

**REPORT ON  
RAILROAD BOXCAR COREML**

**By**  
Singamsetty Bhavitha

# Table of Contents

Introduction.....	3-4
Create ML.....	5-8
Integration of CoreML with IOS app.....	9
➤ Create an Image Classifier Instance .....	9
➤ Create an Image Classification request.....	10
➤ Create a Request Handler .....	10
Conclusion.....	11
References.....	12

# Introduction

Use Core ML to integrate machine learning models into your app. Core ML provides a unified representation for all models. Your app uses Core ML APIs and user data to make predictions, and to train or fine-tune models, all on a person's device.



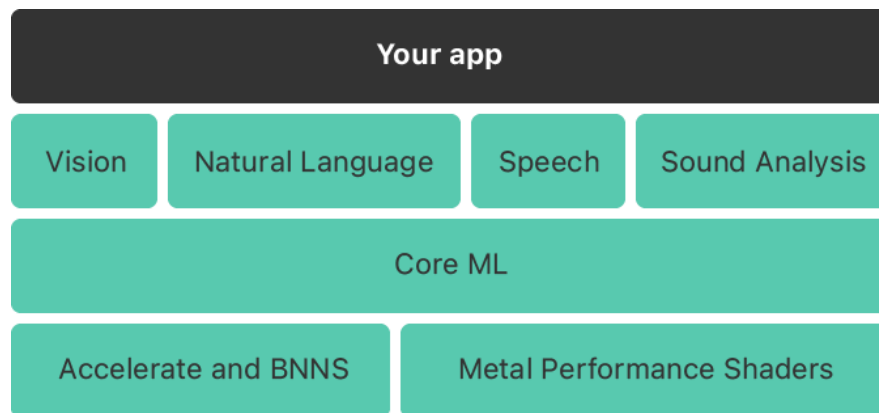
A model is the result of applying a machine learning algorithm to a set of training data. You use a model to make predictions based on new input data. Models can accomplish a wide variety of tasks that would be difficult or impractical to write in code. For example, you can train a model to categorize photos, or detect specific objects within a photo directly from its pixels.

You build and train a model with the Create ML app bundled with Xcode. Models trained using Create ML are in the Core ML model format and are ready to use in your app. Alternatively, you can use a wide variety of other machine learning libraries and then use Core ML Tools to convert the model into the Core ML format. Once a model is on a person's device, you can use Core ML to retrain or fine-tune it on-device, with that person's data.

Core ML optimizes on-device performance by leveraging the CPU, GPU, and Neural Engine while minimizing its memory footprint and power consumption. Running a model strictly on a person's device removes any need for a network connection, which helps keep a person's data private and your app responsive.

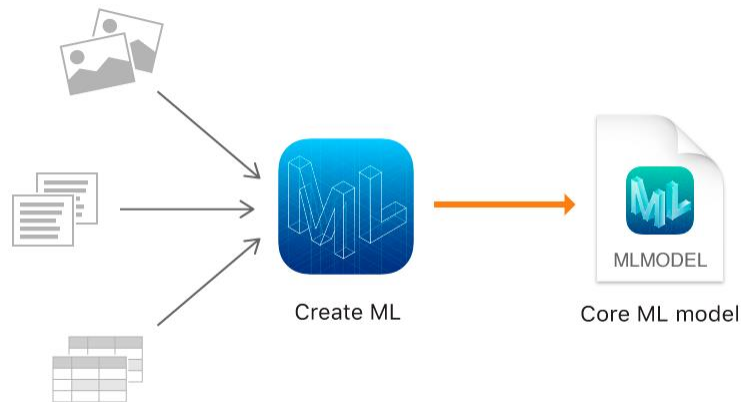
The framework is the foundation for domain-specific frameworks and functionality. It supports Vision for analyzing images, Natural Language for processing text, Speech for converting audio to text, and Sound Analysis for

identifying sounds in audio. Core ML itself builds on top of low-level primitives like Accelerate and BNNS, as well as Metal Performance Shaders.



# Create ML

Use Create ML with familiar tools like Swift and macOS playgrounds to create and train custom machine learning models on your Mac. You can train models to perform tasks like recognizing images, extracting meaning from text, or finding relationships between numerical values.



You train a model to recognize patterns by showing it representative samples. For example, you can train a model to recognize dogs by showing it lots of images of different dogs. After you've trained the model, you test it out on data it hasn't seen before, and evaluate how well it performed the task. When the model is performing well enough, you're ready to integrate it into your app using [Core ML](#).



Create ML leverages the machine learning infrastructure built in to Apple products like Photos and Siri. This means your image classification and natural language models are smaller and take much less time to train.

# Integrating of CoreML model with ios app

Integrating Apple's Vision framework with a Core ML model enhances your application by facilitating image and video analysis. Vision helps to prepare and handle image data before it's processed by a Core ML model, streamlining tasks such as image classification, object detection, or even more complex image-based predictions.

## Create an Image Classifier Instance:

At launch, the ImagePredictor class creates an image classifier singleton by calling its createImageClassifier() type method.

```
/// - Tag: name
static func createImageClassifier() -> VNCoreMLModel {
    // Use a default model configuration.
    let defaultConfig = MLModelConfiguration()

    // Create an instance of the image classifier's wrapper class.
    let imageClassifierWrapper = try? MobileNet(configuration: defaultConfig)

    guard let imageClassifier = imageClassifierWrapper else {
        fatalError("App failed to create an image classifier model instance.")
    }

    // Get the underlying model instance.
    let imageClassifierModel = imageClassifier.model

    // Create a Vision instance using the image classifier's model instance.
    guard let imageClassifierVisionModel = try? VNCoreMLModel(for: imageClassifierModel) else {
        fatalError("App failed to create a `VNCoreMLModel` instance.")
    }

    return imageClassifierVisionModel
}
```

## Create an Image Classification request:

The Image Predictor class creates an image classification request — a `VNCoreMLRequest` instance — by passing the shared image classifier model instance and a request handler to its initializer.

```
// Create an image classification request with an image classifier model.  
  
let imageClassificationRequest = VNCoreMLRequest(model: ImagePredictor.imageClassifier,  
                                                completionHandler: visionRequestHandler)  
  
imageClassificationRequest.imageCropAndScaleOption = .centerCrop
```

## Create an Request Handler:

The Image Predictor's `makePredictions(for photo, ...)` method creates a `VNImageRequestHandler` for each image by passing the image and its orientation to the initializer.

```
let handler = VNImageRequestHandler(cgImage: photoImage, orientation: orientation)
```



## **CONCLUSION**

Thus, we have demonstrated the process of developing a straightforward custom image classifier app for iOS using CoreML. By employing the CreateML app within Xcode, we can generate a machine learning model by supplying training, testing, and configuration data. After creating the model, we incorporated it into the iOS app by integrating it with the Vision framework and executing the Vision request. The results of the classification were then managed and displayed on the front end, showcasing various degrees of rusting.

## REFERENCES

[https://developer.apple.com/documentation/coreml/getting\\_a\\_core\\_ml\\_model](https://developer.apple.com/documentation/coreml/getting_a_core_ml_model)

<https://www.youtube.com/watch?v=3MXYwpifQOM&t=219s>

[https://www.youtube.com/watch?v=6\\_k0kXhRajo&t=853s](https://www.youtube.com/watch?v=6_k0kXhRajo&t=853s)