Processing Passport Image Annotations Using The Elastic Stack

December 14, 2024

Prepared by Yara Mahfouz

Contents

1 Tools Setup

1.1 Elasticsearch

Create a new docker network.

```
docker network create elastic
```

Pull the elasticsearch docker image.

```
docker pull docker.elastic.co/elasticsearch/elasticsearch:8.16.1
```

Start an elasticsearch container.

```
docker run --name es01 --net elastic -p 9200:9200 -it -m 1GB docker.elastic
    .co/elasticsearch/elasticsearch:8.16.1
```

Optionally generate elasticsearch password and enrollment token if needed.

```
docker exec -it es01 /usr/share/elasticsearch/bin/elasticsearch-reset-
    password -u elastic
```

```
docker exec -it es01 /usr/share/elasticsearch/bin/elasticsearch-create-
    enrollment-token -s kibana
```

Store password in an environment variable.

```
export ELASTIC_PASSWORD="your_password"
```

Copy the http_ca.crt SSL certificate from the container to your local machine.

```
docker cp es01:/usr/share/elasticsearch/config/certs/http_ca.crt .
```

1.2 Kibana

Pull the Kibana Docker image.

```
docker pull docker.elastic.co/kibana/kibana:8.16.1
```

Start a kibana container.

```
docker run --name kib01 --net elastic -p 5601:5601 docker.elastic.co/kibana
    /kibana:8.16.1
```

1.3 Logstash

Pull the logstash docker image.

```
docker pull docker.elastic.co/logstash/logstash:8.16.1
```

Create a logstash.yml file, Logstash requires a logstash.yml file for configuration.

```
mkdir -p ~/pipeline/config
nano ~/pipeline/config/logstash.yml
```

Add the following minimal configuration to logstash.yml.

```
http.host: "0.0.0.0"
path.config: /usr/share/logstash/pipeline
```

Ensure that the /pipeline/ directory contains a valid pipeline configuration file.

```
nano ~/pipeline/logstash.conf
```

Add a basic configuration in logstash.conf for testing.

```
input {
  stdin {}
}

output {
  stdout {
    codec => rubydebug
  }
  elasticsearch {
    hosts => ["https://<es01_container_id>:9200"]
    user => "elastic"
    password => "<your_password>"
    ssl => true
    cacert => "/usr/share/logstash/config/http_ca.crt"
  }
}
```

Ensure the Elasticsearch CA certificate is in the /pipeline/config/ directory.

```
docker cp es01:/usr/share/elasticsearch/config/certs/http_ca.crt ~/pipeline
    /config/
```

Re-run the Logstash container with the updated configuration.

```
docker run --rm -it -v ~/pipeline/:/usr/share/logstash/pipeline/ -v ~/
    pipeline/config/:/usr/share/logstash/config/ --network elastic docker.
    elastic.co/logstash/logstash:8.16.1
```

2 Data Ingestion

Before ingestion, an explicit index mapping is defined to customize data analysis and prevent dynamic mapping errors. The image title is stored in two formats: text and keyword. For the text field, a custom tokenizer is implemented to split the title on non-alphanumeric characters, optimizing search for specific components of the title, such as an image ID. The keyword field is normalized to lowercase, enabling case-insensitive exact matches.

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X PUT https://
    localhost:9200/labeled_images_yara_mahfouz -H "Content-Type:
    application/json" -d '
{
  "settings": {
    "analysis": {
      "tokenizer": {
        "custom_image_tokenizer": {
          "type": "pattern",
          "pattern": "[^a-zA-Z0-9]+"
        }
```

```
    },
    "analyzer": {
      "custom_image_analyzer": {
        "type": "custom",
        "tokenizer": "custom_image_tokenizer",
        "filter": ["lowercase"]
      }
    },
    "normalizer": {
      "lowercase_normalizer": {
        "type": "custom",
        "filter": ["lowercase"]
      }
    }
  }
},
"mappings": {
  "properties": {
    "document": {
      "properties": {
        "image": {
          "type": "text",
          "analyzer": "custom_image_analyzer",
          "fields": {
            "keyword": {
              "type": "keyword",
              "normalizer": "lowercase_normalizer"
            }
          }
        },
        "annotations": {
          "type": "nested",
          "properties": {
            "label": {
              "type": "keyword",
              "normalizer": "lowercase_normalizer"
            },
            "class": {
              "type": "keyword",
              "normalizer": "lowercase_normalizer"
            },
            "coordinates": {
              "properties": {
                "x": { "type": "float" },
                "y": { "type": "float" },
                "width": { "type": "float" },
                "height": { "type": "float" }
              }
            }
          }
        },
        "extracted_date": {
          "type": "date",
          "format": "yyyy-MM-dd"
```

```
          }
        }
      }
    }
  }
}'
```

Data ingestion is handled using Logstash, configured to monitor real-time changes in the file. Whenever new data is added, Logstash appends it as a document to the Elasticsearch index. The directory containing the data files is monitored using the Python script below. When a new data file is added, its content is automatically appended to the file that Logstash monitors, ensuring it is indexed as a document in Elasticsearch.

```
import os
import json
from watchdog.observers import Observer
from watchdog.events import FileSystemEventHandler

# Directory containing the input JSON files
input_directory = r"D:\Users\yara.mahfouz\Downloads\global_data"

# Path to the output JSON file
output_file_path = r"D:\Users\yara.mahfouz\Downloads\global_data\data.jsonl
    "

# Path to the processed files log
processed_log_path = r"D:\Users\yara.mahfouz\Downloads\global_data\
    processed_files.log"

# Ensure the output file and processed log exist
if not os.path.exists(output_file_path):
    with open(output_file_path, 'w') as f:
        f.write('')

if os.path.exists(processed_log_path):
    with open(processed_log_path, 'r') as log_file:
        processed_files = set(log_file.read().splitlines())
else:
    processed_files = set()

def process_file(file_path):
    file_name = os.path.basename(file_path)
    if file_name in processed_files:
        return  # Skip if already processed

    try:
        with open(file_path, 'r') as input_file:
            json_data = json.load(input_file)

            # Append each item in the JSON data to the output file
            if isinstance(json_data, list):
                with open(output_file_path, 'a') as output:
                    for item in json_data:
                        output.write(json.dumps(item) + "\n")
```

```
                print(f"Processed file: {file_name}")
                # Log the processed file
                with open(processed_log_path, 'a') as log_file:
                    log_file.write(file_name + "\n")
                processed_files.add(file_name)
            else:
                print(f"File {file_name} does not contain a list of JSON
                    objects.")
    except json.JSONDecodeError as e:
        print(f"Error decoding JSON in file {file_name}: {e}")
    except Exception as e:
        print(f"Unexpected error processing file {file_name}: {e}")

class JSONFileHandler(FileSystemEventHandler):
    def on_created(self, event):
        # Process only JSON files
        if event.is_directory or not event.src_path.endswith('.json'):
            return
        print(f"New file detected: {event.src_path}")
        process_file(event.src_path)

if __name__ == "__main__":
    # Process existing files in the directory
    for file_name in os.listdir(input_directory):
        file_path = os.path.join(input_directory, file_name)
        if os.path.isfile(file_path) and file_name.endswith('.json'):
            process_file(file_path)

    # Set up the file system observer
    event_handler = JSONFileHandler()
    observer = Observer()
    observer.schedule(event_handler, input_directory, recursive=False)

    print(f"Monitoring directory: {input_directory}")
    observer.start()

    try:
        while True:
            pass  # Keep the script running
    except KeyboardInterrupt:
        observer.stop()
    observer.join()
```

The configuration file (logstash.conf) below defines the input source, the transformations applied to the data, and the output destination.

```
input {
  file {
    path => "/mnt/d/Users/yara.mahfouz/Downloads/global_data/data.json"
    codec => json {
      target => "[document]"
    }
    start_position => "beginning"
    sincedb_path => "/dev/null"
    mode => "tail"
```

```
  }
}

filter {
  ruby {
    code => '
      months_mapping = {
        "January" => "01", "February" => "02", "March" => "03", "April" =>
            "04", "May" => "05", "June" => "06",
        "July" => "07", "August" => "08", "September" => "09", "October" =>
             "10", "November" => "11", "December" => "12"
      }

      digits_mapping = {
        "Zero" => "0", "One" => "1", "Two" => "2", "Three" => "3", "Four"
            => "4", "Five" => "5",
        "Six" => "6", "Seven" => "7", "Eight" => "8", "Nine" => "9"
      }

      # Adding the class property to annotations
      annotations = event.get("[document][annotations]")
      if annotations
        annotations.each do |annotation|
          label = annotation["label"]
          if digits_mapping.key?(label)
            annotation["class"] = "Digit"
          elsif months_mapping.key?(label)
            annotation["class"] = "Month"
          else
            annotation["class"] = "Unknown"
          end
        end
        event.set("[document][annotations]", annotations)
      end

      def extract_date(annotations, digits_mapping, months_mapping)
        date_parts = annotations.map { |ann| { "label" => ann["label"], "x"
             => ann["coordinates"]["x"] } }
        sorted_date_parts = date_parts.sort_by { |part| part["x"] }
        alpha_date_parts = sorted_date_parts.map { |part| part["label"] }

        month_index = alpha_date_parts.find_index { |part| months_mapping.
            key?(part) }
        if month_index == 2
          if alpha_date_parts[0..1].all? { |part| digits_mapping.key?(part)
              }
            alpha_date_parts.reverse!
          end
        end

        numeric_date_parts = alpha_date_parts.map do |part|
          if digits_mapping.key?(part)
            digits_mapping[part]
          elsif months_mapping.key?(part)
```

```
            months_mapping[part]
          else
            part
          end
        end

        month_index = numeric_date_parts.find_index { |part| months_mapping
            .value?(part) }
        if month_index && month_index >= 2
          year = numeric_date_parts[0...month_index].join("")
          month = numeric_date_parts[month_index]
          day = numeric_date_parts[(month_index + 1)..].join("").rjust(2,
              "0")
          return "#{year}-#{month}-#{day}"
        end

        return "Invalid Date"
      end

      if annotations
        extracted_date = extract_date(annotations, digits_mapping,
            months_mapping)
        event.set("[document][extracted_date]", extracted_date)
      end

    ,
  }
}

output {
  stdout {
    codec => rubydebug
  }
  elasticsearch {
    hosts => ["https://df44ed1e3e2c:9200"]
    user => "elastic"
    password => "tOuOKU-2wUNDBbmD=dAj"
    ssl => true
    cacert => "/usr/share/logstash/config/http_ca.crt"
    index => "labeled_images_yara_mahfouz"
    document_id => "%{[document][image]}"
  }
}
```

The input source is a JSON file containing data annotations for passport stamp images. The filter applies two key transformations:

- **Class Extraction:** Identifies and extracts the class property from the existing label according to whether it represents a digit or a month.
- **Date Extraction and Formatting:** Derives the date from the image by analyzing the bounding box positions of each date element (considering their order and alignment) and formats them into a valid date structure.

The following command is the way to view the mapping after creation.

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X GET https://
    localhost:9200/labeled_images_yara_mahfouz/_mapping?pretty
```

After applying the transformations, the documents are stored in the Elasticsearch index. The following command launches the Logstash container, mounts the input file, and configures Logstash to monitor the file for updates and store the data:

```
docker run --rm -it   -v ~/pipeline/logstash.conf:/usr/share/logstash/
    pipeline/logstash.conf   -v ~/pipeline/config/:/usr/share/logstash/
    config/   -v /mnt/d/Users/yara.mahfouz/Downloads/global_data/data.json
    :/mnt/d/Users/yara.mahfouz/Downloads/global_data/data.json   --network
    elastic   docker.elastic.co/logstash/logstash:8.16.1
```

The following command is the way to view the data after storage.

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X GET https://
    localhost:9200/labeled_images_yara_mahfouz/_search?pretty
```

3 Elasticsearch Endpoints

3.1 Retrieve all documents

Retrieves all documents stored in the index.

Request

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X GET https://
    localhost:9200/labeled_images_yara_mahfouz/_search?pretty -H "Content-
    Type: application/json" -d '{
  "_source": ["document"]
}'
```

Response

```
"hits" : [
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_94_stamp_jpg.rf.02d6ad2c17a337bbeaf2bd9970865528.jpg",
    "_score" : 1.0,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
      "document" : {
        "annotations" : [
          {
            "class" : "Digit",
            "label" : "Two",
            "coordinates" : {
              "x" : 442.6671,
              "y" : 178.5774,
              "height" : 47.1714,
```

```
        "width" : 32.8823
      }
    },
    {
      "class" : "Digit",
      "label" : "Two",
      "coordinates" : {
        "x" : 392.8623,
        "y" : 182.5083,
        "height" : 42.5184,
        "width" : 31.4388
      }
    },
    {
      "class" : "Month",
      "label" : "June",
      "coordinates" : {
        "x" : 309.5948,
        "y" : 186.4393,
        "height" : 52.9475,
        "width" : 103.6197
      }
    },
    {
      "class" : "Digit",
      "label" : "Four",
      "coordinates" : {
        "x" : 216.205,
        "y" : 192.0549,
        "height" : 50.0594,
        "width" : 45.875
      }
    },
    {
      "class" : "Digit",
      "label" : "Two",
      "coordinates" : {
        "x" : 182.5206,
        "y" : 197.3497,
        "height" : 47.1714,
        "width" : 43.9501
      }
    },
    {
      "class" : "Digit",
      "label" : "Zero",
      "coordinates" : {
        "x" : 160.0643,
        "y" : 193.9803,
        "height" : 19.2536,
        "width" : 26.6267
      }
    },
    {
```

```
            "class" : "Digit",
            "label" : "Two",
            "coordinates" : {
              "x" : 134.5604,
              "y" : 198.8739,
              "height" : 45.4065,
              "width" : 42.3461
            }
          }
        ],
        "extracted_date" : "2024-06-22",
        "image" : "cropped_94_stamp_jpg.rf.02d6ad2c17a337bbeaf2bd9970865528
           .jpg"
      }
    }
  }.......
```

3.2 Search for documents by image title

Searches for documents using all or part of image title. The custom tokenizer allows partial searches based on meaningful segments, like an image ID embedded in the title.

Request

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X GET https://
    localhost:9200/labeled_images_yara_mahfouz/_search?pretty -H "Content-
    Type: application/json" -d '
{
  "_source": ["document"],
  "query": {
    "match": {
      "document.image": "98"
    }
  }
}'
```

Response

```
"hits" : [
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_98_stamp_jpg.rf.0999826b7e768060c006119ef1e8d6b6.jpg",
    "_score" : 2.0794415,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
      "document" : {
        "extracted_date" : "2024-10-01",
        "image" : "cropped_98_stamp_jpg.rf.0999826b7e768060c006119ef1e8d6b6
           .jpg",
        "annotations" : [
          {
            "label" : "One",
```

```
      "class" : "Digit",
      "coordinates" : {
        "y" : 194.4744,
        "x" : 402.5231,
        "height" : 62.4918,
        "width" : 21.8991
      }
    },
    {
      "label" : "October",
      "class" : "Month",
      "coordinates" : {
        "y" : 191.8498,
        "x" : 313.1815,
        "height" : 65.7414,
        "width" : 108.7904
      }
    },
    {
      "label" : "Four",
      "class" : "Digit",
      "coordinates" : {
        "y" : 190.6,
        "x" : 226.6745,
        "height" : 66.7412,
        "width" : 33.2363
      }
    },
    {
      "label" : "Two",
      "class" : "Digit",
      "coordinates" : {
        "y" : 191.4748,
        "x" : 202.8093,
        "height" : 63.9916,
        "width" : 29.4879
      }
    },
    {
      "label" : "Zero",
      "class" : "Digit",
      "coordinates" : {
        "y" : 189.3501,
        "x" : 178.5693,
        "height" : 28.7462,
        "width" : 26.9889
      }
    },
    {
      "label" : "Two",
      "class" : "Digit",
      "coordinates" : {
        "y" : 191.9123,
        "x" : 156.5116,
```

```
            "height" : 63.6166,
            "width" : 31.3104
          }
        }
      ]
    }
  }
]
```

3.3 Search for documents containing a specific label

Request

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X GET https://
    localhost:9200/labeled_images_yara_mahfouz/_search?pretty -H "Content-
    Type: application/json" -d '{
  "_source": ["document"],
  "query": {
    "nested": {
      "path": "document.annotations",
      "query": {
        "match": {
          "document.annotations.label": "July"
        }
      }
    }
  }
}'
```

Response

```
"hits" : [
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_87_stamp_jpg.rf.beb396ed0fc843c8093228582f1cff9c.jpg",
    "_score" : 3.9252682,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
      "document" : {
        "annotations" : [
          {
            "class" : "Digit",
            "label" : "Six",
            "coordinates" : {
              "x" : 1105.042,
              "y" : 516.7615,
              "height" : 153.5426,
              "width" : 90.7726
            }
          },
          {
```

```
      "class" : "Digit",
      "label" : "Zero",
      "coordinates" : {
        "x" : 1004.4935,
        "y" : 522.9528,
        "height" : 58.6103,
        "width" : 59.1616
      }
    },
    {
      "class" : "Month",
      "label" : "july",
      "coordinates" : {
        "x" : 796.0026,
        "y" : 554.3217,
        "height" : 197.294,
        "width" : 248.3867
      }
    },
    {
      "class" : "Digit",
      "label" : "Two",
      "coordinates" : {
        "x" : 518.6981,
        "y" : 561.4044,
        "height" : 173.6464,
        "width" : 83.5985
      }
    },
    {
      "class" : "Digit",
      "label" : "Two",
      "coordinates" : {
        "x" : 401.383,
        "y" : 572.0801,
        "height" : 172.3929,
        "width" : 83.52
      }
    },
    {
      "class" : "Digit",
      "label" : "Zero",
      "coordinates" : {
        "x" : 464.2801,
        "y" : 563.9602,
        "height" : 77.8842,
        "width" : 77.1909
      }
    },
    {
      "class" : "Digit",
      "label" : "Four",
      "coordinates" : {
        "x" : 594.3063,
```

```
              "y" : 557.4073,
              "height" : 168.8344,
              "width" : 85.9543
            }
          }
        ],
        "extracted_date" : "2024-07-06",
        "image" : "cropped_87_stamp_jpg.rf.beb396ed0fc843c8093228582f1cff9c
            .jpg"
      }
    }
  }
]
```


3.4 Retrieve Only Specific Fields

Retrieve only the image field and its extracted date for performance optimization.

Request

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X GET https://
    localhost:9200/labeled_images_yara_mahfouz/_search?pretty -H "Content-
    Type: application/json" -d '{
  "_source": ["document.image", "document.extracted_date"]
}'
```

Response


```
"hits" : [
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_94_stamp_jpg.rf.02d6ad2c17a337bbeaf2bd9970865528.jpg",
    "_score" : 1.0,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
      "document" : {
        "extracted_date" : "2024-06-22",
        "image" : "cropped_94_stamp_jpg.rf.02d6ad2c17a337bbeaf2bd9970865528
            .jpg"
      }
    }
  }.......
```


3.5 Count annotations by label

Counts the number of annotations for each label.

Request

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X GET https://
    localhost:9200/labeled_images_yara_mahfouz/_search?pretty -H "Content-
    Type: application/json" -d '{
  "size": 0,
  "aggs": {
    "nested_annotations": {
      "nested": {
        "path": "document.annotations"
      },
      "aggs": {
        "annotations_per_label": {
          "terms": {
            "field": "document.annotations.label"
          }
        }
      }
    }
  }
}'
```

Response

```
"aggregations" : {
    "nested_annotations" : {
      "doc_count" : 75,
      "annotations_per_label" : {
        "doc_count_error_upper_bound" : 0,
        "sum_other_doc_count" : 6,
        "buckets" : [
          {
            "key" : "two",
            "doc_count" : 30
          },
          {
            "key" : "zero",
            "doc_count" : 12
          },
          {
            "key" : "four",
            "doc_count" : 8
          },
          {
            "key" : "one",
            "doc_count" : 7
          },
          {
            "key" : "three",
            "doc_count" : 4
          },
          {
            "key" : "october",
            "doc_count" : 3
          },
          {
            "key" : "march",
```

```
            "doc_count" : 2
          },
          {
            "key" : "april",
            "doc_count" : 1
          },
          {
            "key" : "february",
            "doc_count" : 1
          },
          {
            "key" : "five",
            "doc_count" : 1
          }
        ]
      }
    }
}
```

3.6 Search for images containing annotations in a specific area

Searches for images containing annotations within a defined range and returns only the matching annotations.

Request

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X GET https://
    localhost:9200/labeled_images_yara_mahfouz/_search?pretty -H "Content-
    Type: application/json" -d '{
  "_source": ["document.image"],
  "query": {
    "nested": {
      "path": "document.annotations",
      "query": {
        "bool": {
          "must": [
            { "range": { "document.annotations.coordinates.x": { "gte": 50,
                "lte": 100 } } },
            { "range": { "document.annotations.coordinates.y": { "gte": 50,
                "lte": 100 } } }
          ]
        }
      },
      "inner_hits": {}
    }
  }
}'
```

Response

```
"hits" : [
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_97_stamp_jpg.rf.4ddb4166bc0785d64d768715c6473867.jpg",
```

```
    "_score" : 2.0,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
      "document" : {
        "image" : "cropped_97_stamp_jpg.rf.4ddb4166bc0785d64d768715c6473867
            .jpg"
      }
    },
    "inner_hits" : {
      "document.annotations" : {
        "hits" : {
          "total" : {
            "value" : 2,
            "relation" : "eq"
          },
          "max_score" : 2.0,
          "hits" : [
            {
              "_index" : "labeled_images_yara_mahfouz",
              "_id" : "cropped_97_stamp_jpg.rf.4
                  ddb4166bc0785d64d768715c6473867.jpg",
              "_nested" : {
                "field" : "document.annotations",
                "offset" : 5
              },
              "_score" : 2.0,
              "_source" : {
                "class" : "Digit",
                "label" : "One",
                "coordinates" : {
                  "x" : 97.5248,
                  "y" : 89.4168,
                  "height" : 32.23,
                  "width" : 9.4658
                }
              }
            },
            {
              "_index" : "labeled_images_yara_mahfouz",
              "_id" : "cropped_97_stamp_jpg.rf.4
                  ddb4166bc0785d64d768715c6473867.jpg",
              "_nested" : {
                "field" : "document.annotations",
                "offset" : 6
              },
              "_score" : 2.0,
              "_source" : {
                "class" : "Digit",
                "label" : "Four",
                "coordinates" : {
                  "x" : 79.541,
                  "y" : 85.1384,
```

```
                "height" : 32.8004,
                "width" : 16.5382
              }
            }
          }
        ]
      }
    }
  }
]
```

3.7 Search for images sorted by extracted date

Searches for specific images sorted by extracted date from earliest to latest.

Request

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X GET https://
    localhost:9200/labeled_images_yara_mahfouz/_search?pretty -H "Content-
    Type: application/json" -d '{
  "_source": ["document.image", "document.extracted_date"],
  "query": {
    "nested": {
      "path": "document.annotations",
      "query": {
        "match": { "document.annotations.label": "october" }
      }
    }
  },
  "sort": [
    { "document.extracted_date": { "order": "desc" } }
  ]
}'
```

Response

```
"hits" : [
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_98_stamp_jpg.rf.0999826b7e768060c006119ef1e8d6b6.jpg",
    "_score" : null,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
      "document" : {
        "extracted_date" : "2024-10-01",
        "image" : "cropped_98_stamp_jpg.rf.0999826b7e768060c006119ef1e8d6b6
            .jpg"
      }
    },
    "sort" : [
      1727740800000
```

```
    ]
  },
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_97_stamp_jpg.rf.4ddb4166bc0785d64d768715c6473867.jpg",
    "_score" : null,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
      "document" : {
        "extracted_date" : "2022-10-14",
        "image" : "cropped_97_stamp_jpg.rf.4ddb4166bc0785d64d768715c6473867
            .jpg"
      }
    },
    "sort" : [
      1665705600000
    ]
  },
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_90_stamp_jpg.rf.5839bc9c7ec43fcd0234ca0f03c5664a.jpg",
    "_score" : null,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
      "document" : {
        "extracted_date" : "2022-10-14",
        "image" : "cropped_90_stamp_jpg.rf.5839bc9c7ec43fcd0234ca0f03c5664a
            .jpg"
      }
    },
    "sort" : [
      1665705600000
    ]
  }
]
```

3.8 Count annotations by month

Count the number of annotations for each month.

Request

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X GET https://
    localhost:9200/labeled_images_yara_mahfouz/_search?pretty -H "Content-
    Type: application/json" -d '{
  "size": 0,
  "aggs": {
    "annotations_per_month": {
      "nested": {
```

```
        "path": "document.annotations"
      },
      "aggs": {
        "month_filter": {
          "filter": {
            "term": { "document.annotations.class": "month" }
          },
          "aggs": {
            "month_count": {
              "terms": {
                "field": "document.annotations.label",
                "size": 12
              }
            }
          }
        }
      }
    }
  }
}'
```

Response

```
"aggregations" : {
    "annotations_per_month" : {
      "doc_count" : 75,
      "month_filter" : {
        "doc_count" : 11,
        "month_count" : {
          "doc_count_error_upper_bound" : 0,
          "sum_other_doc_count" : 0,
          "buckets" : [
            {
              "key" : "october",
              "doc_count" : 3
            },
            {
              "key" : "march",
              "doc_count" : 2
            },
            {
              "key" : "april",
              "doc_count" : 1
            },
            {
              "key" : "february",
              "doc_count" : 1
            },
            {
              "key" : "january",
              "doc_count" : 1
            },
            {
              "key" : "july",
              "doc_count" : 1
```

```
            },
            {
              "key" : "june",
              "doc_count" : 1
            },
            {
              "key" : "november",
              "doc_count" : 1
            }
          ]
        }
      }
    }
}
```

3.9 Find images containing a specific date

Find images with an extracted date of 2024-03-31 for example.

Request

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X GET https://
    localhost:9200/labeled_images_yara_mahfouz/_search?pretty -H "Content-
    Type: application/json" -d '{
  "_source": ["document.image"],
  "query": {
    "term": { "document.extracted_date": "2024-03-31" }
  }
}'
```

Response

```
"hits" : [
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_103_stamp_jpg.rf.91f0c7a887510a248a8ad02e3ce73308.jpg"
       ,
    "_score" : 1.0,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
      "document" : {
        "image" : "cropped_103_stamp_jpg.rf.91
           f0c7a887510a248a8ad02e3ce73308.jpg"
      }
    }
  }
]
```

3.10 Search for images withing a date range

Search for images within the first half of 2024, for example.

Request

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X GET https://
    localhost:9200/labeled_images_yara_mahfouz/_search?pretty -H "Content-
    Type: application/json" -d '{
  "_source": ["document.image", "document.extracted_date"],
  "query": {
    "range": {
      "document.extracted_date": {
        "gte": "2024-01-01",
        "lte": "2024-06-31"
      }
    }
  }
}'
```

Response

```
"hits" : [
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_94_stamp_jpg.rf.02d6ad2c17a337bbeaf2bd9970865528.jpg",
    "_score" : 1.0,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
      "document" : {
        "extracted_date" : "2024-06-22",
        "image" : "cropped_94_stamp_jpg.rf.02d6ad2c17a337bbeaf2bd9970865528
            .jpg"
      }
    }
  },
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_103_stamp_jpg.rf.91f0c7a887510a248a8ad02e3ce73308.jpg"
        ,
    "_score" : 1.0,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
      "document" : {
        "extracted_date" : "2024-03-31",
        "image" : "cropped_103_stamp_jpg.rf.91
            f0c7a887510a248a8ad02e3ce73308.jpg"
      }
    }
  },
  {
```

```
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_95_stamp_jpg.rf.8c19f17aa88786ff8efaed4609cb94fd.jpg",
    "_score" : 1.0,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
      "document" : {
        "extracted_date" : "2024-03-07",
        "image" : "cropped_95_stamp_jpg.rf.8c19f17aa88786ff8efaed4609cb94fd
            .jpg"
      }
    }
  }
]
```

3.11 Count images extracted per year

Count the number of images for each year of the extracted date.

Request

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X GET https://
    localhost:9200/labeled_images_yara_mahfouz/_search?pretty -H "Content-
    Type: application/json" -d '{
  "size": 0,
  "aggs": {
    "images_per_year": {
      "date_histogram": {
        "field": "document.extracted_date",
        "calendar_interval": "year"
      }
    }
  }
}'
```

Response

```
"aggregations" : {
    "images_per_year" : {
      "buckets" : [
        {
          "key_as_string" : "2022-01-01",
          "key" : 1640995200000,
          "doc_count" : 3
        },
        {
          "key_as_string" : "2023-01-01",
          "key" : 1672531200000,
          "doc_count" : 3
        },
        {
          "key_as_string" : "2024-01-01",
          "key" : 1704067200000,
```

```
          "doc_count" : 5
        }
      ]
    }
}
```

3.12 Retrieve images with weekend dates

Filters images based on whether the extracted date corresponds to Friday (5) or Saturday (6)

Request

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X GET https://
    localhost:9200/labeled_images_yara_mahfouz/_search?pretty -H "Content-
    Type: application/json" -d '{
  "_source": ["document.image", "document.extracted_date"],
  "query": {
    "script": {
      "script": {
        "source": "def day = doc[\"document.extracted_date\"].value.
            getDayOfWeek().getValue(); day == 5 || day == 6",
        "lang": "painless"
      }
    }
  }
}'
```

Response

```
"hits" : [
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_94_stamp_jpg.rf.02d6ad2c17a337bbeaf2bd9970865528.jpg",
    "_score" : 1.0,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
      "document" : {
        "extracted_date" : "2024-06-22",
        "image" : "cropped_94_stamp_jpg.rf.02d6ad2c17a337bbeaf2bd9970865528
            .jpg"
      }
    }
  },
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_87_stamp_jpg.rf.beb396ed0fc843c8093228582f1cff9c.jpg",
    "_score" : 1.0,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
```

```
      "document" : {
        "extracted_date" : "2024-07-06",
        "image" : "cropped_87_stamp_jpg.rf.beb396ed0fc843c8093228582f1cff9c
            .jpg"
      }
    }
  },
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_97_stamp_jpg.rf.4ddb4166bc0785d64d768715c6473867.jpg",
    "_score" : 1.0,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
      "document" : {
        "extracted_date" : "2022-10-14",
        "image" : "cropped_97_stamp_jpg.rf.4ddb4166bc0785d64d768715c6473867
            .jpg"
      }
    }
  },
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_90_stamp_jpg.rf.5839bc9c7ec43fcd0234ca0f03c5664a.jpg",
    "_score" : 1.0,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
      "document" : {
        "extracted_date" : "2022-10-14",
        "image" : "cropped_90_stamp_jpg.rf.5839bc9c7ec43fcd0234ca0f03c5664a
            .jpg"
      }
    }
  }
]
```


3.13 Retrieve images closest to a Specific date

Sort images by proximity to a specific date, like getting

Request

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X GET https://
    localhost:9200/labeled_images_yara_mahfouz/_search?pretty -H "Content-
    Type: application/json" -d '{
  "_source": ["document.image", "document.extracted_date"],
  "query": {
    "range": {
      "document.extracted_date": {
        "lte": "2023-01-01"
```

```
      }
    }
  },
  "sort": [
    { "document.extracted_date": { "order": "desc" } }
  ]
}'
```

Response


```
"hits" : [
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_97_stamp_jpg.rf.4ddb4166bc0785d64d768715c6473867.jpg",
    "_score" : null,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
      "document" : {
        "extracted_date" : "2022-10-14",
        "image" : "cropped_97_stamp_jpg.rf.4ddb4166bc0785d64d768715c6473867
            .jpg"
      }
    },
    "sort" : [
      1665705600000
    ]
  },
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_90_stamp_jpg.rf.5839bc9c7ec43fcd0234ca0f03c5664a.jpg",
    "_score" : null,
    "_ignored" : [
      "event.original.keyword"
    ],
    "_source" : {
      "document" : {
        "extracted_date" : "2022-10-14",
        "image" : "cropped_90_stamp_jpg.rf.5839bc9c7ec43fcd0234ca0f03c5664a
            .jpg"
      }
    },
    "sort" : [
      1665705600000
    ]
  },
  {
    "_index" : "labeled_images_yara_mahfouz",
    "_id" : "cropped_100_stamp_jpg.rf.bbda0baef032f0e0f0efdcc05812240c.jpg"
        ,
    "_score" : null,
    "_ignored" : [
      "event.original.keyword"
    ],
```

```
    "_source" : {
      "document" : {
        "extracted_date" : "2022-04-12",
        "image" : "cropped_100_stamp_jpg.rf.
            bbda0baef032f0e0f0efdcc05812240c.jpg"
      }
    },
    "sort" : [
      1649721600000
    ]
  }
]
```

3.14 Count the number of images for each month

Count the number of images for each month of the extracted date.

Request

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X GET https://
    localhost:9200/labeled_images_yara_mahfouz/_search?pretty -H "Content-
    Type: application/json" -d '{
  "size": 0,
  "aggs": {
    "images_per_month": {
      "date_histogram": {
        "field": "document.extracted_date",
        "calendar_interval": "month"
      }
    }
  }
}'
```

Response

```
"aggregations" : {
    "images_per_month" : {
      "buckets" : [
        {
          "key_as_string" : "2022-04-01",
          "key" : 1648771200000,
          "doc_count" : 1
        },
        {
          "key_as_string" : "2022-05-01",
          "key" : 1651363200000,
          "doc_count" : 0
        },
        {
          "key_as_string" : "2022-06-01",
          "key" : 1654041600000,
          "doc_count" : 0
        },
        {
```

```
        "key_as_string" : "2022-07-01",
        "key" : 1656633600000,
        "doc_count" : 0
      },
      {
        "key_as_string" : "2022-08-01",
        "key" : 1659312000000,
        "doc_count" : 0
      },
      {
        "key_as_string" : "2022-09-01",
        "key" : 1661990400000,
        "doc_count" : 0
      },
      {
        "key_as_string" : "2022-10-01",
        "key" : 1664582400000,
        "doc_count" : 2
      },
      {
        "key_as_string" : "2022-11-01",
        "key" : 1667260800000,
        "doc_count" : 0
      },
      {
        "key_as_string" : "2022-12-01",
        "key" : 1669852800000,
        "doc_count" : 0
      },
      {
        "key_as_string" : "2023-01-01",
        "key" : 1672531200000,
        "doc_count" : 1
      },
      {
        "key_as_string" : "2023-02-01",
        "key" : 1675209600000,
        "doc_count" : 1
      },
      {
        "key_as_string" : "2023-03-01",
        "key" : 1677628800000,
        "doc_count" : 0
      },
      {
        "key_as_string" : "2023-04-01",
        "key" : 1680307200000,
        "doc_count" : 0
      },
      {
        "key_as_string" : "2023-05-01",
        "key" : 1682899200000,
        "doc_count" : 0
      },
```

```
        {
          "key_as_string" : "2023-06-01",
          "key" : 1685577600000,
          "doc_count" : 0
        },
        {
          "key_as_string" : "2023-07-01",
          "key" : 1688169600000,
          "doc_count" : 0
        },
        {
          "key_as_string" : "2023-08-01",
          "key" : 1690848000000,
          "doc_count" : 0
        },
        {
          "key_as_string" : "2023-09-01",
          "key" : 1693526400000,
          "doc_count" : 0
        },
        {
          "key_as_string" : "2023-10-01",
          "key" : 1696118400000,
          "doc_count" : 0
        },
        {
          "key_as_string" : "2023-11-01",
          "key" : 1698796800000,
          "doc_count" : 1
        },
        {
          "key_as_string" : "2023-12-01",
          "key" : 1701388800000,
          "doc_count" : 0
        },
        {
          "key_as_string" : "2024-01-01",
          "key" : 1704067200000,
          "doc_count" : 0
        },
        {
          "key_as_string" : "2024-02-01",
          "key" : 1706745600000,
          "doc_count" : 0
        },
        {
          "key_as_string" : "2024-03-01",
          "key" : 1709251200000,
          "doc_count" : 2
        },
        {
          "key_as_string" : "2024-04-01",
          "key" : 1711929600000,
          "doc_count" : 0
```

```
        },
        {
          "key_as_string" : "2024-05-01",
          "key" : 1714521600000,
          "doc_count" : 0
        },
        {
          "key_as_string" : "2024-06-01",
          "key" : 1717200000000,
          "doc_count" : 1
        },
        {
          "key_as_string" : "2024-07-01",
          "key" : 1719792000000,
          "doc_count" : 1
        },
        {
          "key_as_string" : "2024-08-01",
          "key" : 1722470400000,
          "doc_count" : 0
        },
        {
          "key_as_string" : "2024-09-01",
          "key" : 1725148800000,
          "doc_count" : 0
        },
        {
          "key_as_string" : "2024-10-01",
          "key" : 1727740800000,
          "doc_count" : 1
        }
      ]
    }
}
```

4 Dashboard

4.1 Indexing

To create visualizations on kibana, flattening the index was necessary. Below is the mapping for the new flattened index.

```
curl --cacert http_ca.crt -u elastic:$ELASTIC_PASSWORD -X PUT https://
    localhost:9200/labeled_images_yara_mahfouz_flattened -H "Content-Type:
    application/json" -d '{
  "settings": {
    "analysis": {
      "normalizer": {
        "lowercase_normalizer": {
          "type": "custom",
          "filter": ["lowercase"]
        }
```

```
      }
    }
  },
  "mappings": {
    "properties": {
      "annotation_class": {
        "type": "keyword",
        "normalizer": "lowercase_normalizer"
      },
      "annotation_coordinates_height": {
        "type": "float"
      },
      "annotation_coordinates_width": {
        "type": "float"
      },
      "annotation_coordinates_x": {
        "type": "float"
      },
      "annotation_coordinates_y": {
        "type": "float"
      },
      "annotation_label": {
        "type": "keyword",
        "normalizer": "lowercase_normalizer"
      },
      "extracted_date": {
        "type": "date",
        "format": "yyyy-MM-dd"
      },
      "image": {
        "type": "keyword",
        "normalizer": "lowercase_normalizer"
      }
    }
  }
}'
```

The configuration file defines a Logstash pipeline that reads data from the original Elasticsearch index, splits nested annotations into individual documents, flattens fields like class and coordinates into a simpler structure, and sends the transformed data to the flattened index. It uses HTTPS connections with SSL certificate verification disabled for simplicity and outputs processed data to both Elasticsearch and the console for debugging.

```
input {
  elasticsearch {
    hosts => ["https://df44ed1e3e2c:9200"]
    index => "labeled_images_yara_mahfouz"
    user => "elastic"
    password => "tOuOKU-2wUNDBbmD=dAj"
    ssl => true  # Use SSL
    ssl_certificate_verification => false
    schedule => "* * * * *"  # Poll every minute
  }
}
```

```
filter {
  split {
    field => "[document][annotations]"
  }
  mutate {
    add_field => {
      "annotation_class" => "%{[document][annotations][class]}"
      "annotation_label" => "%{[document][annotations][label]}"
      "annotation_coordinates_x" => "%{[document][annotations][coordinates
          ][x]}"
      "annotation_coordinates_y" => "%{[document][annotations][coordinates
          ][y]}"
      "annotation_coordinates_width" => "%{[document][annotations][
          coordinates][width]}"
      "annotation_coordinates_height" => "%{[document][annotations][
          coordinates][height]}"
      "image" => "%{[document][image]}"
      "extracted_date" => "%{[document][extracted_date]}"
    }
    # Generate a unique ID using image name, x, and y coordinates
    add_field => {
      "doc_id" => "%{[document][image]}_%{[document][annotations][
          coordinates][x]}_%{[document][annotations][coordinates][y]}"
    }
    # Remove unnecessary fields
    remove_field => ["document", "[document][annotations]"]
  }
}

output {
  elasticsearch {
    hosts => ["https://df44ed1e3e2c:9200"]
    index => "labeled_images_yara_mahfouz_flattened"
    user => "elastic"
    password => "tOuOKU-2wUNDBbmD=dAj"
    ssl => true
    ssl_certificate_verification => false
    document_id => "%{doc_id}"  # Use the generated unique ID as the
        Elasticsearch document ID
  }
  stdout {
    codec => rubydebug
  }
}
```

The Docker command runs a Logstash container with the specified pipeline configuration

```
docker run --rm -it   -v ~/pipeline/elastic.conf:/usr/share/logstash/
    pipeline/elastic.conf   -v ~/pipeline/config/:/usr/share/logstash/
    config/   --network elastic   docker.elastic.co/logstash/logstash
    :8.16.1
```

4.2 Visualizations

This dashboard offers an overview of the annotated image data, presenting trends, distributions, and key metrics. It visualizes entry counts, label frequencies, and class proportions while highlighting patterns over time and across weekdays. The dashboard provides a clear summary of the data, helping to identify general trends and insights at a glance.

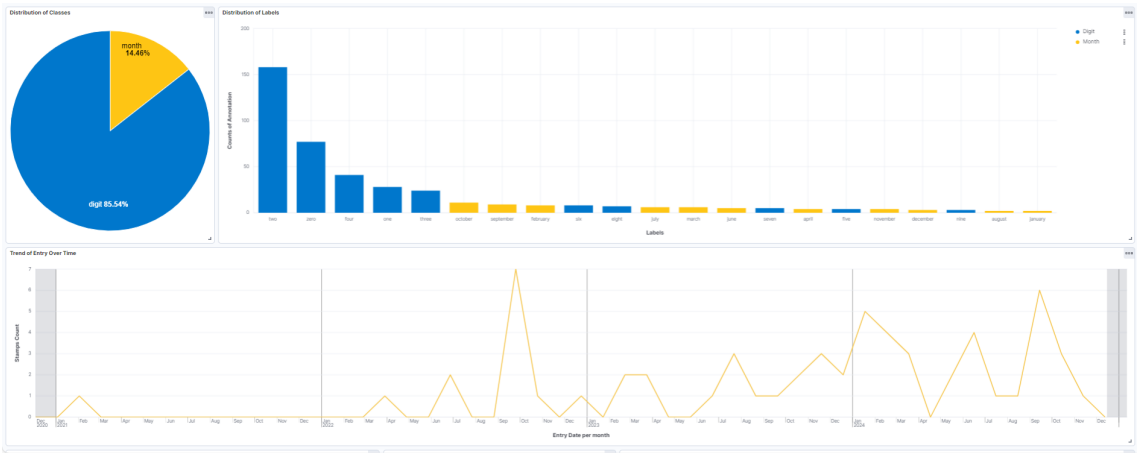

Figure 1: First Half of The Dahsboard.

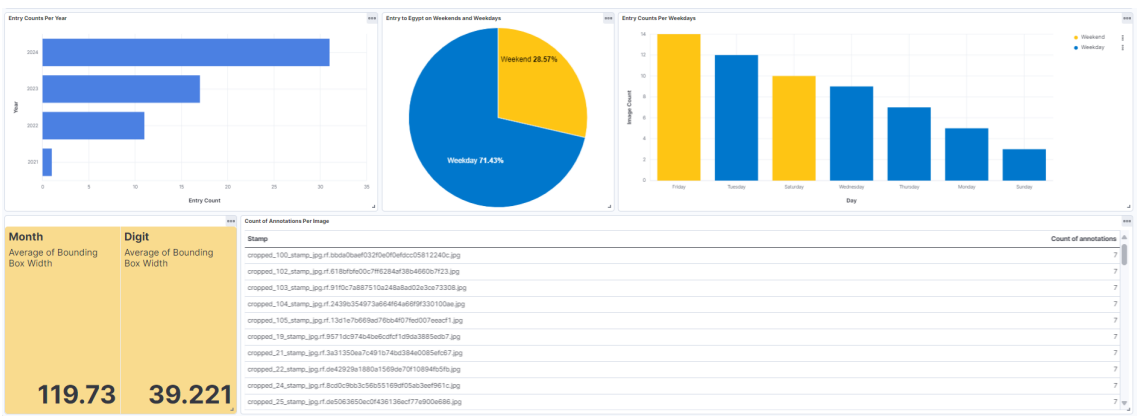

Figure 2: Second Half of The Dahsboard.

Distribution of Labels

The count of annotations for different labels categorized as "Digit" and "Month."

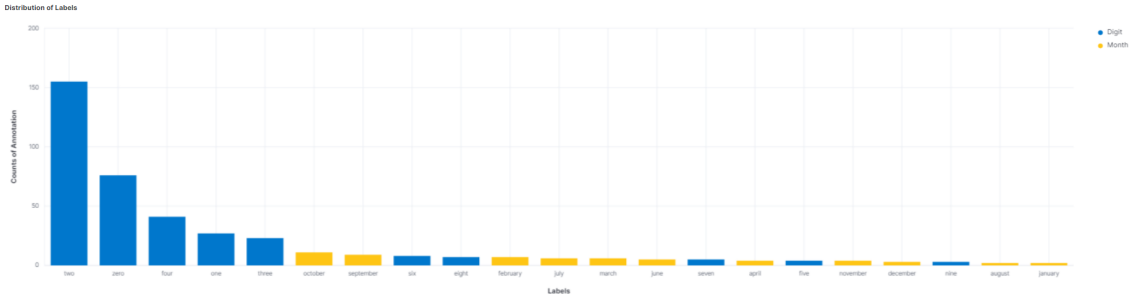

Figure 3: Distribution of Labels visualization.

"Digit" labels dominate, with "two" being the most frequent. "Month" labels are less common, with October and September leading. Useful for understanding distribution of annotations in data.

Distribution of Classes

Proportions of annotations categorized into "Digit" and "Month."

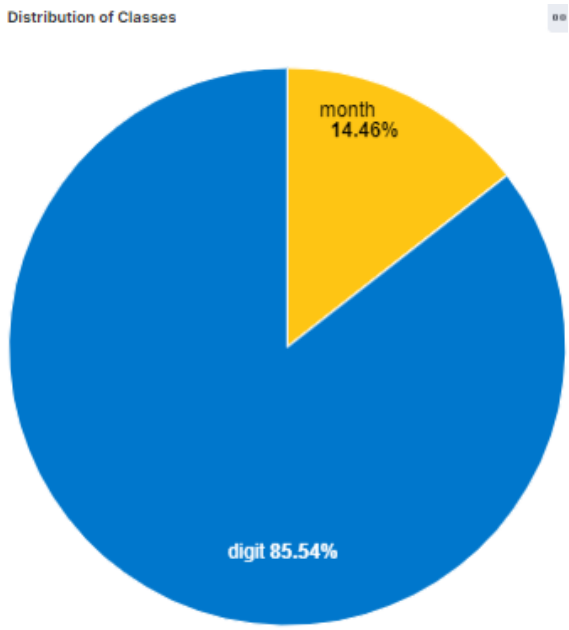

Figure 4: Distribution of Classes visualization.

The majority (85.54%) of annotations belong to the "Digit" class, while "Month" accounts for 14.46%. Useful for understanding distribution of annotations in data.

Trend of Entry Over Time

Tracks entry counts per month over several years.

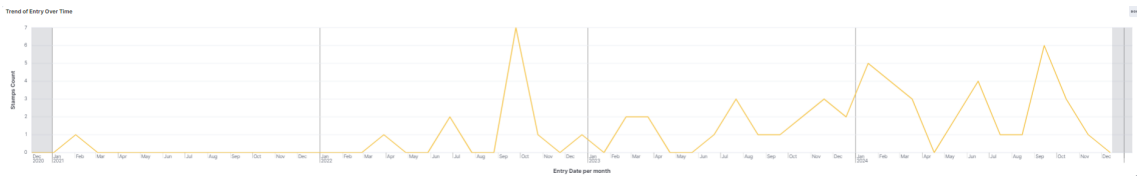

Figure 5: Trend of Entry Over Time visualization.

Peaks in activity are visible in certain months, such as October 2022 and February 2024. Useful for understanding patterns of visiting egypt over time, intervals can be customized in the dashboard.

Entry Counts Per Year

The number of entries recorded for each year.

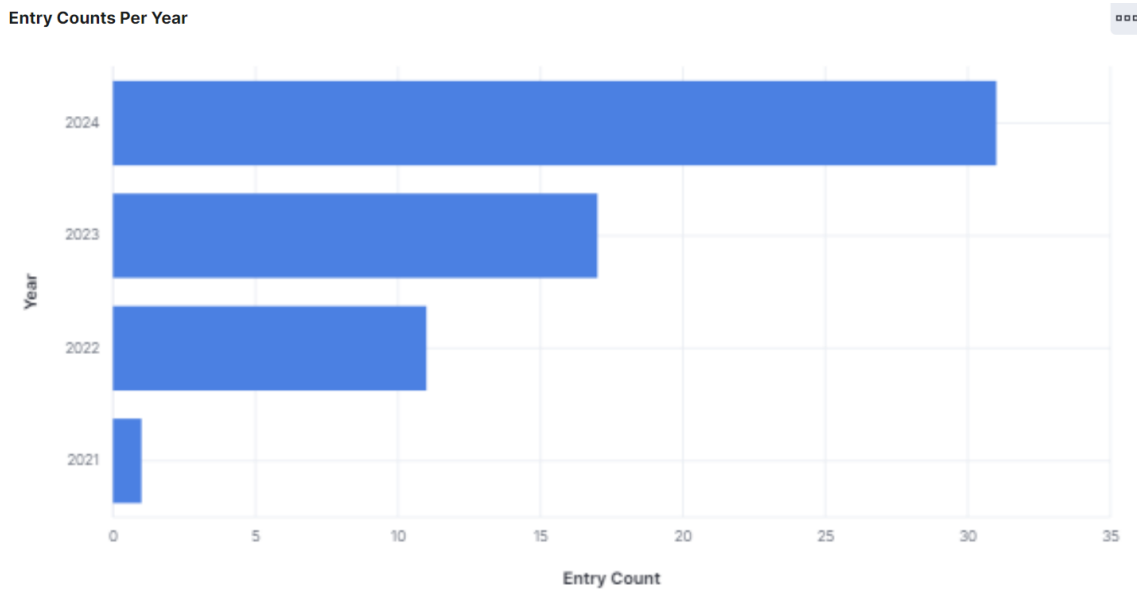

Figure 6: Entry Counts Per Year visualization.

Entries increase over time, with the highest count in 2024, indicating growing data volume or activity. Useful for understanding yearly patterns of traveling to Egypt.

Entry to Egypt on Weekends and Weekdays
Percentage of entries made on weekdays versus weekends.

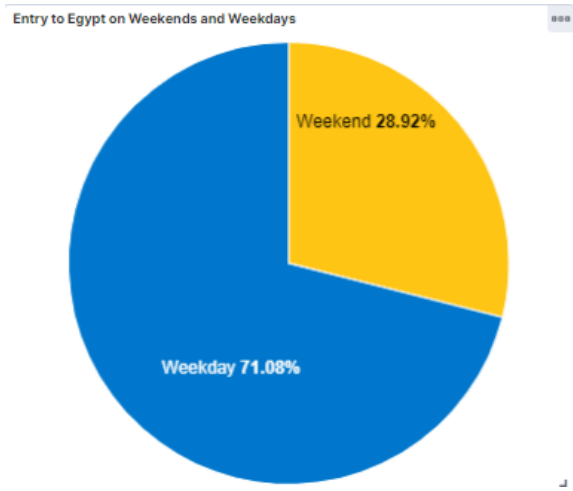

Figure 7: Entry to Egypt on Weekends and Weekdays visualization.

Most entries occur on weekdays (71.08%), while weekends account for 28.92%. Useful for understanding traveling patterns for Weekends vs. Weekdays.

Entry Counts Per Weekdays
Displays the count of entries per day of the week, split into weekdays and weekends.

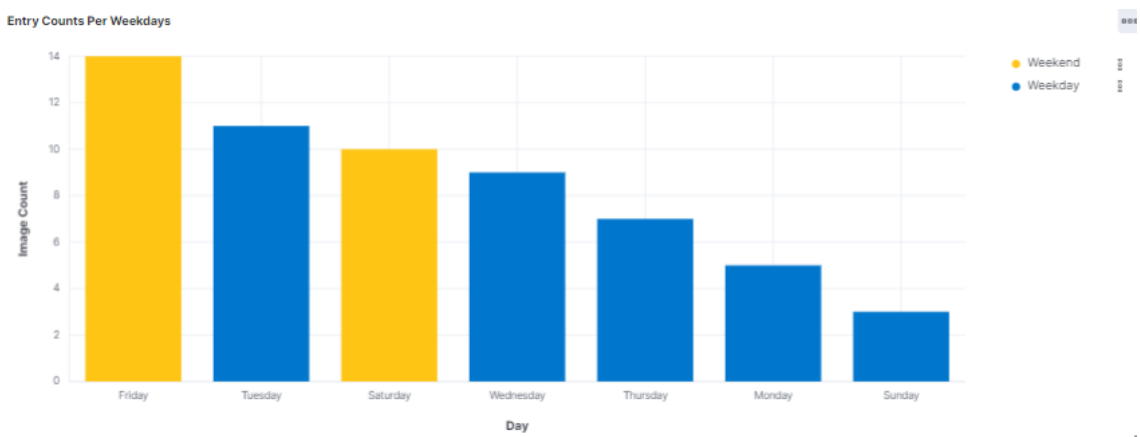

Figure 8: Entry Counts Per Weekdays visualization.

Entries are highest on Fridays (weekend) and Tuesdays (weekday). Useful for understanding traveling patterns for weekdays.

Average of Bounding Box Width
Compares the average width of bounding boxes for labels categorized as "Month" and "Digit".

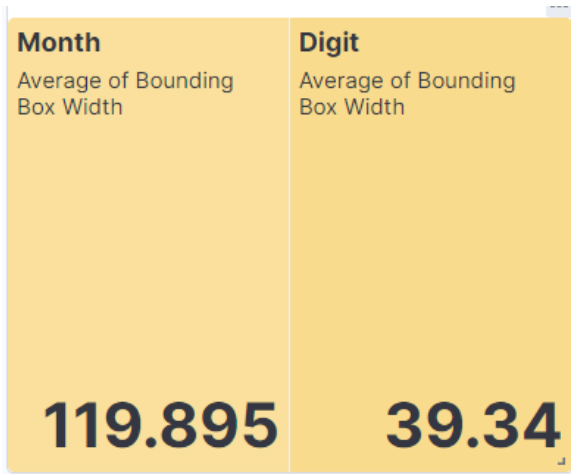

Figure 9: Average of Bounding Box Width visualization.

Bounding boxes for "Month" labels are wider on average (119.89) compared to "Digit" labels (39.34), indicating larger visual areas for "Month" entities. This can be useful for deriving the type of class when it is unknown due to technical issues or some other matters.

Count of Annotations Per Stamp
Counts the number of bounding boxes for each stamp image.

Count of Annotations Per Image

Stamp	Count of annotations
cropped_93_stamp_jpg.rf.d55840c6899e28f73a6a09116663d7a9.jpg	7
cropped_94_stamp_jpg.rf.02d6ad2c17a337bbeaf2bd9970865528.jpg	7
cropped_96_stamp_jpg.rf.e37dfbca6174c41c9218e6bf3fcf06d0.jpg	7
cropped_97_stamp_jpg.rf.4ddb4166bc0785d64d768715c6473867.jpg	7
cropped_99_stamp_jpg.rf.7ea8d1d5d94609d16b5e6d3542a43cd6.jpg	7
cropped_34_stamp_jpg.rf.60ae3b7067d277a07f78984a1e672d9b.jpg	6
cropped_53_stamp_jpg.rf.314caed4be1f59656a94251859acf5cd.jpg	6
cropped_72_stamp_jpg.rf.83796156c1f712a13af6ae1dd682724d.jpg	6
cropped_95_stamp_jpg.rf.8c19f17aa88786ff8efaed4609cb94fd.jpg	6
cropped_98_stamp_jpg.rf.0999826b7e768060c006119ef1e8d6b6.jpg	6

Figure 10: Count of Annotations Per Stamp visualization.

Some dates contain 7 boxes and some contain 6 boxes. This is because some of the days in the date is a one-digit day while others are two digits. This could have some technical value depending on what the data will be used for.