Visual search with Azure AI

EMEA AI GBB team

18th of October 2022

Visual search use-cases

You see a person in a magazine and want to find and buy the same clothes.

You find a product in a website you like and want to find the same product in a different color.

You find a spare part in your company's warehouse that does not have any reference. You want to identify this spare part with its part number and/or additional information (spare parts recognition use-case).

Sometimes marking with serial numbers or bar codes/QR codes is not possible for all the products. So, you need to visual search application to quickly identify the product reference using only an image.

You find a product in a store, but the price is too expensive. You want to find similar products that are more affordable.

You find a nice article you want to buy. But this product is not available anymore. You want to find similar products that you can bought.

You want to have a visual recommendation in your merchant website for a better customer experience.

I would like to search for similar paintings of this work that I have seen in the museum.

Visual search?

- Visual search — **the ability to initiate a search query using an image captured by the camera lens on a mobile device or using any existing image** — has increasingly become a channel that can drive consumers from becoming aware of a product to making a purchase.

- **Gartner classifies visual search as an emerging technology**, which puts it right on par with findings from eMarketer survey suggesting that few consumers “regularly” use it.

- **On average, only 3% regularly use visual search and only 10% have used it in the past, according to the findings**. On the other site of the spectrum, 7% are familiar with the technology, according to an eMarketer ecommerce survey conducted in June 2019 by Bizrate Insights and published in August 2019.

Visual search techniques

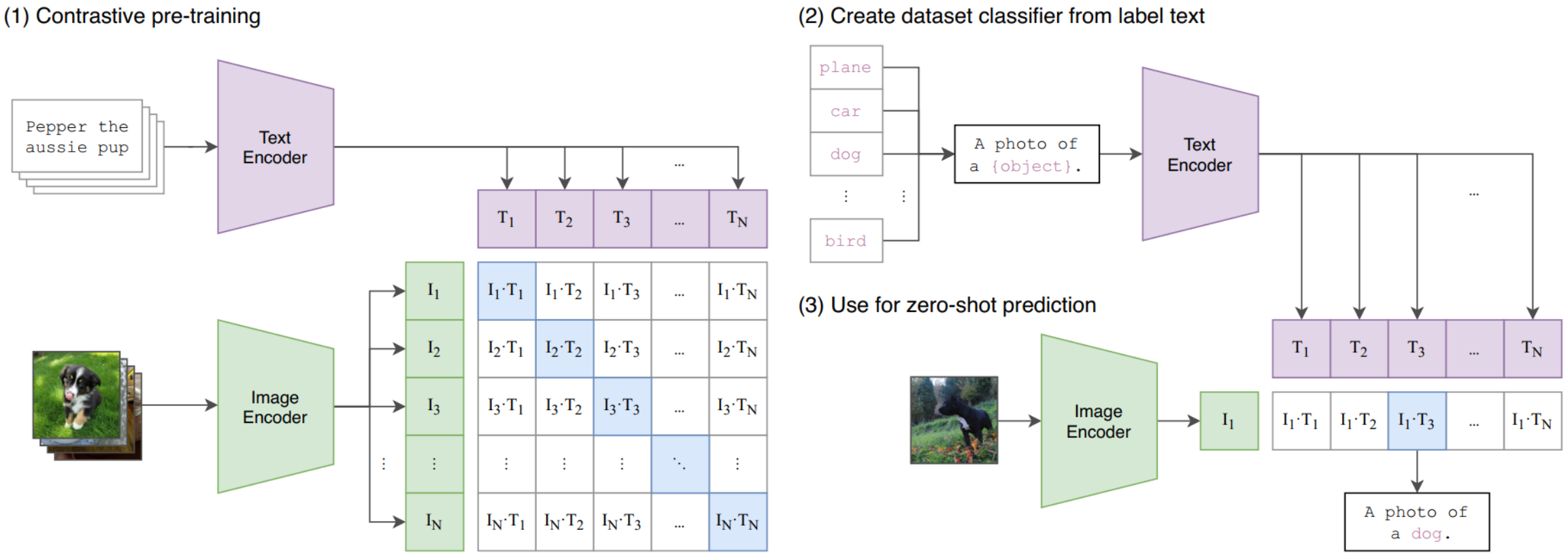
- The goal of this is Azure AI asset is to **enable search over Text and Images using Azure Cognitive Search.**

- The technique was inspired by a research article in 2016 which show how to **convert vectors (embeddings) to text which allows the Azure Cognitive Search service** to leverage the inverted index to quickly find the most relevant items.

- This technique has shown to be incredibly effective and easy to implement. We are using **Sentence Transformers, which is an OpenAI Clip model** wrapper and some **Azure AI cognitive services (Azure Computer Vision)** for OCR and automatic image descriptions and attributes generation.

What is OpenAI Clip?

[CLIP: Connecting Text and Images (openai.com)]



Open AI Clip – an example

	text	confidence
0	games console	0.273042
1	Xbox	0.265951
2	PS5	0.261705
3	Sony	0.261019
4	play station	0.255410
5	Microsoft	0.233499
6	controller	0.233241
7	white controller	0.227006
8	black controller	0.219858
9	truck	0.177364
10	apple	0.174288
11	fish	0.172075
12	Miami	0.171725
13	car	0.168142
14	street	0.167989
15	guitar	0.147125

```
Image file: ./images_unsplash_25kphotos/--2IBUMom1I.jpg
Width = 640 Height = 853
Size: 78.5 kB Date: 2021-02-21 22:29:10
```

–2IBUMom1I.jpg

```
imgfile = "./images_unsplash_25kphotos/--2IBUMom1I.jpg"
vectext = vec2Text.imageEmbedding(imgfile, model=model)
vectext
```

```
array([ 5.62955499e-01,  2.77076483e-01,  1.49947733e-01, -4.27496612e-01,
       -8.11115652e-02, -2.03692347e-01,  4.18303907e-03,  6.78090692e-01,
       -4.21069711e-01, -1.87757015e-02,  2.84420818e-01, -2.84125209e-01,
        6.20087028e-01, -6.90742612e-01,  3.09186876e-01, -1.44741684e-02,
       -1.08902669e+00, -2.58051455e-02,  5.62377393e-01, -1.25617191e-01,
        3.35174203e-01,  3.02467763e-01,  1.75208330e-01, -4.13278222e-01,
       -4.04964805e-01,  2.16677040e-02, -4.77743745e-02, -3.60322893e-02,
       -3.27542067e-01,  1.35492384e-02, -1.76186830e-01,  3.00059438e-01,
       -2.82461166e-01,  1.66423023e-02,  2.51844972e-02,  7.83817321e-02,
       -4.04500812e-01,  2.86668837e-01,  3.10988903e-01,  1.48953408e-01,
       -4.99043316e-01, -2.66104937e-03, -2.40858778e-01, -1.20103344e-01,
        4.17896986e-01, -5.51473618e-01,  1.17683932e-01,  5.77473998e-01,
        1.45307705e-02, -2.79569149e-01,  5.95383942e-01,  3.34215015e-01,
        5.78037798e-01, -8.01847652e-02, -3.37295562e-01, -1.87236145e-02,
        1.22385673e-01,  1.12165213e-01, -5.88083923e-01,  1.63273424e-01,
       -2.36464858e-01, -1.61946446e-01,  5.06792702e-02,  3.20442468e-02,
       -1.17926411e-01, -7.84655809e-02,  1.08443990e-01,  1.11933911e+00,
       -1.11482032e-01, -1.57323226e-01, -6.97274208e-01,  5.80847263e-02,
        1.49179488e-01. -3.04919153e-01. -6.41463995e-02.  4.55385596e-02.
```

The objective is to embed all our existing catalog of images.
Then the objects embedding are converted into a set of **fake terms** and all the results are stored into an **Azure Cognitive Search index** for handling all the search requests. For example, if an embedding looked like [-0,21, .123, ..., .876], this might be converted to a set of fake terms such as: “A1 B3 … FED0”. This is what is sent as the search query to Azure Cognitive Search.

From an image to a VecText

Similar image 1 | score =217.21
./images/catalog_images/catalog_image_00028.jpg
h: 853 w: 640 | 2022-09-29 09:27:24 | 58.1 kB

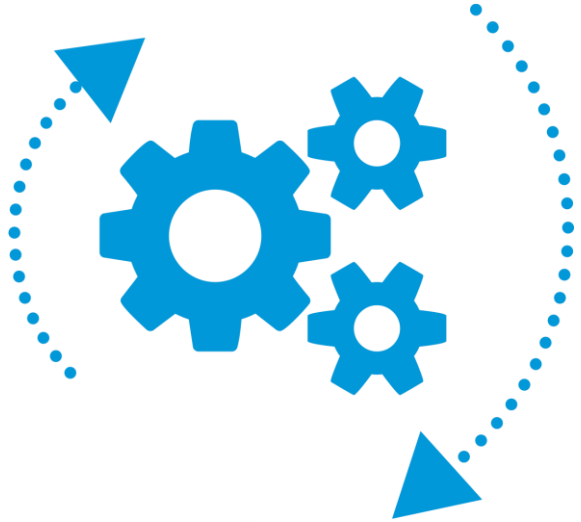
Similar image 2 | score =193.83
./images/catalog_images/catalog_image_00318.jpg
h: 853 w: 640 | 2022-09-29 10:04:38 | 59.8 kB

Similar image 3 | score =179.28
./images/catalog_images/catalog_image_00317.jpg
h: 767 w: 576 | 2022-09-29 10:04:40 | 45.1 kB

Similar image 4 | score =178.04
./images/catalog_images/catalog_image_00029.jpg
h: 853 w: 640 | 2022-09-29 09:27:24 | 69.2 kB

An example

Visual search Process



We have here a collection of catalog images.

For each of these images, we will embed them using **sentence transformers**. Sentence transformer can be used to map images and texts to the same vector space.

As model, we use the **openai CLIP model** which was trained on a large set of images and image alt texts.

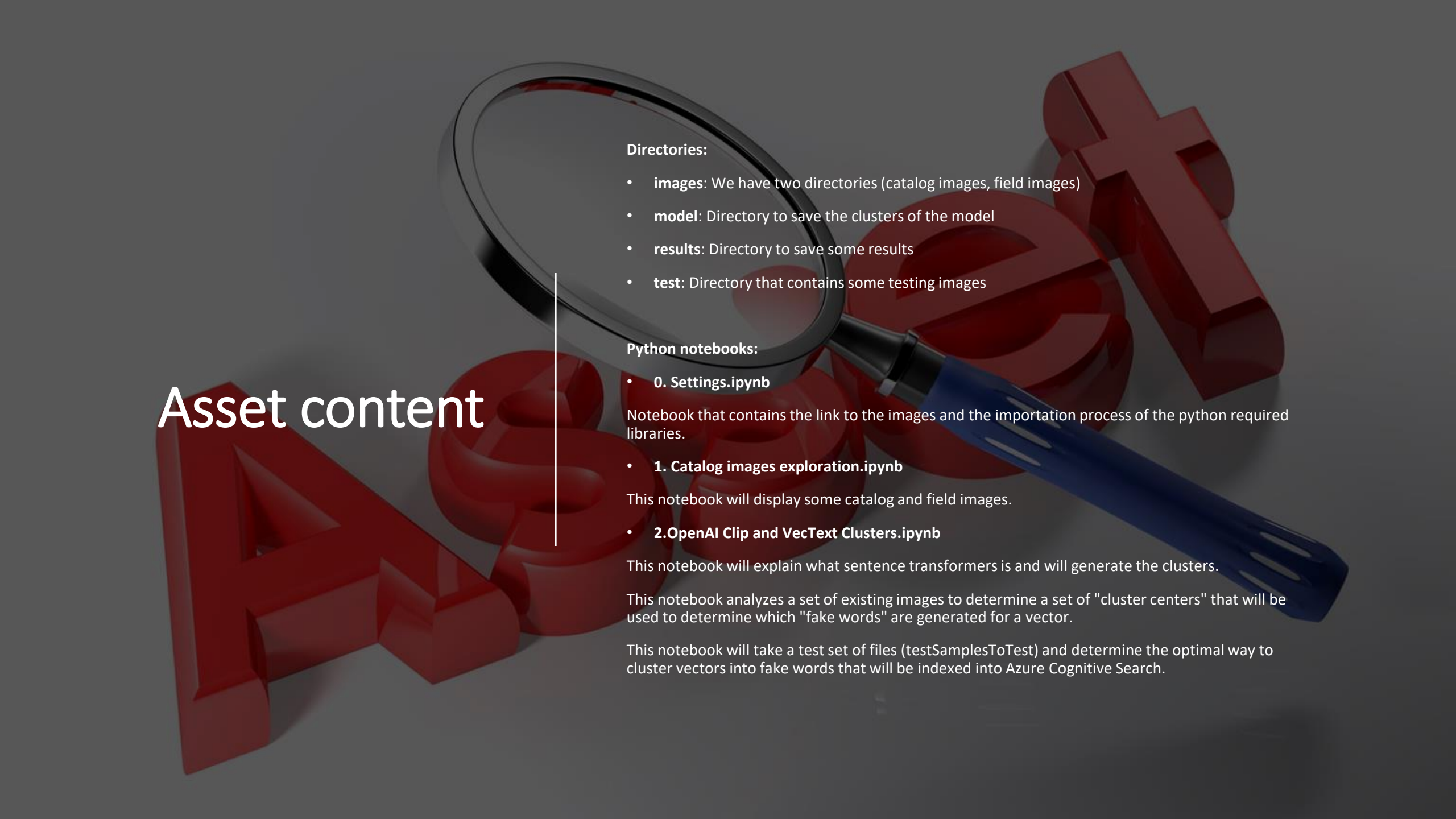
We can retrieve any text from these images using **azure read API** (if any text is available).

We can retrieve any text information from any **bar code or QR code** (if any).

All these information will be **ingested into an azure cognitive search index.**

Then if you have a field image, you can embed it and extract any text/barcode information and call the azure cognitive search index to retrieve any similar images using **vectext.**

Asset content



Directories:

- **images**: We have two directories (catalog images, field images)
- **model**: Directory to save the clusters of the model
- **results**: Directory to save some results
- **test**: Directory that contains some testing images

Python notebooks:

- **0. Settings.ipynb**

Notebook that contains the link to the images and the importation process of the python required libraries.

- **1. Catalog images exploration.ipynb**

This notebook will display some catalog and field images.

- **2.OpenAI Clip and VecText Clusters.ipynb**

This notebook will explain what sentence transformers is and will generate the clusters.

This notebook analyzes a set of existing images to determine a set of "cluster centers" that will be used to determine which "fake words" are generated for a vector.

This notebook will take a test set of files (testSamplesToTest) and determine the optimal way to cluster vectors into fake words that will be indexed into Azure Cognitive Search.

Asset content

- **3. VecText generation.ipynb**

This notebook will generate the vectext embedding for all the catalog images.

- **4. BarCode Information extraction.ipynb**

This notebook will detect any barcode or QR code from the catalog images and will extract the information.

- **5. Azure CV for OCR, tags, colors and captions.ipynb**

This notebook will use Azure Computer Vision or OCR, colors, tags and caption extraction for each of the catalog images.

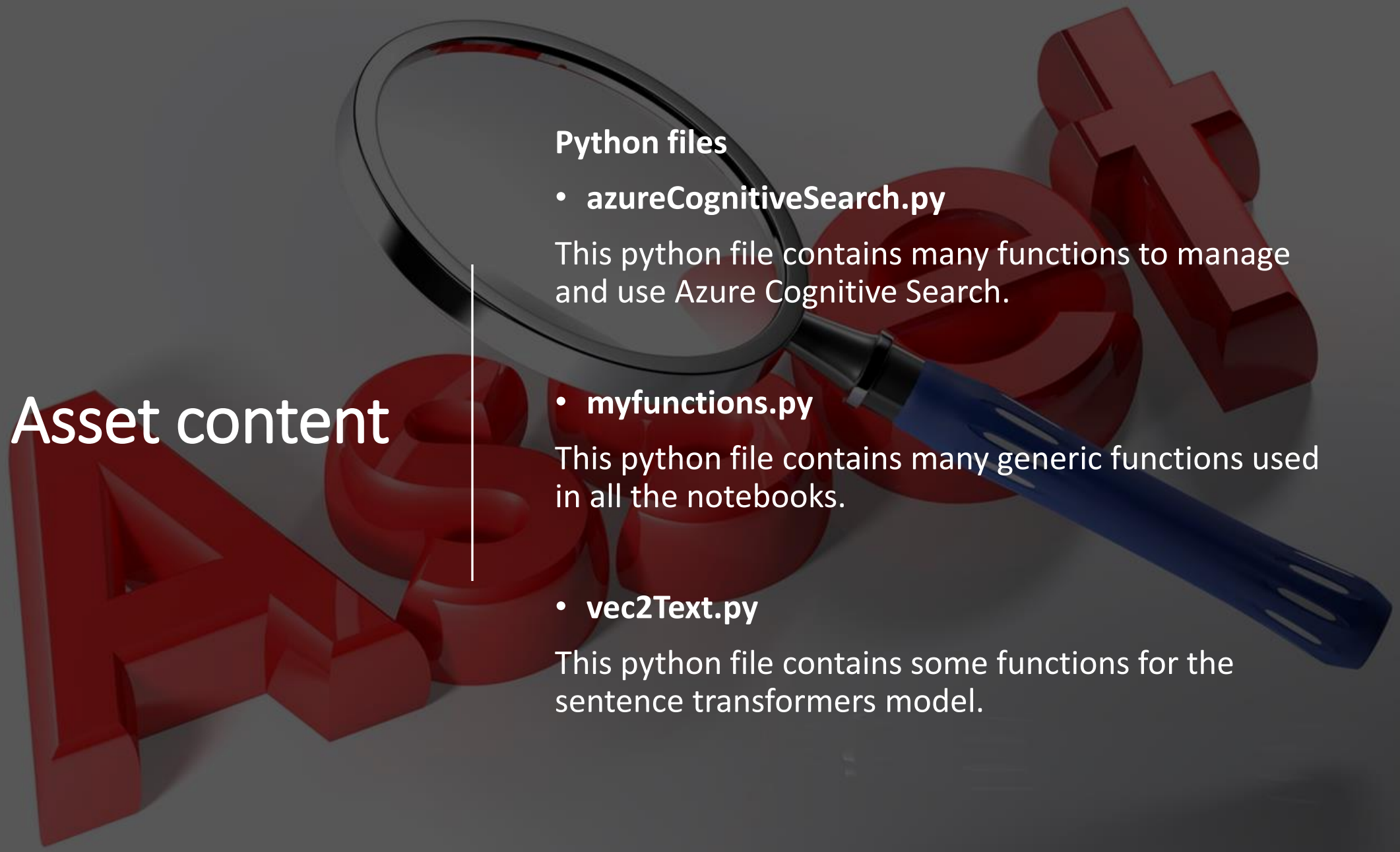
- **6. Azure Cognitive Search Index Generation.ipynb**

This notebook will show how to ingest all the information into an Azure Cognitive Search index.

- **7. Calling Azure Cognitive Search.ipynb**

We can now test the index using some images similarity visual search or free text queries using azure Cognitive Search.

Asset content



Python files

- **azureCognitiveSearch.py**

This python file contains many functions to manage and use Azure Cognitive Search.

- **myfunctions.py**

This python file contains many generic functions used in all the notebooks.

- **vec2Text.py**

This python file contains some functions for the sentence transformers model.

Prerequisites

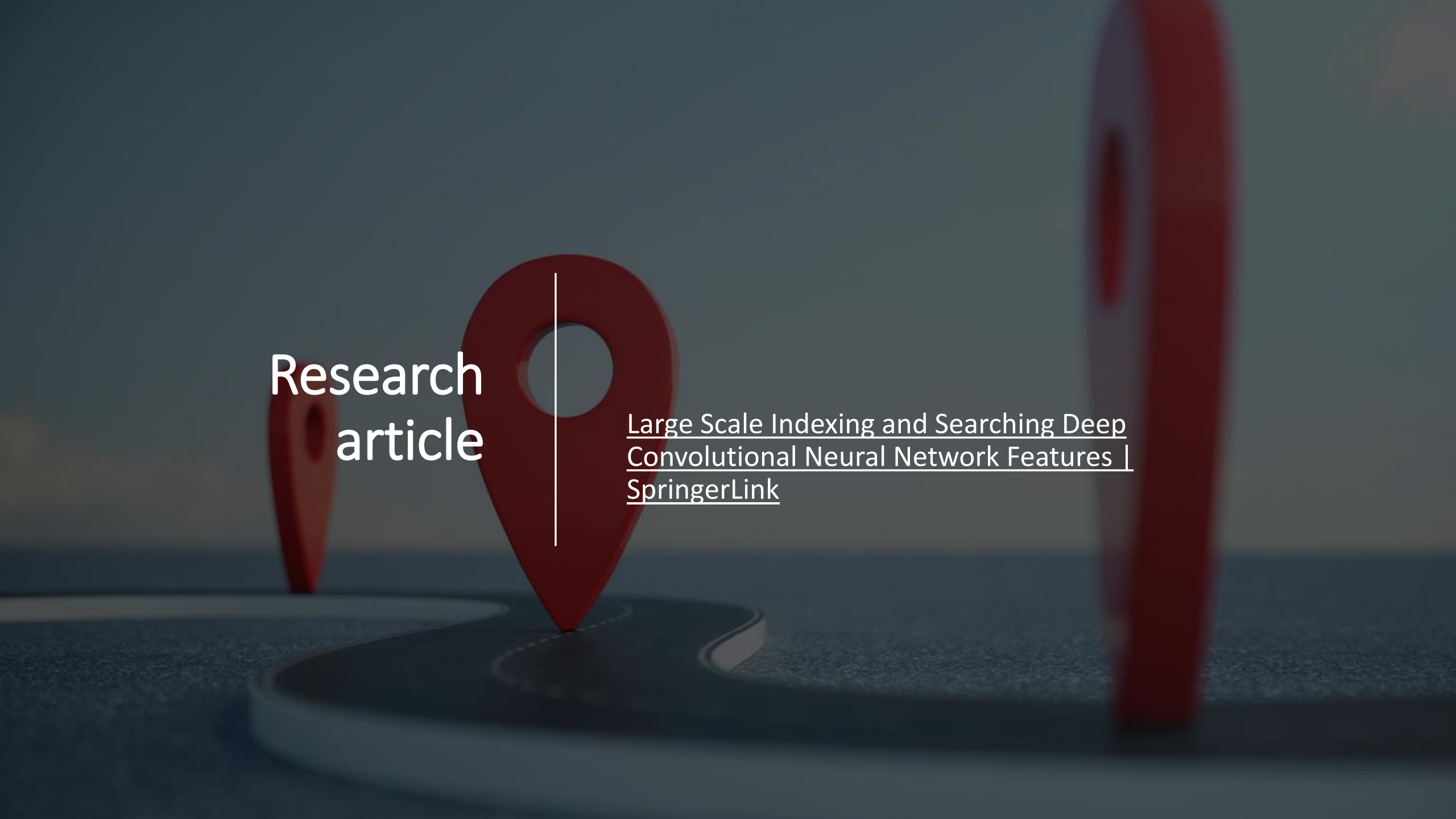
We need to have access to your products, objects or images data.

We need multiple images per product in the “catalog” collection of image (between 5 to 10).

Good quality of images is required (good lightning, no blurring…).

Try as much as possible to avoid images quality bias between “catalog” images and “field” images (light, blur, definition, colors, …).

We need to avoid as much as possible to have multiple products in the field image as compared to catalog images where we usually have 1 product per 1 image.



Research article

Large Scale Indexing and Searching Deep Convolutional Neural Network Features | SpringerLink

Reference

- https://azure.microsoft.com/en-us/products/search/
- https://azure.microsoft.com/en-us/products/cognitive-services/computer-vision/#overview
- https://learn.microsoft.com/en-us/azure/cognitive-services/Computer-vision/how-to/call-read-api
- https://zbar.sourceforge.net/
- https://github.com/liamca/vector-search

Where to download this asset?

Links: https://github.com/retkowsky/azure_visual_search_toolkit

Contacts:

Serge Retkowsky

EMEA AI GBB serge.retkowsky@microsoft.com

Thank you

Appendix

Process

Find Optimal Number of Clusters per Feature

[-0.1, 0.3 ,… 0.4]

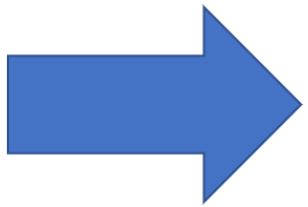
[0.9, -0.7,… -0.8]

.

.

.

[0.3, 0.2,… 0.2]



[5]

[6]

.

.

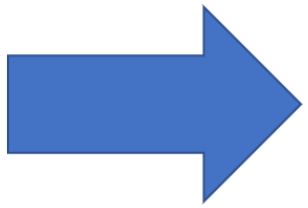
.

[5]

This is needed because you want to be able to map numbers that are close to a single term

Process

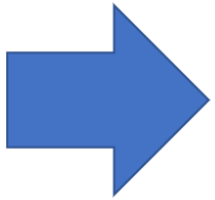
Find Clusters Centers for each Feature

[-0.1, 0.3 ,… 0.4]	[5]		[-0.15, 0.22 ,… 0.41]
[0.9, -0.7,… -0.8]	[6]		[-.32, -0.73,… 0.85]
.	.		.
.	.		.
.	.		.
[0.3, 0.2,… 0.2]	[5]		[-.7, 0.02,… 0.52]

Process

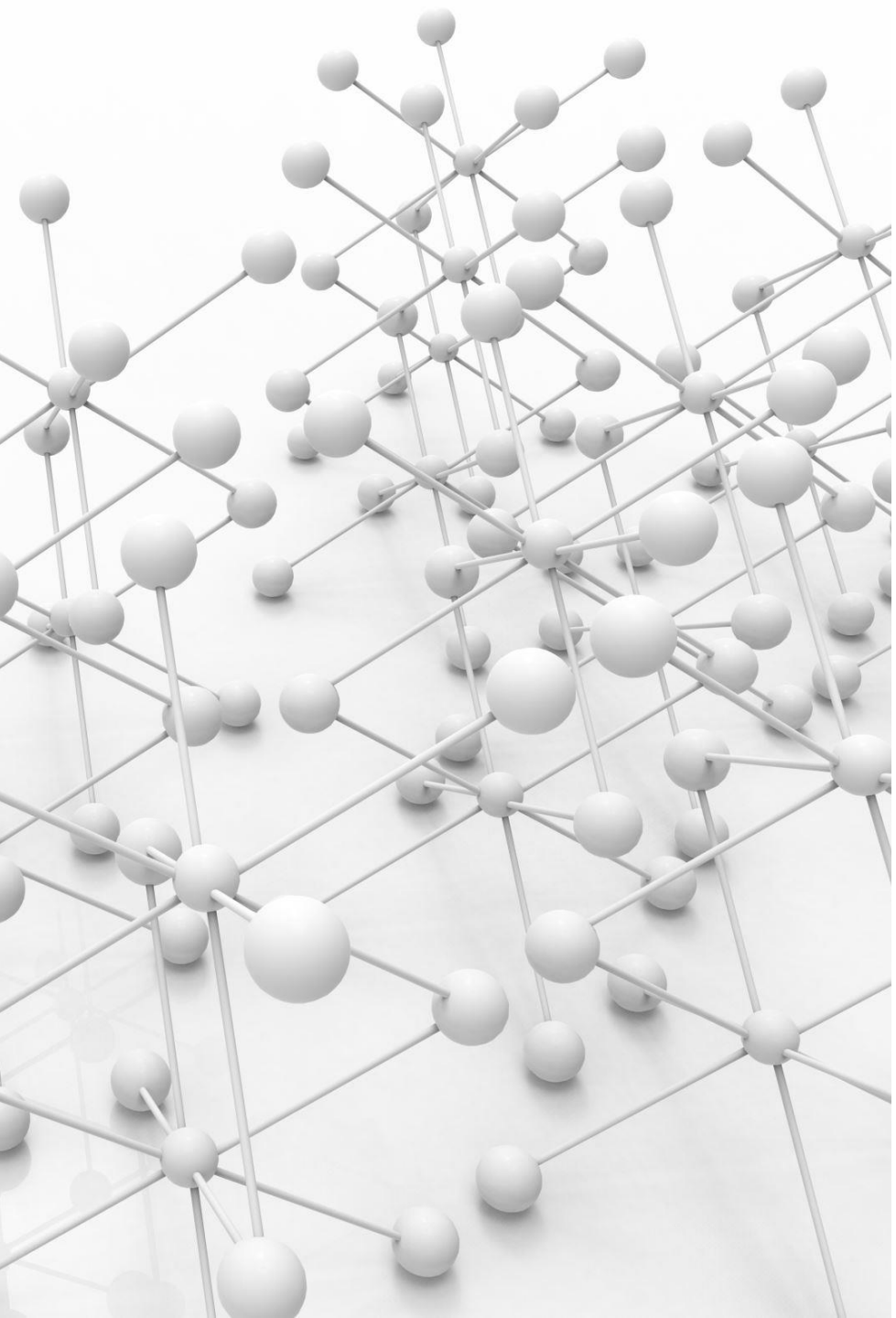
Define a Fake Term for each Cluster Center based on Number of Clusters and Save Them

[-0.15, 0.22 ,… 0.41]	[5]	[A0, A1,… A5]
[-.32, -0.73,… 0.85]	[6]	[B0, B1,… B6]
.	.	.
.	.	.
.	.	.
[-.7, 0.02,… 0.52]	[5]	[FED0, FED1,… FED5]



About OpenAI Clip

- **CLIP is the first multimodal** (in this case, vision and text) model tackling computer vision and was recently released by OpenAI on January 5, 2021.
- From the OpenAI CLIP repository**, "CLIP (Contrastive Language-Image Pre-Training) is a neural network trained on a variety of (image, text) pairs**. It can be instructed in natural language to predict the most relevant text snippet, given an image, without directly optimizing for the task, similarly to the zero-shot capabilities of GPT-2 and 3."
- CLIP is a neural network model. **It is trained on 400,000,000 (image, text) pairs**. An (image, text) pair might be a picture and its caption. So this means that there are 400,000,000 pictures and their captions that are matched up, and this is the data that is used in training the CLIP model**. "It can predict the most relevant text snippet, given an image."** You can input an image into the CLIP model, and it will return for you the likeliest caption or summary of that image.



About OpenAI Clip

- Pros:
  - A neural network model built on **hundreds of millions of images and captions**,
  - Can return the **best caption given an image,** and
  - Has impressive **"zero-shot" capabilities**, making it able to accurately predict entire classes it's never seen before (*zero-shot model* allows us to classify data, which wasn't used to build a model)
- Cons:
  - **Bias** coming from the training datasets
  - **Images Classification, object detection, instance segmentation can be more efficient** for some use-cases

Open AI Clip Demos

- https://github.com/retkowsky/visual_search_openai_clip
- https://github.com/retkowsky/Finding-duplicated-images-with-Sentence-Transformers

```
In [22]: search("Musée du Louvre", 2)

Your query: Musée du Louvre

Results:

1 - Catalog image ID: images/img (990).jpg  with score = 0.30649
```

```
2 - Catalog image ID: images/img (975).jpg  with score = 0.30247
```

```
Done in 0.12015 secs
```

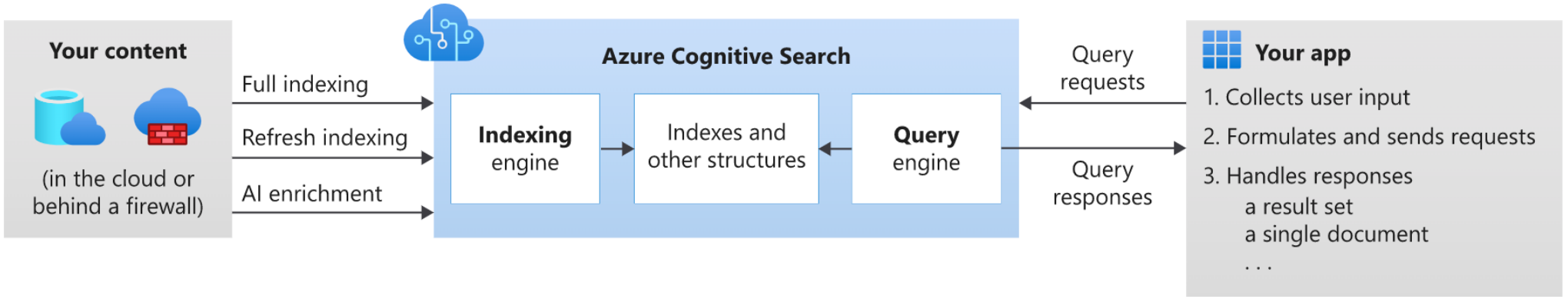
```
find_duplicates()

Searching images duplicates...

****************** Duplicate # 1 ******************

Both images are duplicates:
images/image (5).jpg and images/image (6).jpg

 images/image (5).jpg
```

```
 images/image (6).jpg
```

```
images/image (5).jpg | size = 7.0 kB | date : Wed Sep 21 08:52:44 2022
images/image (6).jpg | size = 7.0 kB | date : Wed Sep 21 08:52:37 2022


****************** Duplicate # 2 ******************

Both images are duplicates:
images/image (8).jpg and images/image (9).jpg

 images/image (8).jpg
```

```
 images/image (9).jpg
```

```
images/image (8).jpg | size = 22.1 kB | date : Wed Sep 21 08:52:32 2022
images/image (9).jpg | size = 15.6 kB | date : Wed Sep 21 08:52:42 2022
```



Azure Cognitive Search

- Azure Cognitive Search (formerly known as "Azure Search") is a cloud search service that gives developers infrastructure, APIs, and tools for building a rich search experience over private, heterogeneous content in web, mobile, and enterprise applications.
- Rich indexing, with lexical analysis and optional AI enrichment for content extraction and transformation
- Rich query syntax for text search, fuzzy search, autocomplete, geo-search and more
- Programmability through REST APIs and client libraries in Azure SDKs
- Azure integration at the data layer, machine learning layer, and AI (Cognitive Services)

Microsoft

Thank you