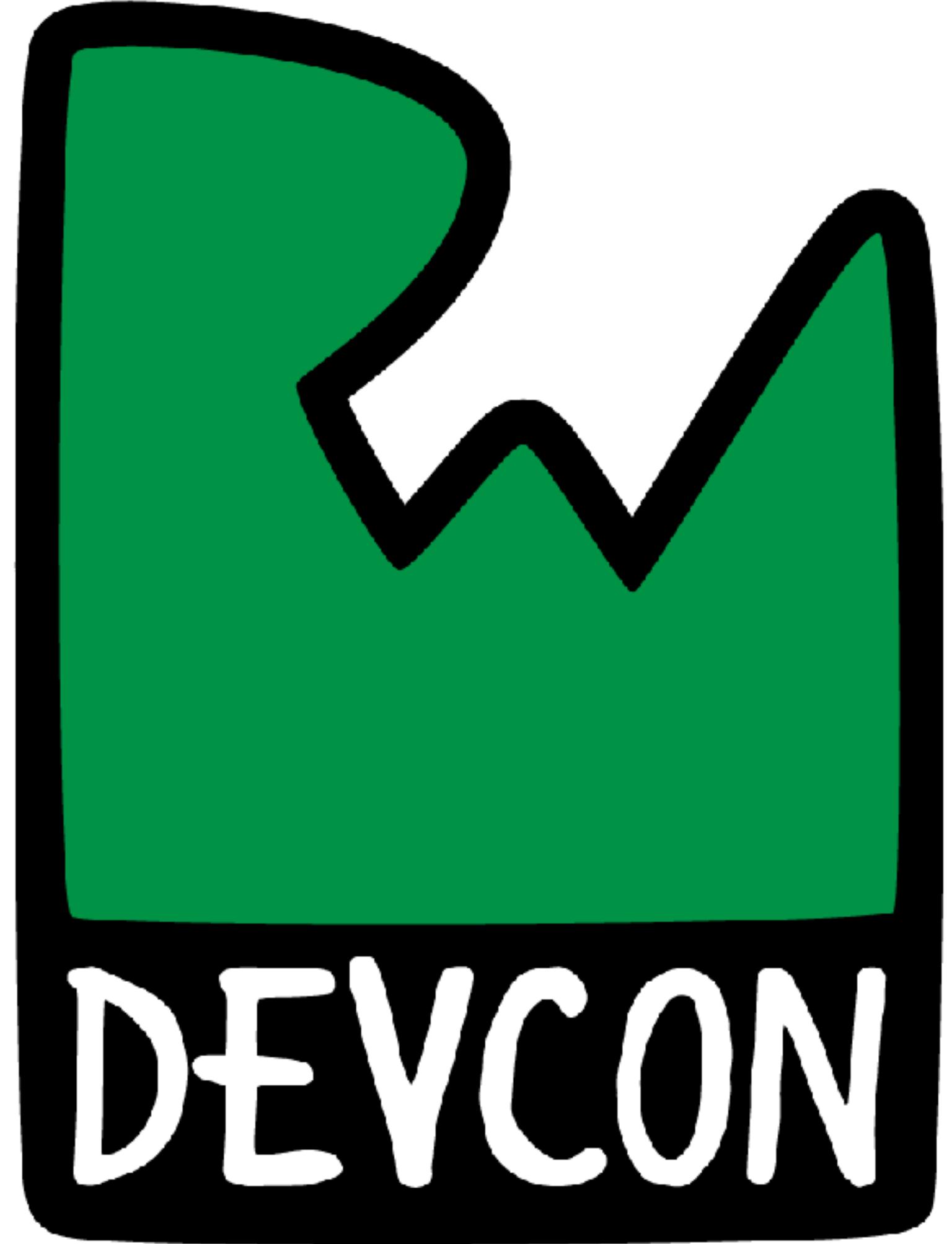


CoreML+Vision



OUT OF THE BOX

PATRICK KWETE



R
W

AUDREY TAM



watchOS
by Tutorials

THIRD EDITION

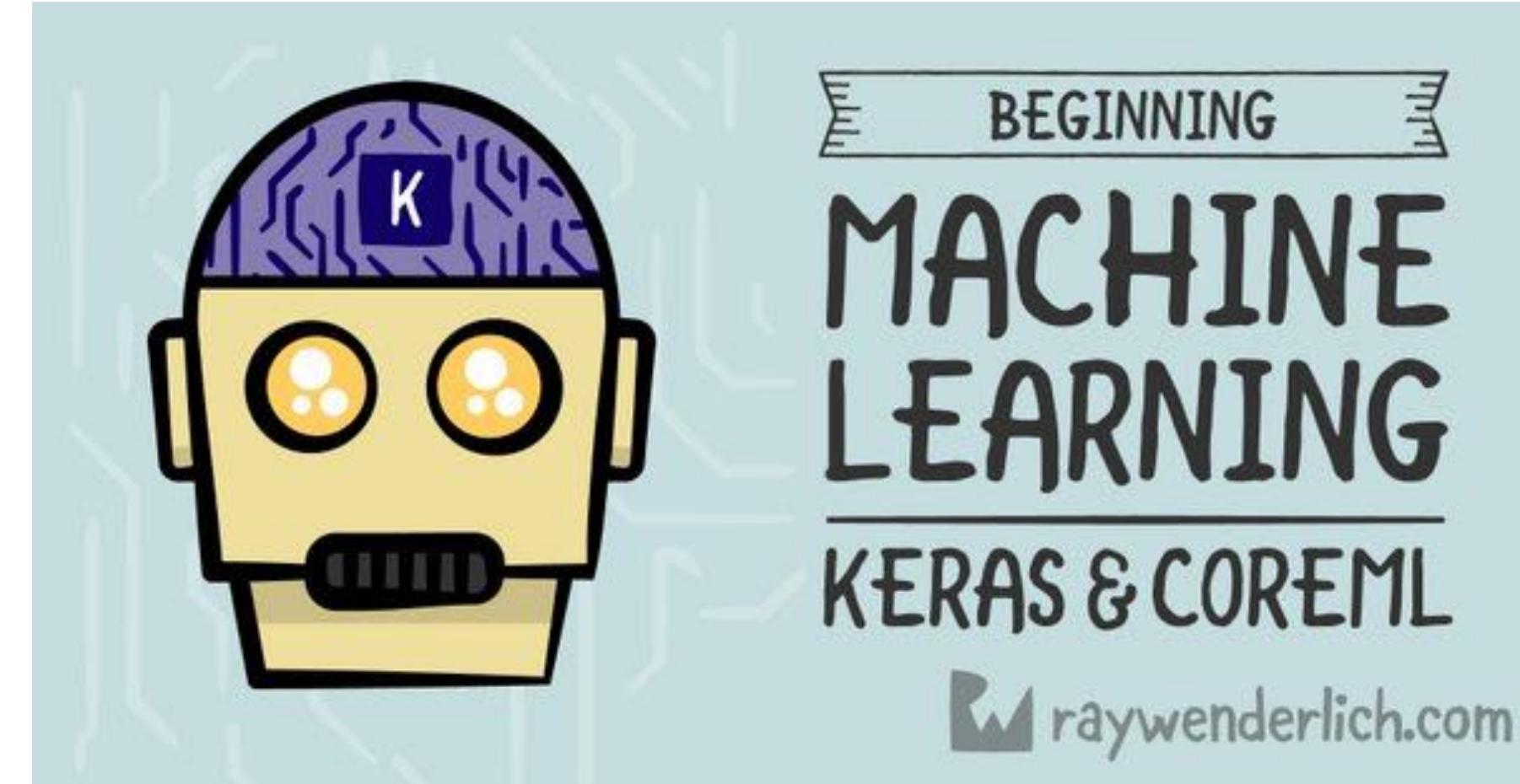


WATCHOS
CORE
BLUETOOTH
TUTORIAL
 raywenderlich.com



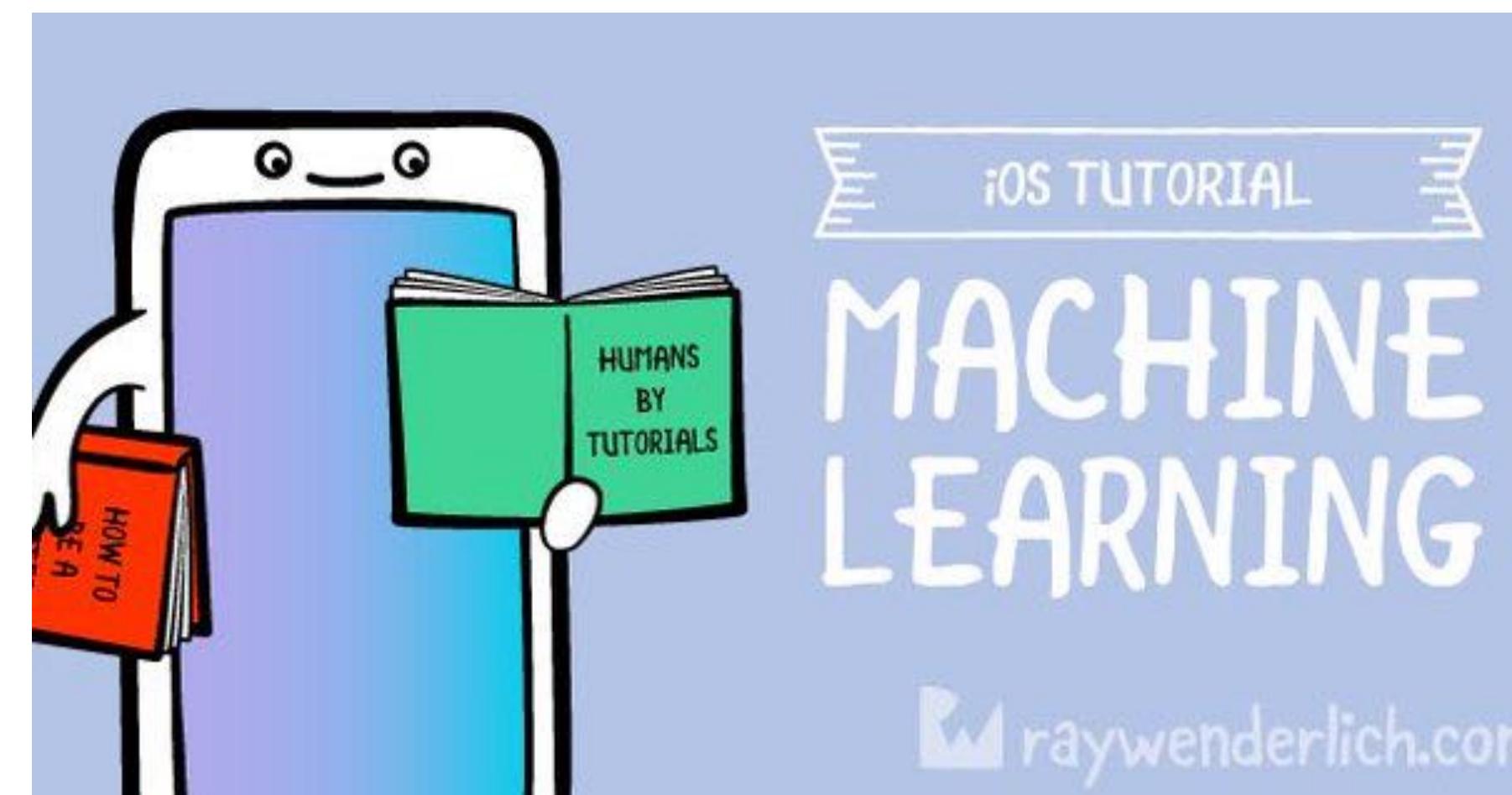
WATCHOS 4
TUTORIAL
PART 1
GETTING STARTED

raywenderlich.com



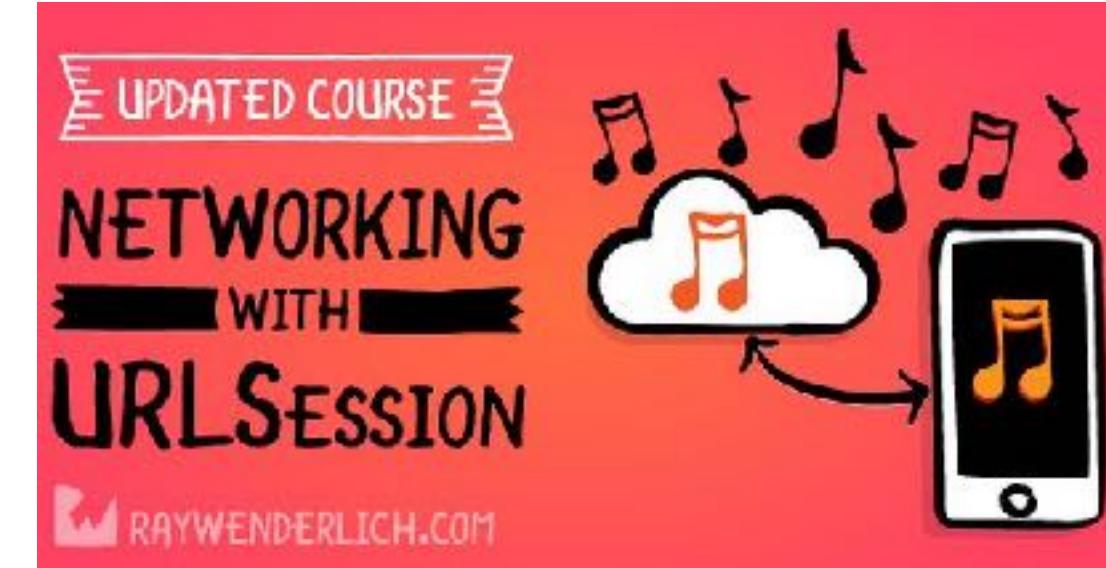
BEGINNING
**MACHINE
LEARNING**
KERAS & COREML

raywenderlich.com



iOS TUTORIAL
**MACHINE
LEARNING**

raywenderlich.com



UPDATED COURSE
**NETWORKING
WITH
URLSESSION**

raywenderlich.com



HOW TO
**DEVELOP
& DESIGN
FOR iPhone X**

raywenderlich.com



MAPKIT
GETTING STARTED

raywenderlich.com



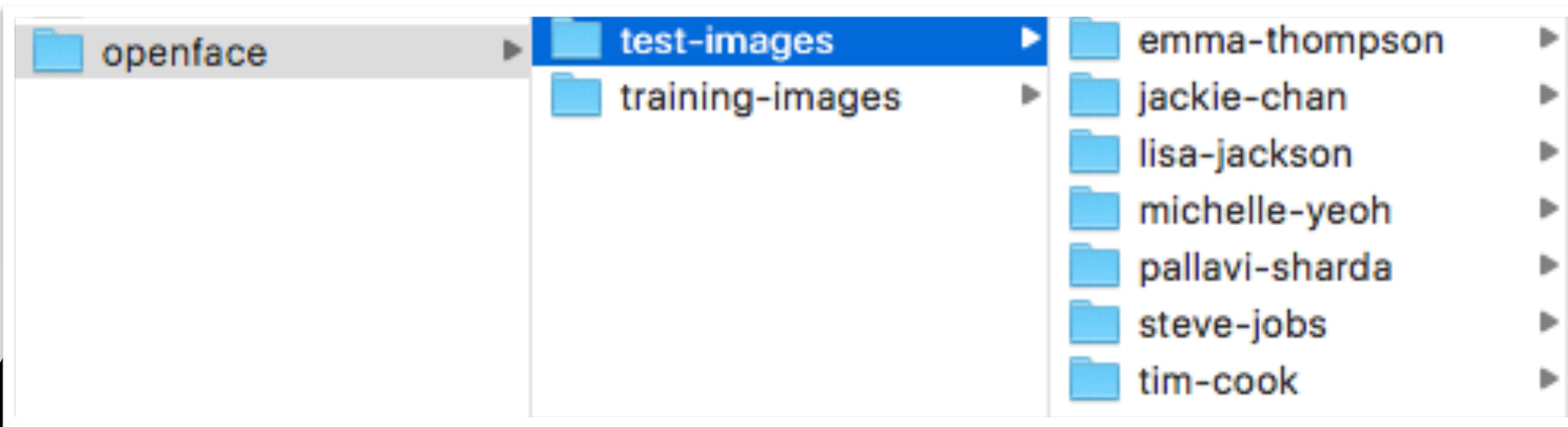
WORKSHOP OVERVIEW

- ⚙️ **Demo 1:** Vision
- ⚙️ **Demo 2:** Face Recognition
- ⚙️ **Demo 3:** Recognize *my* face!



DOCKER & OPENFACE

```
[Audreys-MBP-4:~ amt1$ docker images]
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
<none>              <none>   e478c26739f8  2 months ago  510MB
python               2.7-slim  b0259cf63993  4 months ago  138MB
bamos/openface       latest   7e537bef9341  9 months ago  2.54GB
Audreys-MBP-4:~ amt1$
```



WHAT IS MACHINE LEARNING?

In 1959, Arthur Samuel, a pioneer in the field of machine learning (ML) defined it as the "field of study that gives computers the ability to learn without being explicitly programmed".



Will Wilson @WAWilsonIV · Oct 31

"What's the **difference** between **AI** and **ML**?"

"It's **AI** when you're raising money, it's **ML** when you're trying to hire people."



65



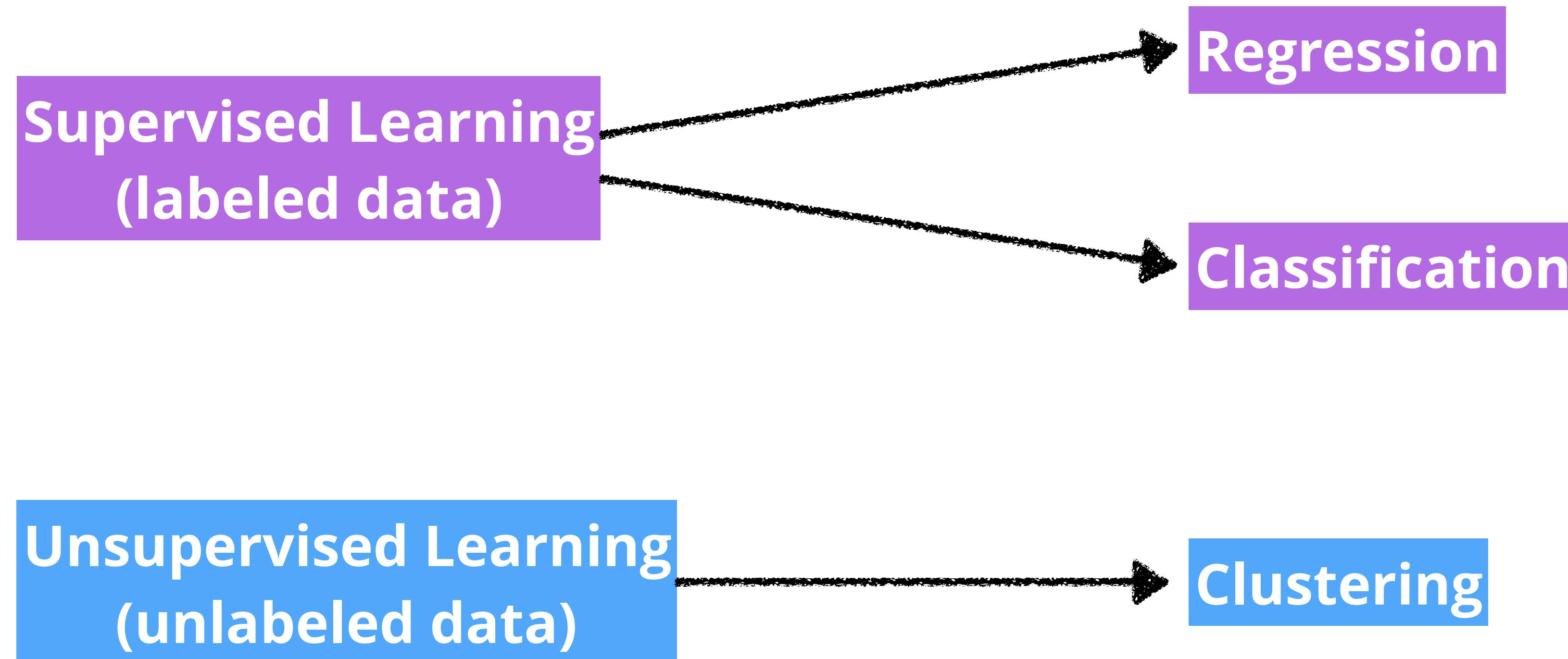
4.6K



9.5K



WHAT IS MACHINE LEARNING?

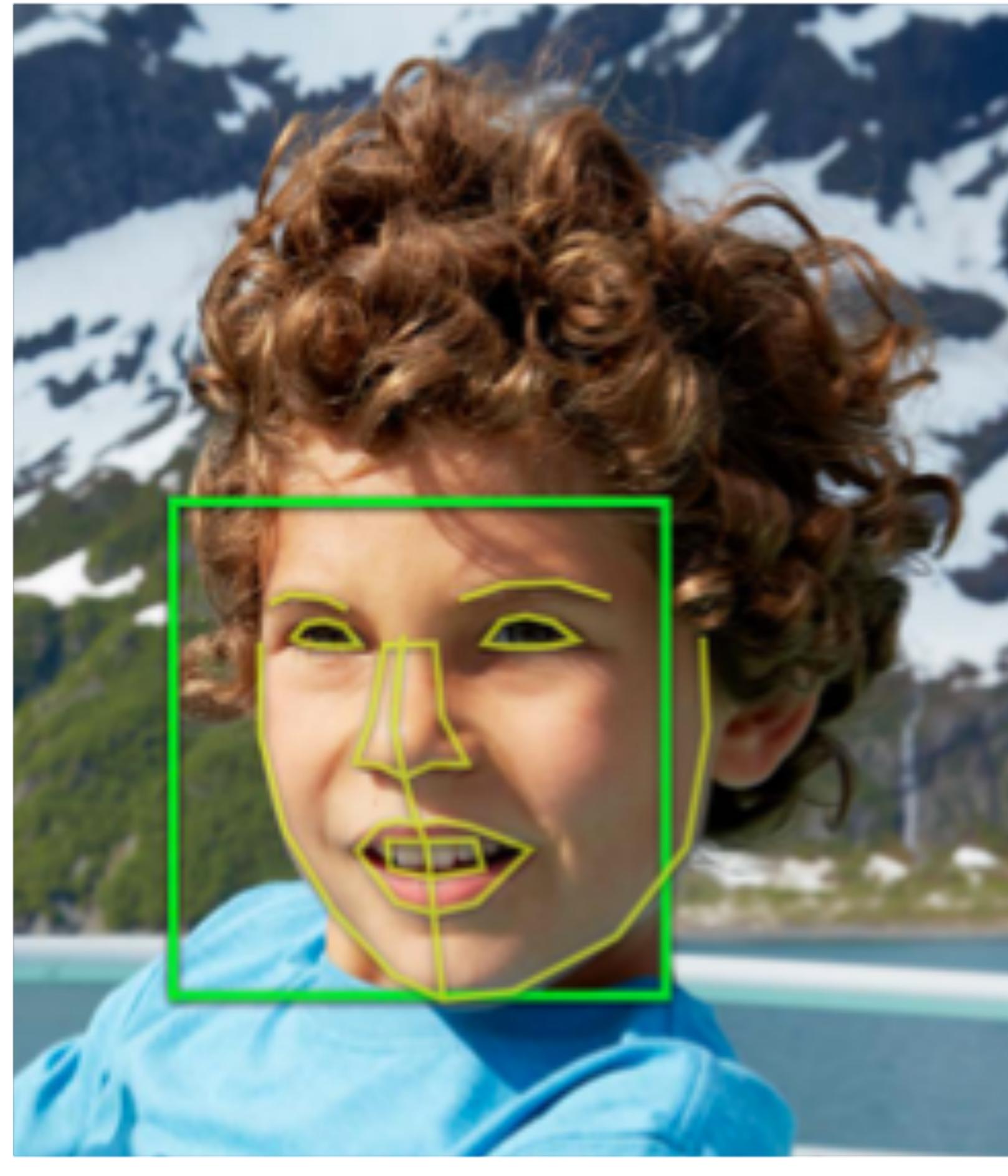
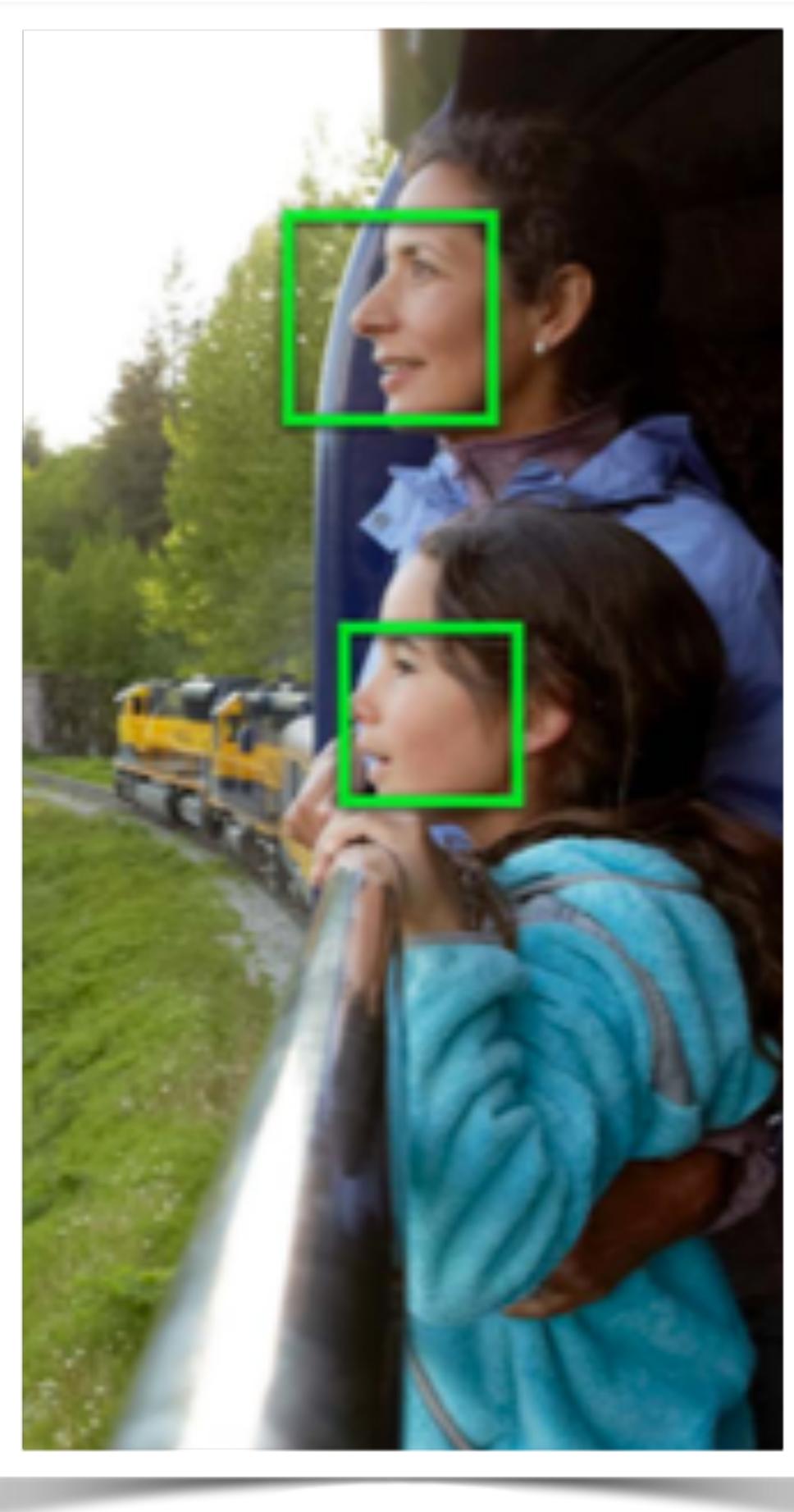


APPLICATION AREAS

- ⚙️ Computational Biology
- ⚙️ Energy Production
- ⚙️ Natural Language Processing
- ⚙️ Computational Finance
- ⚙️ Predictive maintenance
- ⚙️ Computer Vision



VISION FRAMEWORK



VNIMAGEREQUESTHANDLER



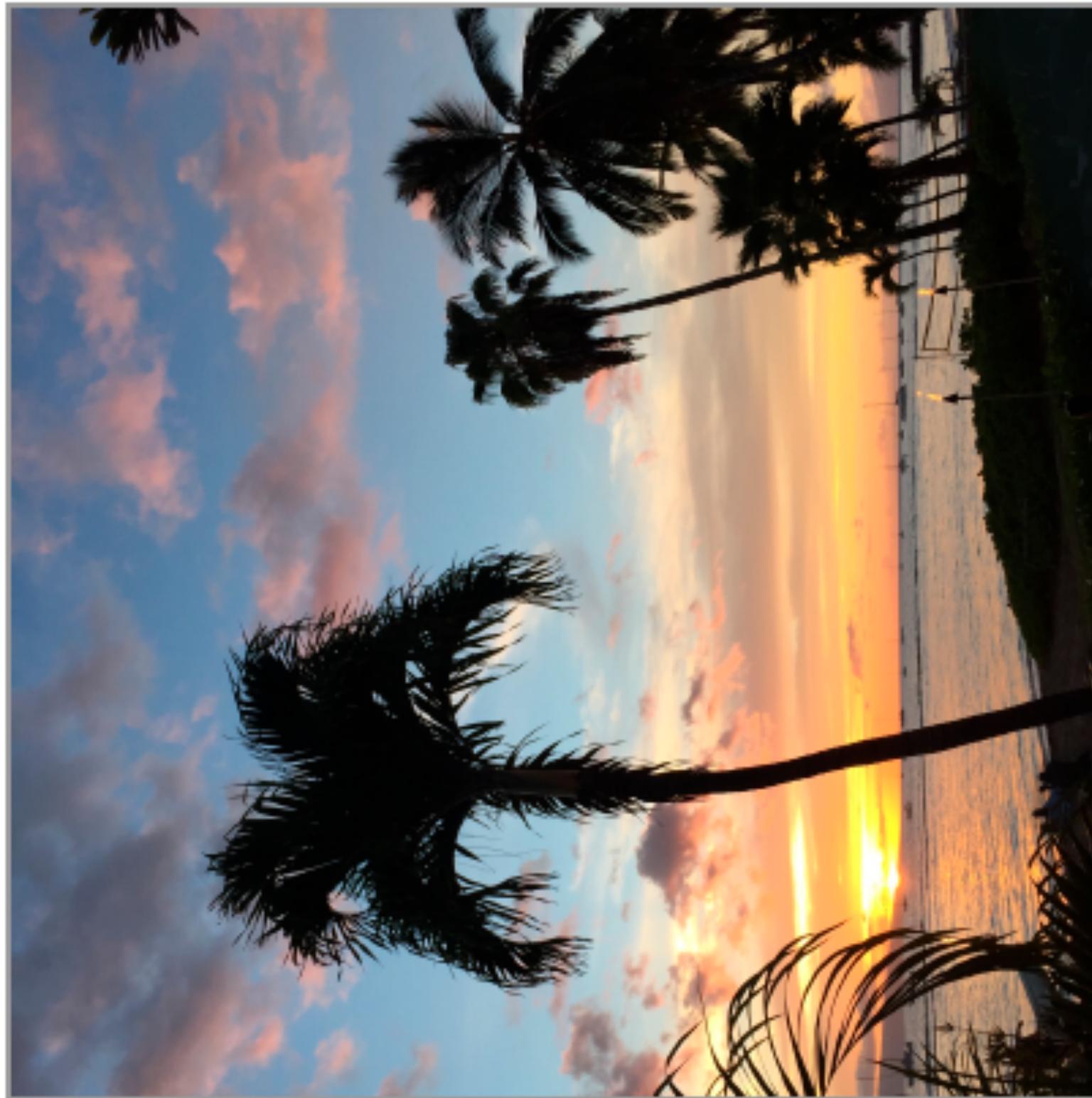
```
let imageHandler = VNImageRequestHandler(cgImage: cgImage,  
orientation: orientation)  
let faceRequest = VNDetectFaceRectanglesRequest(  
completionHandler: handleFaces)  
try? imageHandler.perform([faceRequest])
```

VNSEQUENCEREQUESTHANDLER

```
let sequenceHandler = VNSequenceRequestHandler()  
let trackRequest = VNTrackObjectRequest(  
    detectedObjectObservation: lastObservation,  
    completionHandler: handleFaces)  
try? sequenceHandler.perform([trackRequest], on: pixelBuffer,  
    orientation: exifOrientation)
```



ORIENTATION



Uncorrected image



Intended display



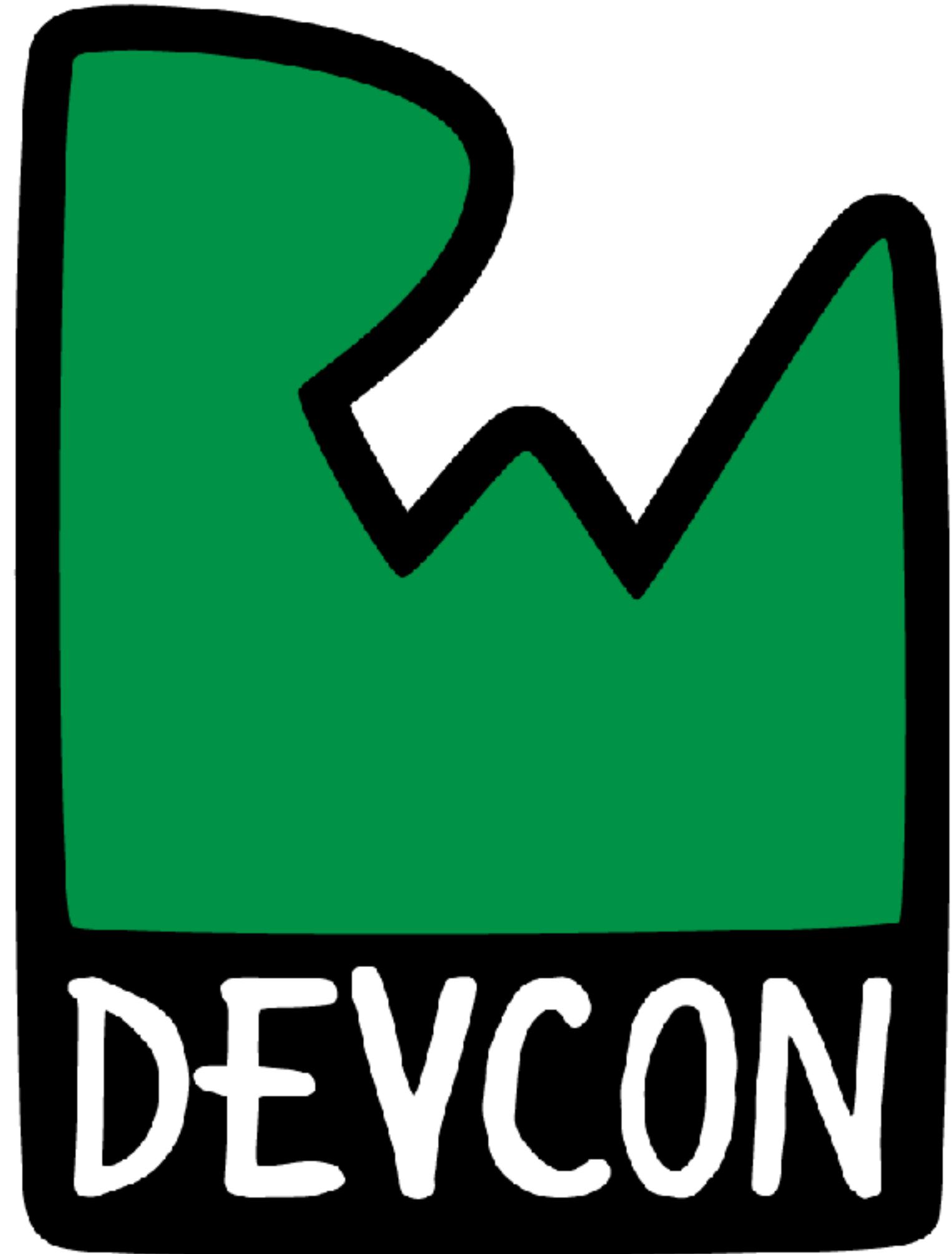


R
W

DEMO 1



Workshop: CoreML + Vision



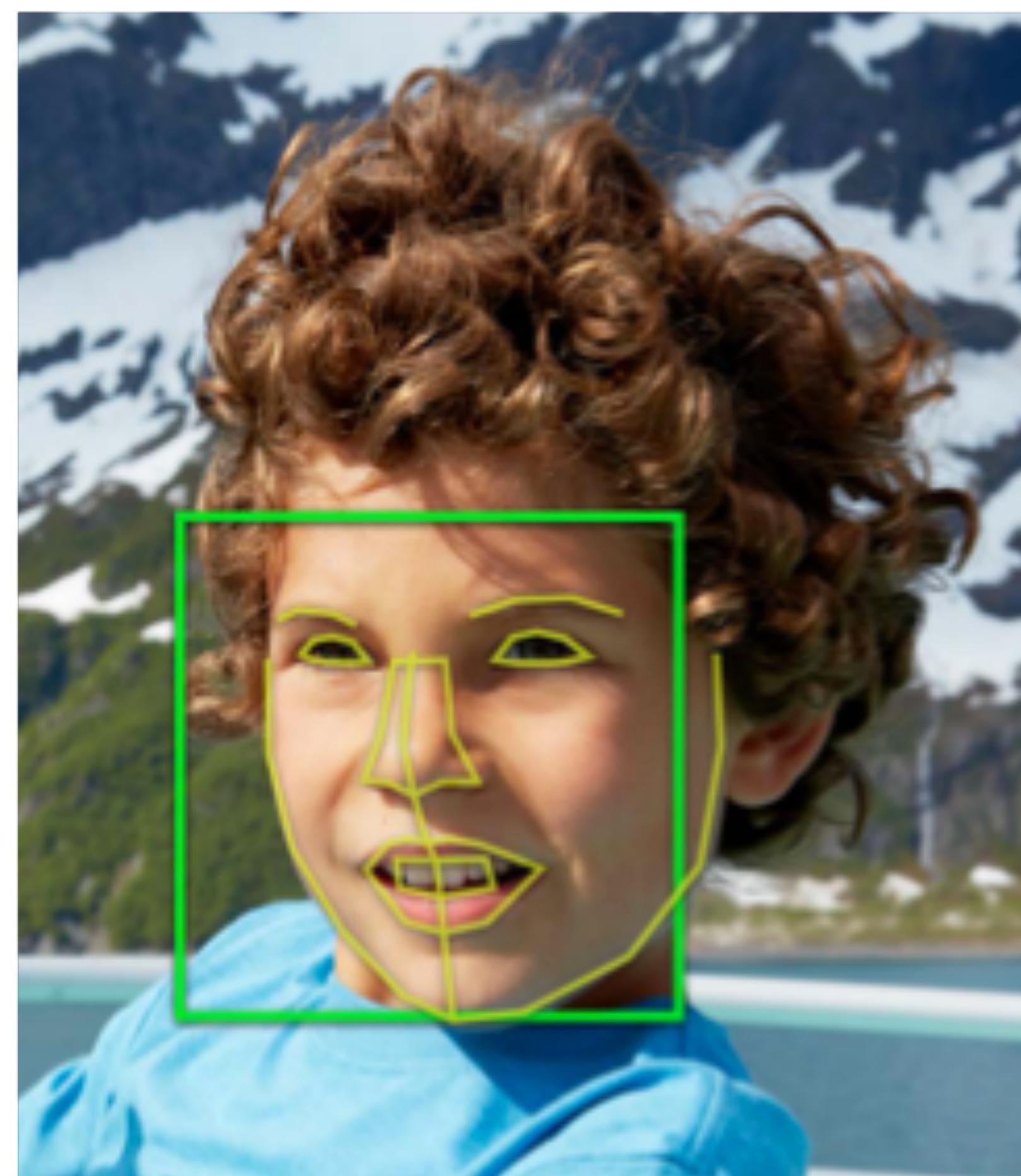
FACE RECOGNITION

FIND FACES



R
W

TRANSFORM WITH FACIAL LANDMARKS



ENCODE FACE: EMBEDDING

Input Image



128 Measurements Generated from Image

| | | | |
|---------------------|---------------------|--------------------|----------------------|
| 0.097496084868908 | 0.045223236083984 | -0.1281466782093 | 0.032084941864014 |
| 0.12529824674129 | 0.060309179127216 | 0.17521631717682 | 0.020976085215807 |
| 0.030809439718723 | -0.01981477253139 | 0.10801389068365 | -0.00052163278451189 |
| 0.036050599068403 | 0.065554238855839 | 0.0731306001544 | -0.1318951100111 |
| -0.097486883401871 | 0.1226262897253 | -0.029626874253154 | -0.0059557510539889 |
| -0.0066401711665094 | 0.036750309169292 | -0.15958009660244 | 0.043374512344599 |
| -0.14131525158882 | 0.14114324748516 | -0.031351584941149 | -0.053343612700701 |
| -0.048540540039539 | -0.061901587992907 | -0.15042643249035 | 0.078198105096817 |
| -0.12567175924778 | -0.10568545013666 | -0.12728653848171 | -0.076289616525173 |
| -0.061418771743774 | -0.074287034571171 | -0.065365232527256 | 0.12369467318058 |
| 0.046741496771574 | 0.0061761881224811 | 0.14746543765068 | 0.056418422609568 |
| -0.12113650143147 | -0.21055991947651 | 0.0041091227903962 | 0.089727647602558 |
| 0.061606746166945 | 0.11345765739679 | 0.021352224051952 | -0.0085843298584223 |
| 0.061989940702915 | 0.19372203946114 | -0.086726233363152 | -0.022388197481632 |
| 0.10904195904732 | 0.084853030741215 | 0.09463594853878 | 0.020696049556136 |
| -0.019414527341723 | 0.0064811296761036 | 0.21180312335491 | -0.050584398210049 |
| 0.15245945751667 | -0.16582328081131 | -0.035577941685915 | -0.072376452386379 |
| -0.12216668576002 | -0.0072777755558491 | -0.036901291459799 | -0.034365277737379 |
| 0.083934605121613 | -0.059730969369411 | -0.070026844739914 | -0.045013956725597 |
| 0.087945111095905 | 0.11478432267904 | -0.089621491730213 | -0.013955107890069 |
| -0.021407851949334 | 0.14841195940971 | 0.078333757817745 | -0.17898085713387 |
| -0.018298890441656 | 0.049525424838066 | 0.13227833807468 | -0.072600327432156 |
| -0.011014151386917 | -0.051016297191381 | -0.14132921397686 | 0.0050511928275228 |
| 0.0093679334968328 | -0.062812767922878 | -0.13407498598099 | -0.014829395338893 |
| 0.058139257133007 | 0.0048638740554452 | -0.039491076022387 | -0.043765489012003 |
| -0.024210374802351 | -0.11443792283535 | 0.071997955441475 | -0.012062266469002 |
| -0.057223934680223 | 0.014683869667351 | 0.05228154733777 | 0.012774495407939 |
| 0.023535015061498 | -0.081752359867096 | -0.031709920614958 | 0.069833360612392 |
| -0.0098039731383324 | 0.037022035568953 | 0.11009479314089 | 0.11638788878918 |
| 0.020220354199409 | 0.12788131833076 | 0.18632389605045 | -0.015336792916059 |
| 0.0040337680839002 | -0.094398014247417 | -0.11768248677254 | 0.10281457751989 |
| 0.051597066223621 | -0.10034311562777 | -0.040977258235216 | -0.082041338086128 |

FIND CLOSEST MATCH

| | | | |
|---------------------|---------------------|--------------------|----------------------|
| 0.097496084868908 | 0.045223236083984 | -0.1281466782093 | 0.032084941864014 |
| 0.12529824674129 | 0.060309179127216 | 0.175216317176882 | 0.020976085215807 |
| 0.030809439718723 | -0.01981477253139 | 0.10801389068365 | -0.00052163278451189 |
| 0.038050509068403 | 0.065554238855839 | 0.0731306001544 | -0.1318951100111 |
| -0.097486883401871 | 0.1226262897253 | -0.029626874253154 | -0.0059557510539889 |
| -0.0066401711665094 | 0.036750309169292 | -0.15958009660244 | 0.043374512344599 |
| -0.14131525158882 | 0.14114324748516 | -0.031351584941149 | -0.053343612700701 |
| -0.048540540039539 | -0.061901587992907 | -0.15042643248035 | 0.078198105096817 |
| -0.12567175924778 | -0.10568545013866 | -0.12728653848171 | -0.076289616525173 |
| -0.061418771743774 | -0.074287034571171 | -0.065365232527256 | 0.12389467318058 |
| 0.046741496771574 | 0.00617618812248111 | 0.14746543765068 | 0.056418422609668 |
| -0.12113850143147 | -0.21058991947851 | 0.0041091227903962 | 0.089727647602558 |
| 0.061806748166945 | 0.11345765739679 | 0.021352224051952 | -0.0085843298584223 |
| 0.06186748166945 | 0.19372203948114 | -0.086726233363152 | -0.022388197481632 |
| 0.10904195904732 | 0.084853030741215 | 0.09463594853878 | 0.02096049556136 |
| -0.019414527341723 | 0.0064811296761036 | 0.21180312335491 | -0.050584398210049 |
| 0.152454957521667 | -0.16582328081131 | -0.035577941685915 | -0.072376452386379 |
| -0.12216668576002 | -0.007277755558491 | -0.036901281458799 | -0.03465224051952 |
| 0.083934605121613 | -0.059730969369411 | -0.070026844739914 | -0.045013956725597 |
| 0.087945111095905 | 0.11478432267904 | -0.089621491730213 | -0.013955107890069 |
| -0.021407851948334 | 0.14841195940971 | 0.078333757817745 | -0.17898085713387 |
| -0.018298890441656 | 0.049525424838066 | 0.13227833807468 | -0.07260327432156 |
| -0.011014151386917 | -0.051016297191381 | -0.14132921397686 | 0.0050511928275228 |
| 0.0093679334968328 | -0.062812767922878 | -0.13407438698099 | -0.014829395338893 |
| 0.058139257133007 | 0.0048638740554452 | -0.039491076022387 | -0.043785489012003 |
| -0.024210374802351 | -0.11443792283535 | 0.071997955441475 | -0.012062266469002 |
| -0.05723934680223 | 0.014683869687351 | 0.05228154733777 | 0.012774495407939 |
| 0.023535015061498 | -0.081752359867098 | -0.031709920614958 | 0.089833380612392 |
| -0.0098039731383324 | 0.037022035568953 | 0.11009479314089 | 0.11638788878918 |
| 0.020220354199409 | 0.12788131833076 | 0.18632389605045 | -0.015336792916059 |
| 0.0040337680839002 | -0.094398014247417 | -0.11768248677254 | 0.10281457751989 |
| 0.051597066223621 | -0.10034311562777 | -0.040977258235216 | -0.082041338086128 |



| | | | |
|---------------------|---------------------|--------------------|----------------------|
| 0.097496084868908 | 0.045223236083984 | -0.1281466782093 | 0.032084941864014 |
| 0.12529824674129 | 0.060309179127216 | 0.175216317176882 | 0.020976085215807 |
| 0.030809439718723 | -0.01981477253139 | 0.10801389068365 | -0.00052163278451189 |
| 0.038050509068403 | 0.065554238855839 | 0.0731306001544 | -0.1318951100111 |
| -0.097486883401871 | 0.1226262897253 | -0.029626874253154 | -0.0059557510539889 |
| -0.0066401711665094 | 0.036750309169292 | -0.15958009660244 | 0.043374512344599 |
| -0.14131525158882 | 0.14114324748516 | -0.031351584941149 | -0.053343612700701 |
| -0.048540540039539 | -0.061901587992907 | -0.15042643248035 | 0.078198105096817 |
| -0.12567175924778 | -0.10568545013866 | -0.12728653848171 | -0.076289616525173 |
| -0.061418771743774 | -0.074287034571171 | -0.065365232527256 | 0.12389467318058 |
| 0.046741496771574 | 0.00617618812248111 | 0.14746543765068 | 0.056418422609668 |
| -0.12113850143147 | -0.21058991947851 | 0.0041091227903962 | 0.089727647602558 |
| 0.083934605121613 | -0.059730969369411 | -0.070026844739914 | -0.045013956725597 |
| 0.087945111095905 | 0.11478432267904 | -0.089621491730213 | -0.013955107890069 |
| -0.021407851948334 | 0.014683869687351 | 0.05228154733777 | 0.012774495407939 |
| -0.018298890441656 | 0.049525424838066 | 0.13227833807468 | -0.07260327432156 |
| -0.011014151386917 | -0.051016297191381 | -0.14132921397686 | 0.0050511928275228 |
| 0.0093679334968328 | -0.062812767922878 | -0.13407438698099 | -0.014829395338893 |
| 0.058139257133007 | 0.0048638740554452 | -0.039491076022387 | -0.043785489012003 |
| -0.024210374802351 | -0.11443792283535 | 0.071997955441475 | -0.012062266469002 |
| -0.05723934680223 | 0.014683869687351 | 0.05228154733777 | 0.012774495407939 |
| 0.023535015061498 | -0.081752359867098 | -0.031709920614958 | 0.089833380612392 |
| -0.0098039731383324 | 0.037022035568953 | 0.11009479314089 | 0.11638788878918 |
| 0.020220354199409 | 0.12788131833076 | 0.18632389605045 | -0.015336792916059 |
| 0.0040337680839002 | -0.094398014247417 | -0.11768248677254 | 0.10281457751989 |
| 0.051597066223621 | -0.10034311562777 | -0.040977258235216 | -0.082041338086128 |
| 0.097496084868908 | 0.045223236083984 | -0.1281466782093 | 0.032084941864014 |
| 0.12529824674129 | 0.060309179127216 | 0.175216317176882 | 0.020976085215807 |
| 0.030809439718723 | -0.01981477253139 | 0.10801389068365 | -0.00052163278451189 |
| 0.038050509068403 | 0.065554238855839 | 0.0731306001544 | -0.1318951100111 |
| -0.097486883401871 | 0.1226262897253 | -0.029626874253154 | -0.0059557510539889 |
| -0.0066401711665094 | 0.036750309169292 | -0.15958009660244 | 0.043374512344599 |
| -0.14131525158882 | 0.14114324748516 | -0.031351584941149 | -0.053343612700701 |
| -0.048540540039539 | -0.061901587992907 | -0.15042643248035 | 0.078198105096817 |
| -0.12567175924778 | -0.10568545013866 | -0.12728653848171 | -0.076289616525173 |
| -0.061418771743774 | -0.074287034571171 | -0.065365232527256 | 0.12389467318058 |
| 0.046741496771574 | 0.00617618812248111 | 0.14746543765068 | 0.056418422609668 |
| -0.12113850143147 | -0.21058991947851 | 0.0041091227903962 | 0.089727647602558 |
| 0.083934605121613 | -0.059730969369411 | -0.070026844739914 | -0.045013956725597 |
| 0.087945111095905 | 0.11478432267904 | -0.089621491730213 | -0.013955107890069 |
| -0.021407851948334 | 0.014683869687351 | 0.05228154733777 | 0.012774495407939 |
| -0.018298890441656 | 0.049525424838066 | 0.13227833807468 | -0.07260327432156 |
| -0.011014151386917 | -0.051016297191381 | -0.14132921397686 | 0.0050511928275228 |
| 0.0093679334968328 | -0.062812767922878 | -0.13407438698099 | -0.014829395338893 |
| 0.058139257133007 | 0.0048638740554452 | -0.039491076022387 | -0.043785489012003 |
| -0.024210374802351 | -0.11443792283535 | 0.0719979554414 | |

CORE ML



```
let model = MyModel()  
if let prediction = try? model.prediction(input: MyModelInput) {  
    print(prediction.classLabel)  
}
```



LUMINA

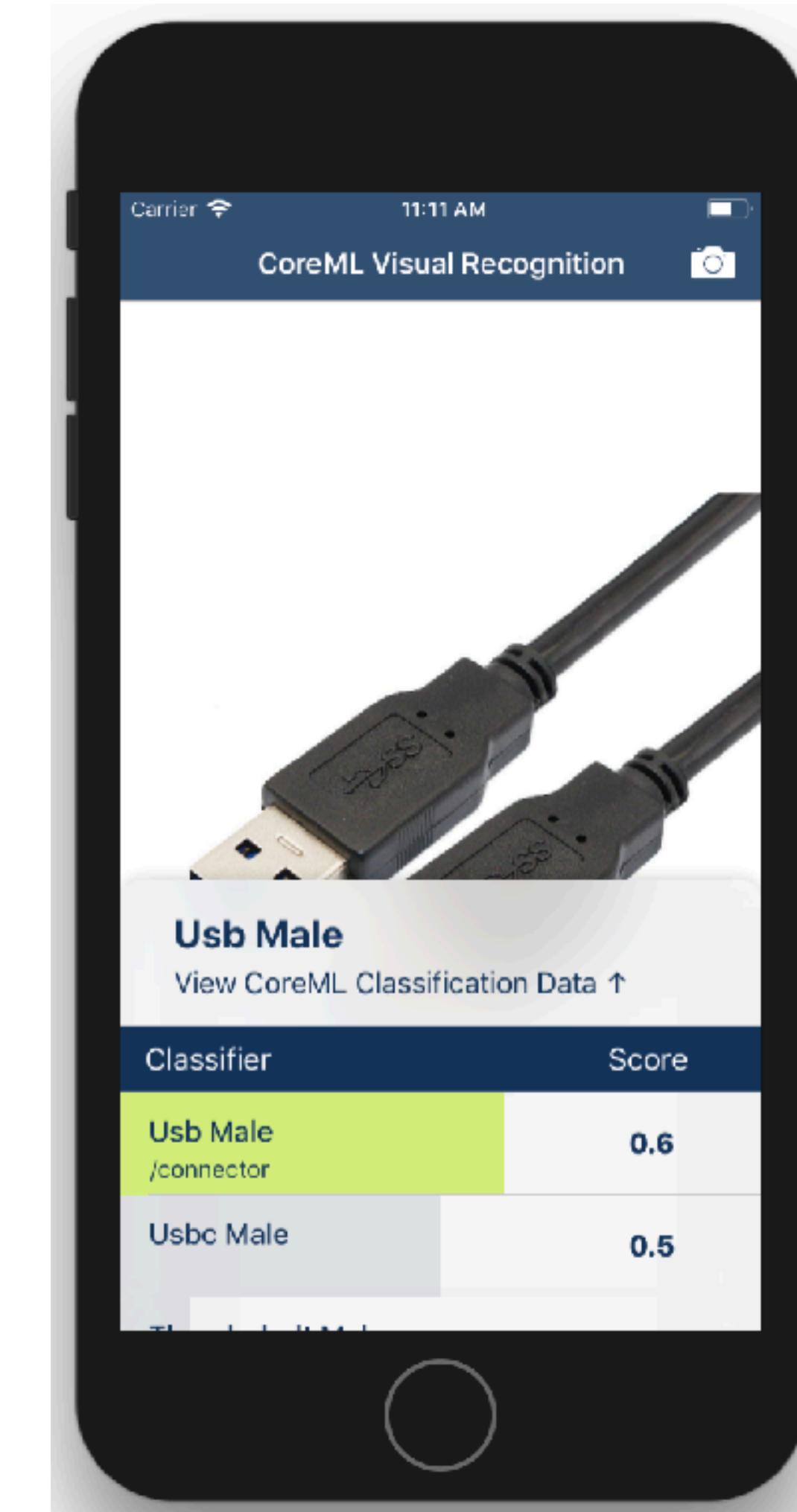
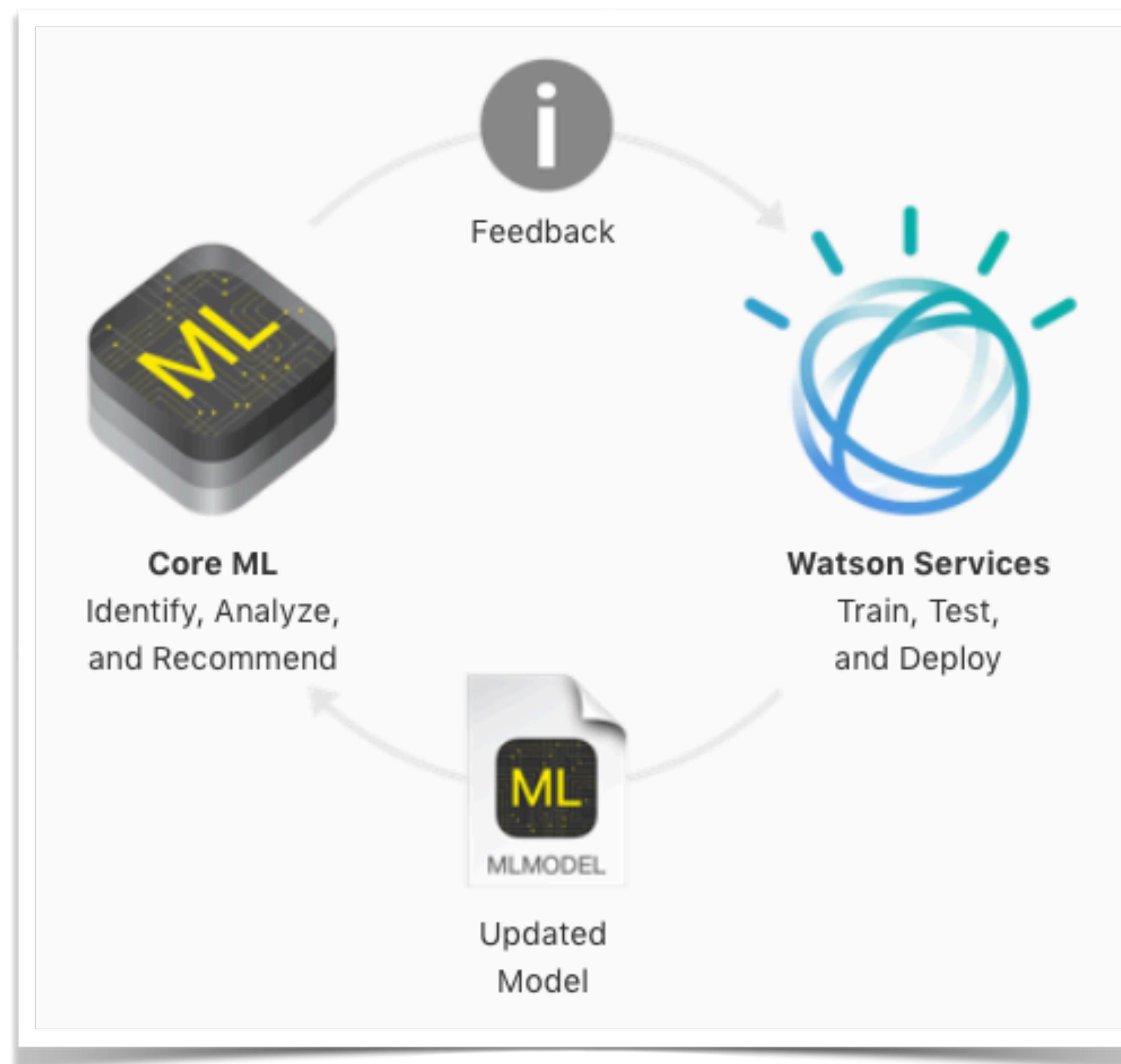


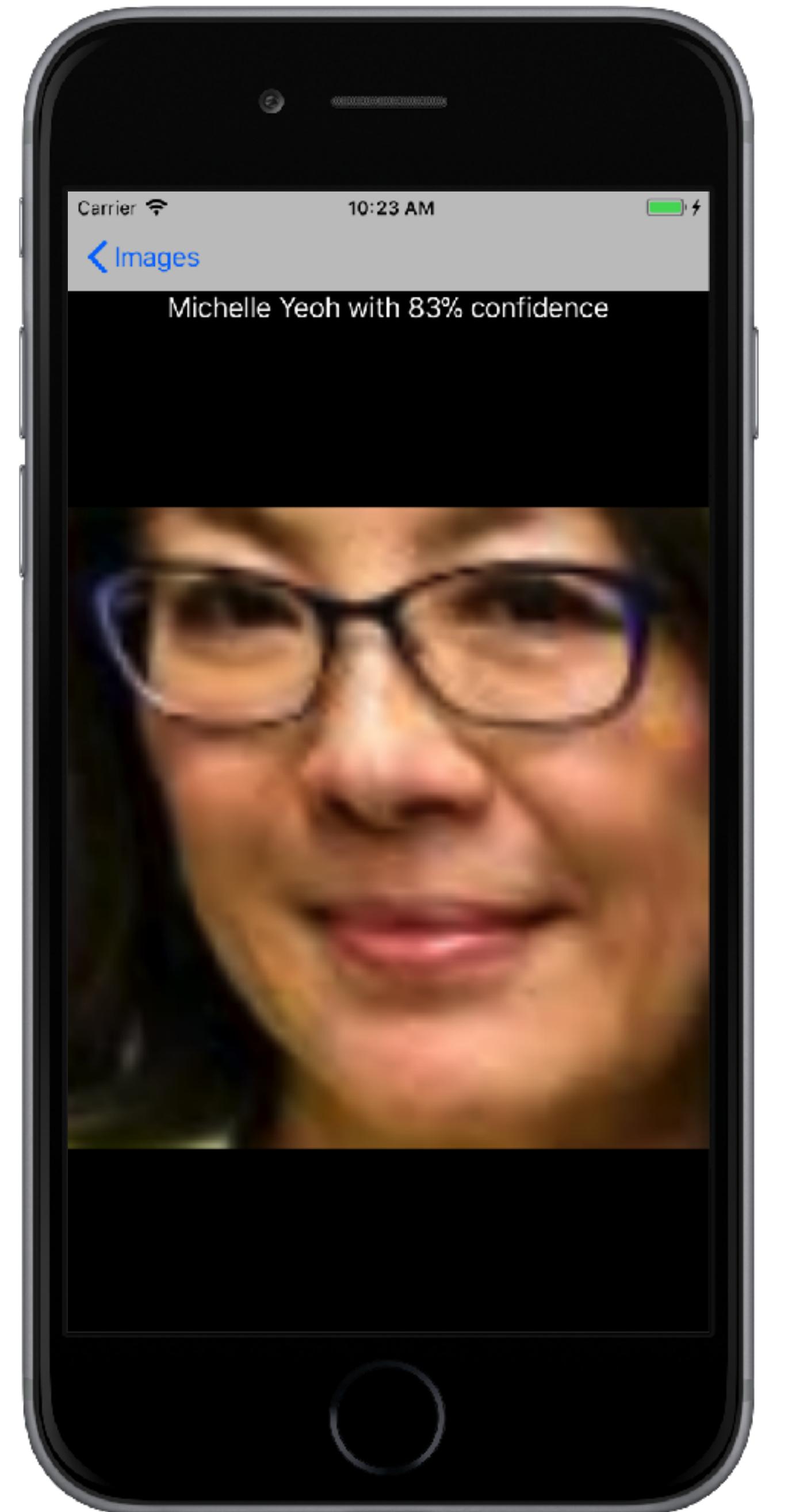
KITURA™



R
W

IBM WATSON SERVICES FOR CORE ML



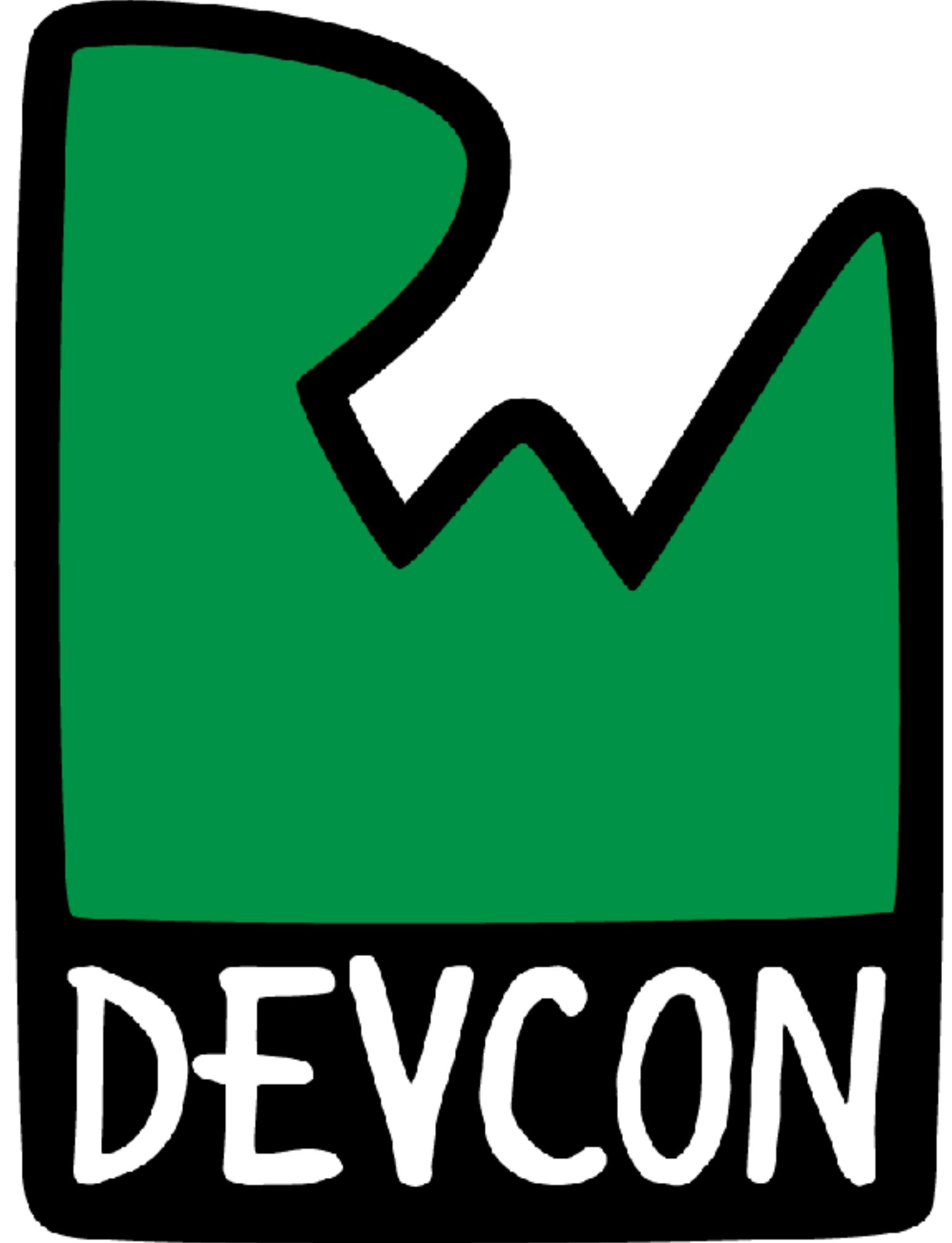


R
W

DEMO 2



Workshop: CoreML + Vision



RECOGNIZE MY FACE!

COLLECT DATA

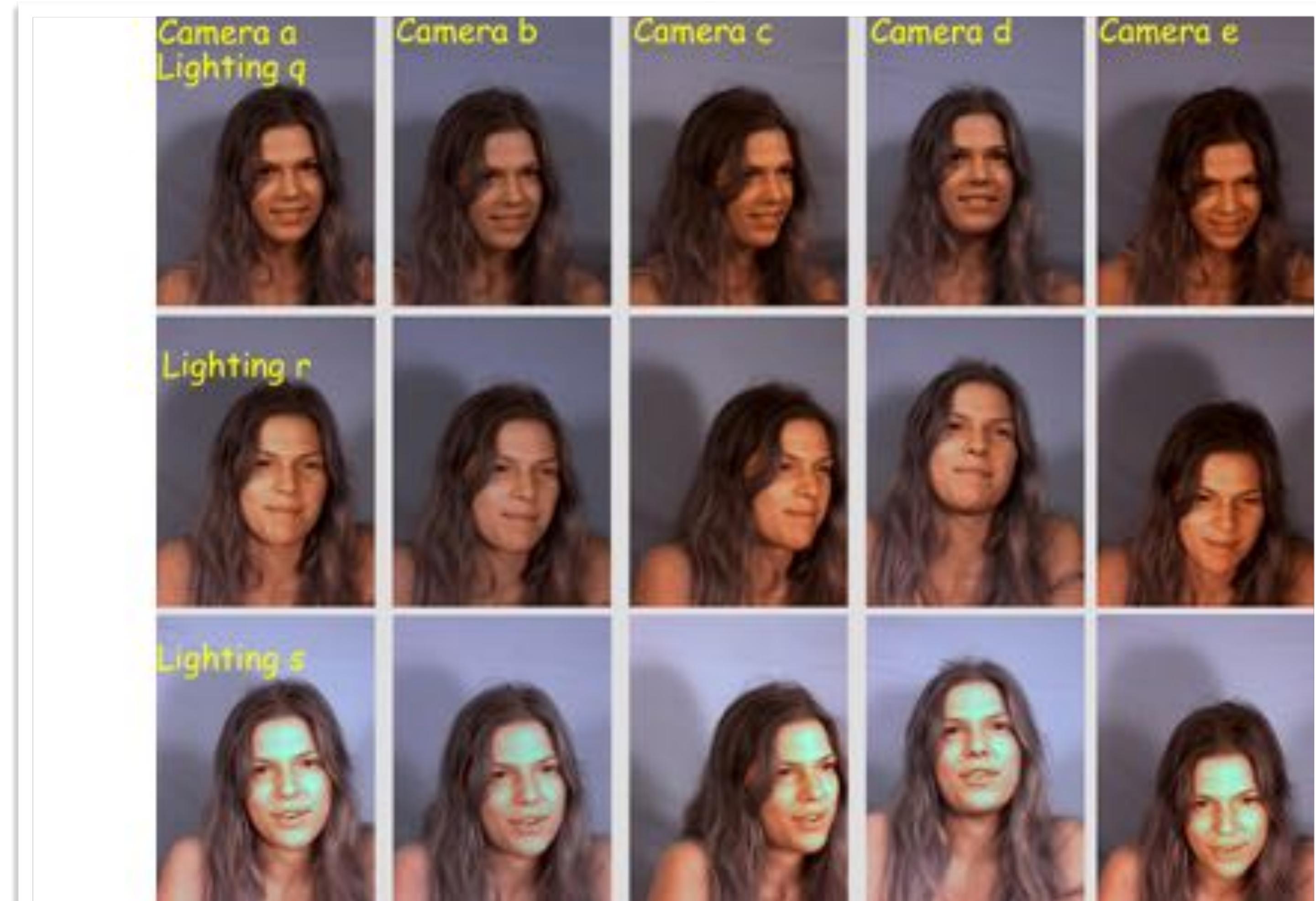


COLLECT DATA: How Much Data?

The amount of data required depends on the complexity of the problem and learning algorithm.



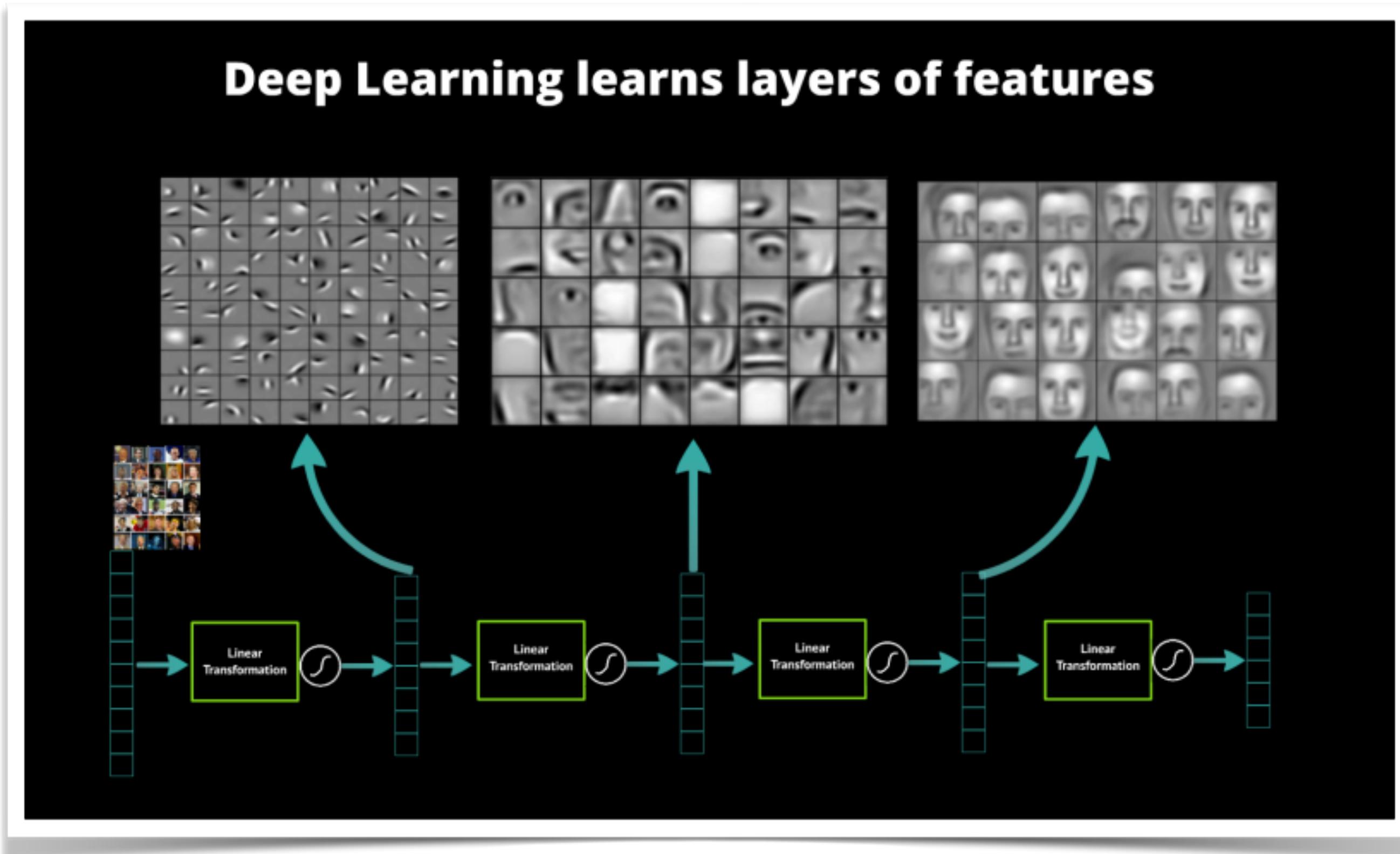
PREPROCESS DATA



R
W

FEATURE SELECTION

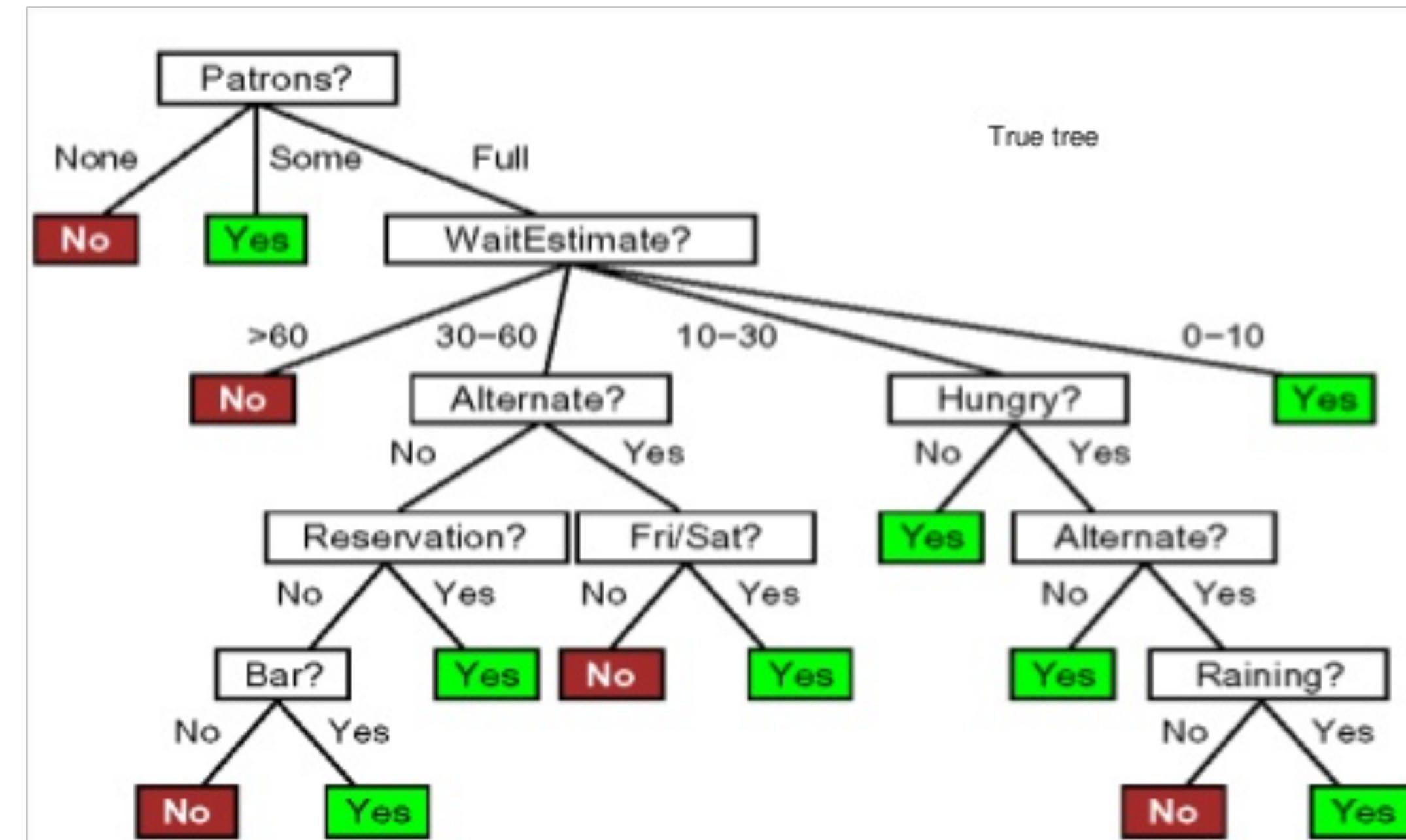
Decide what features from input will be the criteria for analysis.



BUILD & TRAIN MODEL

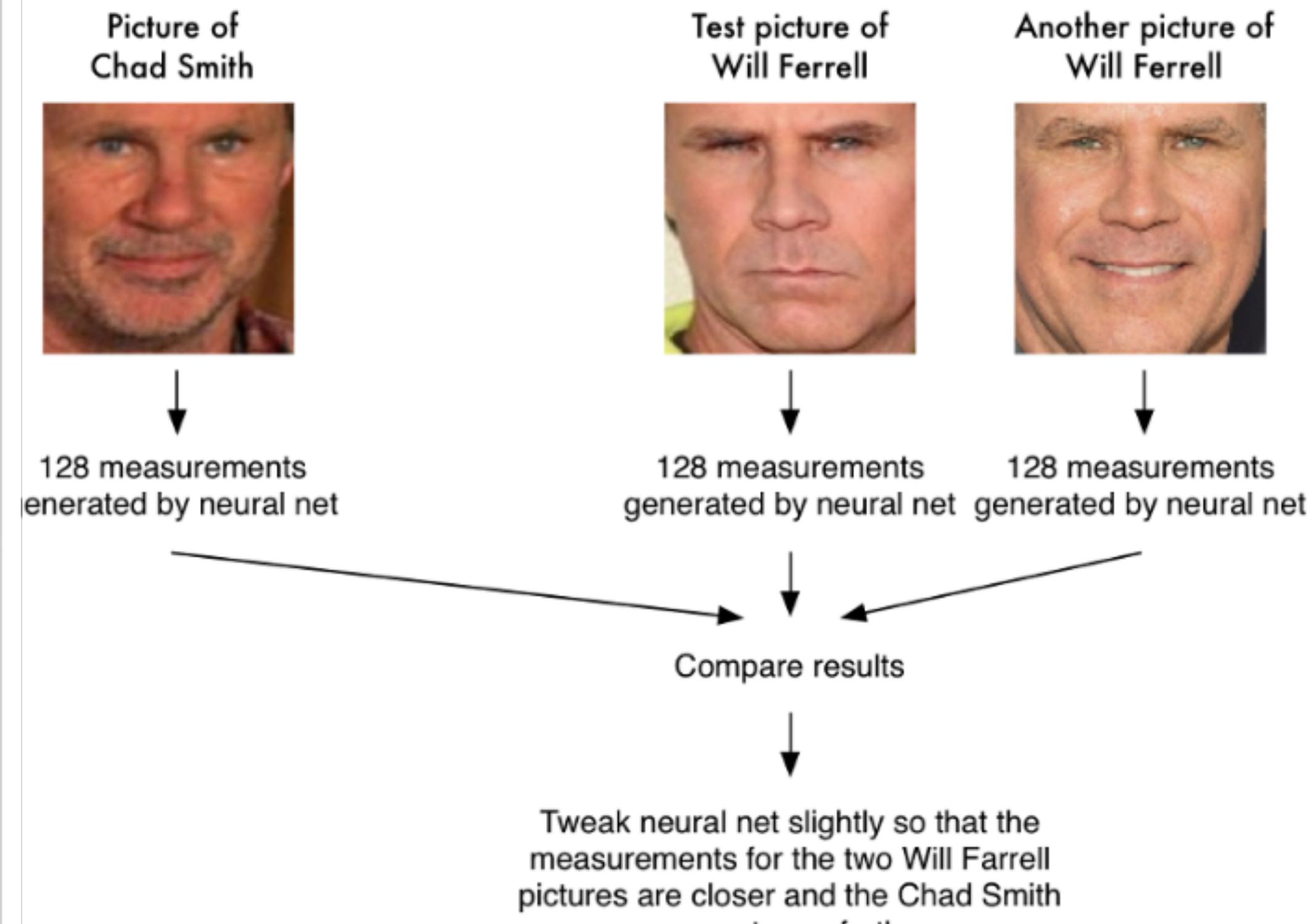
Different models are better suited for solving different problems.

A learned decision tree



BUILD & TRAIN MODEL

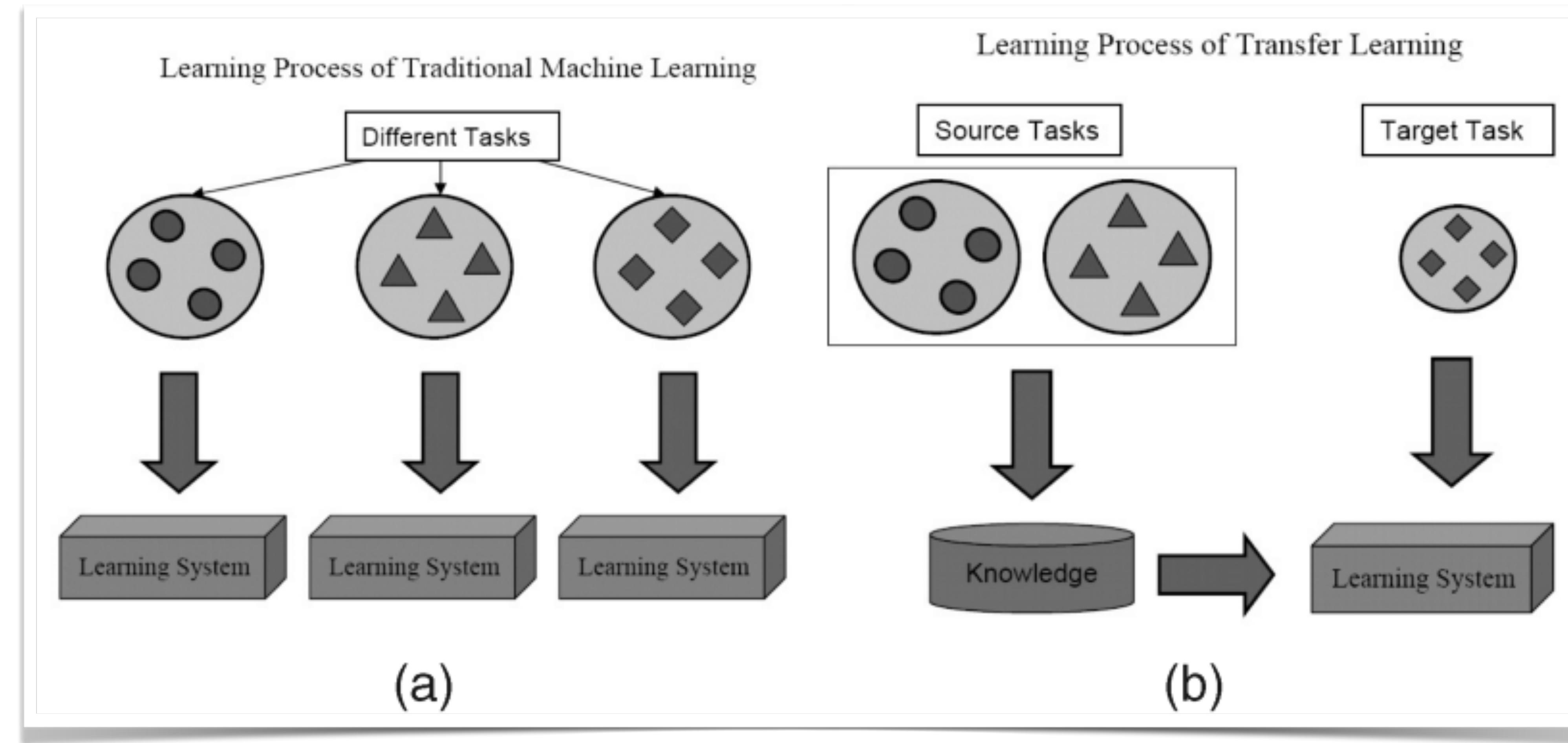
A single 'triplet' training step:



For neural nets, the training process works by repeating the training step millions of time.



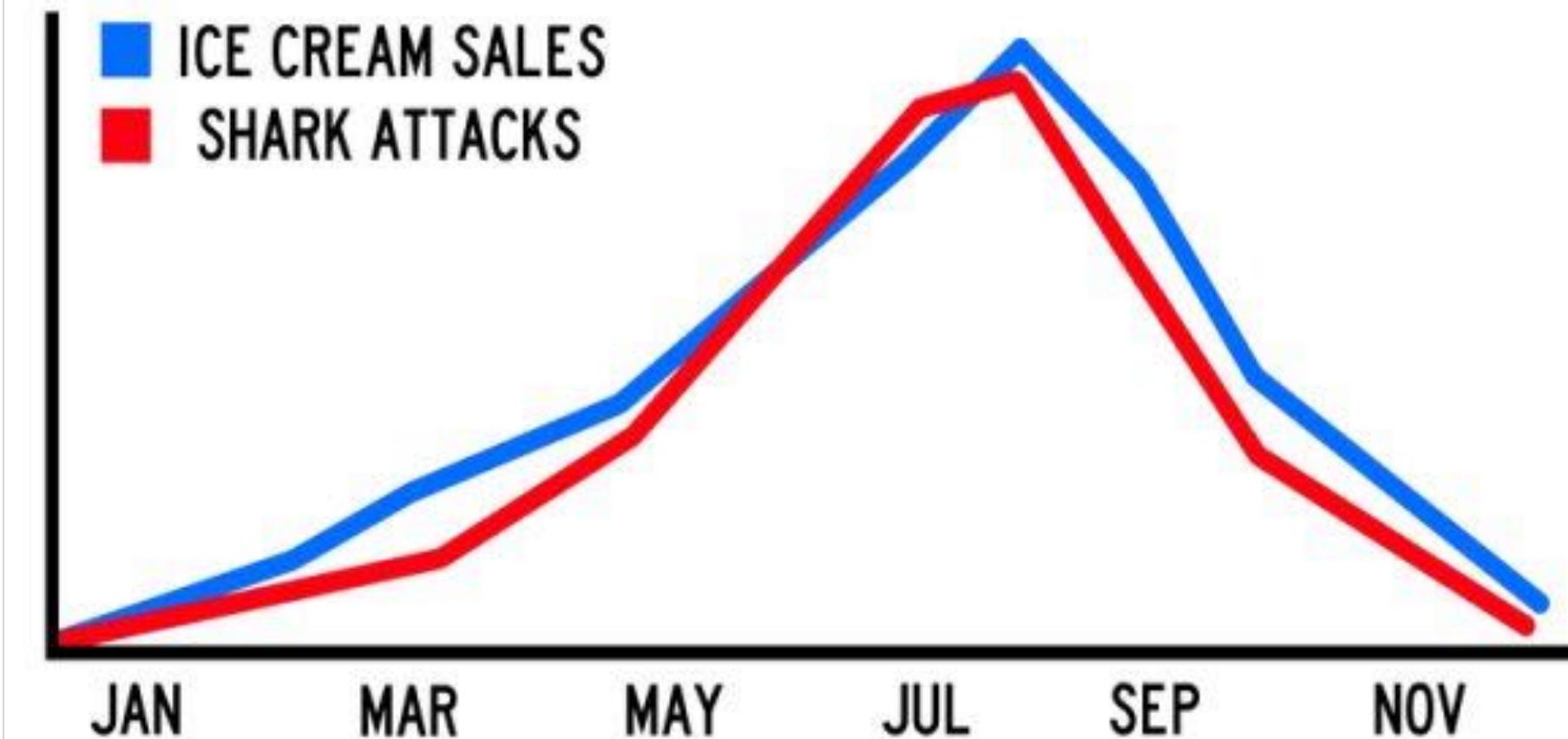
BUILD & TRAIN Model: TRANSFER LEARNING



“WITH GREAT POWER COMES GREAT RESPONSIBILITY”

Building a Model is equivalent to mining multiple correlations from your data.

CORRELATION IS NOT CAUSATION!



Both ice cream sales and shark attacks increase when the weather is hot and sunny, but they are not caused by each other (they are caused by good weather, with lots of people at the beach, both eating ice cream and having a swim in the sea)



TRAINING A MODEL RESPONSIBLY



Don't let any unintended negative biases into your model.

COREMLTOOLS

There are a number of model types that can be converted to CoreML format from popular machine learning tools such as:

- ⚙️ Keras
- ⚙️ Caffe
- ⚙️ scikit-learn
- ⚙️ libsvm
- ⚙️ XGBoost

CoreMLTools, a python package, helps convert models to CoreML format.



COREMLTOOLS

"CoreMLTools is used to convert trained models from popular ML tools into Core ML format (.mlmodel)"



Run an appropriate conversion script, to get your model into CoreML model format(".mlmodel")

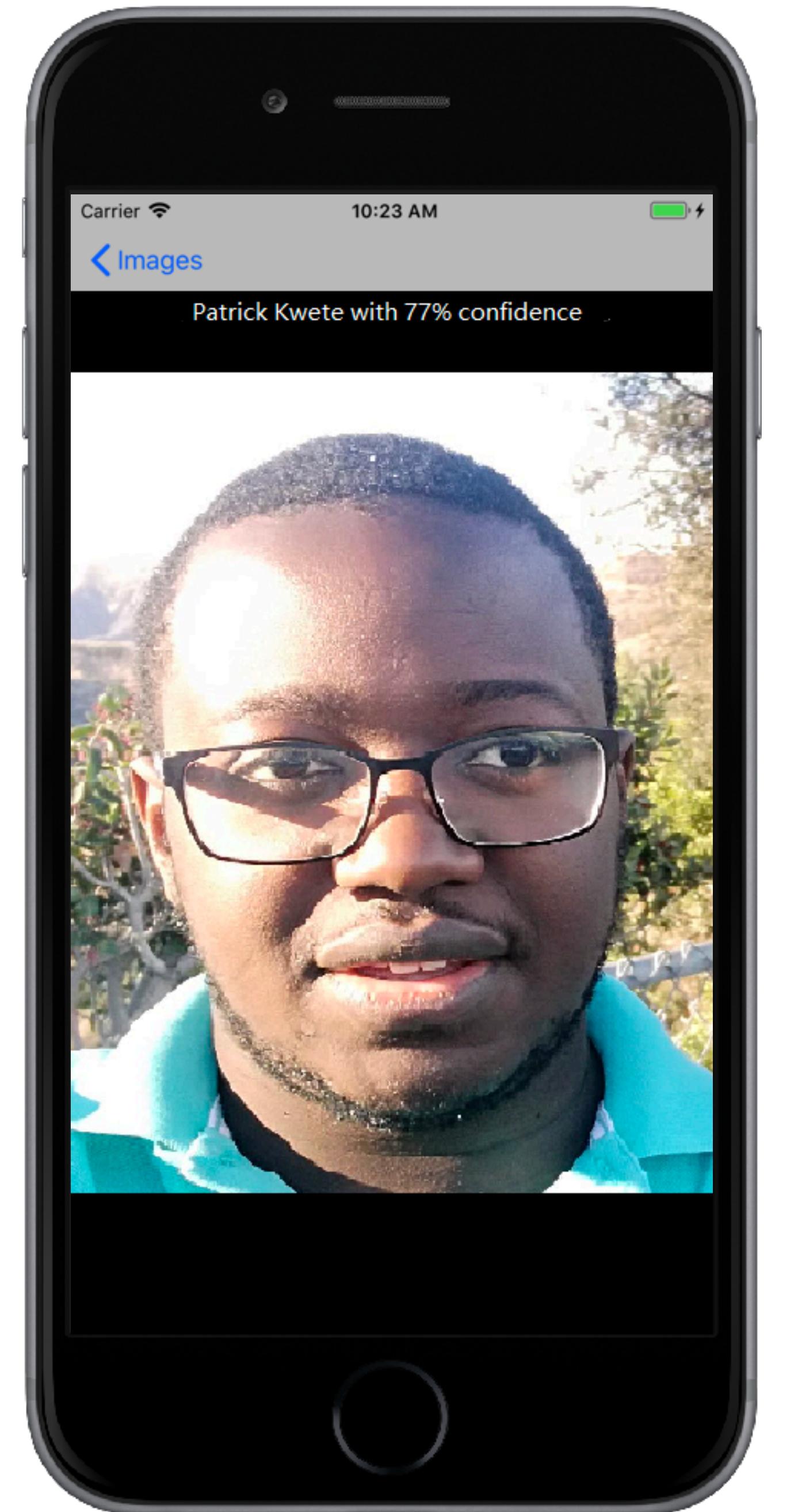


INTEGRATE YOUR COREML MODEL



```
let model = OpenFaceClassifier()  
if let prediction = try? model.prediction(/* parameters */) {  
    print(prediction.identifier)  
}
```



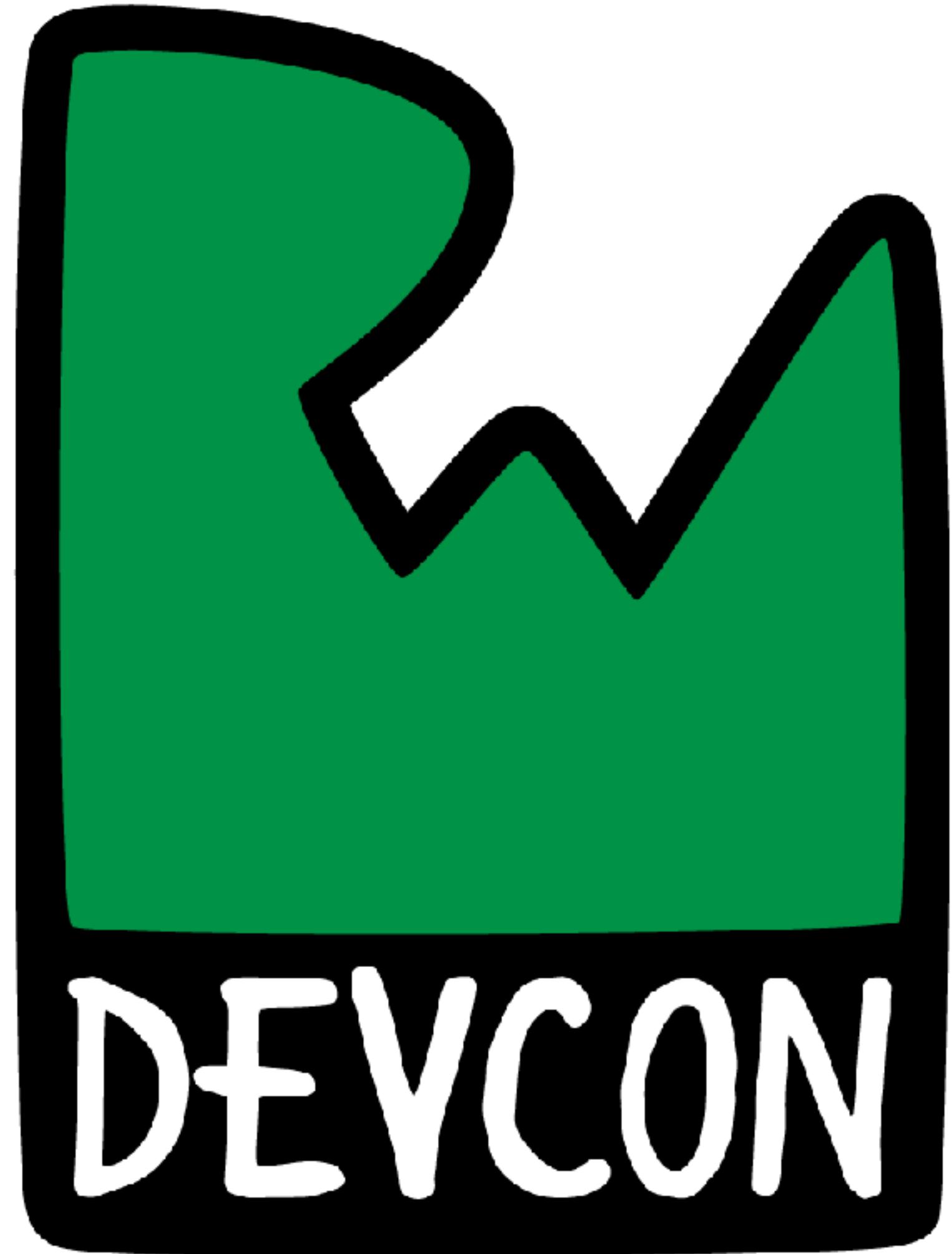


R
W

DEMO 3



Workshop: CoreML + Vision



CONCLUSION

WHAT You LEARNED

- ⚙️ **Demo 1:** Vision
- ⚙️ **Demo 2:** Face Recognition
- ⚙️ **Demo 3:** Recognize *my* face!



WHERE To Go FROM HERE?

- ⚙️ [https://
developer.apple.com/
machine-learning/](https://developer.apple.com/machine-learning/)
- ⚙️ Twitter:
@mataharimau
@pbkwete

