output_Playing-Cards-Images-Object-Detection-Dataset

October 26, 2025

0.1 Step1: Data Exploration

Objective: To explore and analyze the Playing Cards dataset in order to understand its structure and characteristics before the modeling phase.

Main Goals: - Perform an initial assessment of data quality and completeness. - Identify attribute types and their distributions. - Detect missing, inconsistent, or anomalous values. - Visualize data through appropriate plots to support interpretation.

Methods and Tools: - Statistical summaries using descriptive measures (mean, median, standard deviation, etc.). - Visual exploration with histograms, boxplots, and scatter plots. - Correlation analysis between features.

Outcome: - Clear overview of the dataset and its key properties. - Identification of relevant features and potential preprocessing needs. - Foundation for subsequent steps in the Intelligent System design process, following the KDD methodology (selection, preprocessing, modeling, and evaluation).

0.1.1 0) Setup

0.1.2 1) Images Dimensions and Aspect Ratios

We analyze the dimensions (width, height) of the images and compute their aspect ratios.

This helps us understand the variability in image sizes and shapes.

We then plot histograms of widths, heights, and aspect ratios.

```
<IPython.core.display.HTML object>
```

```
 Processed 637 bounding boxes
```

```
================================================================================
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
 Class       Instances

 6h                  21
 Jh                  16
```

2h	16
Ad	16
Jc	16

<IPython.core.display.HTML object>

Class	Instances
4c	9
7d	9
9c	9
3h	6
6c	3

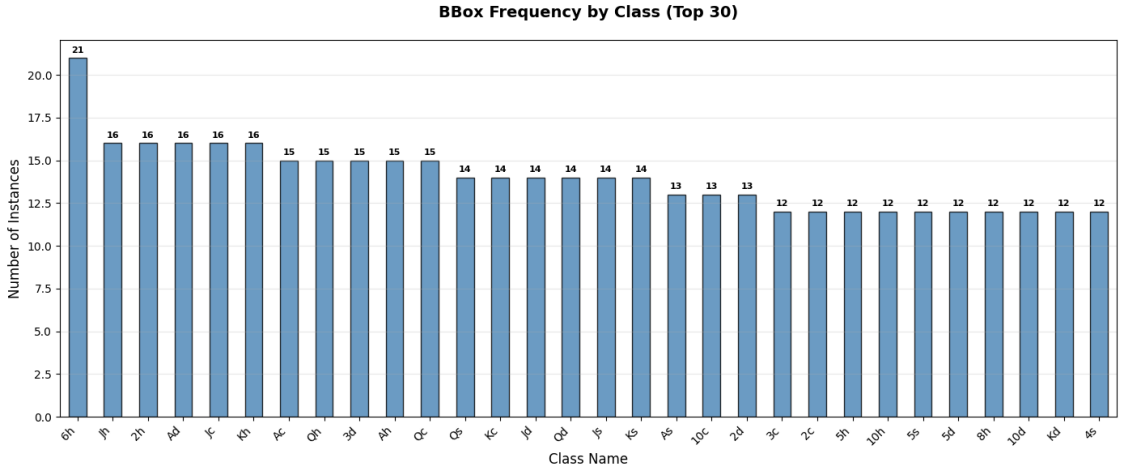
<IPython.core.display.HTML object>

Maximum instances per class:	21
Minimum instances per class:	3
Mean instances per class:	12.25
Median instances per class:	12.0
Total unique classes:	52

==========

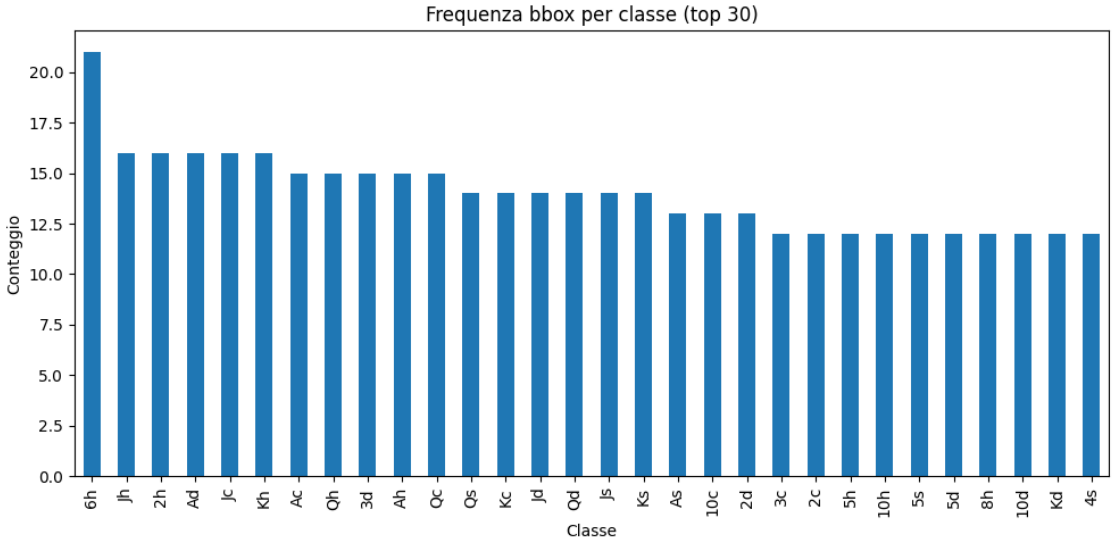
<IPython.core.display.HTML object>

<IPython.core.display.HTML object>



0.1.3 2) Classes Distribution Analysis

```
======================================================================
                        GLOBAL CLASS STATISTICS
======================================================================

Most frequent classes (Top 5):


  Class       Instances

  6h                 21
  Jh                 16
  2h                 16
  Ad                 16
  Jc                 16


Least frequent classes (Bottom 5):


  Class       Instances

  4c                  9
  7d                  9
  9c                  9
  3h                  6
  6c                  3


CLASS DISTRIBUTION STATISTICS
----------------------------------------------------------------------
Max instances per class: 21
Min instances per class: 3
Mean instances per class: 12.25
Total unique classes: 52
======================================================================
```



0.1.4 7) Bounding Box Analysis

We check for anomalies in the labels such as: - bounding boxes with normalized coordinates outside [0,1] - bounding boxes with zero area (width or height equal to zero) - images without any bounding boxes

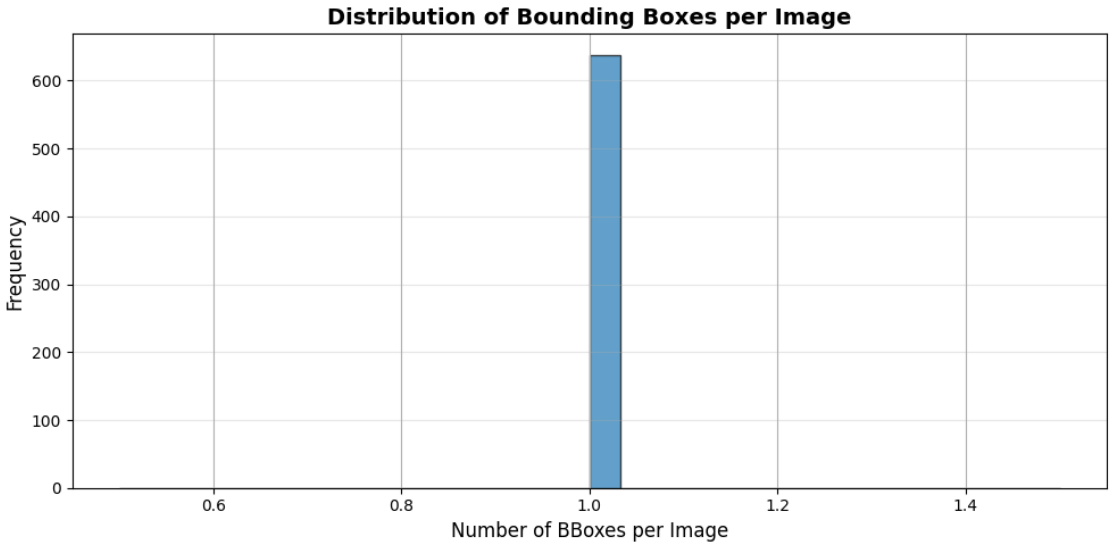
```
<IPython.core.display.HTML object>
```

```
Total bounding boxes:                                  637
BBoxes with coordinates outside [0,1]:                   0
BBoxes with near-zero area:                              0
BBoxes with very large dimensions (w>0.9 or h>0.9):    266
Images without any bbox:                          0.00% (0 / 637)
```

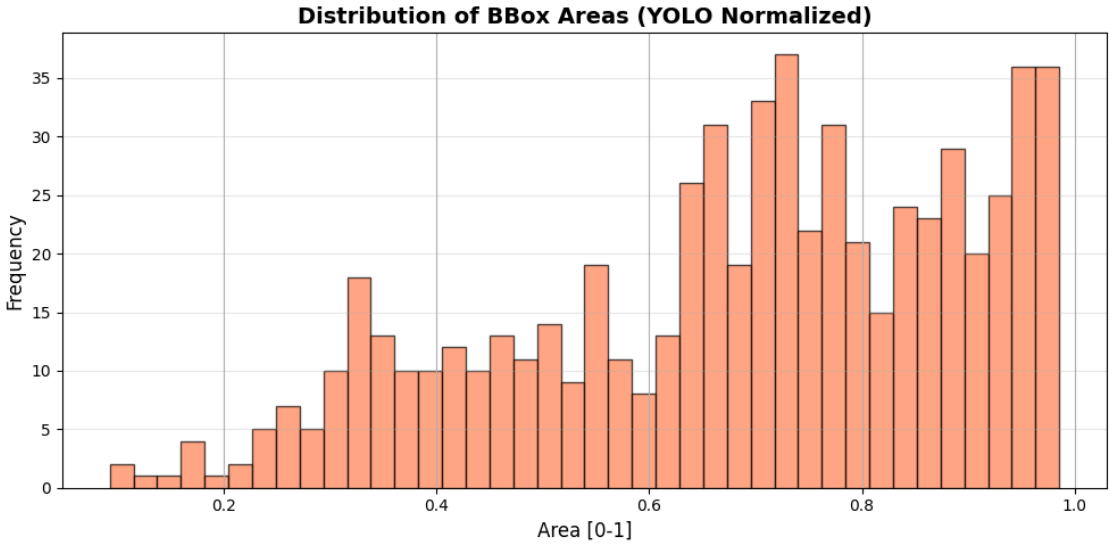
```
================================================================================
```

```
<IPython.core.display.HTML object>
```

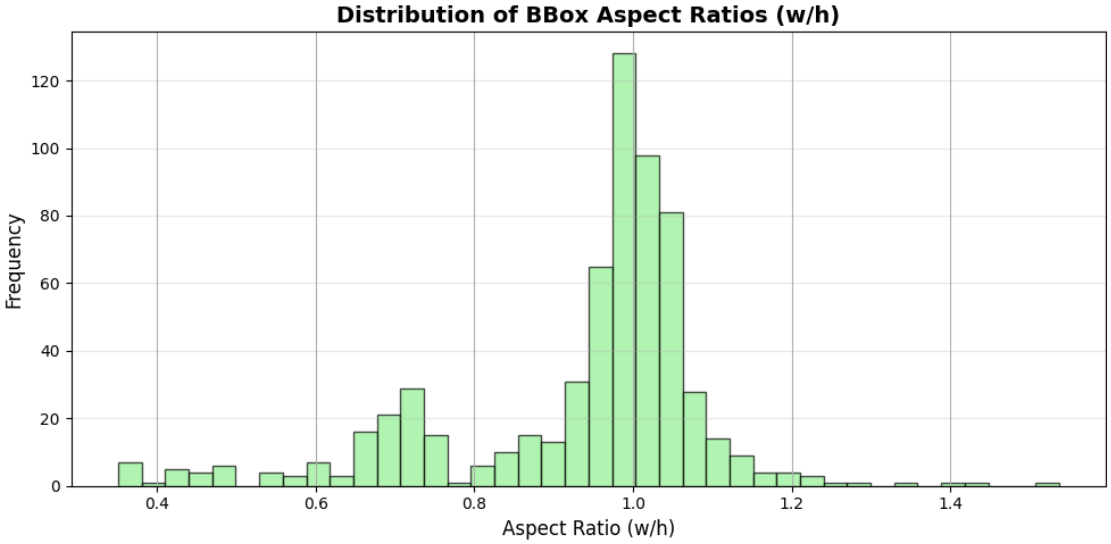
```
<IPython.core.display.HTML object>
```



<IPython.core.display.HTML object>

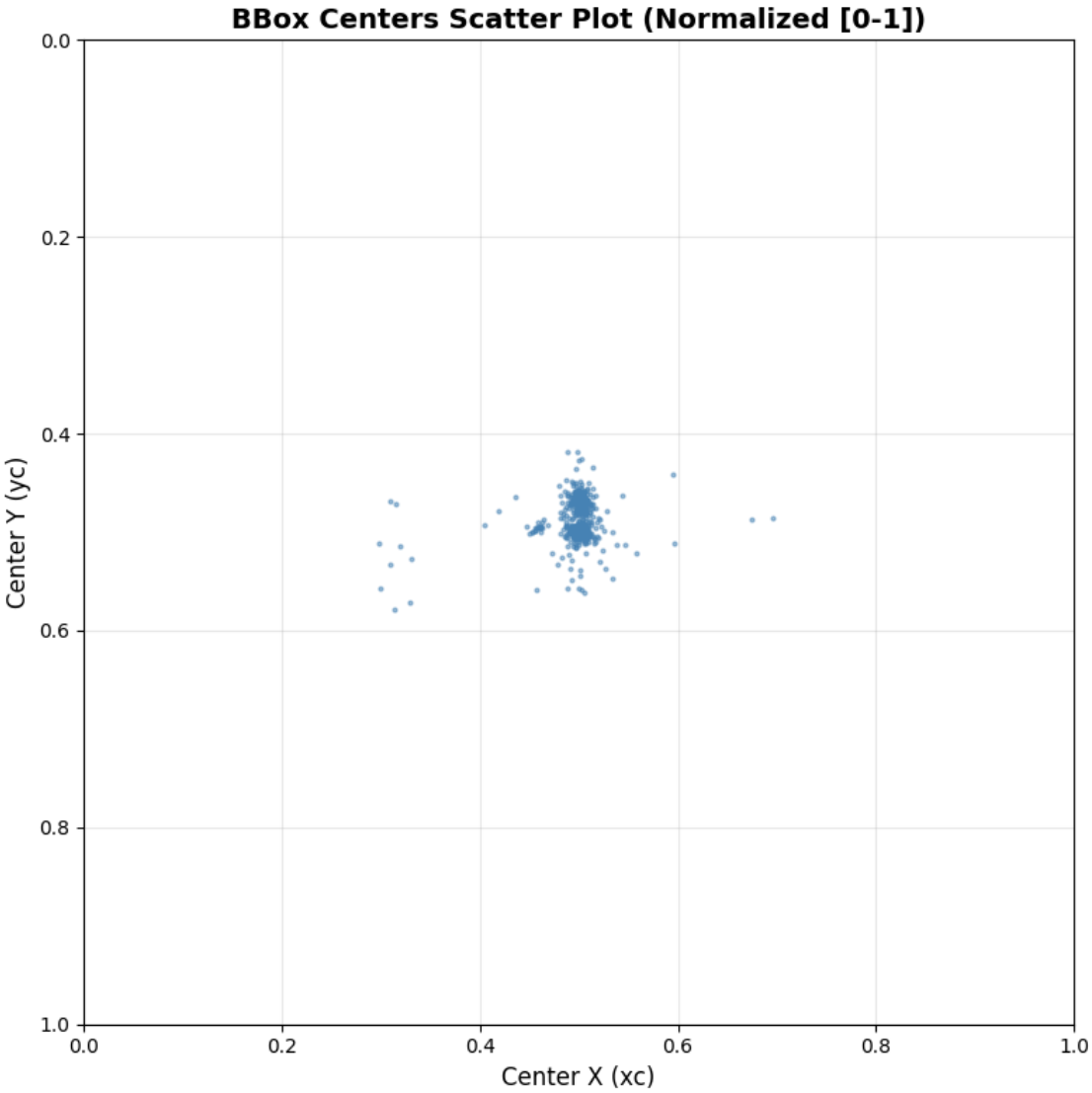


<IPython.core.display.HTML object>

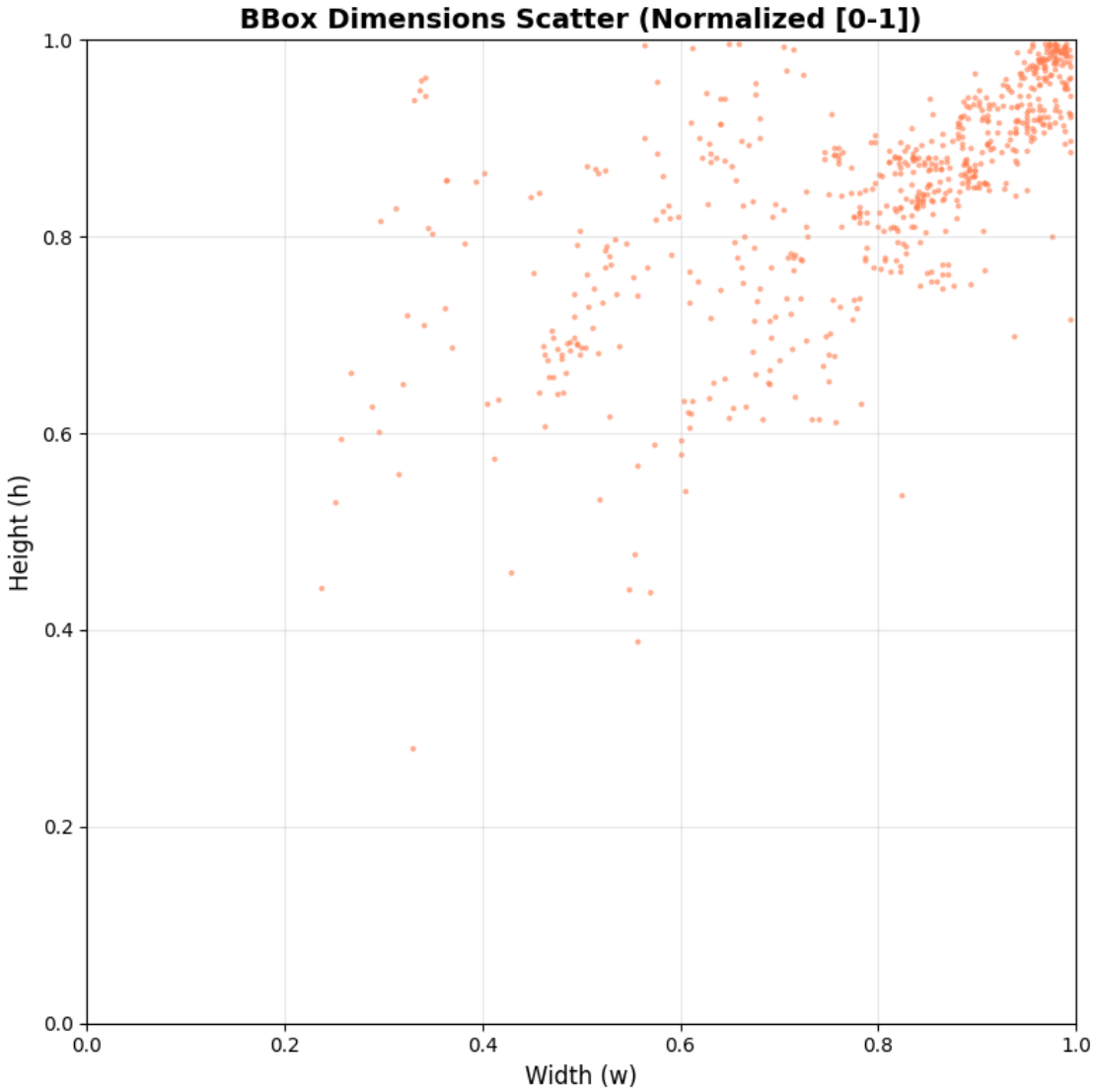


<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

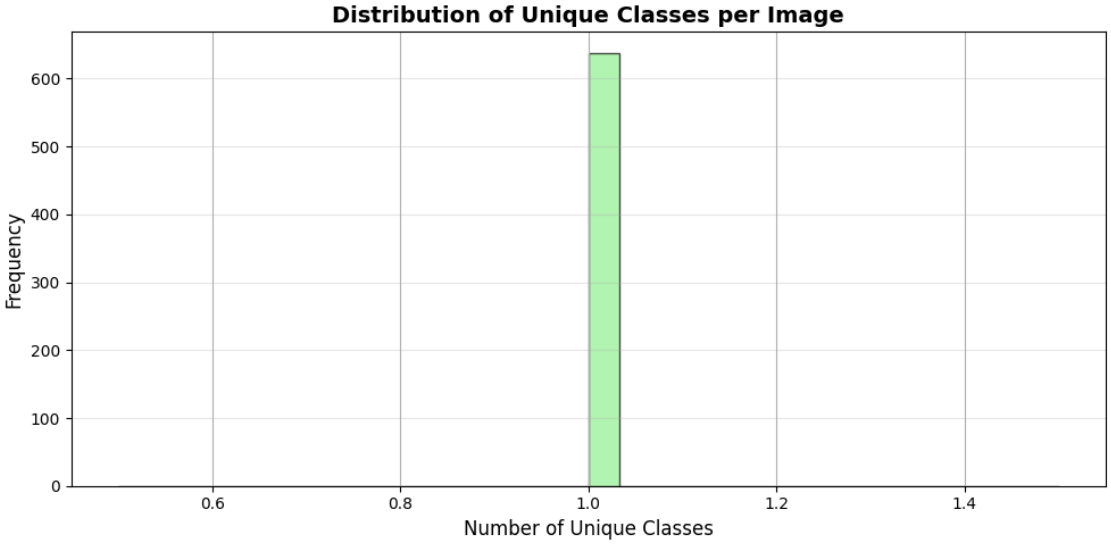


<IPython.core.display.HTML object>



<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

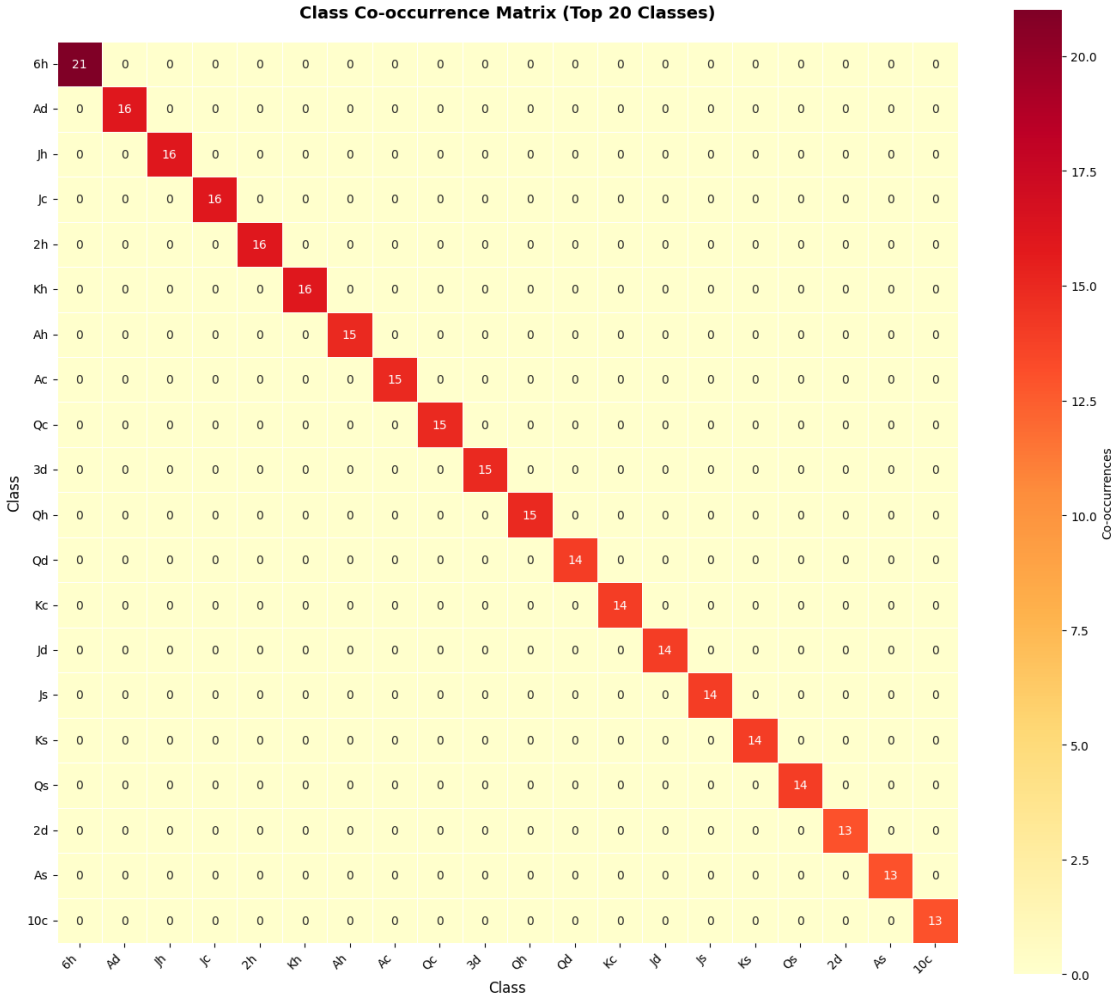


0.1.5 3) Data Visualization

We visualize a few sample images from each split with their corresponding bounding boxes drawn on them.

This helps us qualitatively assess the quality of the annotations and get a better understanding of the dataset content.

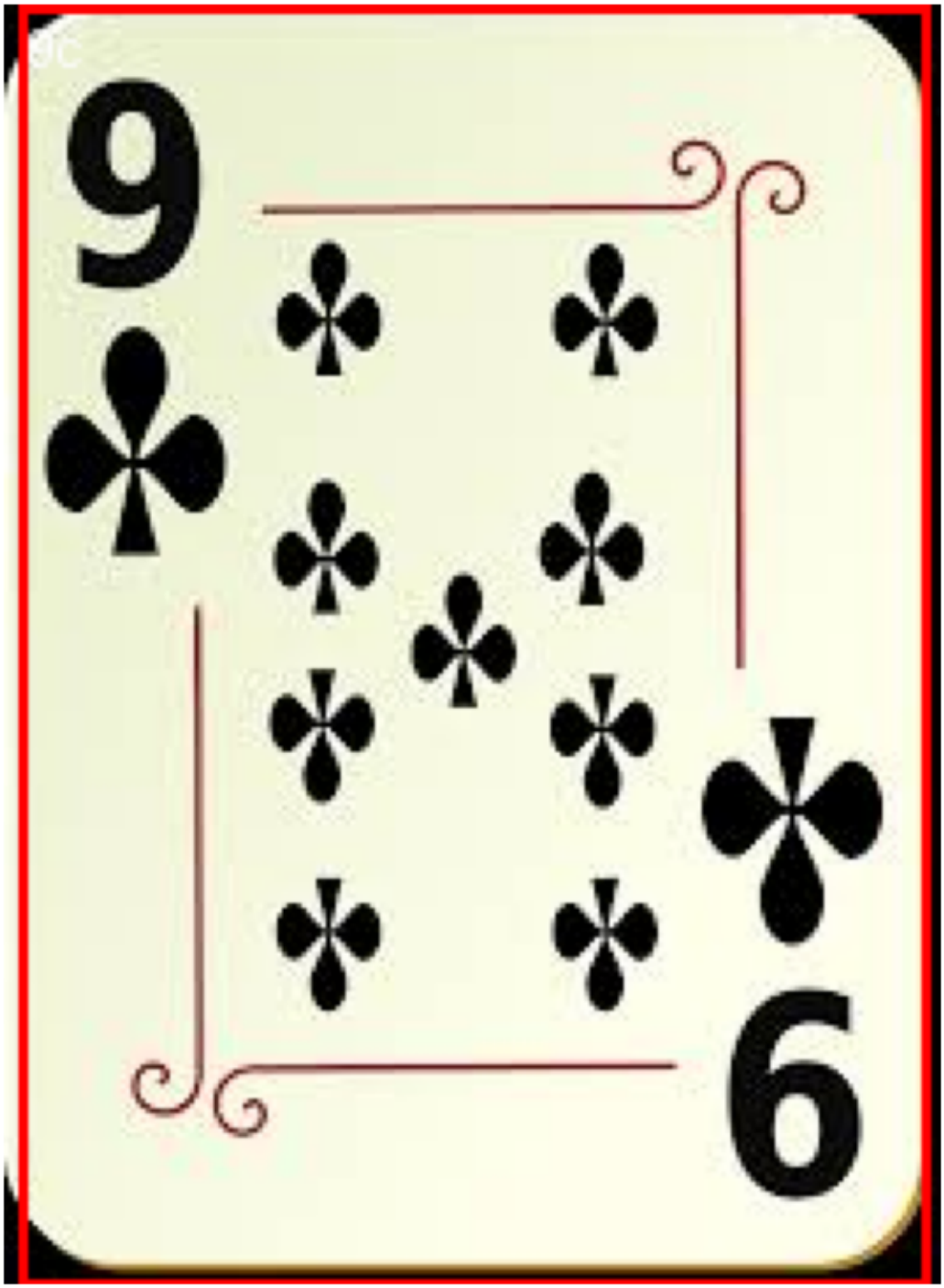
```
<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

Co-occurrence matrix computed: 52x52 classes

<IPython.core.display.HTML object>

No co-occurrences found (no image contains more than one class).

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

No conditional relationships found (each image likely has a single class).

<IPython.core.display.HTML object>
```



<IPython.core.display.HTML object>

ALL: k40.jpg
(1 bounding boxes)

ALL: C93.jpg
(1 bounding boxes)



ALL: H89.jpg
(1 bounding boxes)