

[ML on GCP C8] Training with Pre-built ML Models using Cloud Vision API and AutoML

2 hours

Free

[Rate Lab](#)

Overview

Duration is 1 min

In this lab, you will experiment with pre-built models so there's no coding. First we'll start with the pre-trained Vision API where we don't need to bring our own data and then we'll progress into AutoML for more sophisticated custom labelling that we need.

What you learn

In this lab, you learn how to:

- Setup API key for ML Vision API
- Invoke the pretrained ML Vision API to classify images
- Review label predictions from Vision API
- Train and evaluate custom AutoML Vision image classification model
- Predict with AutoML on new image

Setup

For each lab, you get a new GCP project and set of resources for a fixed time at no cost.

1. Make sure you signed into Qwiklabs using an **incognito window**.

2. Note the lab's access time (for example, **02:00:00** and make sure you can

finish in that time block.

There is no pause feature. You can restart if needed, but you have to start at the beginning.


3. When ready, click





4. Note your lab credentials. You will use them to sign in to Cloud Platform Console.

Open Google Console

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. [Learn more.](#)

Username
google2876526_student@qwiklabs.n 

Password
TG959yrKDX 

GCP Project ID
qwiklabs-gcp-0855e773352d3560 

[New to labs? View our introductory video!](#)

5. Click **Open Google Console**.

6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

If you use other credentials, you'll get errors or **incur charges**.

7. Accept the terms and skip the recovery resource page.

Do not click **End Lab** unless you are finished with the lab or want to restart it. This clears your work and removes the project.

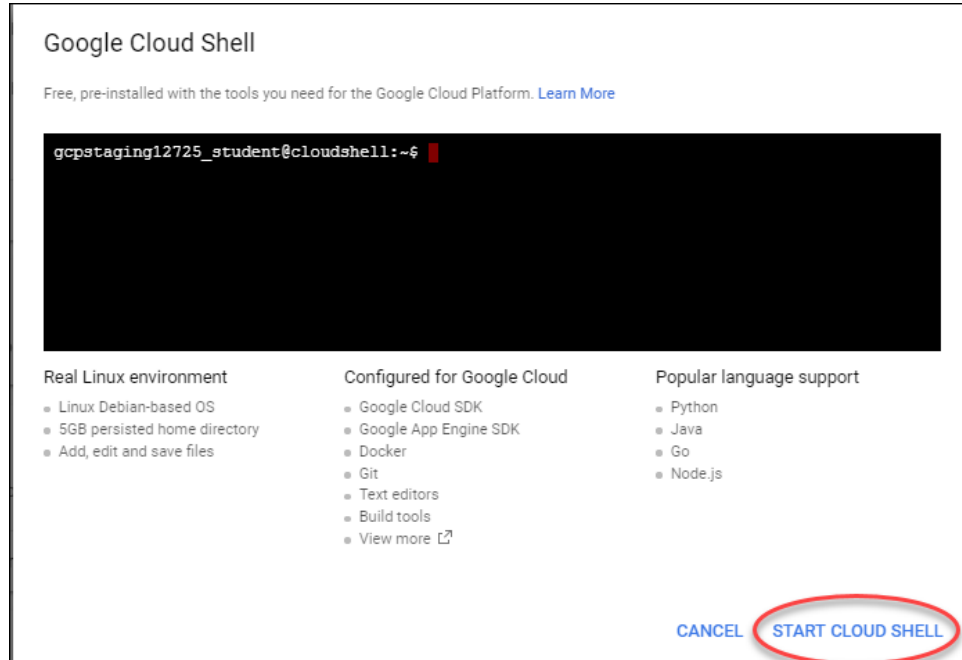
Activate Google Cloud Shell

Google Cloud Shell provides command-line access to your GCP resources.

From the GCP Console click the **Cloud Shell** icon on the top right toolbar:

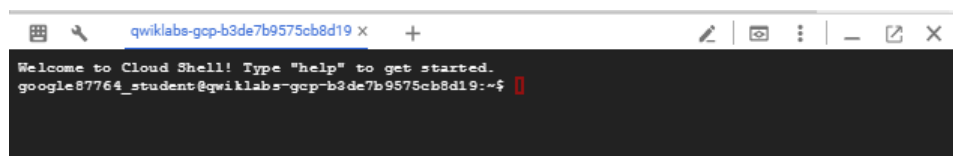


Then click **START CLOUD SHELL**:



You can click **START CLOUD SHELL** immediately when the dialog comes up instead of waiting in the dialog until the Cloud Shell provisions.

It takes a few moments to provision and connects to the environment:



The Cloud Shell is a virtual machine loaded with all the development tools you'll need. It offers a persistent 5GB home directory, and runs on the Google Cloud, greatly enhancing network performance and authentication.

Once connected to the cloud shell, you'll see that you are already authenticated and the project is set to your *PROJECT_ID*:

```
gcloud auth list
```

Output:

```
Credentialed accounts:
- <myaccount>@<mydomain>.com
```

(active)

Note: `gcloud` is the powerful and unified command-line tool for Google Cloud Platform. Full documentation is available on [Google Cloud gcloud Overview](#). It comes pre-installed on Cloud Shell and supports tab-completion.

```
gcloud config list project
```

Output:

```
[core]
project = <PROJECT_ID>
```

Enable Vision API and create API Key

Duration is 1 min

To get an API key:

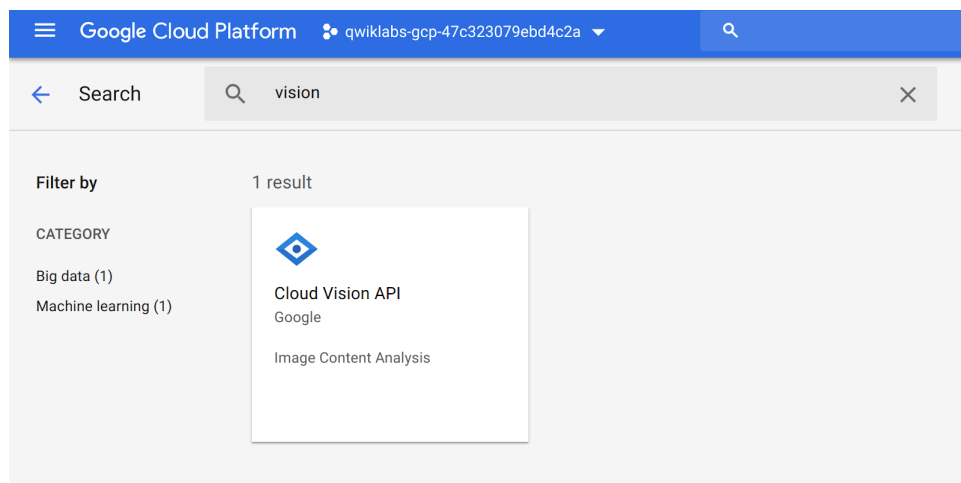
Step 1

In your GCP Console, click on the **Navigation menu** (), select **APIs and**

services and select **Library**.

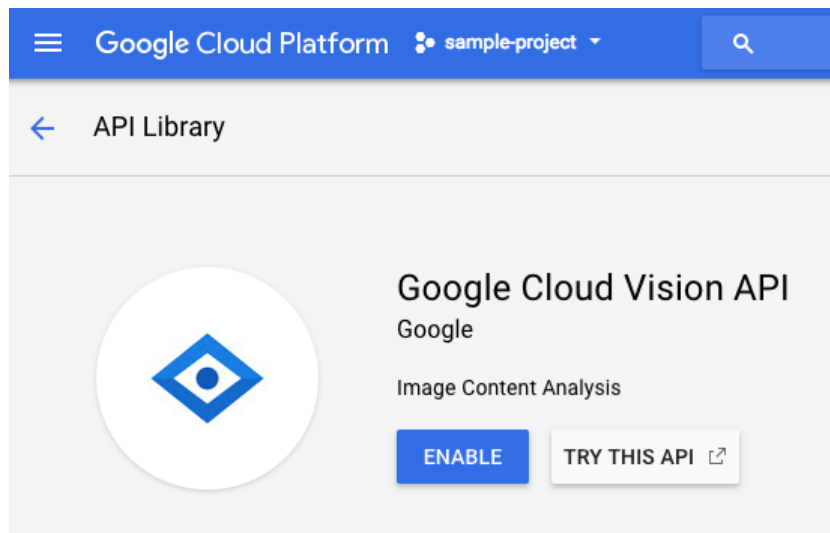
Step 2

In the search box, type **vision** to find the **Cloud Vision API** and click on the hyperlink.



Step 3

Click **Enable** if necessary.



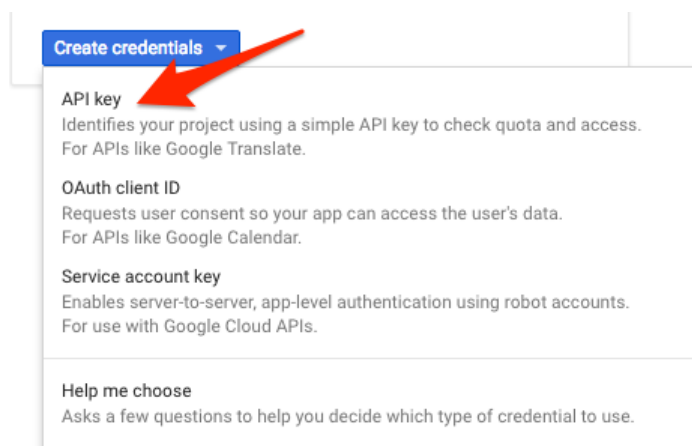
Step 4

In your GCP Console, click on the **Navigation menu** (), select **APIs &**

Services and select **Credentials**.

Step 5

If you do not already have an API key, click the **Create credentials** button and select **API key**. Once created, copy the API key and then click **Close**.



Step 6

In Cloud Shell, export your API key as environment variable. Be sure to replace **<YOUR_API_KEY>** with the key you just copied.


```
export API_KEY=<YOUR_API_KEY>
```

Create storage bucket and store data file

Duration is 2 min

Create a bucket using the GCP console:

Step 1

In your GCP Console, click on the **Navigation menu** (), and

select **Storage**.

Step 2

Click on **Create bucket**.

Step 3

Choose a Regional bucket and set a unique name (use your project ID because it is unique). Then, click **Create**.

Step 4

Download the image below by right-clicking and saving it locally (save it as cirrus.png):



Step 5

Upload the file you just downloaded into the storage bucket you just created using the **upload files** button.

Step 6

In Cloud Shell, run the command below to make the file publicly accessible.

```
gsutil acl ch -u AllUsers:R  
gs://<YOUR-BUCKET>/*
```

Click the **Public** link to confirm the file loads correctly (refresh bucket if needed).

Buckets / qwiklabs-gcp-ab2ba3bed6f2808b

<input type="checkbox"/> Name	Size	Type	Storage class	Last modified	Public access	Encryption
<input checked="" type="checkbox"/> cirrus.png	136.56 KB	image/png	Regional	9/4/18, 10:46 PM	Public	Google-managed key

Label detection with Vision API

Step 1

First, you will create a Vision API request in a json file. Using gcloud or your preferred command line editors (nano, vim, or emacs) create a **request.json** file and inserting the following:

Note: Replace **my-bucket-name** with the name of your storage bucket.

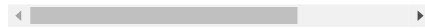
```
{  
  "requests": [  
    {  
      "image": {  
        "source": {  
          "gcsImageUri":  
            "gs://my-bucket-  
name/cirrus.png"  
        }  
      },  
      "features": [  
        {  
          "type":  
            "LABEL_DETECTION",  
          "maxResults": 10  
        }  
      ]  
    }  
  ]  
}
```

Save the file.

Step 2

The label detection method will return a list of labels (words) of what's in your image. Call the Vision API with curl:

```
curl -s -X POST -H "Content-Type: application/json" --data-binary @request.json https://vision.googleapis.com/v1/image:annotate?key=${API_KEY}
```



Your response should look something like the following:

```
{
  "responses": [{
    "labelAnnotations": [{
      "mid":
      "/m/01bqvp",
      "description": "sky",
      "score": 0.9867201,
      "topicality": 0.9867201
    }, {
      "mid":
      "/m/0csby",
      "description": "cloud",
      "score": 0.97132415,
      "topicality": 0.97132415
    }, {
      "mid":
      "/m/01g5v",
      "description": "blue",
      "score": 0.9683707,
      "topicality": 0.9683707
    }, {
      "mid":
      "/m/02q7ylj",
      "description": "daytime",
      "score": 0.9555285,
      "topicality": 0.9555285
    }, {
      "mid":
      "/m/01ctsf",
      "description": "atmosphere",
      "score": 0.92822105,
      "topicality": 0.92822105
    }, {
      "mid":
      "/m/0csh5",
      "description": "cumulus",
      "score": 0.8386173,
      "topicality": 0.8386173
    }
  ]
}
```



```

        "mid":
        "/g/1lk2xz7mr",

        "description": "meteorological
phenomenon",

        "score": 0.75660443,

        "topicality": 0.75660443
    }, {
        "mid":
        "/m/026fm63",

        "description": "calm",

        "score": 0.72833425,

        "topicality": 0.72833425
    }, {
        "mid":
        "/m/03w43x",

        "description": "computer
wallpaper",

        "score": 0.6601879,

        "topicality": 0.6601879
    }, {
        "mid":
        "/m/0d1n2",

        "description": "horizon",

        "score": 0.63659215,

        "topicality": 0.63659215
    }
    ]
}

```

Note that the Vision API does recognize it's an image with SKY and CLOUD but the type of cloud is incorrectly labeled as a cumulus cloud. We need a more specific model with our own labeled training data to get a more accurate model.

Setup AutoML Vision

AutoML Vision enables you to train machine learning models to classify your images according to your own defined labels. In this section, we will upload images of clouds to Cloud Storage and use them to train a custom model to recognize different types of clouds (cumulus, cumulonimbus, etc.).

Step 1

In your GCP Console, click on the **Navigation menu** (), click on **Vision**.

ARTIFICIAL INTELLIGENCE



Talent Solution



ML Engine



Natural Language



Translation



Vision 



Step 2

Select the GCP account created by qwiklabs (if prompted) and allow AutoML access
:

AutoML wants to access your Google Account

 gcpstaging22415_student@qwiklabs.net

This will allow AutoML to:

- View and manage your data across Google Cloud Platform services ⓘ
- View your data across Google Cloud Platform services ⓘ

Make sure you trust AutoML

You may be sharing sensitive info with this site or app. Learn about how AutoML will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your [Google Account](#).

[Learn about the risks](#)

Cancel

Allow

Step 3

Click on **Get started with AutoML**.

Create a custom model

Build your dataset with AutoML Vision. Add your own images and train a model to recognize categories you define. No machine learning experience required.

→ Get started with AutoML

→ Learn about AutoML Vision

Use the pre-trained model

Use Google's proven pre-trained model for general image labeling; face, logo, and landmark detection; and more. Or use the Product Search API to detect products in images, using your provided catalog.

→ Learn about the Vision API

Step 4

Choose the correct GCP project created by qwiklabs and click **Continue**.

Specify Google Cloud project

❗ Please select a valid project id to continue

Project ID

qwiklabs-gcp-97b69f48fa211335

Make sure to use your project's ID, and not its name or number. [Learn more](#)


Need to create a new Google Cloud Platform project? [Get started](#)

CONTINUE

Step 5

Next, Click on **Go To Billing** and choose to **Go to linked billing account**:

Finish setting up your Google Cloud project

 **Let's set up your project and grant AutoML Vision access**

You'll only have to do these steps once for your project.

1. Enable billing

You'll need to enable billing for your Google Cloud project to create custom models.

[GO TO BILLING](#)

2. Enable the required APIs and service accounts

Clicking "Set up now" will also create a bucket on Google Cloud Storage to store your models' images. You can al

[SET UP NOW](#)

[MANUAL SETUP](#)

[CHECK AGAIN](#)

[SELECT DIFFERENT PROJECT](#)

Billing

You have multiple billing accounts

Billing account "Development Qwiklabs Internal Master" is linked to this project

[Go to linked billing account](#) or

[Manage billing accounts](#)


Step 6

Confirm the step was successful.

Overview

Development Qwiklabs Internal Ma... ▼

 RENAME BILLING ACCOUNT

 Project "qwiklabs-gcp-dabd0aa7da42381e" is linked to billing account "Development Qwiklabs Internal Master". Y


Billing account ID: 00FB9B-16ECAA-B33D37

Billing management for "qwiklabs-gcp-dabd0aa7da42381e"


Project name	Project ID
qwiklabs-gcp-dabd0aa7da42381e	qwiklabs-gcp-dabd0aa7da42381e ⋮

Step 7

Now setup the necessary APIs and service accounts by clicking on **Set Up Now**.

 AutoML Vision **BETA**

Finish setting up your Google Cloud project

 **Let's set up your project and grant AutoML Vision access**


You'll only have to do these steps once for your project.



- 1. Enable billing**
You'll need to enable billing for your Google Cloud project to create custom models.
[GO TO BILLING](#)
- 2. Enable the required APIs and service accounts**
Clicking "Set up now" will also create a bucket on Google Cloud Storage to store your models' images. You can al
[SET UP NOW](#) [MANUAL SETUP](#)

[CHECK AGAIN](#) [SELECT DIFFERENT PROJECT](#)

Step 8

You will be redirected on to the AutoML Vision console.

 AutoML Vision **BETA** [+ NEW DATASET](#)

Datasets

Stage training files

Step 1

Back on your GCP console, check under storage buckets to confirm a new bucket created by AutoML Vision API. The name is similar to your project id, with the suffix *vcm* (for example : *qwiklabs-gcp-dabd0aa7da42381e-vcm*).

[Columns ▾](#)

Buckets

<input type="checkbox"/> Name	Default storage class ?
<input type="checkbox"/> qwiklabs-gcp-dabd0aa7da42381e	Regional
<input type="checkbox"/> qwiklabs-gcp-dabd0aa7da42381e-vcn	Regional

Copy the new bucket name so you can use it in the next step.

Step 2

Set the bucket as an environment variable.

```
export BUCKET=
<YOUR_AUTOML_BUCKET>
```

Step 3

Next, using the `gsutil` command line utility for Cloud Storage, copy the training images into your bucket:

```
gsutil -m cp -r gs://automl-
codelab-clouds/* gs://{BUCKET}
```



After the copy, confirm that you have 3 folders in your storage bucket.

qwiklabs-gcp-dabd0aa7da42381e-vcn

[Objects](#) [Overview](#) [Permissions](#)

[Upload files](#) [Upload folder](#) [Create folder](#) [Delete](#)

[Buckets](#) / qwiklabs-gcp-dabd0aa7da42381e-vcn

<input type="checkbox"/> Name
<input type="checkbox"/>  cirrus/
<input type="checkbox"/>  cumulonimbus/
<input type="checkbox"/>  cumulus/

Create dataset

Now that your training data is in Cloud Storage, you need a way for AutoML Vision to find them. To do this you'll create a CSV file where each row contains a URL to a training image and the associated label for that image. This CSV file has been created for you, you just need to update it with your bucket name.

Step 1

To do that, copy this file to your Cloud Shell instance:

```
gsutil cp gs://automl-codelab-  
metadata/data.csv .
```

Step 2

Then run the following command to update the CSV with the files in your project:

```
sed -i -e  
"s/placeholder/${BUCKET}/g"  
./data.csv
```

Step 3

Now you're ready to upload this file to your Cloud Storage bucket:

```
gsutil cp ./data.csv  
gs://${BUCKET}
```


Confirm that you see the CSV file in your bucket.

Step 4

Navigate back to the [AutoML Vision UI](#).

Create a custom model

Build your dataset with AutoML Vision. Add your own images and train a model to recognize categories you define. No machine learning experience required.

 Get started with AutoML

Note: If you've previously created a dataset with AutoML vision, you will see a list of datasets instead. In this case, click **+ New Dataset**.

Type "clouds" for the Dataset name.

Choose **Select a CSV file on Cloud Storage** and enter the URL of the file you just uploaded - `gs://your-project-name-vcn/data.csv`

Create dataset

Dataset name
clouds



Import images

To build a custom model, you first need to import a set of images to train it. Generally the more images the better. Each image should be categorized with a label (labels are essential for telling the model how to identify an image). Each category should have at least 100 images for best results.

Processed images will be stored on Cloud Storage.

☐ Upload images from your computer

Supports JPG, PNG, ZIP. Maximum 500 files per upload.

SELECT FILES

☒ Select a CSV file on Cloud Storage

The CSV file should be a list of paths to your images on GCS and their labels, if available.

`gs://your-project-name-vcn/data.csv`

For this example, leave "enable multi-label classification" unchecked. In your own projects, you may want to check this box if you want to assign multiple labels per image.

Classification type

☐ Enable multi-label classification

If you have images that may require multiple labels, enable this setting now. Typically requires more training images per label to get good model results.

Select **Create Dataset**.

CREATE DATASET

CANCEL

It will take around 2 minutes for your images to finish importing. Once the import has completed, you'll be brought to a page with all the images in your dataset.

Inspect images

Step 1

After the import completes, you will see the Images tab.

The screenshot shows the 'AutoML Vision' interface with the 'clouds' dataset. The top navigation bar includes 'clouds', 'ADD IMAGES', 'LABELS STATS', and 'EXPORT DATA'. The main interface has tabs for 'IMAGES', 'TRAIN', 'EVALUATE', and 'PREDICT'. The 'IMAGES' tab is active, showing a list of images on the left and a grid of image thumbnails on the right. The list on the left includes 'All images' (60), 'Labeled' (60), and 'Unlabeled' (0). Below this is a filter section with a search bar and a list of labels: 'cirrus' (20), 'cumulonimbus' (20), and 'cumulus' (20). An 'Add label' link is also present. The right side of the interface shows a filter bar with the text 'Type to filter images...', a message box stating 'Make sure to consider best practices for preparing the best training data for your model', and a 'Select all images' button. Below these are two rows of image thumbnails, each labeled 'cumulus'.

Label	Count
All images	60
Labeled	60
Unlabeled	0
cirrus	20
cumulonimbus	20
cumulus	20

Try filtering by different labels (i.e. click cumulus) to review the training images:

This screenshot shows the 'Type to filter...' dropdown menu. It contains a search bar with the text 'Type to filter...' and a list of labels: 'cirrus' (20), 'cumulonimbus' (20), and 'cumulus' (20). An 'Add label' link is also present.

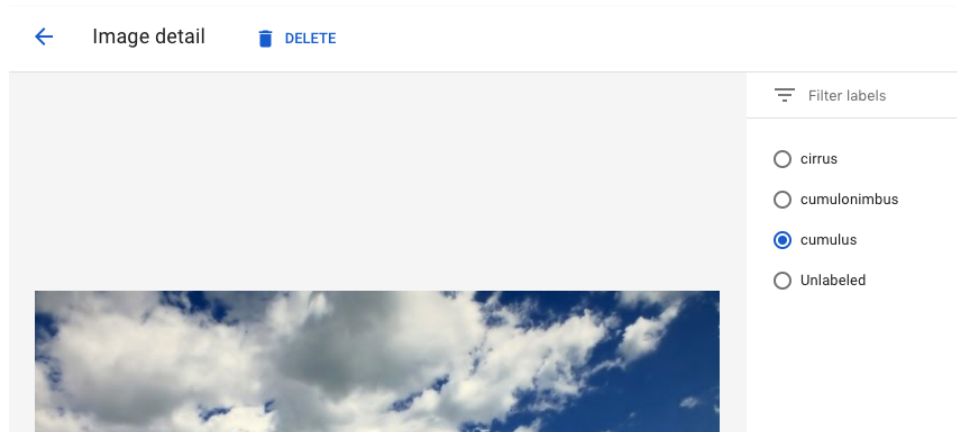
Label	Count
cirrus	20
cumulonimbus	20
cumulus	20

Note: If you were building a production model, you'd want *at least* 100 images per label to ensure high accuracy. This is just a demo so we only used 20

images so that our model will train quickly.

Step 2

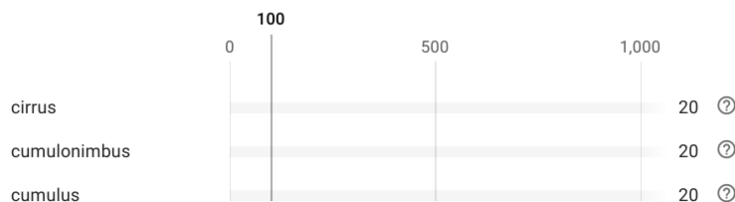
If any image is labeled incorrectly you can click on them to switch the label or delete the image from your training set:



To see a summary of how many images you have for each label, click on Label stats. You should see the following show up on the left side of your browser.

Label stats

Each label should have at least 100 images assigned. Fewer images often result in inaccurate precision and recall scores.



Note: If you are working with a dataset that isn't already labeled, AutoML Vision provides an in-house human labeling service.

Train your model

You're ready to start training your model! AutoML Vision handles this for you automatically, without requiring you to write any of the model code.

Step 1

To train your clouds model, go to the **Train** tab and click **Start Training**.

Enter a name for your model, or use the default auto-generated name, and click **Start Training**.

IMAGES

TRAIN

EVALUATE

PREDICT

Try labeling more images before training

Each label should have at least 100 images assigned. Fewer images often result in inaccurate precision and recall scores.

	100		
cirrus	<div><div></div></div>	20	?
cumulonimbus	<div><div></div></div>	20	?
cumulus	<div><div></div></div>	20	?

Your images will be automatically split into training and test sets, so you can evaluate your model's performance. Unlabeled images will not be used.

START TRAINING

Train new model

Model name

clouds_v20180629232046

Base model

None



Total budget

1 compute hour



To increase your budget, you need at least 1000 labeled images

Data summary

60 labeled images, 3 labels

Your model's success depends on how long you allow it to train, and the number of images and labels in your dataset.

CANCEL

START TRAINING

Since this is a small dataset, it will only take around **5 minutes** to complete.

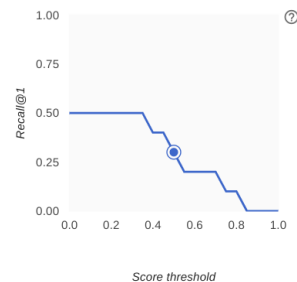
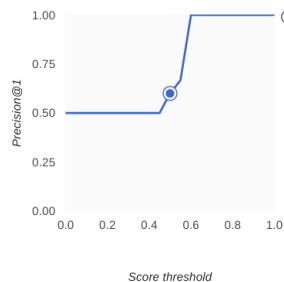
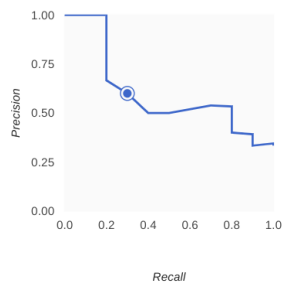
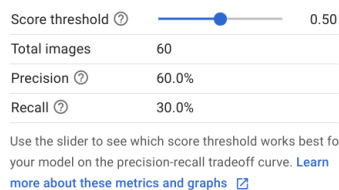
Evaluate your model

Step 1

In the **Evaluate** tab, you'll see information about AUC, precision and recall of the model.



You can also play around with **Score threshold**:



Finally, scroll down to take a look at the **Confusion matrix**.

Confusion matrix

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in orange).

True label	Predicted label			
		cumulus	cirrus	cumulonimbus
cumulus	-	75.0%	25.0%	
cirrus	-	100.0%	-	
cumulonimbus	33.3%	-	66.7%	

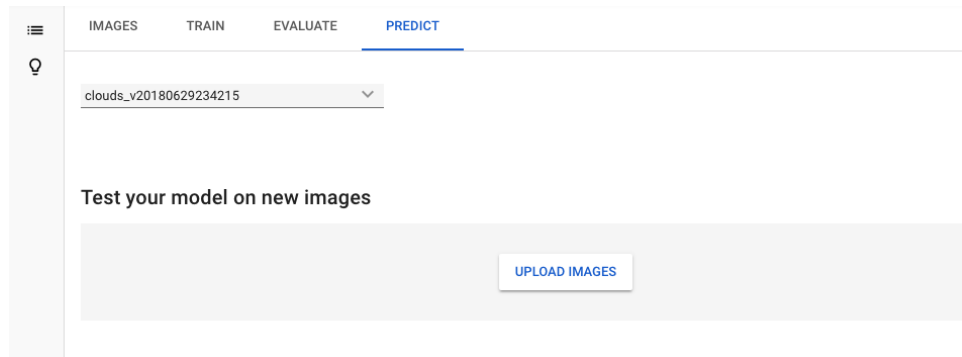
All of this provides some common machine learning metrics to evaluate your model accuracy and see where you can improve your training data. Since the focus for this lab was not on accuracy, skip to the prediction section, but feel free to browse the accuracy metrics on your own.

Generate predictions

Now it's time for the most important part: generating predictions on your trained model using data it hasn't seen before.

Step 1

Navigate to the **Predict** tab in the AutoML UI:



There are a few ways to generate predictions. In this lab, you'll use the UI to upload images. You'll see how your model does classifying these two images (the first is a cirrus cloud, the second is a cumulonimbus).

Step 2

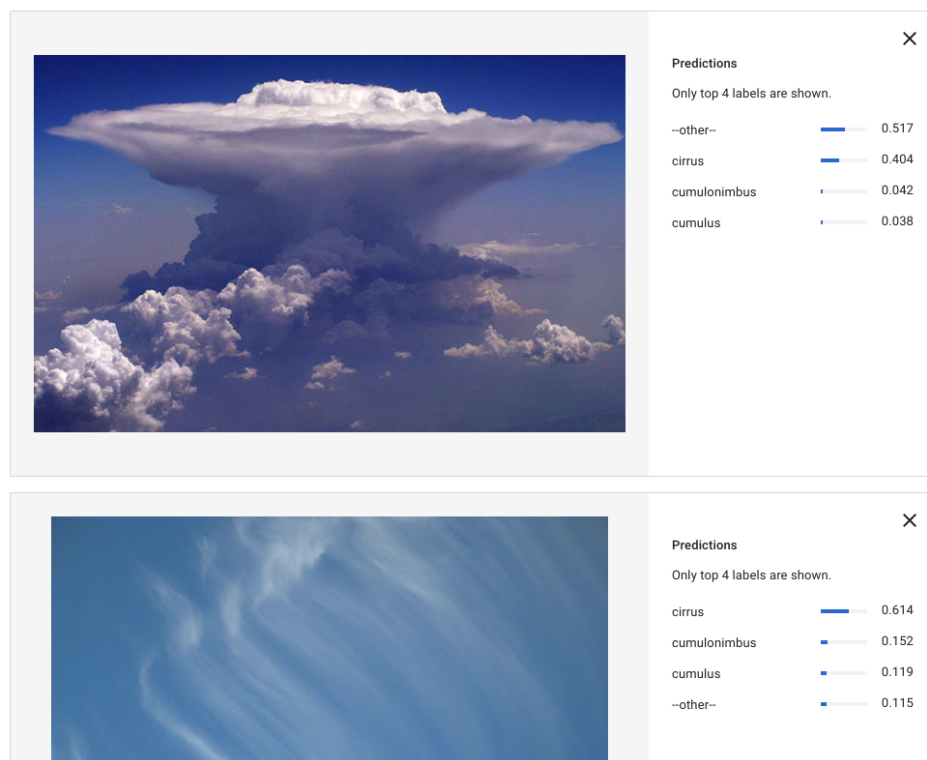
Download these images by right-clicking on each of them:





Step 3

Return to the UI, select **upload images** and upload them to the online prediction UI. When the prediction request completes you should see something like the following:



Pretty cool - the model classified each type of cloud correctly! Does your trained model do better than the 57% CIRRUS cloud above?

Note: In addition to generating predictions in the AutoML UI, you can also use the REST API or the Python client to make prediction requests on your trained model. Check out the tabs for each to see some sample code. You can try it out

by copy/pasting these commands into Cloud Shell and providing an image URL.

End your lab

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you've used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.

Last Tested Date: 12-13-2018

Last Updated Date: 12-14-2018

©2019 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.