

Watson Visual Recognition API

Upkar Lidder, IBM

IBM Code

@lidderupk
<https://github.com/lidderupk/>



CALL FOR CODE

COMMIT TO THE CAUSE. PUSH FOR CHANGE.

Call for Code inspires developers to solve **pressing global problems** with **sustainable software solutions**, delivering on their vast potential to do good.

Bringing together NGOs, academic institutions, enterprises, and startup developers to compete build effective **disaster mitigation solutions**, with a focus on health and well-being.

The American Red Cross, and the United Nations Office of Human Rights combine for the **Call for Code Award** to elevate the profile of developers.

Award winners will receive **long-term support** through **The Linux Foundation**, **financial prizes**, the **opportunity to present their solution to leading VCs**, and will deploy their solution through **IBM's Corporate Service Corps**.

Developers will jump-start their project with dedicated **IBM Code Patterns**, combined with **optional enterprise technology** to build projects over the course of three months.

Judged by the world's most **renowned technologists**, the **grand prize** will be presented in **October** at an Award Event.

<http://callforcode.org>

IBM Code Patterns

<https://developer.ibm.com/code/patterns/>

IBM Code Code ▾ Content ▾

All code patterns

Technologies

28 Analytics

9 API Management

54 Artificial Intelligence

23 Blockchain

18 Container

Orchestration

17 Containers

28 Data Science

5 Data Store

25 Databases

1 DevOps

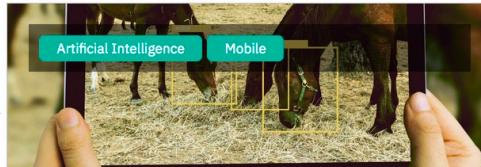


Build an iOS game powered by Core ML and Watson Visual Recognition

Use Watson Visual Recognition and Core ML to create a Kitura-based iOS game that has a user search for a predetermined list of objects.

⇒ [Read more](#)

⇒ [Get the code](#)



Deploy a Core ML model with Watson Visual Recognition

With Core ML, developers can integrate a trained machine learning model into an application. Watson Visual Recognition now supports exporting trained Core ML models. This code pattern shows you how to create a Core ML model using Watson Visual Recognition, which is then bundled into an application.

⇒ [Read more](#)

⇒ [Get the code](#)



Identify cities from space

Create a custom classifier with IBM Watson Visual Recognition that identifies several cities based on their images at night.

⇒ [Read more](#)

⇒ [Get the code](#)



Recycle with Watson

Create an iOS phone application that uses a Watson Visual Recognition custom classifier to sort waste into three categories (landfill, recycling, or compost).

⇒ [Read more](#)

⇒ [Get the code](#)

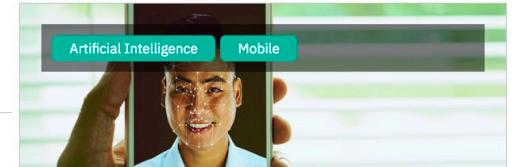


Optimize your visual recognition classification

Build training and prediction apps with Java, Watson IoT Platform, Node-RED, and Watson Visual Recognition service.

⇒ [Read more](#)

⇒ [Get the code](#)



Create an augmented reality application with facial detection

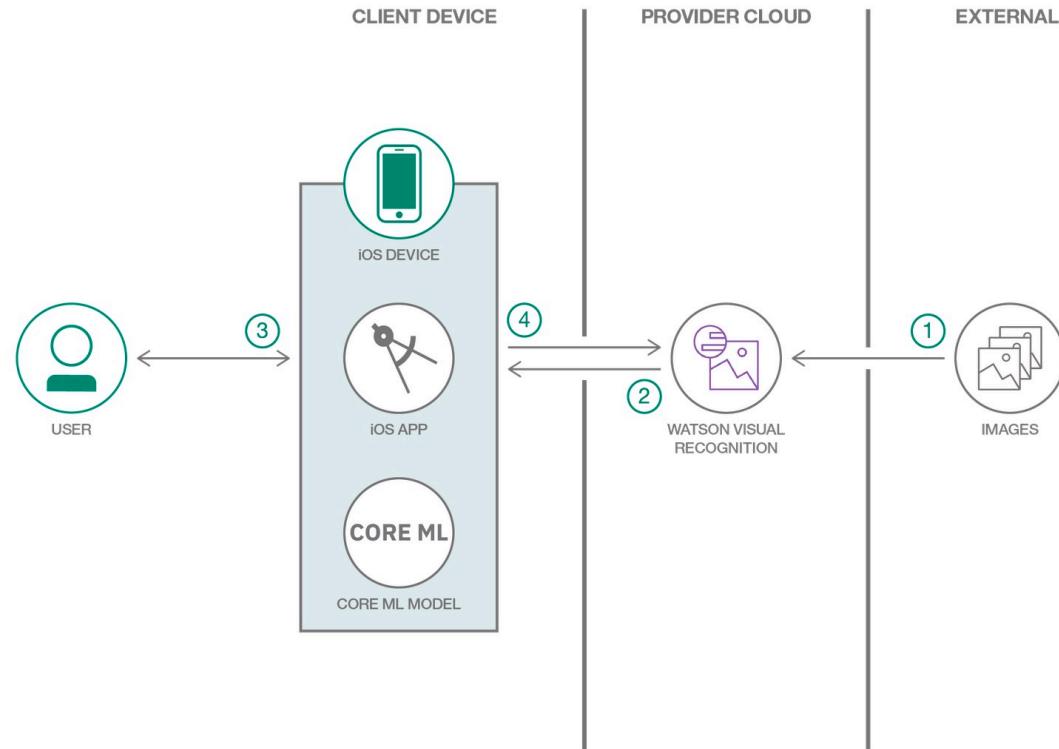
Create an app that combines ARKit with Watson Visual Recognition and a Cloudant database to give you a complete augmented reality experience.

⇒ [Read more](#)

⇒ [Get the code](#)

Classify offline with Watson Visual Recognition and Core ML

Flow



(1) Import and tag images.

(2) Train, test and deploy a Watson Visual Recognition model for Core ML.

(3) Run the application using to classify image using the Core ML model on the device.

(4) Get feedback from the user/device for iterative training in Watson.

IBM Cloud

All Categories

Search

Filter

Infrastructure

Compute

Storage

Network

Security

Containers

VMware

Platform

Boilerplates

APIs

Application Services

Blockchain

Cloud Foundry Apps

Data & Analytics

DevOps

Finance

Functions

Integrate

Internet of Things

Mobile

Security

Storage

Watson

Build cognitive apps that help enhance, scale, and accelerate human expertise.



Watson Assistant (formerly Conversation)

Add a natural language interface to your application to automate interactions with your end users. Common

Lite

IBM



Discovery

Add a cognitive search and content analytics engine to applications.

Lite

IBM



Knowledge Catalog

Discover, catalog, and securely share enterprise data.

Lite

IBM



Knowledge Studio

Build custom models to teach Watson the language of your domain.

Lite

IBM



Language Translator

Translate text from one language to another, adapt translation models to your custom domain.

Lite

IBM



Machine Learning

IBM Watson Machine Learning - make smarter decisions, solve tough problems, and improve user outcomes.

Lite

IBM



Natural Language Classifier

Natural Language Classifier performs natural language classification on question texts. A user would be able to:

IBM



Natural Language Understanding

Analyze text to extract meta-data from content such as concepts, entities, emotion, relations, sentiment and more.

Lite

IBM



Speech to Text

Low-latency, streaming transcription

Lite

IBM



Text to Speech

Synthesizes natural-sounding speech from text.

Lite

IBM



Visual Recognition

Find meaning in visual content! Analyze images for scenes, objects, faces, and other content. Choose a default mo

Lite

IBM



Watson Studio

Embed AI and machine learning into your business. Create custom models using your own data.

Lite

IBM

FEEDBACK

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

UPKAR LIDDER's Acco... UL

Projects / medicine-visual-recognition / Default Custom Model

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

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

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.

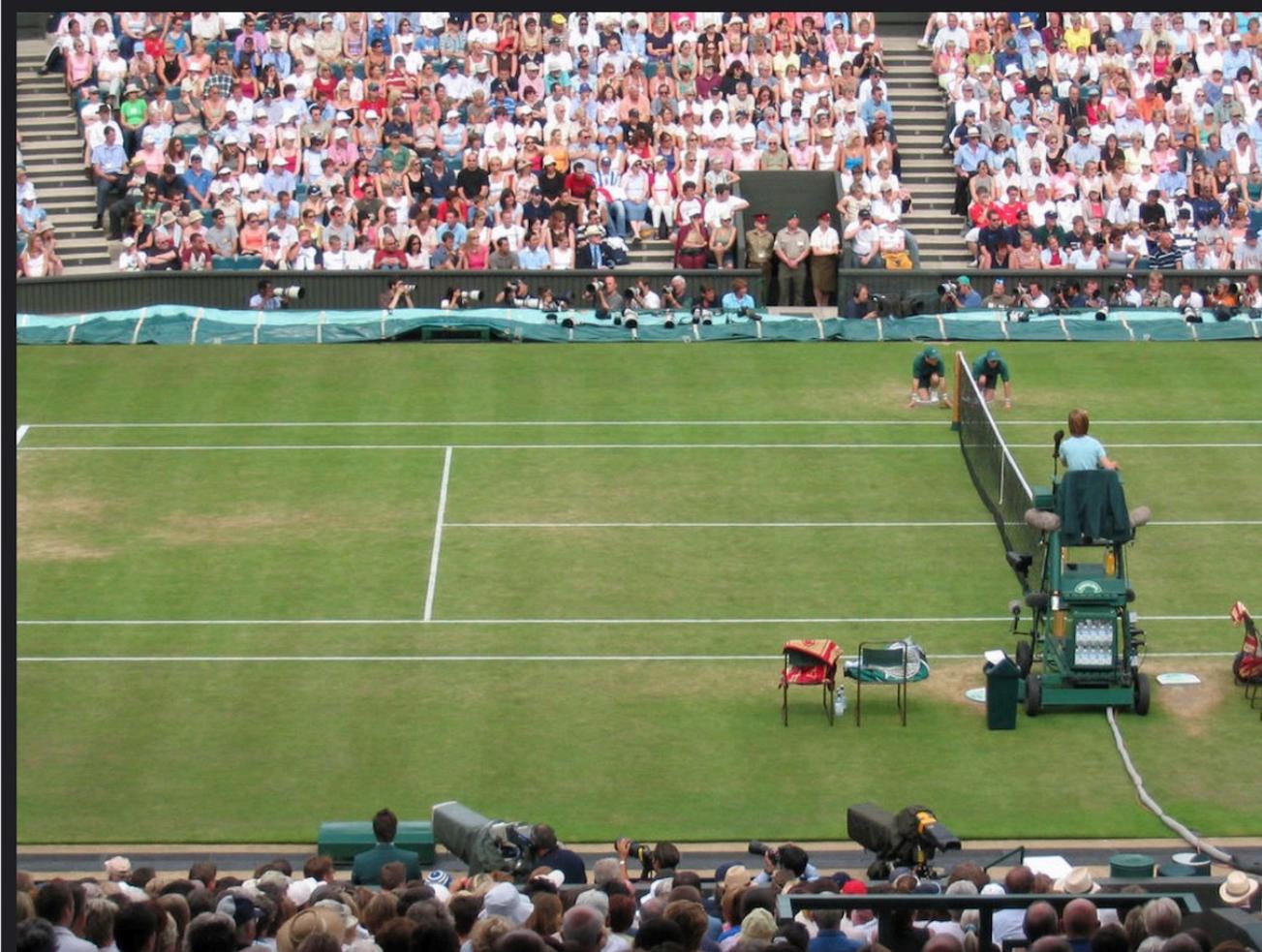
Requires training:

- 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](#)

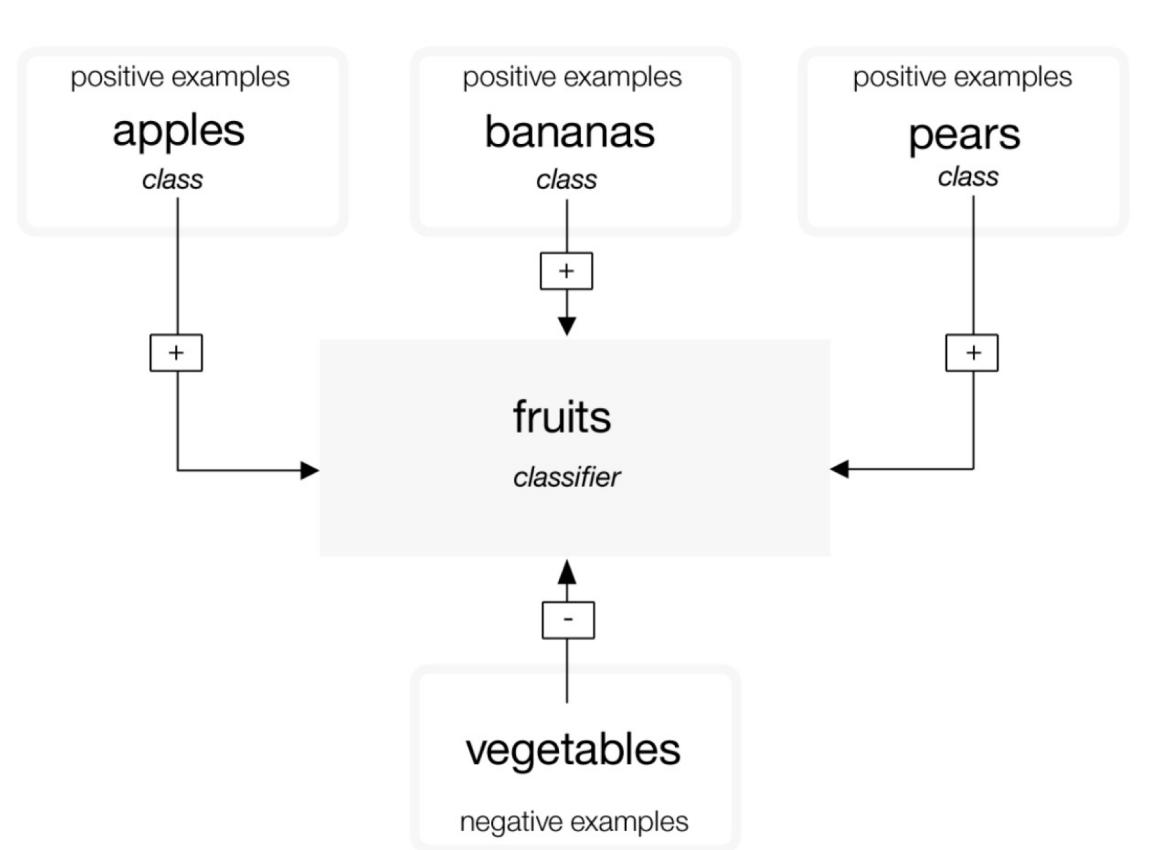
```
{  
  "images": [  
    {  
      "classifiers": [  
        {  
          "classifier_id": "default",  
          "name": "default",  
          "classes": [  
            {  
              "class": "plant",  
              "score": 0.631  
            },  
            {  
              "class":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

classifier_ids=food

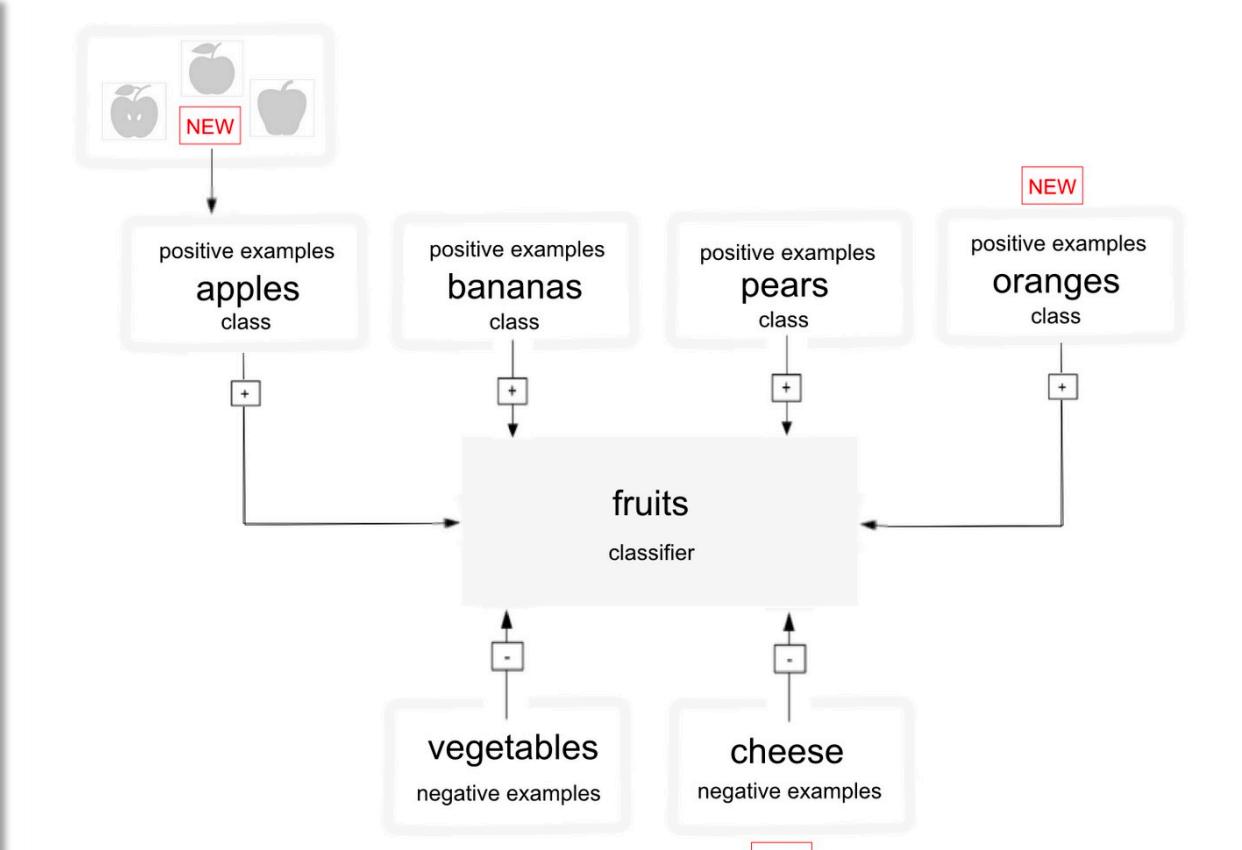
```
{  
  "images": [  
    {  
      "classifiers": [  
        {  
          "classifier_id": "food",  
          "name": "food",  
          "classes": [  
            {  
              "class": "raspberry",  
              "score": 0.934,  
              "type_hierarchy": "/fruit/berry/raspberry"  
            },  
            {  
              "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



Watson API Explorer

Visual Recognition

The IBM Watson™ Visual Recognition service uses deep learning algorithms to identify scenes, objects, and faces in images you upload to the service. You can create and train a custom classifier to identify subjects that suit your needs.

For more information about this service, see the IBM® Cloud docs.

<https://console.bluemix.net/docs/services/visual-recognition/getting-started.html>

General

Show/Hide | List Operations | Expand Operations

GET /v3/classify

Classify an image

POST /v3/classify

Classify images

Face

Show/Hide | List Operations | Expand Operations

GET /v3/detect_faces

Detect faces in an image

POST /v3/detect_faces

Detect faces in images

Custom

Show/Hide | List Operations | Expand Operations

GET /v3/classifiers

Retrieve a list of classifiers

POST /v3/classifiers

Create a classifier

DELETE /v3/classifiers/{classifier_id}

Delete a classifier

GET /v3/classifiers/{classifier_id}

Retrieve classifier details

POST /v3/classifiers/{classifier_id}

Update a classifier

Core ML

Show/Hide | List Operations | Expand Operations

GET /v3/classifiers/{classifier_id}/core_ml_model

Retrieve a Core ML model of a classifier

User data

Show/Hide | List Operations | Expand Operations

DELETE /v3/user_data

Delete labeled data

DEMO

✉ ulidder@us.ibm.com

🐦 twitter.com/lidderupk

🐙 github.com/lidderupk

IBM

Code

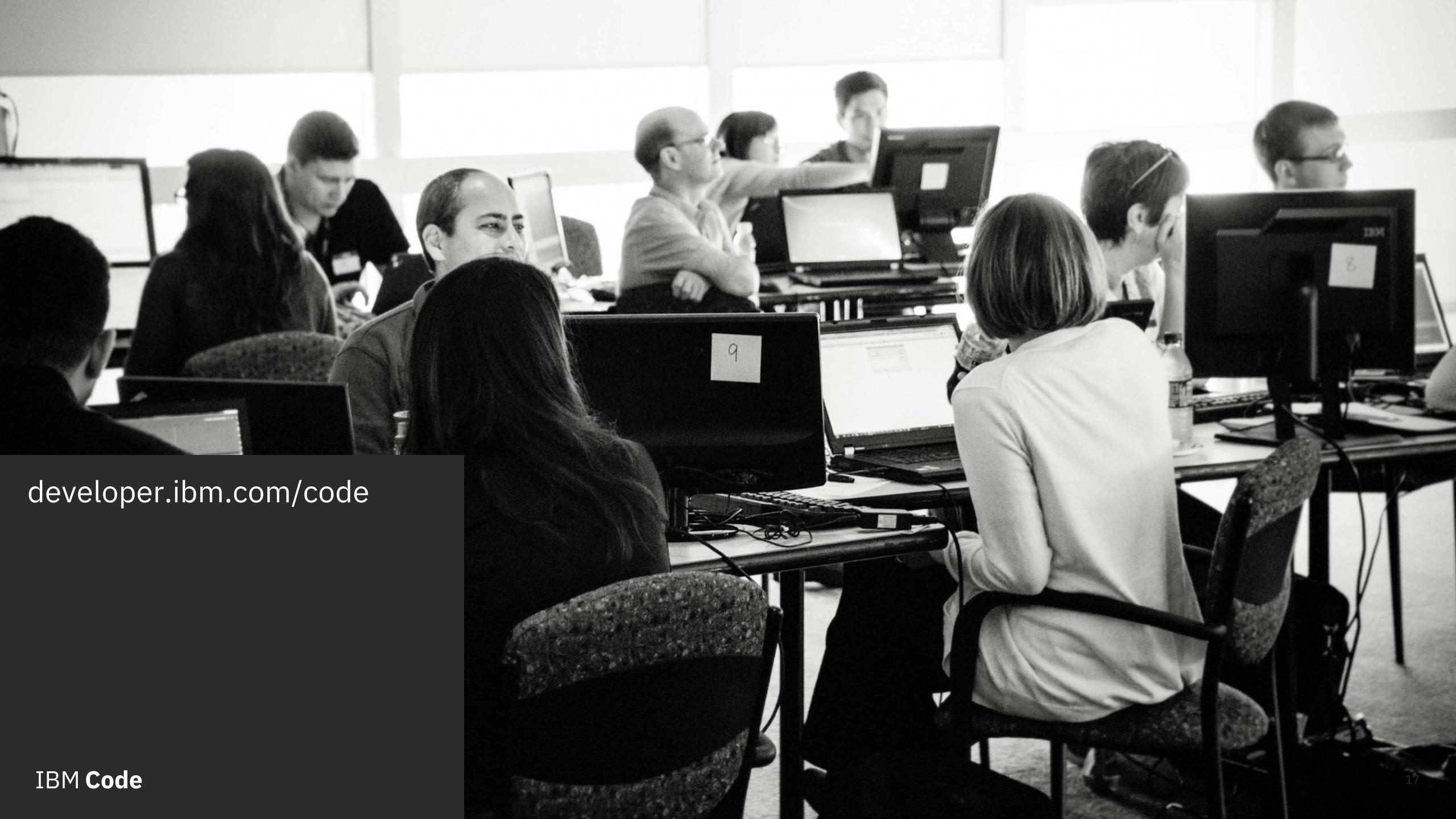
developer.ibm.com/code

<https://ibm.biz/BdYcHK>



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.



developer.ibm.com/code



Difficult Use Cases

While Watson Visual Recognition is highly flexible, there have been a number of recurring use cases that we've seen the API either struggle on or require significant pre/post-work from the user.

- **Face Recognition:** Visual Recognition is capable of face detection (detecting the presence of faces) not face recognition (identifying individuals).
- **Detecting details:** Occasionally, users want to classify an image based on a small section of an image or details scattered within an image. Because Watson analyzes the entire image when training, it may struggle on classifications that depend on small details. Some users have adopted the strategy of breaking the image into pieces or zooming into relevant parts of an image.
- **Emotion:** Emotion classification (whether facial emotion or contextual emotion) is not a feature currently supported by Visual Recognition. Some users have attempted to do this through custom classifiers, but this is an edge case and we cannot estimate the accuracy of this type of training.

Size Limitations

There are size limitations for training calls and data:

- The service accepts a maximum of 10,000 images or 100 MB per .zip file
- The service requires a minimum of 10 images per .zip file.
- The service accepts a maximum of 256 MB per training call.
- Minimum recommended size of an image is 32X32 pixels.

There are also size limitations when classifying images or detecting faces:

- Limitations for the methods to classify images:
 - Maximum image size is 10 MB.
 - Maximum .zip file size is 100 MB with up to 20 images.
- Limitations for the methods to detect faces:
 - Maximum image size is 10 MB.
 - Maximum .zip file size is 100 MB with up to 15 images.

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