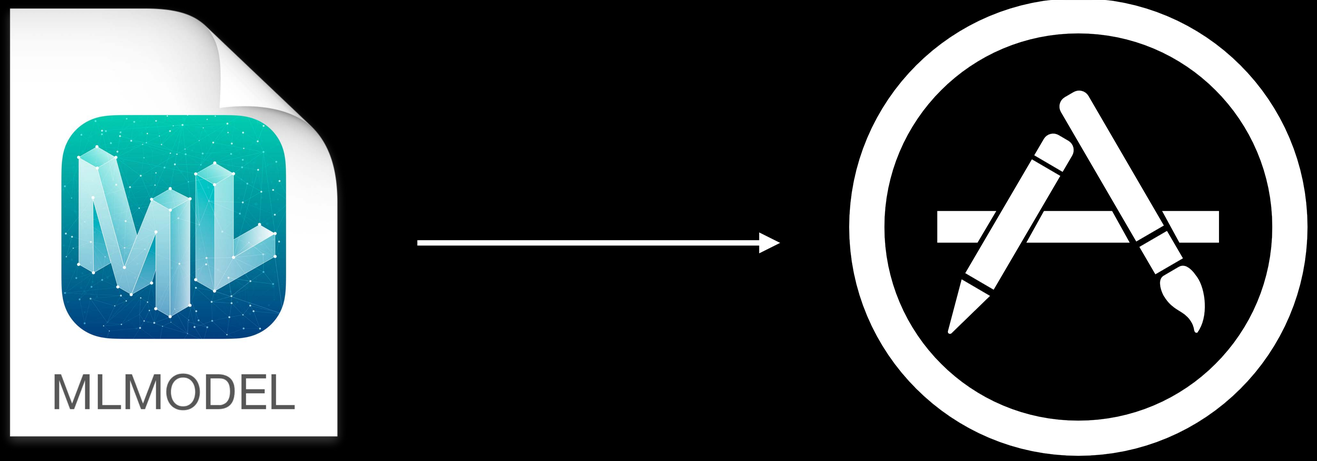
#WWDC18

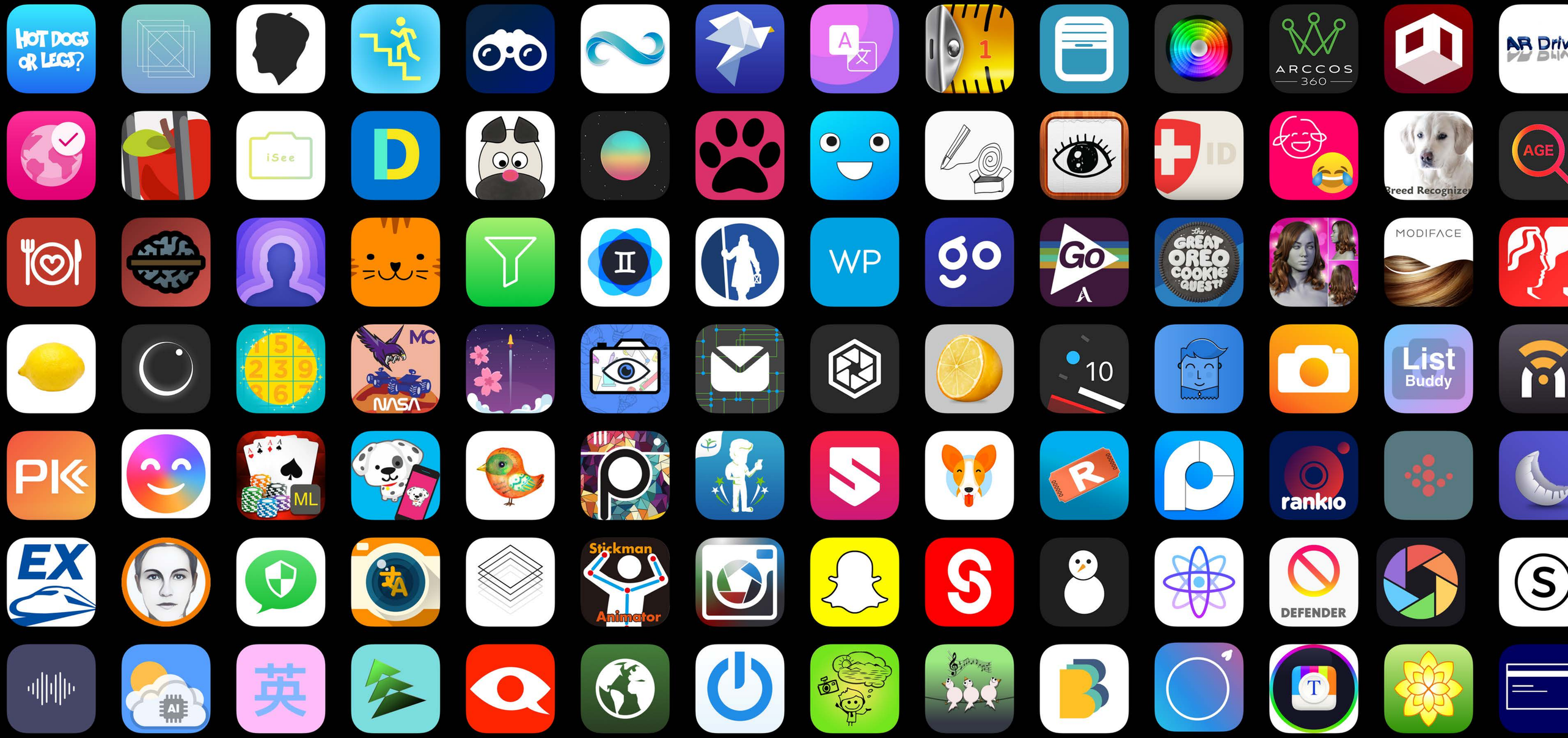
What's New in Core ML

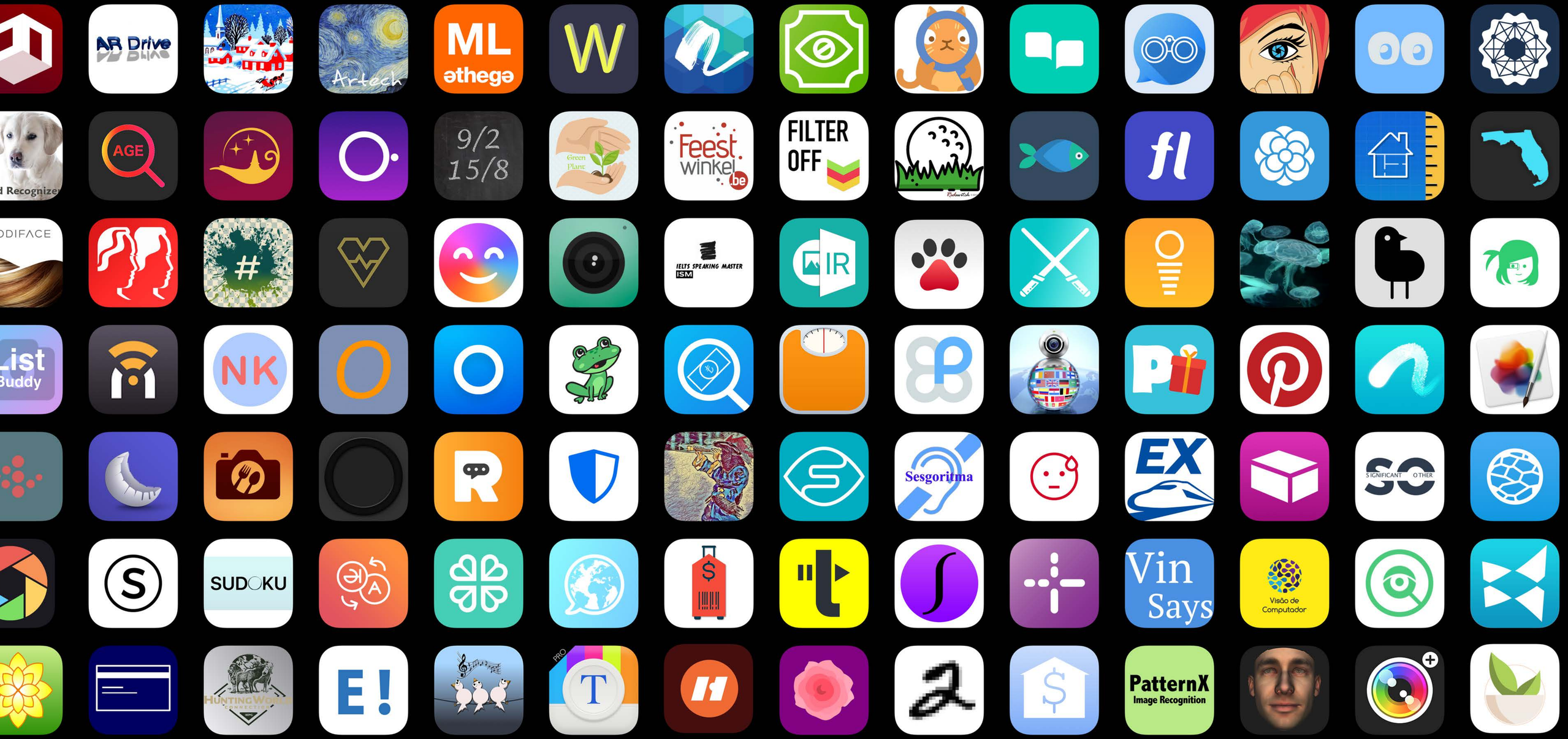
Part two

Aseem Wadhwa, Core ML
Sohaib Qureshi, Core ML

Core ML




HOT DOGS OR LEGS?
ARCCOS 360
iSee
Breed Recognizer
AGE
WP
Go
GREAT OREO COOKIE QUEST!
MODIFACE
List Buddy
NASA
MC
ML
rankio
DEFENDER
Stickman Animator
AI
英

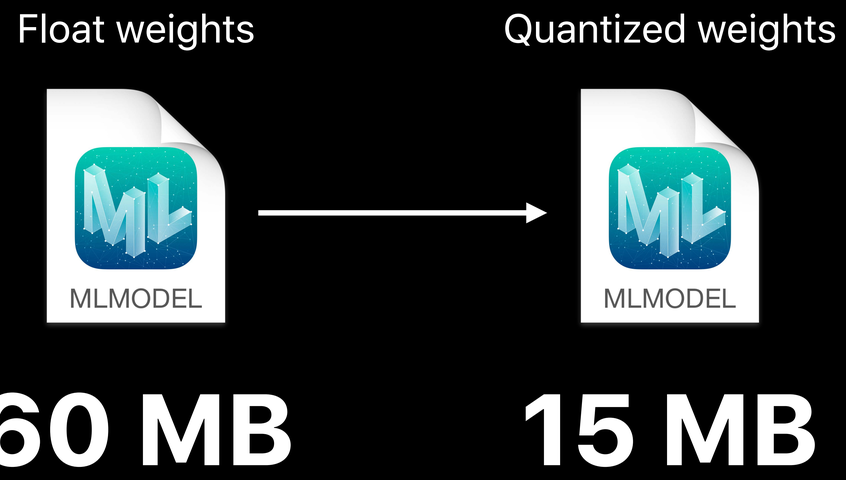

AR Drive
Artech
ML
athega
W
AGE
9/2
15/8
Green
Plant
Feest
winkel
be
FILTER
OFF
fl
IELTS SPEAKING MASTER
ISM
IR
NK
Sesgoritma
EX
SIGNIFICANT
OTHER
SUDOKU
Vin
Says
Visão de
Computador
HUNTING WORLD
E!
PRO
PatternX
Image Recognition

Part One Recap

Model size

Performance

Customization

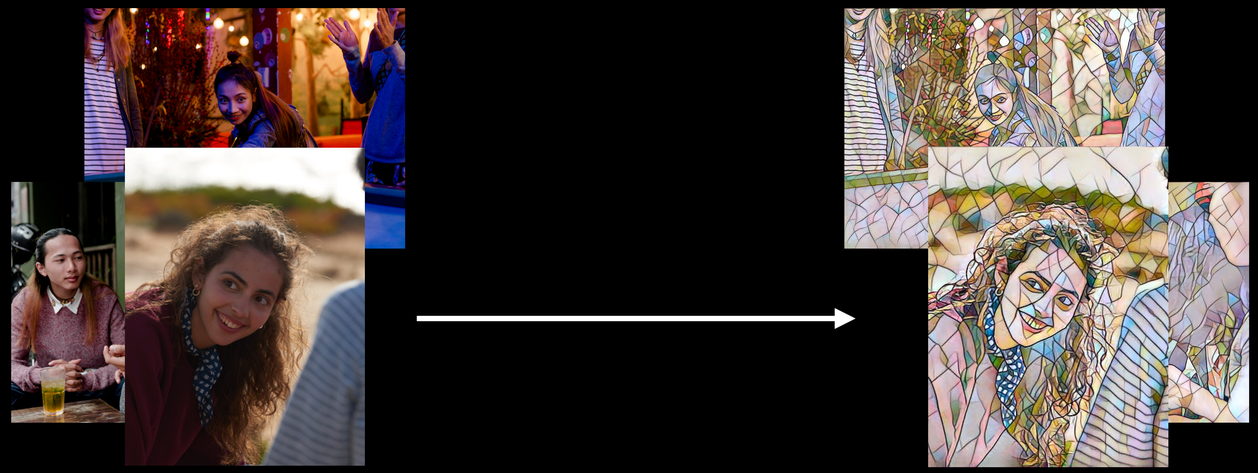


Part One Recap

Model size

Performance

Customization

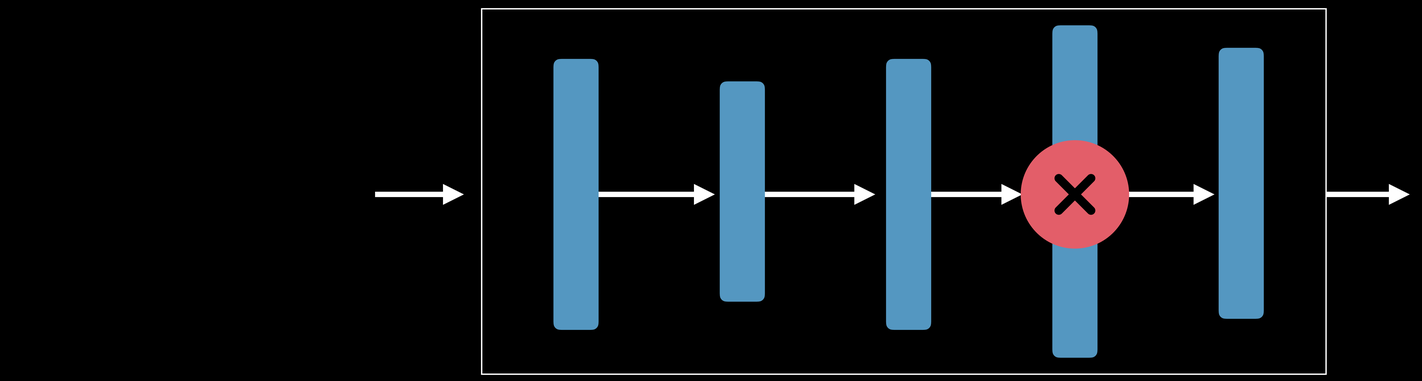


Part One Recap

Model size

Performance

Customization



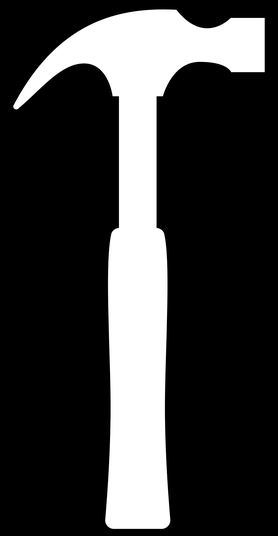
Part One Recap

Model size

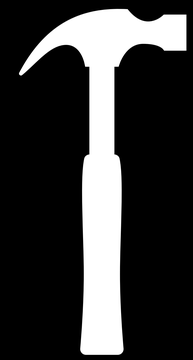
Performance

Customization

Part Two

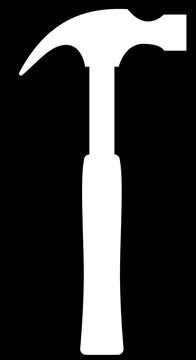


Agenda



Agenda

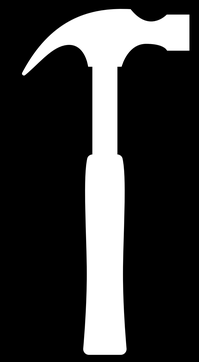
Core ML Tools ecosystem



Agenda

Core ML Tools ecosystem

Quantization utilities

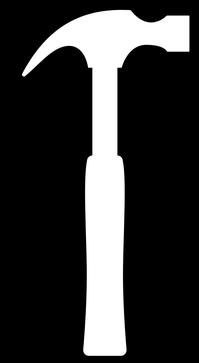


Agenda

Core ML Tools ecosystem

Quantization utilities

Custom conversion



Core ML Tools ecosystem

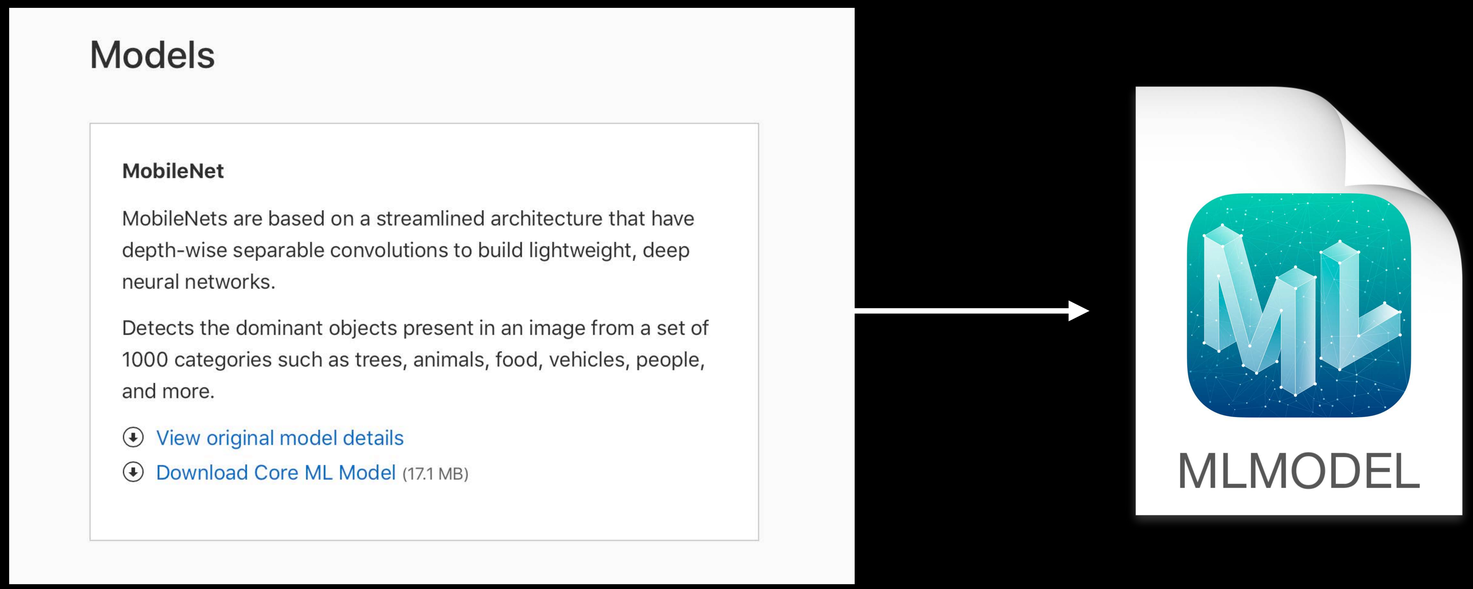
Quantization utilities

Custom conversion

?

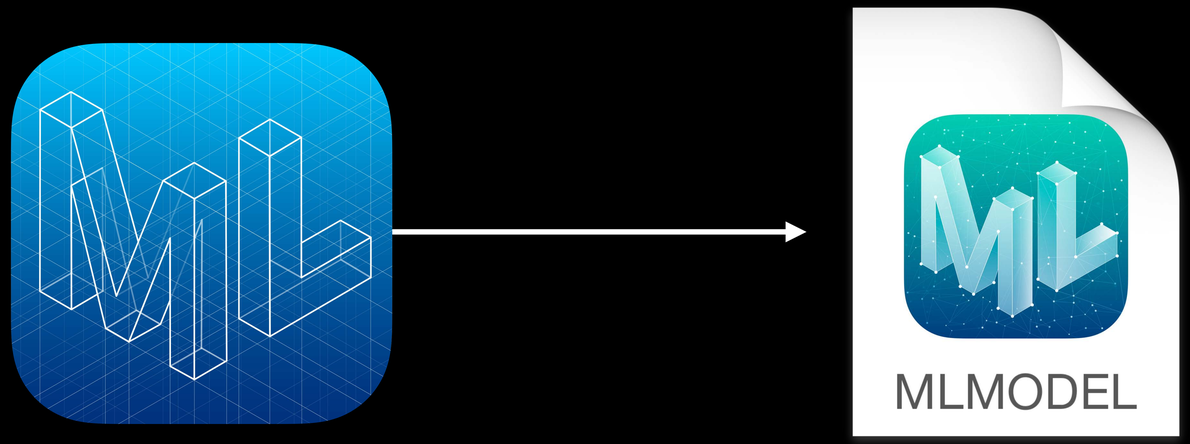

MLMODEL

Download



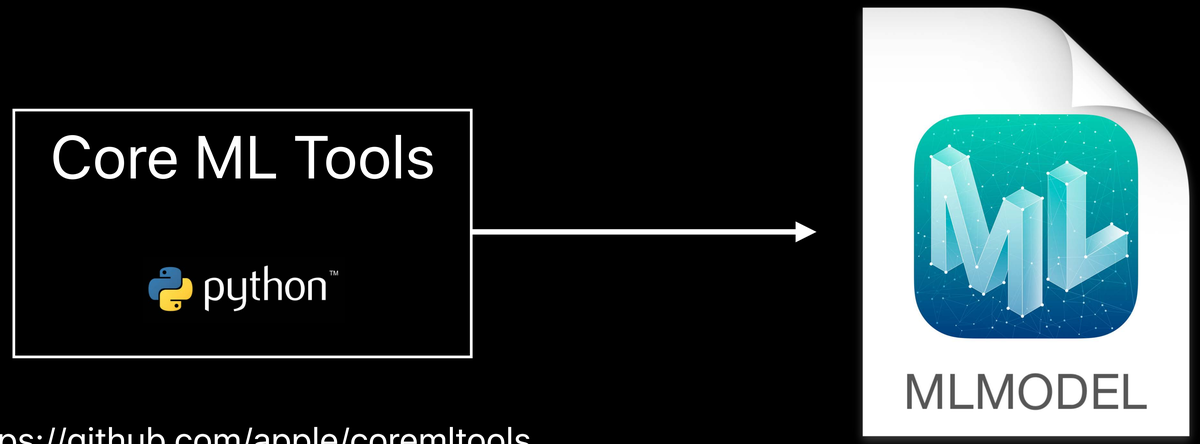
https://developer.apple.com/machine-learning/

NEW



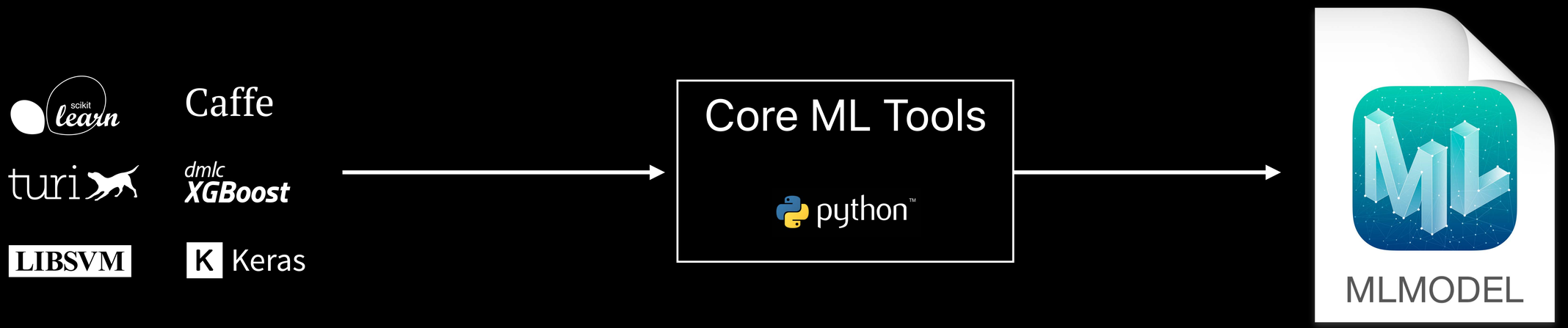
Create ML

Last Year

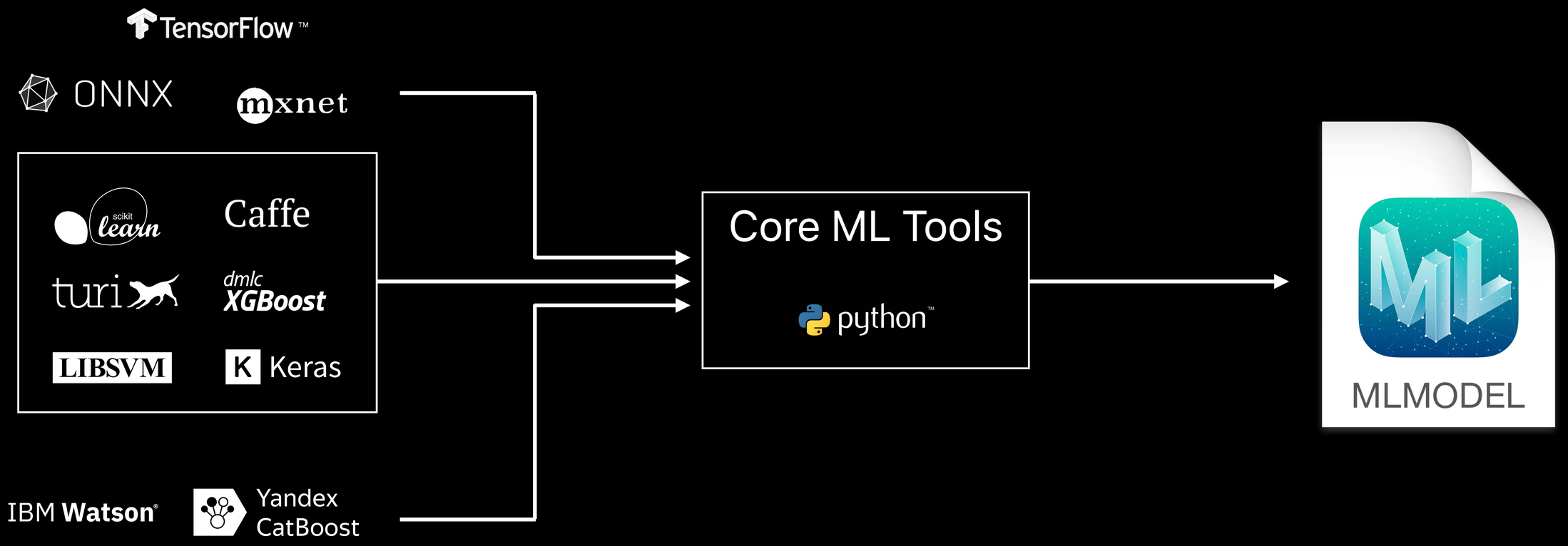


https://github.com/apple/coremltools

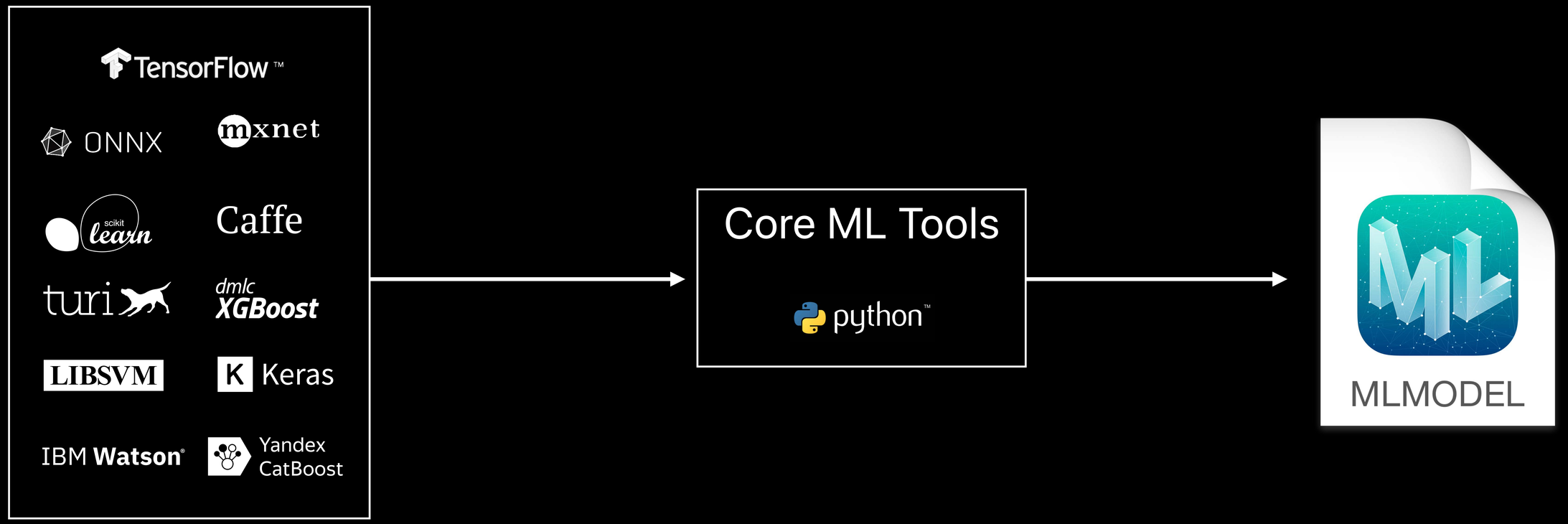
Last Year


scikit learn
Caffe
turi
dmlc XGBoost
LIBSVM
K Keras
Core ML Tools
python™
MLMODEL

Today


TensorFlow™
ONNX
mxnet
scikit learn
Caffe
turi
dmlc XGBoost
LIBSVM
Keras
IBM Watson®
Yandex CatBoost
Core ML Tools
python™
MLMODEL

Today


TensorFlow™
ONNX
mxnet
scikit learn
Caffe
turi
dmlc XGBoost
LIBSVM
Keras
IBM Watson
Yandex CatBoost
Core ML Tools
python™
MLMODEL

TensorFlow Converter

https://github.com/tf-coreml/tf-coreml

In collaboration with Google

TensorFlow Converter

https://github.com/tf-coreml/tf-coreml

Support for custom layers

TensorFlow Converter

https://github.com/tf-coreml/tf-coreml

Support for Quantized TensorFlow models

(coming soon!)

TensorFlow Converter

https://github.com/onnx/onnx-coreml

In collaboration with Facebook and Prisma

TensorFlow Converter

https://github.com/onnx/onnx-coreml

Core ML Tools ecosystem

Quantization utilities

Custom conversion

Core ML Tools 2.0

Quantization utilities

Core ML Tools 2.0

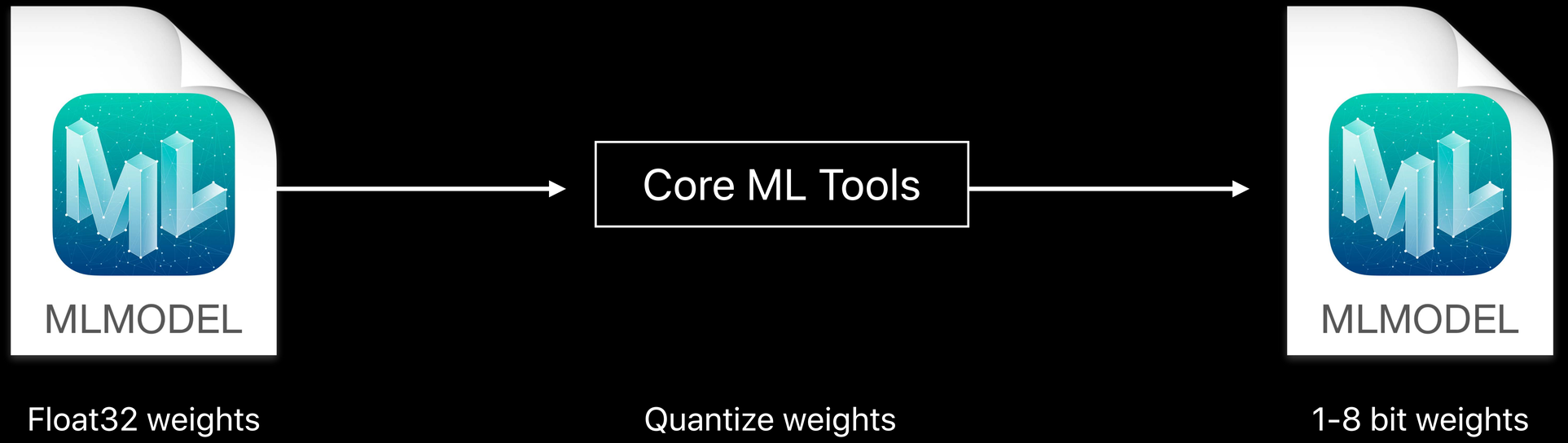
https://github.com/apple/coremltools

Support for latest Core ML .mlmodel specification

Quantization utilities

Flexible shape utilities

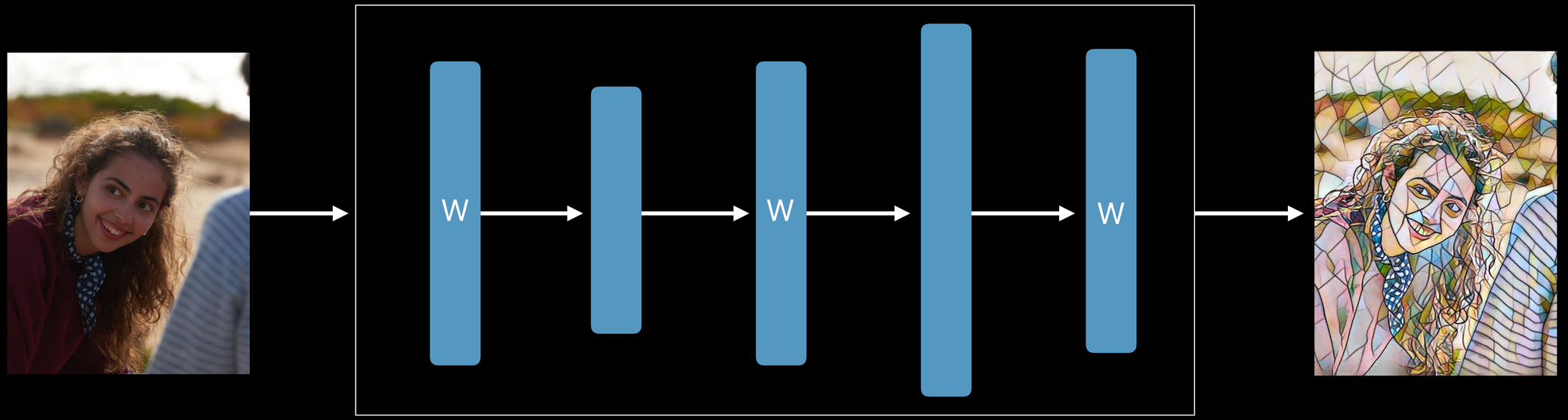
Quantization Utilities



Post-training Quantization

Quantization

Peeking under the hood

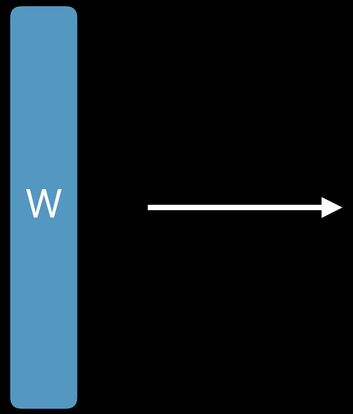


Quantization

Peeking under the hood

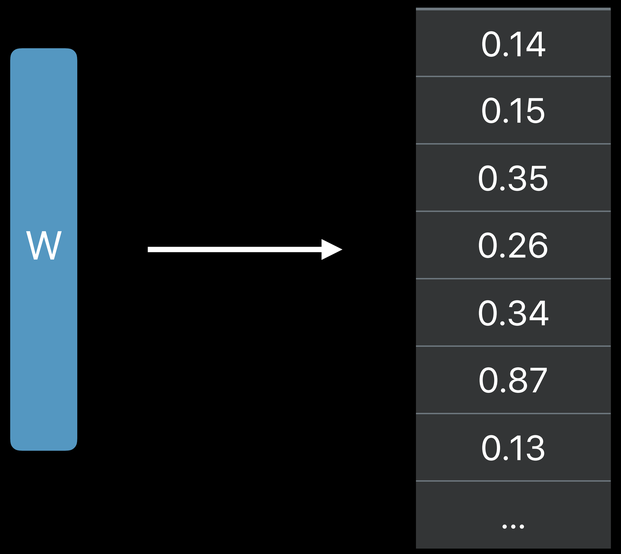
Quantization

Peeking under the hood



Quantization

Peeking under the hood

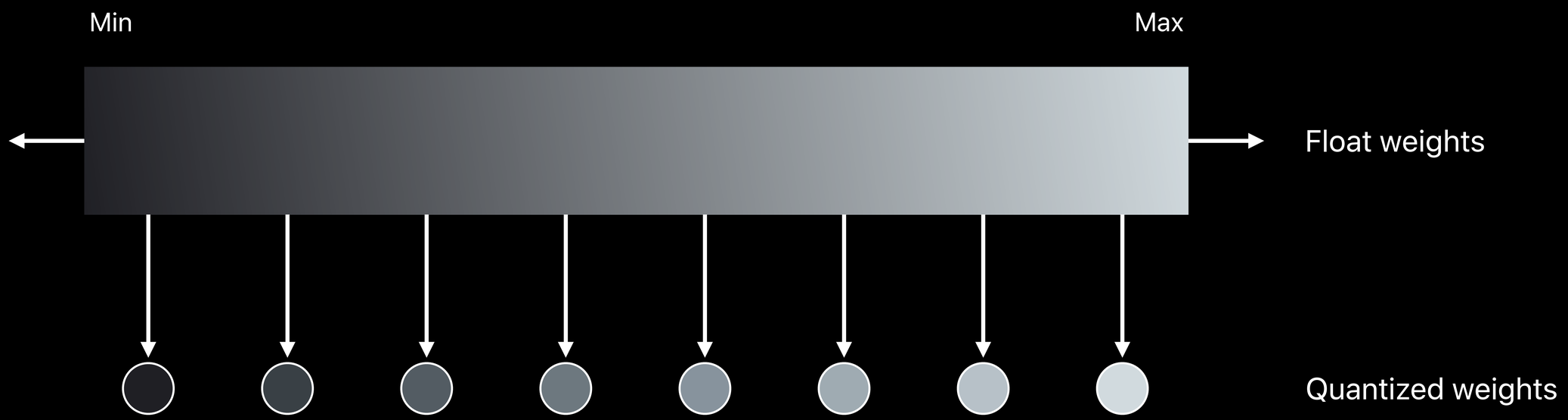


Weight Quantization

Peeking under the hood

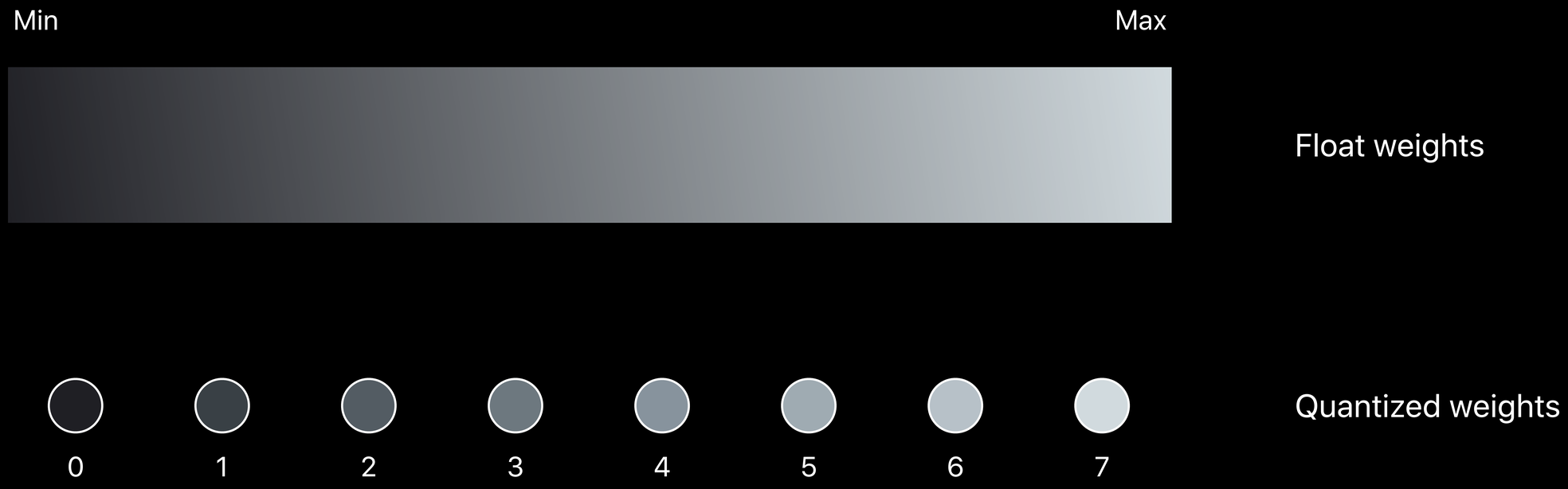
Weight Quantization

Peeking under the hood



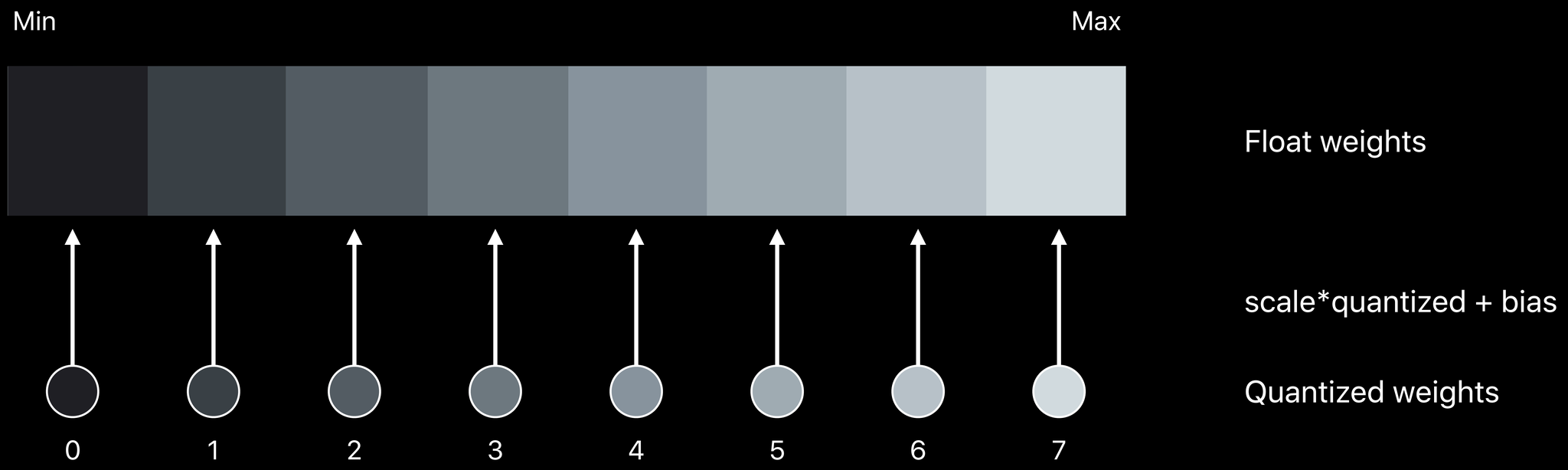
Linear

Three-bit example



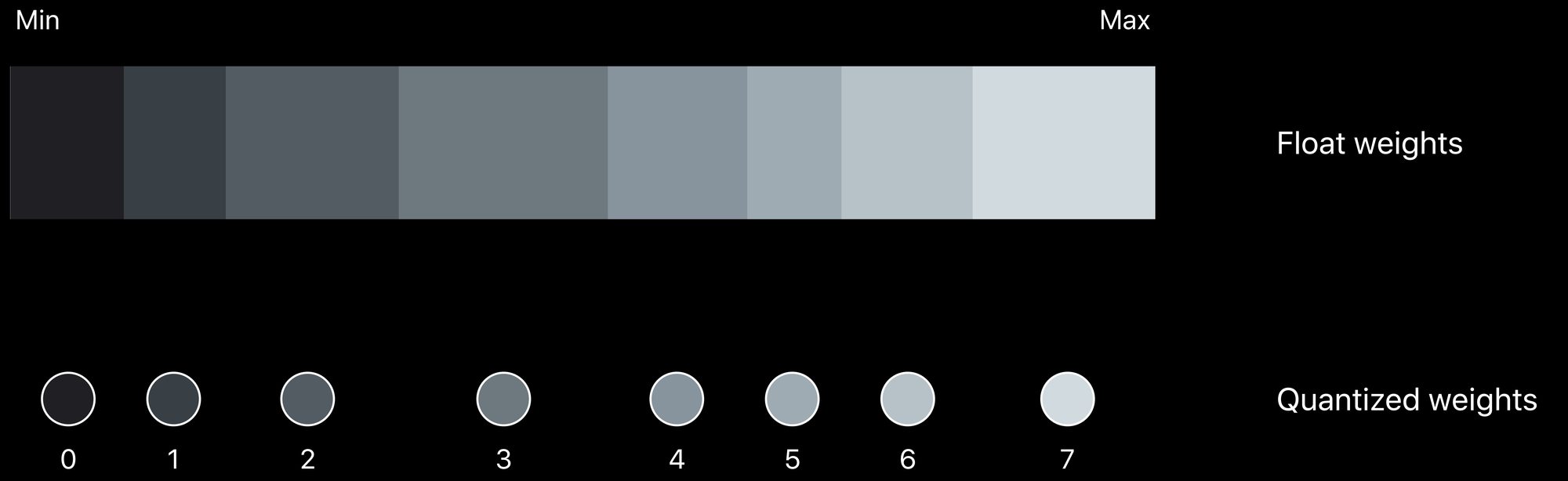
Linear

Three-bit example



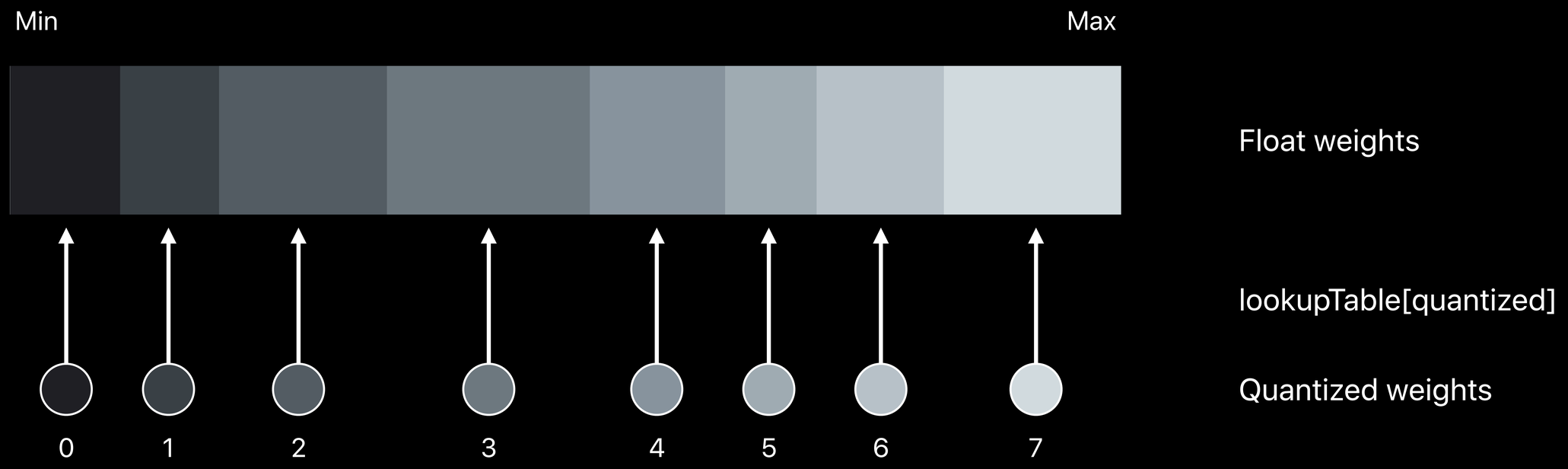
Lookup Table

Three-bit example



Lookup Table

Three-bit example



Quantization Utilities

Quantization Utilities

Decide on precision and algorithm

Quantization Utilities

Decide on precision and algorithm

Let Core ML Tools work its magic!

Demo

Quantization in Core ML Tools

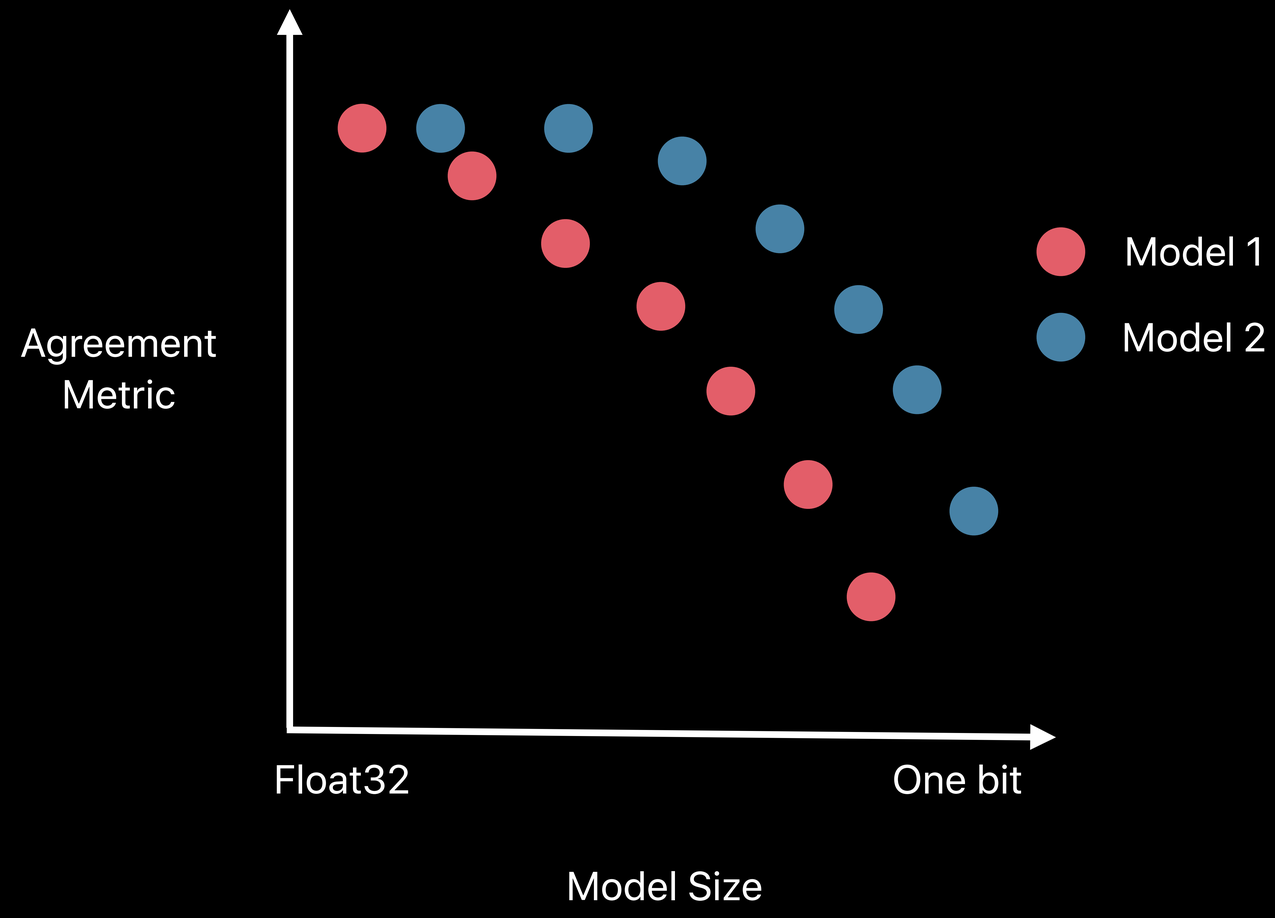
```
// Quantize model using KMeans Lookup Table
quantized_model = quantize_weights(model, 8, 'kmeans')


// Compare Quantized model with original
compare_model(model, quantized_model, './sample_data/')
```

```
// Quantize model using KMeans Lookup Table
quantized_model = quantize_weights(model, 8, 'kmeans')


// Compare Quantized model with original
compare_model(model, quantized_model, './sample_data/')
```

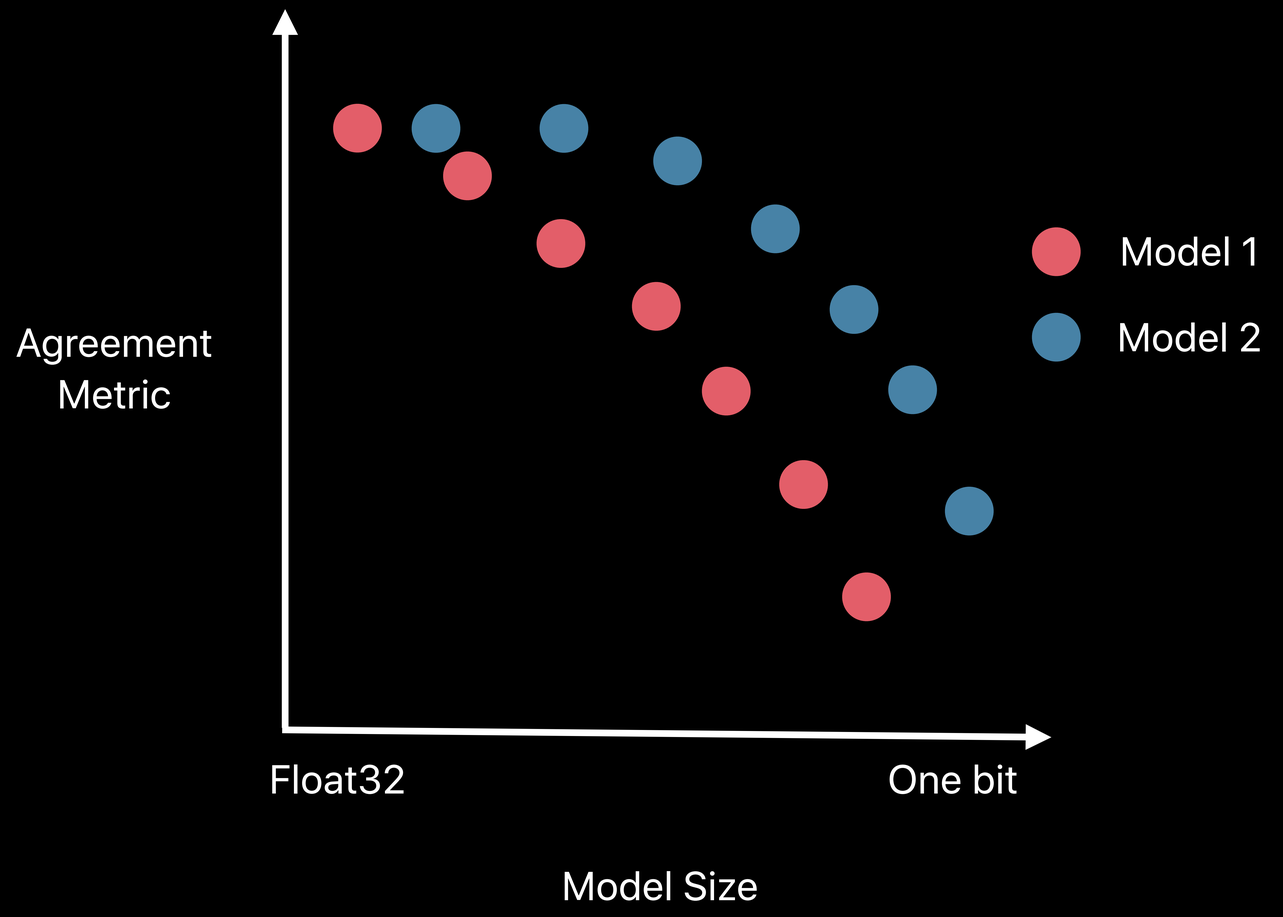
```
// Quantize model using KMeans Lookup Table
quantized_model = quantize_weights(model, 8, 'kmeans')


// Compare Quantized model with original
compare_model(model, quantized_model, './sample_data/')
```

Model Size Versus Agreement

Model Size Versus Agreement



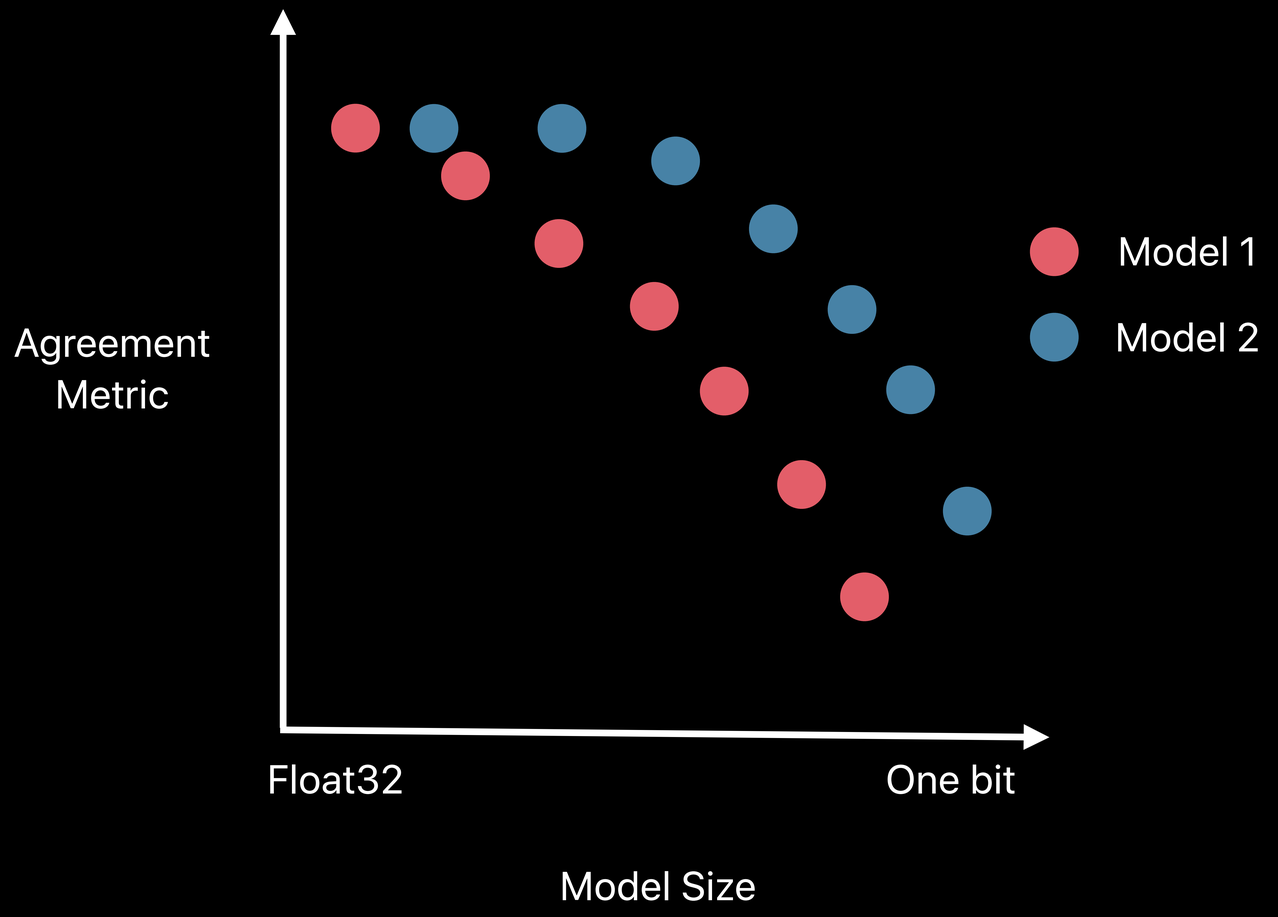
Model Size Versus Agreement



Model Size Versus Agreement

Metrics

- Accuracy
- Signal-to-noise ratio
- Visual inspection
- Model-specific metrics



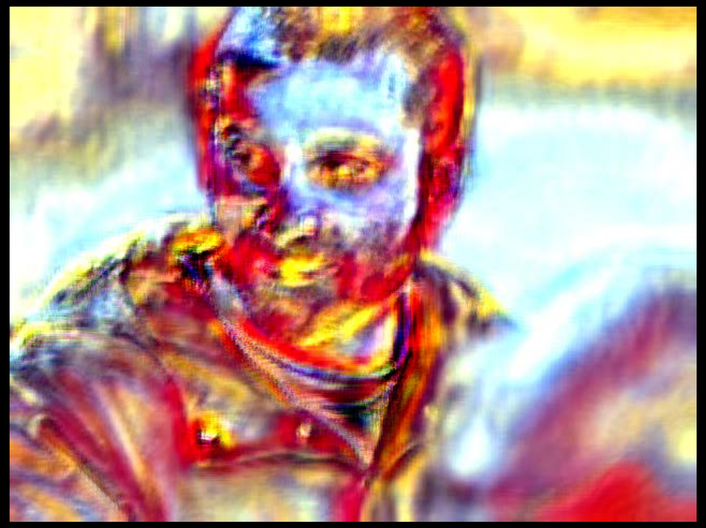
32 bit - 6.7 MB

16 bit - 3.4 MB

8 bit (linear) - 1.7 MB

4 bit (linear) - 857 KB

3 bit (linear) - 647 KB



2 bit (linear) - 437 KB

Core ML tools ecosystem

Quantization utilities

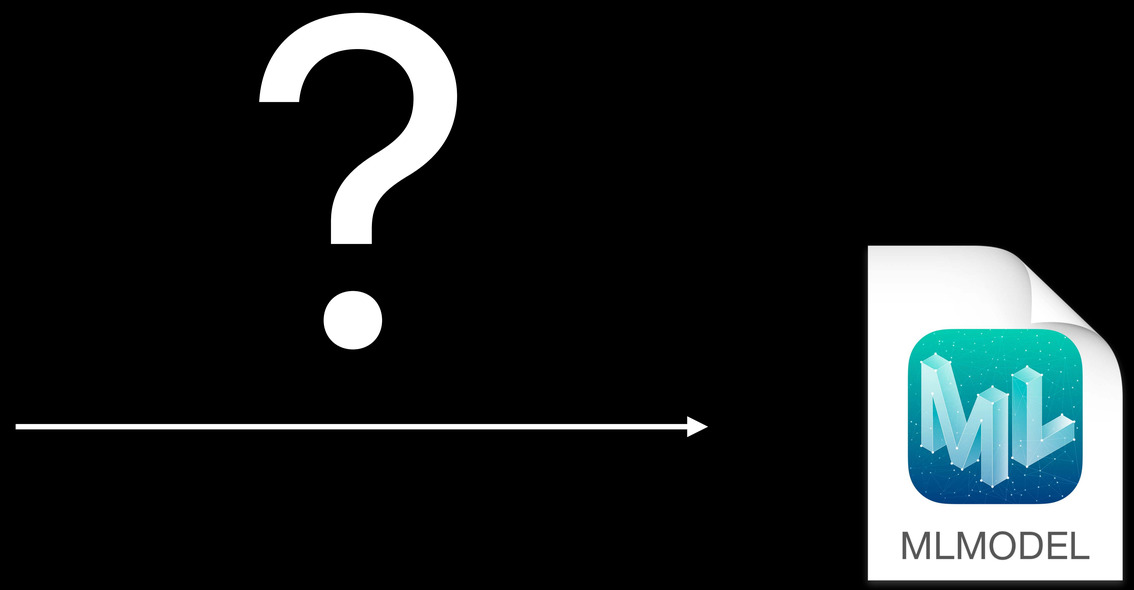
Custom conversion

Custom Conversion

Aseem Wadhwa, Core ML

New Neural Network Layer

New Model Architecture



New Neural Network Layer

New Model Architecture

Customization

New Neural Network Layer

Customization

Model Conversion

Converters: Simple API

Model Conversion

Converters: Simple API

```
import coremltools
coremltools.converters.keras.convert(keras_model)
```

Model Conversion

Converters: Simple API

```
import onnx_coreml
onnx_coreml.convert(onnx_model)
```

Model Conversion

Converters: Simple API

```
import tfcoreml
tfcoreml.convert(tf_model_path=tf_model_path,
                 mlmodel_path=mlmodel_path,
                 output_feature_names=['output:0'])
```

Model Conversion

Converters: Simple API

```
import tfcoreml
tfcoreml.convert(tf_model_path=tf_model_path,
                 mlmodel_path=mlmodel_path,
                 output_feature_names=['output:0'],
        )
```

```
NotImplementedError: Unsupported Ops of type: Tile
```

Model Conversion

Converters: Simple API

```
import tfcoreml
tfcoreml.convert(tf_model_path=tf_model_path,
                 mlmodel_path=mlmodel_path,
                 output_feature_names=['output:0'],
        )
```

Use Custom Layers!

Custom Layer Examples

Image Classifier

▼ **Model Class**

C ImageClassifierModel

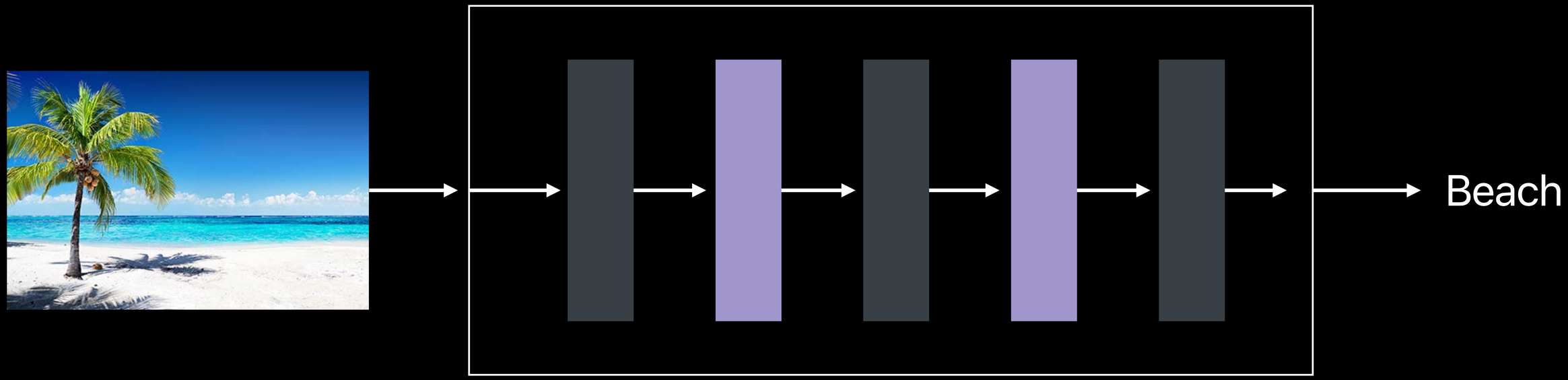
Model is not part of any target. Add the model to a target to enable generation of the model class.

▼ **Model Evaluation Parameters**

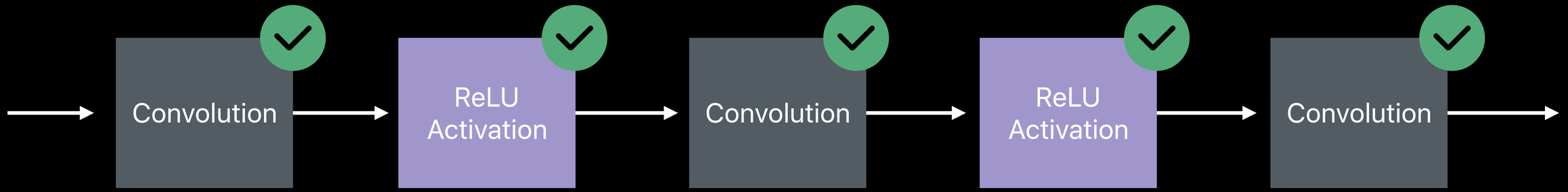
Name	Type	Description
▼ Inputs		
input_image	Image (Color 320 x 240)	Input image
▼ Outputs		
probability_vector	MultiArray (Double 10)	Output probability vector

Beach

Opening the Hood

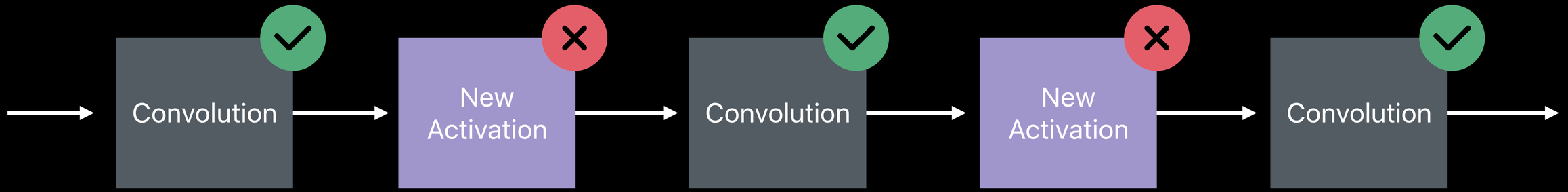


Convertible to Core ML



Supported by Core ML

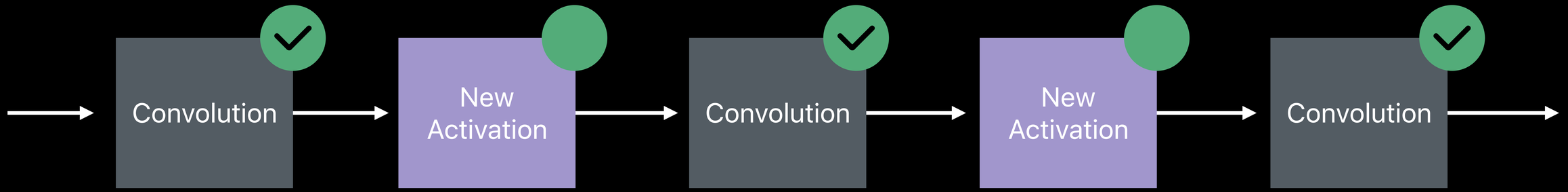
Not convertible to Core ML



Supported by Core ML

Not Supported by Core ML

Custom Layer



Supported by Core ML

Custom Layer in Core ML Model

Image Classifier

▼ **Model Class**

ImageClassifierModel

Model is not part of any target. Add the model to a target to enable generation of the model class.

▼ **Model Evaluation Parameters**

Name	Type	Description
▼ Inputs		
input_image	Image (Color 320 x 240)	Input image
▼ Outputs		
probability_vector	MultiArray (Double 10)	Output probability vector

▼ **Dependencies**

Name	Description
▼ Custom Layers	
AAPLMyNewActivation	A new activation function I want to try out.

Beach

With Custom Activation Layer

Image Classifier

Model Class

ImageClassifierModel

Model is not part of any target. Add the model to a target to enable generation of the model class.

Model Evaluation Parameters

Name	Type	Description
▼ Inputs		
input_image	Image (Color 320 x 240)	Input image
▼ Outputs		
probability_vector	MultiArray (Double 10)	Output probability vector

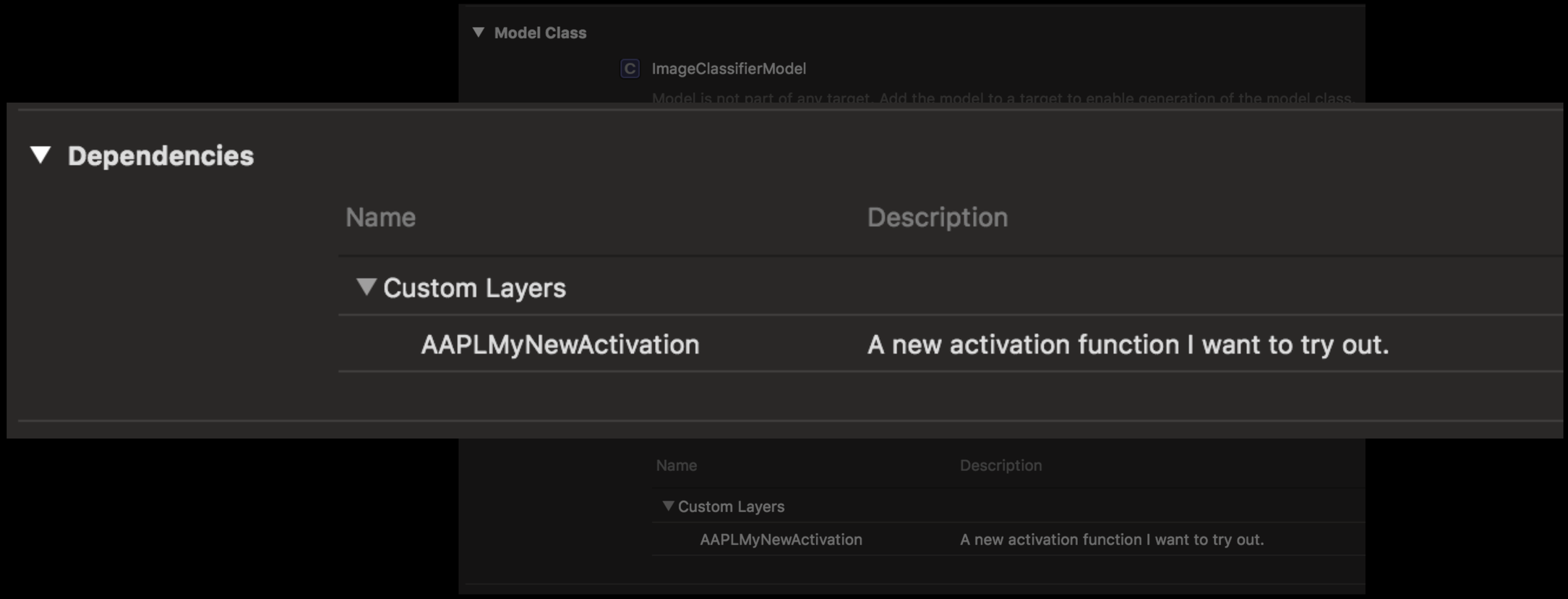
Dependencies

Name	Description
▼ Custom Layers	
AAPLMyNewActivation	A new activation function I want to try out.

Beach

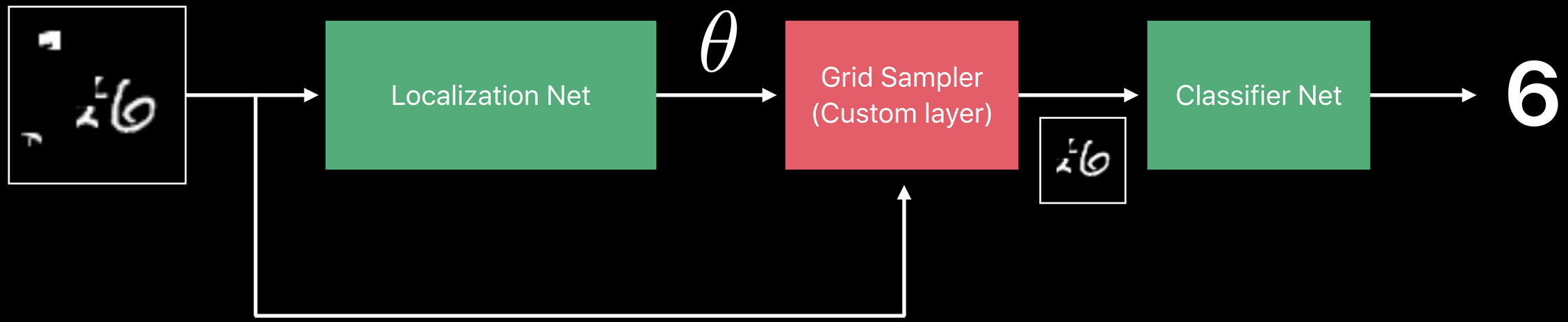
With Custom Activation Layer

Image Classifier



A Simple Classifier

Spatial Transformer Network



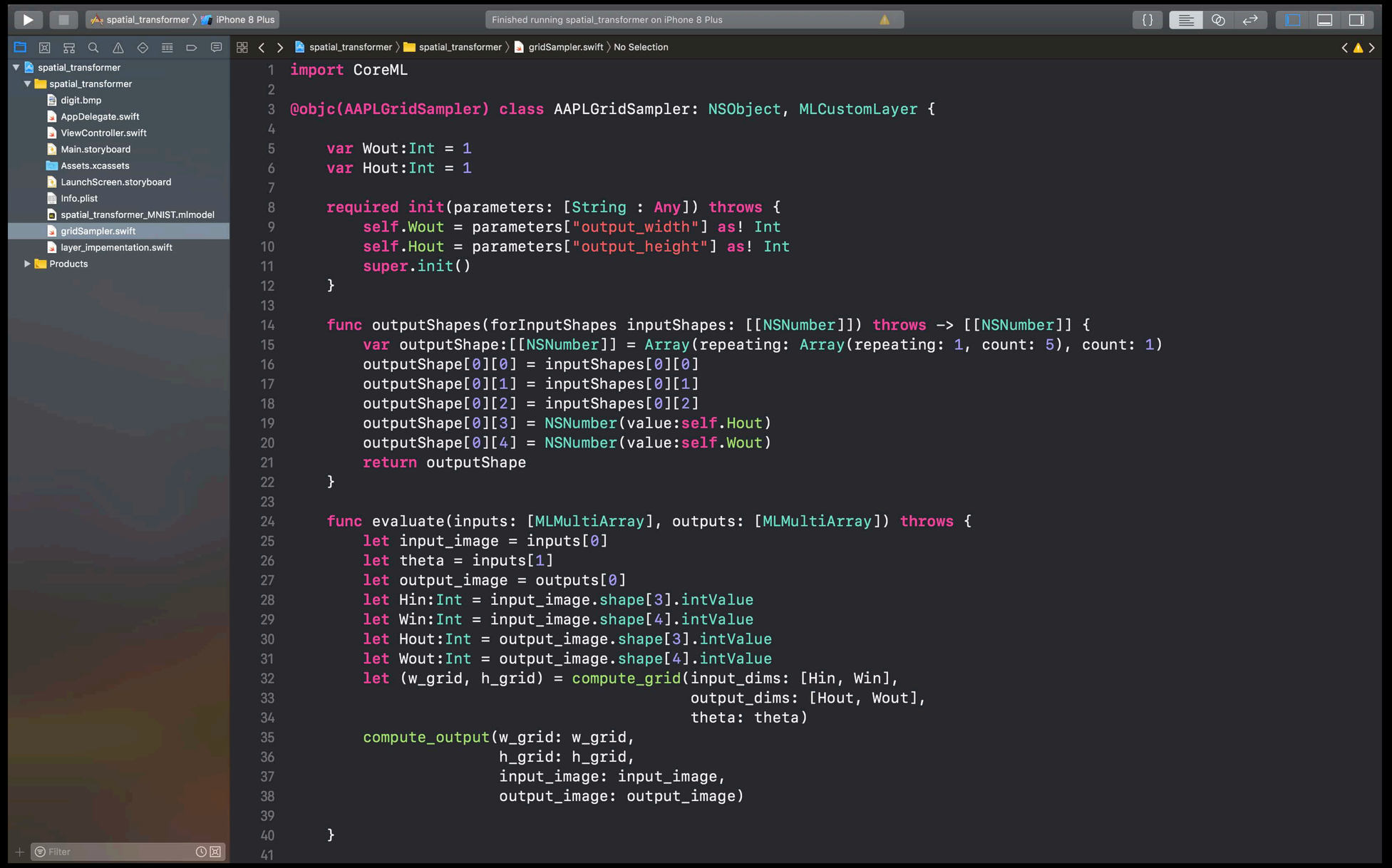
Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." *Advances in neural information processing systems*. 2015.

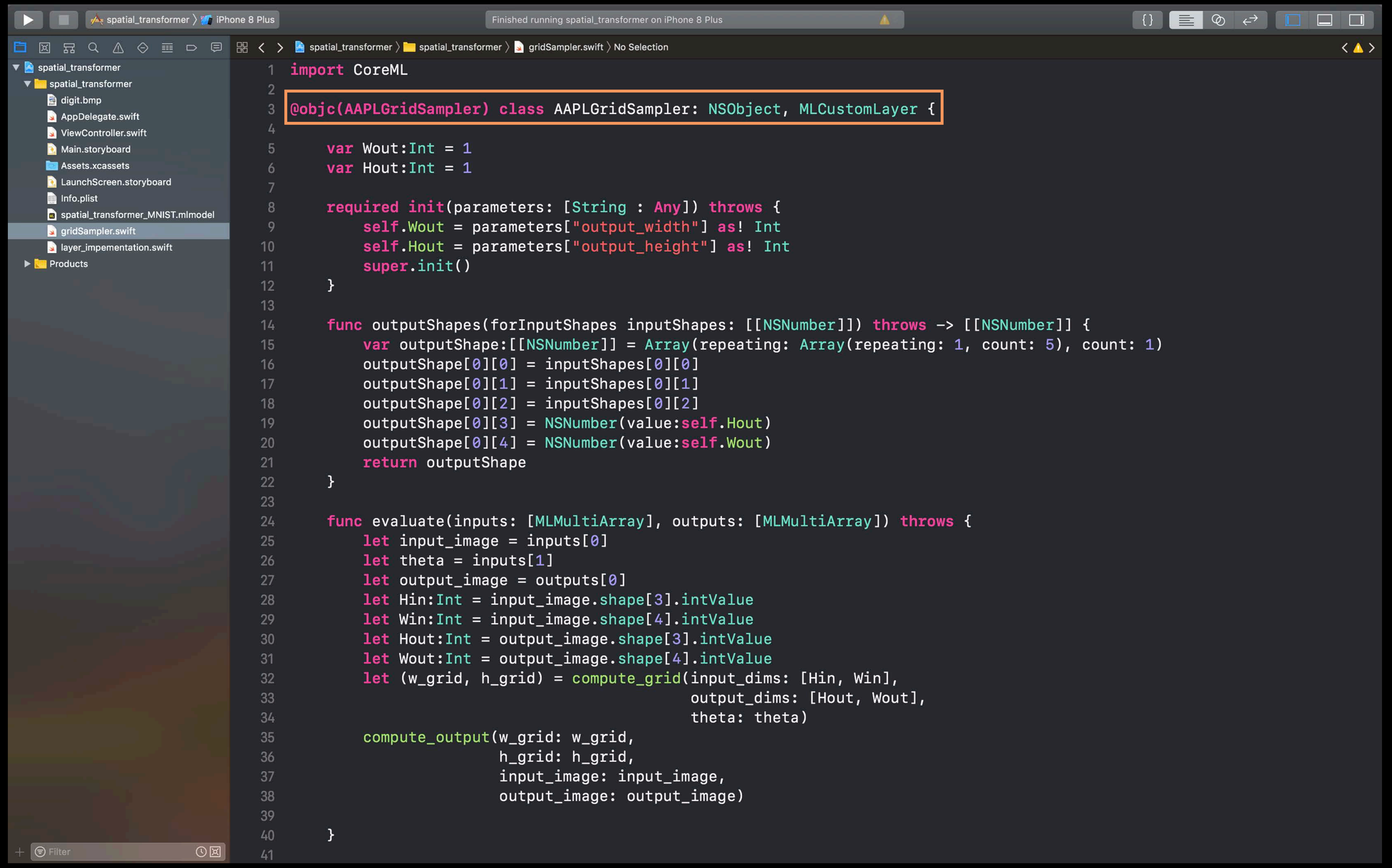
Demo

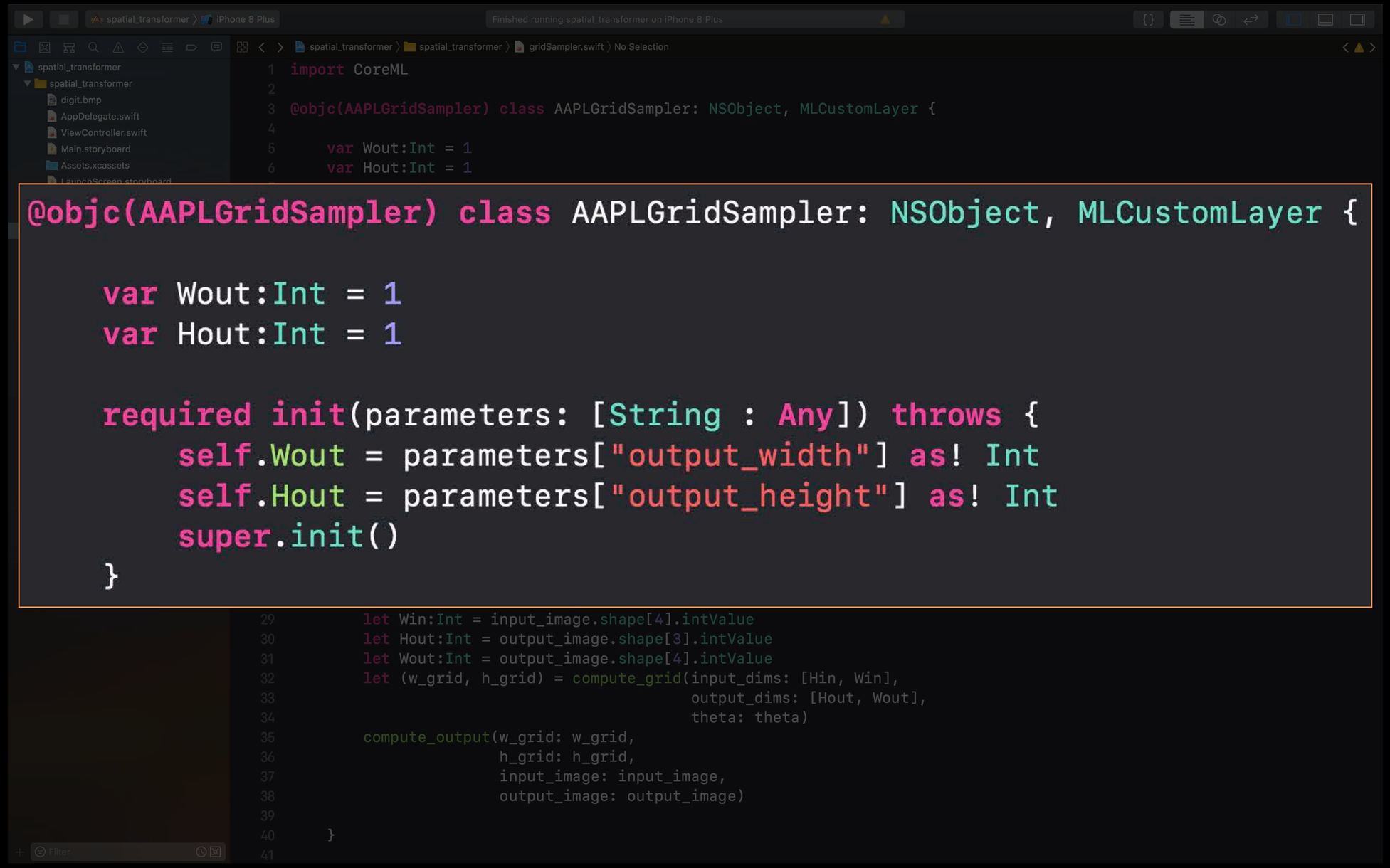
Conversion with Custom Layer

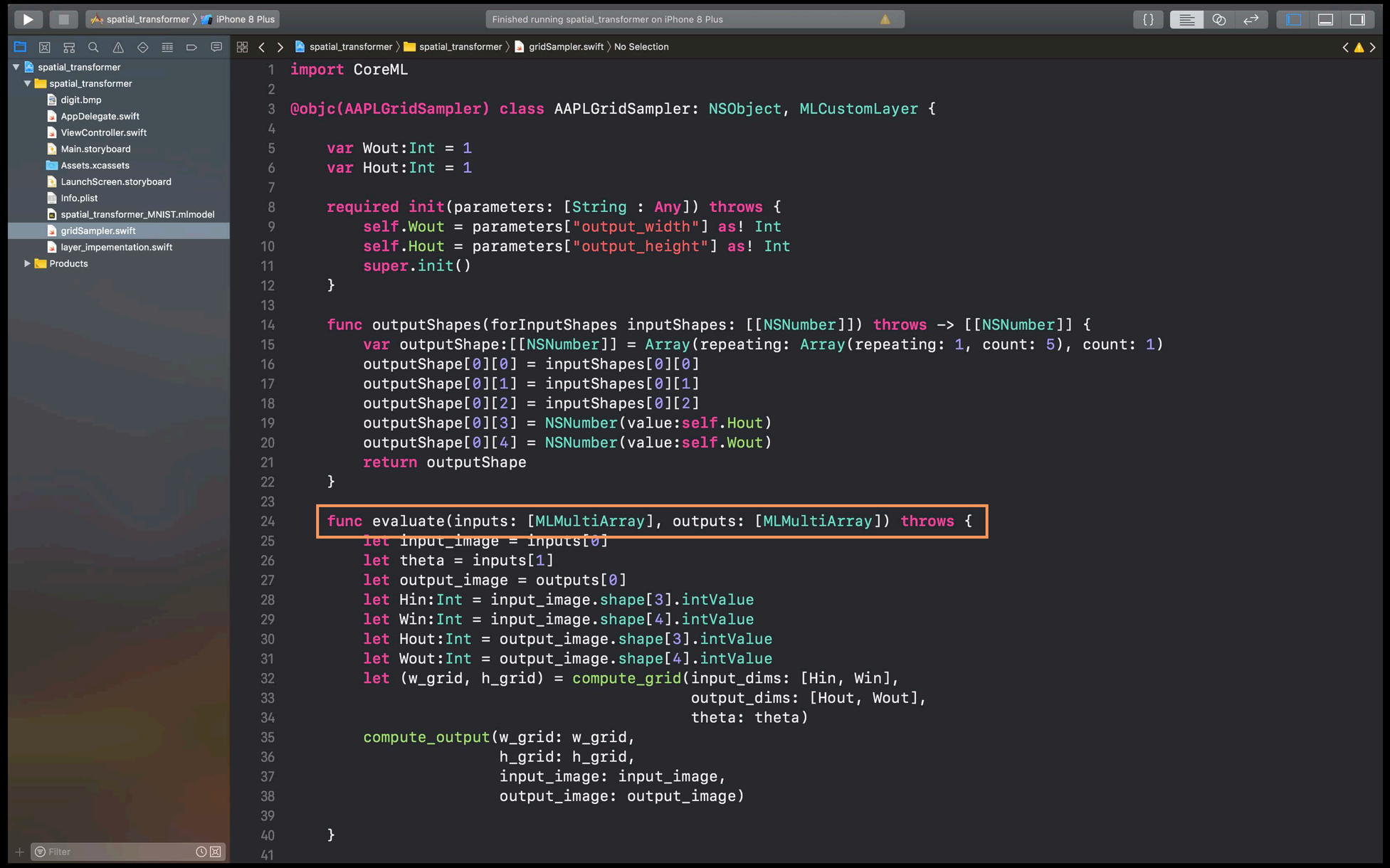
Custom Layer
(Parameters)

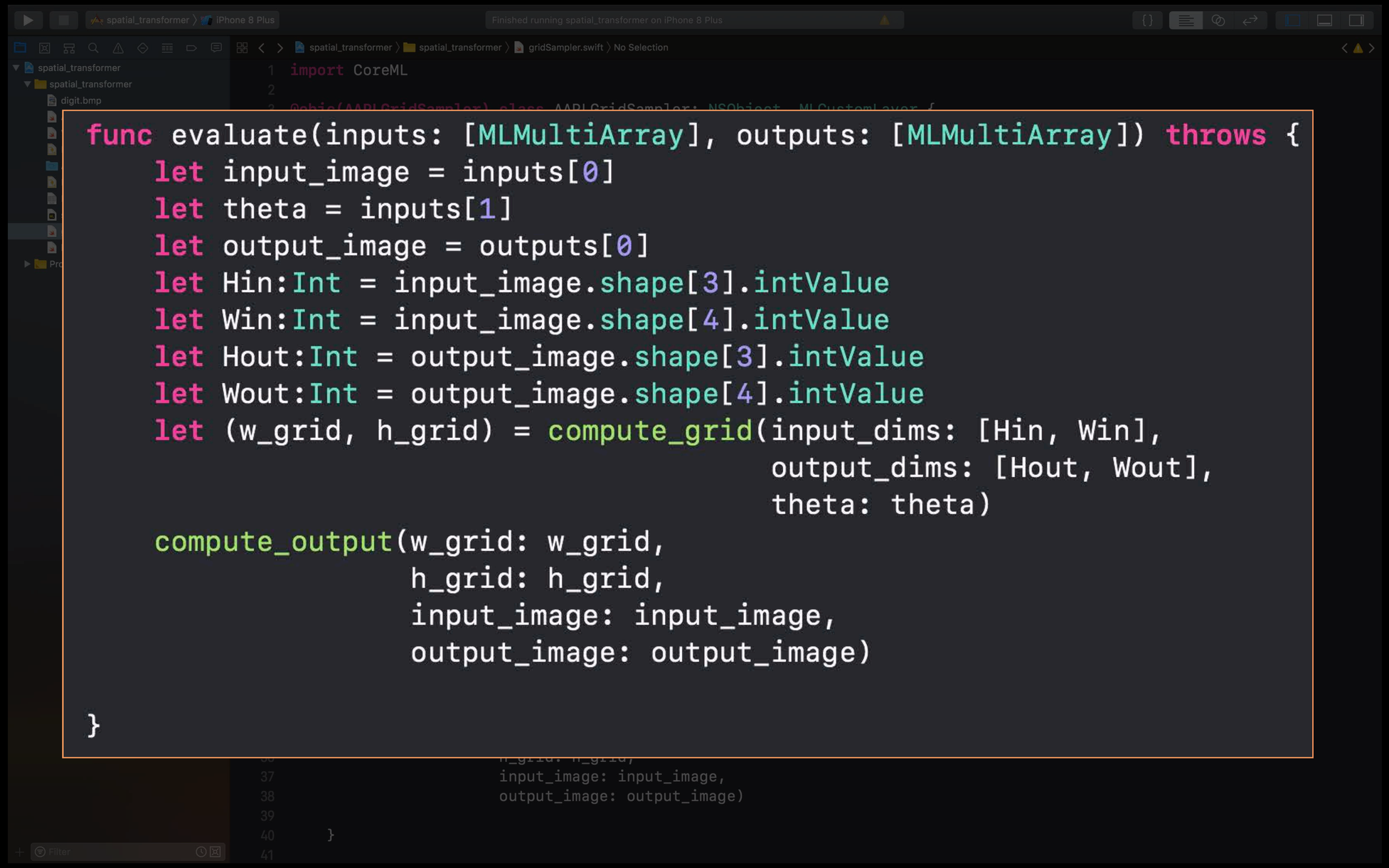
gridSampler.swift

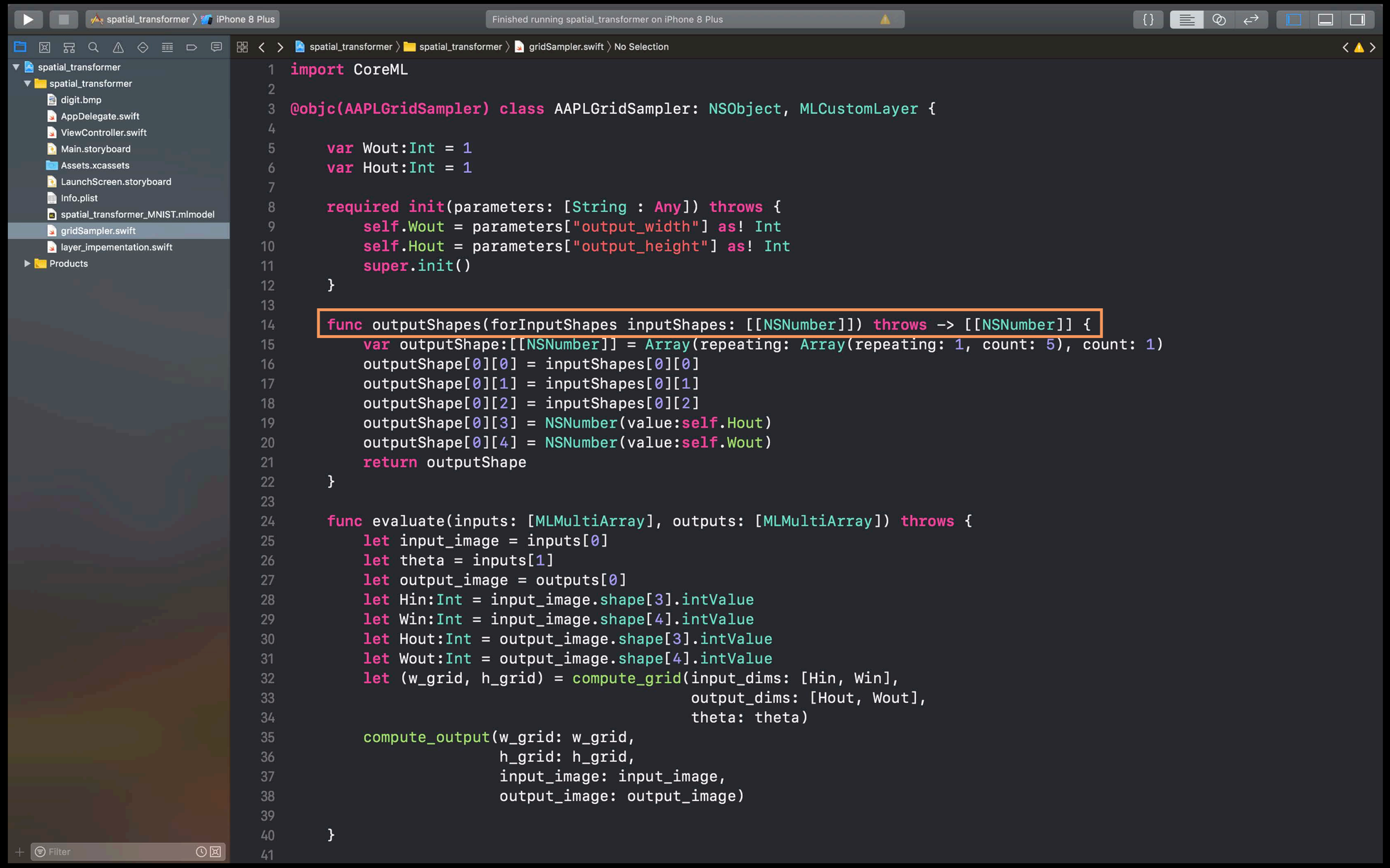

spatial_transformer 〉 iPhone 8 Plus
Finished running spatial_transformer on iPhone 8 Plus
spatial_transformer 〉 spatial_transformer 〉 gridSampler.swift 〉 No Selection
spatial_transformer
spatial_transformer
digit.bmp
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
spatial_transformer_MNIST.mlmodel
gridSampler.swift
layer_impementation.swift
Products
Filter
import CoreML
@objc(AAPLGridSampler) class AAPLGridSampler: NSObject, MLCustomLayer {
var Wout:Int = 1
var Hout:Int = 1
required init(parameters: [String : Any]) throws {
self.Wout = parameters["output_width"] as! Int
self.Hout = parameters["output_height"] as! Int
super.init()
}
func outputShapes(forInputShapes inputShapes: [[NSNumber]]) throws -> [[NSNumber]] {
var outputShape:[[NSNumber]] = Array(repeating: Array(repeating: 1, count: 5), count: 1)
outputShape[0][0] = inputShapes[0][0]
outputShape[0][1] = inputShapes[0][1]
outputShape[0][2] = inputShapes[0][2]
outputShape[0][3] = NSNumber(value:self.Hout)
outputShape[0][4] = NSNumber(value:self.Wout)
return outputShape
}
func evaluate(inputs: [MLMultiArray], outputs: [MLMultiArray]) throws {
let input_image = inputs[0]
let theta = inputs[1]
let output_image = outputs[0]
let Hin:Int = input_image.shape[3].intValue
let Win:Int = input_image.shape[4].intValue
let Hout:Int = output_image.shape[3].intValue
let Wout:Int = output_image.shape[4].intValue
let (w_grid, h_grid) = compute_grid(input_dims: [Hin, Win],
output_dims: [Hout, Wout],
theta: theta)
compute_output(w_grid: w_grid,
h_grid: h_grid,
input_image: input_image,
output_image: output_image)
}

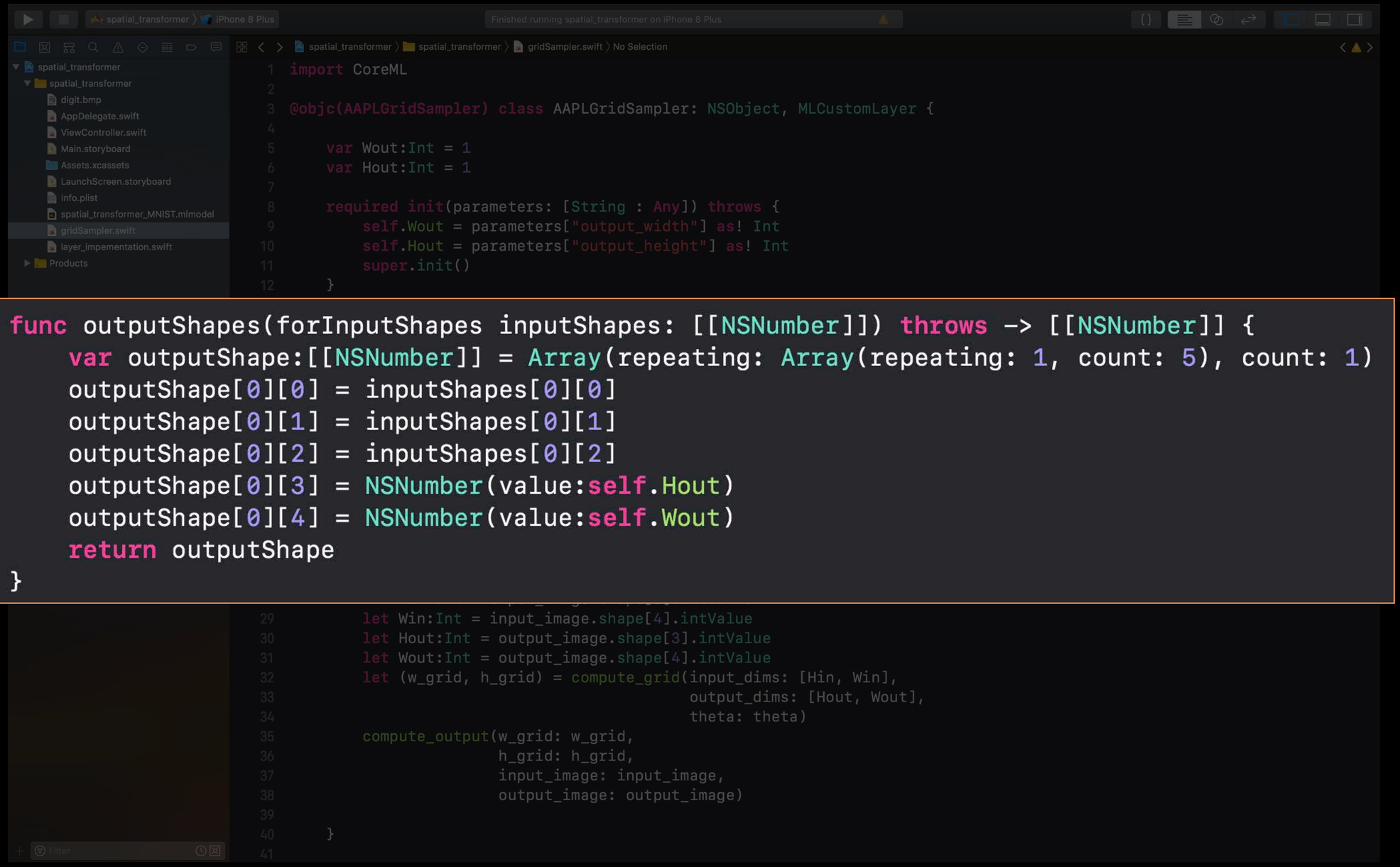

spatial_transformer 〉 iPhone 8 Plus
Finished running spatial_transformer on iPhone 8 Plus
spatial_transformer 〉 spatial_transformer 〉 gridSampler.swift 〉 No Selection
spatial_transformer
spatial_transformer
digit.bmp
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
spatial_transformer_MNIST.mlmodel
gridSampler.swift
layer_impementation.swift
Products
Filter
1 import CoreML
2
3 @objc(AAPLGridSampler) class AAPLGridSampler: NSObject, MLCustomLayer {
4
5 var Wout:Int = 1
6 var Hout:Int = 1
7
8 required init(parameters: [String : Any]) throws {
9 self.Wout = parameters["output_width"] as! Int
10 self.Hout = parameters["output_height"] as! Int
11 super.init()
12 }
13
14 func outputShapes(forInputShapes inputShapes: [[NSNumber]]) throws -> [[NSNumber]] {
15 var outputShape:[[NSNumber]] = Array(repeating: Array(repeating: 1, count: 5), count: 1)
16 outputShape[0][0] = inputShapes[0][0]
17 outputShape[0][1] = inputShapes[0][1]
18 outputShape[0][2] = inputShapes[0][2]
19 outputShape[0][3] = NSNumber(value:self.Hout)
20 outputShape[0][4] = NSNumber(value:self.Wout)
21 return outputShape
22 }
23
24 func evaluate(inputs: [MLMultiArray], outputs: [MLMultiArray]) throws {
25 let input_image = inputs[0]
26 let theta = inputs[1]
27 let output_image = outputs[0]
28 let Hin:Int = input_image.shape[3].intValue
29 let Win:Int = input_image.shape[4].intValue
30 let Hout:Int = output_image.shape[3].intValue
31 let Wout:Int = output_image.shape[4].intValue
32 let (w_grid, h_grid) = compute_grid(input_dims: [Hin, Win],
33 output_dims: [Hout, Wout],
34 theta: theta)
35 compute_output(w_grid: w_grid,
36 h_grid: h_grid,
37 input_image: input_image,
38 output_image: output_image)
39
40 }
41


spatial_transformer
iPhone 8 Plus
Finished running spatial_transformer on iPhone 8 Plus
spatial_transformer
spatial_transformer
digit.bmp
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
spatial_transformer
spatial_transformer
gridSampler.swift
No Selection
import CoreML
@objc(AAPLGridSampler) class AAPLGridSampler: NSObject, MLCustomLayer {
var Wout:Int = 1
var Hout:Int = 1
@objc(AAPLGridSampler) class AAPLGridSampler: NSObject, MLCustomLayer {
var Wout:Int = 1
var Hout:Int = 1
required init(parameters: [String : Any]) throws {
self.Wout = parameters["output_width"] as! Int
self.Hout = parameters["output_height"] as! Int
super.init()
}
let Win:Int = input_image.shape[4].intValue
let Hout:Int = output_image.shape[3].intValue
let Wout:Int = output_image.shape[4].intValue
let (w_grid, h_grid) = compute_grid(input_dims: [Hin, Win],
output_dims: [Hout, Wout],
theta: theta)
compute_output(w_grid: w_grid,
h_grid: h_grid,
input_image: input_image,
output_image: output_image)
}


spatial_transformer 〉 iPhone 8 Plus
Finished running spatial_transformer on iPhone 8 Plus
spatial_transformer 〉 spatial_transformer 〉 gridSampler.swift 〉 No Selection
spatial_transformer
spatial_transformer
digit.bmp
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
spatial_transformer_MNIST.mlmodel
gridSampler.swift
layer_impementation.swift
Products
Filter
import CoreML
@objc(AAPLGridSampler) class AAPLGridSampler: NSObject, MLCustomLayer {
var Wout:Int = 1
var Hout:Int = 1
required init(parameters: [String : Any]) throws {
self.Wout = parameters["output_width"] as! Int
self.Hout = parameters["output_height"] as! Int
super.init()
}
func outputShapes(forInputShapes inputShapes: [[NSNumber]]) throws -> [[NSNumber]] {
var outputShape:[[NSNumber]] = Array(repeating: Array(repeating: 1, count: 5), count: 1)
outputShape[0][0] = inputShapes[0][0]
outputShape[0][1] = inputShapes[0][1]
outputShape[0][2] = inputShapes[0][2]
outputShape[0][3] = NSNumber(value:self.Hout)
outputShape[0][4] = NSNumber(value:self.Wout)
return outputShape
}
func evaluate(inputs: [MLMultiArray], outputs: [MLMultiArray]) throws {
let input_image = inputs[0]
let theta = inputs[1]
let output_image = outputs[0]
let Hin:Int = input_image.shape[3].intValue
let Win:Int = input_image.shape[4].intValue
let Hout:Int = output_image.shape[3].intValue
let Wout:Int = output_image.shape[4].intValue
let (w_grid, h_grid) = compute_grid(input_dims: [Hin, Win],
output_dims: [Hout, Wout],
theta: theta)
compute_output(w_grid: w_grid,
h_grid: h_grid,
input_image: input_image,
output_image: output_image)
}

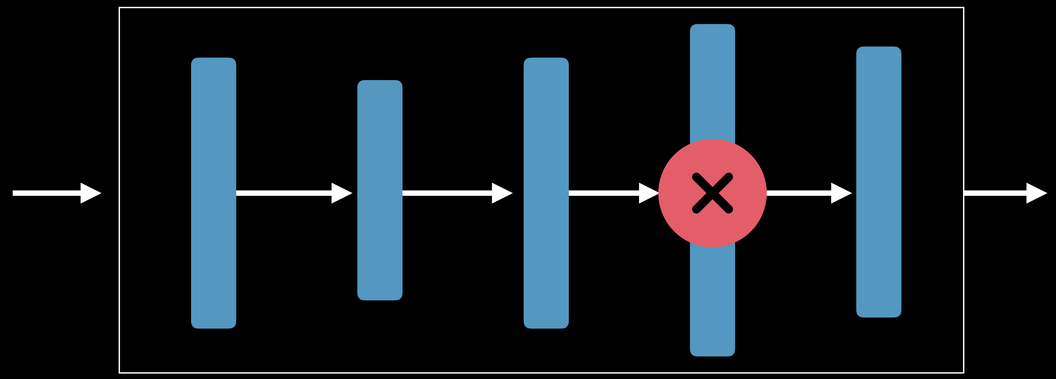

spatial_transformer
iPhone 8 Plus
Finished running spatial_transformer on iPhone 8 Plus
spatial_transformer
spatial_transformer
gridSampler.swift
No Selection
spatial_transformer
spatial_transformer
digit.bmp
import CoreML
func evaluate(inputs: [MLMultiArray], outputs: [MLMultiArray]) throws {
let input_image = inputs[0]
let theta = inputs[1]
let output_image = outputs[0]
let Hin:Int = input_image.shape[3].intValue
let Win:Int = input_image.shape[4].intValue
let Hout:Int = output_image.shape[3].intValue
let Wout:Int = output_image.shape[4].intValue
let (w_grid, h_grid) = compute_grid(input_dims: [Hin, Win],
output_dims: [Hout, Wout],
theta: theta)
compute_output(w_grid: w_grid,
h_grid: h_grid,
input_image: input_image,
output_image: output_image)
}
input_image: input_image,
output_image: output_image)
}


spatial_transformer 〉 iPhone 8 Plus
Finished running spatial_transformer on iPhone 8 Plus
spatial_transformer 〉 spatial_transformer 〉 gridSampler.swift 〉 No Selection
spatial_transformer
spatial_transformer
digit.bmp
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
spatial_transformer_MNIST.mlmodel
gridSampler.swift
layer_impementation.swift
Products
Filter
import CoreML
@objc(AAPLGridSampler) class AAPLGridSampler: NSObject, MLCustomLayer {
var Wout:Int = 1
var Hout:Int = 1
required init(parameters: [String : Any]) throws {
self.Wout = parameters["output_width"] as! Int
self.Hout = parameters["output_height"] as! Int
super.init()
}
func outputShapes(forInputShapes inputShapes: [[NSNumber]]) throws -> [[NSNumber]] {
var outputShape:[[NSNumber]] = Array(repeating: Array(repeating: 1, count: 5), count: 1)
outputShape[0][0] = inputShapes[0][0]
outputShape[0][1] = inputShapes[0][1]
outputShape[0][2] = inputShapes[0][2]
outputShape[0][3] = NSNumber(value:self.Hout)
outputShape[0][4] = NSNumber(value:self.Wout)
return outputShape
}
func evaluate(inputs: [MLMultiArray], outputs: [MLMultiArray]) throws {
let input_image = inputs[0]
let theta = inputs[1]
let output_image = outputs[0]
let Hin:Int = input_image.shape[3].intValue
let Win:Int = input_image.shape[4].intValue
let Hout:Int = output_image.shape[3].intValue
let Wout:Int = output_image.shape[4].intValue
let (w_grid, h_grid) = compute_grid(input_dims: [Hin, Win],
output_dims: [Hout, Wout],
theta: theta)
compute_output(w_grid: w_grid,
h_grid: h_grid,
input_image: input_image,
output_image: output_image)
}


spatial_transformer 〉 iPhone 8 Plus
Finished running spatial_transformer on iPhone 8 Plus
spatial_transformer 〉 spatial_transformer 〉 gridSampler.swift 〉 No Selection
spatial_transformer
spatial_transformer
digit.bmp
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
spatial_transformer_MNIST.mlmodel
gridSampler.swift
layer_impementation.swift
Products
import CoreML
@objc(AAPLGridSampler) class AAPLGridSampler: NSObject, MLCustomLayer {
var Wout:Int = 1
var Hout:Int = 1
required init(parameters: [String : Any]) throws {
self.Wout = parameters["output_width"] as! Int
self.Hout = parameters["output_height"] as! Int
super.init()
}
func outputShapes(forInputShapes inputShapes: [[NSNumber]]) throws -> [[NSNumber]] {
var outputShape:[[NSNumber]] = Array(repeating: Array(repeating: 1, count: 5), count: 1)
outputShape[0][0] = inputShapes[0][0]
outputShape[0][1] = inputShapes[0][1]
outputShape[0][2] = inputShapes[0][2]
outputShape[0][3] = NSNumber(value:self.Hout)
outputShape[0][4] = NSNumber(value:self.Wout)
return outputShape
}
let Win:Int = input_image.shape[4].intValue
let Hout:Int = output_image.shape[3].intValue
let Wout:Int = output_image.shape[4].intValue
let (w_grid, h_grid) = compute_grid(input_dims: [Hin, Win],
output_dims: [Hout, Wout],
theta: theta)
compute_output(w_grid: w_grid,
h_grid: h_grid,
input_image: input_image,
output_image: output_image)
}

Custom Layer

Layer in a Neural Network



Custom Model

New Model

Summary

Core ML Tools 2.0

Core ML Tools 2.0

Rich Core ML Tools ecosystem

Core ML Tools 2.0

Rich Core ML Tools ecosystem

Easy-to-use quantization utilities

Core ML Tools 2.0

Rich Core ML Tools ecosystem

Easy-to-use quantization utilities

Integrate new layers

More Information

https://developer.apple.com/wwdc18/709

Machine Learning Lab	Technology Lab 2	Wednesday 4:00PM
Machine Learning Lab	Technology Lab 12	Friday 2:00PM


WWDC18