cajoux/IMG_20220603_101247.jpg	cajoux
3	dataset_train/cajoux/IMG_20220603_101124.jpg	cajoux
4	dataset_train/cajoux/IMG_20220603_104147.jpg	cajoux
...	...	...
583	dataset_train/pecan/IMG_20220610_104354.jpg	pecan
584	dataset_train/pecan/IMG_20220603_112530.jpg	pecan
585	dataset_train/pecan/IMG_20220603_112316.jpg	pecan
586	dataset_train/pecan/IMG_20220603_112445.jpg	pecan
587	dataset_train/pecan/IMG_20220610_104407.jpg	pecan

588 rows × 2 columns

Figure 1: Dataset Image List

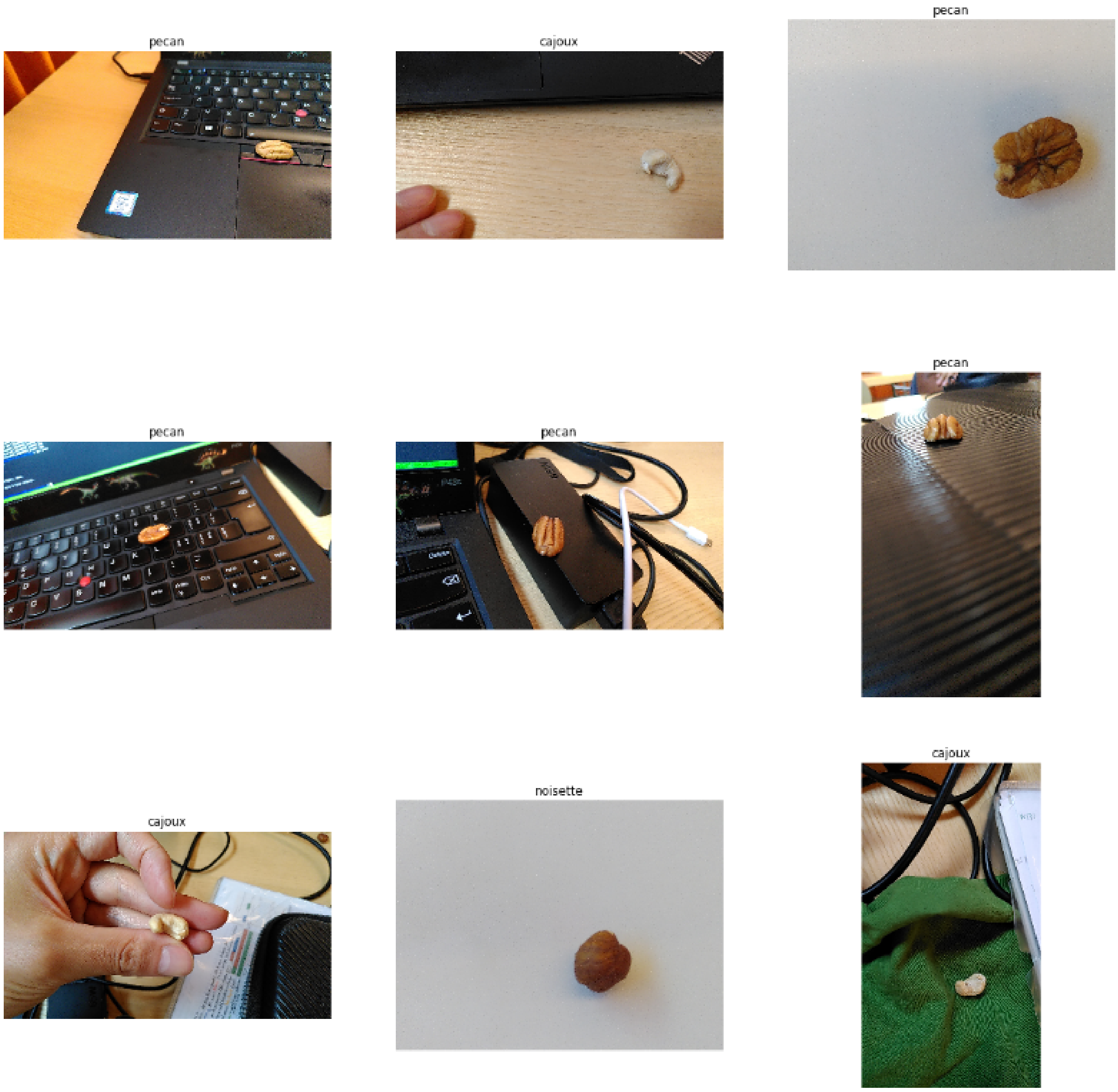


Figure 2: Sample of our dataset

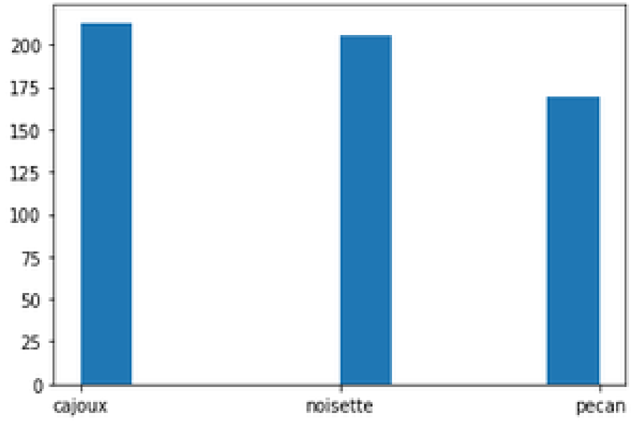


Figure 3: Dataset Histogram

We had to take more and more pictures because our first dataset had too much of the same background. Our final dataset consist of close-up pictures of the nuts with a “neutral” background so that our model can extract clear features plus more realistic picture taken at various angles and backgrounds so that it does not overfit and can work better in realistic conditions.

##TODO: data augmentation, crop, zoom, format, … Our whole images dataset is rescaled and resized in order to always give the same dimensions as input for our model. In order to have slightly different images at each iteration, we applied some data augmentations to our training set. It include RandomFlip, RandomZoom, RandomRotation and RandomContrast. We chose to use value between -0.2 and 0.3 and not higher because we saw that the resultant pictures were too deformed, and thus could maybe mislead our model.

Model creation

We tried a lot of different variation. We started with very simple architectures like a single layer of 20 neurons, 2 layers, 3 layers. We also tried 250 neurons, adding dropout, modifying out data augmentation but our model was still bad in realistic condition. With the previous labs, we were used to have very simple architectures and a small number of neurons but this task is much more complex. We searched the web for examples and common settings for image classification and found out that using thousands of neurons was more common for such a task. We tried adding 2 layers with 1024 neurons each and a third layer with 512 neurons. The graph of the results wasn’t too good as you can see below, but our confusion seemed to indicate that the classification was ok.

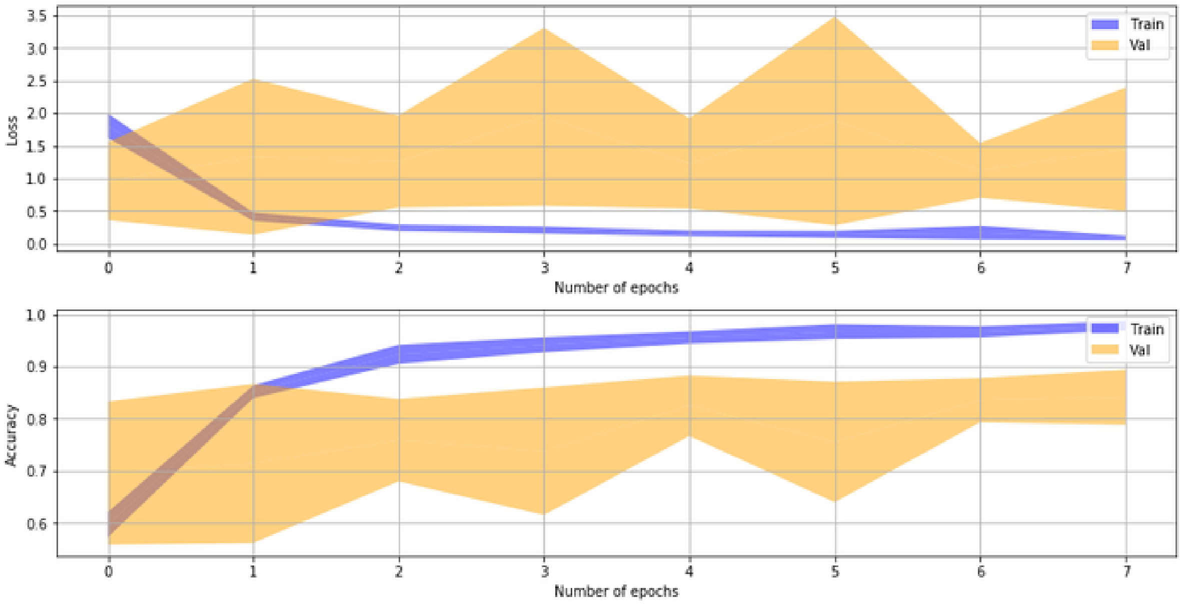


Figure 4: Result Graph

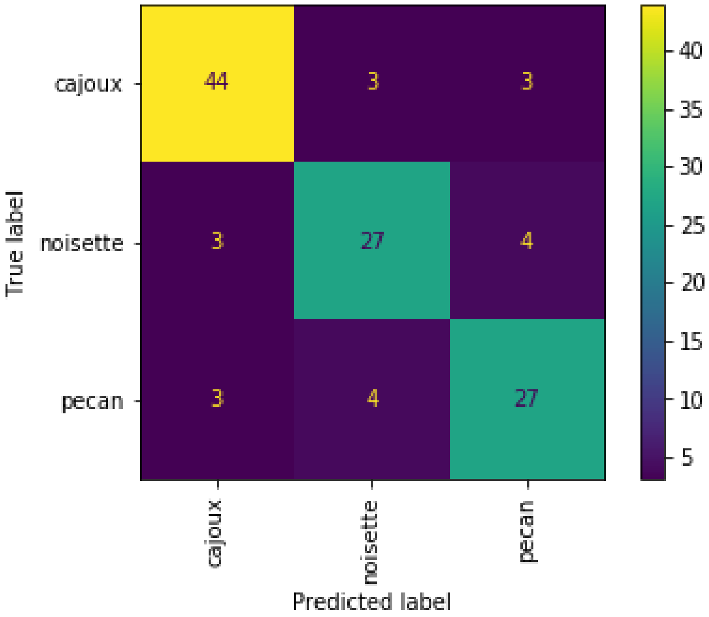


Figure 5: ConfusionMatrix

Our grad-cam also indicated that the model was able to find the nuts and was activated by them.

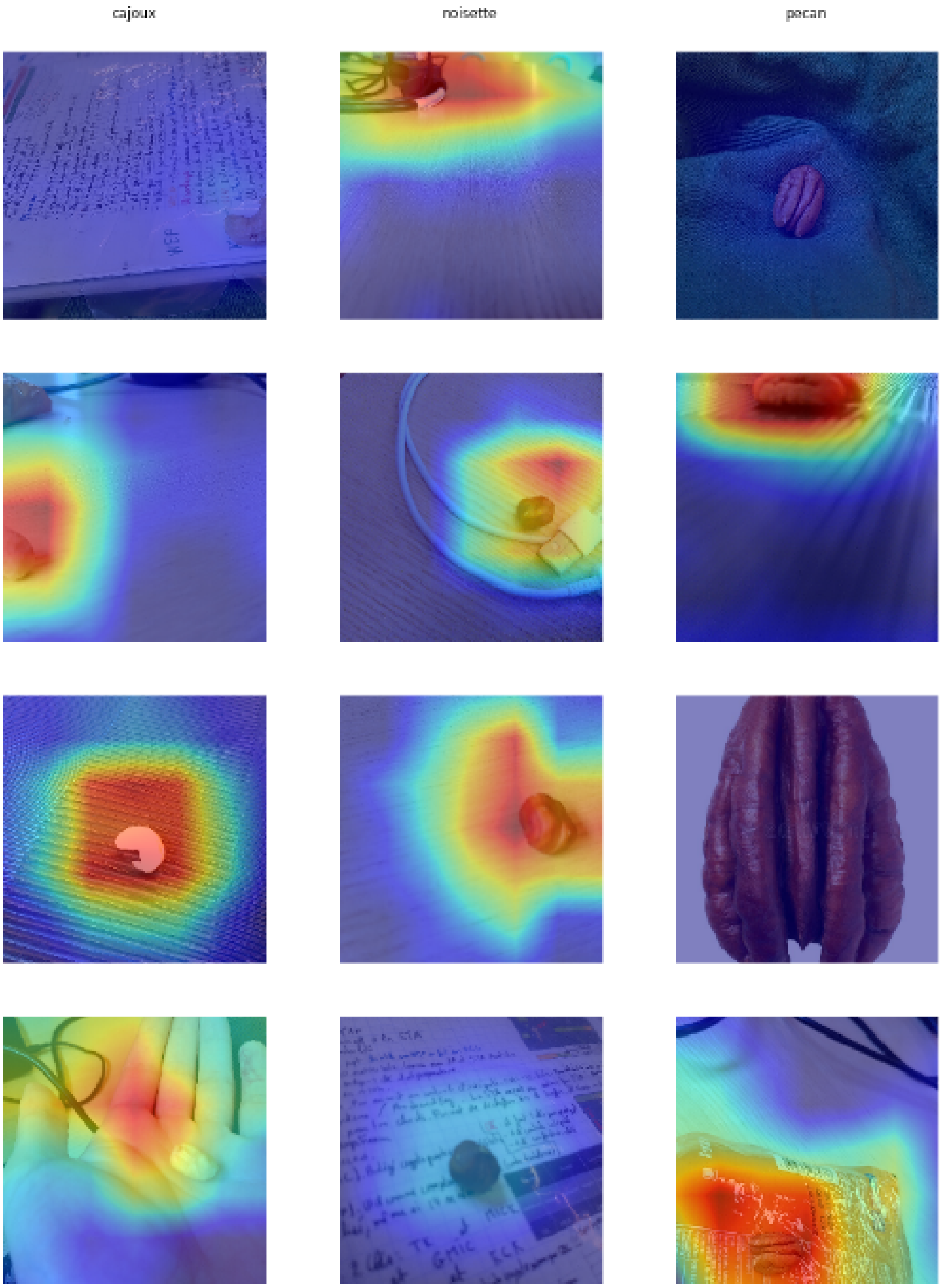
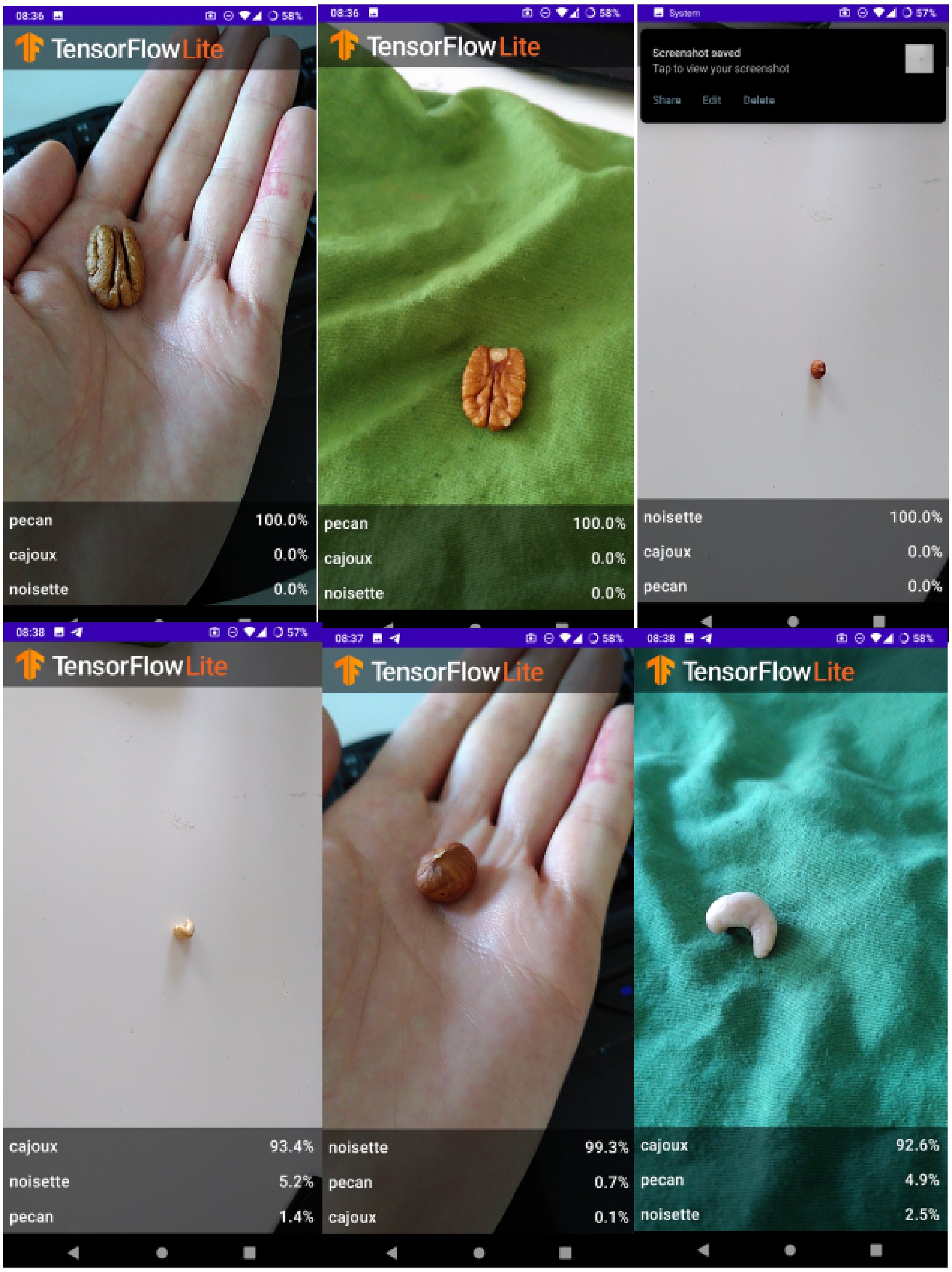


Figure 6: Grad Cam

So we loaded our model on our phones and tried in realistic conditions. It was the first time the results were good using our phones.



```
pecan      100.0%
cajoux       0.0%
noisette     0.0%
```

```
pecan      100.0%
cajoux       0.0%
noisette     0.0%
```

Screenshot saved
Tap to view your screenshot
Share Edit Delete

```
noisette   100.0%
cajoux       0.0%
pecan        0.0%
```

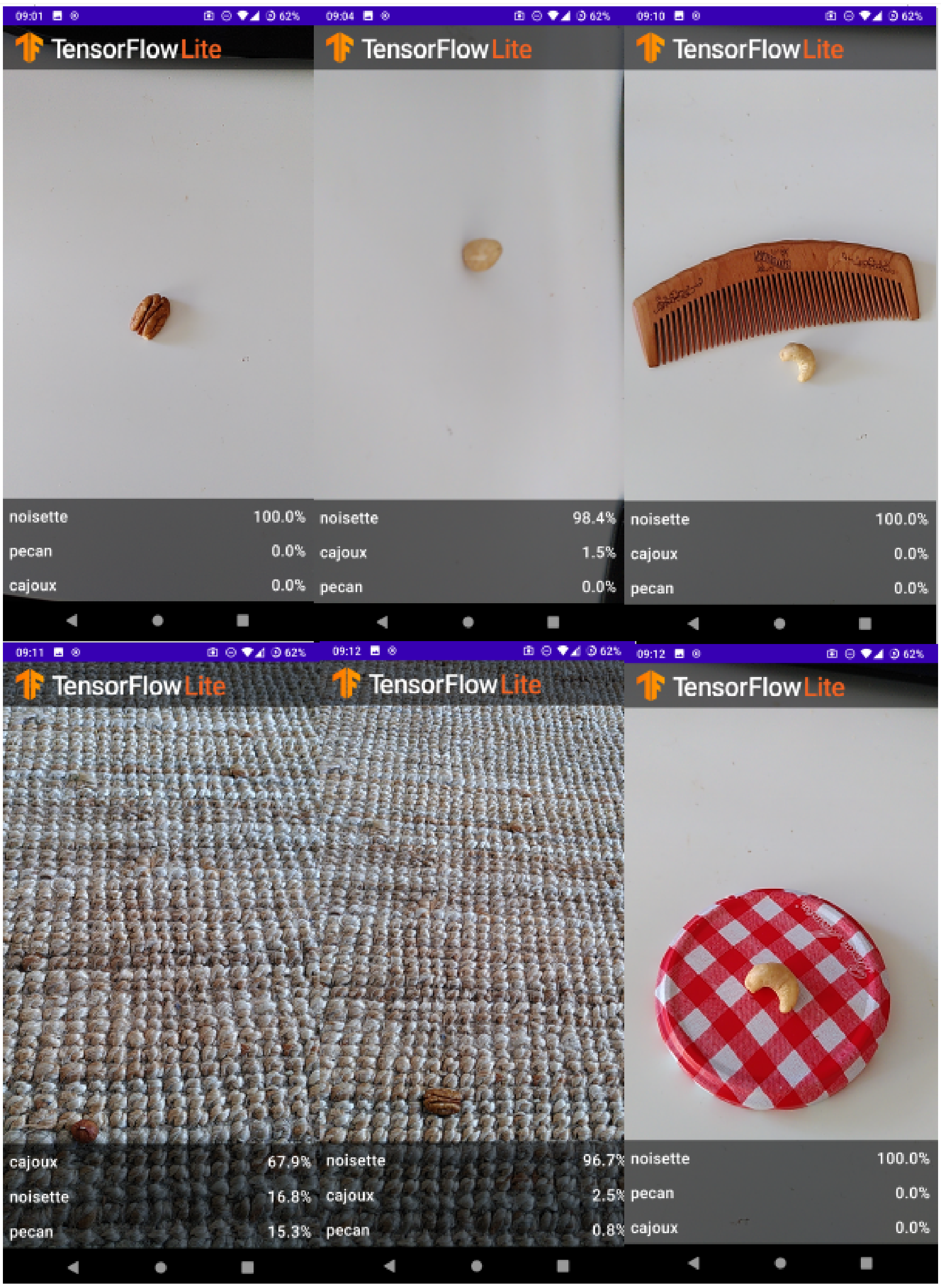
```
cajoux      93.4%
noisette     5.2%
pecan        1.4%
```

```
noisette    99.3%
pecan        0.7%
cajoux       0.1%
```

```
cajoux      92.6%
pecan        4.9%
noisette     2.5%
```

Figure 7: PhoneResults Good classification

Still, our model is very easy to trick by changing the angle or the background.



noisette 100.0%
pecan 0.0%
cajoux 0.0%

noisette 98.4%
cajoux 1.5%
pecan 0.0%

noisette 100.0%
cajoux 0.0%
pecan 0.0%

cajoux 67.9%
noisette 16.8%
pecan 15.3%

noisette 96.7%
cajoux 2.5%
pecan 0.8%

noisette 100.0%
pecan 0.0%
cajoux 0.0%

Figure 8: PhoneResults Bad classification

Parameters

Final model :

```
Total params: 5,145,667
Trainable params: 3,299,843
Non-trainable params: 1,845,824
```

Our final model consist of the frozen MobileNetV2 layers plus the 4th last layers we added :

- Dense(1024, activation="relu")
- Dense(1024, activation="relu")
- Dense(512, activation="relu"),
- Dense(len(LABEL_NAMES), activation="softmax")

Results

##TODO: screenshots, confusion matrix, etc

Conclusion

We tried several configurations for our model (1 layer, 2 layers, 3 layers, 256 neurons, 8 neurons, dropout, etc…) and most of the time, we managed to get high accuracies and fine confusion matrix. But even so, we had problems once our model loaded on the app. Indeed, the app detected each classes with great confidence, but not for the right nut. We believe that our model is too influenced by the background due to a to different images set between our 3 classes. TODO: ARN IS FUN