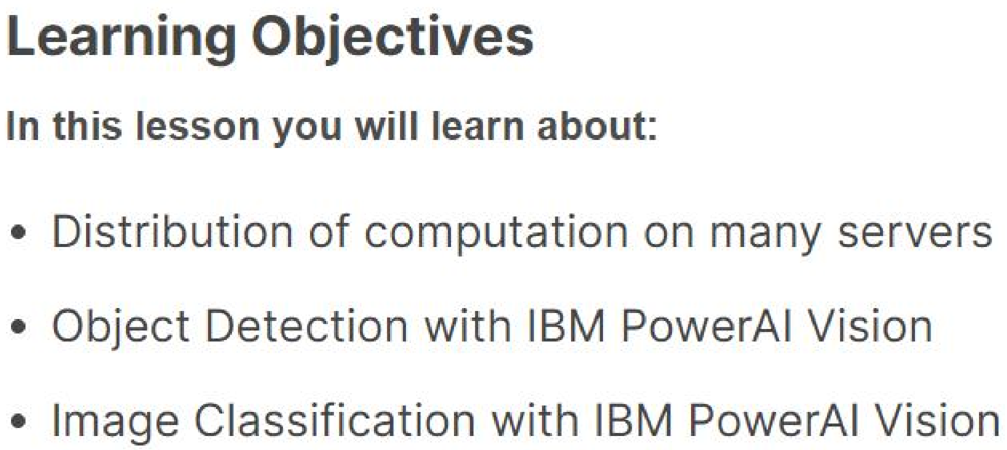
Module 4 - Advanced Deep Learning

- 4.1 Distributed Deep Learning
- 4.2 Deep Learning without Coding: PowerAI vision
- 4.3 Object recognition with PowerAI vision
- 4.4 Assignment - Make your object recognition model
- 4.5 Lab - Make you image classification model
- 4.6 Graded Review Questions (5 Questions)



Learning Objectives

In this lesson you will learn about:

- Distribution of computation on many servers
- Object Detection with IBM PowerAI Vision
- Image Classification with IBM PowerAI Vision

总结：

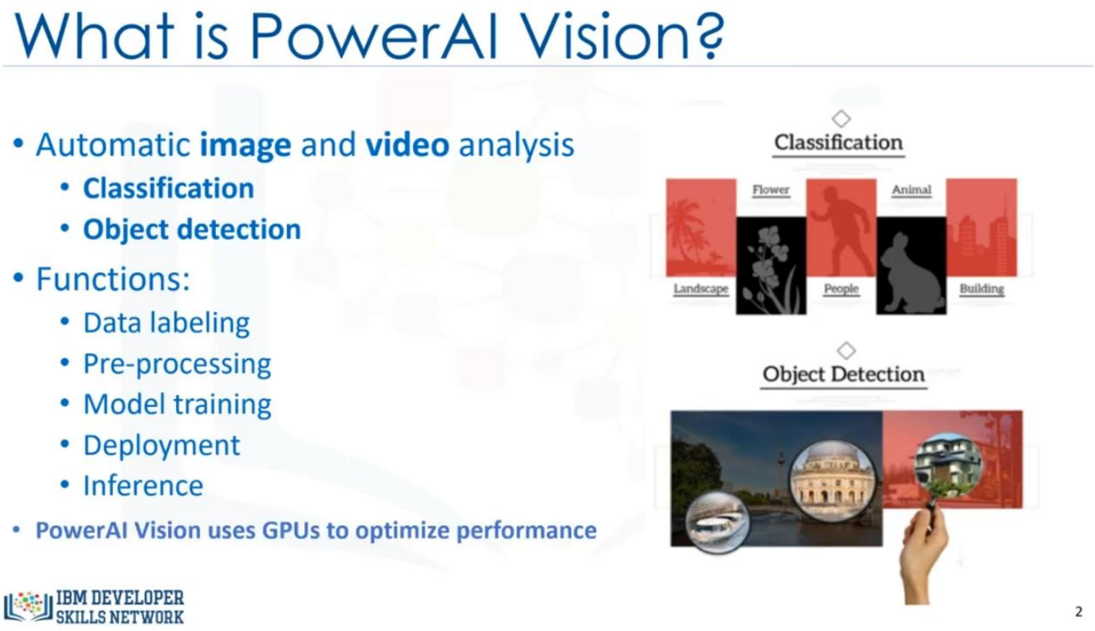
4.1 Distributed Deep Learning

详见下载下来的网页：
D:\KeepStudy\01_Edx\1 Using GPUs to Scale and Speed-up Deep Learning (IBM DL0122EN)**M4_1 Distributed Deep Learning.html**

D:\KeepStudy\01_Edx\1 Using GPUs to Scale and Speed-up Deep Learning (IBM DL0122EN)**M4_1 IBM Research achieves record deep learning performance with new software technology**

4.2 Deep Learning without Coding: PowerAI vision

If you would like to know more about PowerAI vision, you can register for a 3-days trial of PowerAI vision.
https://cocl.us/PowerAIvisionTrial

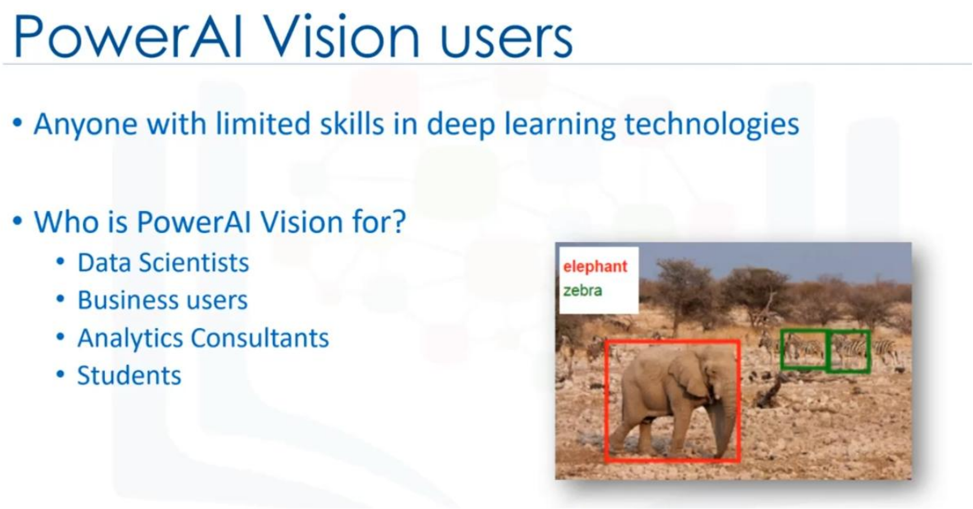


What is PowerAI Vision?

- Automatic **image** and **video** analysis
  - **Classification**
  - **Object detection**
- Functions:
  - Data labeling
  - Pre-processing
  - Model training
  - Deployment
  - Inference
- PowerAI Vision uses GPUs to optimize performance

Hello and Welcome! In this video, we'll show you how you can train your Deep Learning models without having to do any coding at all, by leveraging the **PowerAI Vision platform**.

PowerAI Vision is a development platform that performs automatic image and video analysis, without you having coding or deep learning expertise. The platform supports custom learning of classification and object detection for computer vision, and includes functions like: data labeling, pre-processing, model training, deployment, and inference. PowerAI Vision uses GPUs to optimize performance and accelerate these tasks.

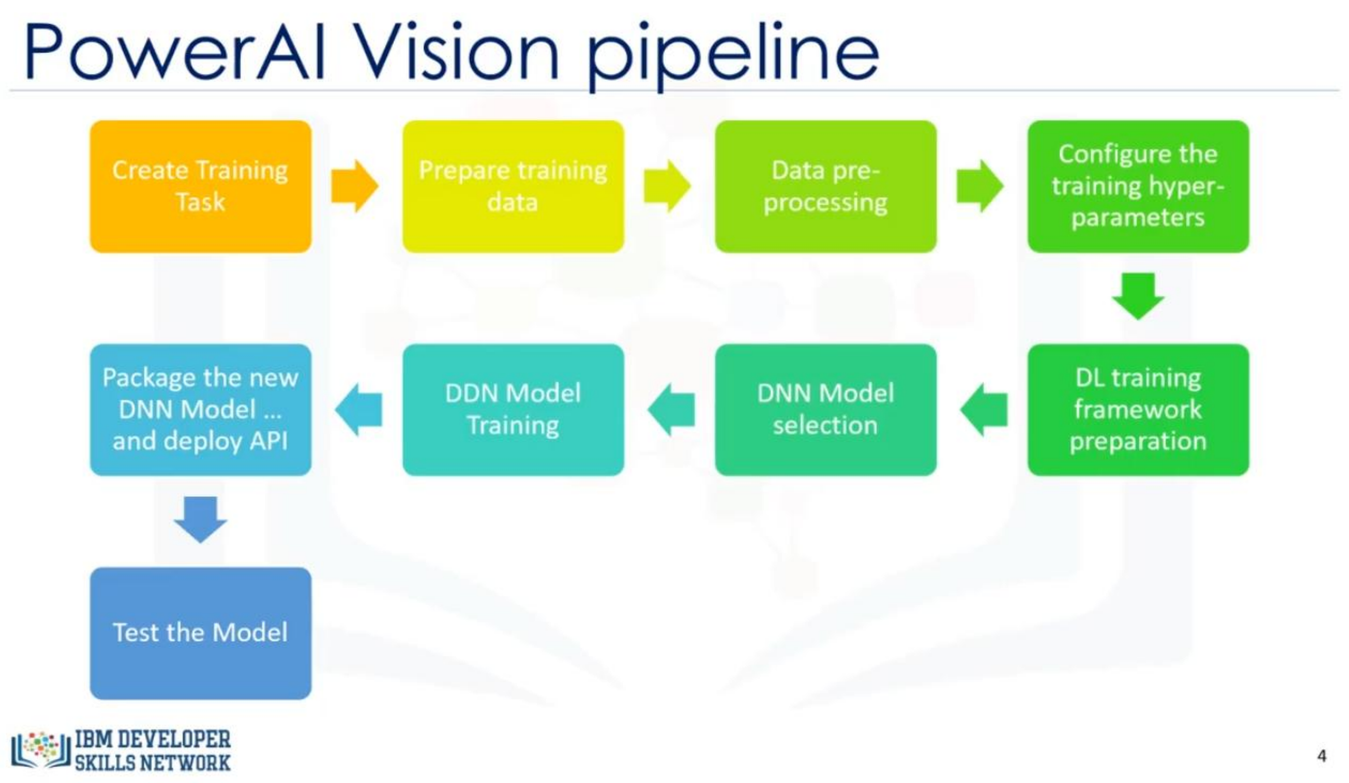


PowerAI Vision users

- Anyone with limited skills in deep learning technologies
- Who is PowerAI Vision for?
  - Data Scientists
  - Business users
  - Analytics Consultants
  - Students

PowerAI Vision includes tools and interfaces for anyone with limited skills in deep learning technologies; and it makes computer vision with deep learning more accessible to users. Data scientists, Business users, Analytics consultants, and Students alike, can benefit from this platform because it automates many of the monotonous and time-consuming tasks within deep learning.

总结：



PowerAI Vision pipeline

Create Training Task
Prepare training data
Data pre-processing
Configure the training hyper-parameters
DL training framework preparation
DNN Model selection
DDN Model Training
Package the new DNN Model … and deploy API
Test the Model

IBM DEVELOPER SKILLS NETWORK

A typical deep learning pipeline normally involves the following nine high-level Deep Learning tasks:

1. Creating the training task
2. Preparing the training data
3. Data pre-processing
4. Configuring the training hyper-parameters
5. Preparing the Deep Learning training framework
6. Selecting the Deep Neural Network Model (or DNN, for short)
7. Training the DNN Model
8. Packaging the new DNN Model … and then deploying the API
9. And the ninth and final step is Testing the Model PowerAI Vision performs many of these tasks automatically for you.

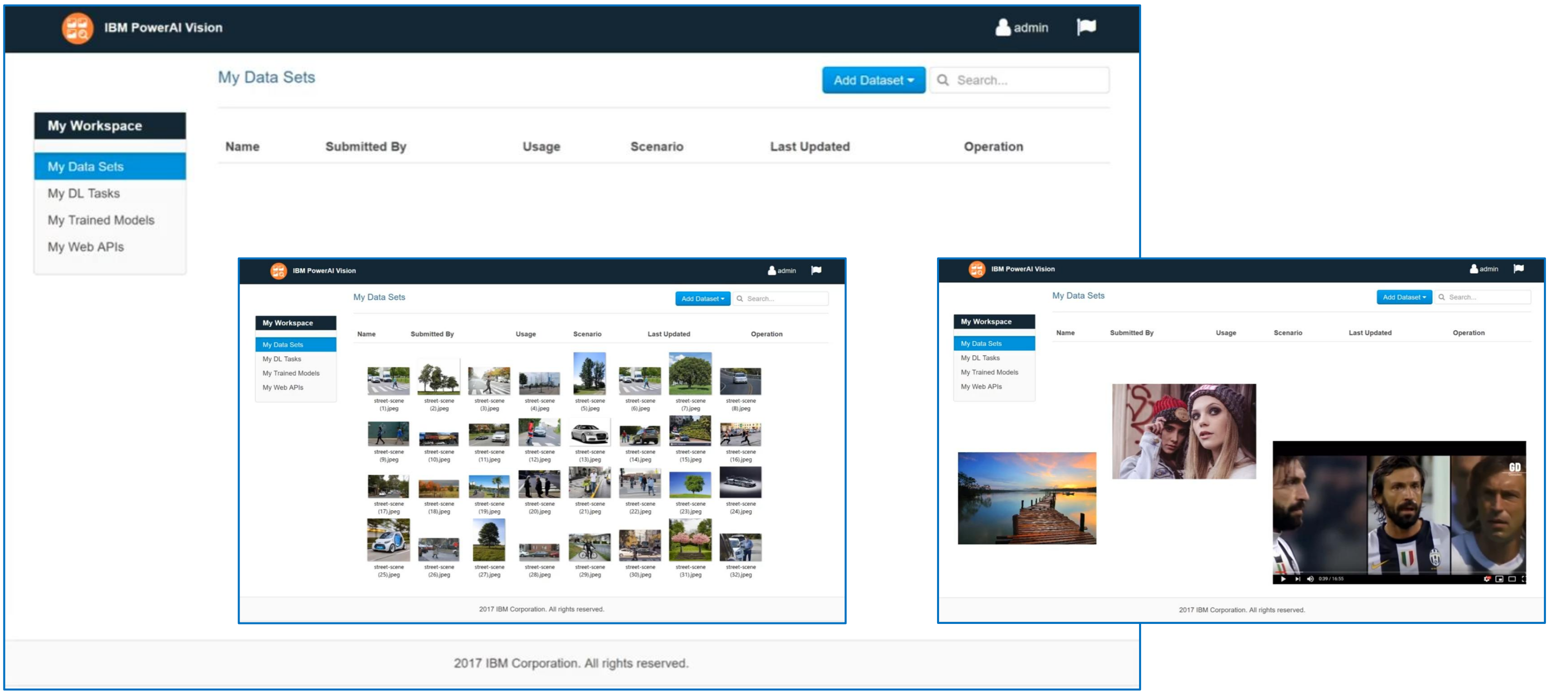
You simply need to upload the training dataset, including the labelled images, and PowerAI will create the model for you so that it's ready to deploy for use on test images -- all without you having to write a single line of code. View the Object Recognition with PowerAI Vision video for a demo of PowerAI.

Thanks for watching this video!

总结：

4.3 Object recognition with PowerAI vision

If you would like to know more about PowerAI vision, you can register for a 3-days trial of PowerAI vision.
https://cocl.us/PowerAIvisionTrial

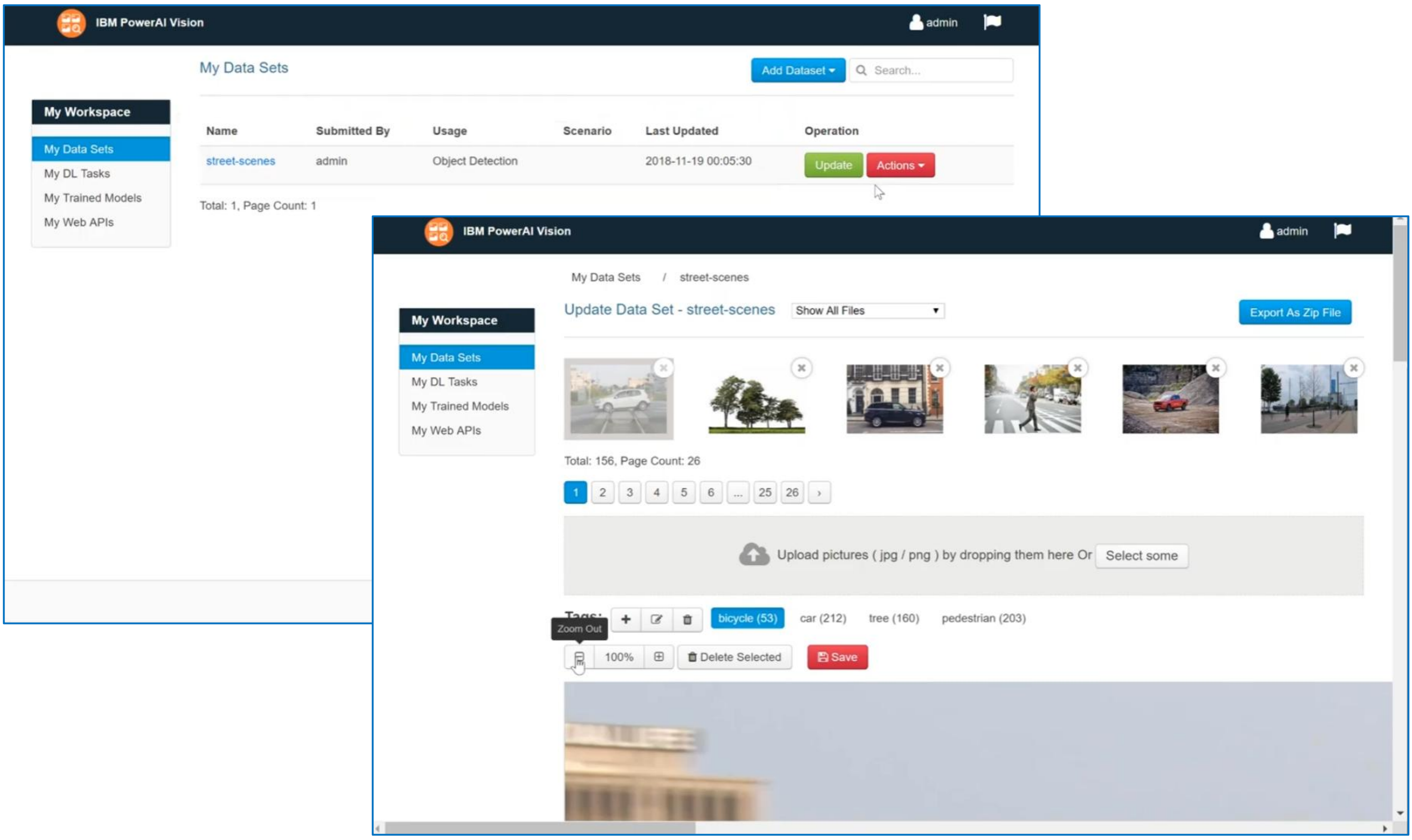


Name	Submitted By	Usage	Scenario	Last Updated	Operation

2017 IBM Corporation. All rights reserved.

Hello! In this video we'll give you a quick demo of PowerAI Vision, a tool that automates the building of deep learning models based on visual data, such as pictures and videos. So, let's see how it works!

This is the PowerAI Vision interface that displays when you click the link from the registration email. Let me show you how easy it is to quickly build a model that detects different objects from a dataset of street scene pictures that contain cars, pedestrians, bicycles, and trees, among other things. Specifically, I'd like to show you how self-driving cars can detect the objects on the street. This is a simple demo, and you can try it out for yourself for detecting other types of objects, for instance, landscapes in images, or people's faces in photos, or even recognizing soccer players in a video.



Name	Submitted By	Usage	Scenario	Last Updated	Operation
street-scenes	admin	Object Detection		2018-11-19 00:05:30	Update Actions ▾

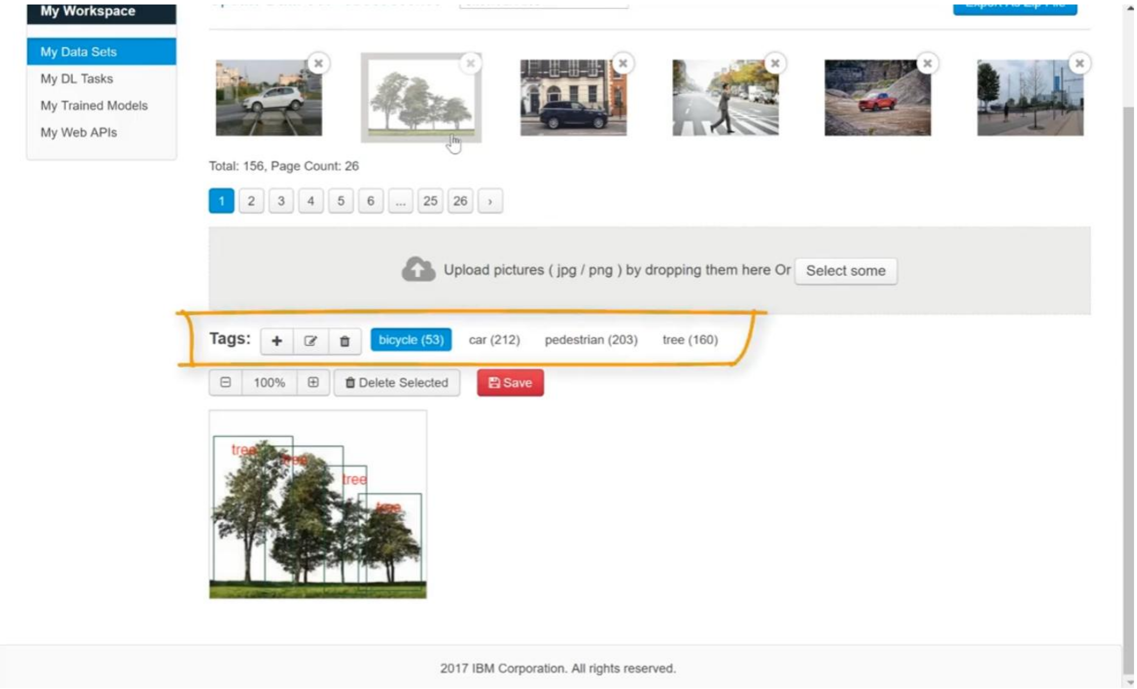
Total: 1, Page Count: 1

Total: 156, Page Count: 26

Upload pictures (jpg / png) by dropping them here Or Select some

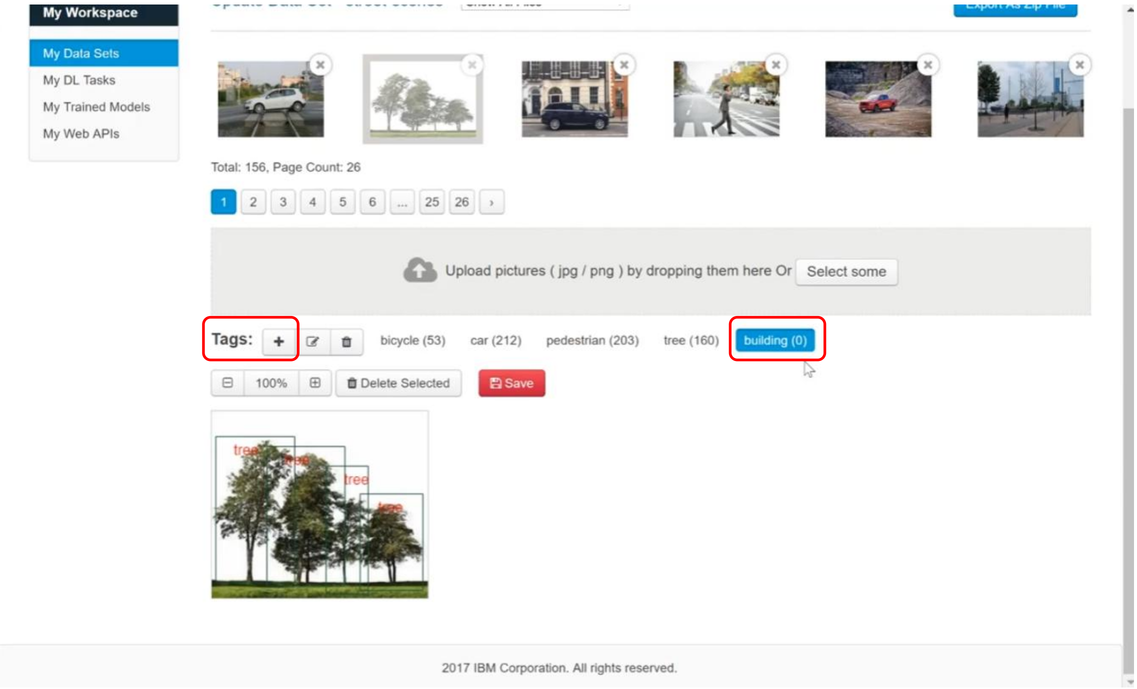
Tags: bicycle (53) car (212) tree (160) pedestrian (203)

总结：



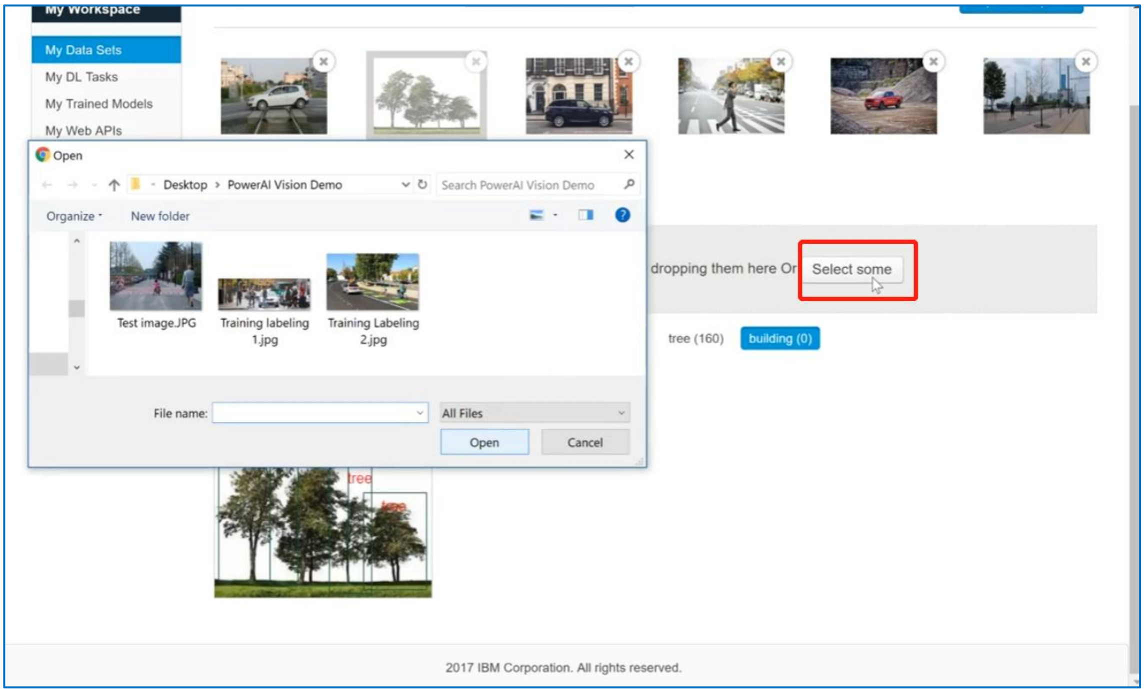
Getting back to our street scene demo, we have already built, labelled, and uploaded a small dataset of street scene pictures, in a dataset called street-scenes, as shown here. Let's click on the dataset. You can see some of the pictures, which have already been labelled. The first one is of a car; we'll zoom in and scroll down to see it. And here's the label. Then, another car. Here's a pedestrian and a small car. Here's a truck labelled as a car. (We can go back and label it as a truck in future.) And in this picture we have two trees and a pedestrian. In the second picture, please note that the trees are labelled individually.

Now let's look at the Tags. As you can see on the Tags bar, we have already created 4 different tags that will identify the different types of objects found in our street pictures. If there's a new object that we want the model to find in this same dataset of pictures, such as buildings, we simply add that tag. Note that you'll need to go through all the pictures to label that new object in each picture where a building is found.

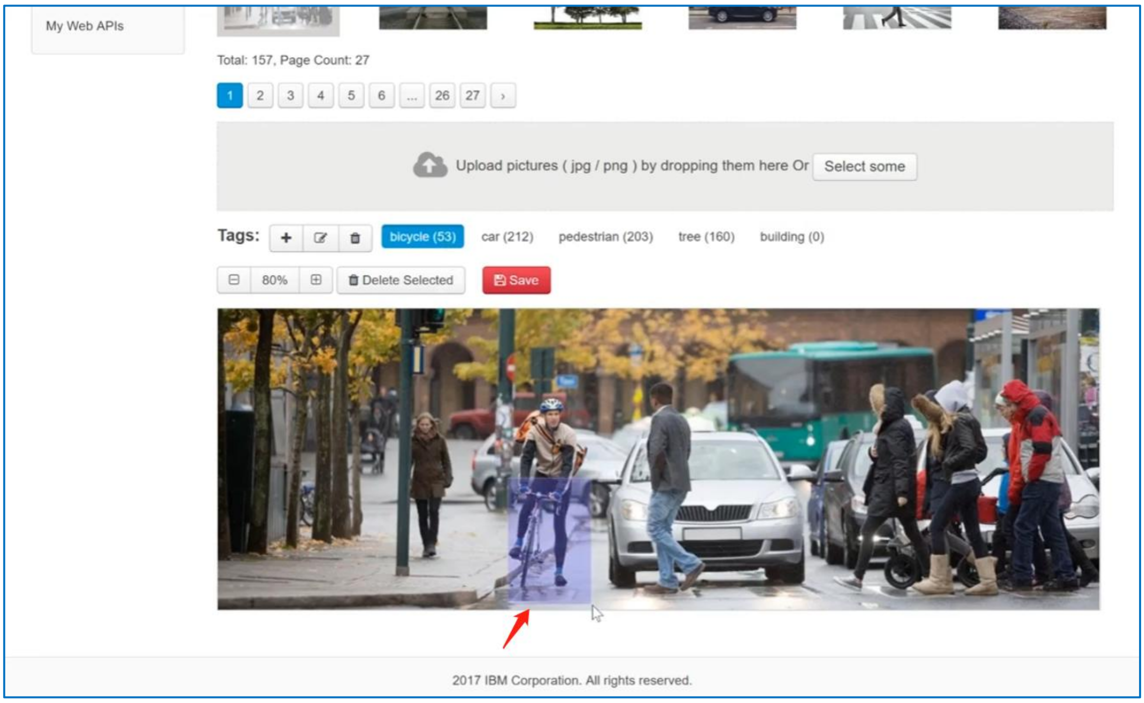


So, let's add the new "building" tag. Simply click the **+ button** next to the Tags heading, as I'm doing here, and add the new tag, "building". You can also delete the tags, but if you do, you will get a warning that all corresponding labels in the dataset will be deleted as well in all the images where you've used that tag.

总结：



2017 IBM Corporation. All rights reserved.



Total: 157, Page Count: 27

Upload pictures (jpg / png) by dropping them here Or Select some

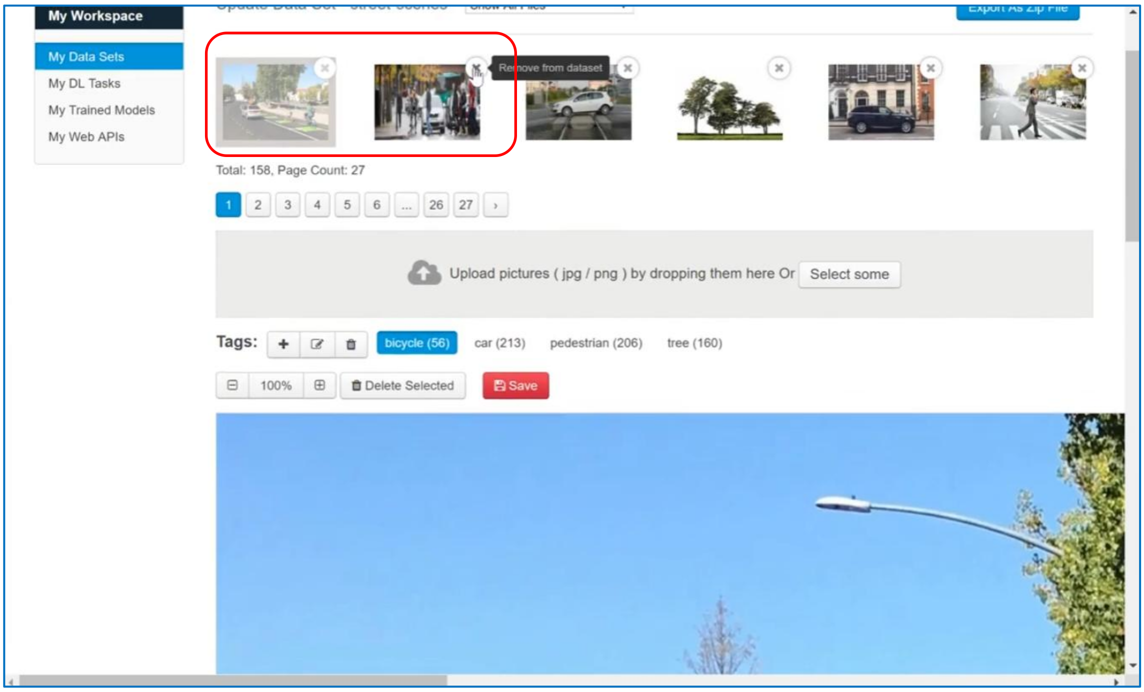
2017 IBM Corporation. All rights reserved.

Now let's add two more pictures to this dataset, and label them using some of these tags. Click the **"Select some" button**, then select the first new image to be added and labelled.
Once the new picture is loaded and displays in PowerAI Vision, first select the tag that corresponds to the object you want to label; we'll start with bicycle. So select the bicycle tag, and then simply draw a bounding box around that object in the image. Here, we just want the bicycle. Now we'll select the pedestrian and label some of the pedestrians. Note that when labelling objects, select one object for each label; for example, if your picture has 3 cars, you would label each of them separately. And now we'll label the car. Note that if you change your mind on any of the labels you add, you can move or resize the bounding box by dragging the whole box - or just the border - to where you want it. Or if you change your mind altogether, you can even delete it by selecting the bounding box you want to delete, and clicking the **"Delete Selected" button**, as I've done here, Then we click **Save**.

Let's try labelling another picture. Again, after it displays, follow the same simple process. Select the tag and identify the corresponding object in the picture; in this picture I'd like to label just the two bicycles and the car; and then I click Save. It's as simple as that!

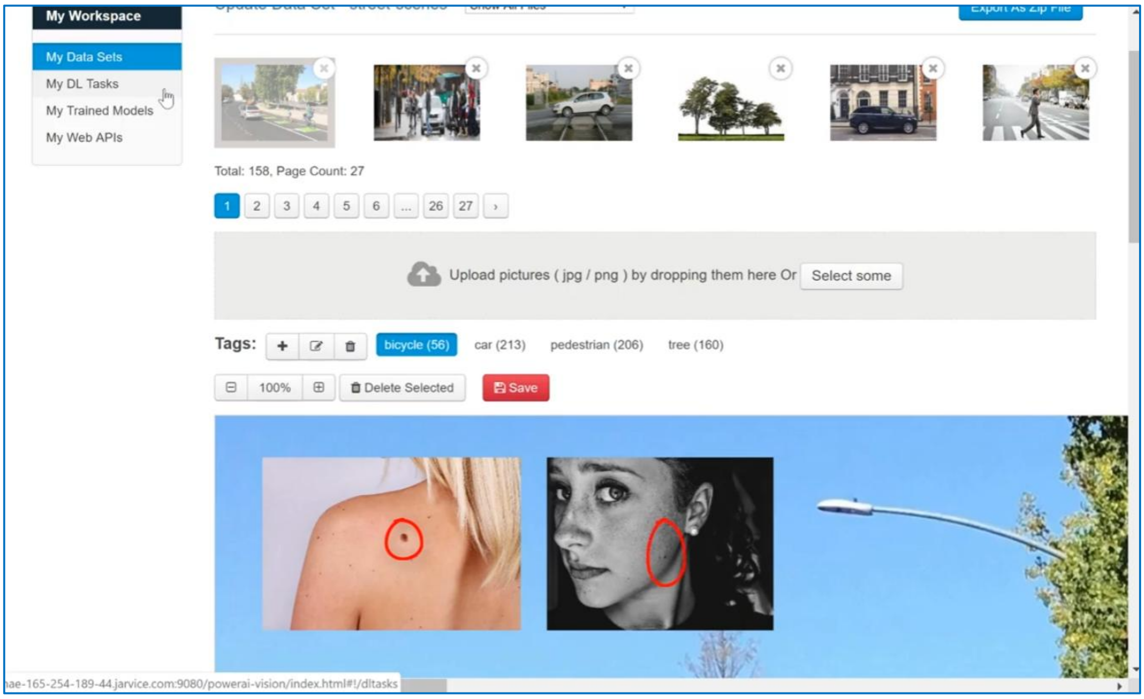
Remember that you can constantly edit and correct the labelling you've applied in your dataset, or add new tags or new pictures as I've shown here.

总结：



Total: 158, Page Count: 27

Upload pictures (jpg / png) by dropping them here Or



Total: 158, Page Count: 27

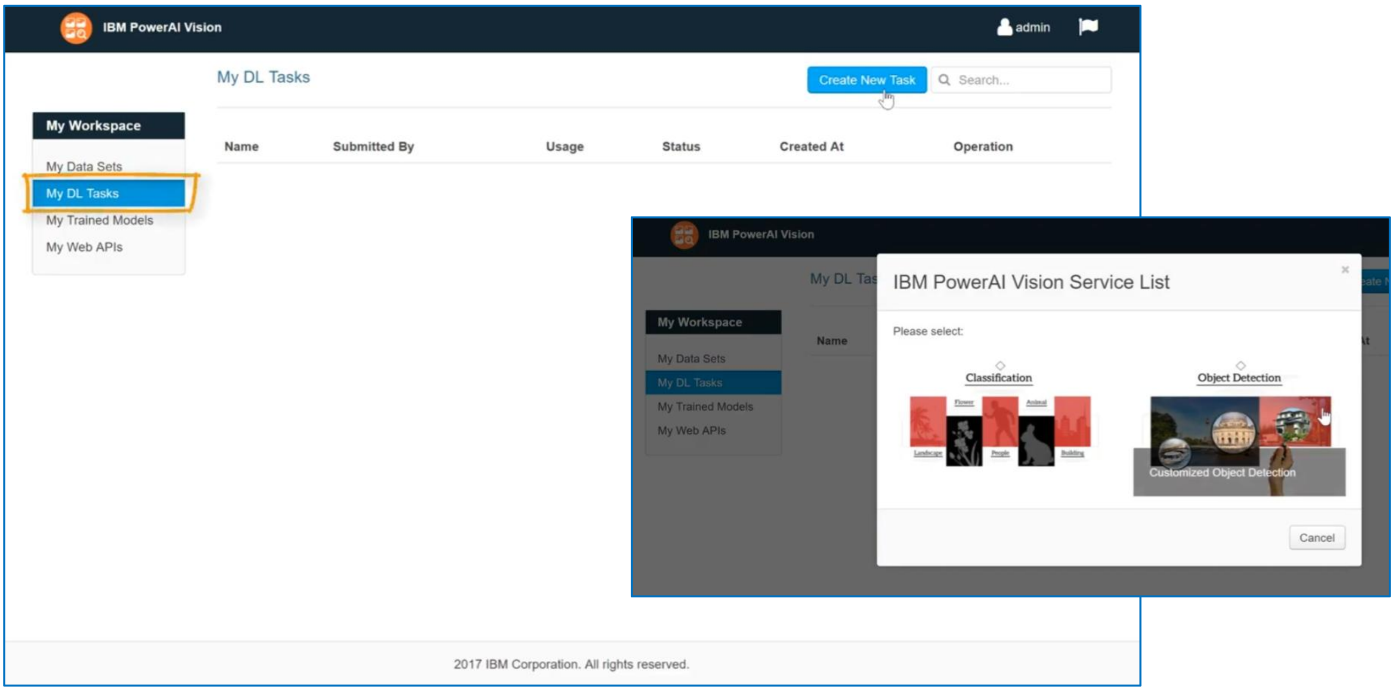
Upload pictures (jpg / png) by dropping them here Or

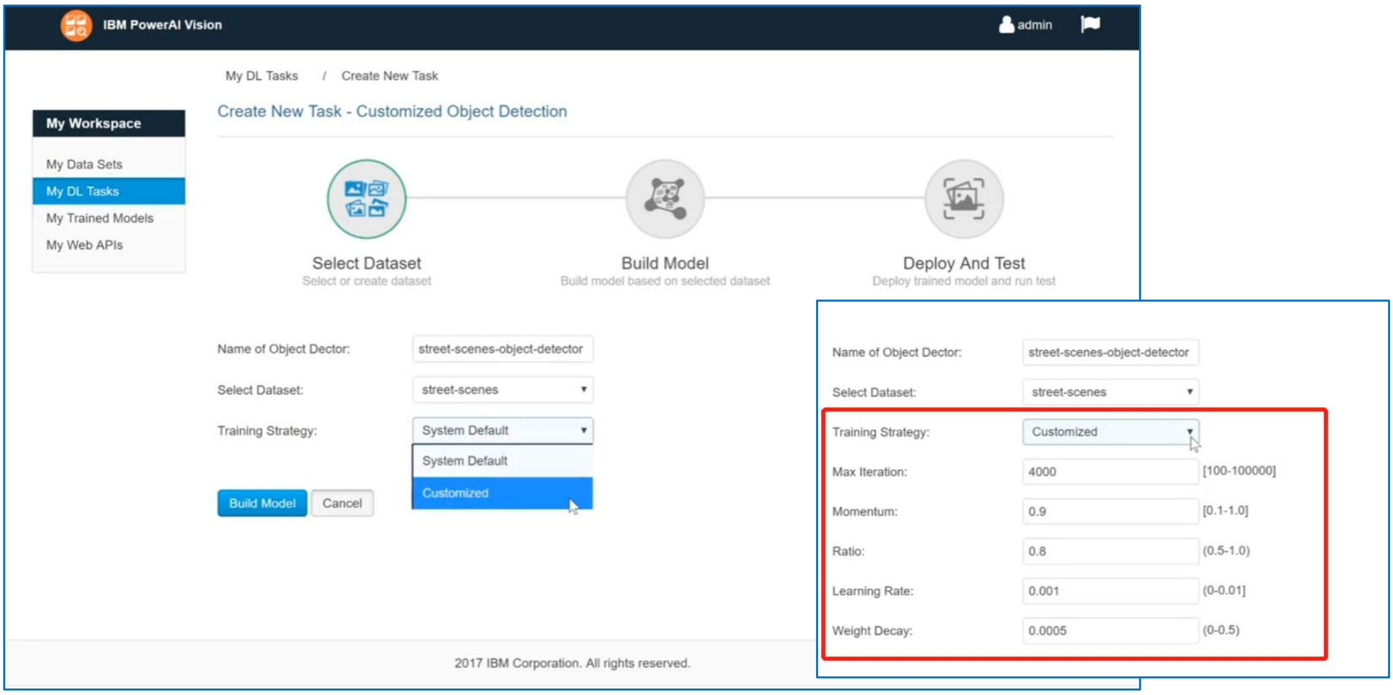
Please note, that you can delete any of the pictures in the dataset, by clicking on the x on the top right corner of the picture you want to delete. These two new labelled images are now part of the street-scene dataset. Our model will be trained on all the images and corresponding labels that we've added. So, as you can see, it's really quite simple to build a training dataset.

An important thing to note is that the labelling can and should be done by the people who understand the subject matter the best, and who know what the objects represent. In these images, anyone can label cars, pedestrians, bicycles and trees, but for more serious subjects or tag names, such as labeling pictures of different moles for cancer detection models, you would really want someone who is trained and understands the different types of moles and different types of models and, thus, can label the pictures accurately and concisely.

And to do all of this, all they would need is their computer and a browser! They don't need to be a data scientist to label pictures accurately.

总结：





Name of Object Dector: street-scenes-object-detector

Select Dataset: street-scenes

Training Strategy: System Default

Name of Object Dector: street-scenes-object-detector

Select Dataset: street-scenes

Training Strategy: Customized

Max Iteration: 4000 [100-100000]

Momentum: 0.9 [0.1-1.0]

Ratio: 0.8 (0.5-1.0)

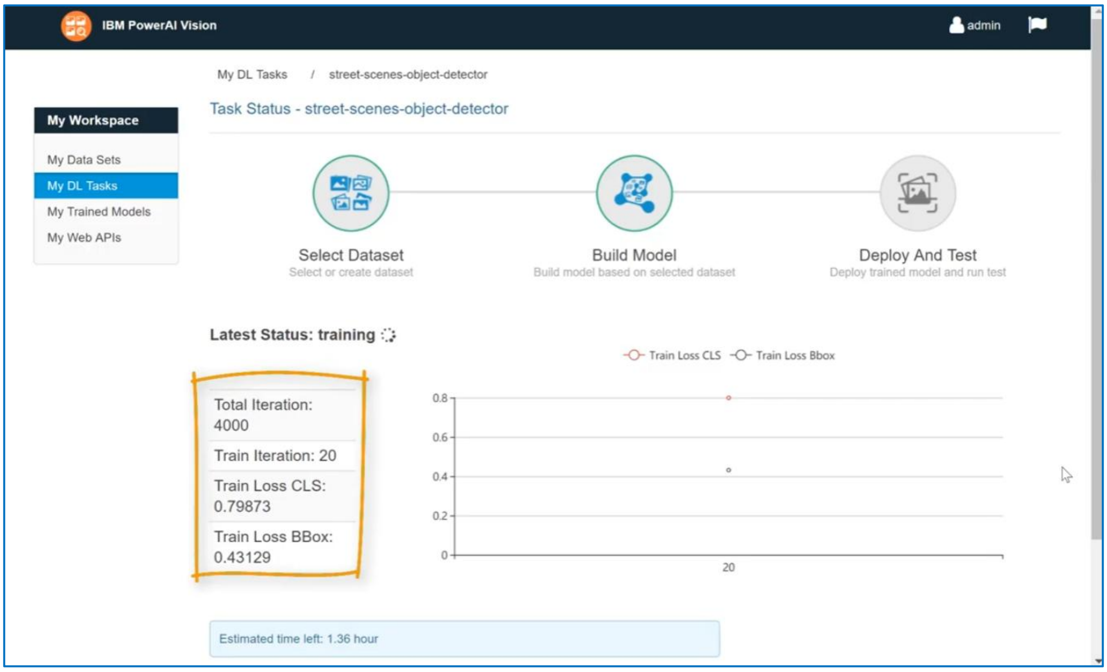
Learning Rate: 0.001 (0-0.01]

Weight Decay: 0.0005 (0-0.5)

Next, to create the deep learning model that can identify street scene objects, we'll need to create a new, deep learning task, which is quite simple. All we need to do is create a new task, selecting Object Detection and giving the task a name. In this case, I will call the task "street-scene-object-detector" so that we know what the resulting model can do.

Note that you can simply use the system defaults, or you can customize the process to determine how many iterations of the training to go through, and which parameters to use. For this demo, let's use the system defaults because they work quite well. Then, confirm that you want to build your model. And the building of the deep learning model starts.

总结：



Task Status - street-scenes-object-detector

Latest Status: training

Total Iteration: 4000

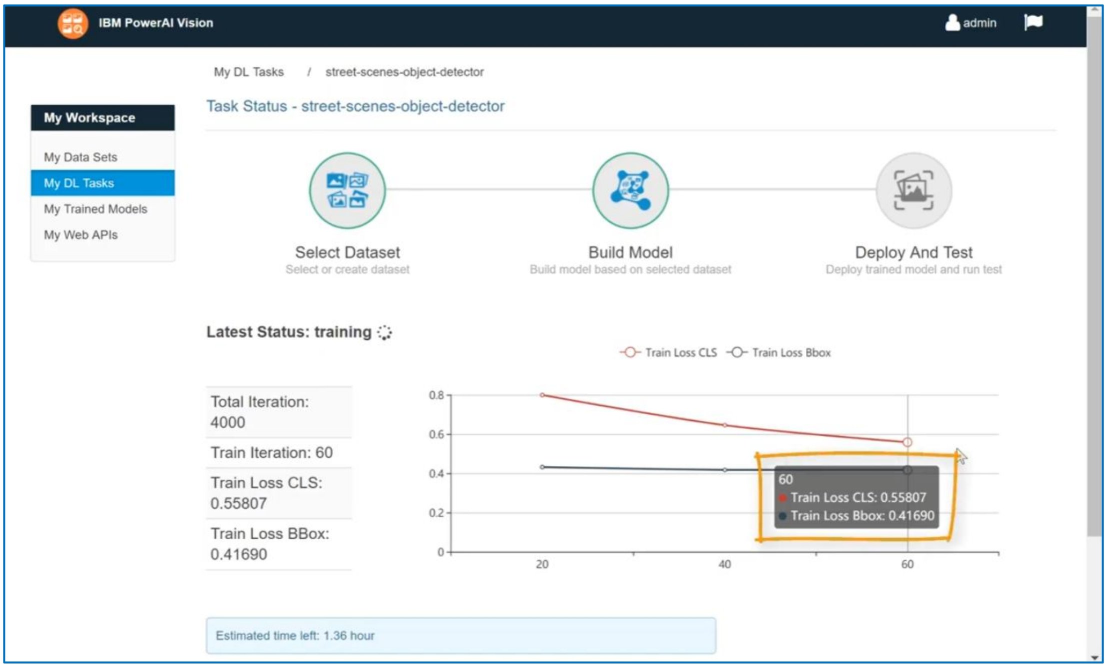
Train Iteration: 20

Train Loss CLS: 0.79873

Train Loss BBox: 0.43129

Estimated time left: 1.36 hour

In the background, what's happening is that 75% of the images in the dataset are being used for training, and the other 25% will be used for testing the model, even though all the images are labelled. That is, the labels in the testing portion are not used in the training of the model. So, as PowerAI Vision builds this model and as it's being trained, it’s using the training images in the dataset itself for training, and the rest for testing, and iterating them 4,000 times, which is the default number of iterations it’ll go through, if you select “System defaults”. This is to build up new insights on new patterns in those images.



Task Status - street-scenes-object-detector

Latest Status: training

Total Iteration: 4000

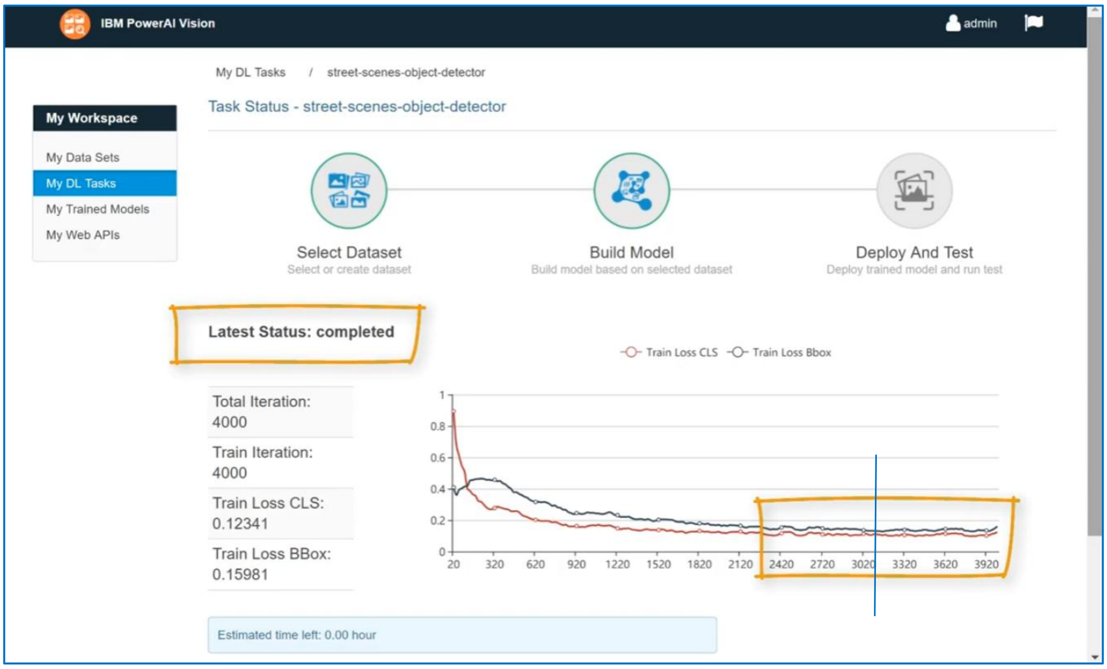
Train Iteration: 60

Train Loss CLS: 0.55807

Train Loss BBox: 0.41690

Estimated time left: 1.36 hour

PowerAI Vision is also regularly comparing them to the test dataset that we've already tagged, to confirm how accurate the model is. And this is all reported back through the process as it goes through each iteration. PowerAI Vision displays the accuracy, as shown here, when you hover over the lines of the graph. This is showing you how many images were lost through the training, and also how many of the test images were misidentified. As PowerAI Vision continues to build the model, this graph will extend until it has reached the 4,000 iterations, showing you accuracy levels, as well as progress over time, at set intervals.



Task Status - street-scenes-object-detector

Latest Status: completed

Total Iteration: 4000

Train Iteration: 4000

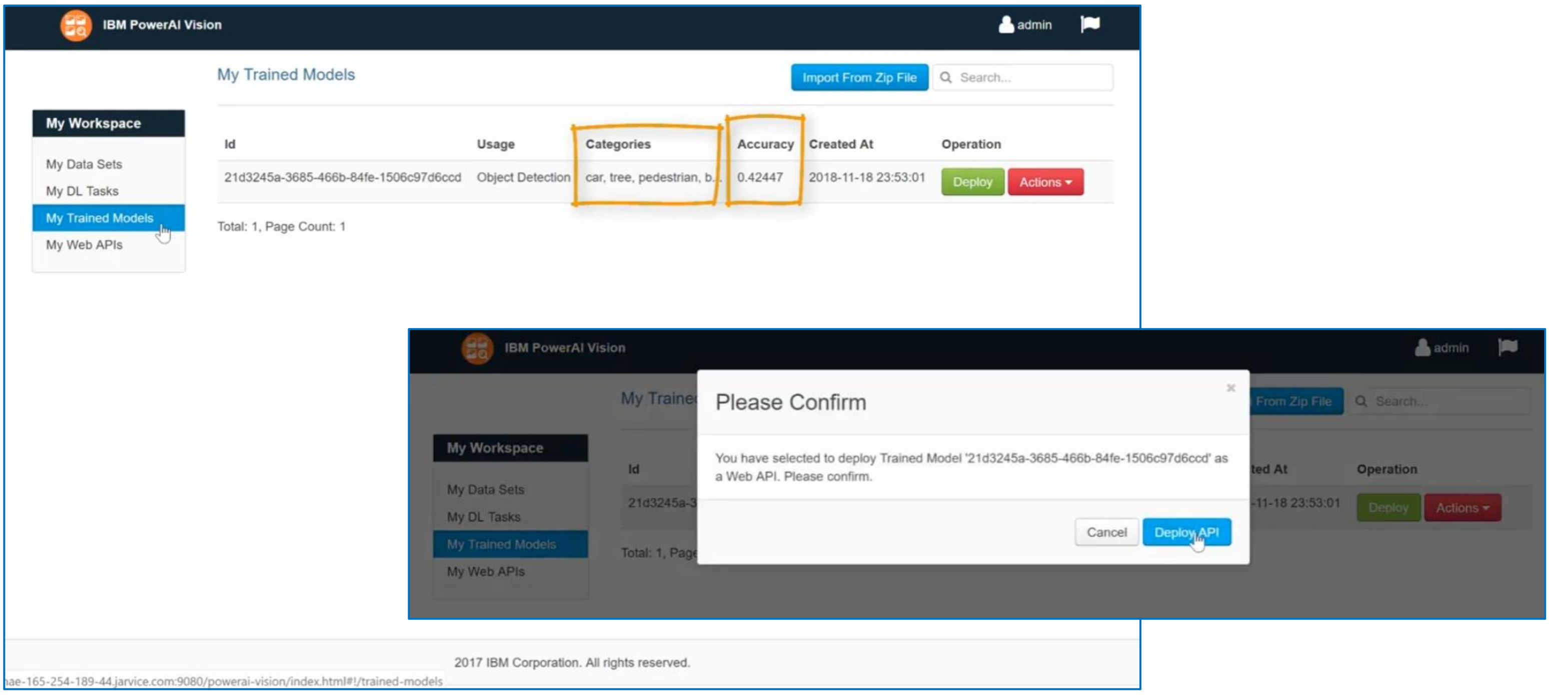
Train Loss CLS: 0.12341

Train Loss BBox: 0.15981

Estimated time left: 0.00 hour

Once the training is finished or completed, as shown here, at the end of the graph, you can see that starting at the mid-point where the two lines converge, the graph shows that the loss is dramatically reduced. In fact, we can also see from this graph that we didn't need the full 4,000 iterations; 3,000 iterations, or so, would have sufficed.

总结：



My Trained Models

Id	Usage	Categories	Accuracy	Created At	Operation
21d3245a-3685-466b-84fe-1506c97d6ccd	Object Detection	car, tree, pedestrian, b...	0.42447	2018-11-18 23:53:01	Deploy Actions

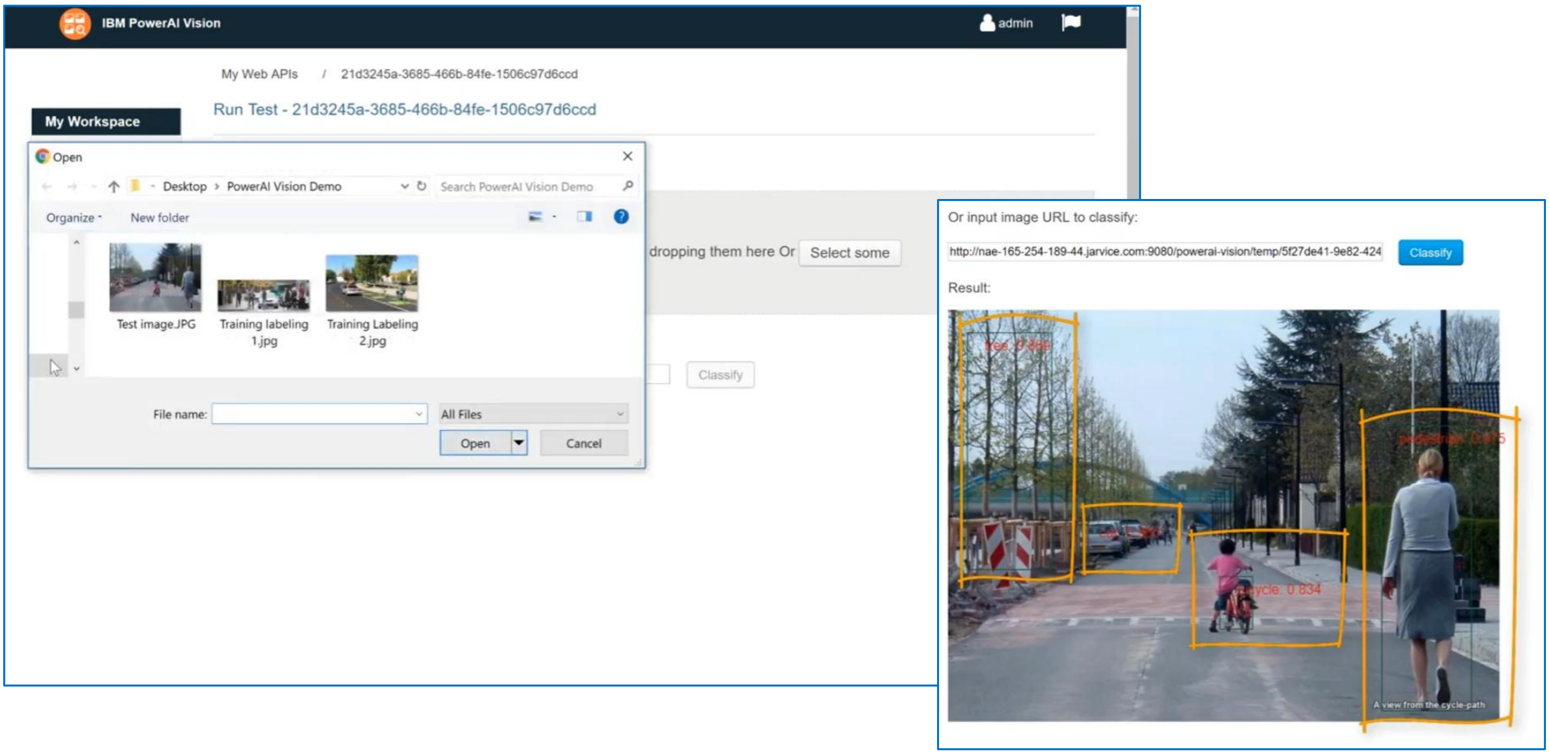
Total: 1, Page Count: 1

Please Confirm

You have selected to deploy Trained Model '21d3245a-3685-466b-84fe-1506c97d6ccd' as a Web API. Please confirm.

2017 IBM Corporation. All rights reserved.

nae-165-254-189-44.jarvice.com:9080/powerai-vision/index.html#!/trained-models



My Web APIs / 21d3245a-3685-466b-84fe-1506c97d6ccd

Run Test - 21d3245a-3685-466b-84fe-1506c97d6ccd

Or input image URL to classify:

http://nae-165-254-189-44.jarvice.com:9080/powerai-vision/temp/5f27de41-9e82-424

Classify

Result:

Once the model is trained, you can see it in the “Trained Models” section, as shown here. You can see what objects it was trained to detect, and it will give you an indication of how accurate it is, based on the information that we provided. That is, based on the results of the test images that were held back for testing.

Now, once it’s trained, we need to deploy the model to make it ready for use, that is, accessible to programmers and developers who can use it to build other applications. In the deploy phase, deploying the model is as simple as clicking the Deploy button from the My Trained Model section.

So, in just a few seconds, we already have a web-facing API that anyone can use, and we’ll use it now, ourselves to test an image that the model has not seen before. Let’s try posting a new unseen image directly to this API to see what objects it identifies. We select a picture that the model hasn’t seen before that contains all or some of the objects that were labelled. The response from the API comes back in JSON format, showing which objects have been detected. As shown here, the numbers after each of the labels indicate the confidence level. From these confidence levels, we can be confident that the model accurately identified the objects that it was trained to detect, with a high confidence level on 4 objects! We can conclude that our model is indeed accurate at selecting objects from street scenes, such as cars, pedestrians, bicycles and trees. Not only that, but it can do so in a matter of seconds! So now you’ve seen the whole process, from beginning to end, and you saw exactly how quick and easy it was to create, deploy and test your model.

In summary, you’ve seen how Power AI Vision, can be used to: create a data set; identify some objects in that dataset, tag and label those objects; build a deep learning model based on that information; and deploy the model in a way that anyone can use! So, now you know how easy it would be to quickly get insights from your own pictures and videos.
Thank-you for watching this video!

总结：

4.4 Assignment - Make your object recognition model

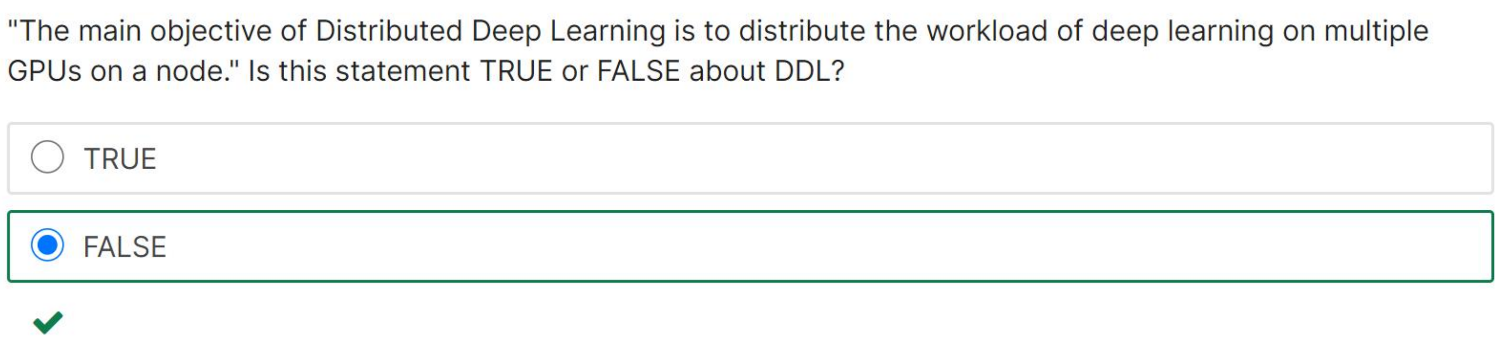
详见下载下来的网页：
D:\KeepStudy\01_Edx\1 Using GPUs to Scale and Speed-up Deep Learning (IBM DL0122EN)**M4_4 Assignment - Make your object recognition model.html**

4.5 Lab - Make you image classification model

详见下载下来的网页：
D:\KeepStudy\01_Edx\1 Using GPUs to Scale and Speed-up Deep Learning (IBM DL0122EN)**M4_5 Lab - Make you image classification model.html**
以及 Web API html 模板和数据集：
D:\KeepStudy\01_Edx\1 Using GPUs to Scale and Speed-up Deep Learning (IBM DL0122EN)\labs\M4_5 Lab - Make you image classification model

总结：

4.6 Graded Review Questions (5 Questions)

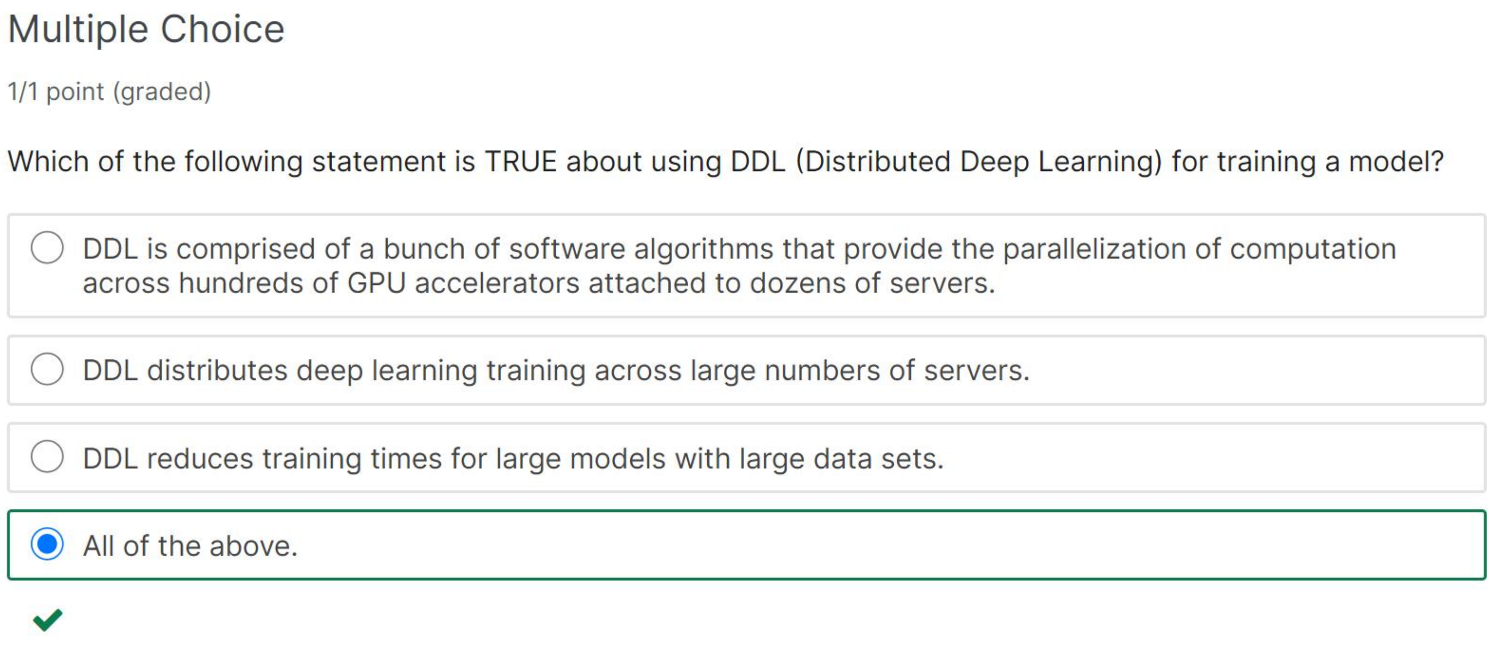


"The main objective of Distributed Deep Learning is to distribute the workload of deep learning on multiple GPUs on a node." Is this statement TRUE or FALSE about DDL?

○ TRUE

● FALSE

✓



Multiple Choice

1/1 point (graded)

Which of the following statement is TRUE about using DDL (Distributed Deep Learning) for training a model?

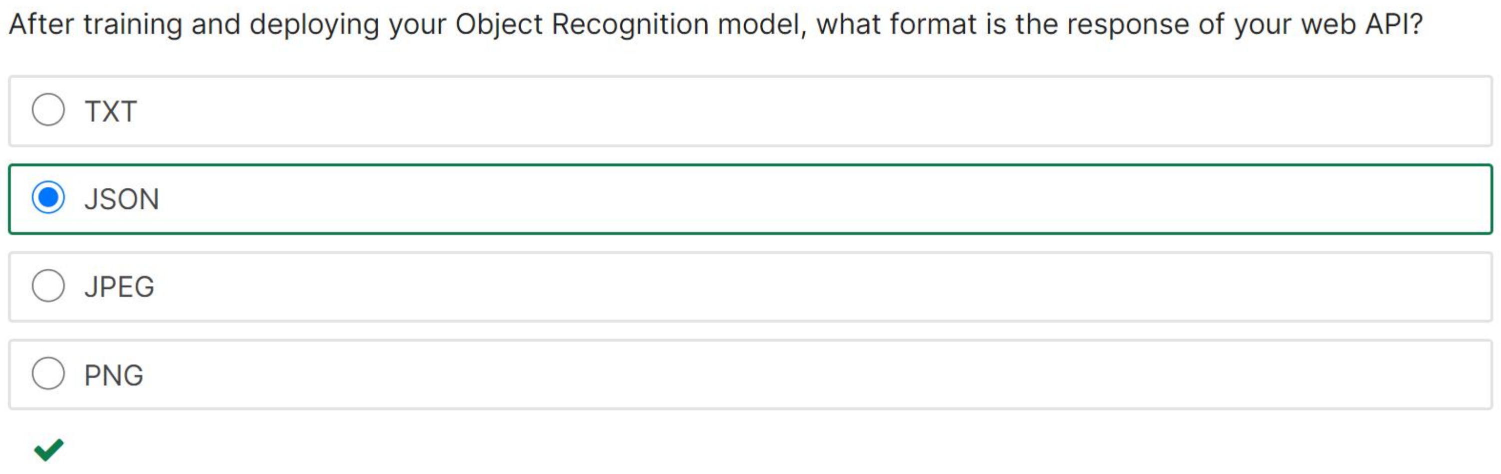
○ DDL is comprised of a bunch of software algorithms that provide the parallelization of computation across hundreds of GPU accelerators attached to dozens of servers.

○ DDL distributes deep learning training across large numbers of servers.

○ DDL reduces training times for large models with large data sets.

● All of the above.

✓



After training and deploying your Object Recognition model, what format is the response of your web API?

○ TXT

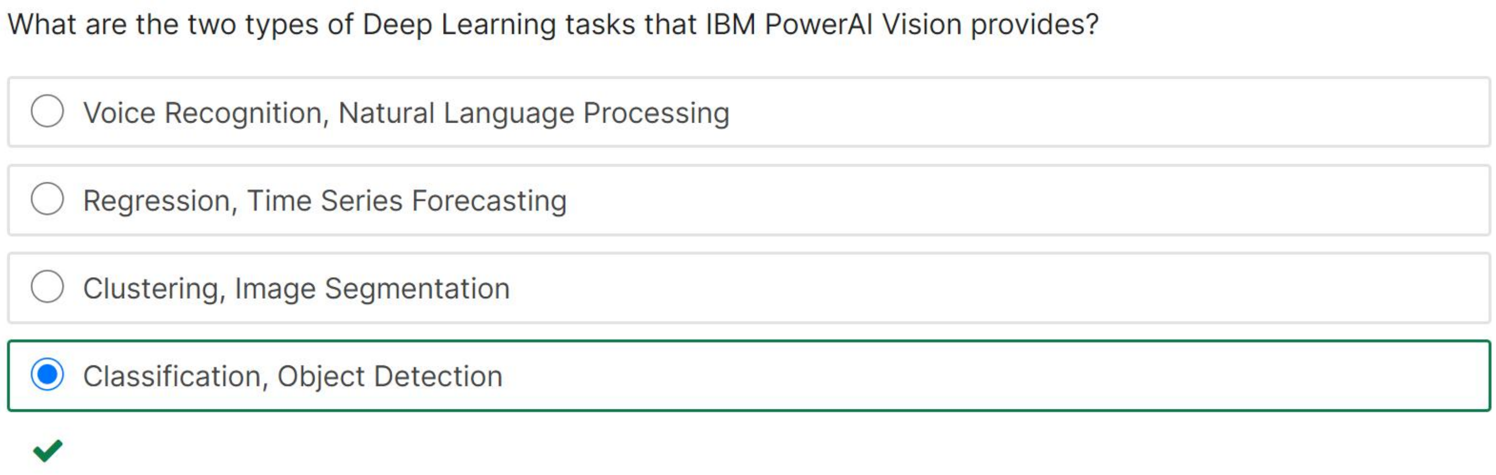
● JSON

○ JPEG

○ PNG

✓

总结：



What are the two types of Deep Learning tasks that IBM PowerAI Vision provides?

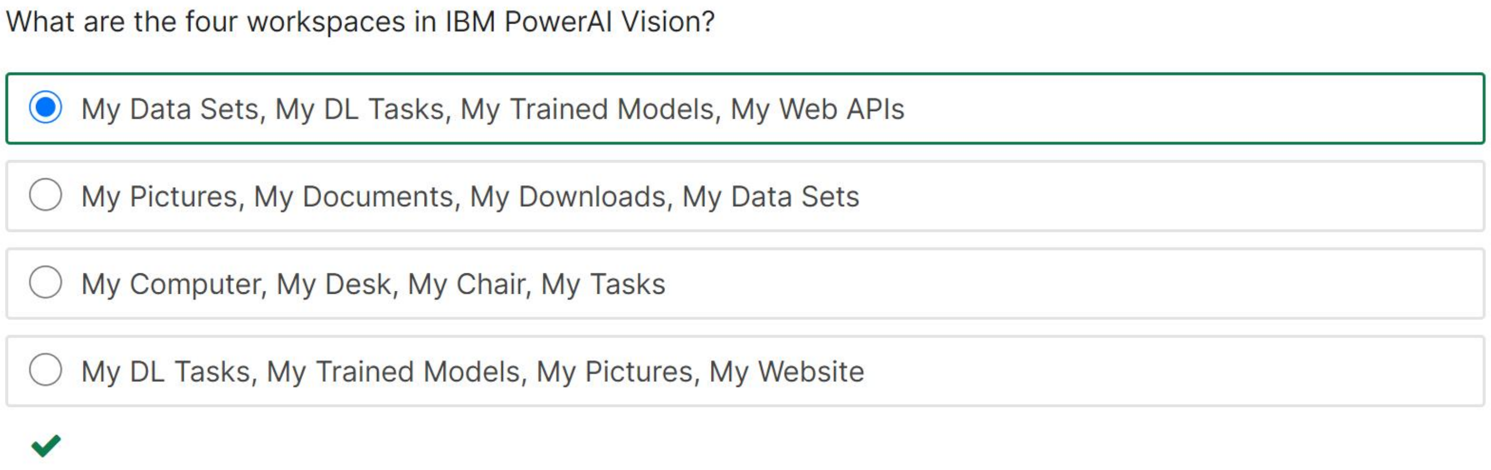
○ Voice Recognition, Natural Language Processing

○ Regression, Time Series Forecasting

○ Clustering, Image Segmentation

● Classification, Object Detection

✓



What are the four workspaces in IBM PowerAI Vision?

● My Data Sets, My DL Tasks, My Trained Models, My Web APIs

○ My Pictures, My Documents, My Downloads, My Data Sets

○ My Computer, My Desk, My Chair, My Tasks

○ My DL Tasks, My Trained Models, My Pictures, My Website

✓

总结：