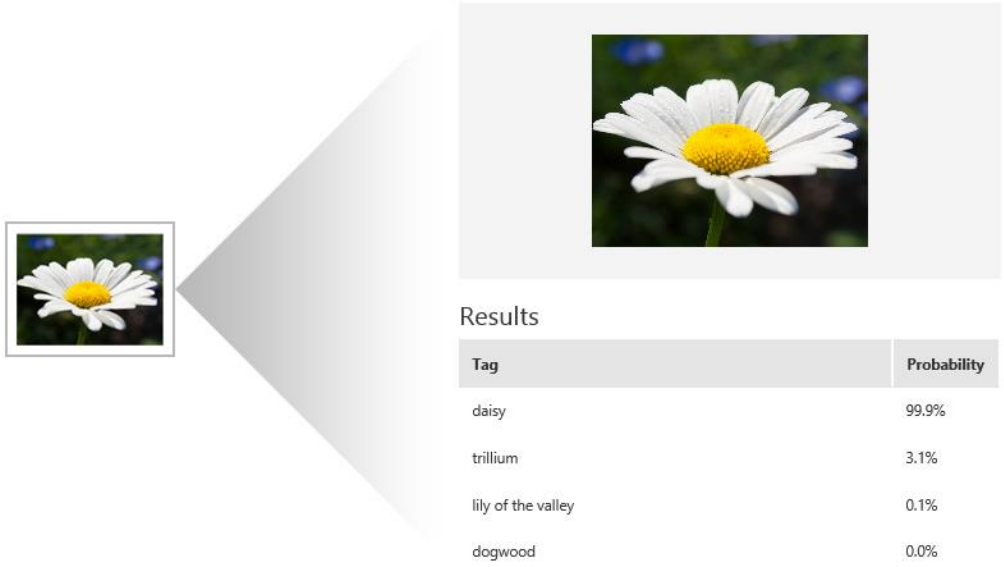
Custom Vision Service

Rachel Phinnemore, Tina Hai, Evan McKenzie
Microsoft Student Partners @ Queen's University

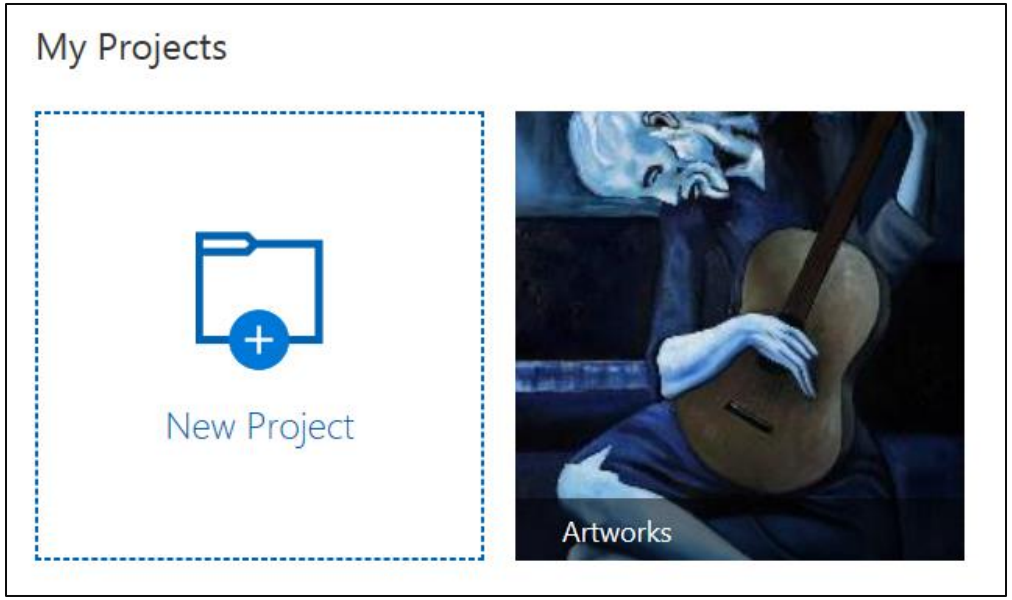
Custom Vision Service

Custom Vision Service leverages the power of machine learning to create and customize state-of-the-art computer vision image recognition models by exposing tools and APIs to build custom image classifiers to training and predict images.



Classifiers and Projects

A **classifier** is a model you build using Custom Vision Service, by using a few training images. **Each classifier you build is in its own project**.



Classifier = Project

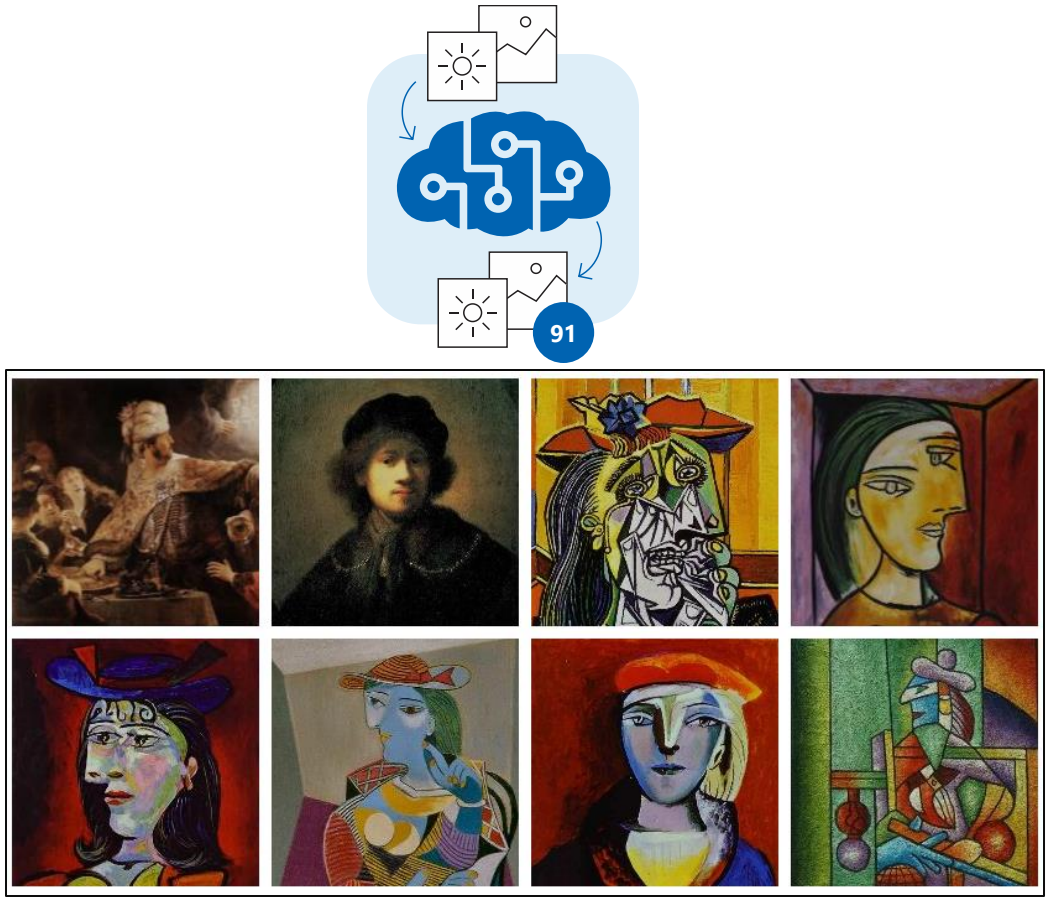
Domains

When you create a project, you select a **domain** for that project. The domain optimizes a classifier for a specific type of object in your images.

- **Food**
  Optimized for dishes you would see on a restaurant menu.
- **Landmark**
  Optimized for recognizable landmarks, both natural and artificial.
- **Retail**
  Optimized for classifying images in a shopping catalog or shopping website.
- **Adult**
  Optimized to better define between adult content and non-adult content.

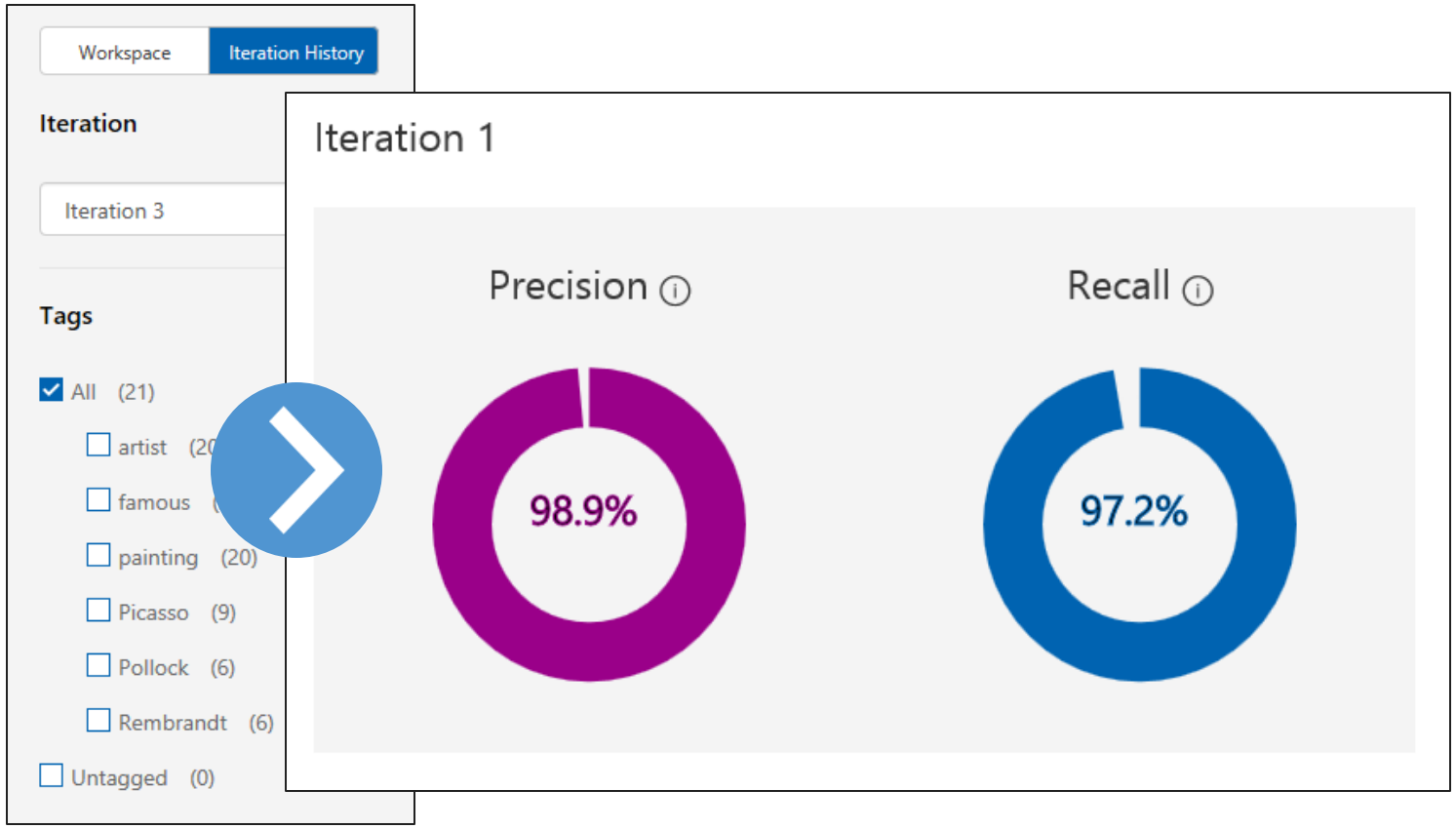
Training Images

To create a high precision classifier, Custom Vision Service needs several **training images**. A training image is a photograph of the image you want Custom Vision Service to classify.



Iteration

Every time you Train or re-train your classifier, you create a new **iteration** of your model.



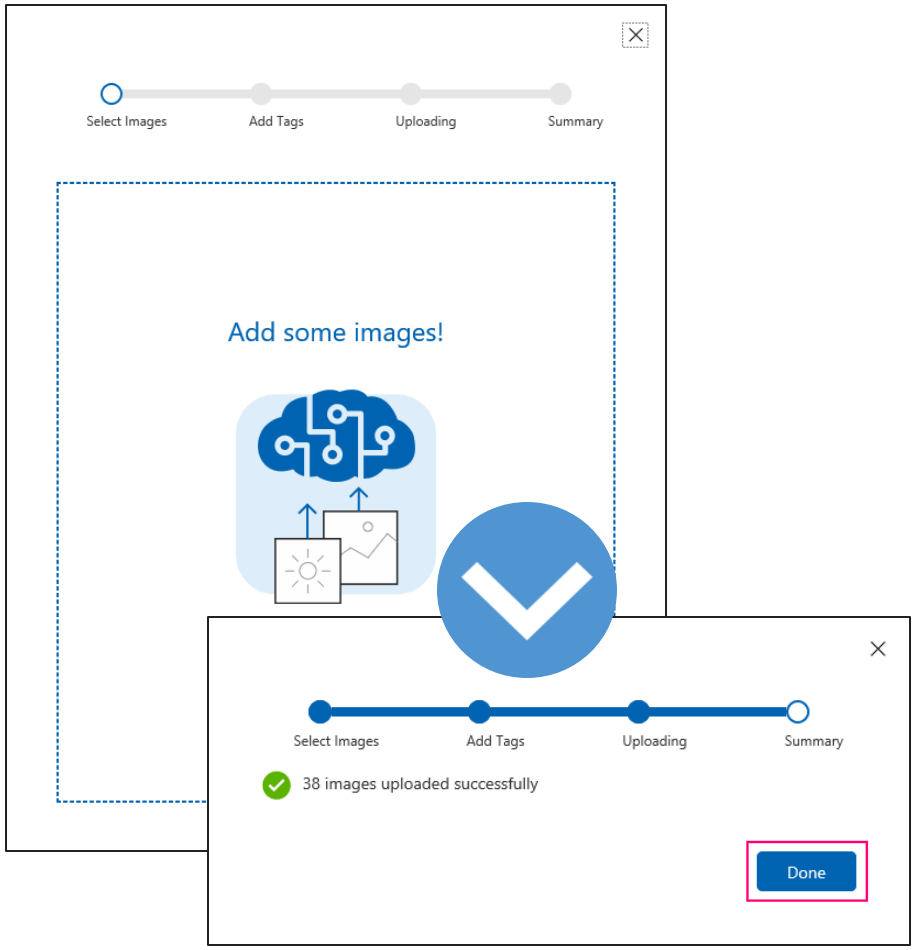
Important Terms

Precision
When you classify an image, how likely is your classifier to correctly classify the image? Out of all images used to train the classifier (Picasso, Rembrandt, and Pollock), what percent did the model get correct? 99 correct tags out of 100 images gives a Precision of 99%.

Recall
Out of all images that should have been classified correctly, how many did your classifier identify correctly? A Recall of 100% would mean, if there were 12 Picasso paintings in the images used to train the classifier, 12 Picasso paintings were found by the classifier.

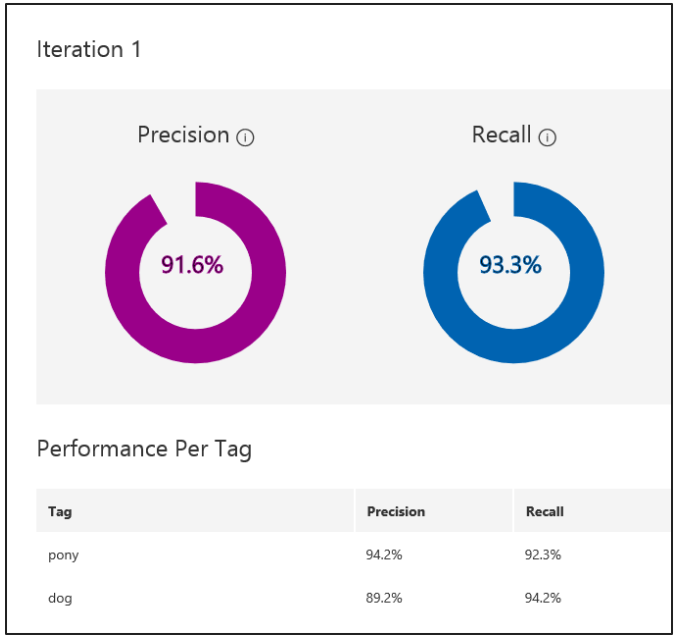
Building a Classifier

- Create a project
- Select a domain
- Add images
- Assign tags to images
- Train the classifier
- Evaluate the classifier



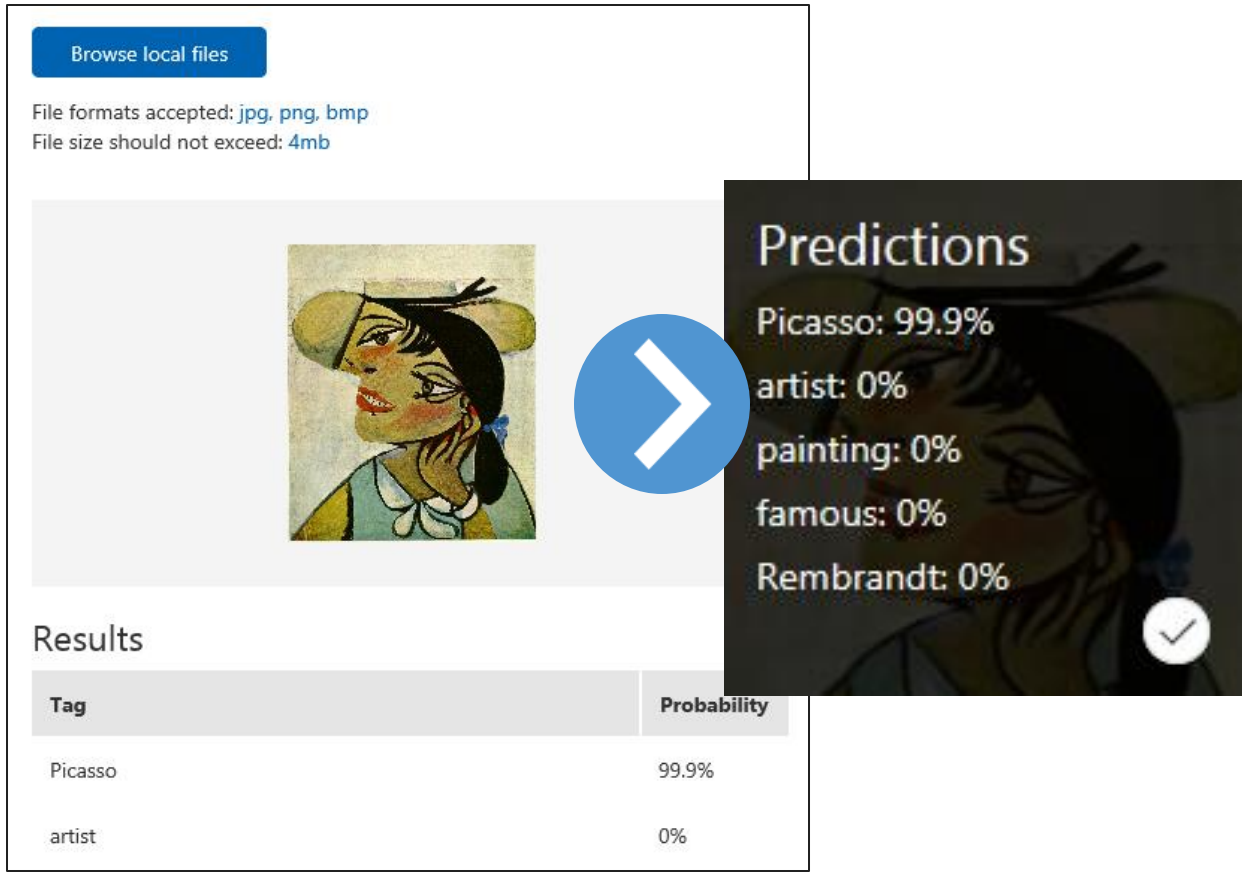
Improving a Classifier

The best way to have a quality classifier is to **add more varied tagged images** (different backgrounds, angles, object size, groups of photos, and variants of types.) Always to train your classifier after you have added more images. **Include images that are representative of what your classifier will encounter in the real world.** Photos in context are better than photos of objects in front of neutral backgrounds, for example.



Testing a Model

After you train your model, you can quickly test it using a locally stored image or an online image. **The test uses the most recently trained iteration.**

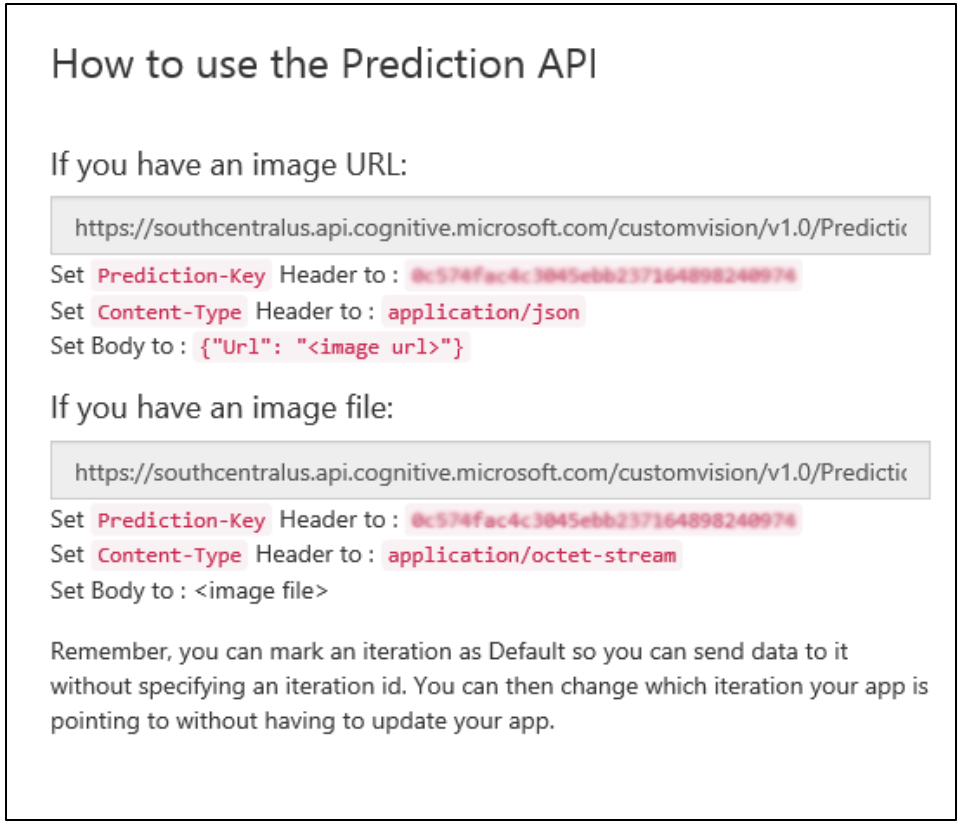


Commonly Used APIs

Training	Prediction
• Create images • Tag images • Create projects • Manage projects • Manage iterations • Create tags • Get account information • Train a project	• Predict images • Predict and save images • Predict image URLs • Predict and save image URLs

Using the Prediction API

After a successful training, the Custom Vision Service can be accessed via an **endpoint** that references the **Project Identifier**, as long as the appropriate **Prediction Key** is passed in the request header.



Prediction API REST Concepts

All actions related to the Custom Vision Service are accessed via **standard REST-based methods**, such as GET and POST against an API endpoint, making it simple to use the Prediction API on any platform or with any programming language.

```
var client = new HttpClient();

// Request headers - replace this example key with your valid
client.DefaultRequestHeaders.Add("Prediction-Key", "13hc77781f

// Prediction URL - replace this example URL with your valid pr
string url = "http://southcentralus.api.cognitive.microsoft.co

HttpResponseMessage response;

// Request body. Try this sample with a locally stored image.
byte[] byteData = GetImageAsByteArray(imageFilePath);

using (var content = new ByteArrayContent(byteData))
{
    content.Headers.ContentType = new MediaTypeHeaderValue("ap
    response = await client.PostAsync(url, content);
    Console.WriteLine(await response.Content.ReadAsStringAsync
}
```

Hands-On Lab

Custom Vision Service

Link: https://docs.microsoft.com/en-us/learn/modules/classify-images-with-custom-vision-service/

Feedback Form:

- https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR_qIoEIl6HpBjLRDb8BfMuxUMTkzRzhaMElaS1daTjhXMlVUQUQ0VU82UC4u