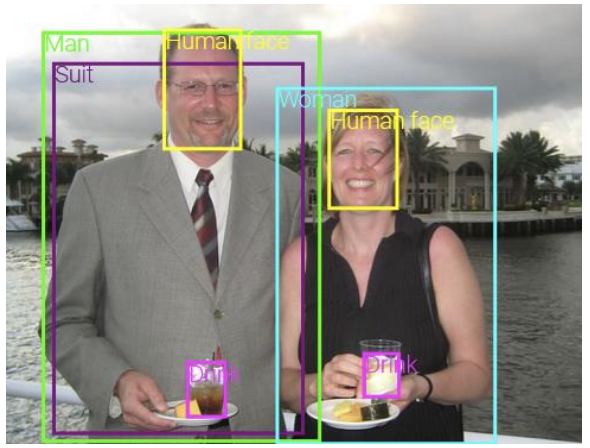
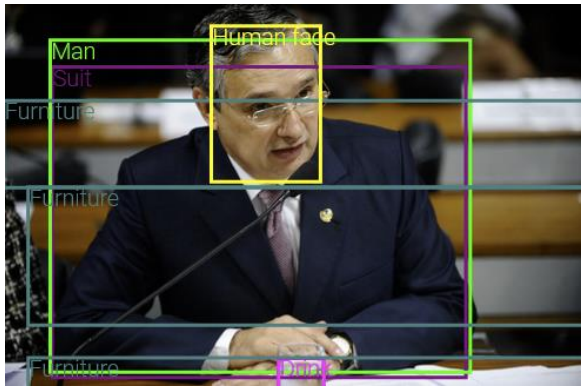
PROJECT #1

By: Nathan, Ashhad, Will

Open Images v7

- ~9 million images, ~20 TB
- 61.4 million labels across 20638 classes
- Creative Commons 2.0 License

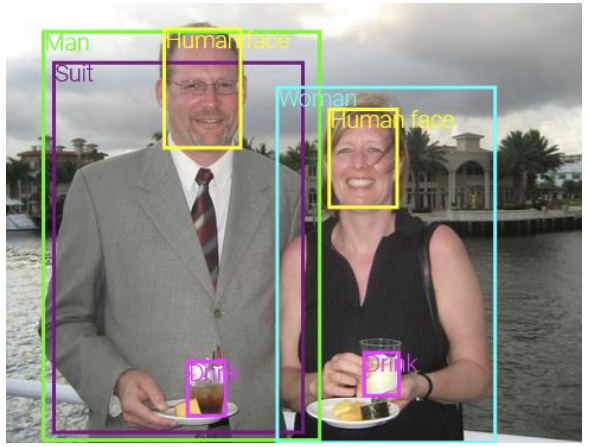


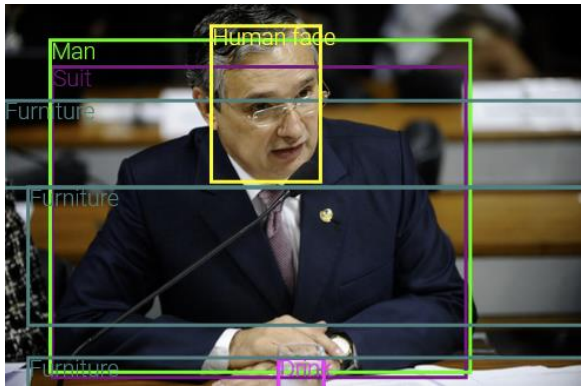


14364	/m/0g5rsyg	Pistol
14365	/m/0836fh	Piston ring
14366	/m/0gw9c2	Pistou
14367	/m/0h5xg	Pit bull
14368	/m/0gwlg1	Pit cave
14369	/m/03mgd6	Pit stop
14370	/m/02s6fs	Pit viper

Open Images v7

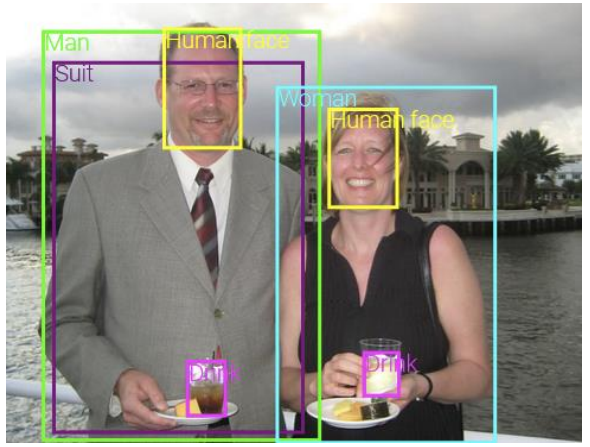
- Motivation:
  - Represents 'real life'
  - Applications in advertising, security, healthcare, accessibility
  - Predefined training, validation, test sets
  - Compare with Google Cloud Vision

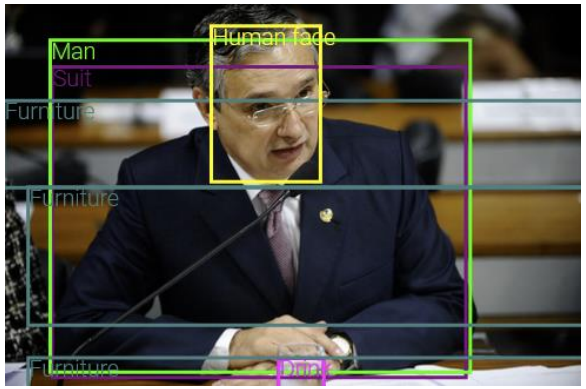




Open Images v7

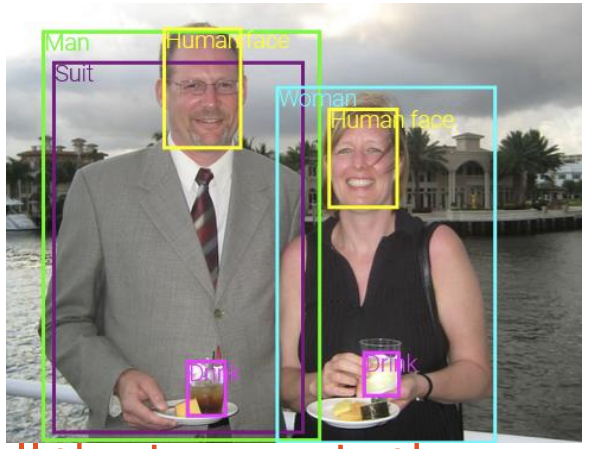
- Challenges:
  - Large size
  - Slow processing
  - Higher complexity





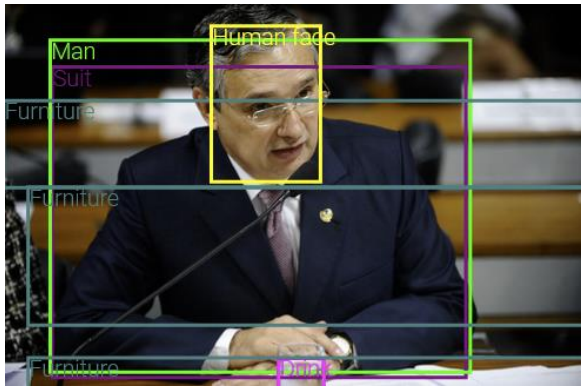
Open Images v7

- Text Corpus :-



Collection of Human-verified and Machine-generated labels of all the images in the dataset along with their IDs

14364	/m/0g5rsyg	Pistol
14365	/m/0836fh	Piston ring
14366	/m/0gw9c2	Pistou
14367	/m/0h5xg	Pit bull
14368	/m/0gwlg1	Pit cave
14369	/m/03mgd6	Pit stop
14370	/m/02s6fs	Pit viper



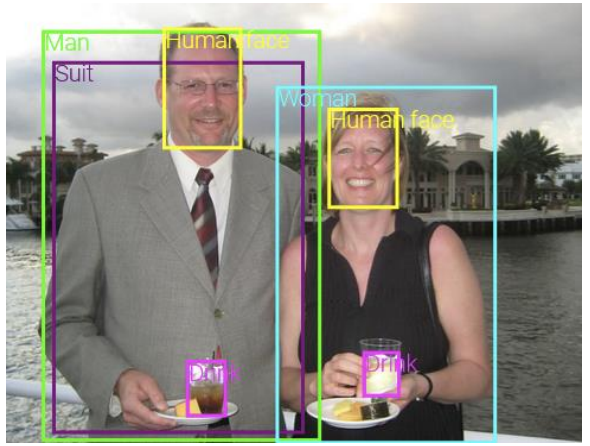
Open Images v7

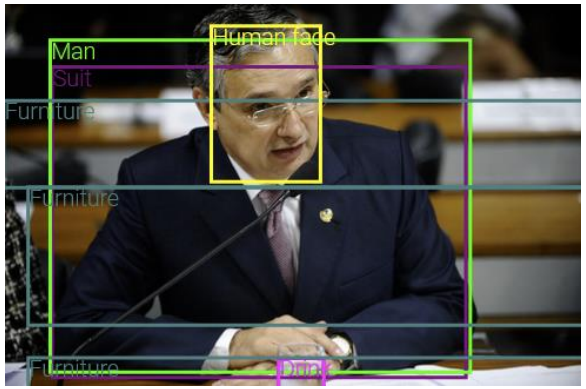
- Objective:-

Gives us an idea of class distribution across the dataset

Can downscale the dataset to the classes

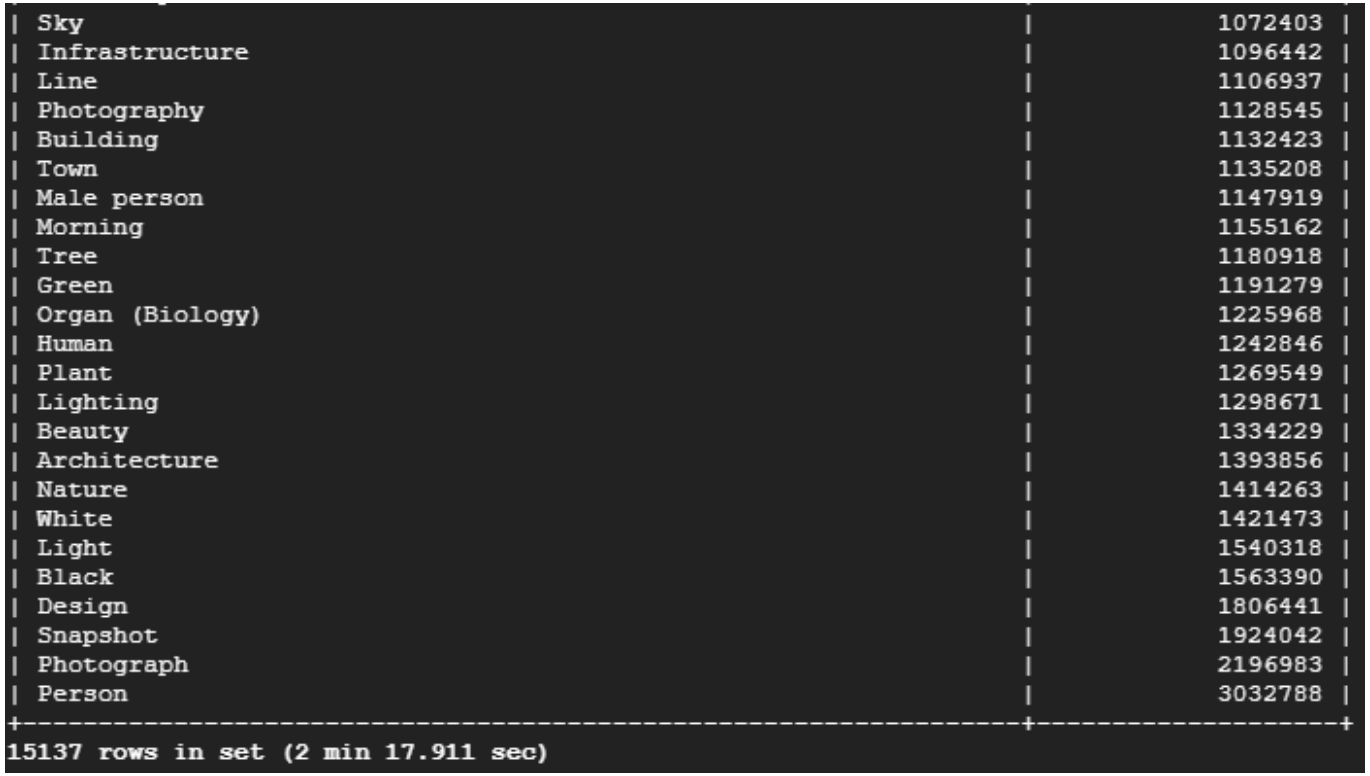
relevant to our topic for faster processing





Defining Classes

- 'Person', 'Sky', 'Tree'
- High frequency
- Avoid "descriptions"
  e.g. 'Photograph', 'Beauty', 'Design'
- Subcategories
  'Male person' within 'person'
- Broad categories
  'Plant' vs 'Tree'



Sky	1072403
Infrastructure	1096442
Line	1106937
Photography	1128545
Building	1132423
Town	1135208
Male person	1147919
Morning	1155162
Tree	1180918
Green	1191279
Organ (Biology)	1225968
Human	1242846
Plant	1269549
Lighting	1298671
Beauty	1334229
Architecture	1393856
Nature	1414263
White	1421473
Light	1540318
Black	1563390
Design	1806441
Snapshot	1924042
Photograph	2196983
Person	3032788

15137 rows in set (2 min 17.911 sec)

Software

- Spark/Pyspark
- MariaDB/SQL
- Google Cloud API

Setup

- Metadata stored separately from image data
- Concatenate image data with metadata to search for keywords

```
DROP TABLE IF EXISTS HumanClassDesc;
CREATE TABLE HumanClassDesc AS
SELECT Human.ImageID, Human.Source, Human.LabelName, Human.Confidence, ClassDesc.DisplayName
FROM Human
INNER JOIN ClassDesc
ON Human.LabelName = ClassDesc.LabelName;

DROP TABLE IF EXISTS MachineClassDesc;
CREATE TABLE MachineClassDesc AS
SELECT Machine.ImageID, Machine.Source, Machine.LabelName, Machine.Confidence, ClassDesc.DisplayName
FROM Machine
INNER JOIN ClassDesc
ON Machine.LabelName = ClassDesc.LabelName;
```

MariaDB

- Person Count: 4015096
- Sky Count: 2150621
- Tree Count: 1913744
- Time: 2 min 52.911 sec

```
Database changed
MariaDB [csvfiles]> SELECT
    ->     sum(case when DisplayName='Person' then 1 else 0 end) as PersonCount,
    ->     sum(case when DisplayName='Sky' then 1 else 0 end) as SkyCount,
    ->     sum(case when DisplayName='Tree' then 1 else 0 end) as TreeCount
    -> FROM HumanClassDesc;
+-------------+----------+-----------+
| PersonCount | SkyCount | TreeCount |
+-------------+----------+-----------+
|      982308 |  1078218 |    732826 |
+-------------+----------+-----------+
1 row in set (44.498 sec)
```

```
MariaDB [csvfiles]>   SELECT
    ->     sum(case when DisplayName='Person' then 1 else 0 end) as PersonCount,
    ->     sum(case when DisplayName='Sky' then 1 else 0 end) as SkyCount,
    ->     sum(case when DisplayName='Tree' then 1 else 0 end) as TreeCount
    -> FROM MachineClassDesc;
+-------------+----------+-----------+
| PersonCount | SkyCount | TreeCount |
+-------------+----------+-----------+
|     3032788 |  1072403 |   1180918 |
+-------------+----------+-----------+
1 row in set (2 min 8.413 sec)
```

Spark

- Person Count: 4015096
- Sky Count: 2150621
- Tree Count: 1913744
- Time: 7 min 34 sec

```
# Create SparkSession
from pyspark.sql import SparkSession
from pyspark.sql.functions import when
from pyspark.sql.functions import sum
import time

start = time.time()
spark:SparkSession = SparkSession.builder.master("local[1]").appName("ParseData").getOrCreate()

dataframe = spark.read.csv("/mnt/disks/Disk1/sql/tables/TotalSet.csv")
dataframe.createOrReplaceTempView("dataset")
spark.sql("SELECT sum(case when _c4='Person' then 1 else 0 end) as PersonCount,sum(case when _c4='Sky' then 1 e
lse 0 end) as SkyCount,sum(case when _c4='Tree' then 1 else 0 end) as TreeCount FROM dataset;").show()

print("Program Elapsed Time")
print(time.time() - start, "seconds")
```

```
+-----------+--------+---------+
|PersonCount|SkyCount|TreeCount|
+-----------+--------+---------+
|    4015096| 2150621|  1913744|
+-----------+--------+---------+

Program Elapsed Time
454.0396361351013 seconds
```

Conclusion

- MariaDB faster by 4 min 41 sec
- Spark better with larger datasets