Mobile App Development
CS4153/CS5143 - Online
Grad Assignment

COREML

According to the article by **IBM Cloud Education**, "**Machine learning** is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy." It is a fast-growing field that seems to be involved in almost every academic and industry discussion in recent times.

COREML is apple's powerhouse for performing machine learning tasks. According to the documentation on developer.apple.com, CoreML applies a machine learning algorithm to create a model which can be used to make predictions on data never seen before by your model.

CoreML is used in a variety of tasks from image classification to object detection, and a whole other machine learning task.

After model creation which are always of mlmodel files, the mlmodel files can be integrated in our project and applications. And whenever the user supply fresh data, the app uses the CoreML APIs with the given data to make predictions.

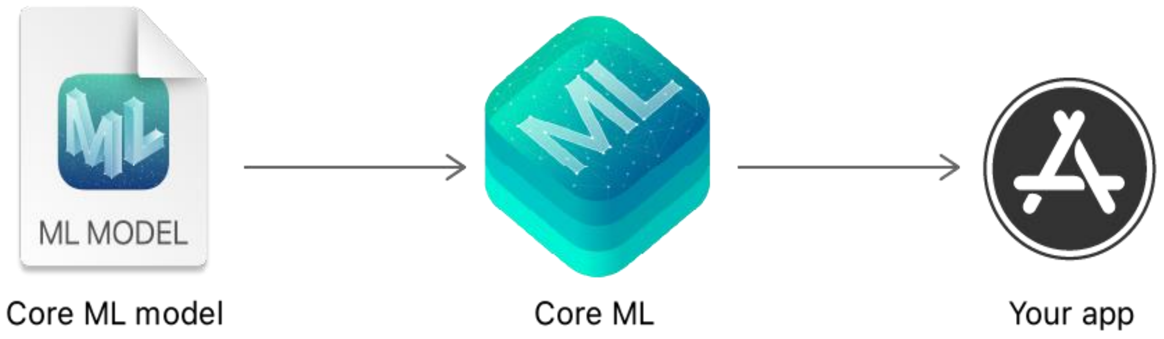


Figure 1: Integrating a coreML mlmodel into your app allows the mlmodel make API calls and run the task for your app. (Image obtained from **https://developer.apple.com/documentation/coreml**)

Core ML is built upon low level primitives called Accelerated and BNS, and Metal Performance Shaders. Several Machine learning tasks are then built on top of CoreML which then let our app take advantage of the various tasks. See below and image from **https://developer.apple.com/documentation/coreml**

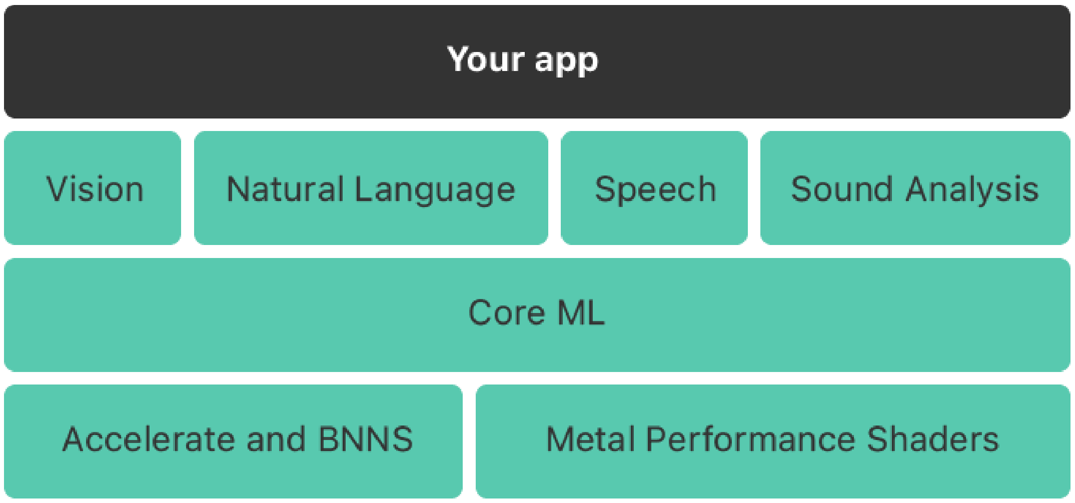


Figure 2: An overview of hierarchical relationship in the COREML framework between our app and COREML.

CreateML

CreateML is another part of the swift ecosystem that does the actual training part of this project. Instead of training the model outside Swift ecosystem and then converting the model into mlmodel format, CreateML ensures that you can train your model end-to-end and get an mlmodel directly.

How Does it Work?

We train a model to recognize patterns in our data be it image, text or sound etc. In this case our image data are put in different folders according to the classes. This is done for both training and testing data. The model is trained on the data and evaluated. Then it is tested on the test set to see how it performs on unseen data.

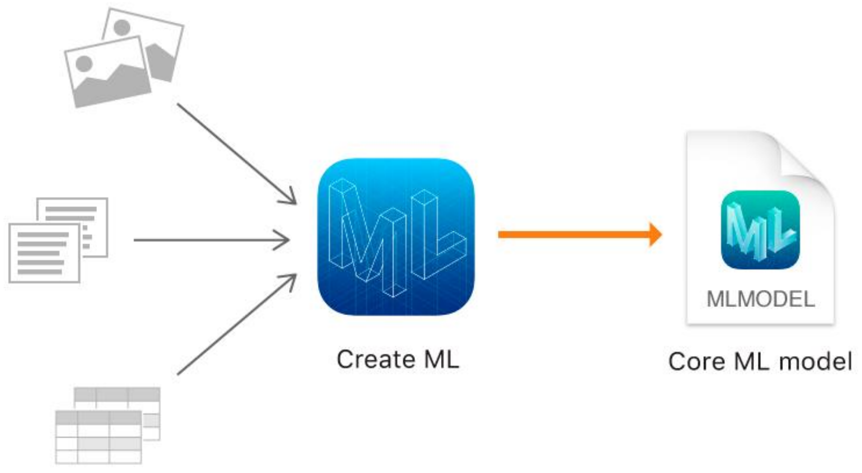


Figure 3: We supply data to CreateML and we receive a prediction model in return.

A lot of time we may have to go back to the drawing board to collect more or better dataset, etc. and so Machine learning task may be described as loop. Collect Data, Train, Evaluate, repeat.

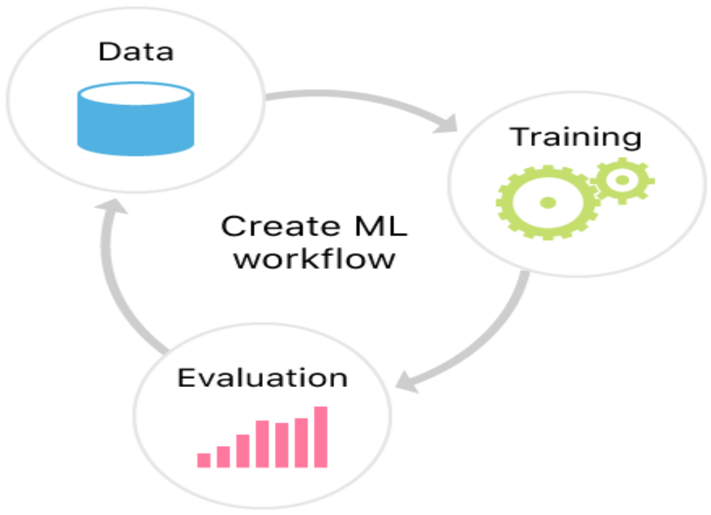


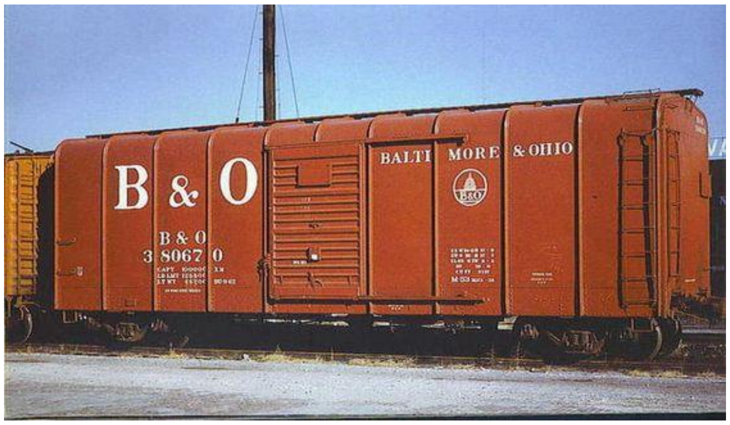
Figure 4: The ML Loop

BoxCars Rust Prediction

For our own project we have harnessed the power of COREML and CREATEML to build an app which can predict whether a Boxcar is

1. Good
2. Rusty
3. Damaged

All images were obtained using the google image search engine **https://images.google.com/**



After we trained our model in the CREATEML environment, the mlmodel is then downloaded and placed in the project navigator.

Once our model is in the navigator, we can create a class of it. Say our model is BoxCarsClassification.mlmodel, we can create an instance of the model as follows

```
let model = BoxCarsClassification()
```

This creates an instance of the model in our swift ios project, and from there we can use the instance to predict our data.