```
!pip install opencv-python
```

```
Requirement already satisfied: opencv-python in e:\anaconda\lib\site-
packages (4.12.0.88)
Requirement already satisfied: numpy<2.3.0,>=2 in e:\anaconda\lib\
site-packages (from opencv-python) (2.1.3)
```

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import cv2
import os
import math
plt.rcParams.update({
    "font.size": 15,
    "font.family": "Arial"
})
bxwidth = 1
```

```
df = pd.read_csv("D:/pml/handwritten/handwritten/english.csv")
```

```
df
```

```
                   image label
0     Img/img001-001.png     0
1     Img/img001-002.png     0
2     Img/img001-003.png     0
3     Img/img001-004.png     0
4     Img/img001-005.png     0
...                  ...   ...
3405  Img/img062-051.png     z
3406  Img/img062-052.png     z
3407  Img/img062-053.png     z
3408  Img/img062-054.png     z
3409  Img/img062-055.png     z

[3410 rows x 2 columns]
```

```
image_folder = "D:/pml/handwritten/handwritten/Img/"
base_path = "D:/pml/handwritten/handwritten/"
```

```
df.columns = df.columns.str.strip()
```

```
print(df.describe())
print(df.info())
```

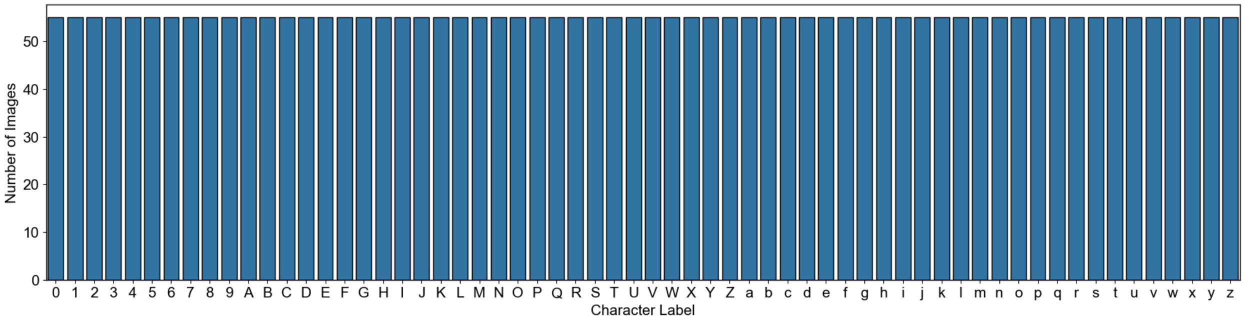
```
                     image label
count                 3410  3410
unique                3410    62
top     Img/img062-055.png     0
freq                     1    55
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3410 entries, 0 to 3409
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   image   3410 non-null   object
 1   label   3410 non-null   object
dtypes: object(2)
memory usage: 53.4+ KB
None
```

```
print(df.isnull().sum())
```

```
image    0
label    0
dtype: int64
```

```
cat_cols = ['label']
rows = math.ceil(len(cat_cols) / 1)

fig, axes = plt.subplots(rows, 1, figsize=(18, 6 * rows))
if rows == 1:
    axes = [axes]
else:
    axes = axes.flatten()

subplot_labels = [f"{chr(97+i)})" for i in range(len(cat_cols))]

for i, col in enumerate(cat_cols):
    ax = axes[i]
    sns.countplot(data=df, x=col, order=sorted(df[col].unique()),
ax=ax, edgecolor="black")

    ax.set_title("") # No title on top
    ax.set_xlabel("Character Label")
    ax.set_ylabel("Number of Images")

    ax.text(0.5, -0.25, f"{subplot_labels[i]} Frequency of each
Character Label",
            ha="center", va="top", transform=ax.transAxes,
fontsize=13)

    for spine in ax.spines.values():
        spine.set_linewidth(bxwidth)

fig.text(0.5, 0.01, "Count plot showing the frequency distribution of
character labels in the dataset.",
         ha="center", fontsize=15)

plt.tight_layout(rect=[0, 0.05, 1, 1])
plt.savefig("label_distribution_handwritten.eps", format="eps",
dpi=600, bbox_inches="tight")
plt.show()
```

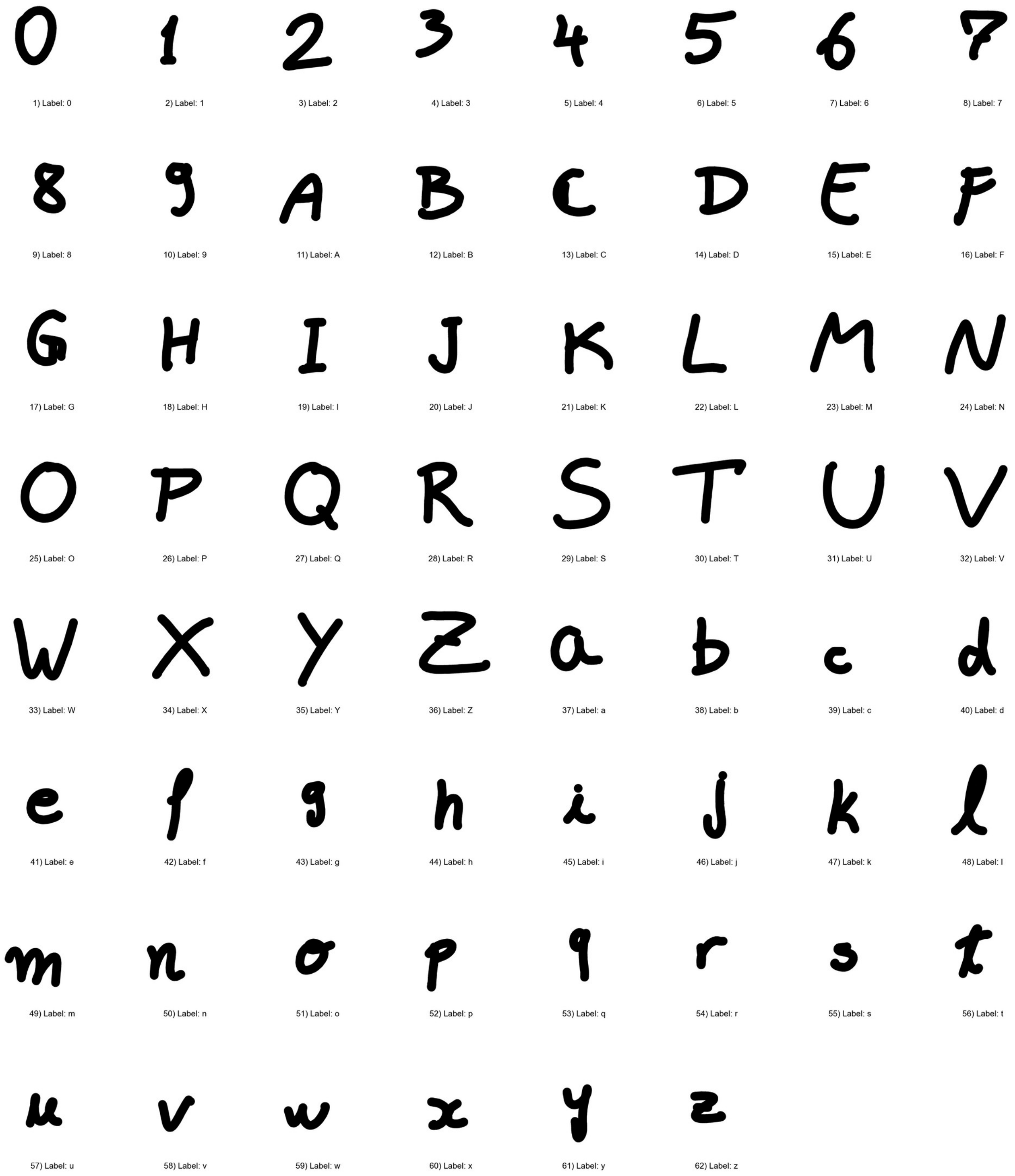


a) Frequency of each Character Label

Count plot showing the frequency distribution of character labels in the dataset.

```
samples = df.groupby('label').first().reset_index()
num_labels = len(samples)
cols_per_row = 8
rows_needed = math.ceil(num_labels / cols_per_row)

fig, axes = plt.subplots(rows_needed, cols_per_row, figsize=(20, 3 *
rows_needed))
axes = axes.flatten()

subplot_labels = [f"{i+1})" for i in range(num_labels)]

for i, (idx, row) in enumerate(samples.iterrows()):
    ax = axes[i]
    img_relative_path = row['image']
    img_full_path = os.path.join(base_path, img_relative_path)

    if os.path.exists(img_full_path):
        image = cv2.imread(img_full_path)
        image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        ax.imshow(image_rgb)
    else:
        ax.text(0.5, 0.5, "File Not Found", ha='center', va='center')

    ax.set_title("")
    ax.axis('off')

    ax.text(0.5, -0.20, f"{subplot_labels[i]} Label: {row['label']}",
            ha="center", va="top", transform=ax.transAxes,
fontsize=11)

    for spine in ax.spines.values():
        spine.set_visible(True)
        spine.set_linewidth(bxwidth)
```

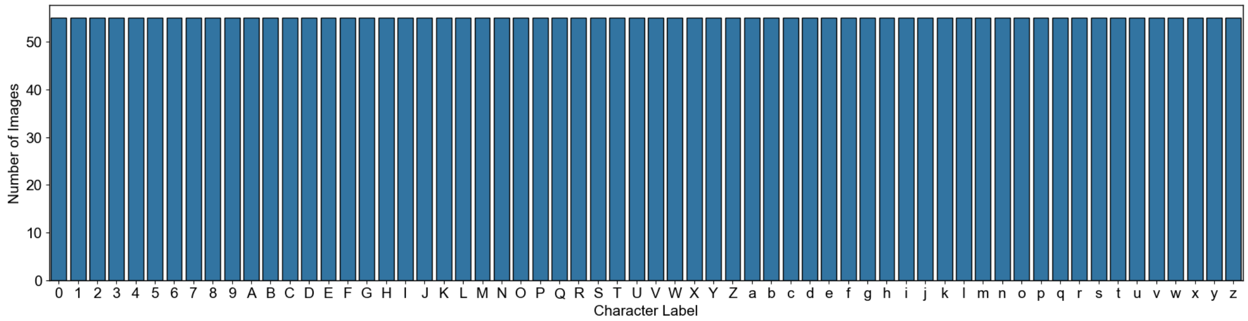
```
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

fig.text(0.5, 0.01, "Figure: Sample images for each unique character
label in the dataset.",
         ha="center", fontsize=15)

plt.tight_layout(rect=[0, 0.05, 1, 1])
plt.savefig("sample_characters_grid.eps", format="eps", dpi=600,
bbox_inches="tight")
plt.show()
```



1) Label: 0
2) Label: 1
3) Label: 2
4) Label: 3
5) Label: 4
6) Label: 5
7) Label: 6
8) Label: 7
9) Label: 8
10) Label: 9
11) Label: A
12) Label: B
13) Label: C
14) Label: D
15) Label: E
16) Label: F
17) Label: G
18) Label: H
19) Label: I
20) Label: J
21) Label: K
22) Label: L
23) Label: M
24) Label: N
25) Label: O
26) Label: P
27) Label: Q
28) Label: R
29) Label: S
30) Label: T
31) Label: U
32) Label: V
33) Label: W
34) Label: X
35) Label: Y
36) Label: Z
37) Label: a
38) Label: b
39) Label: c
40) Label: d
41) Label: e
42) Label: f
43) Label: g
44) Label: h
45) Label: i
46) Label: j
47) Label: k
48) Label: l
49) Label: m
50) Label: n
51) Label: o
52) Label: p
53) Label: q
54) Label: r
55) Label: s
56) Label: t
57) Label: u
58) Label: v
59) Label: w
60) Label: x
61) Label: y
62) Label: z

Figure: Sample images for each unique character label in the dataset.

```
fig, ax = plt.subplots(figsize=(18, 6))

sns.countplot(data=df, x='label', order=sorted(df['label'].unique()),
ax=ax, edgecolor="black")

ax.set_title("")
ax.set_xlabel("Character Label")
ax.set_ylabel("Number of Images")

ax.text(
    0.5, -0.25,
    "a) Frequency of each Character Label in the Dataset",
    ha="center",
    va="top",
    transform=ax.transAxes,
    fontsize=13
)

for spine in ax.spines.values():
    spine.set_linewidth(bxwidth)

fig.text(
    0.5, 0.01,
    "Figure: Bar chart showing the distribution of handwritten
character samples across all labels.",
    ha="center",
    fontsize=15
)

plt.tight_layout(rect=[0, 0.05, 1, 1])
plt.savefig(
    "label_distribution_handwritten.eps",
    format="eps",
    dpi=600,
    bbox_inches="tight"
)
plt.show()
```



a) Frequency of each Character Label in the Dataset

Figure: Bar chart showing the distribution of handwritten character samples across all labels.