



Azure Computer Vision 4.0 (Florence)

Next generation AI with large foundation model

4th of April 2023

Serge Retkowsky | EMEA AI GBB | serge.retkowsky@microsoft.com

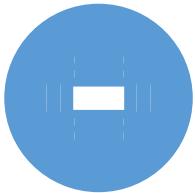
Azure Computer Vision 4.0

7th of March 2023

- We are pleased to announce the ***public preview*** of Microsoft's **Florence foundation model**, trained with billions of text-image pairs and integrated as cost-effective, production-ready computer vision services in [Azure Cognitive Service for Vision](#).
- The improved Vision Services enables developers to create cutting-edge, market-ready, responsible computer vision applications across various industries.
- Customers can now seamlessly digitize, analyze, and connect their data to natural language interactions, unlocking powerful insights from their image and video content to support accessibility, drive acquisition through SEO, protect users from harmful content, enhance security, and improve incident response times.

<https://azure.microsoft.com/en-us/blog/announcing-a-renaissance-in-computer-vision-ai-with-microsofts-florence-foundation-model/>

Florence



Trained with billions
of images and videos



Multi modality for
language & vision



Zero shot capabilities
(millions of objects
can be recognized)



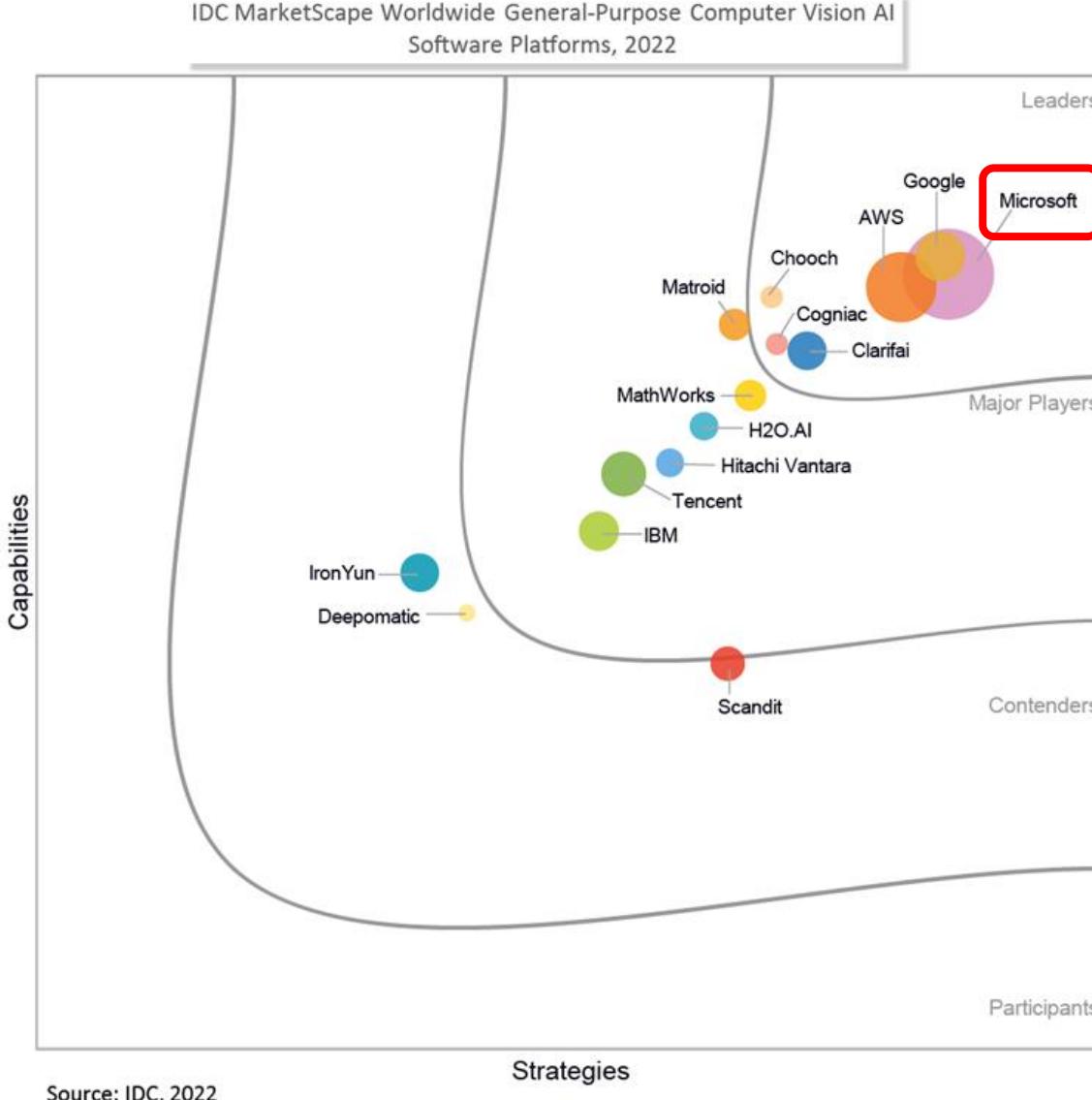
Few shot learning
with Florence for
custom vision models



Enterprise grade
service quality

IDC Computer Vision

2022



[Microsoft named a Leader in the IDC MarketScape: Worldwide General-Purpose Computer Vision AI Software Platform 2022 Vendor Assessment | Azure Blog and Updates | Microsoft Azure](#)

Azure Computer Vision 4 – new features



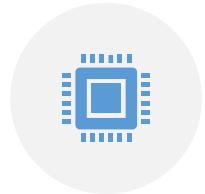
Dense captions: Automatically deliver rich captions, design suggestions, accessible alt-text, SEO optimization, and intelligent photo curation to support digital content.



Image retrieval: Improve search recommendations and advertisements with natural language queries that seamlessly measure the similarity between images and text.



Background removal: Transform the look and feel of images by easily segmenting people and objects from their original background, replacing them with a preferred background scene.

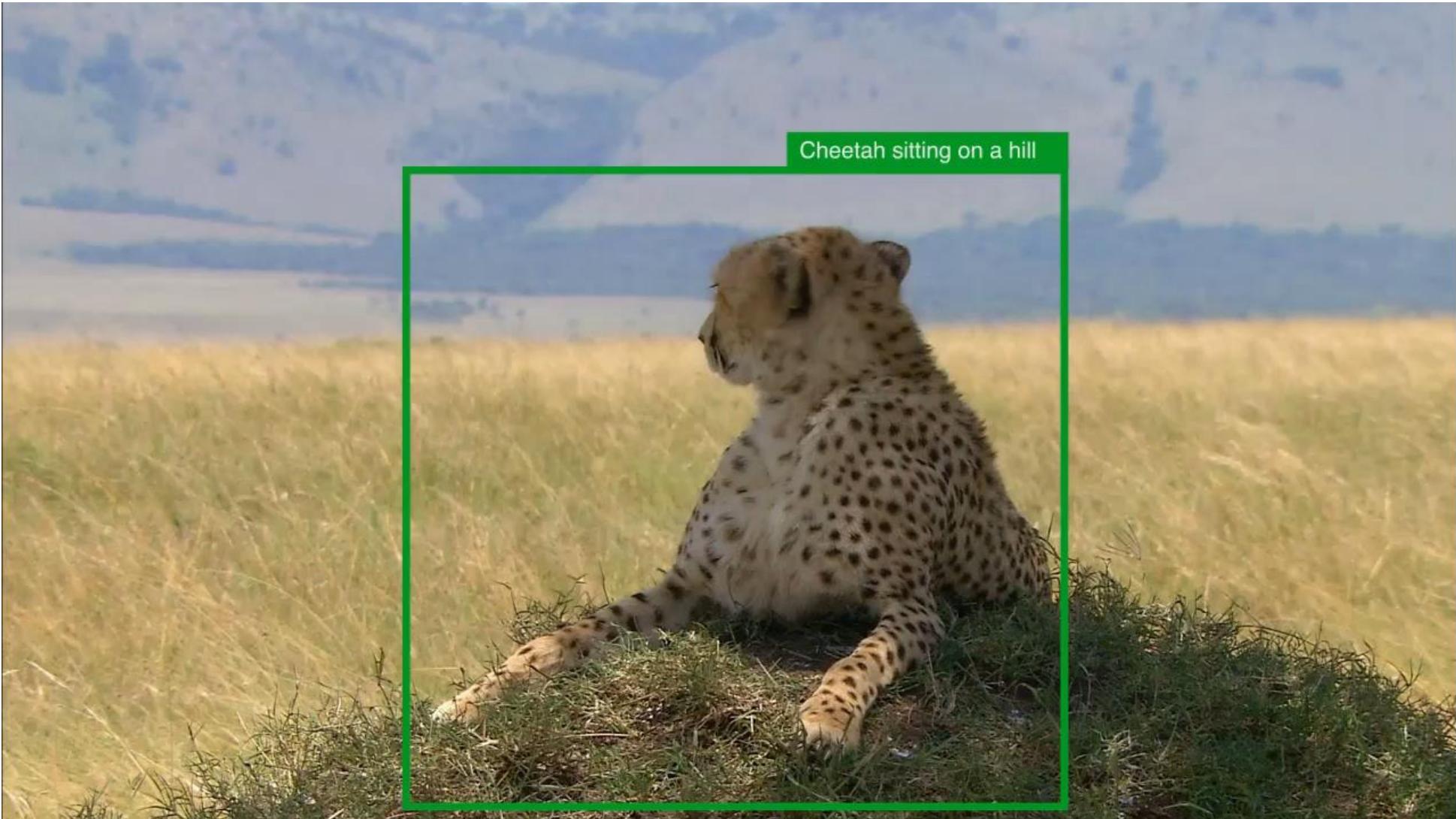


Model customization: Lower costs and time to deliver custom models that match unique business demands at high precision, and with just a handful of images.



Video summarization: Search and interact with video content in the same intuitive way you think and write. Locate relevant content without the need for additional metadata.

Azure Computer Vision 4.0



Azure Vision Studio



Video summary and frame locator

Preview

Generate a brief summary of the main points shown in video. Locate specific keywords and jump to the relevant section.

Try it out



Extract text from images

Extract printed and handwritten style text from images and documents for supported languages.

Try it out



Search photos with natural language

Preview

Retrieve specific moments within your photo album. For example, you can search for: a wedding you attended last summer, your pet, or your favorite city.

Try it out



Add captions to images

Preview

Generate a human-readable sentence that describes the content of an image.

Try it out



Dense captioning

Preview

Generate human-readable captions for all important objects detected in your image

Try it out



Detect common objects in images

Preview

Recognize the location of objects of interest in an image and assign them a label.

Try it out



Extract common tags from images

Preview

Use an AI model to automatically assign one or more labels to an image.

Try it out



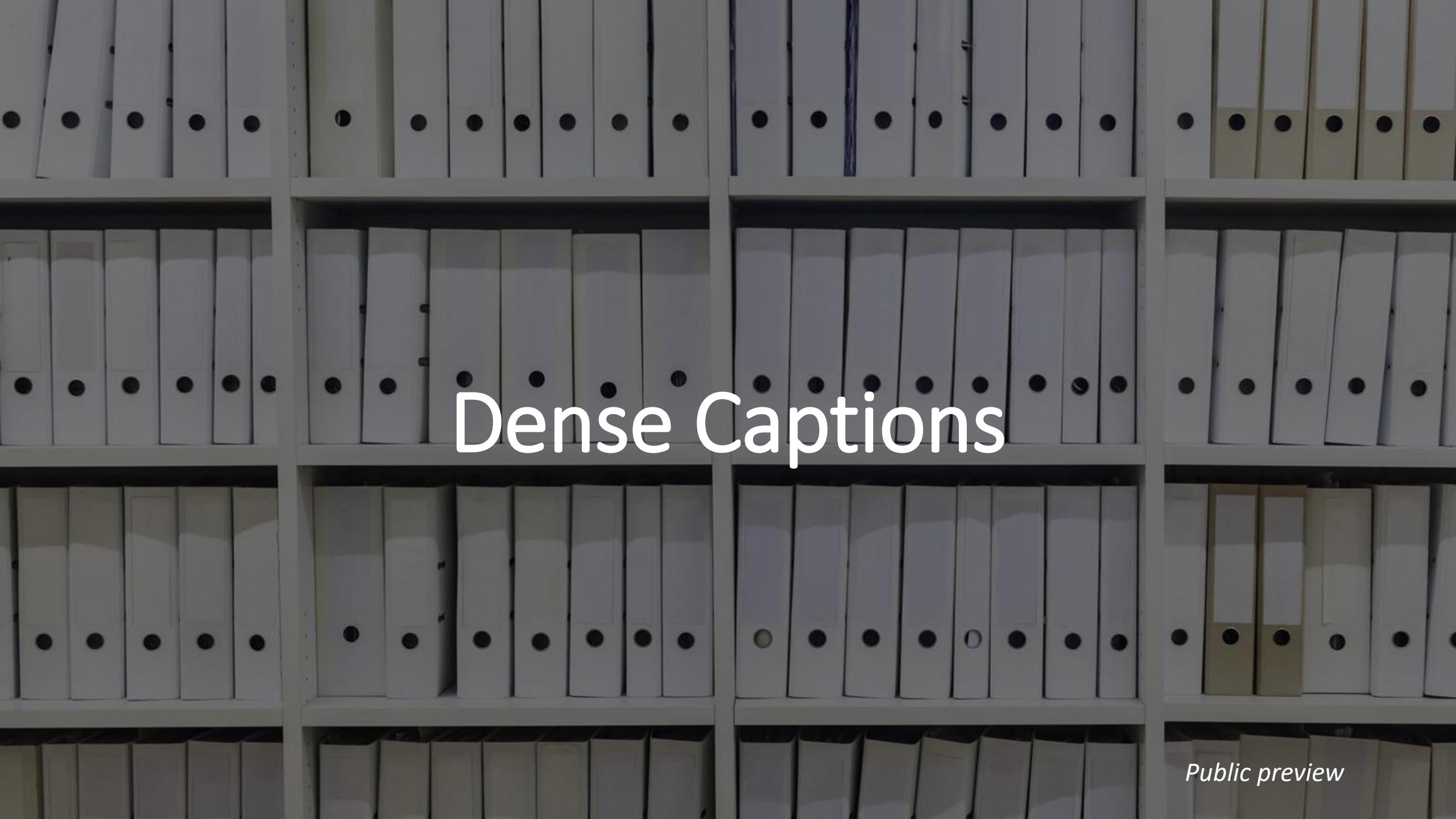
Detect sensitive content in images

Detect sensitive content in images so you can moderate their usage in your applications.

Try it out

New features or upgrades in red
(public preview)

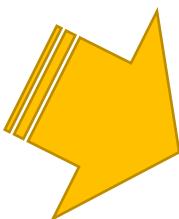
<https://aka.ms/VisionStudio>



Dense Captions

Public preview

Image Analysis V4.0 Dense Captioning



```
print("\033[1;31;34mDense captions:\n")  
  
for idx, value in enumerate(results['denseCaptionsResult']['values'], start=1):  
    print(idx, value['text'], "=", round(value['confidence'], 3))  
  
Dense captions:  
  
1 a fire truck and a car on fire = 0.494  
2 a fire truck on the road = 0.503  
3 a car on fire with smoke coming out of the back = 0.525  
4 a sign with an arrow pointing to the exit = 0.462  
5 a green sign with a white arrow = 0.531  
6 a car parked on the side of a road = 0.495  
7 a person wearing a firefighter uniform = 0.374  
8 a fire truck on the road = 0.517  
9 firemen on the road with a fire truck = 0.309  
10 a pole with a light on it = 0.37
```

Image Analysis V4.0 Dense Captioning



Image Analysis V4.0 Dense Captioning

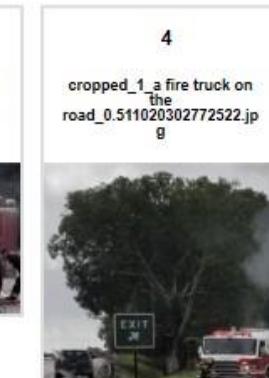
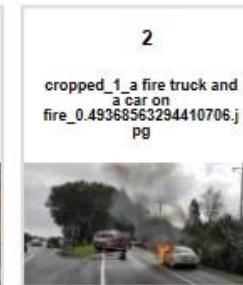
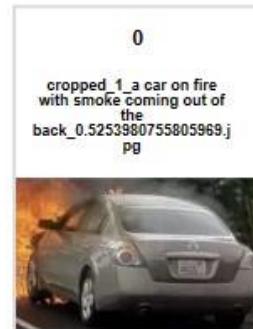
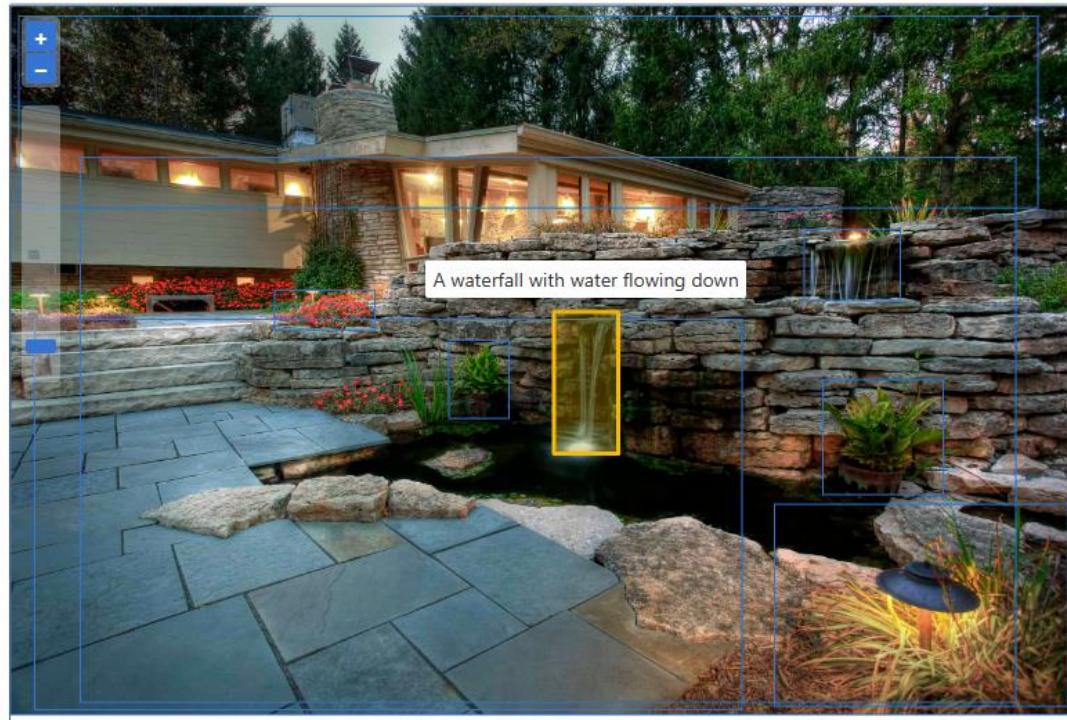


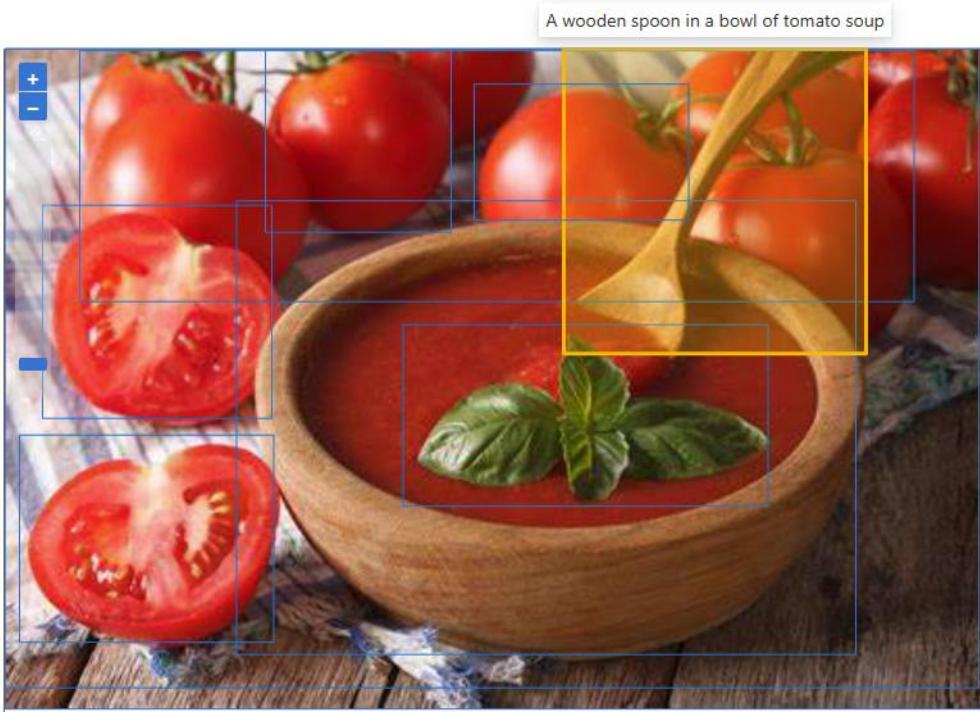
Image Analysis V4.0 Dense Captioning



Detected attributes [JSON](#)

- A house with a waterfall and a pond
- A water fountain in a stone wall
- A potted plant in front of a rock wall
- A building with a roof and trees
- A stone patio with a pond and a waterfall
- A waterfall with water flowing down**
- A close-up of a waterfall
- A close up of red flowers
- A light in the grass
- A plant in a pot

Image Analysis V4.0 Dense Captioning



A wooden spoon in a bowl of tomato soup

Detected attributes

JSON

- A bowl of tomato soup with a spoon
- A bowl of tomato soup with a spoon
- A bowl of tomato soup with a spoon
- A close up of a tomato
- A close up of a tomato
- A close up of a leaf
- A close up of a tomato
- A bowl of tomato soup next to tomatoes
- A wooden spoon in a bowl of tomato soup**
- A close up of a tomato

Image Analysis V4.0 Dense Captioning



[Detected attributes](#) [JSON](#)

- A group of people around a barbecue
- A man sitting on a chair playing a guitar
- A man holding a glass of beer**
- A man in a grey shirt
- A person standing in front of a grill
- A barbecue grill with food on it
- A woman wearing shorts and a flannel shirt
- A man sitting in a chair next to a tent
- A person sitting on a chair
- A close up of food

Image Analysis V4.0 Dense Captioning



[Detected attributes](#) [JSON](#)

- A car on fire in a street with black smoke coming out of it
- A white car with a red light on
- A car on the road
- A building on fire with black smoke
- A tree next to a building
- A car on fire with smoke**
- A car parked on the side of the road
- A window with bars
- A group of cars on fire
- A street light on a building

Image Analysis V4.0 Dense Captioning



[Detected attributes](#) [JSON](#)

- A car crash on the road
- A white van on the road
- A car with a damaged front end**
- A person in a yellow and black jacket
- A blue cone with white stripe
- A blue and white cone
- A white van with a light on the back
- A road with a car parked on it
- A man sweeping the street
- A man in black suit

Demo

<https://github.com/retkowsky/Image-Analysis-with-Azure-Computer-Vision-4.0-Captioning-and-Dense-Captioning>



DEMO

Demo video

https://www.youtube.com/playlist?list=PLy4MOYaxz3vMAA_le5wRUNiCZ9SiF74ex

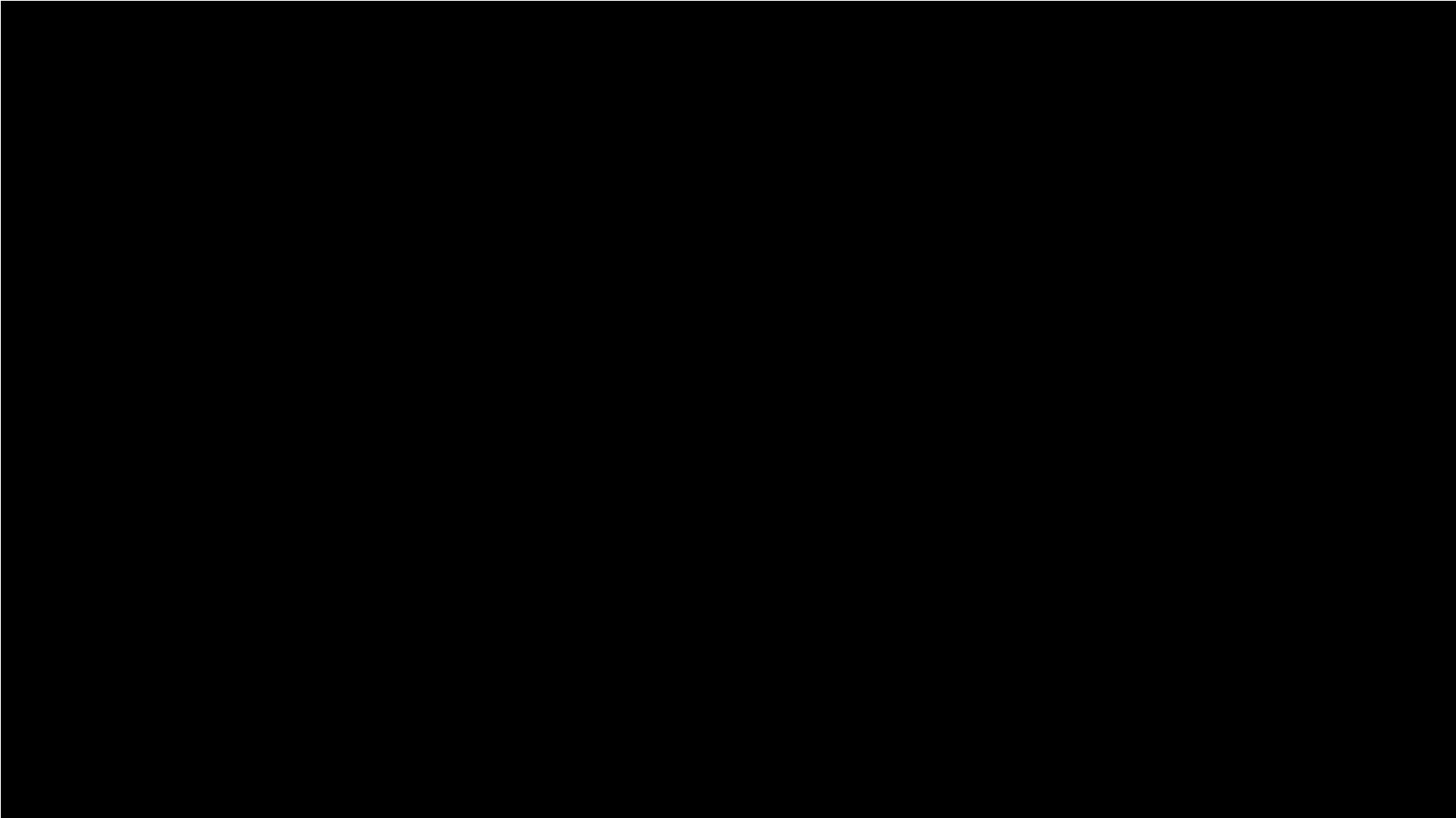




Image Retrieval

Public preview

Image Retrieval

1. Vectorize Images and Text: the Image Retrieval APIs, **VectorizeImage** and **VectorizeText**, can be used to extract feature vectors out of an image or text respectively. The APIs return a single feature vector representing the entire input.
2. Measure similarity: Vector search systems typically use distance metrics, such as cosine distance or Euclidean distance, to compare vectors and rank them by similarity. The [Vision studio](#) demo uses [cosine distance](#) to measure similarity.
3. Retrieve Images: Use the top N vectors similar to the search query and retrieve images corresponding to those vectors from your photo library to provide as the final result.

Image Retrieval

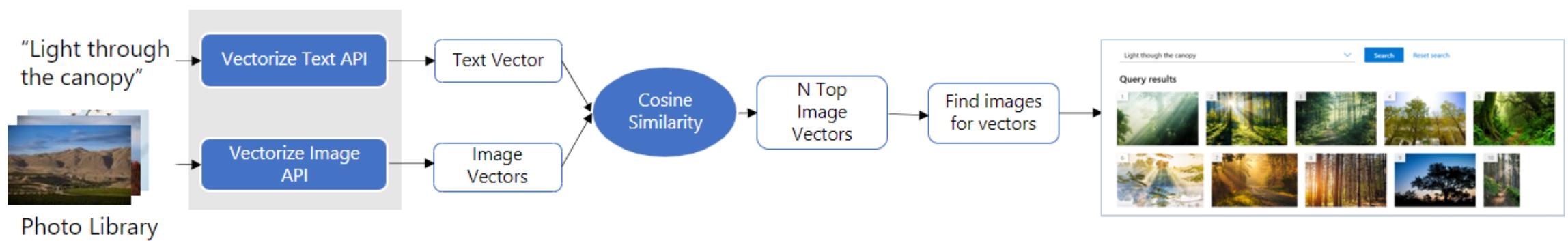


Image Retrieval – Text & image



```
In [18]: prompts = [  
    'PS5', 'Xbox', 'play station', 'Sony', 'controller', 'Microsoft',  
    'games console', 'guitar', 'fish', 'apple', 'car', 'street', 'truck',  
    'Miami', 'black controller', 'white controller'  
]
```

```
In [19]: df = similarity_results(image_emb2, prompts)  
  
cm = sns.light_palette("green", as_cmap=True)  
df.style.background_gradient(cmap=cm)
```

Out[19]:

	prompt	similarity
0	Xbox	0.312210
1	games console	0.310517
2	PS5	0.296529
3	white controller	0.275109
4	Microsoft	0.272174
5	controller	0.267969
6	play station	0.263670
7	black controller	0.242213
8	Sony	0.241703
9	truck	0.133365
10	guitar	0.120511
11	car	0.116488
12	fish	0.115516
13	apple	0.113572
14	Miami	0.112749
15	street	0.097342

Image Retrieval – Text & images



```
In [15]: prompts = [  
    'bird', 'a truck', 'a car', 'a blue car', 'a white car', 'a BMW white car',  
    'a tesla car', 'a mercedes car', 'a man', 'a ford car'  
]
```

```
In [16]: df = similarity_results(image_emb1, prompts)  
  
cm = sns.light_palette("green", as_cmap=True)  
df.style.background_gradient(cmap=cm)
```

Out[16]:

	prompt	similarity
0	a BMW white car	0.348192
1	a white car	0.311123
2	a blue car	0.272143
3	a car	0.267926
4	a tesla car	0.226553
5	a ford car	0.225621
6	a mercedes car	0.220861
7	a truck	0.159232
8	a man	0.151596
9	bird	0.138678

Image Retrieval – Visual search using images

Reference image



Comparison image 1



```
get_cosine_similarity(image_emb4, whitebmw)  
0.945438232657761
```

Comparison image 2



```
get_cosine_similarity(image_emb4, cat)  
0.3304837206034262
```

The most similar image of the reference image is *comparison image 1* with a similarity index = 0.94

Image Retrieval

Cognitive Services | Vision Studio

Vision Studio > Search photos with natural language

Sample image sets Try with your own images

Nature No.of photos: 260

Manufacturing No.of photos: 245

Education No.of photos: 264

Retail No.of photos: 265

Select a retrieval query or create your own

Solar panels in the desert

Search Reset search

Query results

Query results vary from most relevant in the dataset to least relevant. Utilize the slider below to view more or less images based on their relevance to the retrieval query.

Most relevant Least relevant

1 2 3 4 5 6 7

8 9 10 11 12 13 14

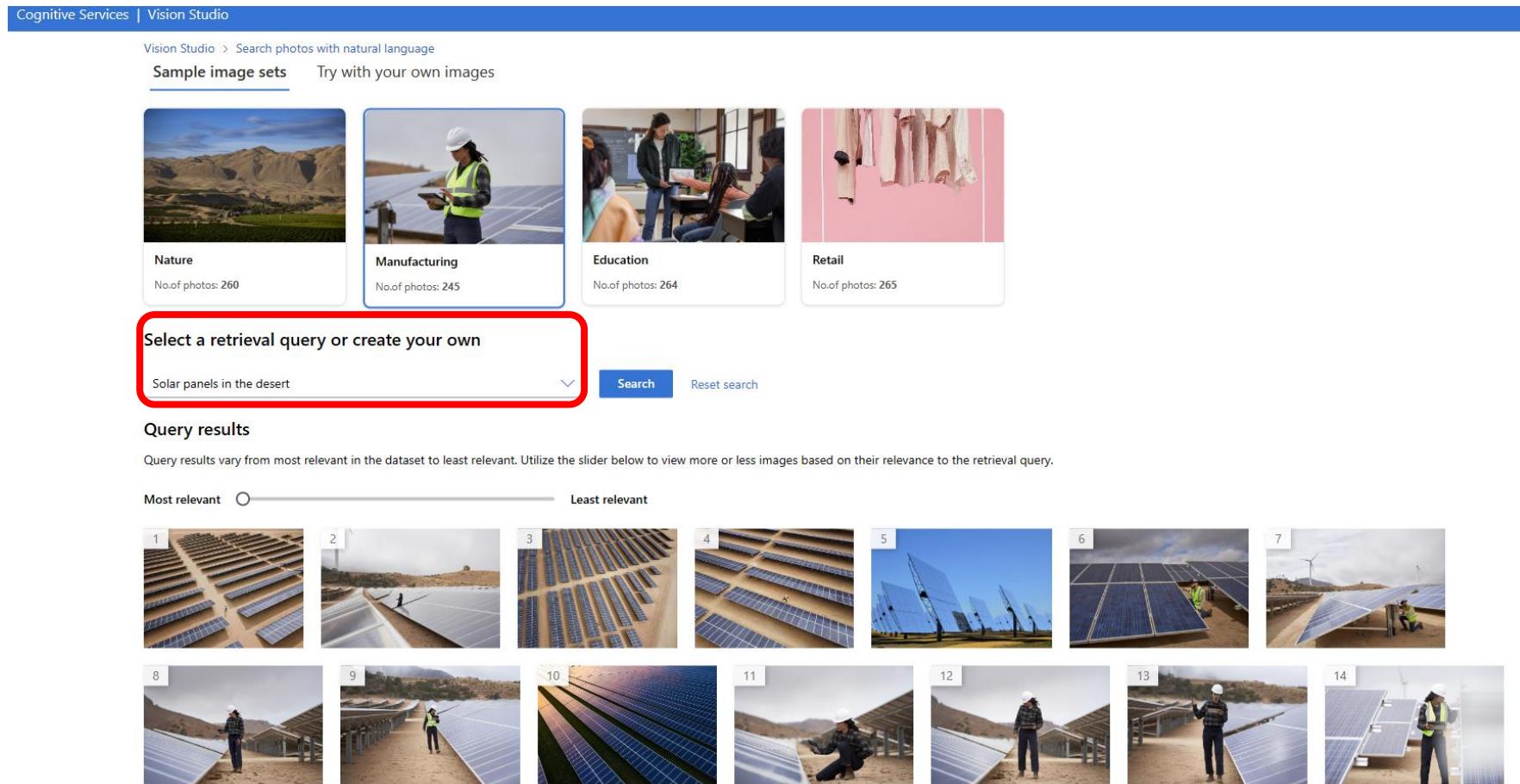


Image Retrieval

Vision Studio > Search photos with natural language

[Sample image sets](#)

Try with your own images



Nature

No.of photos: 260



Manufacturing

No.of photos: 245



Education

No.of photos: 264



Retail

No.of photos: 265

Select a retrieval query or create your own

Boxes on a conveyor belt

Search

Reset search

Query results

Query results vary from most relevant in the dataset to least relevant. Utilize the slider below to view more or less images based on their relevance to the retrieval query.

Most relevant

Least relevant



Note: The sample dataset is limited and may not have precise results based on your search query. Try uploading your own photos to try more complex queries.

Next steps

Use Search photos with natural language in your own application. Start with [sample code](#) and [read documentation](#).

Visual search with Image Retrieval using an image

```
reference_image = field_images[30]  
view_image(reference_image)
```

Image: images/field_images/field_image_00031.jpg



```
topn_list, simil_topn_list = get_topn_images(topn=5)
```

Top 5 images:

001 images/catalog_images/catalog_image_00334.jpg with similarity index = 0.9406693291226688
002 images/catalog_images/catalog_image_00412.jpg with similarity index = 0.9318492521466659
003 images/catalog_images/catalog_image_00086.jpg with similarity index = 0.9116624975370073
004 images/catalog_images/catalog_image_00085.jpg with similarity index = 0.8973578354650728
005 images/catalog_images/catalog_image_00084.jpg with similarity index = 0.8528888453249868

```
view_similar_images()
```

[show html](#)

Top 1 similarity = 0.94067

images/catalog_images/catalog_image_00334.jpg



Top 2 similarity = 0.93185

images/catalog_images/catalog_image_00412.jpg



Top 3 similarity = 0.91166

images/catalog_images/catalog_image_00086.jpg



Top 4 similarity = 0.89736

images/catalog_images/catalog_image_00085.jpg



Top 5 similarity = 0.85289

images/catalog_images/catalog_image_00084.jpg



Visual search with Image Retrieval using a prompt

```
text = "coca cola can"
```

```
df = get_similar_images_using_prompt(text)
```

Done in 0.511 secs

```
topn_list, simil_topn_list = get_topn_images(topn=5)
```

Top 5 images:

```
001 images/catalog_images/catalog_image_00462.jpg with similarity index = 0.3814190065405629  
002 images/catalog_images/catalog_image_00256.jpg with similarity index = 0.37658384707914094  
003 images/catalog_images/catalog_image_00250.jpg with similarity index = 0.37257945224846534  
004 images/catalog_images/catalog_image_00249.jpg with similarity index = 0.36913790014010545  
005 images/catalog_images/catalog_image_00460.jpg with similarity index = 0.36567493876892554
```

```
view_similar_images()
```

[show html](#)

Top 1 similarity = 0.38142

images/catalog_images/catalog_image_00462.jpg



Top 2 similarity = 0.37658

images/catalog_images/catalog_image_00256.jpg



Top 3 similarity = 0.37258

images/catalog_images/catalog_image_00250.jpg



Top 4 similarity = 0.36914

images/catalog_images/catalog_image_00249.jpg



Top 5 similarity = 0.36567

images/catalog_images/catalog_image_00460.jpg



Visual search with Image Retrieval (fashion examples)

```
In [97]: topn_list, simil_topn_list = get_topn_images(topn=topn)
view_similar_images_using_image()
```

Image to search:
download.jpg



Top 3: 0699923102.jpg
Similarity = 0.72736



Top 1: 0621522018.jpg
Similarity = 0.74739



Top 4: 0699923067.jpg
Similarity = 0.72531



Top 2: 0694131008.jpg
Similarity = 0.74719



Top 5: 0648703010.jpg
Similarity = 0.69851



Visual search with Image Retrieval (fashion examples)

Test 10

```
In [125]: query = "Italian cities"
```

```
In [126]: df = get_similar_images_using_prompt(query)
topn = 4
df.head(topn).style.background_gradient(
    cmap=sns.light_palette("green", as_cmap=True))
```

```
Out[126]:
```

	image_file	similarity
9381	fashion/0698273004.jpg	0.241590
5135	fashion/0646592011.jpg	0.222154
4241	fashion/0642051001.jpg	0.220227
9439	fashion/0698328006.jpg	0.215080

```
In [127]: topn_list, simil_topn_list = get_topn_images(topn=topn)
view_similar_images_using_prompt()
```

Top 1: 0698273004.jpg
Similarity = 0.24159



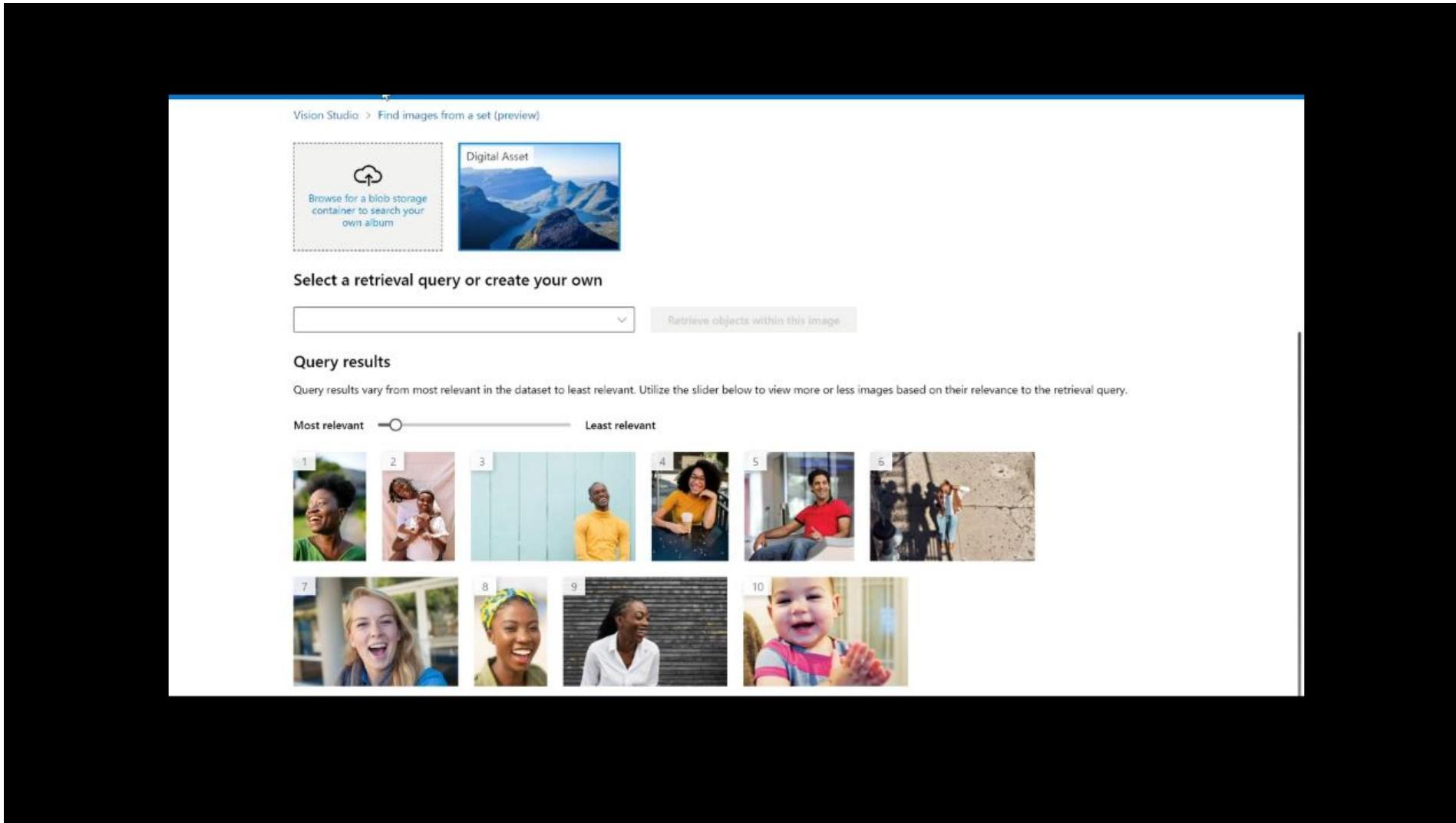
Top 2: 0646592011.jpg
Similarity = 0.22215



Top 3: 0642051001.jpg
Similarity = 0.22023



Find images from a set (Vision studio demo)



Find images from a set (Python SDK)

```
In [41]: field_images = [  
    file for file in glob.glob(IMAGES_DIR + '/field_images/*.*')  
    if file.endswith(('jpeg', 'png', 'jpg', 'JPG', 'JPEG', 'PNG'))]  
  
print("Total number of field images = ", len(image_files))
```

Total number of field images = 466

Test 1

```
In [42]: reference_image = field_images[10]  
  
view_image(reference_image)
```

Image: images/field_images/field_image_00011.jpg



```
In [43]: df = get_similar_images_using_image(reference_image)  
  
Done in 0.810 secs
```

```
In [44]: topn_list, simil_topn_list = get_topn_images(topn=5)
```

Demo

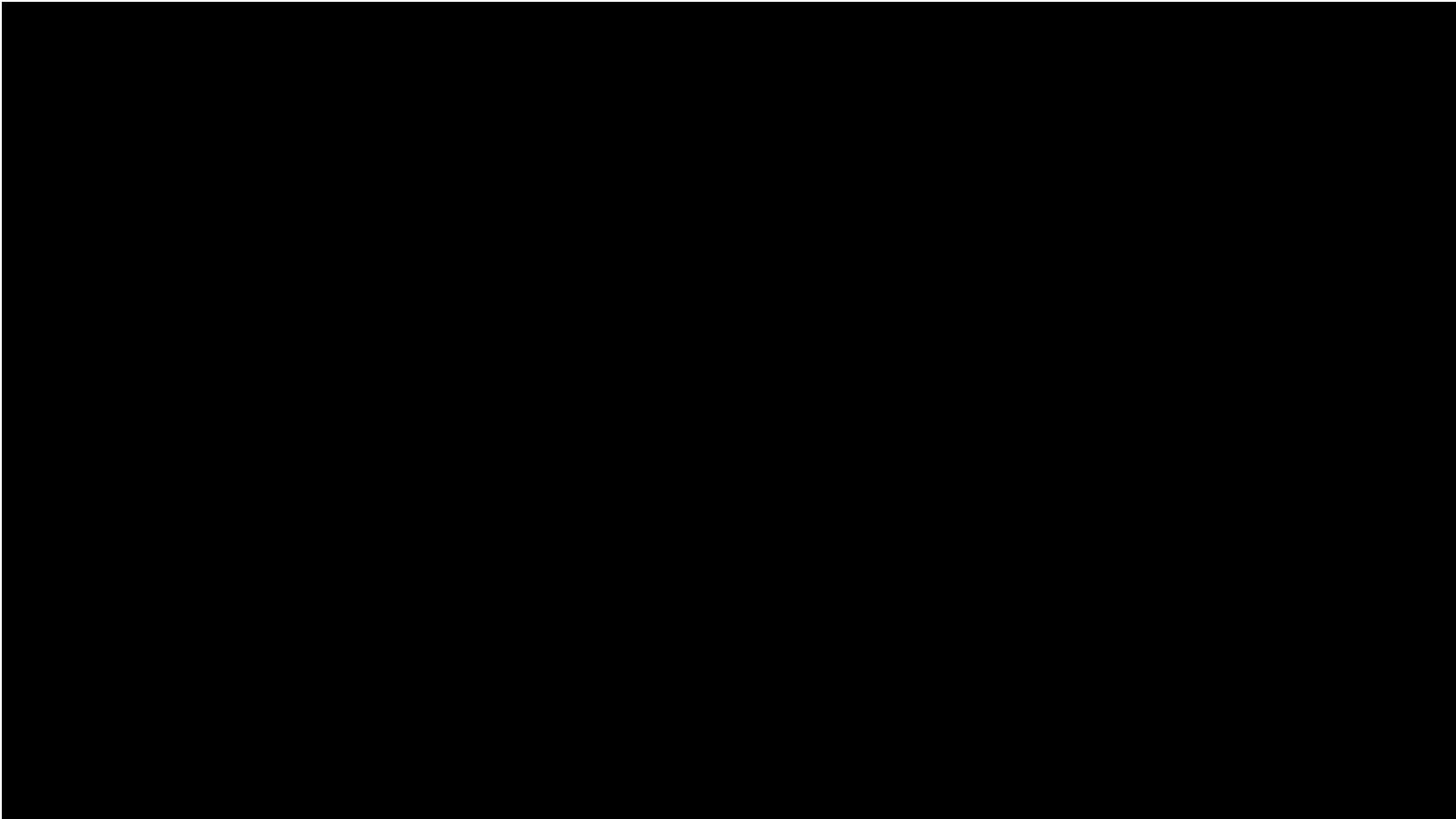
https://github.com/retkowsky/image_retrieval_with_Azure_Computer_Vision_4.0

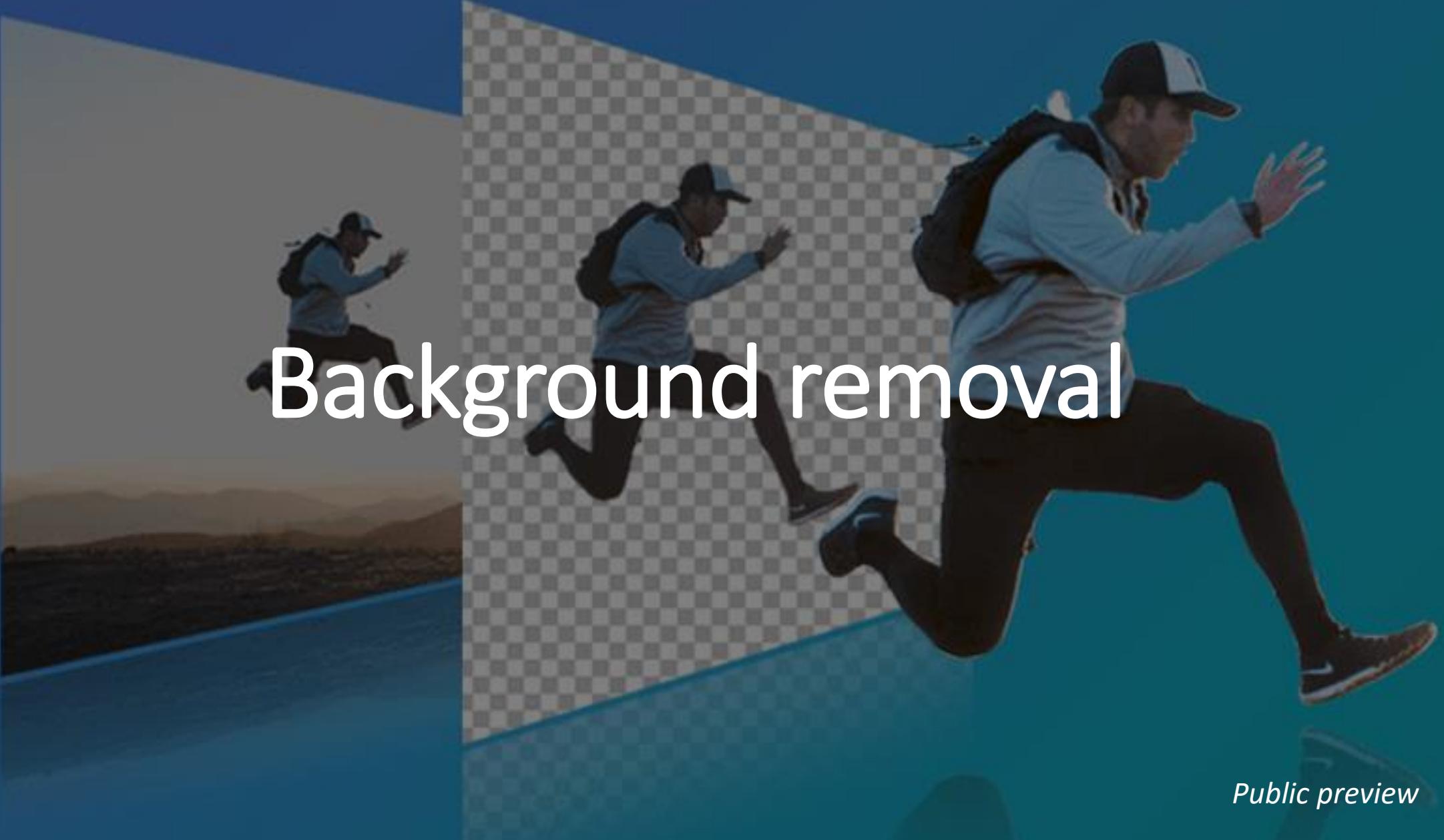


DEMO

Demo video

https://www.youtube.com/playlist?list=PLy4MOYaxz3vMAA_le5wRUNiCZ9Si74ex

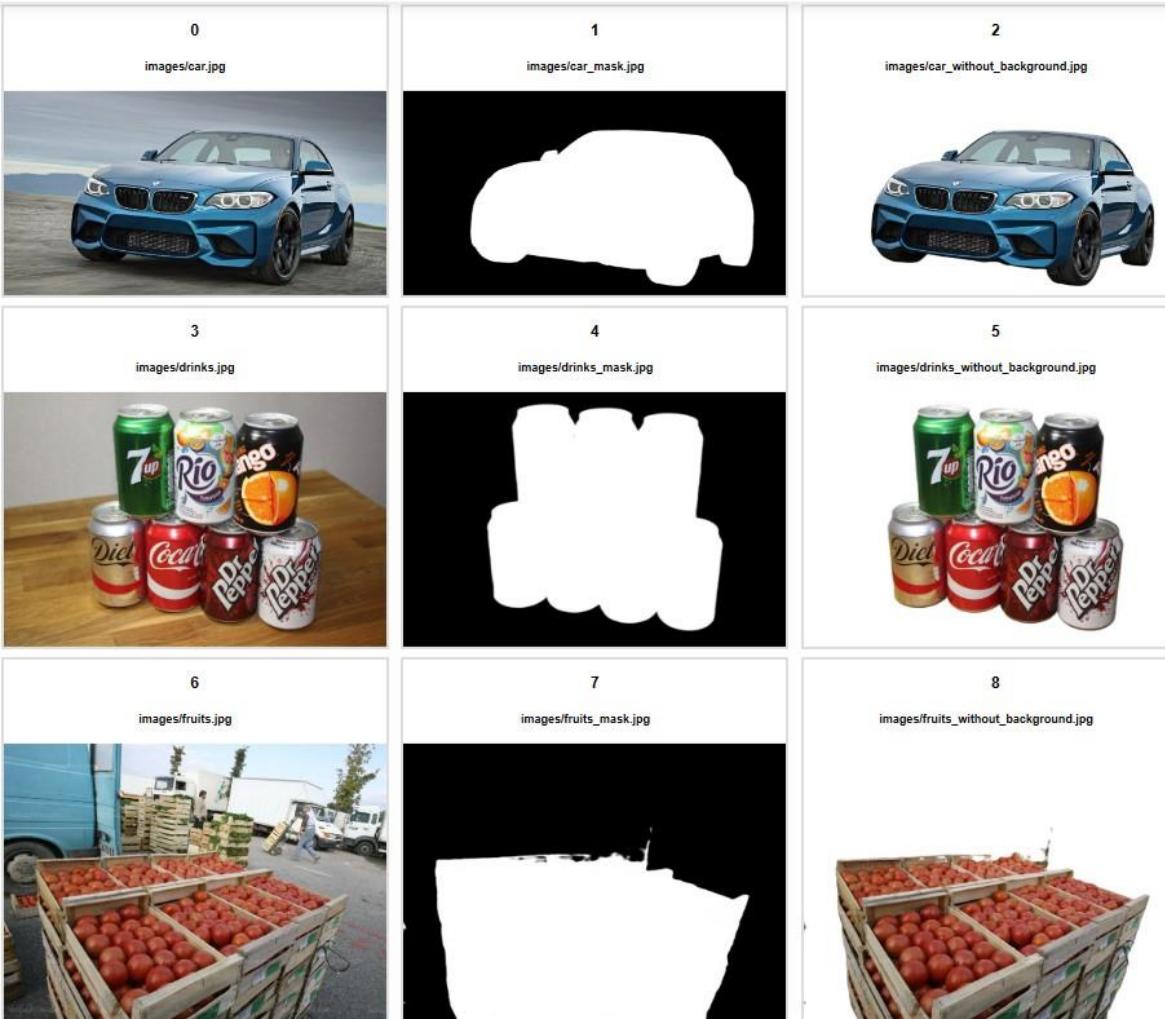




Background removal

Public preview

Background Removal



This option is perfect for removing non useful information from an image before processing it with a computer vision model for example

Background Removal

Background removal with Azure Computer Vision 4

Initial image



Without the background



Background Removal

Background removal with Azure Computer Vision 4

Initial image



Without the background



Demo

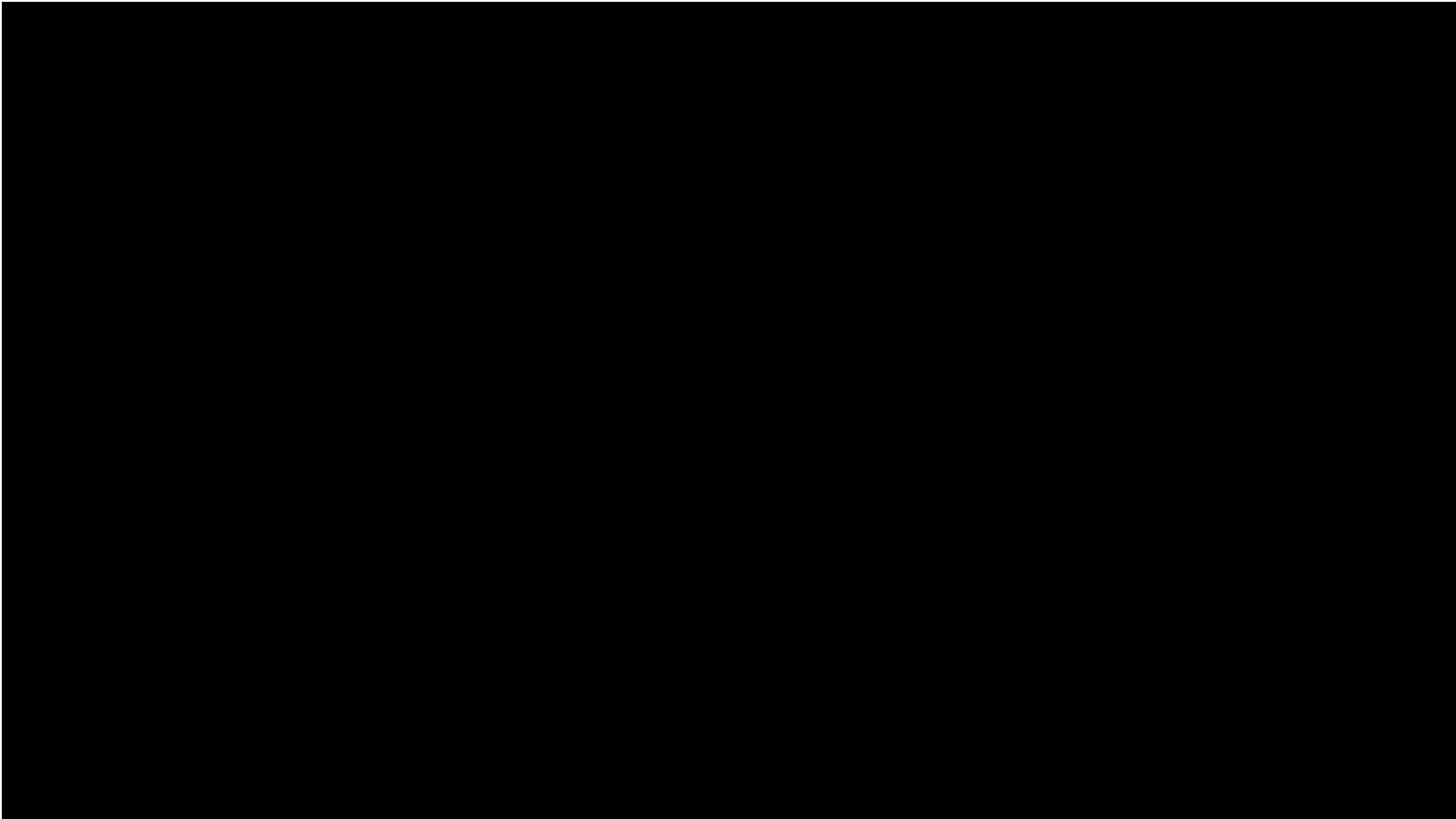
[https://github.com/retkowsky/Re
moving-background-using-Azure-
Computer-Vision-4](https://github.com/retkowsky/Removing-background-using-Azure-Computer-Vision-4)



DEMO

Demo video

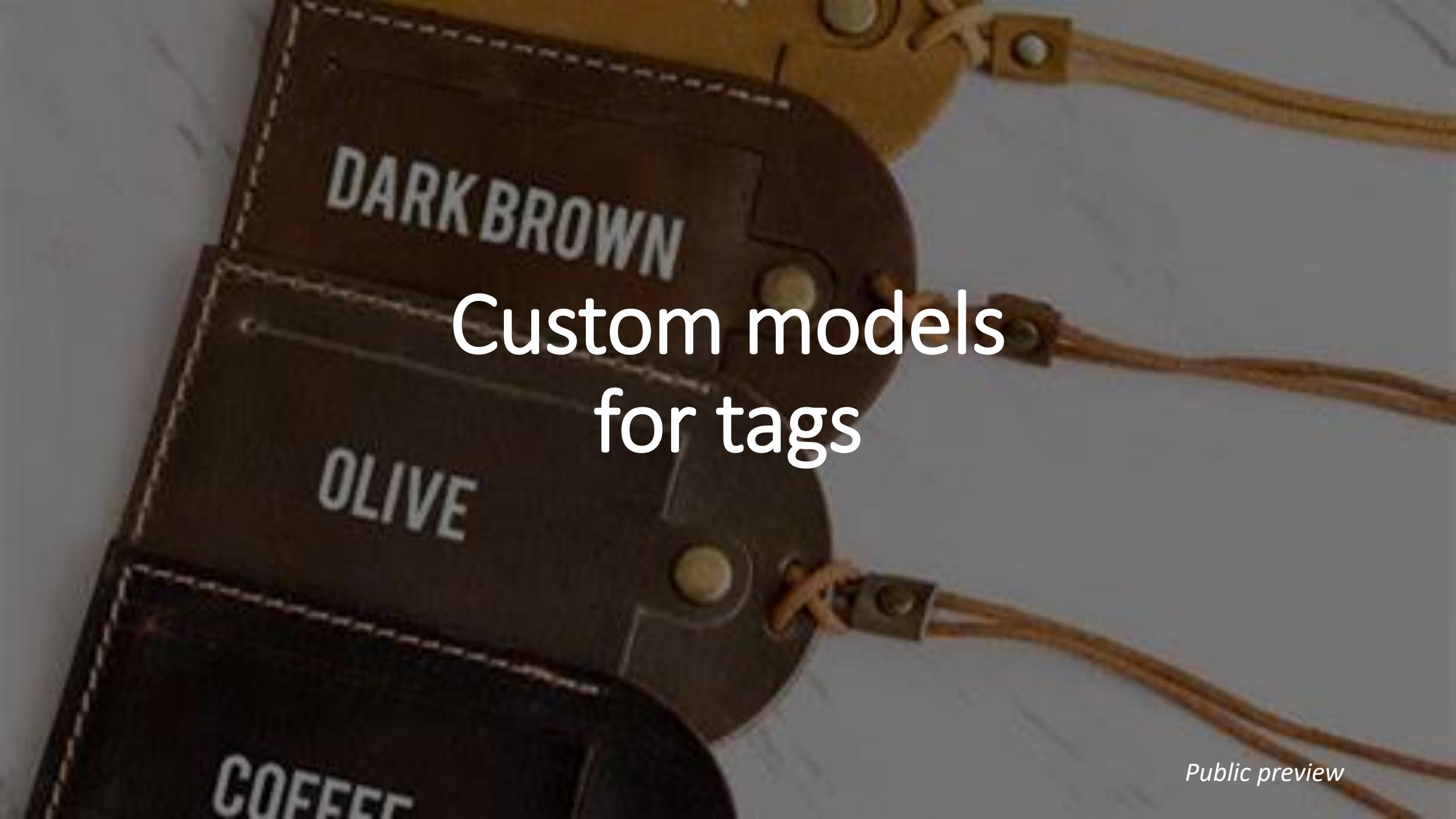
https://www.youtube.com/playlist?list=PLy4MOYaxz3vMAA_le5wRUNiCZ9Si74ex





Custom models for tags & objects

Public preview



Custom models
for tags

Public preview

Extract common tags – standard model

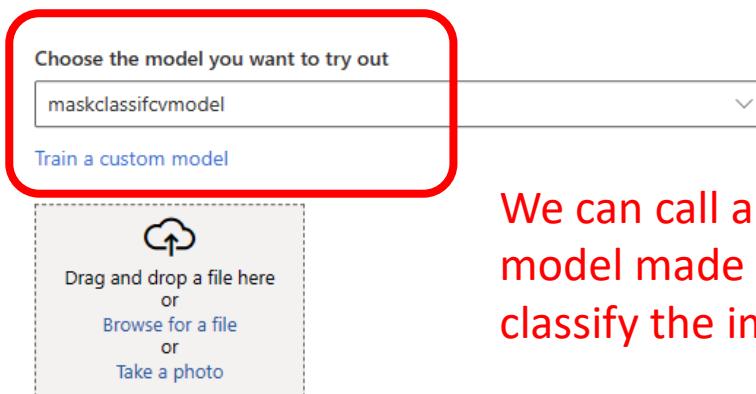


[Detected attributes](#) [JSON](#)

outdoor (99.77%)
clothing (99.73%)
grass (99.66%)
person (99.41%)
footwear (98.50%)
tree (98.43%)
people (95.85%)
picnic (92.11%)
plant (90.72%)
man (89.53%)
group (89.09%)
woman (66.37%)
standing (65.23%)
barbecue (61.63%)
food (59.39%)
summer (52.06%)

Tags results using the standard model from Azure Computer Vision 4

Extract common tags – custom model



We can call a custom image classification model made with Azure Computer Vision to classify the image using some customs tags

Extract common tags – custom model

Cognitive Services | Vision Studio

Vision Studio > My resources > azurecv > Custom models

Custom models that are a part of this resource

+ Train a new model Refresh Delete

Name ↑

- maskclassifcvmodel
- maskod

Train a new model

Select a name and a target scenario for your model

Customize the name of this model *

mymodel

Select the model type *

Please select

Image classification

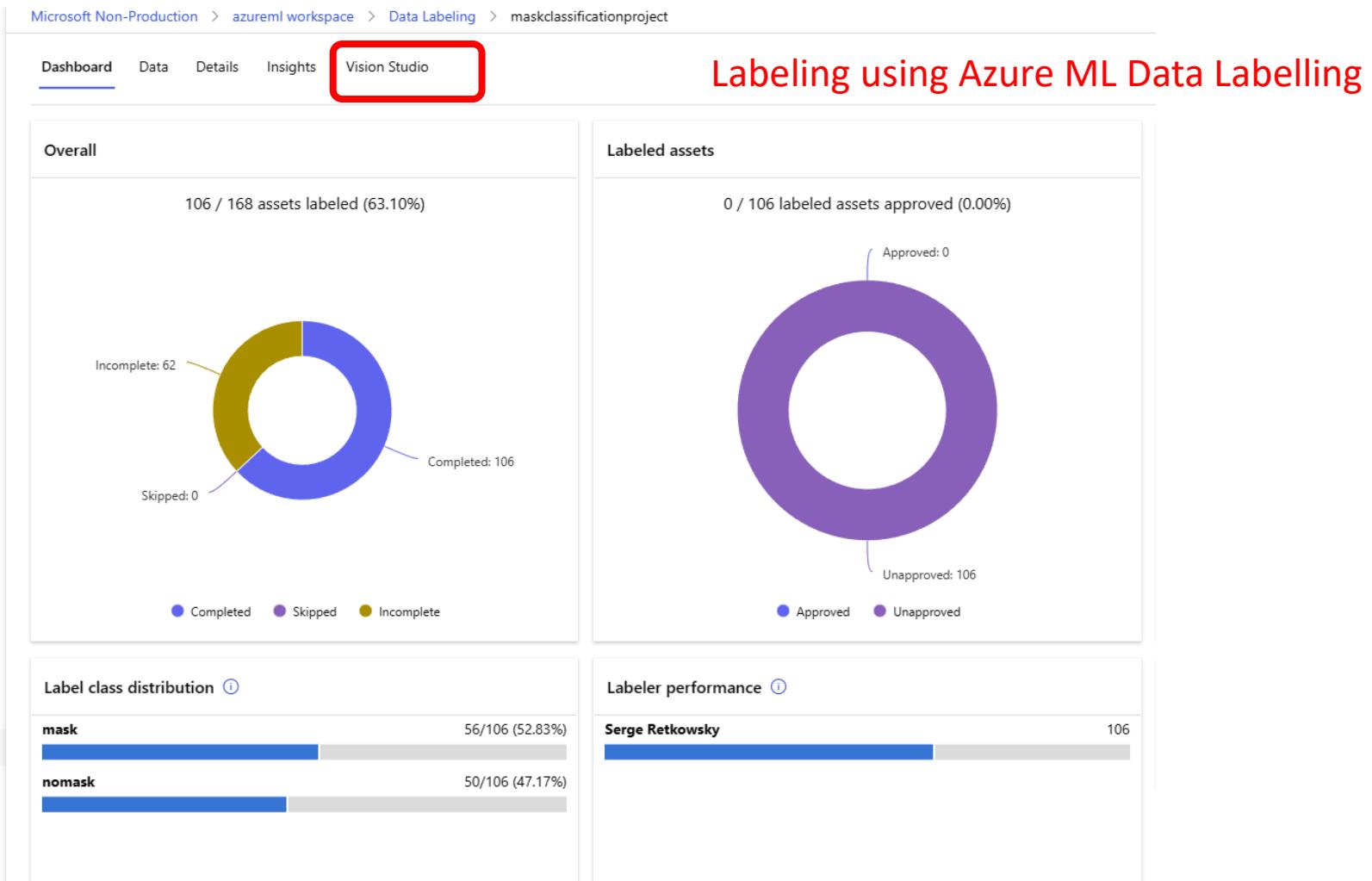
Object detection

Model creation user interface

Next Train model Cancel

The screenshot shows the Microsoft Cognitive Services Vision Studio interface. On the left, there's a sidebar with 'Vision Studio' and 'azurecv' sections, and a 'Custom models' section which is currently selected. In the center, there's a list of existing custom models: 'maskclassifcvmodel' and 'maskod'. A modal window titled 'Train a new model' is open. It asks for a 'Name and scenario' (with 'mymodel' entered) and a 'Select the model type' (with 'Image classification' selected). A red box highlights the text 'Model creation user interface'.

Extract common tags – custom model



Extract common tags – custom model

We can retrieve the labelled Coco file from Azure ML

Cognitive Services | Vision Studio

Vision Studio > My resources > azurecv > Datasets > masksdataset

masksdataset

Refresh Delete

Last modified on 08/03/2023 10:39:22
Dataset type Multi-class image classification (single tag per image)
Blob storage container masks

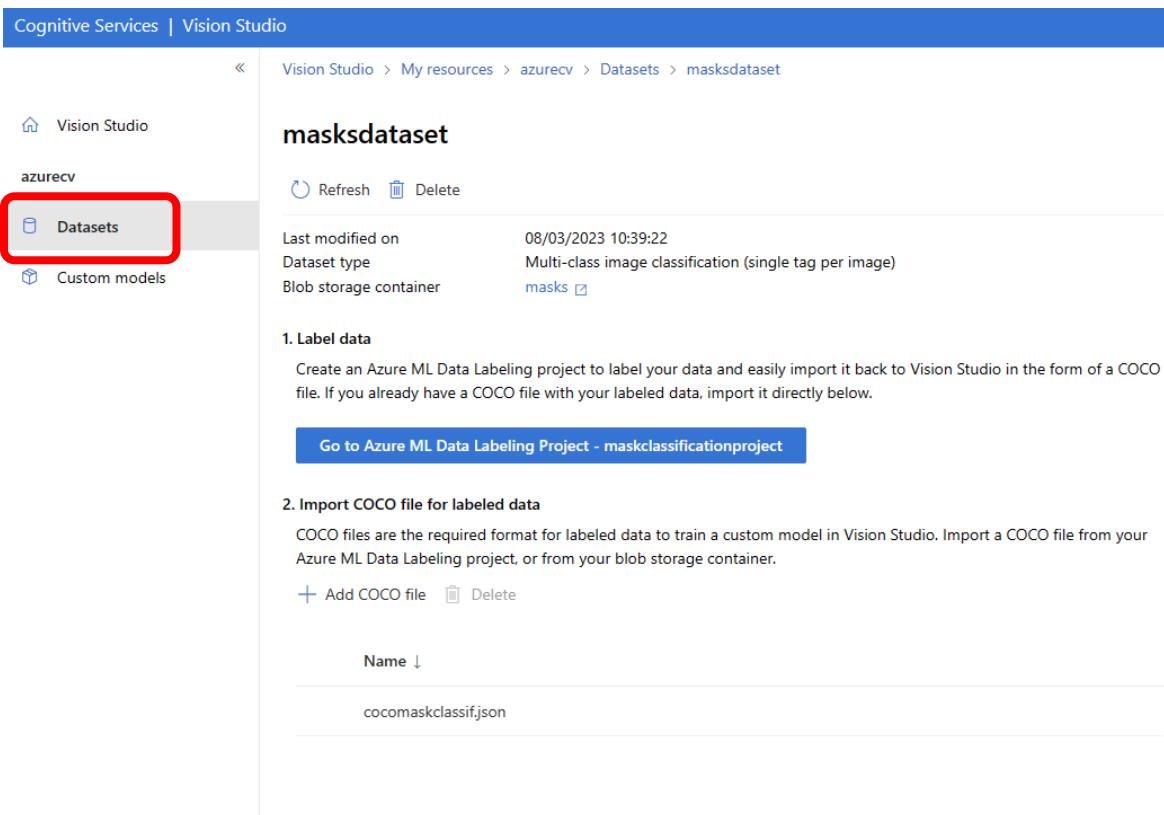
1. Label data
Create an Azure ML Data Labeling project to label your data and easily import it back to Vision Studio in the form of a COCO file. If you already have a COCO file with your labeled data, import it directly below.

[Go to Azure ML Data Labeling Project - maskclassificationproject](#)

2. Import COCO file for labeled data
COCO files are the required format for labeled data to train a custom model in Vision Studio. Import a COCO file from your Azure ML Data Labeling project, or from your blob storage container.

+ Add COCO file Delete

Name ↓
cocomaskclassif.json



Cognitive Services | Vision Studio

Vision Studio > My resources > azurecv > Custom models > maskclassifcvmodel

maskclassifcvmodel

Refresh Cancel Delete

Target scenario Image classification
Finished training on 08/03/2023 12:13:59
Total training time 1 hour(s)
Dataset(s) used to train the model masksdataset
Status Succeeded
Model task kind Multi-Class Classification

Evaluation runs
+ Add new evaluation run Delete evaluation run

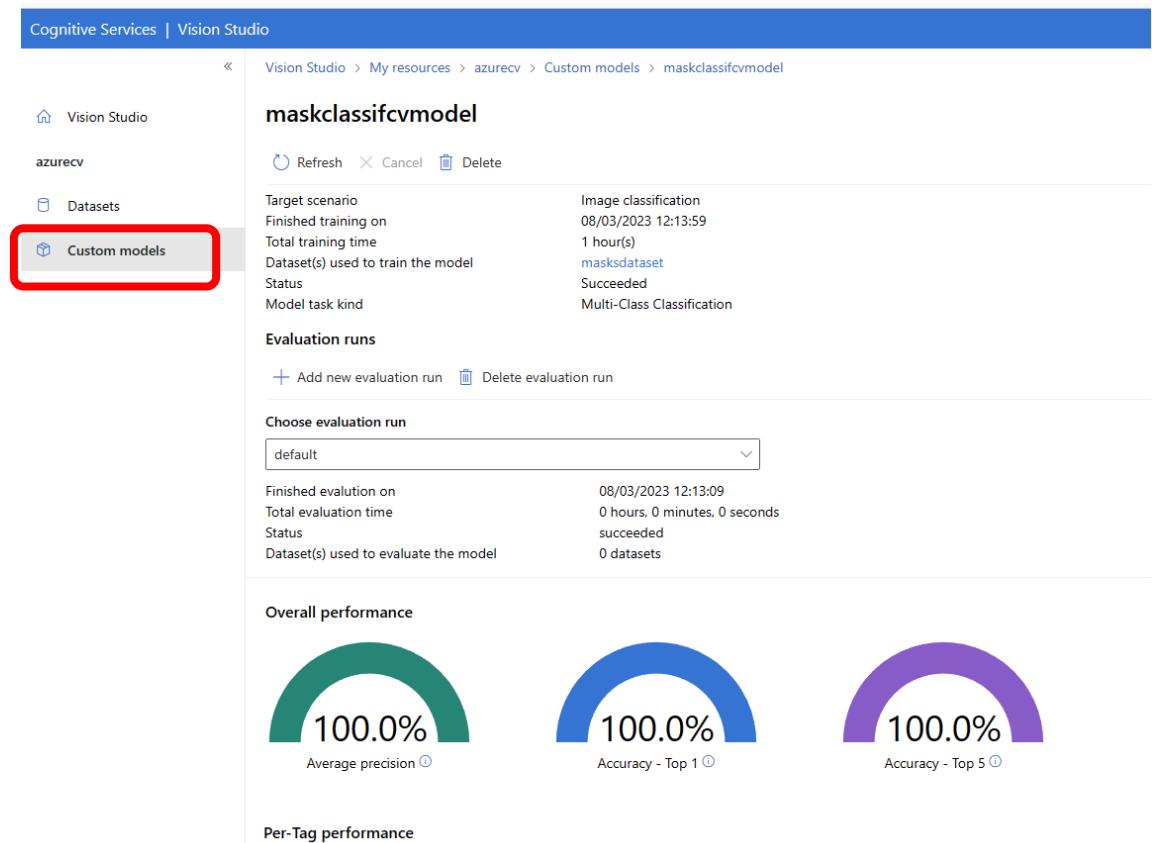
Choose evaluation run default

Finished evalution on 08/03/2023 12:13:09
Total evaluation time 0 hours, 0 minutes, 0 seconds
Status succeeded
Dataset(s) used to evaluate the model 0 datasets

Overall performance

Average precision 100.0% Accuracy - Top 1 100.0% Accuracy - Top 5 100.0%

Per-Tag performance



We can train the model and see the results using Azure Computer Vision

Extract common tags – custom model

Choose the model you want to try out

maskclassifcvmodel

Train a custom model

Drag and drop a file here
or
Browse for a file
or
Take a photo

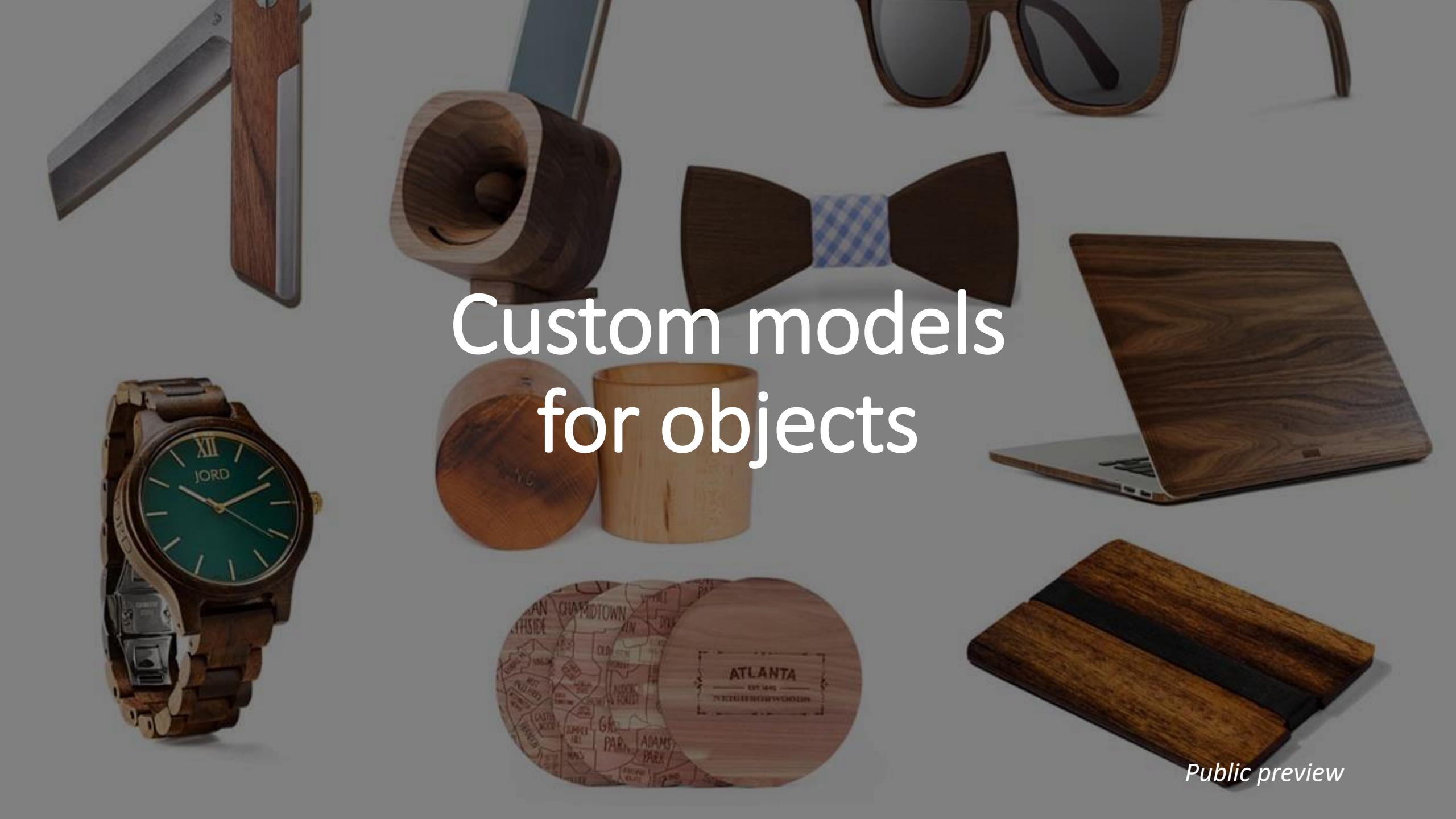
We can select our custom model



Detected attributes JSON

mask (91.31%)
nomask (8.69%)

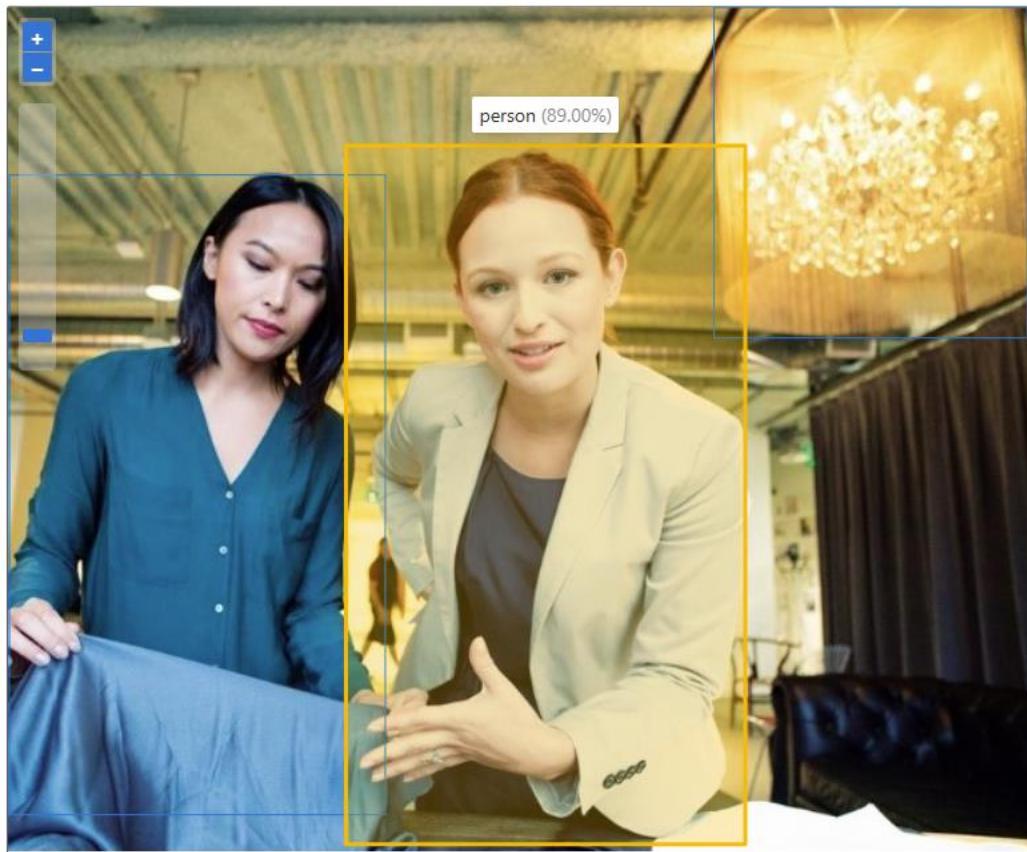
Results of our
custom image
classification model



Custom models
for objects

Public preview

Detect common objects in images – standard model



Detected attributes JSON

Threshold value ⓘ 15

- Lantern (52.30%)
- person (84.90%)
- person (89.00%)

Detect common objects in images – custom model

Cognitive Services | Vision Studio

Vision Studio > My resources > azurecv > Custom models

[Vision Studio](#) [azurecv](#) [Datasets](#) [Custom models](#)

[+ Train a new model](#) [Refresh](#) [Delete](#)

Custom models that are a part of this resource

Name ↑	Last updated	Model type	Status
maskclassifcvmodel	8 hours ago	Multi-Class Classification	Succeeded
maskobjectdetectionmodel	1 hour ago	Object Detection	Succeeded

Detect common objects in images – custom model

Cognitive Services | Vision Studio

Vision Studio > My resources > azurecv > Custom models > maskobjectdetectionmodel

maskobjectdetectionmodel

Refresh Cancel Delete

Target scenario: Object detection
Finished training on: 08/03/2023 19:12:11
Total training time: 4 hour(s)
Dataset(s) used to train the model: maskoddataset
Status: Succeeded
Model task kind: Object Detection

Evaluation runs

Add new evaluation run Delete evaluation run

Choose evaluation run

default

Finished evalution on: 08/03/2023 19:12:05
Total evaluation time: 0 hours, 0 minutes, 0 seconds
Status: succeeded
Dataset(s) used to evaluate the model: 0 datasets

Overall performance

93.9% Mean Average Precision @ 30 ⓘ

86.9% Mean Average Precision @ 50 ⓘ

28.3% Mean Average Precision @ 75 ⓘ

Detect common objects in images – custom model

Cognitive Services | Vision Studio

Vision Studio > Detect common objects in images

Detect common objects in images PREVIEW

Detect and extract bounding boxes based on thousands of recognizable objects and living beings.

[View documentation](#) [View SDK reference](#) [Use the REST API](#) [View samples on Github](#)

Try it out

I acknowledge that this demo will incur usage to resource `azurecv` in my Azure account. [Choose a different resource](#).

ⓘ When running prediction with custom models, you might experience longer than expected latency to receive prediction results. It is not currently

Use one of your own files or choose from a sample below.

Choose the model you want to try out

maskobjectdetectionmodel

Pretrained Vision model

maskobjectdetectionmodel

Drag and drop a file here
or
[Browse for a file](#)
or
[Take a photo](#)

maskobjectdetectionmodel

Detect common objects in images – custom model



Detected attributes JSON

Threshold value 50

nomask (82.96%)
nomask (81.30%)

Detected attributes JSON

Threshold value 50

nomask (82.96%)
nomask (81.30%)

The interface shows the threshold value set to 50. The "nomask" detection for the woman on the right has moved from the first list to the second list, indicating a change in the model's confidence or a re-detection.

Detect common objects in images – custom model



Detected attributes

Threshold value ⓘ

mask (87.16%)
mask (86.13%)
mask (85.89%)
mask (85.11%)
mask (84.96%)
mask (83.98%)
mask (83.30%)
mask (83.25%)
mask (80.03%)
mask (79.83%)
mask (78.47%)
mask (77.29%)
mask (74.85%)
mask (71.58%)
mask (67.04%)
mask (64.55%)



Detected attributes

Threshold value ⓘ

mask (93.51%)
nomask (87.60%)

Custom vision models

Category	Generic image classifier	Generic object detector
Max # training hours	288 (12 days)	288 (12 days)
Max # training images	1,000,000	200,000
Max # evaluation images	1,00,000	100,000
Min # training images per category	2	2
Max # tags per image	multiclass: 1	NA
Max # regions per image	NA	1,000
Max # categories	2,000	1,000
Min # categories	2	1
Max image size (Training)	20 MB	20 MB
Max image size (Prediction)	Sync: 6 MB, Batch: 20 MB	Sync: 6 MB, Batch: 20 MB
Max image width/height (Training)	10,240	10,240
Min image width/height (Prediction)	50	50
Available regions	West US 2, East US, West Europe	West US 2, East US, West Europe
Accepted image types	jpg, png, bmp, gif, jpeg	jpg, png, bmp, gif, jpeg



Video Summary

Public preview

Video summary and frame locator

Factory



0:07

Run a test

Summarize this video

Run

Results

Automatic text summarization

Summarization:

The recently released video showcases the daily operations of a factory or machine shop, focusing on worker safety and efficient use of equipment. Here are the noteworthy events that were captured in the video:

- One worker is seen running, suggesting an urgent matter that required immediate attention.
- Another worker is observed taking a picture, likely for documentation purposes.
- Unfortunately, the video also shows a worker falling to the ground, which emphasizes the potential dangers of working in a factory environment.

Despite these incidents, the video showcases the workers operating various types of machinery, transporting equipment, and performing maintenance tasks. The factory also uses heavy machinery such as a forklift to move boxes and machines, highlighting the importance of specialized equipment in factory settings. The video provides a fascinating insight into the daily routine of workers in a factory and demonstrates the complexity and fast-paced nature of this work environment.

2:56

Video summary and frame locator

paris.mp4



Run a test

Locate specific frames using a search query

Search query

two motorbikes

Run

We can search on any object from the video like *two motorbikes*

Results

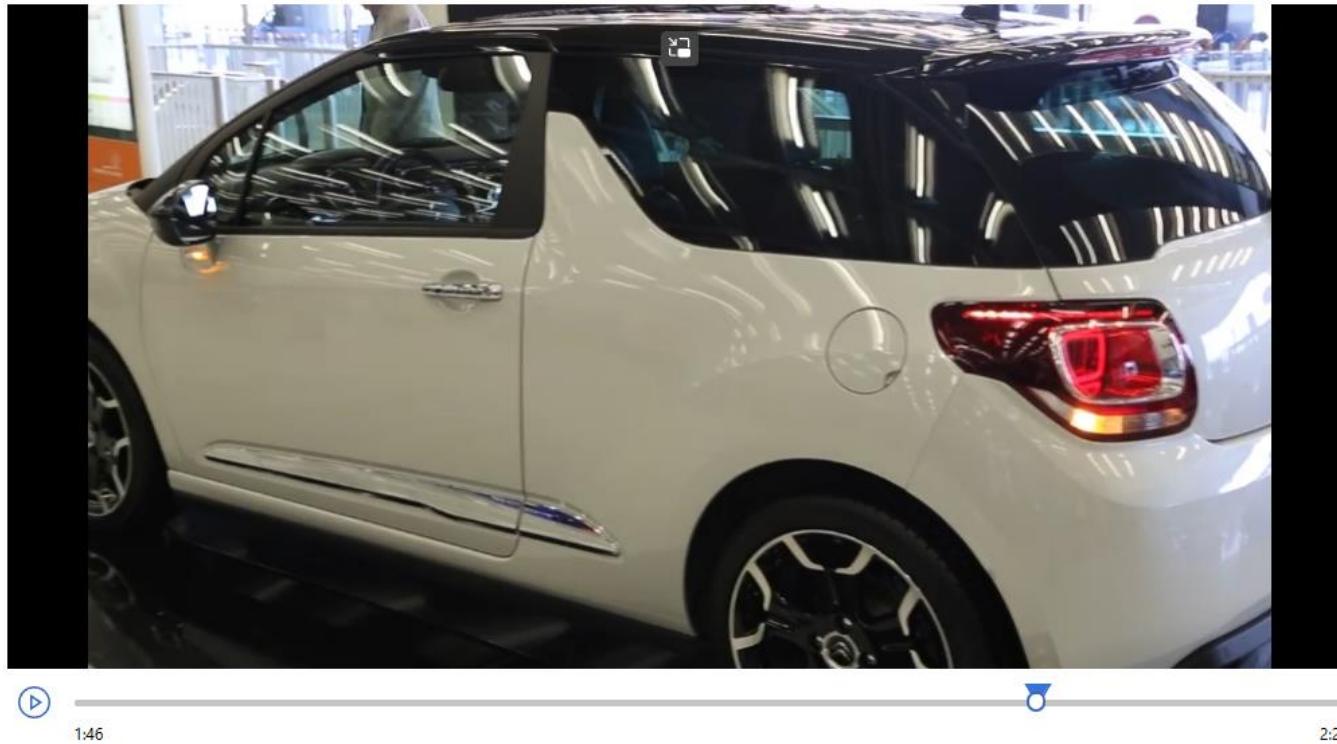
"two motorbikes" appeared at:

- ▼ 00:27.800
- ▼ 00:28.633
- ▼ 00:27.900
- ▼ 00:27.700
- ▼ 00:28.200

Show more

Video summary and frame locator

orly.mp4



Run a test

Locate specific frames using a search query

Search query

car

Run

We can search on any object from the video like a car

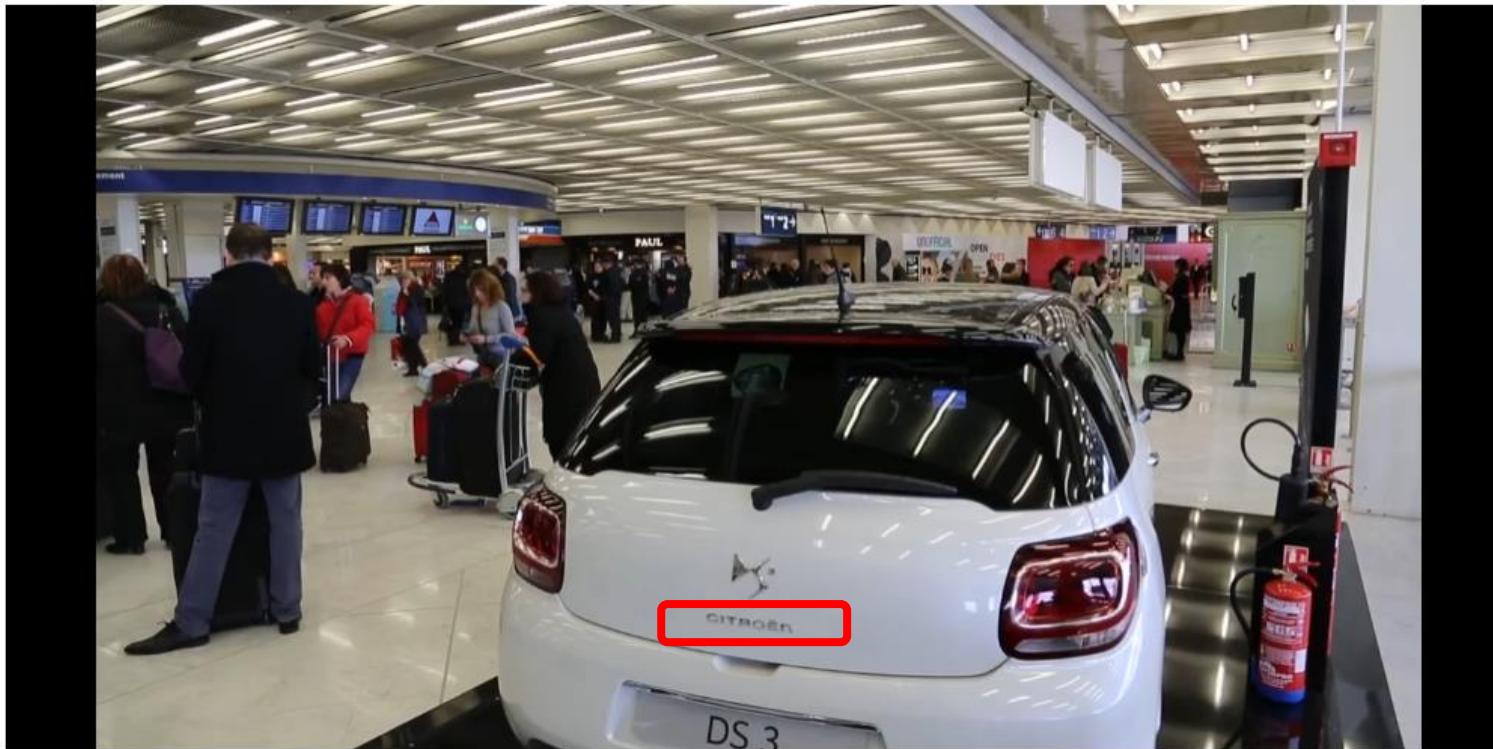
Results

"car" appeared at:

- ▼ 01:46.520
 - ▼ 01:46.800
 - ▼ 01:46.600
 - ▼ 01:46.920
 - ▼ 01:47.200
- Show more

Video summary and frame locator

orly.mp4



1:21



2:21

Run a test

Locate specific frames using a search query

Search query

citroen

Run

We can search on any text from the video like *citroen*

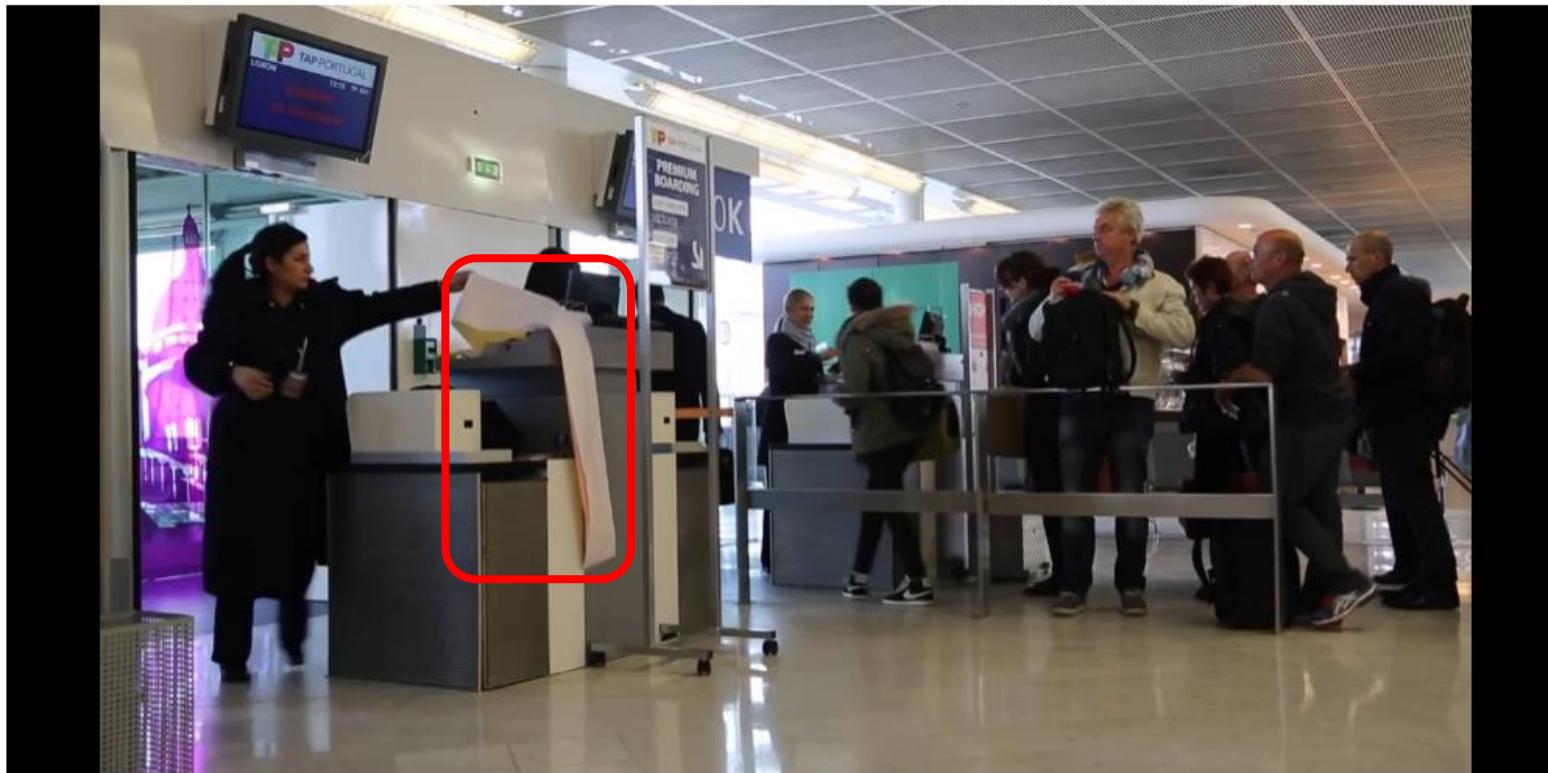
Results

"citroen" appeared at:

- ▼ 01:21.520
 - ▼ 01:21.120
 - ▼ 01:22.120
 - ▼ 01:21.200
 - ▼ 01:21.720
- [Show more](#)

Video summary and frame locator

orly.mp4



2:07



2:21

Run a test

Locate specific frames using a search query

Search query

printer

Run

We can search on any object from the video like a *printer*

Results

"printer" appeared at:

- ▼ 02:07.000
 - ▼ 02:06.920
 - ▼ 02:07.600
 - ▼ 02:07.120
 - ▼ 02:07.520
- Show more

Video summary and frame locator

orly.mp4



Run a test

Locate specific frames using a search query

Search query

cosmétiques

Run

We can search on any object or text like
from the video = « cosmétiques »

Results

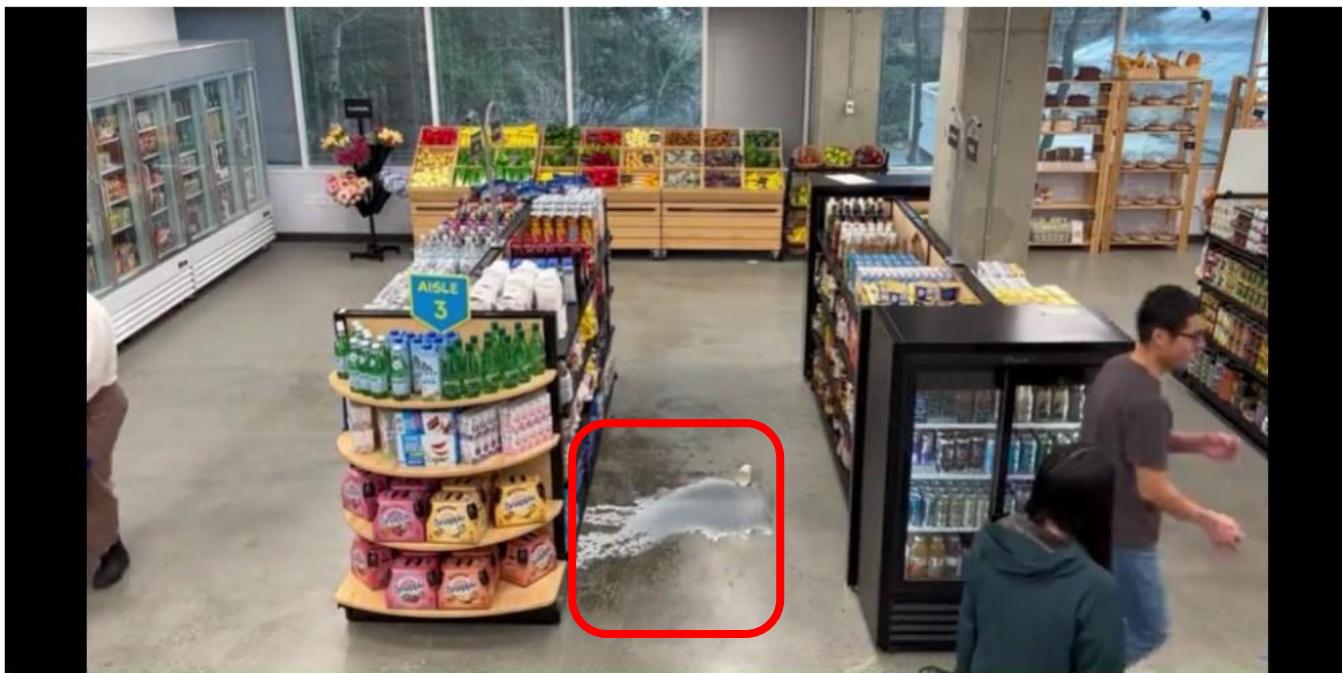
"cosmétiques" appeared at:

- ▼ 02:02.800
- ▼ 02:02.720
- ▼ 02:02.200
- ▼ 02:02.320
- ▼ 02:02.400

Show more

Video summary and frame locator

Retail



Run a test

Locate specific frames using a search query

Search query

White liquid on floor

Run

Results

"White liquid on floor" appeared at:

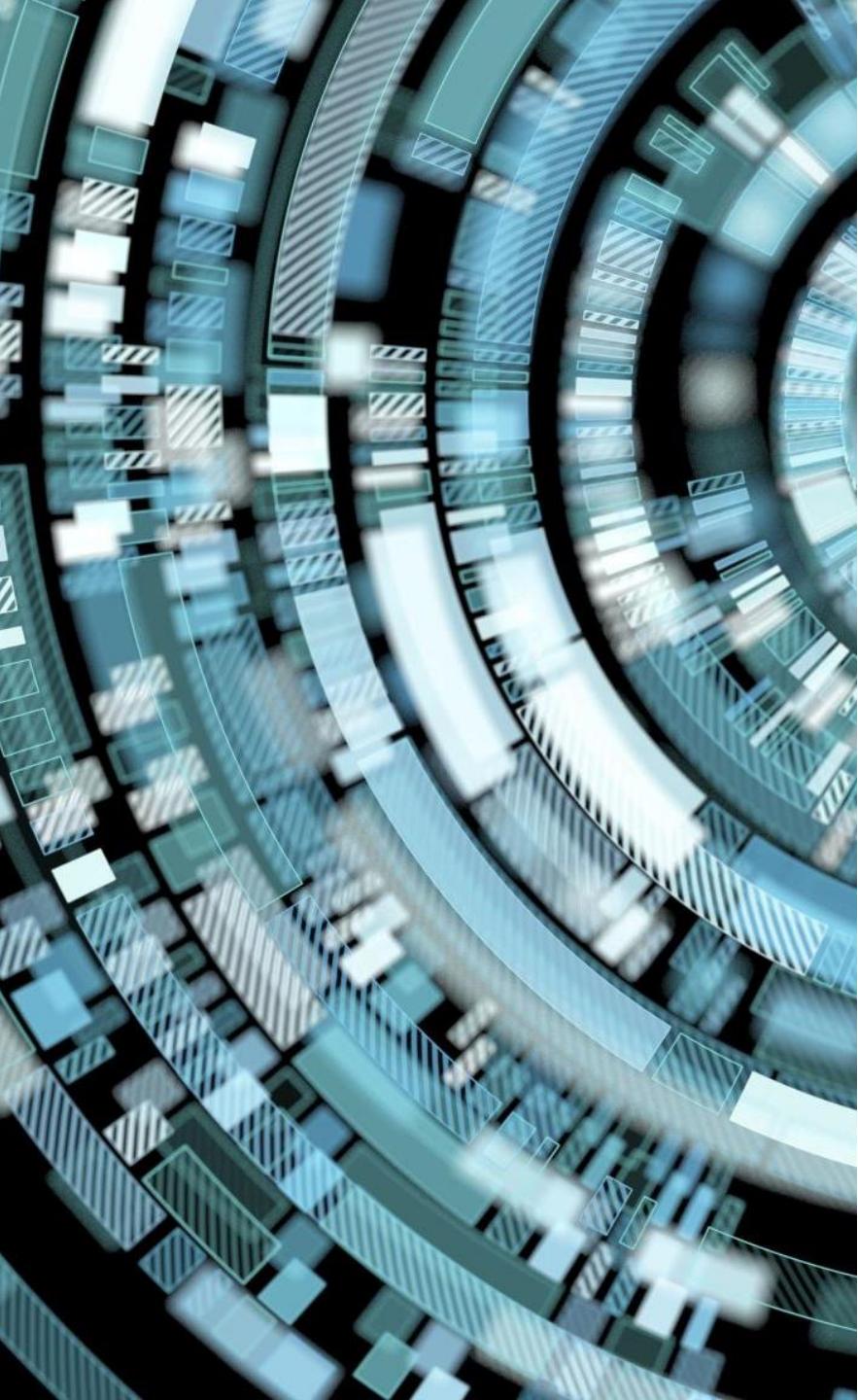
- ▼ 05:15.400
 - ▼ 05:15.500
 - ▼ 04:35.300
 - ▼ 05:17.600
 - ▼ 04:37.100
- Show more

5:15

11:39

The background of the slide features a photograph of a white shelving unit containing three rows of binders. Each binder has a different colored cover (blue, green, yellow, red, grey) and a white page protector. The shelves are filled from left to right with these binders.

Documentation



Documentation

- <http://aka.ms/VisionStudio>
- <https://aka.ms/CognitiveService/Vision>
- [Announcing a renaissance in computer vision AI with Microsoft's Florence foundation model | Azure Blog and Updates | Microsoft Azure](#)
- [Project Florence \(AI\) - Microsoft Research](#)
- [What's new in Computer Vision? - Azure Cognitive Services | Microsoft Learn](#)
- [Quickstart: Image Analysis 4.0 - Azure Cognitive Services | Microsoft Learn](#)

A close-up, high-angle shot of several interlocking metallic gears. The gears have sharp, well-defined teeth and are made of a reflective metal, possibly steel. The lighting is dramatic, highlighting the metallic surfaces and the intricate meshing of the gear teeth. The background is dark and out of focus, making the silver and grey tones of the gears stand out.

Gear up

<https://gearup.microsoft.com/communications/public-preview-azure-cognitive-service-for-vision-features-powered-by-florence>

Demos videos

https://www.youtube.com/playlist?list=PLy4MOYaxz3vMAA_Ie5wRUNiCZ9Sif74
ex

The screenshot shows a YouTube playlist interface. At the top, there's a thumbnail for a video titled "Background removal with Azure Computer Vision 4.0 (Florence)". Below it, another thumbnail for "Image Retrieval with Azure Computer Vision 4.0 (Florence)" is visible. At the bottom, a third thumbnail for "Dense captioning with Azure Computer Vision 4.0 (Florence)" is shown. The main title of the playlist is "Azure Computer Vision 4.0 (FLORENCE)" and it is posted by "Serge Analytics". There are three video thumbnails displayed, each with its title, author, and duration.

Video Title	Author	Duration
Background removal with Azure Computer Vision 4.0 (Florence)	Serge Analytics	1:18
Image Retrieval with Azure Computer Vision 4.0 (Florence)	Serge Analytics	1:55
Dense captioning with Azure Computer Vision 4.0 (Florence)	Serge Analytics	1:37

Python notebooks demos

- <https://github.com/retkowsky/Image-Analysis-with-Azure-Computer-Vision-4.0>
- <https://github.com/retkowsky/Image-Analysis-with-Azure-Computer-Vision-4.0-Captioning-and-Dense-Captioning>
- <https://github.com/retkowsky/Removing-background-using-Azure-Computer-Vision-4>
- https://github.com/retkowsky/image_retrieval_with_Azure_Computer_Vision_4.0



Thank you