CoreML: Everyone's an Artist

Building a Neural Style Transfer app with CoreML

What is Neural Style Transfer?

(Or NST)









Obtaining a CoreMLNST model

2 solutions

Online*

- Github, modelzoo.co...

DIY

- GPU + some ♥ for Python
- Training: Turi Create, Tensorflow, Pytorch...
- Misc: ONNX, coremitools

^{*}check the licence!

Importing a CoreML model

Easy peasy







▼ Machine Learning Model

Name StarryNight

Type Neural Network

Size 854 KB

Author Monogle

Description starry_night, linear 16 bit

License Commercial, all rights reserved

▼ Model Class

C StarryNight ©

Automatically generated Swift model class

▼ Model Evaluation Parameters

Name	Туре	Description
▼ Inputs		
inputImage	Image (Color 720 x 720)	Image to stylize
▼ Outputs		
outputImage	Image (Color 720 x 720)	Stylized image



```
StarryNight.swi
                           C StarryNightInput
                             P inputImage
                             P featureNames
                             M featureValue(for:)
4 // This file was
                                                                not be edite
                             M init(inputImage:)
                           C StarryNightOutput
                             P provider
  import CoreML
                             P outputImage
                             P featureNames
  /// Model Predict
                             M featureValue(for:)
                                                                chOS 4.2, *)
                             M init(outputImage:)
  class StarryNight
                             M init(features:)
                           C StarryNight
       /// Image to
                                                                /pe_32BGRA) im
                             P model
        var inputImag
                             P urlOfModelInThisBundle
        var featureNa
                             M init(contentsOf:)
            get {
                             M init()
                             M init(configuration:)
                             M init(contentsOf:configuration:)
                             M prediction(input:)
        func featureV
                             M prediction(input:options:)
            if (featu
                             M prediction(inputImage:)
                                                                ıtImage)
                             M predictions(inputs:options:)
```

Using a CoreML model

Easy peasy

Specific API for CoreML

- 1. Convert input to CoreML format (ML...)
- 2. Make prediction
- 3. Convert output to target format

Optimizations

- Threading
- Offload memory to disk

Livecoding

Demo for a Neural Style Transfer app

One more thing

Quantization



Resources

Going further

- NST demo sources (github.com/kirualex/NSTDemo)
- Building a Neural Style Transfer app on iOS (bit.ly/building nst ios)
- Using Quantization in iOS 12 (bit.ly/quantization ios)

Apple

- What's new in CoreML part 1 & 2
- A guide to Turi Create

Other

Jcjohnson (github.com/jcjohnson/fast-neural-style)

Thank you



@alexiscreuzot



@kirualex



looq.monoqle.fr