output_Playing-Cards-Labelized-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 48,093 bounding boxes

==

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

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

Class	Instances
7h	992
4d	988

```
     3s                 980
     Ac                 978
     6d                 972


<IPython.core.display.HTML object>


  Class        Instances

     5c                 884
     9h                 884
     Jh                 879
     6h                 873
     10s                867


<IPython.core.display.HTML object>

Maximum instances per class:              992
Minimum instances per class:              867
Mean instances per class:              924.87
Median instances per class:             925.0
Total unique classes:                      52


==============================================================================


<IPython.core.display.HTML object>

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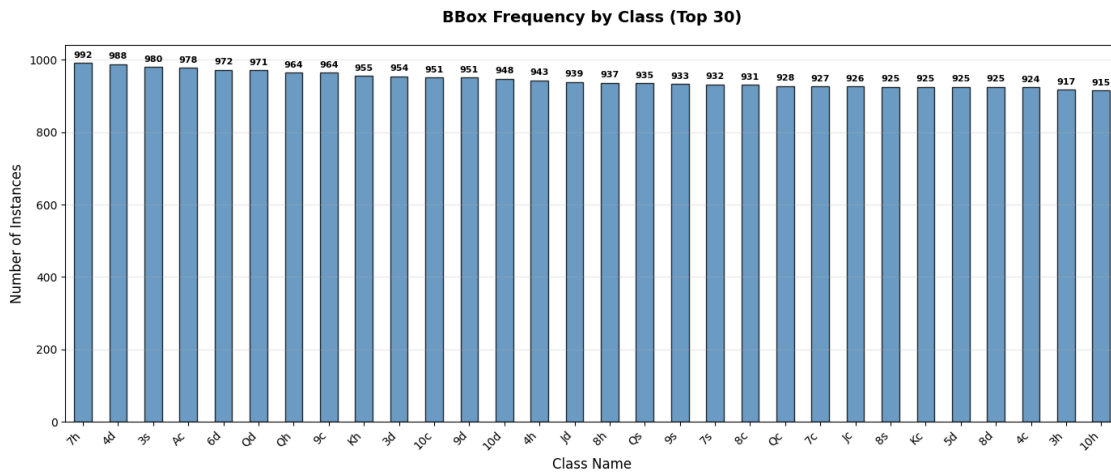

BBox Frequency by Class (Top 30)

0.1.3 2) Classes Distribution Analysis

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


Most frequent classes (Top 5):


  Class        Instances

  7h                 992
  4d                 988
  3s                 980
  Ac                 978
  6d                 972


Least frequent classes (Bottom 5):


  Class        Instances

  5c                 884
  9h                 884
  Jh                 879
  6h                 873
  10s                867


CLASS DISTRIBUTION STATISTICS
------------------------------------------------------------------------
Max instances per class: 992
Min instances per class: 867
Mean instances per class: 924.87
Total unique classes: 52
========================================================================
```
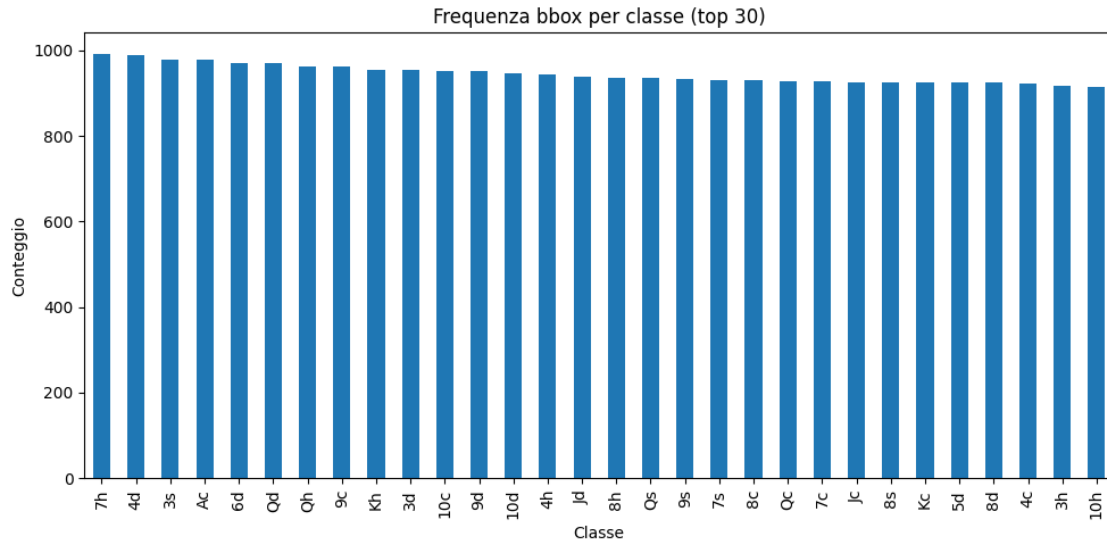
Frequenza bbox per classe (top 30)

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:                            48,093
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):   0
Images without any bbox:               0.00% (0 / 6,000)


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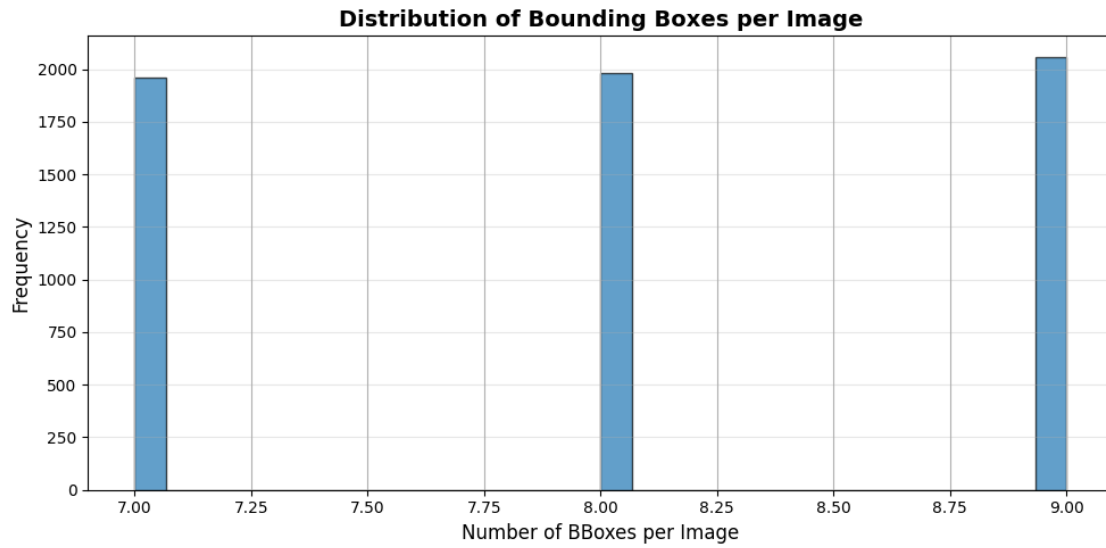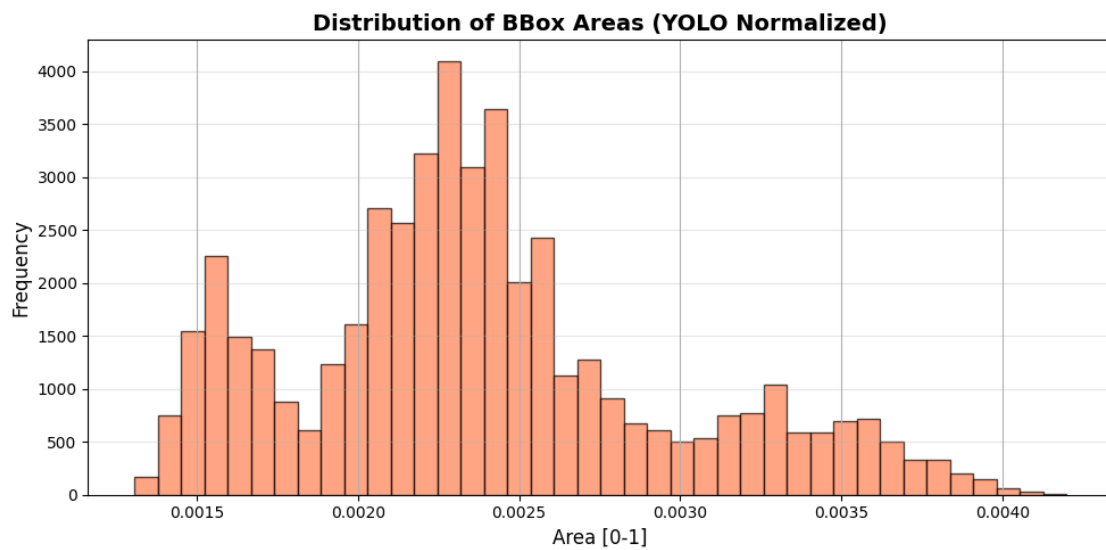
```
<IPython.core.display.HTML object>
```

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

Distribution of Bounding Boxes per Image

<IPython.core.display.HTML object>

Distribution of BBox Areas (YOLO Normalized)

<IPython.core.display.HTML object>

Distribution of BBox Aspect Ratios (w/h)

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

BBox Centers Scatter Plot (Normalized [0-1])

<IPython.core.display.HTML object>

BBox Dimensions Scatter (Normalized [0-1])

<IPython.core.display.HTML object>
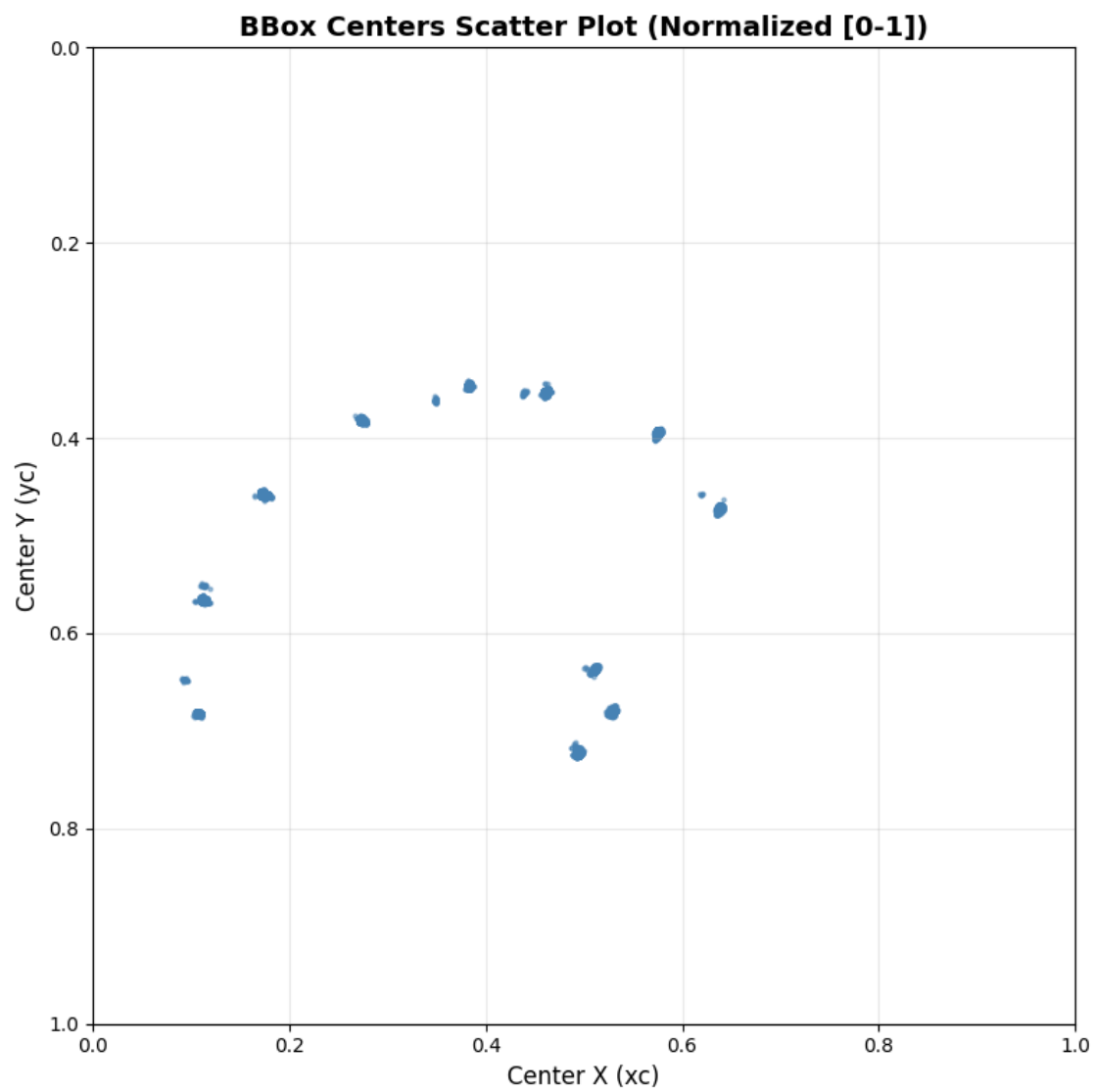
<IPython.core.display.HTML object>
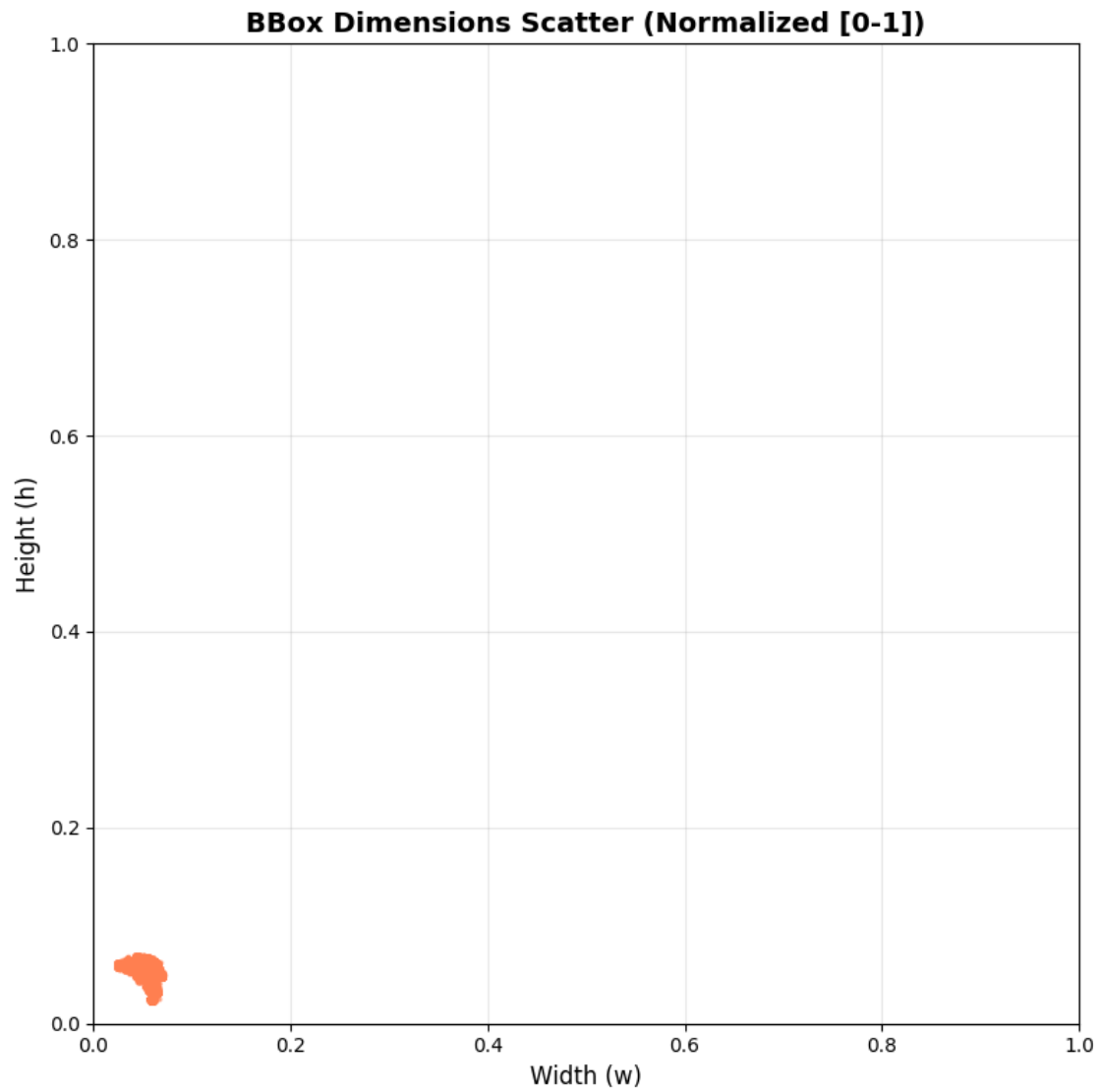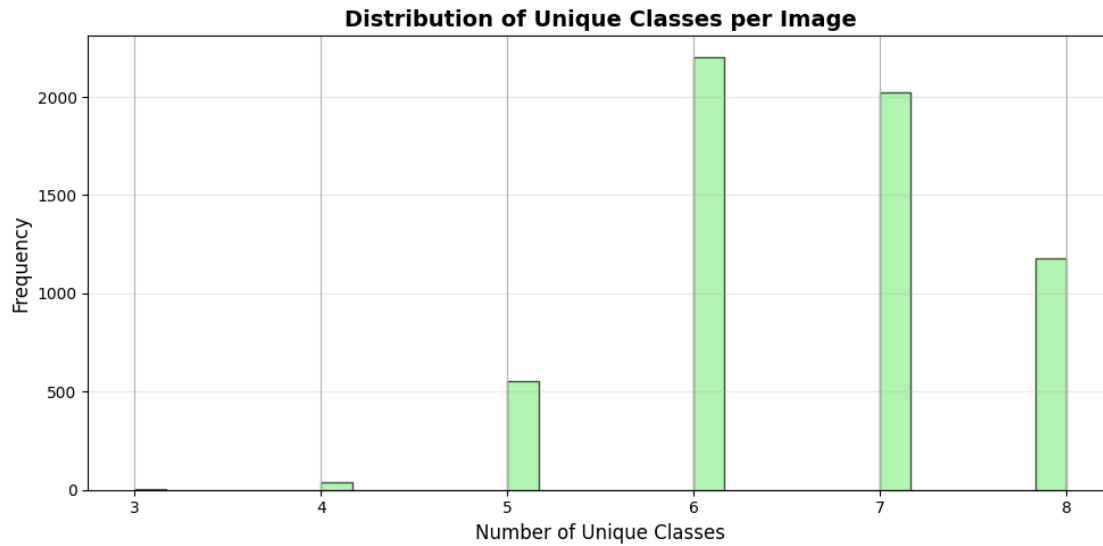
Distribution of Unique Classes per Image

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>
```

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

Class 1	Class 2	Co-occurrences
4d	6d	112
4d	8d	111
4d	Qc	111
2h	Kc	110
10h	3s	110
3d	Kd	109
Qc	Qs	109
7s	Qd	109
4c	6d	108
7h	Jc	107
8s	Qh	107

```
2c          Ac                        107
3s          As                        107
6d          7h                        107
Ac          Qd                        107
```

<IPython.core.display.HTML object>

```
Class 1     Class 2      Co-occurrences

2h          Ac                         66
Kh          Qs                         65
2d          Qc                         65
Ad          Jc                         65
5s          Ah                         64
10h         6h                         64
5c          As                         63
2s          5d                         63
2d          5s                         63
2s          8d                         60
```

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

```
Given Class     Then Class      P(Then|Given) %

10h             3s                       14.78
2h              Kc                       14.77
Kd              3d                       14.59
8d              4d                       14.49
Qc              4d                       14.40
2c              Ac                       14.36
Kc              2h                       14.27
Jc              7h                       14.27
7s              Qd                       14.25
2h              10d                      14.23
10s             7h                       14.21
As              3s                       14.17
Qc              Qs                       14.14
4c              6d                       14.14
Qs              Qc                       14.12
```
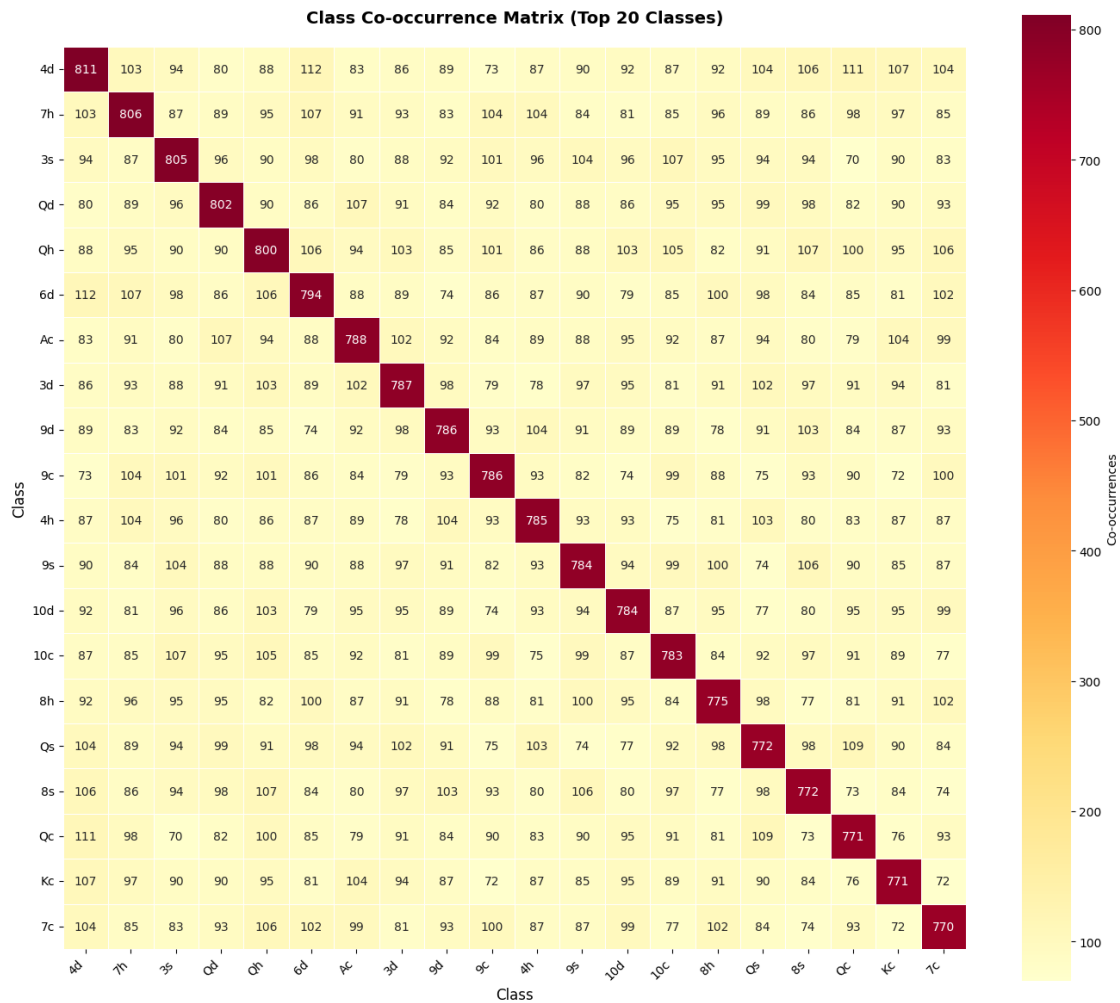
<IPython.core.display.HTML object>

Class Co-occurrence Matrix (Top 20 Classes)

<IPython.core.display.HTML object>