Step 1 : Extract Transform Load :

- For this step I decided to check on the kaggle platform a subject that interested me related to image processing to apply some deep learning.

- I chose this kaggle competition and downloaded their data for the Capstone project :
  - https://www.kaggle.com/moltean/fruits

- I loaded them thanks to keras library and the function image.load_img : Thans to this library, I constructed two functions to load and transform to numpy array those images : load_image(path) for the non DL-algo and load_image_keras for the DL algorithm.

- Those function are used on the "Load Data" part after we get all the path for the training and test dataset.

- Finally on the "Data Cleansing Part", I created y_train and y_test and two dictionnaries to go from the class number to the name of the fruit and vice-versa.

- I permuted the train data and test data to have the class in different order for CNN to train better.

1rst Model : PCA + Random Forest:

Step 2 : Feature Creation

- I flattened the images loaded in a numpy array. I then scalled the train set and applied the same transform to the test dataset.

Step 3 : Model Definition

- I applied a PCA algorithm to guard 90% of the variance of the data set for my next algorithm.
- I use a Random Forest Classifier with 20 estimators.

Step 4: Model Training

- I trained on the permuted Train dataset with a simple .fit()

Step 5: Model Evaluation

- To evaluate, I use a simple accuracy score, but I also checked the Precision, Recall and F1 score for all the classes. (Got them from the function pred_accuracy_each_label(pred, true))
- The function top5_worst give the 5 worst Precision, Recall and F1_score. We could check why it is bad and change our feature extraction for those classes.

2nd Model : Deep Learning :

Step 2 : Feature Creation

- In the function load_image_keras, I added a call to the function keras.applications.vgg16.preprocess_input to transform the raw image to a image well adapted for the CNN VGG16.

- I used the VGG16 network top output (that is just after the last Conv layer) as feature. So, I got for the train and test dataset the features.

Step 3 : Model Definition

- I used a DL network with : 1 Conv 2D layer, 1 MaxPooling2D, 4 Flattens with two Dropouts layers (0.3)

Step 4: Model Training

- As it is a multiclass problem, I choose a categorical cross entropy loss with an Adam optimizer. I fitted on 10 epochs

Step 5: Model Evaluation

- To evaluate, I use a simple accuracy score, but I also checked the Precision, Recall and F1 score for all the classes. (Got them from the function pred_accuracy_each_label(pred, true))
- The function top5_worst give the 5 worst Precision, Recall and F1_score. We could check why it is bad and change our feature extraction for those classes.

Results :

1rst Model :

```
Accuracy with RandomForest:  88.17035584197254

*******************************************
5 worst Precision :
    -  Nectarine : [0.436, 0.354, 0.391]
    -  Apple Golden 3 : [0.549, 0.938, 0.693]
    -  Apple Red 2 : [0.624, 0.707, 0.663]
    -  Apple Golden 1 : [0.64, 0.671, 0.655]
    -  Banana : [0.656, 0.747, 0.699]
*******************************************
5 worst Recall :
    -  Nectarine : [0.436, 0.354, 0.391]
    -  Plum : [0.976, 0.543, 0.698]
    -  Apple Red 1 : [0.691, 0.573, 0.626]
    -  Banana Red : [0.828, 0.578, 0.681]
    -  Apple Red 3 : [0.754, 0.618, 0.679]
*******************************************
5 worst F1_score :
    -  Nectarine : [0.436, 0.354, 0.391]
    -  Apple Red 1 : [0.691, 0.573, 0.626]
    -  Apple Golden 1 : [0.64, 0.671, 0.655]
    -  Apple Red 2 : [0.624, 0.707, 0.663]
    -  Apple Red 3 : [0.754, 0.618, 0.679]
*******************************************
```

2nd Model :

```
Accuracy with DeepLearning:  96.21182404034744
```

```
*****************************************
5 worst Precision :
    -  Pineapple Mini : [0, 0, 0]
    -  Kiwi : [0.51, 1.0, 0.675]
    -  Rambutan : [0.529, 1.0, 0.692]
    -  Chestnut : [0.614, 1.0, 0.761]
    -  Nectarine : [0.841, 1.0, 0.914]
*****************************************
5 worst Recall :
    -  Pineapple Mini : [0, 0, 0]
    -  Cocos : [1.0, 0.024, 0.047]
    -  Peach : [0.914, 0.713, 0.801]
    -  Apple Red 1 : [0.983, 0.726, 0.835]
    -  Mangostan : [0.942, 0.794, 0.862]
*****************************************
5 worst F1_score :
    -  Pineapple Mini : [0, 0, 0]
    -  Cocos : [1.0, 0.024, 0.047]
    -  Kiwi : [0.51, 1.0, 0.675]
    -  Rambutan : [0.529, 1.0, 0.692]
    -  Chestnut : [0.614, 1.0, 0.761]
*****************************************
```

We see that Pineapple Mini is always confounded by Rambutan, it is because the VGG 16 misclassify him. We should also train the VGG if we want better results. We also have the issue with Coco and Kiwis and Peach with Pomgrenate.