

IBM PowerAI Vision

User Guide

TECHNOLOGY PREVIEW v4.0

By IBM Cognitive Systems Group
January 2018



Table of Contents

General Information	2
Main Features	2
Terms and Conditions	2
Technology Preview Capabilities	3
Architectural Overview	4
PowerAI Vision User Interface	5
My Data Sets	5
Copy From Sample Dataset	5
Create a New Dataset	6
Exporting and Importing Data Sets	9
My DL Task	10
My Trained Models	11
My Web APIs	11
Validate Trained Model	11
Using a REST API for Inference	12
Video Object Detection and Labeling	13
Overview	13
Data Set Management	13
Video Operations	14
Video Label Editor	15
Hands-On: Classification with PowerAI Vision	16
Data Preparation	16
Uploading a Data Set	17
Creating a New Training Task	19
Testing the Trained Model	20
Hands-On: Detecting Objects with PowerAI Vision	22
Data Preparation	22
Uploading a Data Set	22
Creating a New Training Task	25
Testing the Trained Model	26
Hands-On: Detecting Objects in Videos with PowerAI Vision	28
Data Preparation	28
Uploading a Data Set	28
Label Video Training Data	30
Export Labels for Training	31
Train an Object Detector Based on Labeled Video Data	32

Testing the Trained Model on Video	33
Autonomous Labeling of Video	35
Appendix A. System Requirements	36
Hardware Requirements	36
Software Requirements	36
Appendix B. Installation and Validation	37
Installation Guide	37
References	39
IBM Copyright Information	40

License Information

The Programs listed below are licensed under the following License Information terms and conditions in addition to the Program license terms previously agreed to by Client and IBM. If Client does not have previously agreed to license terms in effect for the Program, the International License Agreement for Early Release of Programs (Z125-5544-05) applies.

Program Name (Program Number):

IBM PowerAI Vision Technology Preview 1.0, 2.0, 3.0, 4.0 (Early Release)

The following standard terms apply to Licensee's use of the Program.

Test Period

The test period begins on the date that Licensee agrees to the terms of this Agreement and ends on 2018-08-03.

Prohibited Uses

Licensee may not use or authorize others to use the Program, alone or in combination with other services or products, in support of any of the following high risk activities: design, construction, control, or maintenance of nuclear facilities, mass transit systems, air traffic control systems, automotive control systems, weapons systems, or aircraft navigation or communications, or any other activity where program failure could give rise to a material threat of death or serious personal injury.

Export and Import Restrictions

This Program may contain cryptography. Transfer to, or use by, users of the Program may be prohibited or subject to export or import laws, regulations or policies, including those of the United States Export Administration Regulations. Licensee assumes all responsibility for complying with all applicable laws, regulations, and policies regarding the export, import, or use of this Program, including but not limited to, U.S. restrictions on exports or reexports. To obtain the export classification of this Program refer to: <https://www.ibm.com/products/exporting/>.

Third Party Data and Services

The Program may contain links to or be used to access third party data services, databases, web services, software, or other third party content (all, "content"). Access to this content is provided "AS-IS", WITH NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING THE WARRANTY OF TITLE, NON-INFRINGEMENT OR NON-INTERFERENCE AND THE IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Access can be terminated by the relevant third parties at their sole discretion at any time. Licensee may be required to enter into separate agreements with the third parties for the access to or use of such content. IBM is not a party to any such separate agreements and as an express condition of this license Licensee agrees to comply with the terms of such separate agreements.

L/N: L-MCHR-APSJM5

D/N: L-MCHR-APSJM5

P/N: L-MCHR-APSJM5

General Information

The IBM PowerAI Vision platform built on cognitive infrastructure is a new generation of video/image analysis platform. The platform offers built-in deep learning models that learn to analyze images and video streams for classification and object detection.

IBM PowerAI Vision provides tools and interfaces for business analysts, subject matter experts and developers with limited skills in deep learning technologies. The tools assist users to pre-process datasets and label them so they can be utilized to train and validate a model to be deployed in customized solutions that demand image classification and object detection.

IBM PowerAI Vision is supported on Power Systems and leverages GPUs to optimize performance required for training models.

Main Features

IBM PowerAI Vision provides users with image-based deep learning analytics, including the following key features:

1. **Automatic Model Generation:** IBM PowerAI Vision provides the base model to the users, and the enterprise application developers can use the platform to automatically create new training models and cognitive APIs to quickly develop new image recognition applications.
2. **Single-click Model Deployment:** Once the model training is complete, users can deploy the API with one click, and then users could develop applications based on the model.
3. **Data Set Management and Labeling:** IBM PowerAI Vision supports the management of both user raw data and labeled data.
4. **Video Object Detection and Labeling Assistance:** IBM PowerAI Vision supports object detection of video streams and semi-automated labeling of video content.

Terms and Conditions

The IBM PowerAI Vision Technology Preview distribution is released under an Evaluation License under which the software is being provided as-is with no guarantees of the availability of the service, with no protection of your data and no support. Please refer to the licensing information in this document.

Important

IBM PowerAI Vision's Technology Preview includes a 30-day evaluation term.

After 30 days from first run, the technology preview will cease to function.

Contact IBM for details.

Technology Preview Capabilities

IBM PowerAI Vision's Technology Preview Release 3.0 currently includes three main use cases to demonstrate its capabilities:

Static Image Classification: Determine whether an image belongs to one or more classes of images based on overall image contents (e.g. "Determine the species of dog in this image").

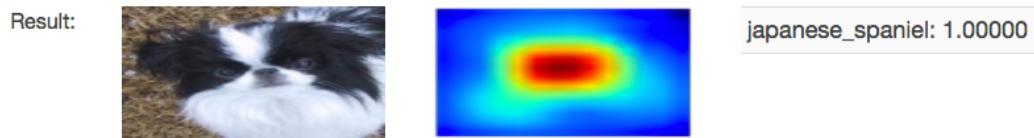


Figure 1: PowerAI Vision Image Classification can detect the overall contents of an image, based on custom training data.

Static Image Object Detection: Determine and label the contents of an image based on user-defined data labels (e.g. "Locate and label all dogs in this image").

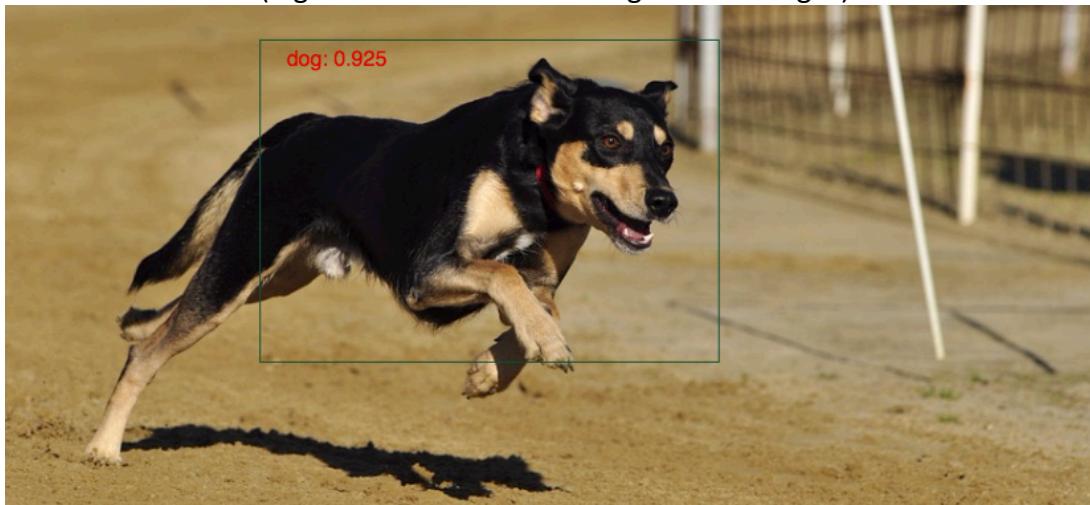


Figure 2: Power AI Vision Object Detection can detect and label instances of objects within an image based on custom training data.

Video Object Detection: Determine and label the contents of an uploaded video or live video stream based on user-defined data labels (e.g. "Locate and label all dogs in this video").

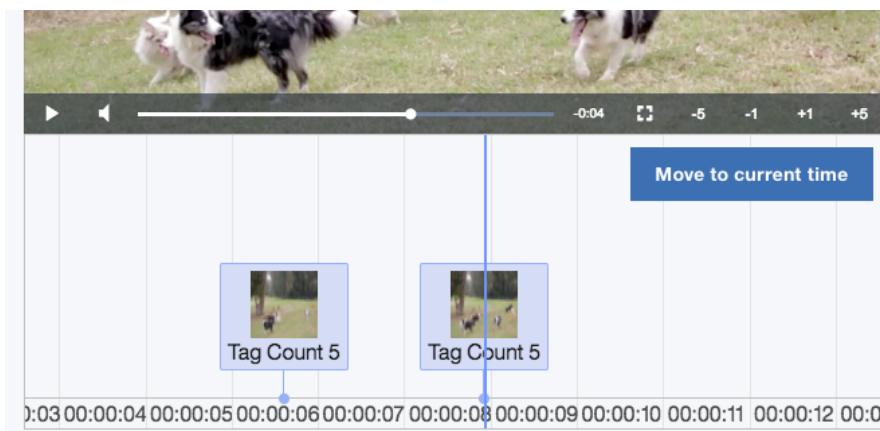


Figure 3: PowerAI Vision Video Object Detection can detect objects within streaming and recorded videos.

Architectural Overview

The architecture of IBM PowerAI Vision consists of Hardware, Resource Management, Deep Learning computation, Service Management, and Application Service layers. Each layer is built around industry-standard technologies.

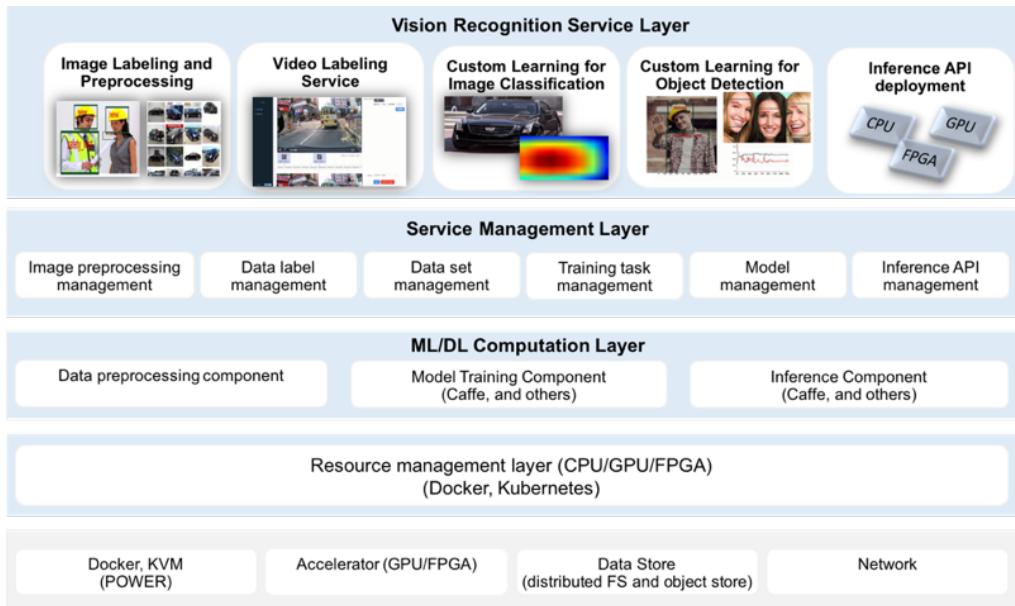


Figure 4: Architecture Diagram of PowerAI Vision

Architectural Layer	Description
Infrastructure Layer	The infrastructure layer consists of hardware systems that support IBM PowerAI Vision, including virtual machines/containers, accelerators (GPUs/FPGAs), storage systems, networks, and so on.
Resource Management Layer	The resource management layer coordinates and schedules all computing resources.
Deep Learning Calculation Layer	The deep learning calculation layer is a layer about deep learning algorithms, including data processing modules, model training modules, and prediction modules.
Service Management Layer	Manage user projects in a graphical interface, including image preprocessing, data annotation management, data set management, training task management, model management, and API management.
Application Service Layer	Located on the top of the IBM PowerAI Vision platform, it is responsible for managing all application-related services, including image labeling and preprocessing services, video annotation services, customized image classification services, and customized object detection services.

Table 1: Overview of PowerAI Vision's Architectural Layers

PowerAI Vision User Interface

IBM PowerAI Vision manages tasks via “workspaces.” Workspaces are included for Data Management, Model Training, Model Deployment, and Video Management. For more information about video processing, refer to the Video Object Detection and Labeling section.

My Data Sets

My Data Sets manages the user’s data sets which are required for training models.

Click the “Update” button to update the images and labels of the dataset, click the “Actions” button to rename and delete the current dataset, or “Mark as Sample Dataset.”

Name	Submitted By	Usage	Scenario	Last Updated	Operation
Birds	poweraivision	Image Classification	Others	2017-07-25 09:36:39	<button>Update</button> <button>Actions</button>

Figure 5: IBM PowerAI Vision’s Data Set Workspace

The user can create a data set for image classification by copying a sample data set, creating a new data set, or importing a previously-exported data set.

Note

IBM PowerAI Vision’s Technology Preview supports JPEG and PNG images.

Copy From Sample Dataset

IBM PowerAI Vision provides the functionality of copying from a sample data set that is pre-bundled in the offering. Users can load pre-bundled sample data set or load data sets marked as sample datasets.

The user can choose to copy a data set from the menu labelled “Copy From Sample Dataset.”

Copy From Sample Dataset

DataSet Name: birds

Sample Dataset: Birds (Image Classification Others)

Cancel Copy Dataset

Figure 6: Sample data sets provide small, pre-built collections of images for learning about the platform.

Create a New Dataset

Creating a new data set is the most common way for users to build their own customized training tasks. At present, IBM PowerAI Vision supports two import methods: uploading individual files, or uploading zip files from local disk. Data sets are targeted to specific use cases such as image classification, which determines the overall contents of an image, or object detection, which discovers and labels instances of specific objects in an image.

Create a Dataset from Image Files for Image Classification

PowerAI Vision Image Classification determines the overall contents of an image based on training categories.

To create a new data set for Image Classification:

1. From the “My Data Sets” view, select “Add Data Set.”



Figure 7: Add Data Set offers multiple options, depending on the use case of the data set.

2. Select “Image Classification.”
3. Name the data set, and select from one of the pre-built scenarios. For most users, choose “Others” to create a general-purpose classifier.

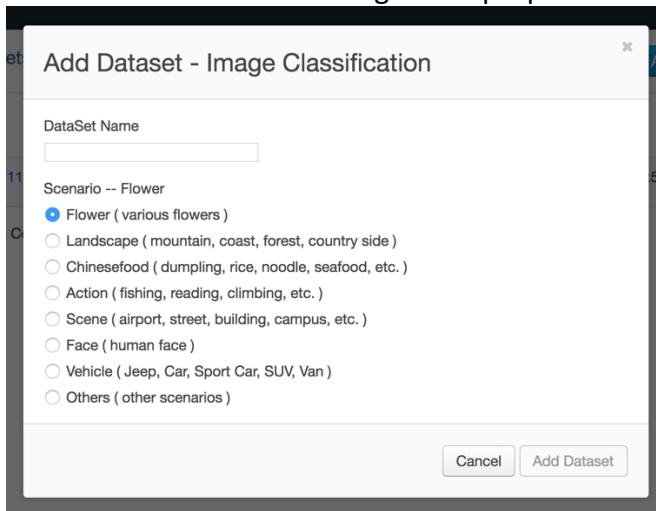


Figure 8: Name data sets as needed based on content, and choose “Others” for a general-purpose classifier.

Note

IBM PowerAI Vision uses transfer learning to accelerate training times. Choosing a specific scenario above will start from a customized model that is well-suited for that task, but may be poorly-suited for other tasks. For most users, choose “Other” to start from a general-purpose model.

- Click on “Add Category” and enter a unique name for each of the categories within the data set.
- Select a category, and upload one or more images for classification. Note that PowerAI Vision supports batch upload (selecting multiple files or drag-and-dropping multiple files).

Important

Training quality improves with larger data sets. A minimum of five images per class is required; at least fifty images per class is recommended.

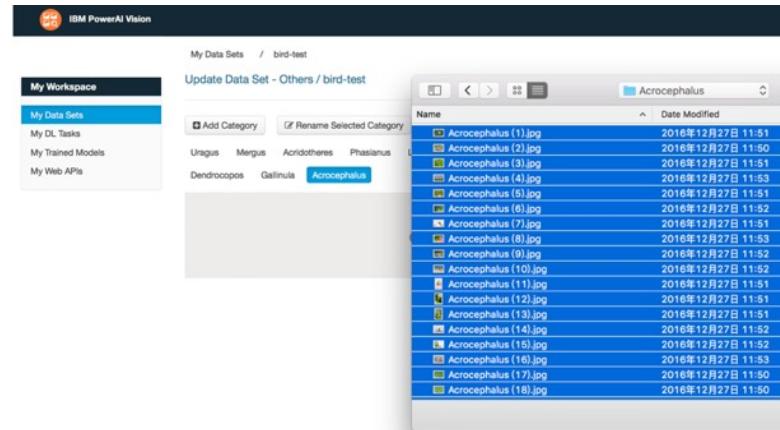


Figure 9: Upload one or more images to the data set.

Note

Users may also rename categories, manage categories or individual images, and delete categories, individual images, or entire collections of images.

Create a Dataset from Image Files for Object Detection

PowerAI Vision Object Detection discovers and labels objects within an image, enabling users and developers to count instances of objects within an image based on customized training.

To create a new data set for object detection:

- From the “My Data Sets” view, select “Add Data Set.”

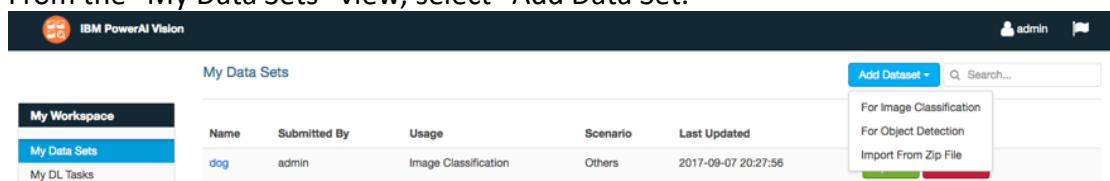


Figure 10: Add Data Set offers multiple options, depending on the use case of the data set.

- Select “Object Detection.”
- Name the data set.
- Upload one or more images by drag-n-drop or multiple selection in a web browser.

Note

Users may upload multiple images simultaneously via web browser.

5. Create one or more tags by clicking the “+” icon to add a new tag. Each tag will represent the training objects within the image based on specific use cases (e.g. dogs, cars, bicycles).

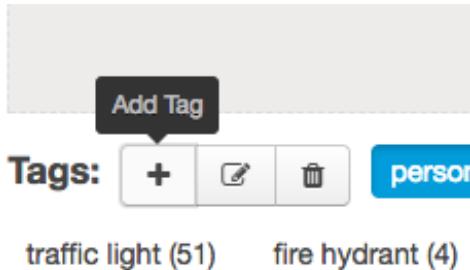


Figure 11: Add tags as needed.

Important

Object detection relies on multiple different images for training. PowerAI Vision requires at least five images for training. At least fifty are suggested.

6. Label the contents of the image by selecting each image, selecting a tag, and using a mouse to drag-n-drop a bounding box around the objects of that type in the image. Select additional tags, and draw additional bounding boxes as needed. Click “Save” for each image.

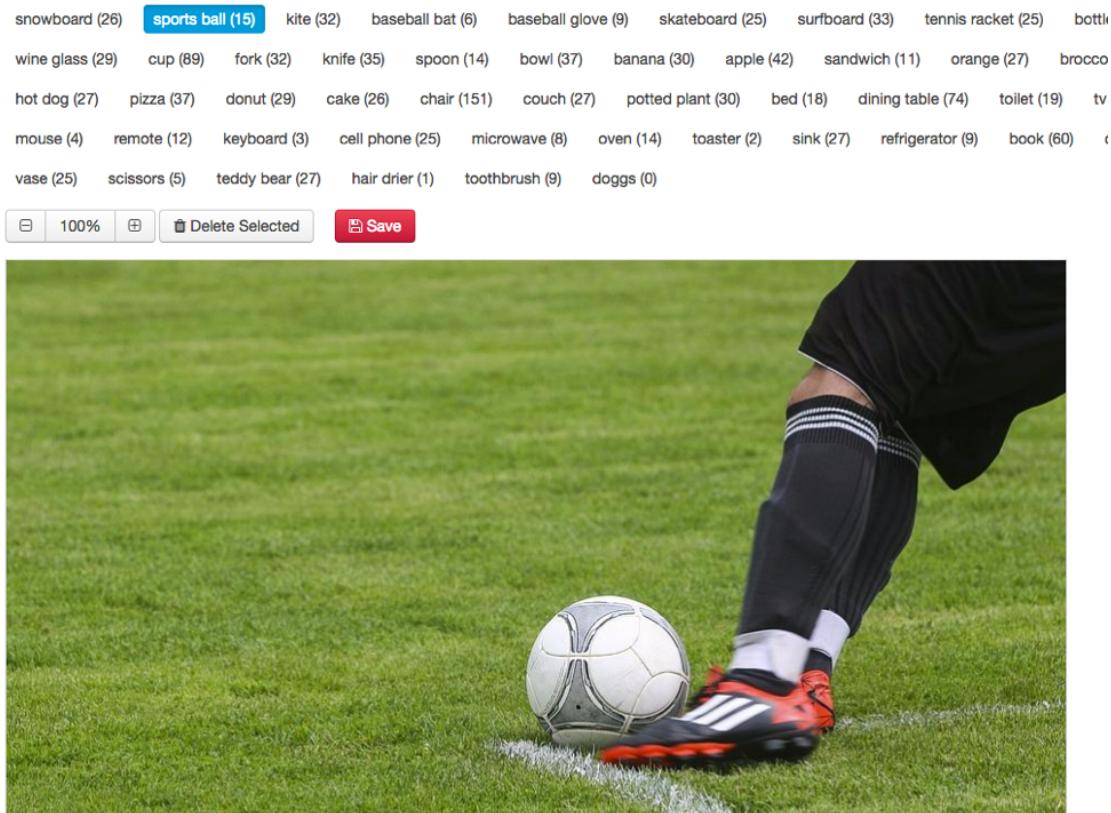


Figure 12: Images by default have no labeled objects. Select a label and drag a box around each instance of that object.

Note

PowerAI Vision shows unsaved bounding boxes in blue, and shows saved tags in green. Be sure to click “save.” PowerAI Vision supports bulk upload of images via REST APIs.

7. Repeat this process for all tags and all images.

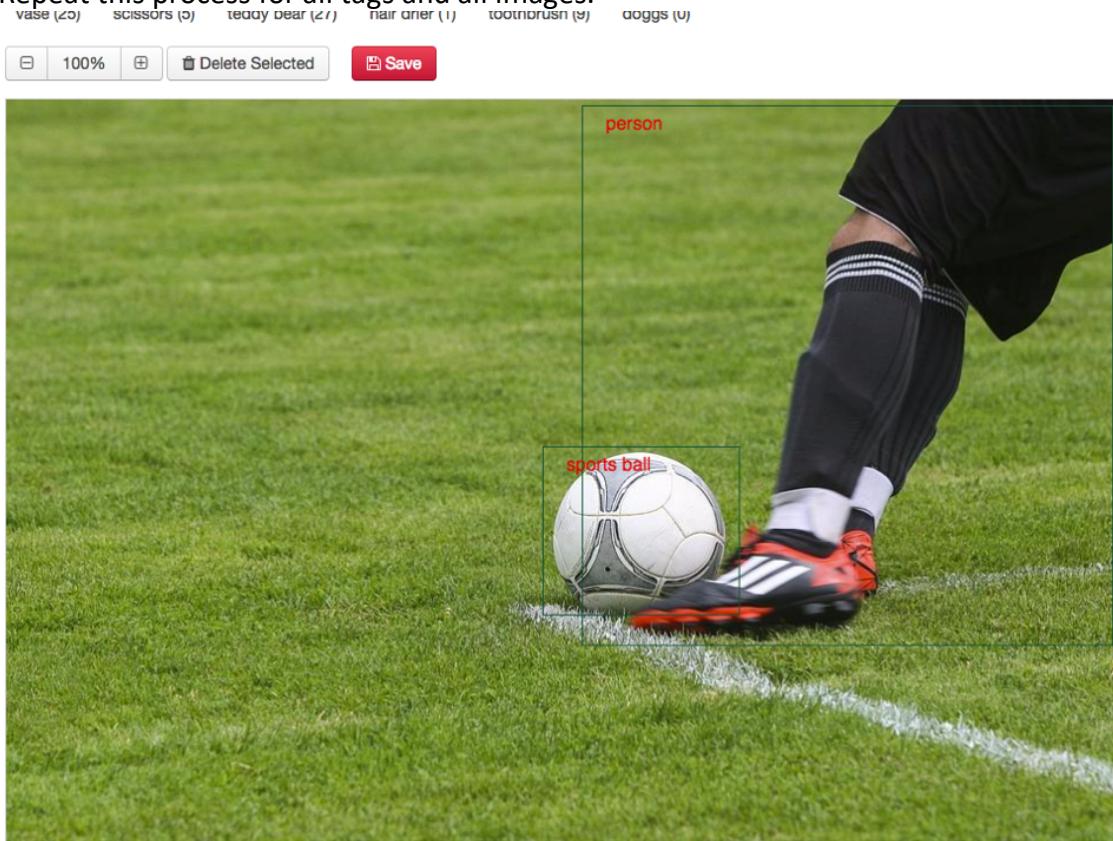


Figure 13: PowerAI Vision labels objects in a training image with green boxes. Users may fine-tune imported bounding boxes by hand.

Exporting and Importing Data Sets

IBM PowerAI Vision enables users to export data sets as zip archives. These may be later reimported into the platform by selecting “My Data Sets” → “Add Data Set” → “Import from Zip File” and uploading or selecting an archive.

A screenshot of the 'My Data Sets' page. At the top, there is a search bar and a 'Search...' button. Below the search bar is a dropdown menu with options: 'Add Dataset', 'For Image Classification', and 'Import From Zip File' (which is highlighted with a blue background). A table lists existing datasets: 'Birds' submitted by 'dlaastest' for 'Image Classification' with a 'Last Updated' date of '2017-07-13 19:41:08'. There are 'Update' and 'Actions' buttons for each row.

Figure 14: Importing a Data Set into PowerAI Vision

Important

Note that the application may only import data sets created by the IBM PowerAI Vision platform. PowerAI Vision does not support zip files from other sources.

A data set may be exported at any time by selecting the data set and selecting “Export as Zip File.” This zip file contains all image data as well as metadata about the image (e.g. label data) necessary to manage the data set.



Figure 15: Exporting a Data Set from PowerAI Vision

My DL Task

Deep learning training jobs are represented in PowerAI Vision as deep learning “tasks.” Users may rename and delete a training task by clicking the “Actions” button.

Name	Submitted By	Usage	Status	Created At	Operation
Birds	poweraivision	Image Classification	deployed	2017-07-21 16:43:39	<button>Details</button> <button>Actions ▾</button>

Figure 16: The Deep Learning Task Workspace manages current and complete training jobs.

Tasks are specific to each data set type. PowerAI Vision’s technology preview supports creating models for Image Classification and Object Detection.

Figure 17: Static image task types available for IBM PowerAI Vision’s Technology Preview include Object Detection and Image Classification.

My Trained Models

Deep Learning tasks result in trained models which may be deployed for use. Users may “Deploy” a model, which creates a unique API endpoint based on that model for inference operations.

The screenshot shows the 'My Trained Models' section of the PowerAI Vision interface. On the left, a sidebar lists 'My Workspace' items: My Data Sets, My DL Tasks, My Trained Models (which is selected and highlighted in blue), and My Web APIs. The main content area has a header 'My Trained Models' with a search bar. Below the header is a table with columns: Id, Usage, Categories, Accuracy, Created At, and Operation. One row is shown: da39806b-d085-4e7e-94e2-eef1e1133b7d, Image Classification, Acridotheres, Acrocephal..., 0.72321, 2017-07-21 16:44:25, with 'Deploy' and 'Actions' buttons. At the bottom, it says 'Total: 1, Page Count: 1'.

Figure 18: PowerAI Vision outputs models which may be deployed as API endpoints.

My Web APIs

PowerAI Vision presents Web APIs for inference operations. Each Web API is deployed in a stateless container, and may be managed independently. Deletion of the deployed API will delete the container without deleting the underlying model.

Validate Trained Model

The screenshot shows the 'My Web APIs' section of the PowerAI Vision interface. The sidebar is identical to Figure 18. The main content area has a header 'My Web APIs' with a 'How to use' link and a search bar. Below is a table with columns: URL, Categories, Accuracy, Deployed At, and Operation. One row is shown: /diapis/23619197-c2d6-4817-85fa-04023b1adbff, Acridotheres, Acrocephal..., 0.92500, 2017-07-25 11:21:43, with 'Run Test' and 'Actions' buttons. At the bottom, it says 'Total: 1, Page Count: 1'.

Figure 19: Trained models may be validated via the GUI or accessed via REST API.

To validate a trained model against a test image by hand:

1. Click on “My Web APIs.”
2. Select the deployed model and click “Run Test.”

Important

If no web APIs are available, ensure a trained model is “deployed.”

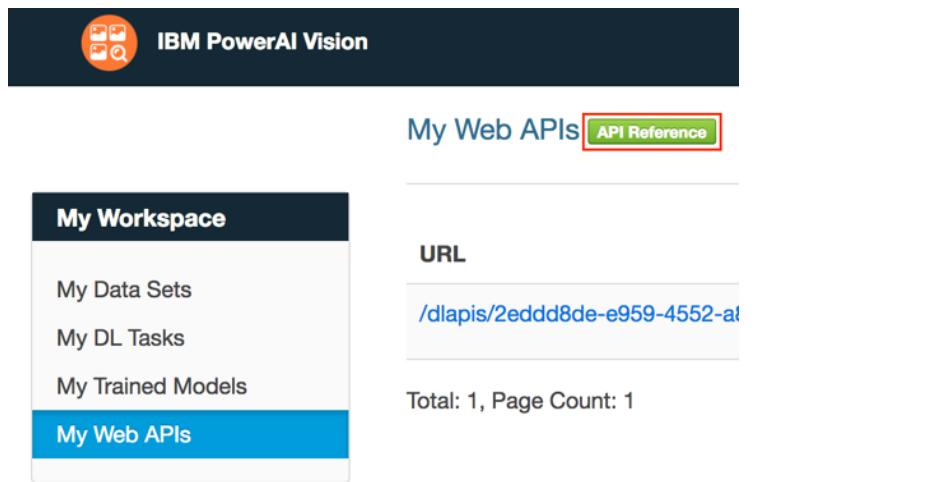
3. Upload a test image by drag-n-drop, selecting an image, or entering a URL.

The screenshot shows a 'Run Test' dialog for the model 'da39806b-d085-4e7e-94e2-eef1e1133b7d'. The top bar shows 'My Web APIs / da39806b-d085-4e7e-94e2-eef1e1133b7d'. The main area has a header 'Run Test - da39806b-d085-4e7e-94e2-eef1e1133b7d'. It says 'Upload a picture to classify:' and features a file upload input field with placeholder text 'Upload pictures (jpg / png) by dropping them here Or Select some'.

Figure 20: PowerAI Vision enables users to test deployed models within the application prior to deployment into production.

Using a REST API for Inference

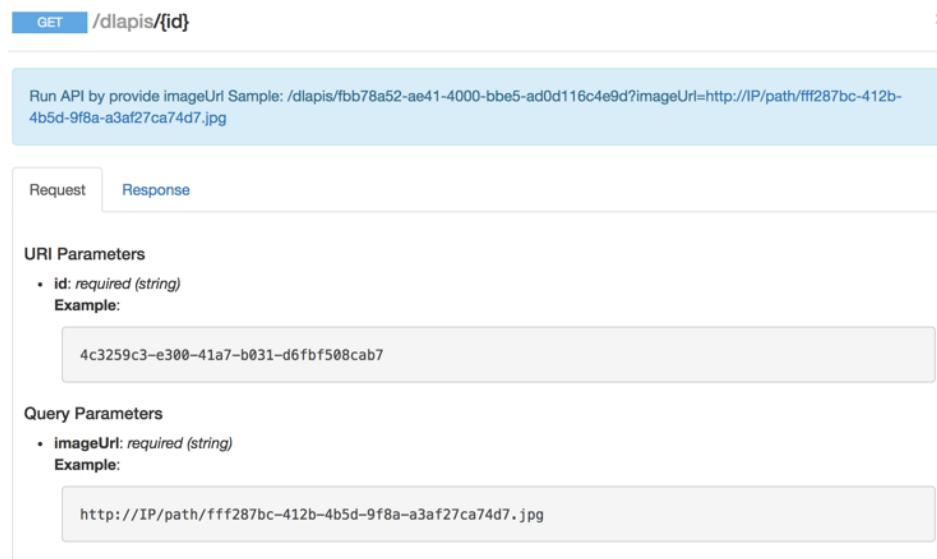
Deployed models may be integrated into a larger application or pipeline via a RESTful API.



The screenshot shows the IBM PowerAI Vision interface. At the top, there's a dark header bar with the IBM logo and the text "IBM PowerAI Vision". Below this is a sidebar titled "My Workspace" containing links for "My Data Sets", "My DL Tasks", "My Trained Models", and "My Web APIs". The "My Web APIs" link is highlighted with a blue background. The main content area has a title "My Web APIs" and a button labeled "API Reference" which is also highlighted with a red box. Below this, there's a section for "URL" with the value "/dlapis/2eddd8de-e959-4552-a1". At the bottom, it says "Total: 1, Page Count: 1".

Figure 21: Click on "API Reference" to learn more about programmatic access to PowerAI Vision deployed models.

Standard PUT/GET verbs are documented within the PowerAI GUI under the “API Reference” page.



The screenshot shows the "API Reference" page for the endpoint "/dlapis/{id}". At the top, it says "Run API by provide imageUrl Sample: /dlapis/fbb78a52-ae41-4000-bbe5-ad0d116c4e9d?imageUrl=http://IP/path/fff287bc-412b-4b5d-9f8a-a3af27ca74d7.jpg". Below this, there are tabs for "Request" and "Response". Under "URI Parameters", it lists "id: required (string)" with an example "4c3259c3-e300-41a7-b031-d6fbf508cab7". Under "Query Parameters", it lists "imageUrl: required (string)" with an example "http://IP/path/fff287bc-412b-4b5d-9f8a-a3af27ca74d7.jpg".

Figure 22: The IBM PowerAI Vision GUI includes information about how PUTs and GETs function for each deployed endpoint.

Video Object Detection and Labeling

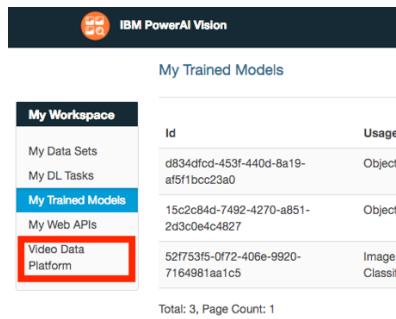
IBM PowerAI Vision supports labeling of objects in video streams or recorded files. Labeling of training data is semi-automated; IBM PowerAI Vision supports assisted labeling of video streams based on learned deep learning models.

Important

IBM PowerAI Vision Technology Preview 3 supports only H264 MP4 files, and works best with Google Chrome.

Overview

Video Object Detection and Labeling uses a dedicated user interface for managing of video assets, but still relies on the core IBM PowerAI Vision application for training and deployment of deep learning models. Video assets are organized into Data Sets, which contain one or more videos. Selected video frames may be extracted for training a model. To get started, click on “Video Data Platform” to access the tech preview’s dedicated Video Management User Interface.

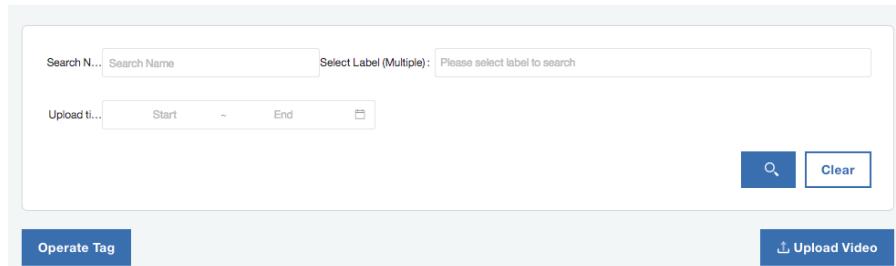


The screenshot shows the IBM PowerAI Vision interface. At the top, there's a dark header bar with the IBM logo and the text "IBM PowerAI Vision". Below it, a navigation sidebar on the left has several options: "My Data Sets", "My DL Tasks", "My Trained Models" (which is currently selected and highlighted in blue), "My Web APIs", and "Video Data Platform" (which is also highlighted with a red box). The main content area is titled "My Trained Models" and displays a table with three rows of data. The columns are "Id" and "Usage". The first row has Id "d834dfcd-453f-440d-8a19-af511bcc23a0" and Usage "Object I". The second row has Id "15c2b84d-7492-4270-a851-2d3c0e4c4827" and Usage "Object I". The third row has Id "52f753f5-0f72-406e-9920-7164981aa1c5" and Usage "Image Classifier". At the bottom of the table, it says "Total: 3, Page Count: 1".

Figure 23: Click on the “Video Data Platform” link to access the main portal for video asset management.

Data Set Management

IBM PowerAI Vision’s Video Data Platform organizes data into Data Sets, which are made up of one or more video files. Files within a data set may be tagged individually. Tags must be exported to the main PowerAI Vision platform for training and deployment. Videos may be sorted by tag, by file name, and filtered by properties like upload timestamp.



This screenshot shows the "Operate Tag" interface of the Video Data Platform. At the top, there are search and filter fields: "Search N..." with a dropdown "Search Name", "Select Label (Multiple)" with a dropdown "Please select label to search", and a date range selector "Upload till..." with "Start" and "End" fields and a calendar icon. Below these are two buttons: "Operate Tag" on the left and "Upload Video" with a camera icon on the right. At the bottom center are "Search" and "Clear" buttons.

Figure 24: Videos may be filtered by a variety of properties.

Video Operations

Users must label videos for training. Initial labeling may be performed manually, or autonomously based on deployed models.

The screenshot shows a table of uploaded video files with columns for Name, Tags, Status, Upload Time, Size, Time, Task, Auto Detect Result, and Operation. Two videos are listed: 'puppy.mp4' (Untagged) and 'border-colies' (Tagged). The 'border-colies' row has a red badge with '105' indicating the number of tagged objects. The 'Operation' column for each video contains three icons: a three-dot menu, a blue briefcase labeled 'Add Label', and a white briefcase icon. A sidebar on the right lists five operations: Add Label, Auto Detect, Export Labels, Auto Labeling, and Delete.

Name	Tags	Status	Upload Time	Size	Time	Task	Auto Detect Result	Operation
puppy.mp4		Untagged	2017/11/17 11:1	3.85 MB	0:00:15			
border-colies	dog	Tagged	2017/11/17 11:0	6.8 MB	0:00:13			

10 items per page | 1 - 2 of 2 items 1 of 1 pages <

Add Label
Auto Detect
Export Labels
Auto Labeling
Delete

Figure 25: Users may label videos automatically or manually by clicking on the "operation" icon to the right of the video.

Video Operations	Description
Add Label	Start here and add labels to newly-uploaded videos. Use this option to add or edit additional labeled objects, and to build a training data set of tagged videos.
Auto Detect	Validate an object detector's performance by tagging a video based on a trained model.
Export Labels	Export labels to the main PowerAI platform for training and model deployment.
Auto Labeling	If an object detector is already deployed, PowerAI Vision can auto-label a video based on periodic samples (e.g. once per second). Use this for machine-assisted labeling of training data.
Delete	Delete individual video files from a data set. Any trained models using the video will remain. Export labels and re-train a new model to reflect changes.

Table 2: Summary of available operations for PowerAI Vision's Video Data Platform

Video Label Editor

To train a model based on videos, *at least* five frames must be labeled with objects. The label editor shows the video clip, and allows a user to “capture” a single frame for labeling below.

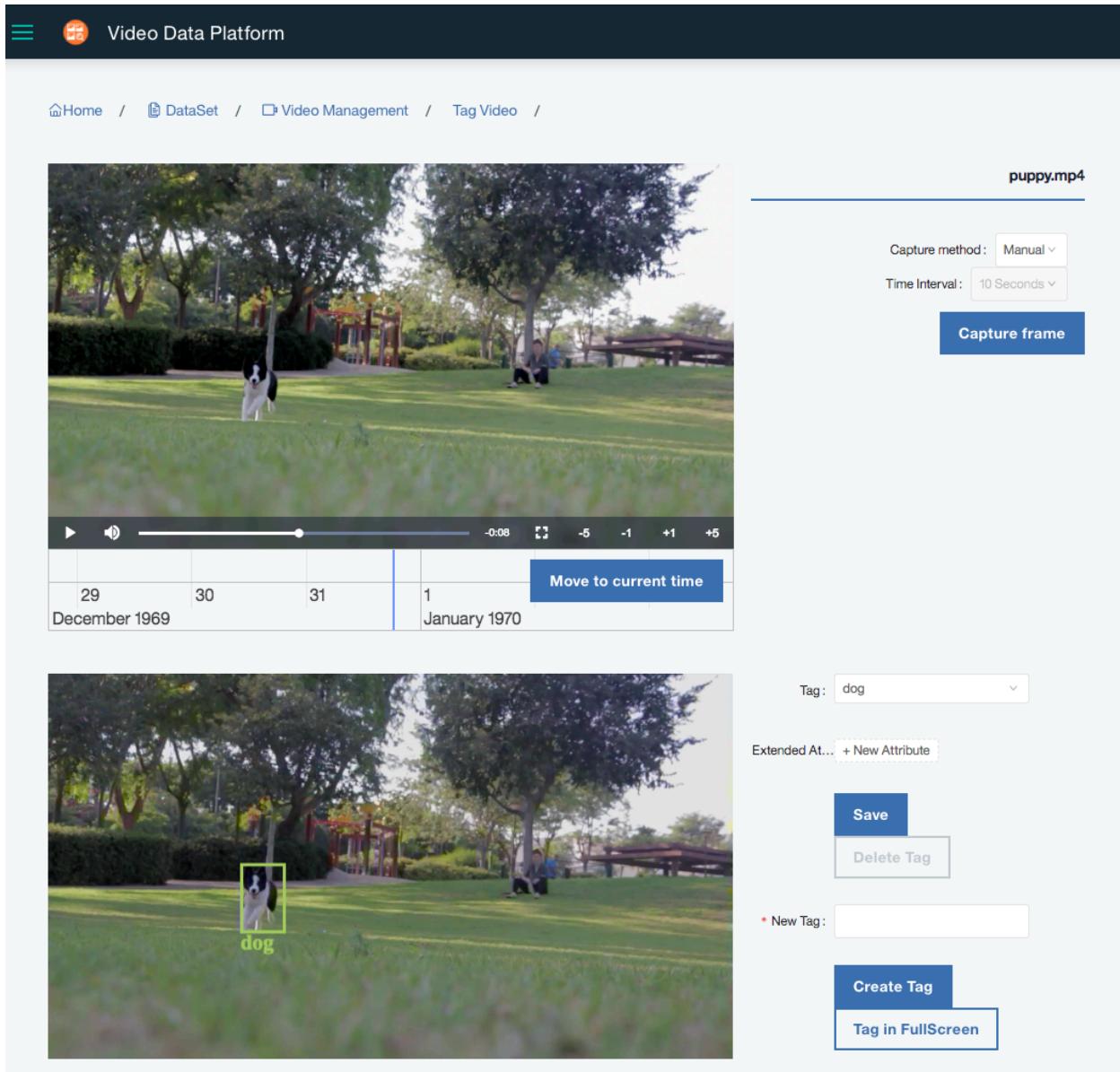


Figure 26: PowerAI Vision allows a user to edit labeled video frames for training of deep learning models.

Hands-On: Classification with PowerAI Vision

Learn how to prepare data, build a data set, set up training tasks, and test a trained model through a real bird classification task.

The goal of this example is to train a model to classify images of birds into groups based on their physiological similarities. Once the model is trained with a known dataset, users can upload new datasets to auto classify the birds into respective categories.

A video of the process is made available on YouTube:

<https://www.youtube.com/watch?v=qHZRnswzqUI>

Data Preparation

Data preparation usually includes two types of data, training data and test data. The main purpose of training data is to make the neural network learn features of the object, to build the classification model. The purpose of the test data is to validate accuracy of the trained model.

Note

1. *Training data and test data should be different pictures.*
2. *IBM PowerAI Vision currently only supports JPEG or PNG format images.*

First prepare some of the bird pictures for the training data set.



Figure 27: Example Data sets can include an arbitrary number of classes.



Figure 28: Each collection of images as shown here includes multiple examples of the same species of bird.

Uploading a Data Set

On the IBM PowerAI Vision platform, select "My Data Sets" → "Add Dataset" → "For Image Classification", and name the dataset as "Birds".

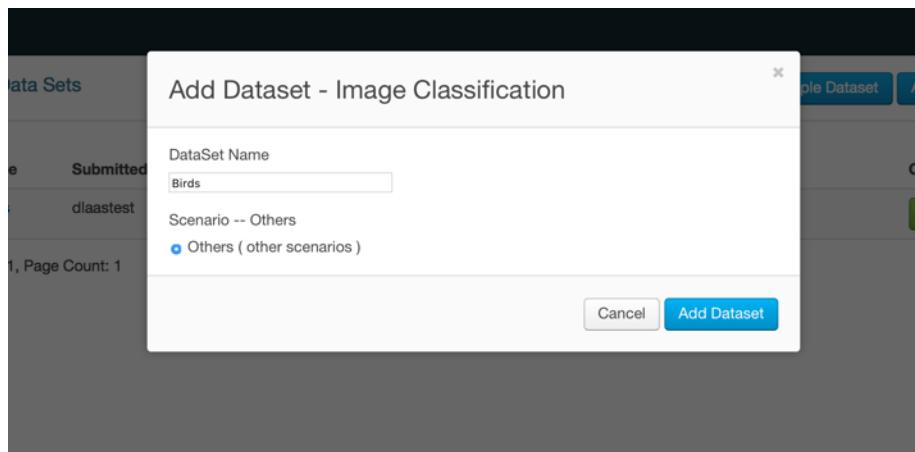


Figure 29: Create a new data set for image classification.

Click on "Add Dataset" and name the first bird category as "Acridotheres" and select the image of the Acridotheres bird from local directory upload it to this newly-created category.

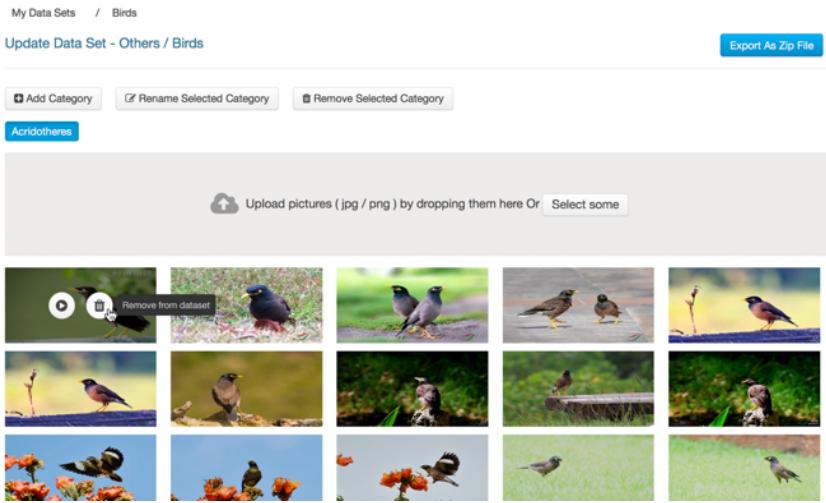


Figure 30: PowerAI Vision provides a gallery view for images in a data set category.

Repeat this process for other categories from the data set.

Figure 31: Click on each category to view images that belong to that category within the data set.

Notes

1. If user has copied an existing dataset, images will be preloaded.
2. Data sets may be exported out of the platform via the "Export as Zip file" option.
3. Zip files exported from IBM PowerAI Vision include metadata about the data set. These zip files may be imported back into the IBM PowerAI Vision platform.

Creating a New Training Task

Select "My DL Tasks", "Create New Task" on the IBM PowerAI Vision and select "Classification". The parameter settings page for the classification task will then be entered.

The classifier name can be set according to the user's needs, such as "Birds," and the dataset, we select the "Birds" dataset just created.

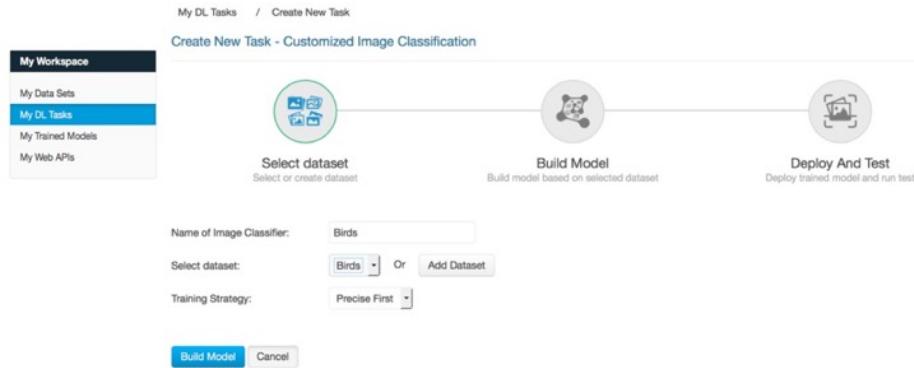


Figure 32: Creating a new Deep Learning Task offers several options, depending on the task type.

There are three training options available to users, Speed First, Precise First and Customized.

- **Speed First:** The training speed is the preferred goal for the system to consider the strategy and is suitable for scenarios where the user wants to get a rough training result quickly and without too much of the accuracy of the model. For example, users want to quickly check the data set in the picture is suitable for this project.
- **Precise First:** With accuracy as the preferred goal of the system to consider the strategy. This option is the default option for the system and is suitable for most training programs. This method takes more time than option for "Speed First."
- **Customized:** IBM PowerAI Vision supports the deep learning developer to adjust some of the training parameters. Since each parameter adjustment will impact the time to train, it is recommended that the user carefully modify the parameters.

Select "Precise First" and click "Build Model." IBM PowerAI Vision will display the real-time status of Train Loss, Test Loss, Accuracy, and other parameters in a graphical way to assist user to judge the training effect. It also provides the estimated remaining training time.

Once the training is complete, click on "Deploy and Test" to publish the model as an API for testing.

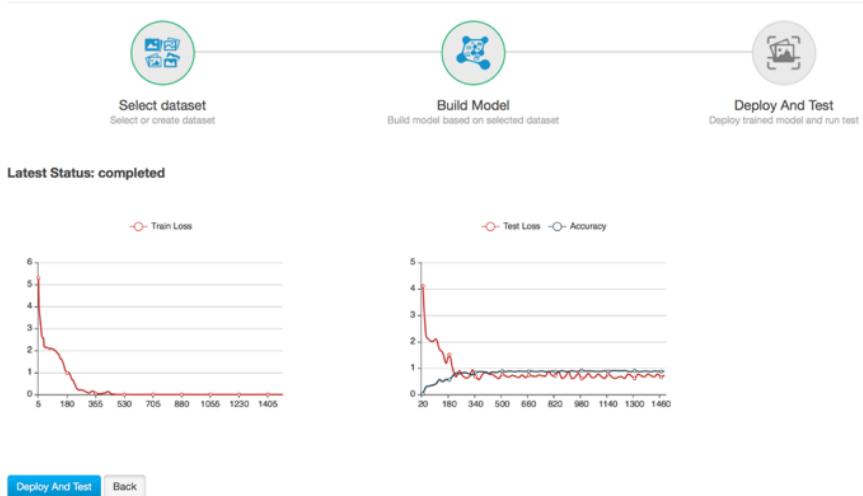


Figure 33: PowerAI Vision displays various metrics showing real-time training job status as models converge over time.

Testing the Trained Model

After the classification model training is completed, the user can call the API of the model through its own application, and test the model effect through the test page provided by PowerAI Vision.

On the IBM PowerAI Vision main page, select “My Web APIs” to enter the API Management page. “My Web APIs” page lists all the APIs that have been published by the user, the URL of each API, Categories, Accuracy and Deployed At. Click the "Run Test" button to access the API test page.

Figure 34: Selecting "Run Test" to try out the API by uploading an image or attempting to classify an image hosted at a URL.

Click "Run Test" to enter the API test page. IBM PowerAI Vision supports the user through the URL or upload the local image from the local directory to test the deployed API, here we upload a local picture as an example.

Click "Select Some" to select a test picture from the local directory and upload it. The test result will display the uploaded picture and the test result heat map, and give the classification and accuracy of the bird. For example, the following test bird pictures, belonging to the Pycnonotus category, the accuracy is 0.925.

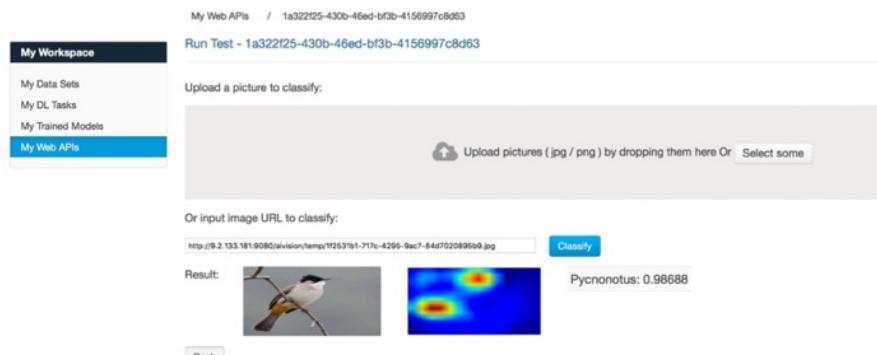


Figure 35: Heat Maps show features used to detect the class of the image.

The heat map of the test results can assist the user in judging whether the model has correctly learned the features of this classification. The red area of the heat map corresponds to the area of the picture feature. For example, the red area of the test picture above corresponds to the bird's neck and body position, indicating that the model has learned the correct features.

Hands-On: Detecting Objects with PowerAI Vision

This section describes how to prepare data, build a data set, set up training tasks, and test a trained model through an object detection task.

Data Preparation

Data preparation usually includes both training and test images, and bounding box object tags.

Note

1. *Training data and test data should be different pictures.*
2. *IBM PowerAI Vision currently only supports JPEG or PNG format images.*

First prepare some of the images for the training data set. If bounding box data is not available for the training set, PowerAI Vision supports manually drawing object bounding boxes as needed.

Uploading a Data Set

On the IBM PowerAI Vision platform, select "My Data Sets" → "Add Dataset" → "For Object Detection", and name the dataset.

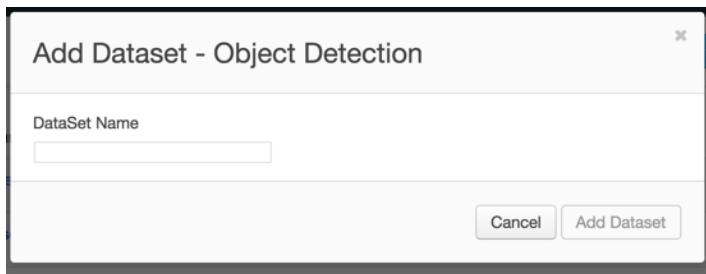


Figure 36: Create a new data set for object detection

Click on "Add Dataset."

Batch upload one or more images into the data set.

The screenshot shows the IBM PowerAI Vision interface for managing data sets. On the left, a sidebar titled 'My Workspace' includes 'My Data Sets' (selected), 'My DL Tasks', 'My Trained Models', and 'My Web APIs'. The main area displays a 'sample object set' with two images shown. Below the images, text indicates 'Total: 5002, Page Count: 834' and a page navigation bar from 1 to 834. A large central box contains a cloud icon and the text 'Upload pictures (.jpg / .png) by dropping them here Or Select some'. Below this is a 'Tags:' section with a '+' button, a trash bin icon, and a search icon. A list of tags is displayed, with 'dog (244)' highlighted in blue. Other tags include person (10902), motorcycle (306), airplane (253), bus (260), train (187), truck (376), boat (465), traffic light (501), fire hydrant (68), stop sign (83), parking meter (51), bench (378), bird (415), cat (204), horse (321), sheep (371), cow (411), elephant (233), bear (67), zebra (250), giraffe (205), backpack (368), umbrella (523), handbag (553), tie (213), suitcase (260), frisbee (121), skis (283), snowboard (162), kite (378), baseball bat (107), baseball glove (118), skateboard (285), surfboard (271), sports ball (288), tennis racket (223), bottle (1051), wine glass (272), fork (256), spoon (226), bowl (630), banana (371), sandwich (162), and orange (300).

Figure 37: PowerAI Vision provides a gallery view for images in the object detection data set. Note that images are not organized by category, as multiple categories of objects may be tagged in each image.

Create one or more tags for the data set by clicking on the “+” sign next to the “Tags” list. An arbitrary number of tags may exist.

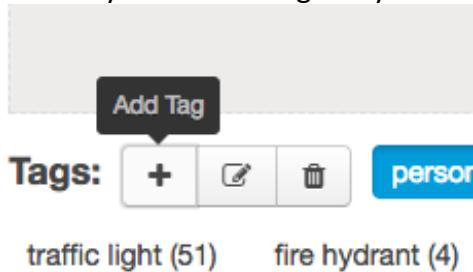


Figure 38: Add tags as needed.

Label the contents of the image by selecting each image, selecting a tag, and using a mouse to drag-n-drop a bounding box around the objects of that type in the image. Select additional tags, and draw additional bounding boxes as needed. Click “Save” for each image. Multiple instances of objects may appear in each image (e.g. multiple dogs), and bounding boxes may overlap (e.g. an object may partially-obscure another object).

snowboard (26) sports ball (15) kite (32) baseball bat (6) baseball glove (9) skateboard (25) surfboard (33) tennis racket (25) bottle (29) wine glass (29) cup (89) fork (32) knife (35) spoon (14) bowl (37) banana (30) apple (42) sandwich (11) orange (27) broccoli (27) hot dog (27) pizza (37) donut (29) cake (26) chair (151) couch (27) potted plant (30) bed (18) dining table (74) toilet (19) tv (4) mouse (4) remote (12) keyboard (3) cell phone (25) microwave (8) oven (14) toaster (2) sink (27) refrigerator (9) book (60) vase (25) scissors (5) teddy bear (27) hair drier (1) toothbrush (9) doggs (0)

100% Delete Selected Save



Figure 39: Images by default have no labeled objects. Select a label and drag a box around each instance of that object.

Note

Unsaved bounding boxes will be shown in a blue. Saved tags are shown in green. Be sure to click “save.” PowerAI Vision supports bulk upload of images via REST APIs.

Repeat this process for all tags and all images.

vase (29) scissors (5) teddy bear (27) hair dryer (1) toothbrush (9) doggs (0)

100% Delete Selected

Save



Figure 40: Objects are labeled in a training image with green boxes.

Creating a New Training Task

To train a new model, select "My DL Tasks" → "Create New Task" and select "Object Detection." Name the detector task as needed. You may customize object detection training parameters as needed, or select default values. Start the training job, and wait for it to complete.

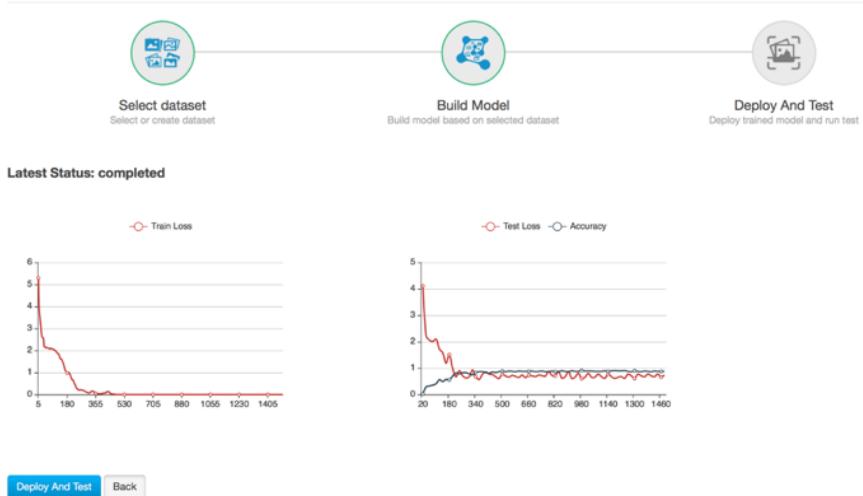


Figure 41: PowerAI Vision displays various metrics showing real-time training job status as models converge over time.

Testing the Trained Model

After object detection training is completed, the user can call an API of the model through its own application, and test the model effect through the test page provided by PowerAI Vision.

From “My Trained Models” choose “Deploy” to create a RESTful endpoint.

On the IBM PowerAI Vision main page, select “My Web APIs” to enter the API Management page. “My Web APIs” page lists all the APIs that have been published by the user, the URL of each API, Categories, Accuracy and Deployed At. Click the "Run Test" button to access the API test page.

URL	Categories	Accuracy	Deployed At	Operation
/diapis/23619197-c2d6-4817-85fa-04023b1adbf9	Acridotheres, Acrocephal...	0.92500	2017-07-25 11:21:43	Run Test Actions ▾

Figure 42: Selecting "Run Test" to try out the API by uploading an image or attempting to classify an image hosted at a URL.

Tests can run on images uploaded from the local client, or on images that are hosted via HTTP. To try out the endpoint, by drag-n-dropping a file for upload, or by entering an image URL.

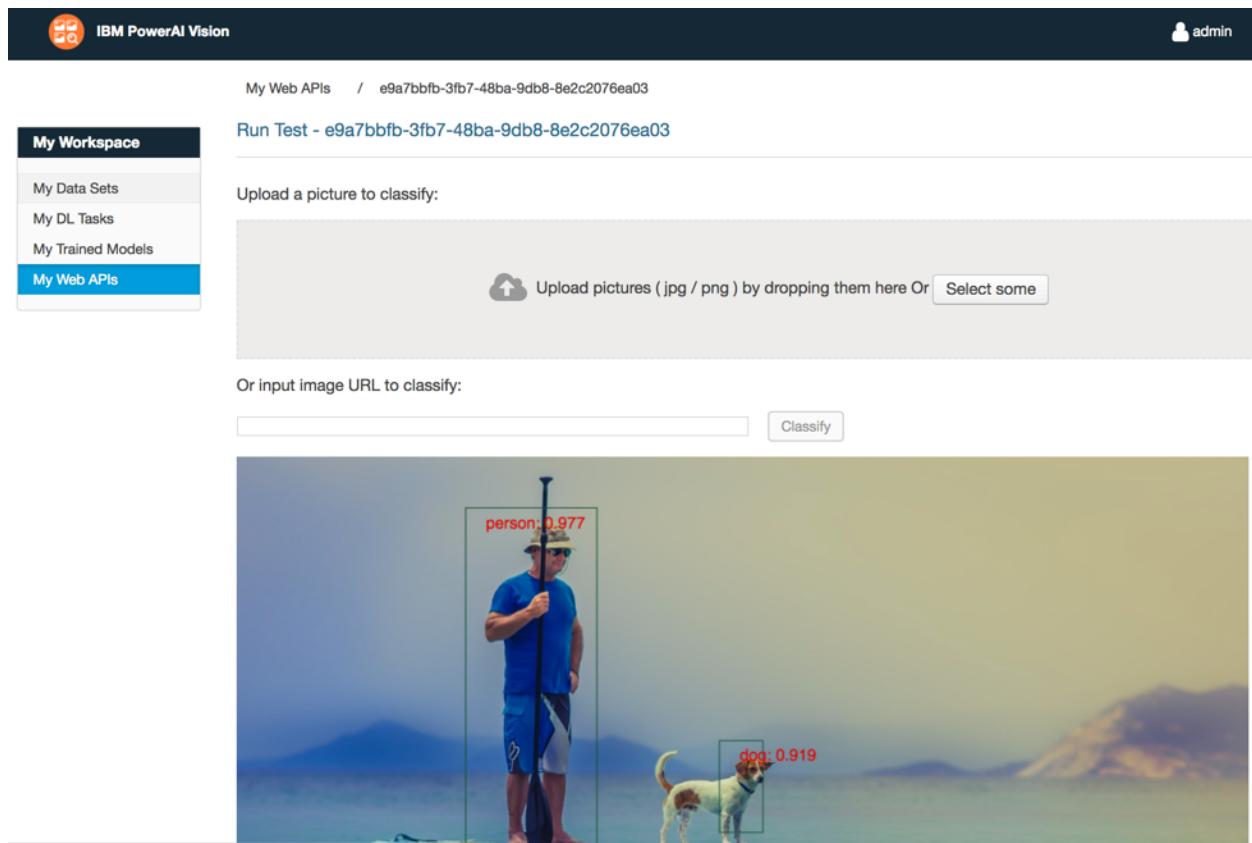


Figure 43: PowerAI Vision returns bounding boxes around objects found inside the image, labels each object, and returns a confidence score for each found tag.

Each object detected by PowerAI Vision is labeled with X,Y pixel coordinates, an object class, and a confidence score. Underlying responses are returned in JSON.

Hands-On: Detecting Objects in Videos with PowerAI Vision

This section describes how to prepare videos, build a data set, set up training tasks, and test a trained model through a video object detection task.

Data Preparation

Data preparation usually includes both training and test video files.

Note

1. *Training data and test data should be different video files.*
2. *IBM PowerAI Vision currently only supports H264 MP4 video files.*

First prepare some of the videos for the training data set. While videos may have a very large number of frames, a relatively small number should be labeled as initial training data. To expand a data set, IBM PowerAI Vision supports semi-autonomous labeling after an initial model is trained.

Uploading a Data Set

On the IBM PowerAI Vision platform, select "Video Data Platform" and select "Data Set."

The screenshot shows the IBM PowerAI Vision interface. On the left, there's a sidebar with 'My Workspace' (selected), 'My Data Sets', 'My DL Tasks', 'My Trained Models' (selected), 'My Web APIs', and 'Video Data Platform' (highlighted with a red box). The main area is titled 'My Trained Models' and lists three entries:

Id	Usage
d834dfcd-453f-440d-8a19-af5f1bcc23a0	Object I
15c2c84d-7492-4270-a851-2d3c0e4c4827	Object I
52f753f5-0f72-406e-9920-7164981aa1c5	Image Classifier

Total: 3, Page Count: 1

Figure 44: Click on the "Video Data Platform" link to access the main portal for video asset management.

Select "Create Data Set," name the new data set, and click "Confirm."

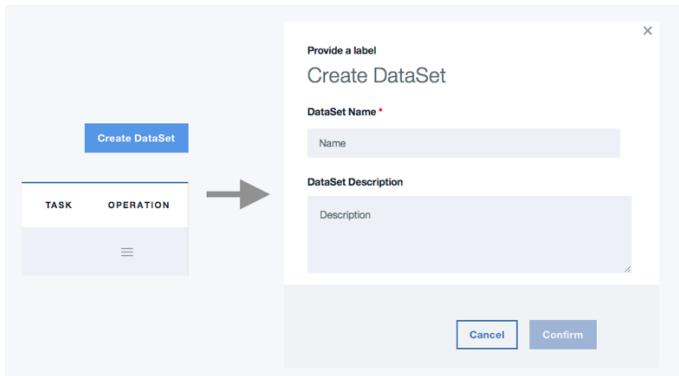


Figure 45: Click “Create Dataset” and then name the new data set.

Add a file to the data set by left-clicking on the “operation” menu icon to the right of the new data set name, and choosing “Manage Videos.”

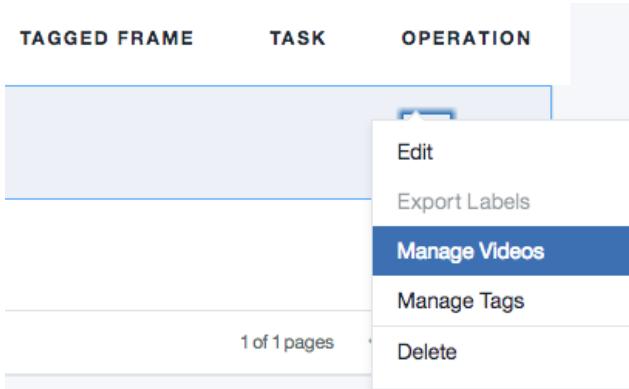


Figure 46: Click on the “menu” icon for the data set, and choose “Manage Videos” to add or delete videos from the data set.

Begin by uploading training videos. Select “Upload Video,” navigate to an MP4 video file, and choose “Open” or “OK” depending on your operating system and browser. Repeat this process for each training file as needed.

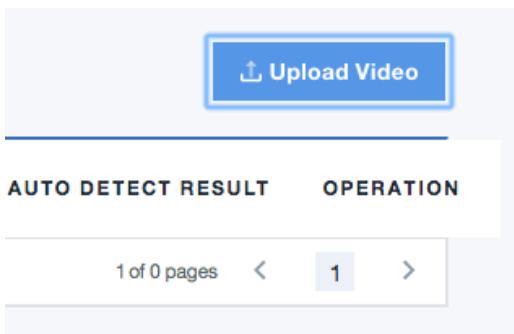


Figure 47: Select “Upload Video” to add new videos for training.

Label Video Training Data

Training a deep learning model based on video requires selecting at least five frames from the data set, and labeling objects within the frames.

Note

Video files are made of hundreds or thousands of frames. A minimum of five frames must be labeled. Object recognition performance improves as more frames are labeled. Multiple videos may be labeled, but only one is required.

Click on the “operation” menu icon for a video, and choose “Add Label” to launch the video labeling editor.

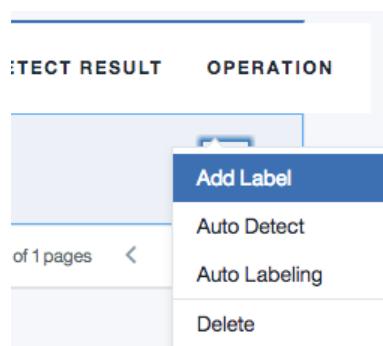


Figure 48: Click on "Add Label" for a video to select frames and label video contents.

The video label editor shows a copy of the video. Click the “play” triangle icon to play the video. PowerAI Vision includes two ways to capture frames:

- Manual: Users select representative frames based on best judgement at any timestamp
- Auto: PowerAI Vision selects frames periodically (e.g. “every 10 seconds”)

Choose “Manual,” drag the video preview slider to an interesting frame, and click “Capture Frame.”

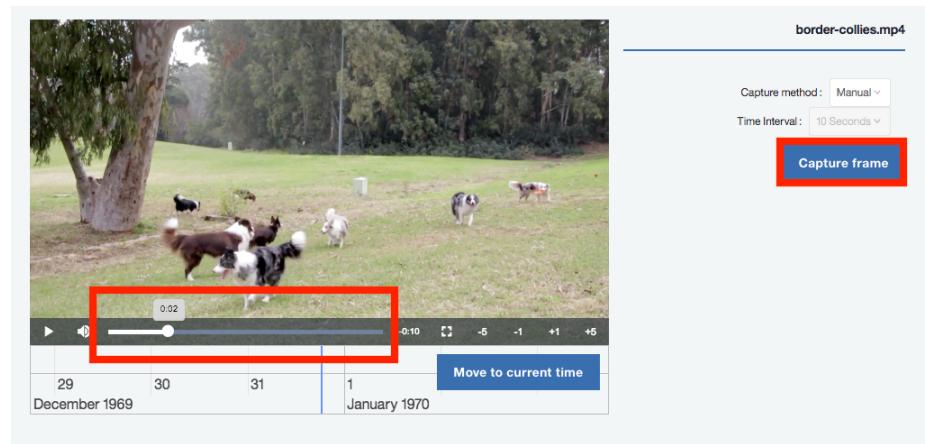


Figure 49: Frames must be "captured" prior to labeling.

Create at least one new object type tag. A data set may have many types of tags. Multiple different tags can appear in the same video. Tags may also overlap. Click “Save” to save the tags for the current frame before proceeding to the next frame.

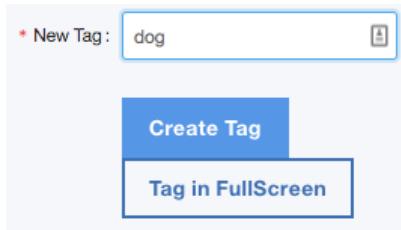


Figure 50: Create at least one new tag before attempting to label objects in a video.

Use the slider to choose a new frame, and repeat this process until at least five frames are labeled.

After at least one tag is “saved” in a video, a new timeline will appear showing the labeled frames from the video. Click on each thumbnail to go to that frame, and edit any tags in the frame as needed (adding, moving, or deleting). Click “Save” to save updates as needed.

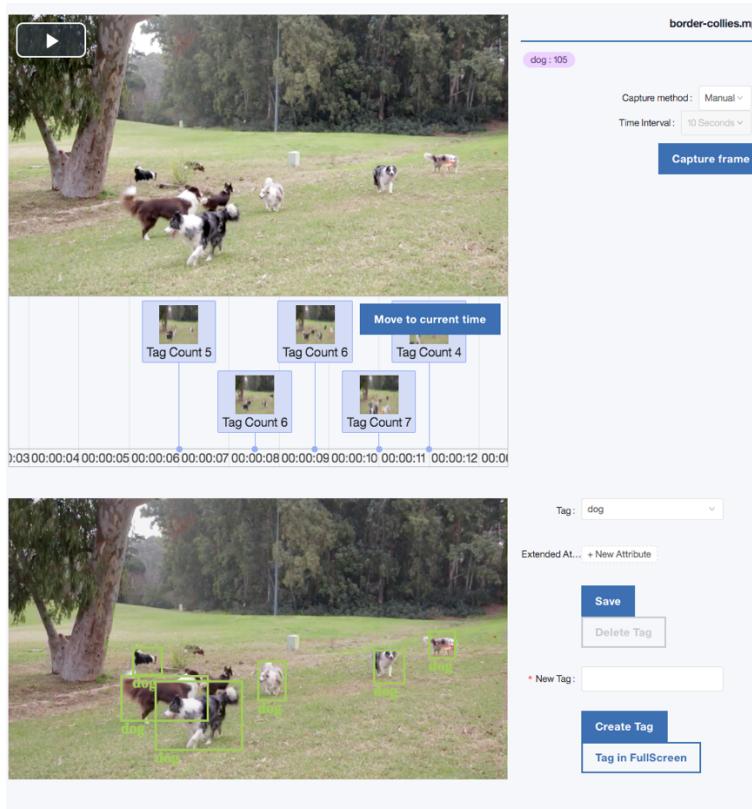


Figure 51: PowerAI Vision will show labeled frames in a separate timeline, showing how many tags are in each frame.

Export Labels for Training

Click on “Data Set” to return to the list of all data sets. Select the target data set, and choose “Export Labels” to set up a data set in PowerAI Vision for training a deep learning model.

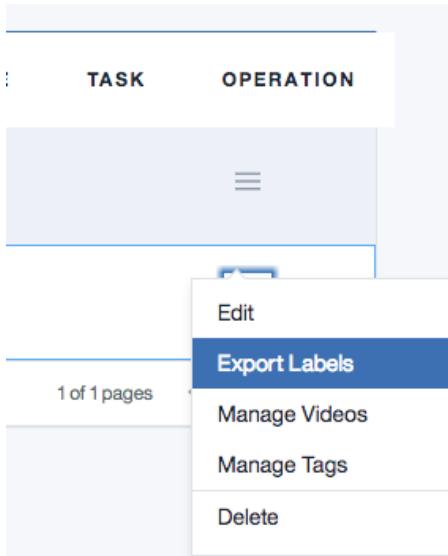


Figure 52: Data must be "exported" from the Video platform to PowerAI Vision's base application for model training and deployment.

Note

Model management, training lifecycle management, and deployment of models occurs in the PowerAI Vision UI as an "Object Detection" data set, task, and API.

Train an Object Detector Based on Labeled Video Data

Labeled video data appears in PowerAI Vision's main UI under "My Data Sets," and follows a naming convention of "dataset_yyyymmddhhmmss" based on the date the data set was exported. If a video data set is exported multiple times, new static image data sets will be created, allowing users to train different versions of models based on each data set's contents.

Important

Changes to the data set in the PowerAI Vision Main Object Detection UI will not be reflected in the PowerAI Vision Video UI. Likewise, changes in the Video UI must be "exported" to the Main UI.

Refresh the PowerAI Vision Main UI, and select "My Data Sets" to view the newly-created data set based on video data.

Figure 53: Click on "My Data Sets" to view the exported object detection data from the PowerAI Video platform.

As with static image object detection, train a new model by selecting "My DL Tasks" → "Create New Task" and select "Object Detection." Name the detector task as needed. Start the training job, and wait for it to complete.



Figure 54: PowerAI Vision displays various metrics showing real-time training job status as models converge over time.

Testing the Trained Model on Video

After object detection training is completed, from "My Trained Models" choose "Deploy" to create a RESTful endpoint.

On the IBM PowerAI Vision main page, select "My Web APIs" to enter the API Management page. Click the "Run Test" button to access the API test page.

URL	Categories	Accuracy	Deployed At	Operation
/diapis/23619197-c2d6-4817-85fa-04023b1adbf9	Acridotheres, Acrocephal...	0.92500	2017-07-25 11:21:43	<button>Run Test</button> <button>Actions ▾</button>

Figure 55: Selecting "Run Test" to try out the API by uploading an image or attempting to classify an image hosted at a URL.

To test the model on video, select "API Showcase" and "Video Detection."

Figure 56: Choose "API Showcase" and "Video Detection" to upload a test video.

Choose “Upload Video,” select a video for upload, then choose “Detect” to find all labeled objects in a validation video.

Note

Objects found during the test will NOT be added to the training data set. To automatically discover and add labels to a data set, see “Autonomous Labeling” below.



Figure 57: Click “Detect” to start the object recognition job for the test video.

After a few moments, a labeled video player will appear below. Click “Play” to view the labeled video.

Note

For efficiency of validation, frames will be labeled once per second. PowerAI Vision can label each frame in a video if desired via API calls.

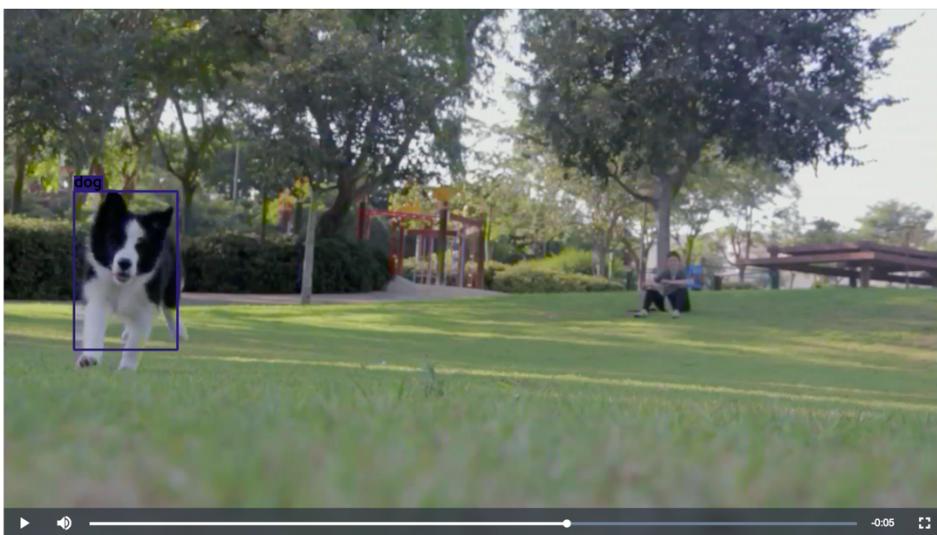
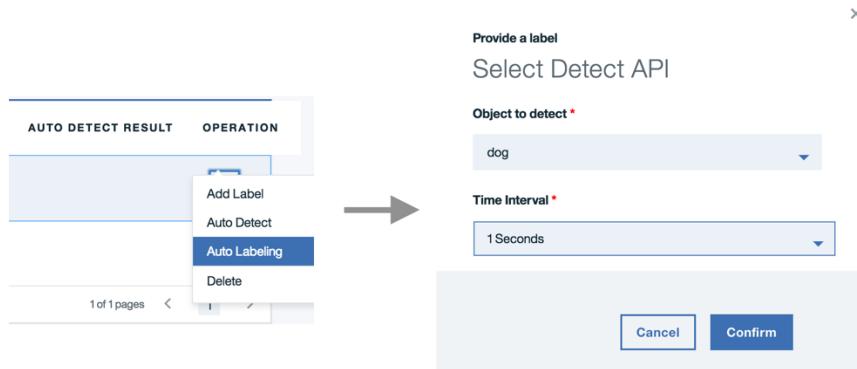


Figure 58: Validation video showing labeled objects based on a model trained earlier.

Autonomous Labeling of Video

With a trained model, PowerAI Vision can assist in labeling of additional frames in a video, greatly-expanding a training set. In the PowerAI Vision Video Data Platform’s “Dataset” view, choose a video, choose the “menu” icon, and choose “Auto Labeling.” Select an object detector, choose a time interval, and click “Confirm” to start auto-detection.



During the auto-labeling process, the video dashboard will show a real-time indicator of newly-detected objects in each frame. Once complete, click on the “menu” icon for the video, and choose “Add label” to review the automatically-created labels. IBM PowerAI Vision’s auto-labeling improves as training data grows, but is not perfect. Be sure to review the automatically-detected labels and edit or delete incorrectly-labeled objects, being sure to click “Save” to save changes.

After reviewing the newly-expanded training data, repeat the “export” process above for the data set, and train a new model based on the expanded data set. Repeat validation and observe any changes in accuracy.

Note

IBM PowerAI Vision supports object detection on a video stream for validation purposes. As with a static image object detector, deployment of the object detection API must fit into a broader image processing pipeline.

Appendix A. System Requirements

Hardware Requirements

- ✓ CPU: POWER8, at least 8 hardware cores
- ✓ Memory: 128GB memory
- ✓ GPU: At least 1 GPU; Tesla K40 or newer GPU. GPU memory of 1 GPU core > 4GB.
- ✓ Disk: > 20GB
- ✓ Network: At least 1 TCP/IP interface

Software Requirements

- ✓ Ubuntu 16.04 or RHEL74
- ✓ Docker.io 1.12.0 or newer, or docker-ce
- ✓ docker-compose v1.7 or newer
- ✓ NVidia GPU driver 384.81 or newer

Note

PowerAI Vision's Technology Preview release is limited to a single system. For multi-node deployments, contact IBM.

Appendix B. Installation and Validation

Installation Guide

- ✓ **Step 1: Prerequisite installation**
 - NVIDIA GPU driver (>384.81), either docker.io, docker-ce, or docker-ee, and docker-compose should be installed.
- ✓ **Step 2: Uninstall Any Previous PowerAI Vision Installs (If Applicable)**

Note

Please uninstall any previous installations prior to updating to a newer version. If IBM PowerAI Vision is not installed, skip this step.

- Stop PowerAI Vision

```
# /opt/powerai-vision/bin/powerai_vision_stop.sh
```
- Purge previously-installed images and data

```
# /opt/powerai-vision/bin/purge_image.sh  
# /opt/powerai-vision/bin/purge_data.sh
```
- Uninstall IBM PowerAI Vision

```
# dpkg -r powerai-vision
```
- Delete Data Directories

```
# rm -rf /opt/powerai-vision/data
```

- ✓ **Step 2: IBM PowerAI Vision Installation**

- Install IBM PowerAI Vision deb package.

Ubuntu:
\$ sudo dpkg -i powerai-vision_1.x.x_ppc64el.deb
RHEL:
sudo yum install powerai-vision-1.x.x.rpm
- Verify that necessary TCP/IP ports are free.

\$ sudo /opt/powerai-vision/sbin/check_port.sh
- Generate Docker images for PowerAI Vision.

\$ sudo /opt/powerai-vision/bin/build_image.sh
- Configure the externally-accessible IP address of the server.

\$ vi /opt/powerai-vision/bin/config.sh

Note

Edit the EXTERNAL_IP and K8S_SERVICE_IP_RANGE as needed for your deployment.

- ✓ **Step 3: IBM PowerAI Vision service start**
 - Start the IBM PowerAI Vision service.
 \$ sudo /opt/powerai-vision/bin/powerai_vision_start.sh
 - In a web browser, visit the PowerAI Vision GUI to get started.
- ✓ **Step 4: IBM PowerAI Vision service stop**
 - To stop IBM PowerAI Vision:
 \$ sudo /opt/powerai-vision/bin/powerai_vision_stop.sh
- ✓ **Step 5: (Optional) Reset configuration settings (preserving data sets)**
 - To reset service configuration settings (e.g. IP addresses) start PowerAI Vision with `RESET_CONFIG=true` set:
 \$ sudo RESET_CONFIG=true /opt/powerai-vision/bin/powerai_vision_start.sh

References

A series of videos are made available on YouTube to help you build models and understand the practical use cases of the deep learning models. We recommend you review the following:

- [PowerAI Vision Demo: Build your own image classification API easily](#)
- [PowerAI Vision demo: Enable deep learning capability from data center to car](#)
- [PowerAI Vision demo: Build your own object detection API easily](#)
- [Download sample dataset for classifying breeds of dogs from stanford.edu](#)

IBM Copyright Information

© Copyright IBM Corporation 2017

IBM Corporation
Route 100
Somers, NY 10589

Produced in the United States of America
January 2018

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

THE INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.

The client is responsible for ensuring compliance with laws and regulations applicable to it. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the client is in compliance with any law or regulation.