Datensatz_Visualisierung

February 12, 2021

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras import datasets, layers, models
```

[2]:
```
(X_train10, y_train10), (X_test10, y_test10)     = tf.keras.datasets.cifar10.
 ↪load_data()
(X_train100, y_train100), (X_test100, y_test100) = tf.keras.datasets.cifar100.
 ↪load_data()
```

[3]:
```
print('Images Shape: {}'.format(X_train10.shape))
print('Labels Shape: {}'.format(y_train10.shape))
print('Images Shape: {}'.format(X_train100.shape))
print('Labels Shape: {}'.format(y_train100.shape))
```

```
Images Shape: (50000, 32, 32, 3)
Labels Shape: (50000, 1)
Images Shape: (50000, 32, 32, 3)
Labels Shape: (50000, 1)
```

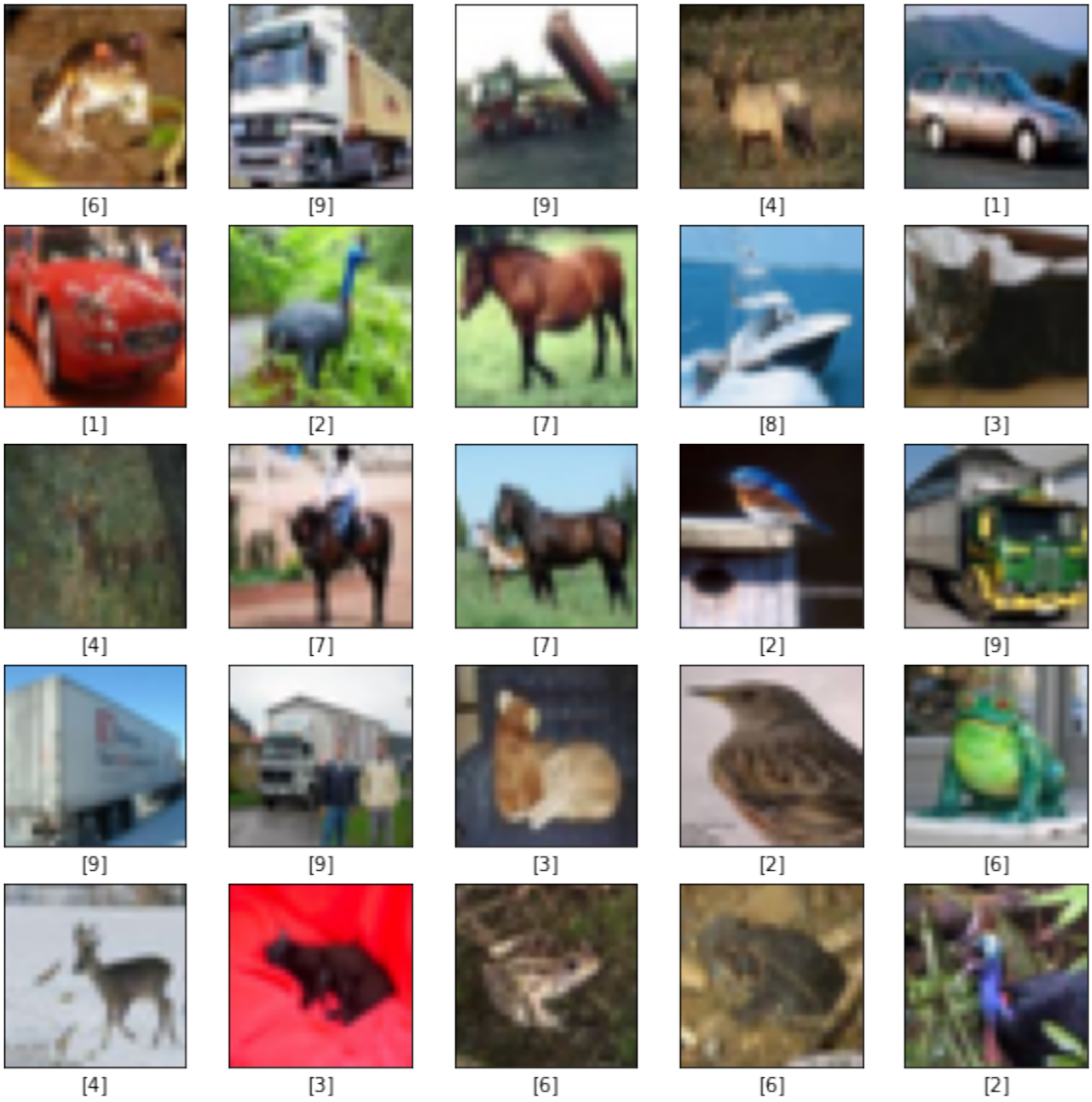
[4]:
```
print(y_train10[:10])
```

```
[[6]
 [9]
 [9]
 [4]
 [1]
 [1]
 [2]
 [7]
 [8]
 [3]]
```

[5]:
```
plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
```

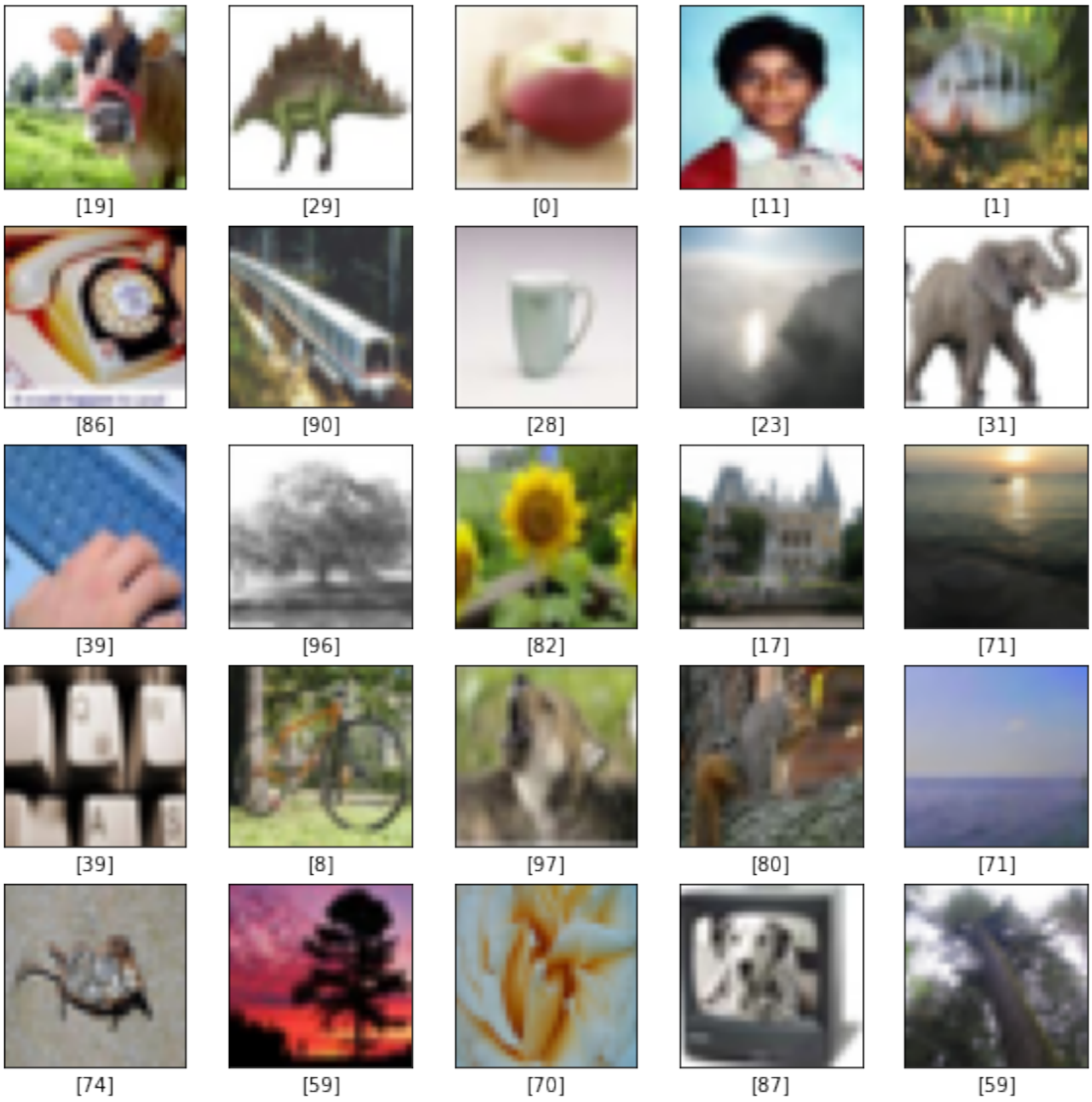
```
    plt.imshow(X_train10[i], cmap=plt.cm.binary)
    plt.xlabel(y_train10[i])
plt.show()
```

```
D:\Anaconda\lib\site-packages\matplotlib\text.py:1163: FutureWarning:
elementwise comparison failed; returning scalar instead, but in the future will
perform elementwise comparison
  if s != self._text:
```



[6]:
```
plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
```

```
    plt.yticks([])
    plt.grid(False)
    plt.imshow(X_train100[i], cmap=plt.cm.binary)
    plt.xlabel(y_train100[i])
plt.show()
```

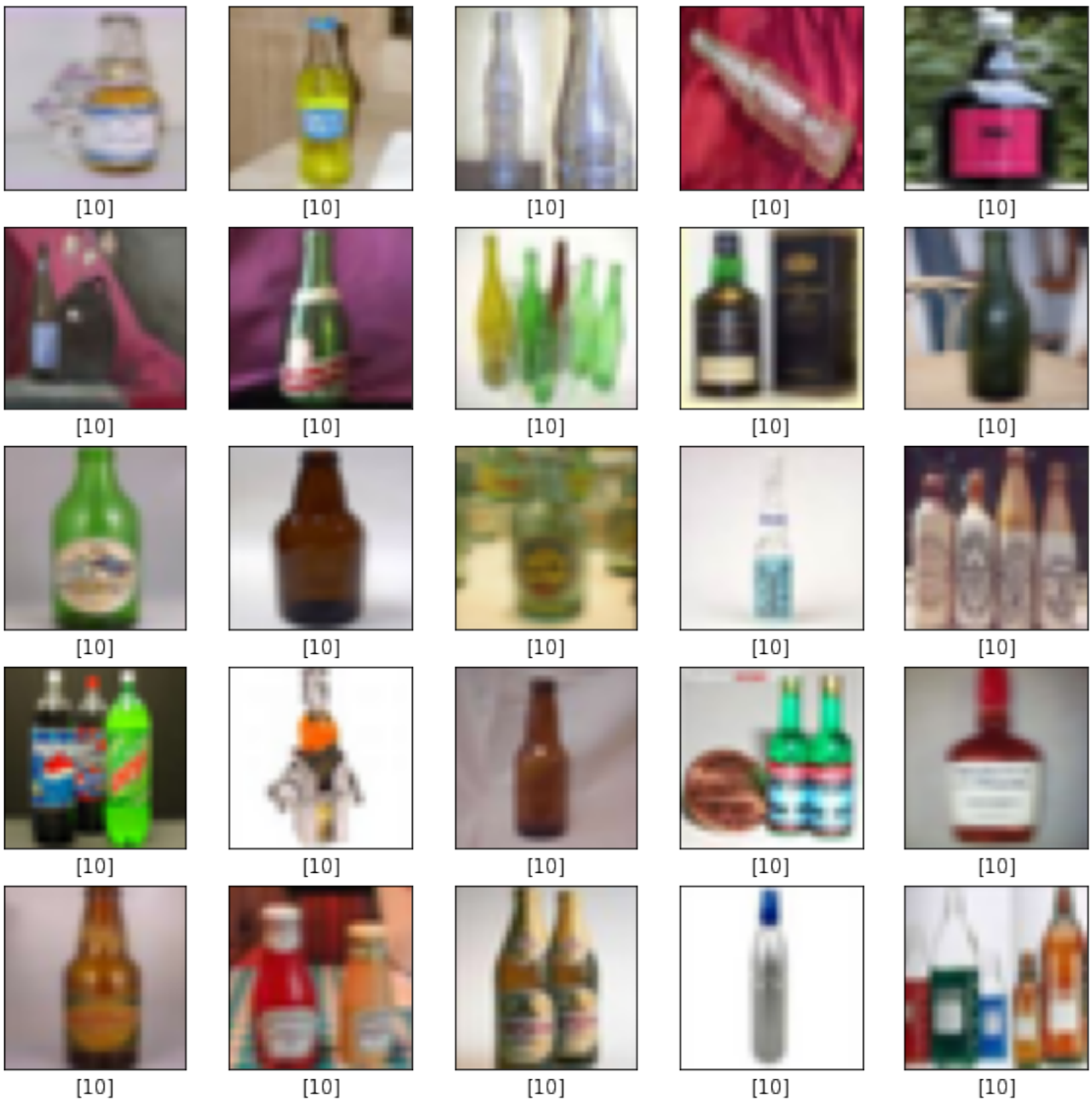


[7]:
```
idx         = (y_train100 == 9).reshape(X_train100.shape[0])
X_train100 = X_train100[idx]
y_train100 = y_train100[idx]
for i in range(len(y_train100)):
    y_train100[i]=10
```

[8]:
```
len(X_train100)
```

[8]: 500

[9]:
```
plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(X_train100[i], cmap=plt.cm.binary)
    plt.xlabel(y_train100[i])
plt.show()
```



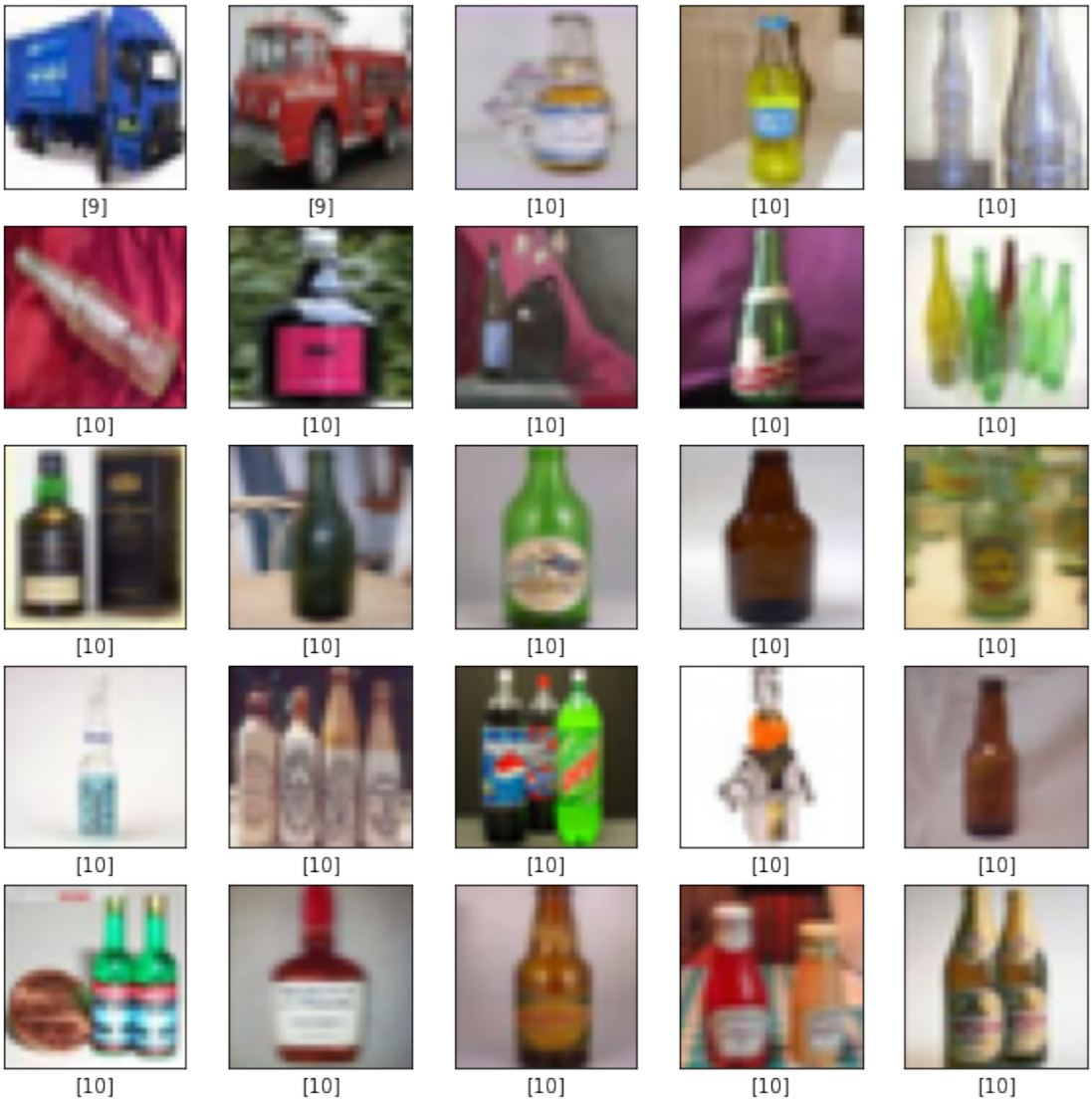
[10]:
```
X_train10_red = [None]*5000
y_train10_red = [None]*5000

for i in range(10):
    idx = (y_train10 == i).reshape(X_train10.shape[0])
    x   = X_train10[idx]
    y   = y_train10[idx]
    X_train10_red[i*500:i*500+500] = x[0:500]
    y_train10_red[i*500:i*500+500] = y[0:500]
```

[11]:
```
X_train = np.concatenate((X_train10_red, X_train100))
y_train = np.concatenate((y_train10_red, y_train100))
```

[12]:
```
len(X_train)
```

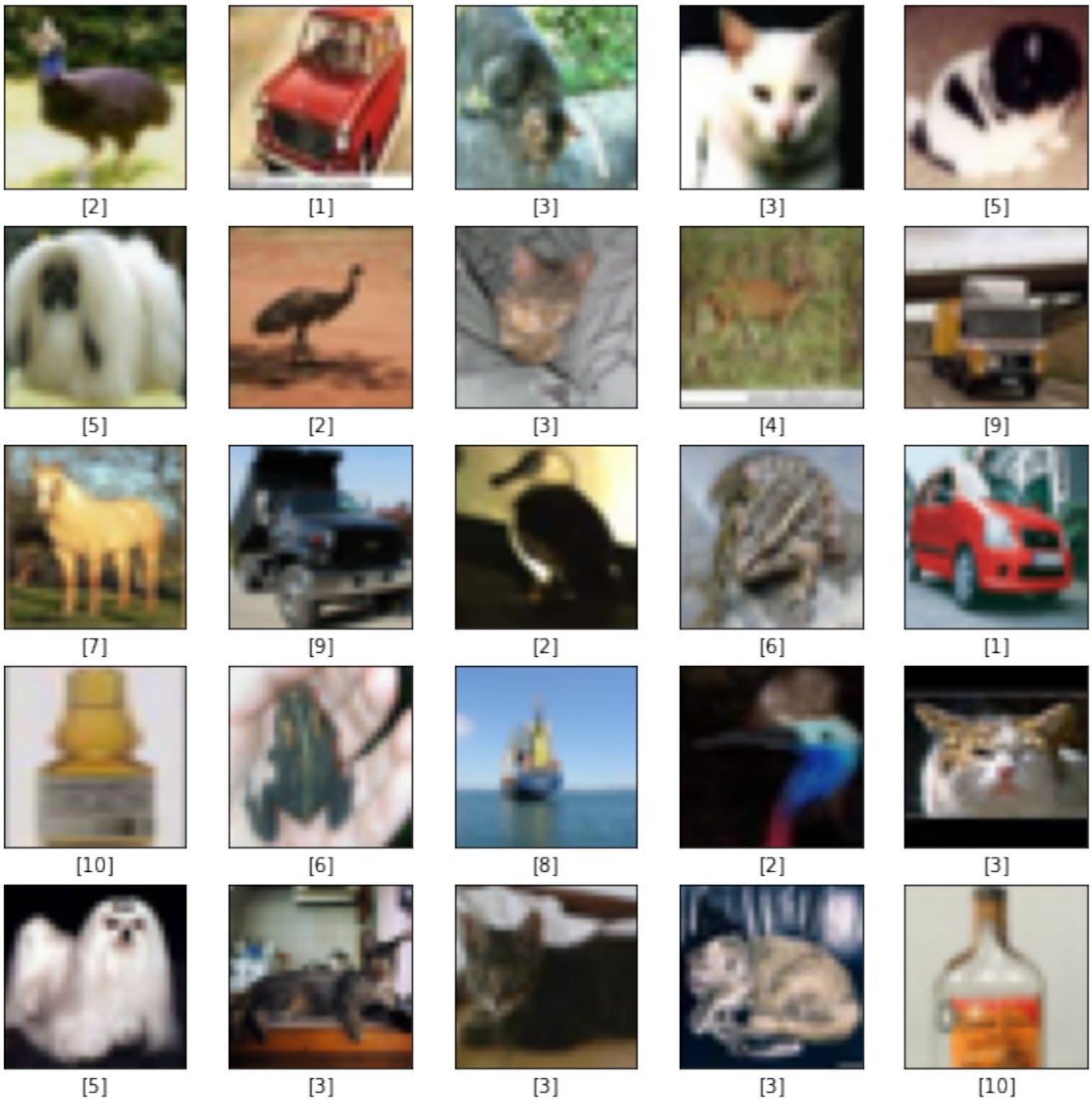
[12]: 5500

[13]:
```
plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(X_train[i+4998], cmap=plt.cm.binary)
    plt.xlabel(y_train[i+4998])
plt.show()
```



[14]:
```
shuffler = np.random.permutation(len(X_train))
X_train  = X_train[shuffler]
y_train  = y_train[shuffler]
```

[15]:
```
plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(X_train[i+4998], cmap=plt.cm.binary)
    plt.xlabel(y_train[i+4998])
plt.show()
```



[16]:
```
X_train[1,1:3,1:3,1]
```

[16]:
```
array([[143, 148],
       [145, 149]], dtype=uint8)
```

[17]:
```
X_train[1,0:3,0:3,1]
```

[17]:
```
array([[131, 140, 145],
       [133, 143, 148],
       [135, 145, 149]], dtype=uint8)
```

[18]:
```
print(X_train.shape)
print(y_train.shape)
```

```
print(y_train[1])
print(X_train[1].shape)
```

```
(5500, 32, 32, 3)
(5500, 1)
[10]
(32, 32, 3)
```

[ ]: