Elasticsearch is for the Birds

Identifying feathered-friends using vector-embedded images and similarity search

Justin Castilla

@Castilla_Codes

justin.castilla@elastic.co

Senior Developer Advocate

Thank you for having me, Voxel51!

elastic

Motivation

Birds have a vast set of unique visual qualities perfectly suited for vector database exploration.

Observable Traits

- size
- color
- texture
- shape
- habitat
- behavior






First, acquire a source of bird data

- 525 bird species folders
- 150-200 images each
- 224 × 224 pixels
- CSV file with name and scientific name
- Misspelled folders/species names
- Lack of apostrophes
- Wrong names
- ALLCAPS

kaggle™

Second: clean the bird data

- Python scripting heaven!
- Capitalize bird names
- Cross-reference bird names with wikimedia using wptools client
- Use wptools to verify names and retrieve descriptions
- Correct unverified names
- Align all names with wikimedia

But How does Elastic fit in?

- the company behind the open-source search engine Elasticsearch
- powers search, logging, observability, security, and analytics solutions
- offers traditional search (BM25) as well as a flexible vector search platform

We need an index to keep our birds organized

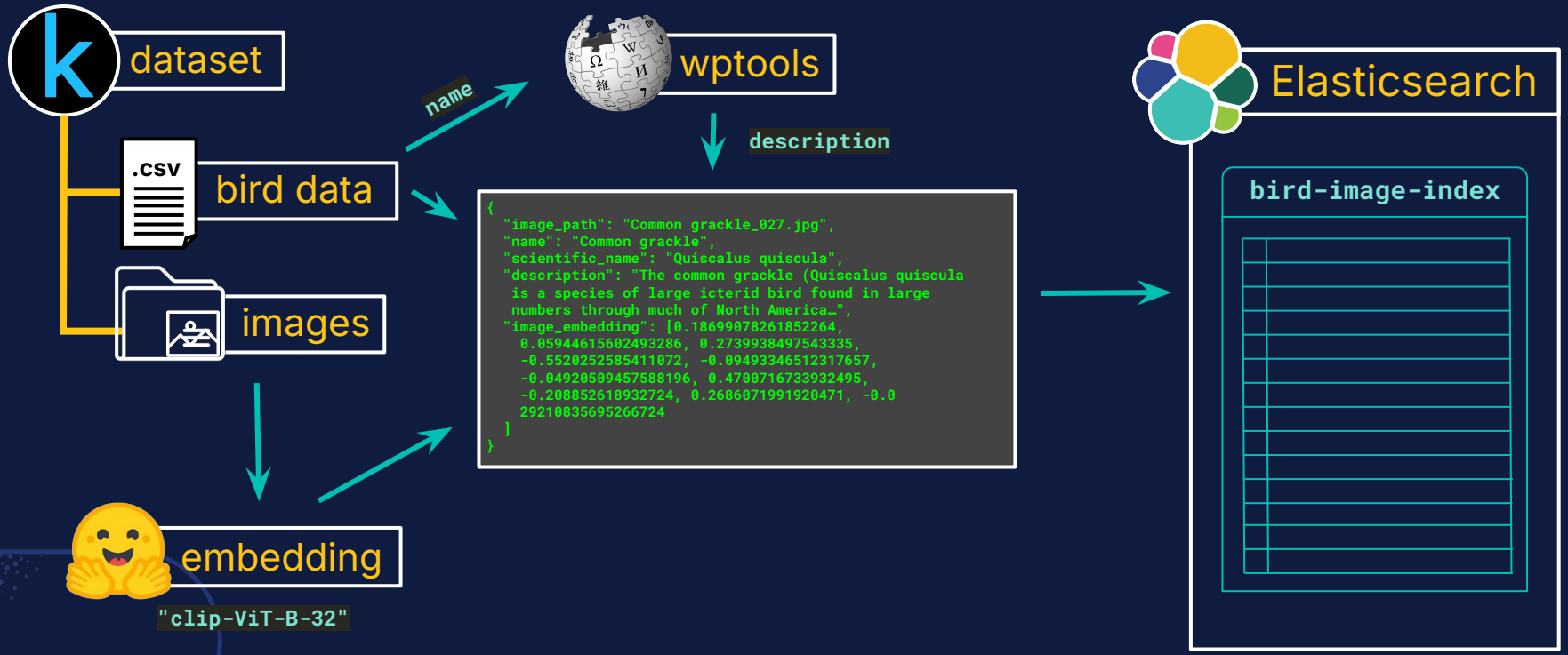
Create bird-image-index

name

image_embedding

```
PUT /bird-image-index
{
  "mappings": {
    "properties": {
      "name": { "type": "keyword" },
      "scientific_name": { "type": "keyword" },
      "description": { "type": "text" },
      "image_embedding": {
        "type": "dense_vector",
        "dims": 512,
        "index": true,
        "similarity": "cosine"
      },
      "image_path": { "type": "keyword" },
    }
  }
}
```

Populating the Elasticsearch Vector Database



Run the code: embedding an image

```
from sentencew_transformers import SentenceTransformer
from PIL import Image


model = SentenceTransformer("clip-ViT-B-32")

image_path = "Common grackle_170.jpg"
temp_image = Image.open(image_path)
embedded_image = model.encode(temp_image).tolist()
```

Run the code: add one doc to index

```
from elasticsearch import Elasticsearch
import wptools

es_db = Elasticsearch(
  hosts="<SUPER SECRET ENDPOINT>",
  api_key="<SUPER SECRET API KEY>")

bird = {
    "name": "Common grackle",
    "scientific_name": "Quiscalus quiscula",
    "image_path": image_path,
    "image_embedding": embedded_image,
}

bird["description"] = wptools.page(bird["name"]).get_parse().data["description"]

es_db.index(index="bird-image-index", body=bird)
```

Run the code: add bulk docs to index

```
operations = []
for bird in birds:
    operations.append({"index": {"_index": "updated-birds"}})
    operations.append(bird)

batch_size = 500

for i in range(0, len(operations), batch_size):
    chunk = operations[i : i + batch_size]
    es_db.bulk(body=operations[i : i + batch_size])
```

Querying the Elastic Vector Database

embedding

```
[
  0.18699078261852264,
  0.0594615602493286,
  0.2739938497543335,
  -0.550252585411072,
  -0.093346512317657,
  -0.040509457588196,
  0.4700716733932495,
  -0.208852618932724
  0.2686071991920471,
  -0.020835695266724
]
```

Query

Elasticsearch

bird-image-index

```
{
  "name": "Common grackle",
  "scientific_name": "Quiscalus quiscula",
  "score": 95.38994,
  "Image_path": "Common grackle_170.jpg
}
```

Results

Run the code: create a vector query

```
temp_image = Image.open(query_image_path)
query_embedding = model.encode(temp_image).tolist()

knn = {
    "field": "image_embedding",
    "k": 5,
    "num_candidates": 100,
    "query_vector": query_embedding,
    "boost": 100,
}

# The fields to retrieve from the matching documents
fields = ["name", "scientific_name", "image_path", "description"]

results = es_db.search(index="bird-image-index", body={"knn": knn, "_source": fields})
```

Run the code: actually run some code!

Further developments

Live Camera identification: a Raspberry Pi camera trained on a birdfeeder to identify and log visitors and notify of new species encountered

PR to include image models in Elasticsearch trained models in hosting capabilities

Enhance dataset with more information and a bird-trained model to create a RAG interface

Add more birds!

Before you fly off!

Try this code locally!

github.com/justincastilla/elastic-bird

Thank you!

Email: justin.castilla@elastic.co
Twitter: @Castilla_Codes