

IBM Watson Visual Recognition Workflow with Cloudinary

Aditi Madan, Cloudinary

<https://github.com/aditimadan-Cloudinary>
aditi@cloudinary.com

IBM Developer

Upkar Lidder, IBM

@lidderupk

<https://github.com/lidderupk/>
ulidder@us.ibm.com

Slides: <https://github.com/lidderupk/watson-vr-cloudinary-12.10.18>

IBM Cloud: <https://ibm.biz/BdYqM6>

1500 Drones Are Ready To Fly

IBM Developer Drone Challenge

Challenge runs from November 12 through December 16 and those 18 years of age or older residing in the US and Canada are eligible to enter

Enter at <https://developer.ibm.com/contest>

Once per week during the 5 weeks of the contest a random drawing

Will be held to determine the winners

(watch for the drawing on Twitch)

Winners will receive a [DJI Tello programmable drone](#),

an [IBM Developer T-shirt](#), and an [IBM Developer laptop sticker](#)

Winners will also receive a special code to unlock more challenges and given the opportunity to contribute back to the challenge



#IBMDroneDrop



Catalog

Search the catalog...

Filter

All Categories

- Compute
- Containers
- Networking
- Storage
- AI**
- Analytics
- Databases
- Developer Tools
- Integration
- Internet of Things
- Security and Identity
- Starter Kits
- Web and Mobile
- Web and Application

AI

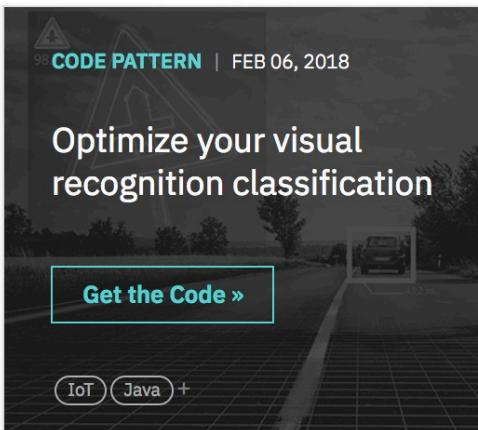
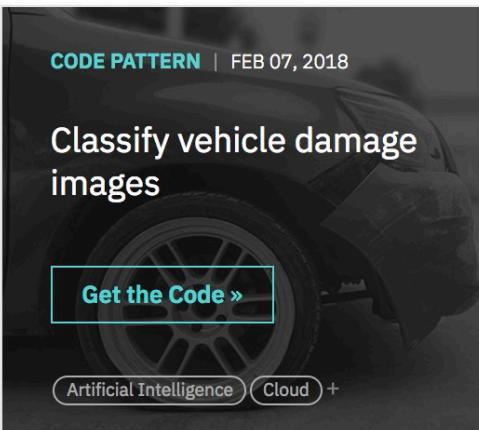
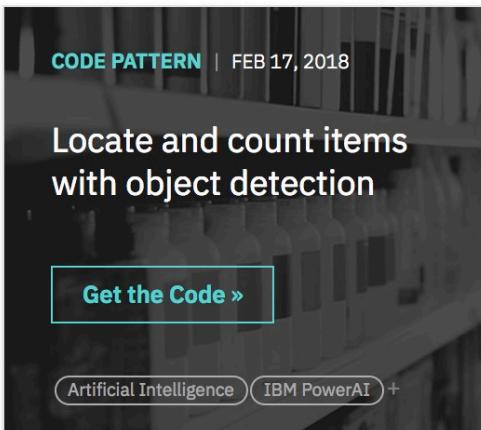
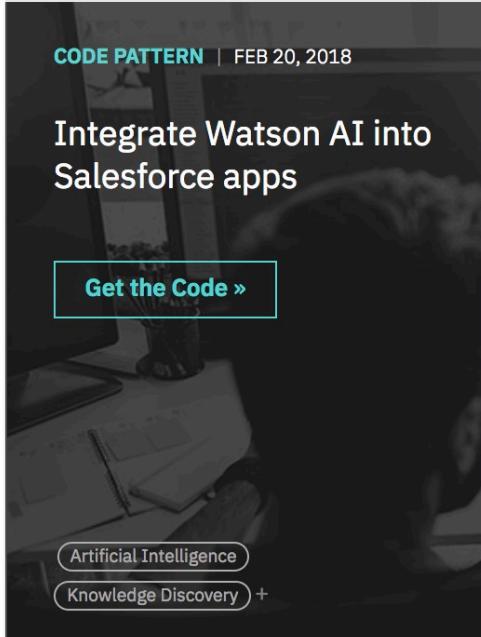
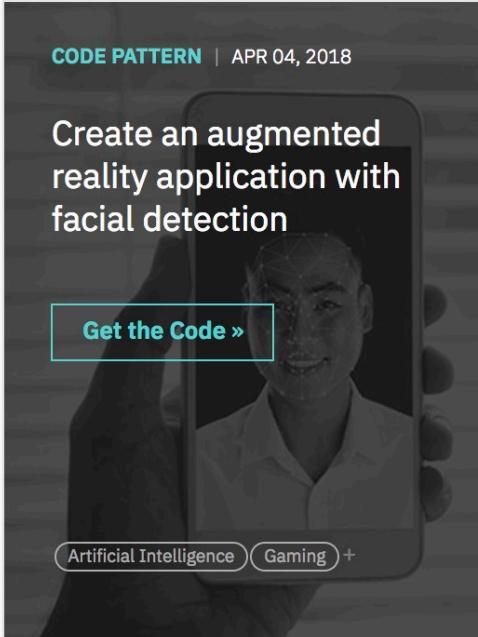
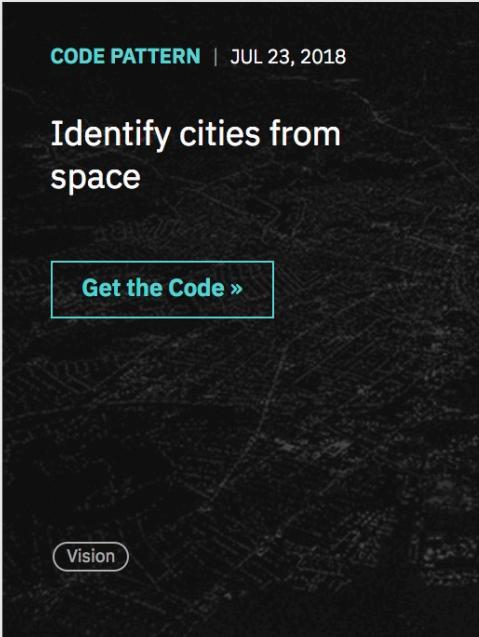
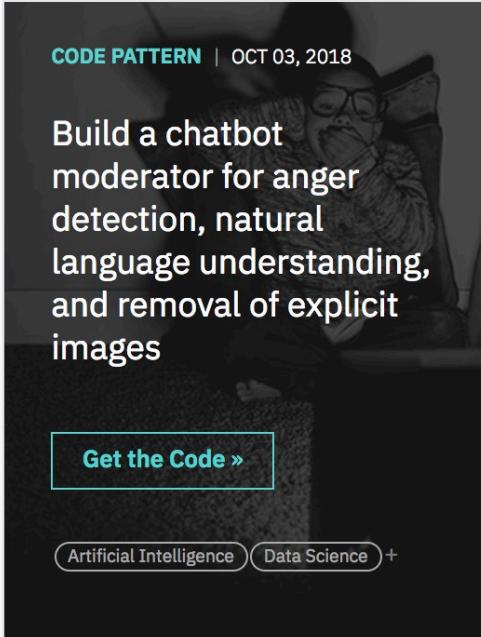
Watson Assistant (formerly Conversation)	AI OpenScale	Compare Comply	Discovery	Knowledge Catalog
Lite • IBM Watson Assistant a platform that allows developers and non-technical users to collaborate on building conversational AI-powered assistants.	Lite • IBM IBM AI OpenScale is an enterprise-grade environment for AI infused applications that provides enterprises with visibility into how AI is...	IBM • Beta Process governing documents to convert, identify, classify, and compare important elements	Lite • IBM Add a cognitive search and content analytics engine to applications.	Lite • IBM Discover, catalog, and securely share enterprise data.
Lite • IBM Teach Watson the language of your domain.	Lite • IBM Translate text, documents, and websites from one language to another. Create industry or region-specific translations via the service's customization...	Lite • IBM IBM Watson Machine Learning - make smarter decisions, solve tough problems, and improve user outcomes.	IBM Natural Language Classifier uses advanced natural language processing and machine learning techniques to create custom classification models...	Lite • IBM Analyze text to extract meta-data from content such as concepts, entities, emotion, relations, sentiment and more.
Lite • IBM The Watson Personality Insights derives insights from transactional and social media data to identify psychological traits	Lite • IBM Low-latency, streaming transcription	Lite • IBM Synthesizes natural-sounding speech from text.	Lite • IBM Tone Analyzer uses linguistic analysis to detect three types of tones from communications: emotion, social, and language. This insight can be the...	Lite • IBM Find meaning in visual content! Analyze images for scenes, objects, faces, and other content. Choose a default model off the shelf, or create your own...
Lite • IBM Embed AI and machine learning into your business. Create custom models using your own data.	Third Party The accelerated deep learning platform for enterprise. Built on the IBM PowerAI platform, powered by Nimbix.			

IBM Code Patterns

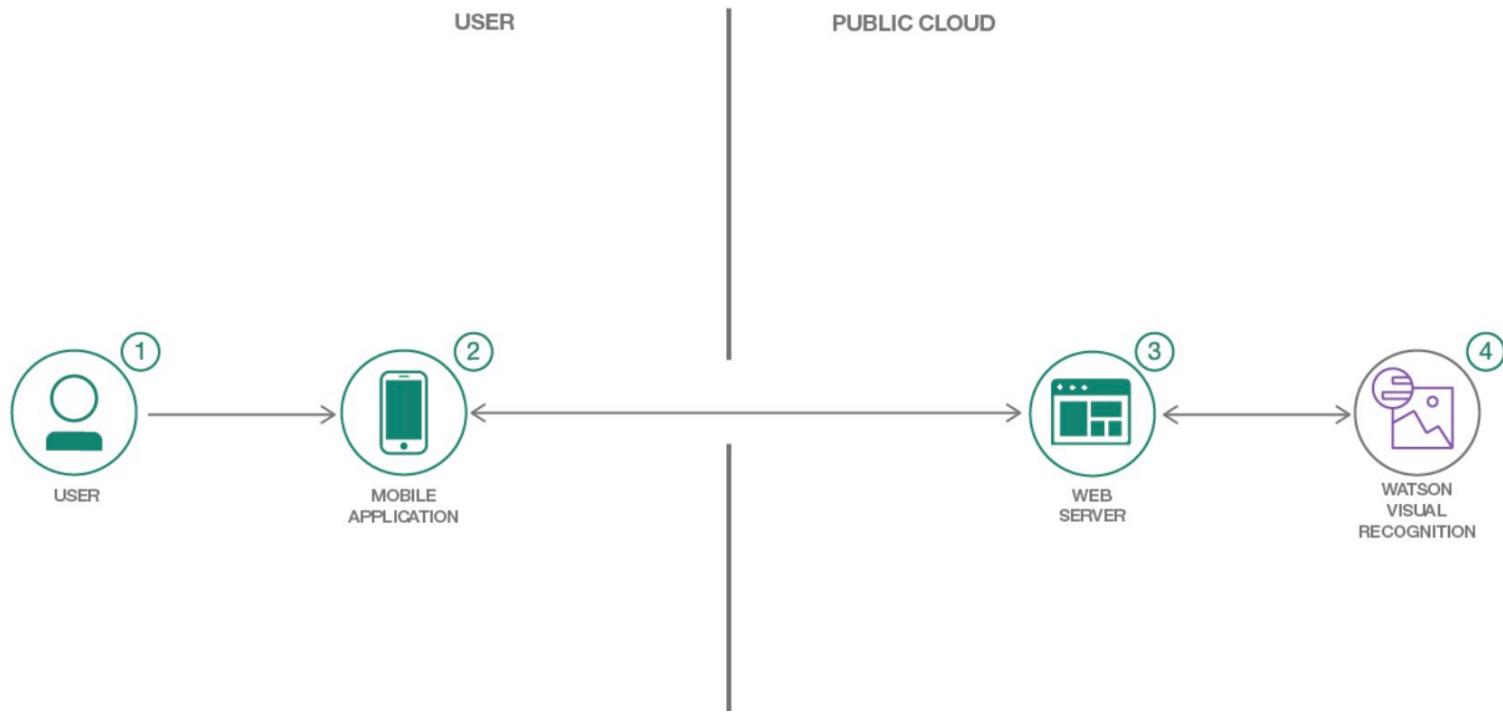
<https://developer.ibm.com/patterns/> -> Try it out !

- Analytics
- API Management
- Application Lifecycle Management
- Artificial Intelligence
- Blockchain
- Containers
- Continuous Integration
- Continuous Delivery
- Continuous Testing
- Conversation
- Data Governance
- Data Management
- Data Science
- Messaging
- Microservices
- Mobile Development
- Natural Language Processing
- Node.js
- Object Storage
- Platform as a Service
- Predictive Analytics
- Python
- Quantum Computing
- Reactive Programming
- Responsive Design
- Security

- Data Stores
- Databases
- Deep Learning
- DevOps
- Governance
- Infrastructure
- IoT
- Java
- Java Platforms
- JavaScript
- Knowledge Discovery
- Linux
- Machine Learning
- Serverless
- SOA
- Software Development
- Speech and Empathy
- Storage
- Streaming Analytics
- Swift on the Server
- Systems
- Virtual Reality
- Vision
- Web Analytics
- Web Development

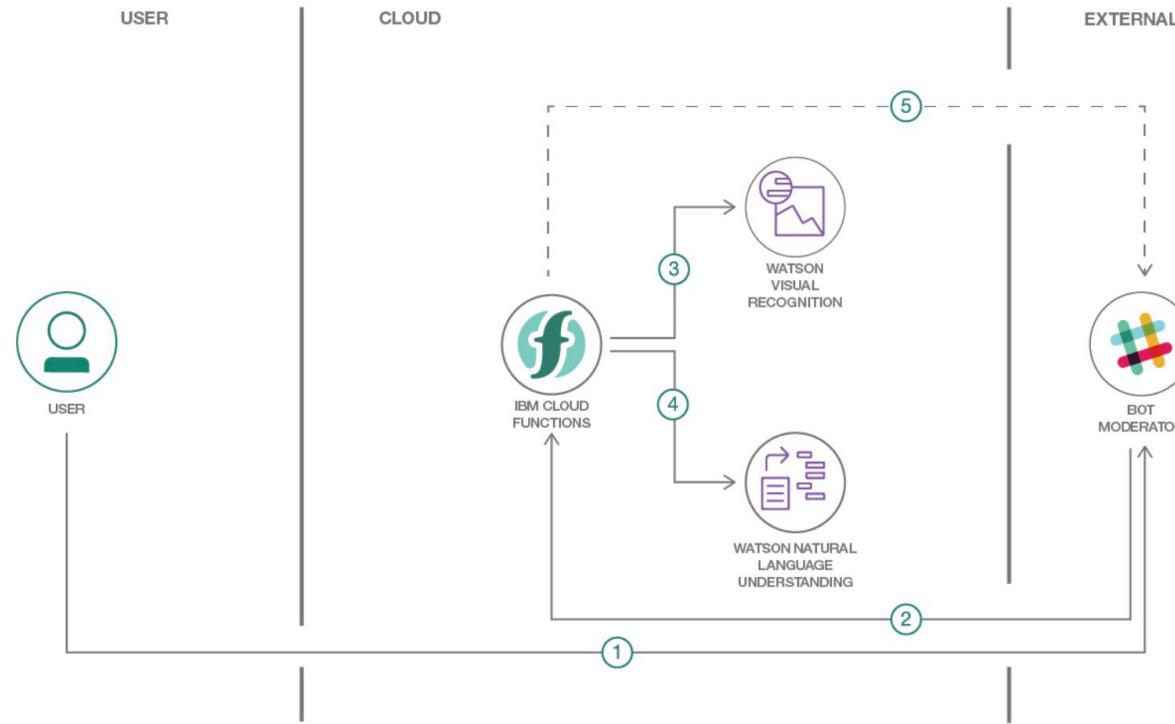


Classify vehicle damage images



1. The user captures an image with the mobile application.
2. The user sends the image on the mobile phone to the server application running in the cloud.
3. The server sends the image to Watson Visual Recognition service for analysis.
4. Watson Visual Recognition service classifies the image and returns the information to the server.

Chatbot - Anger detection, NLU, and removal of explicit images



1. The user interacts with the Slack app and either sends a text or uploads an image.
2. A bot passes the text or image used in Slack to an IBM Cloud Functions API.
3. Watson Visual Recognition categorizes the uploaded image using the default and explicit classifiers.
4. Watson Natural Language Processing categorizes the text if text is sent as part of the Slack communication.
5. IBM Cloud Functions receives the response, and if the text is not polite, the bot sends a message to the Slack user to be more polite using the [Slack post message API](#). If an image that is considered explicit is used, the image is deleted by IBM Cloud Functions using the [Slack files delete API](#).

Watson Studio

IBM Watson

UPKAR LIDDER's Acco... UL

Get started ▾



Welcome Upkar!

Watson Studio is part of IBM Watson.

Try out other IBM Watson apps.

Get started with key tasks

New project Refine data New notebook Deep learning Hide ▾

Recently updated projects View all (5) + New project

NAME	ROLE	COLLABORATORS	DATE CREATED	LAST UPDATED
cable-test	Admin	UL	Jul 20, 2018	Jul 20, 2018
medicine-visual-recognition	Admin	UL	Jul 13, 2018	Jul 13, 2018
wildfire	Admin	UL	Jul 12, 2018	Jul 12, 2018
traffic-code-pattern	Admin	UL	Jul 10, 2018	Jul 10, 2018
raspberry-pi	Admin	UL	Jun 26, 2018	Jun 26, 2018

IBM Watson

Projects Tools Community Services Manage Support Docs

Projects / medicine-visual-recognition / Default Custom Model

Get started ▾

Default Custom Model

Associated Service : upkar-watson-medicine-classification

My classes (4) All images (47)

Drag and drop zip files from your project.

4 classes | 0 incomplete classes | 5 unclassified images

Total file size: 70.0/250 MB

Search classes

Create a class

mucinex 19 images

negative (recommended) 0 images

Use the negative class to train the model on images that do not depict the visual subject of any of the positive classes.

nyquil 10 images

vitaminc 13 images

Model is not yet ready to train. Learn why.

Train Model i

1. Upload to project

To add files to your project, drop .zip files here or [Browse](#)

2. Add from project

Drag .zip files from your project to the training area to add them to your model.

0 selected

mucinex.zip 13 Jul 2018, 3:57:06 pm 30.74 MB

nyquil.zip 13 Jul 2018, 3:57:05 pm 16.97 MB

vitaminc.zip 13 Jul 2018, 3:56:23 pm 22.28 MB

IBM Developer

8

Classifiers/Models

A set of built-in models provides highly accurate results without training:

- General model: Default classification from thousands of classes.
- Face model: Facial analysis with age and gender.
- Explicit model (Beta): Whether an image is inappropriate for general use.
- Food model (Beta): Specifically for images of food items.
- Text model (Private beta): Text extraction from natural scene images.

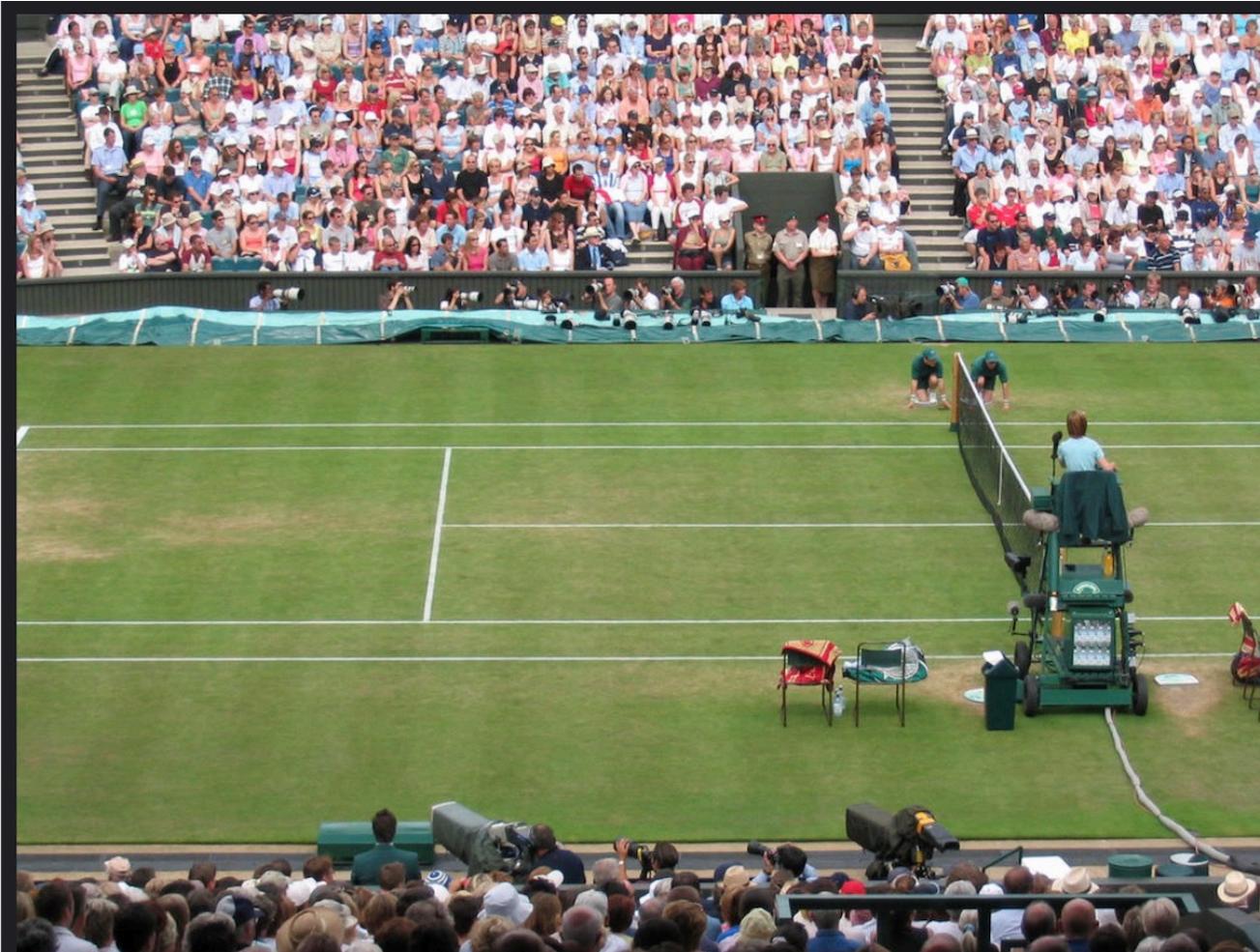
Requires training:

- Custom model: Train a custom classifier by providing positive and negative images.

General Model

- Animals (including birds, reptiles, amphibians, etc.)
- Person and people-oriented information and activities
- Food (including cooked food and beverages)
- Plants (including trees, shrubs, aquatic plants, vegetables)
- Sports
- Nature (including many types of natural formations, geological structures)
- Transportation (land, water, air)
- And many more, including furnishings, fruits, musical instruments, tools, colors, gadgets, devices, instruments, weapons, buildings, structures and man-made objects, clothing and garments, and flowers, among others.

General Model Example



```
{  
  "images": [  
    {  
      "classifiers": [  
        {  
          "classifier_id": "default",  
          "name": "default",  
          "classes": [  
            {  
              "class": "tennis",  
              "score": 0.762,  
              "type_hierarchy": "/sport/athletic game/court game/tennis"  
            },  
            {  
              "class": "court game",  
              "score": 0.774  
            },  
            {  
              "class": "athletic game",  
              "score": 0.779  
            },  
            {  
              "class": "sport",  
              "score": 0.786  
            },  
            {  
              "class": "ballpark",  
              "score": 0.732,  
              "type_hierarchy": "/sports stadium/ballpark"  
            },  
            {  
              "class": "sports stadium",  
              "score": 0.817  
            },  
            {  
              "class": "greenishness color",  
              "score": 0.716  
            }  
          ]  
      },  
      "image": "sport.jpg"  
    ]  
  ],  
  "images_processed": 1,  
  "custom_classes": 0  
}
```

Face Detect



[Photo by Tom Kelly / CC BY](#)

```
{  
  "images": [  
    {  
      "faces": [  
        {  
          "age": {  
            "min": 42,  
            "max": 45,  
            "score": 0.8267146  
          },  
          "face_location": {  
            "height": 240,  
            "width": 198,  
            "left": 587,  
            "top": 258  
          },  
          "gender": {  
            "gender": "MALE",  
            "score": 0.99999547  
          }  
        },  
        {  
          "age": {  
            "min": 20,  
            "max": 22,  
            "score": 0.99971515  
          },  
          "face_location": {  
            "height": 257,  
            "width": 209,  
            "left": 1275,  
            "top": 330  
          },  
          "gender": {  
            "gender": "MALE",  
            "score": 0.9667457  
          }  
        }  
      ]  
    }  
  ],  
  "image": "family.jpg"  
},  
  "images_processed": 1  
}
```

Face Detect



Photo by Alan Kotok / [CC BY](#)

classify

```
{  
  "images": [  
    {  
      "classifiers": [  
        {  
          "classifier_id": "default",  
          "name": "default",  
          "classes": [  
            {  
              "class": "day school",  
              "score": 0.678,  
              "type_hierarchy": "/building/school/day schoo  
            },  
            {  
              "class": "school",  
              "score": 0.678  
            },  
            {  
              "class": "building",  
              "score": 0.678  
            },  
            {  
              "class": "claret red color",  
              "score": 0.733  
            },  
            {  
              "class": "alizarine red color",  
              "score": 0.672  
            }  
          ]  
        },  
        {"image": "cuba.jpg"}  
      ],  
      "images_processed": 1,  
      "custom_classes": 0  
    }  
  ]  
}
```

detect_faces

```
{  
  "images": [  
    {  
      "faces": [  
        {  
          "age": {  
            "min": 0,  
            "max": 12,  
            "score": 0.36798656  
          },  
          "face_location": {  
            "height": 129,  
            "width": 103,  
            "left": 1229,  
            "top": 352  
          },  
          "gender": {  
            "gender": "MALE",  
            "score": 0.68076503  
          }  
        },  
        {  
          "age": {  
            "min": 54,  
            "max": 57,  
            "score": 0.79777867  
          },  
          "face_location": {  
            "height": 86,  
            "width": 75,  
            "left": 995,  
            "top": 76  
          },  
          "gender": {  
            "gender": "MALE",  
            "score": 0.96644753  
          }  
        }  
      ]  
    }  
  ]  
}
```

Food

classifier_ids=""



[Photo](#) by [Lola Williams](#) / [CC BY](#)

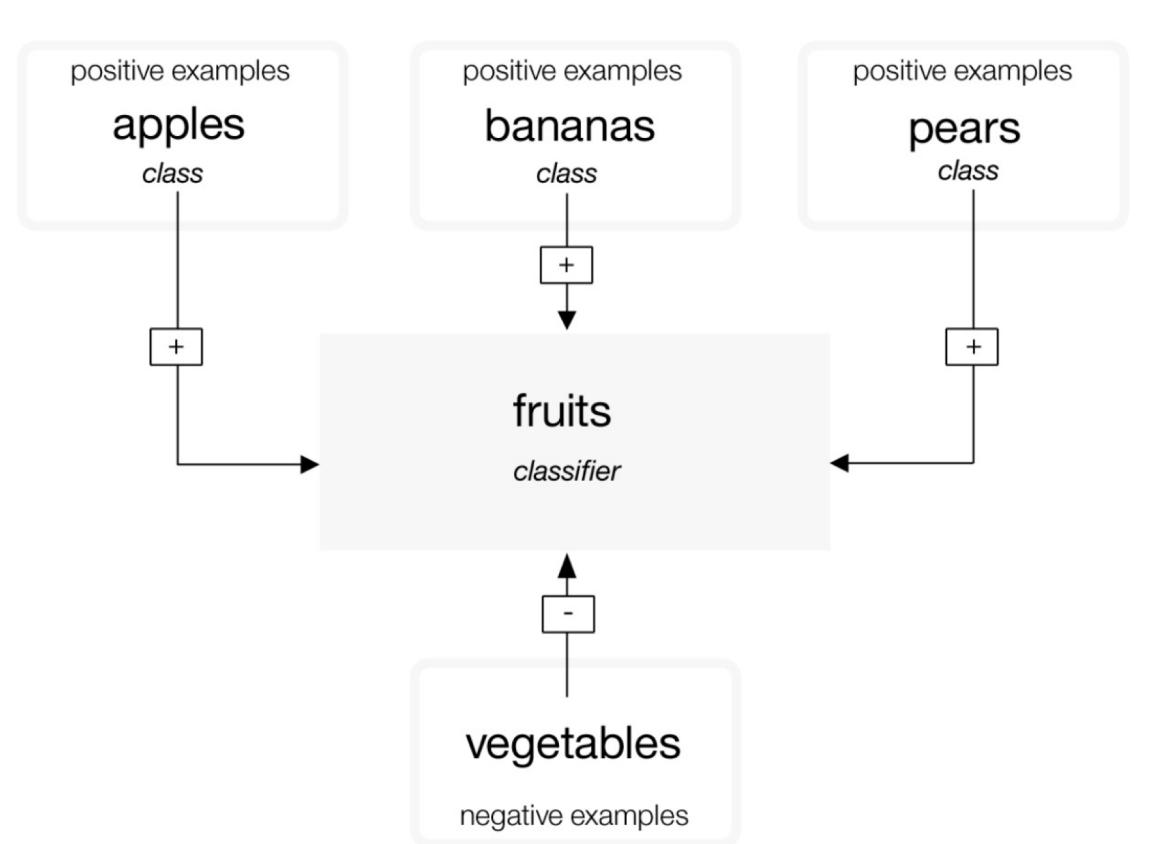
```
{  
  "images": [  
    {  
      "classifiers": [  
        {  
          "classifier_id": "default",  
          "name": "default",  
          "classes": [  
            {  
              "class": "plant",  
              "score": 0.631  
            },  
            {  
              "class": "berry",  
              "score": 0.6  
            },  
            {  
              "class": "dark red color",  
              "score": 0.911  
            }  
          ]  
        },  
        {"image": "fruit2.jpg"}  
      ],  
      "images_processed": 1,  
      "custom_classes": 0  
    }  
  ]  
}
```

classifier_ids=food

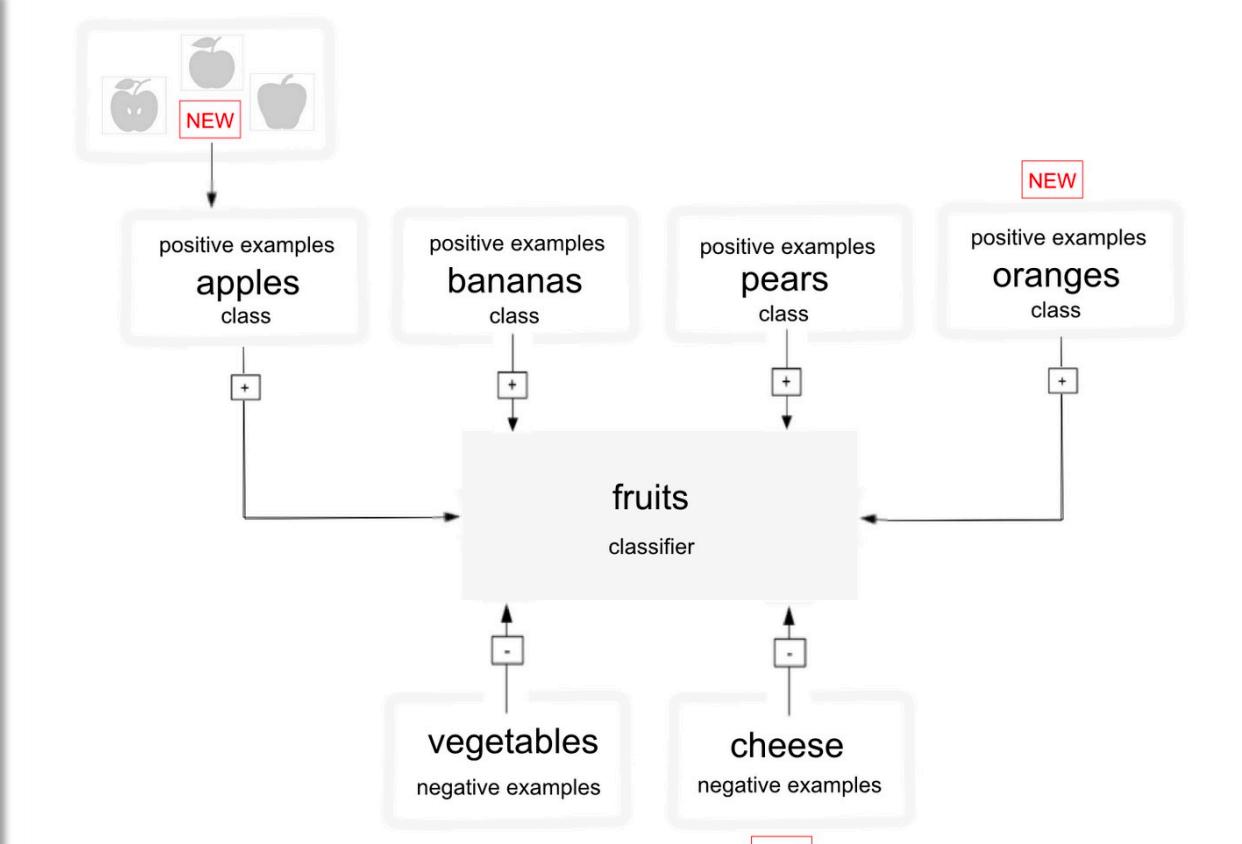
```
{  
  "images": [  
    {  
      "classifiers": [  
        {  
          "classifier_id": "food",  
          "name": "food",  
          "classes": [  
            {  
              "class": "raspberry",  
              "score": 0.934,  
              "type_hierarchy": "/fruit/berry/raspberry"  
            },  
            {  
              "class": "berry",  
              "score": 0.95  
            },  
            {  
              "class": "fruit",  
              "score": 0.95  
            }  
          ]  
        },  
        {"image": "fruit2.jpg"}  
      ],  
      "images_processed": 1,  
      "custom_classes": 0  
    }  
  ]  
}
```

Custom Models

Initial Training



Update Training



DEMO

- IBM Cloud
- Visual Recognition API
- Watson Studio
- Sample apps



Testing the custom model in UI

Data assets

0 asset selected.

NAME	TYPE	SERVICE	CREATED BY	LAST MODIFIED	ACTIONS
mucinex.zip	Data Asset	Project	UPKAR LIDDER	13 Jul 2018, 3:57:07 pm	⋮
nyquil.zip	Data Asset	Project	UPKAR LIDDER	13 Jul 2018, 3:57:05 pm	⋮
vitaminc.zip	Data Asset	Project	UPKAR LIDDER	13 Jul 2018, 3:56:24 pm	⋮

Models

Natural Language Classifier models **BETA** [+ New Natural Language Classifier model](#)

NAME	MODEL ID	SERVICE INSTANCE	LAST MODIFIED	ACTIONS
You don't have any Natural Language Classifier models yet.				

Visual Recognition models [+ New Visual Recognition model](#)

NAME	MODEL ID	SERVICE INSTANCE	LAST MODIFIED	ACTIONS
Medicine-Model	MedicinexModel_257596841	Visual Recognition	24 Jul 2018, 4:26:45 pm	⋮

Summary

Model ID	MedicinexModel_257596841
Status	Ready
Explanation	This model is ready for use.
Created by	ulidder@us.ibm.com
Created on	7/24/2018, 4:24:04 PM
Number of classes	3
Number of images	42

Classes

CLASS	NUMBER OF EXAMPLES
mucinex	19
nyquil	10
vitaminc	13

Projects / medicine-visual-recognition / Medicine-Model

Threshold: 0.0

Class: mucinex, nyquil, vitaminc



mucinex-1.JPG

mucinex	0.90
vitaminc	0.03
nyquil	0.01



test-emergenc-3.JPG

vitaminc	0.76
nyquil	0.38
mucinex	0.12



test-micinex-1.JPG

mucinex	0.85
vitaminc	0.19
nyquil	0.05



test-mucinex-3.JPG

mucinex	0.90
vitaminc	0.04
nyquil	0.01



test-vitaminc-1.JPG

mucinex	0.89
vitaminc	0.10
nyquil	0.01



test-emergenc-1.JPG

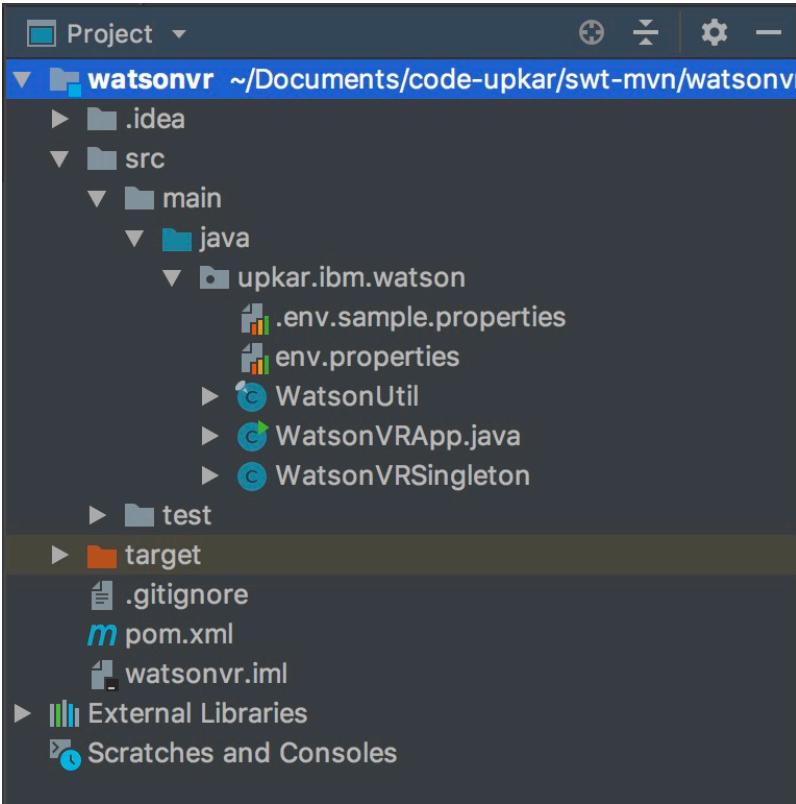
vitaminc	0.91
nyquil	0.00
mucinex	0.00



test-emergenc-2.JPG

vitaminc	0.88
mucinex	0.08

Java Application



```
<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <java.version>1.8</java.version>
</properties>

<repositories>
    <repository>
        <id>maven-eclipse-repo</id>
        <url>http://maven-eclipse.github.io/maven</url>
    </repository>
</repositories>

<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.1</version>
            <configuration>
                <source>${java.version}</source>
                <target>${java.version}</target>
            </configuration>
        </plugin>
    </plugins>
</build>
<dependencies>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.11</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>org.eclipse.swt</groupId>
        <artifactId>org.eclipse.swt.cocoa.macosx.x86_64</artifactId>
        <version>4.6.1</version>
    </dependency>
    <dependency>
        <groupId>com.ibm.watson.developer_cloud</groupId>
        <artifactId>java-sdk</artifactId>
        <version>6.9.0</version>
    </dependency>
</dependencies>
```

1	VERSION=
2	API_KEY=

Java Application – Util Class

```
package upkar.ibm.watson;

import com.ibm.watson.developer_cloud.visual_recognition.v3.VisualRecognition;
import com.ibm.watson.developer_cloud.visual_recognition.v3.model.ClassifiedImages;
import com.ibm.watson.developer_cloud.visual_recognition.v3.model.Classifiers;
import com.ibm.watson.developer_cloud.visual_recognition.v3.model.ClassifyOptions;
import com.ibm.watson.developer_cloud.visual_recognition.v3.model.ListClassifiersOptions;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

public final class WatsonUtil {

    private WatsonUtil(){
        }

    static ClassifiedImages classify(String filePath, double threshold, Boolean checkCustom) throws FileNotFoundException {...}

    static Classifiers getClassifiers() {...}
}
```

Java Application – Default Classifier



The application interface displays a photograph of a shop on Bleeker Street. The shop has a green sign that reads "NEWSPAPERS - TOBACCO" flanked by two Coca-Cola signs. Below the sign, the shop is labeled "SOOTTA" and its address is "233 BLEECKER ST.". A bicycle is parked in front of the shop. The background shows a brick building with windows and air conditioning units.

Result

default

retail store	:0.962
building	:0.962
shop	:0.954
deli	:0.82
marketplace	:0.796
brick red color	:0.563
convenience store	:0.528
tobacco shop	:0.5
Indian red color	:0.391

Pick the classifiers: Default Food Face medicines

0.18

ANALYZE

Java Application – Custom Classifier



Result

medicines

mucinex :0.805
vitaminc :0.378

default

bandage :0.778
medical dressing :0.778
alizarine red color :0.755
reddish brown color :0.689
Band Aid :0.641
food :0.599
beverage :0.599
paint :0.598
food product :0.596
adhesive bandage :0.5

Pick the classifiers: Default Food Face medicines

0.18

ANALYZE

Swift Application

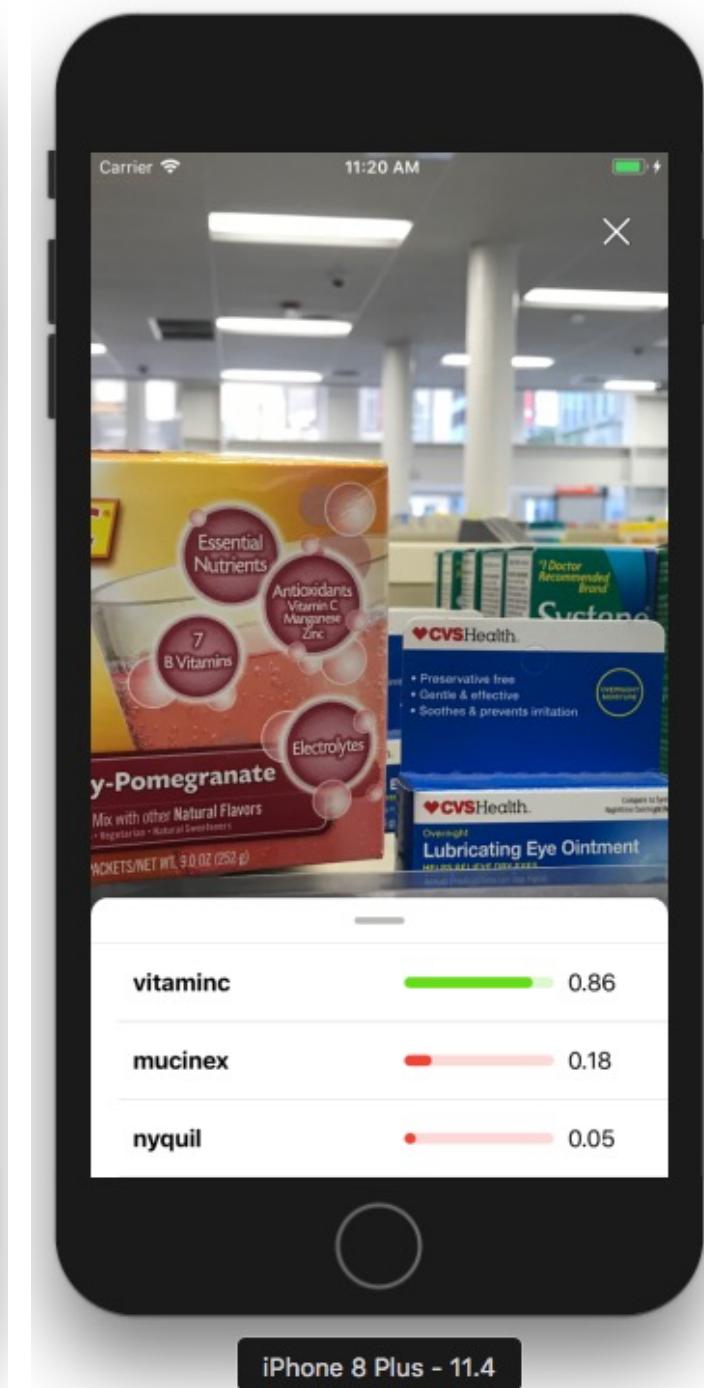
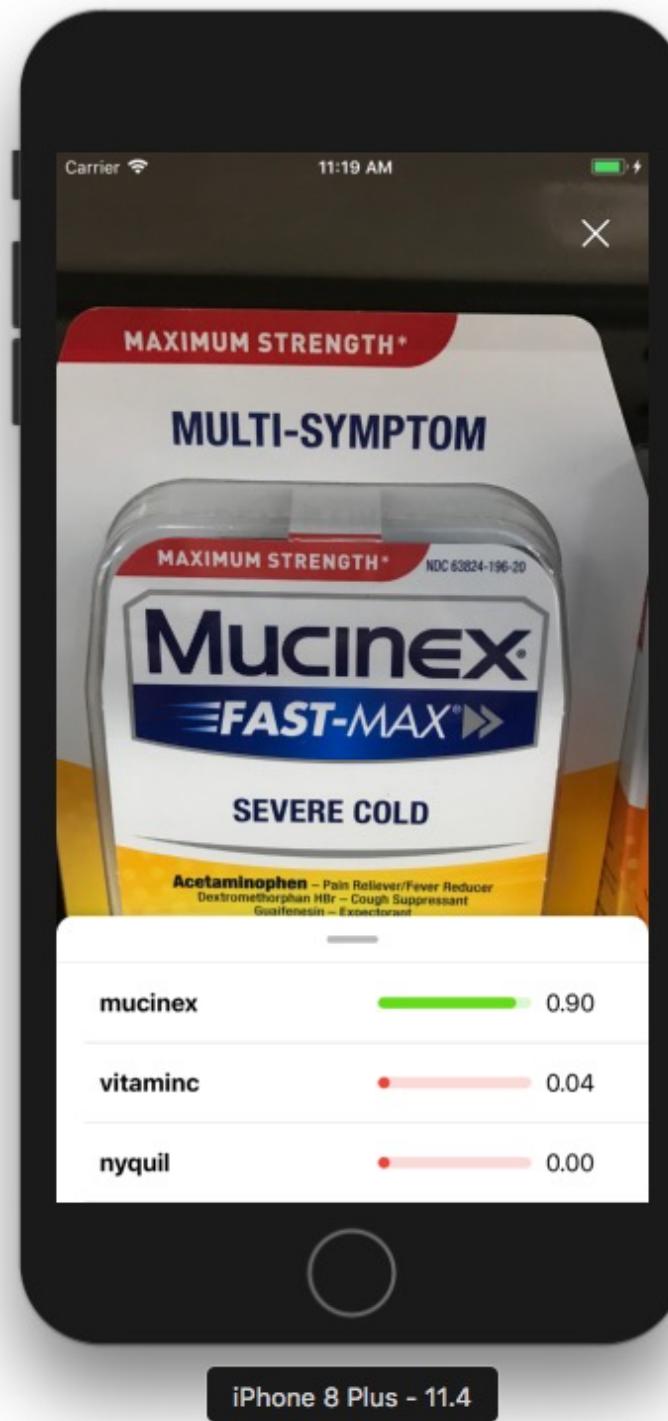
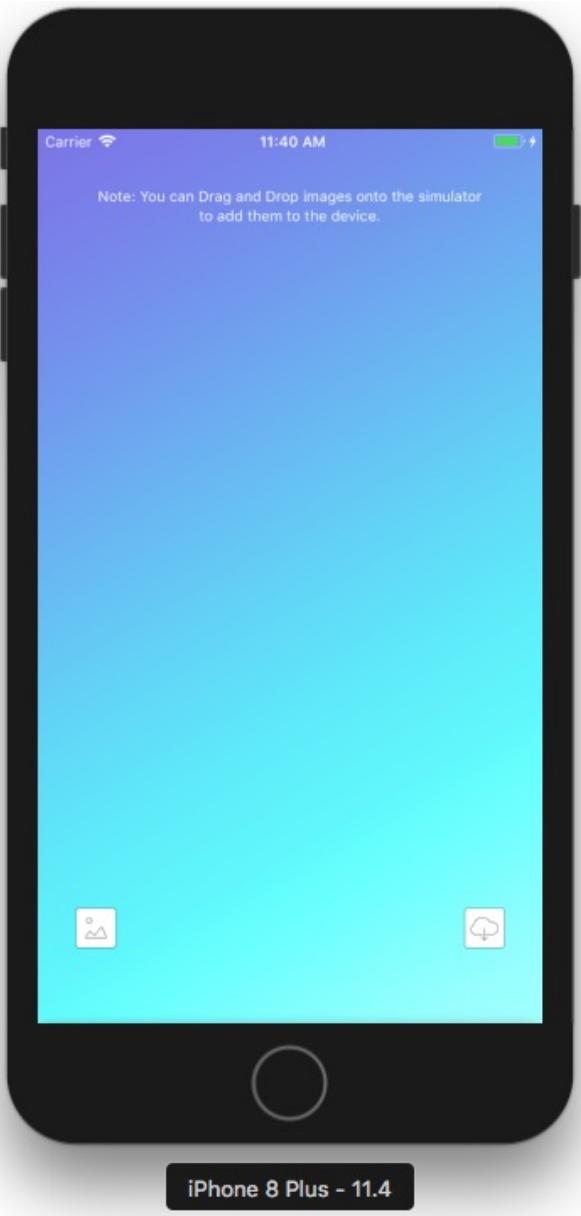
```
21 struct VisualRecognitionConstants {  
22     // Instantiation with `api_key` works only with Visual Recognition service instances created before May 23, 2018. Visual Recognition  
23     // instances created after May 22 use the IAM `apikey`.  
24     static let apikey = "-----" // The IAM apikey  
25     static let apiKey = "" // The apikey  
26     static let modelIDs = ["medicines_253364822"]  
27     static let version = "2018-03-19"  
28 }
```

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure under "Core ML Vision Custom". A file named "medicines_253364822.mlmodel" is highlighted with a red box.
- Code Editor:** Displays the "ImageClassificationViewController.swift" file. The code implements a classification function that uses a local model to identify images and updates the UI on the main thread.
- Search Bar:** At the top, it shows the current path: Core ML Vision Custom > Core ML Vision Custom > ImageClassificationViewController.swift > push(results:position:).

```
138     func classifyImage(_ image: UIImage, localThreshold: Double = 0.0) {  
139         showResultsUI(for: image)  
140  
141         let failure = { (error: Error) in  
142             DispatchQueue.main.async {  
143                 self.showAlert("Could not classify image", alertMessage: error.localizedDescription)  
144                 self.resetUI()  
145             }  
146         }  
147  
148         let imageCentered = cropToCenter(image: image)  
149  
150         visualRecognition.classifyWithLocalModel(image: imageCentered, classifierIDs: VisualRecognitionConstants.modelIDs, threshold:  
151             localThreshold, failure: failure) { classifiedImages in  
152  
153             // Make sure that an image was successfully classified.  
154             guard let classifiedImage = classifiedImages.images.first else {  
155                 return  
156             }  
157  
158             // Update UI on main thread  
159             DispatchQueue.main.async {  
160                 // Push the classification results of all the provided models to the ResultsTableView.  
161                 self.push(results: classifiedImage.classifiers)  
162             }  
163         }  
164     }
```

Swift Application



Custom Models - Tips and Tricks

- Provide training examples that are similar to what you plan to analyze.
 - Example: training with tiger in a zoo and then using the model to classify tiger in the wild. The background, surroundings matter, angle, lighting, distance, size of subject, etc matter.
- If the quality and content of training data is the same, then classifiers/classes that are trained on more images will do better.
- Time vs accuracy - the benefits of more images plateaus around 5000 images.
- Parallelize by groups of images. You can train a class asynchronously.
- Recommend 150-200 images per .zip file with image size of 320 x 320. Do not need high resolution.
- Specify only classifiers you want the result for using the “classifier_id” parameter.
- How do you scale to hundreds of classes each with hundreds of training pictures ?

Thank you

 ulidder@us.ibm.com

 twitter.com/lidderupk

 github.com/lidderupk

 developer.ibm.com

