test_dataset_visual

June 12, 2021

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from preprocessing import dataset as ds
from preprocessing import test_dataset as t_ds
from preprocessing import constants as const
```

[2]:
```
training_dataset = ds.Dataset(const.TRAINING_PATH, name="Training",␣
↪include_augmented=True, height=400, width=400, channels=3)
```

```
100%|      | 140/140 [01:28<00:00,  1.57it/s]
100%|      | 8/8 [00:09<00:00,  1.20s/it]

Training Dataset contains 13608 total images.
Training Dataset contains 9855 augmented images.
####################
Processing labels…
####################

including augmented images…
Done processing labels.
```

[3]:
```
# import pandas as pd
# codes, uniques = pd.factorize(training_dataset.optotypes)
# print(codes)
# print(uniques)
```

[4]:
```
# def test_label(dataset_attribute, constant_attribute, size=None):
#     assert np.all(dataset_attribute == constant_attribute)
#     if size:
#         assert np.size(dataset_attribute) == size
#
# print(np.unique(training_dataset.optotypes))
# print(np.unique(training_dataset.acuities))
# test_label(np.unique(training_dataset.optotypes), np.array(const.optotypes))
```

[5]:
```
testing_dataset = ds.Dataset(const.TESTING_PATH, name="Testing", height=400,␣
↪width=400, channels=3)
```

```
# t_ds.test_labels(testing_dataset, "Test")
```

```
100%|      | 140/140 [01:32<00:00,  1.52it/s]
100%|      | 8/8 [00:08<00:00,  1.09s/it]

Testing Dataset contains 12834 total images.
Testing Dataset contains 0 augmented images.
####################
Processing labels…
####################
Done processing labels.
```

[6]:
```
print(len(const.optotypes))
print(np.unique(training_dataset.optotypes).shape)
```

```
59
(1,)
```

[7]:
```
def test_random_index(path, ax, index=None):
    if index is None:
        index = np.random.randint(0, dataset.images.shape[0])

    ax.axis('off')
    ax.imshow(dataset.images[index]/255)
    ax.set_title(
        ␣
 ↪"Acuity={ac}\nCharacter={c}\nOptotype={o}\nAngle={a}\nSize={s}\nDistortion={d}\nAugmentatio
 ↪format(o=dataset.optotypes[index],
                                                                                ␣
 ↪                  ac=dataset.acuities[index],
                                                                                ␣
 ↪                                  c = dataset.character[index],
                                                                                ␣
 ↪     a=dataset.angles[index],
                                                                                ␣
 ↪     s=dataset.sizes[index],
                                                                                ␣
 ↪     d=dataset.distortions[index],
                                                                                ␣
 ↪     ia=dataset.augmented[index]))
```

[8]:
```
testing_acuities = np.load("/Users/brookeryan/Developer/BaldiLab/Visual-Acuity/
 ↪np_data/Testing/acuities.npy", allow_pickle=True)
print(np.unique(testing_acuities))
print(np.unique(testing_acuities).shape)
```

```
['A' 'C' 'E' 'ETDRS' 'ETL-face' 'ETL-x' 'HOTV' 'L' 'NL' 'NPV' 'P' 'SSa'
 'SSl' 'Teller' 'W']
```

```
(15,)
```

1 Testing

[13]:
```
testing_images = np.load("/Users/brookeryan/Developer/BaldiLab/Visual-Acuity/
↪np_data/Testing/images.npy")
acuities = np.load("/Users/brookeryan/Developer/BaldiLab/Visual-Acuity/np_data/
↪Testing/acuities.npy", allow_pickle=True)
angles = np.load("/Users/brookeryan/Developer/BaldiLab/Visual-Acuity/np_data/
↪Testing/angles.npy")
augmented = np.load("/Users/brookeryan/Developer/BaldiLab/Visual-Acuity/np_data/
↪Testing/augmented.npy", allow_pickle=True)
character = np.load("/Users/brookeryan/Developer/BaldiLab/Visual-Acuity/np_data/
↪Testing/character.npy", allow_pickle=True)
distortions = np.load("/Users/brookeryan/Developer/BaldiLab/Visual-Acuity/
↪np_data/Testing/distortions.npy", allow_pickle=True)
optotypes = np.load("/Users/brookeryan/Developer/BaldiLab/Visual-Acuity/np_data/
↪Testing/optotypes.npy", allow_pickle=True)
sizes = np.load("/Users/brookeryan/Developer/BaldiLab/Visual-Acuity/np_data/
↪Testing/sizes.npy", allow_pickle=True)

fig, axs = plt.subplots(4, 4, figsize=(12, 19))

for ax in axs.ravel():
    index = np.random.randint(0, testing_images.shape[0])
    ax.axis('off')
    ax.imshow(testing_images[index]/255)
    ax.set_title(
        ␣
↪"Acuity={ac}\nCharacter={c}\nOptotype={o}\nAngle={a}\nSize={s}\nDistortion={d}\nAugmentatio
↪format(o=optotypes[index],
                                                                                ␣
↪                   ac=acuities[index],
                                                                                ␣
↪                                  c = character[index],
                                                                                ␣
↪      a=angles[index],
                                                                                ␣
↪      s=sizes[index],
                                                                                ␣
↪      d=distortions[index],
                                                                                ␣
↪      ia=augmented[index]))

plt.show()
```

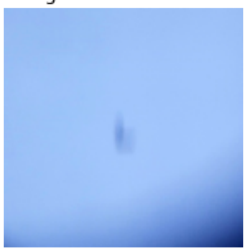
Acuity=C
Character=alpha
Optotype=C
Angle=315.0
Size=S
Distortion=high
Augmentation=None

Acuity=ETDRS
Character=alpha
Optotype=V
Angle=0.0
Size=S
Distortion=high
Augmentation=None

Acuity=C
Character=alpha
Optotype=C
Angle=135.0
Size=S
Distortion=low
Augmentation=None

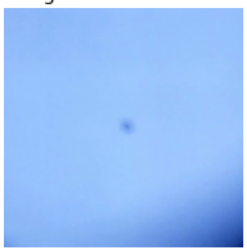
Acuity=P
Character=wingding
Optotype=circle
Angle=0.0
Size=M
Distortion=high
Augmentation=None

Acuity=SSl
Character=alpha
Optotype=L
Angle=0.0
Size=M
Distortion=high
Augmentation=None

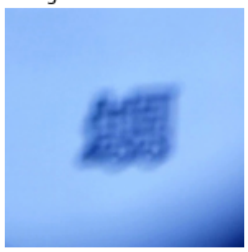


Acuity=A
Character=wingding
Optotype=duck
Angle=0.0
Size=S
Distortion=low
Augmentation=None

Acuity=SSl
Character=alpha
Optotype=D
Angle=0.0
Size=S
Distortion=low
Augmentation=None



Acuity=W
Character=wingding
Optotype=train
Angle=0.0
Size=L
Distortion=high
Augmentation=None



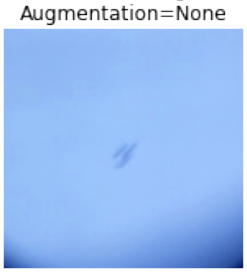
Acuity=SSl
Character=alpha
Optotype=Z
Angle=0.0
Size=S
Distortion=low
Augmentation=None

Acuity=SSl
Character=alpha
Optotype=Z
Angle=0.0
Size=S
Distortion=low
Augmentation=None

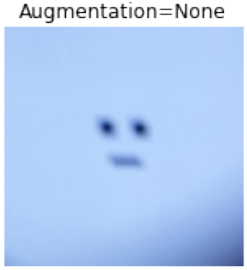
Acuity=C
Character=alpha
Optotype=C
Angle=135.0
Size=S
Distortion=high
Augmentation=None

Acuity=A
Character=wingding
Optotype=horse
Angle=0.0
Size=S
Distortion=low
Augmentation=None

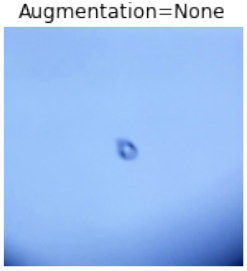
Acuity=ETL-face
Character=wingding
Optotype=smile-line
Angle=0.0
Size=S
Distortion=high
Augmentation=None



Acuity=ETL-face
Character=wingding
Optotype=flat-square
Angle=0.0
Size=L
Distortion=high
Augmentation=None



Acuity=SSl
Character=alpha
Optotype=D
Angle=0.0
Size=M
Distortion=low
Augmentation=None



Acuity=C
Character=alpha
Optotype=C
Angle=90.0
Size=S
Distortion=high
Augmentation=None

2 Training Dataset

[14]:
```
testing_images = np.load("/Users/brookeryan/Developer/BaldiLab/Visual-Acuity/
↪np_data/Training/images.npy")
acuities = np.load("/Users/brookeryan/Developer/BaldiLab/Visual-Acuity/np_data/
↪Training/acuities.npy", allow_pickle=True)
angles = np.load("/Users/brookeryan/Developer/BaldiLab/Visual-Acuity/np_data/
↪Training/angles.npy")
augmented = np.load("/Users/brookeryan/Developer/BaldiLab/Visual-Acuity/np_data/
↪Training/augmented.npy", allow_pickle=True)
character = np.load("/Users/brookeryan/Developer/BaldiLab/Visual-Acuity/np_data/
↪Training/character.npy", allow_pickle=True)
distortions = np.load("/Users/brookeryan/Developer/BaldiLab/Visual-Acuity/
↪np_data/Training/distortions.npy", allow_pickle=True)
optotypes = np.load("/Users/brookeryan/Developer/BaldiLab/Visual-Acuity/np_data/
↪Training/optotypes.npy", allow_pickle=True)
sizes = np.load("/Users/brookeryan/Developer/BaldiLab/Visual-Acuity/np_data/
↪Training/sizes.npy", allow_pickle=True)

fig, axs = plt.subplots(4, 4, figsize=(12, 19))

for ax in axs.ravel():
    index = np.random.randint(0, testing_images.shape[0])
    ax.axis('off')
    ax.imshow(testing_images[index]/255)
    ax.set_title(
        ␣
↪"Acuity={ac}\nCharacter={c}\nOptotype={o}\nAngle={a}\nSize={s}\nDistortion={d}\nAugmentatio
↪format(o=optotypes[index],
                                                                              ␣
↪                   ac=acuities[index],
                                                                              ␣
↪                                  c = character[index],
                                                                              ␣
↪     a=angles[index],
                                                                              ␣
↪     s=sizes[index],
                                                                              ␣
↪     d=distortions[index],
                                                                              ␣
↪     ia=augmented[index]))
```

```
plt.show()
```

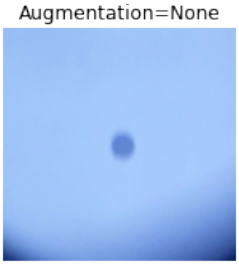
Acuity=ETL-face
Character=wingding
Optotype=smile-square
Angle=0.0
Size=L
Distortion=low
Augmentation=Bright.Png

Acuity=ETL-x
Character=wingding
Optotype=x-square
Angle=0.0
Size=L
Distortion=low
Augmentation=Bright.Png

Acuity=SSl
Character=alpha
Optotype=Z
Angle=0.0
Size=L
Distortion=low
Augmentation=None

Acuity=W
Character=wingding
Optotype=cup
Angle=0.0
Size=L
Distortion=low
Augmentation=Contrast.Png

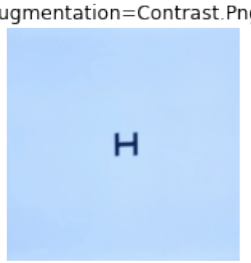
Acuity=Teller
Character=teller
Optotype=SregularCircle
Angle=135.0
Size=S
Distortion=low
Augmentation=None



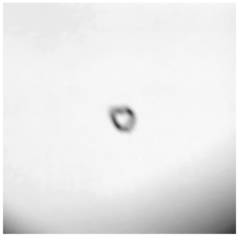
Acuity=E
Character=alpha
Optotype=E
Angle=180.0
Size=L
Distortion=low
Augmentation=None

Acuity=A
Character=wingding
Optotype=tree
Angle=0.0
Size=L
Distortion=low
Augmentation=Contrast.Png

Acuity=HOTV
Character=alpha
Optotype=H
Angle=0.0
Size=L
Distortion=low
Augmentation=Contrast.Png



Acuity=P
Character=wingding
Optotype=apple
Angle=0.0
Size=L
Distortion=low
Augmentation=Gray.Png



Acuity=SSa
Character=alpha
Optotype=P
Angle=0.0
Size=L
Distortion=low
Augmentation=Bright.Png

Acuity=ETL-x
Character=wingding
Optotype=x-diamond
Angle=0.0
Size=L
Distortion=low
Augmentation=Gray.Png

Acuity=P
Character=wingding
Optotype=star
Angle=0.0
Size=L
Distortion=low
Augmentation=Contrast.Png

Acuity=NL
Character=numeric
Optotype=5
Angle=0.0
Size=L
Distortion=low
Augmentation=Bright.Png

Acuity=ETL-x
Character=wingding
Optotype=+blank
Angle=0.0
Size=L
Distortion=low
Augmentation=Contrast.Png

Acuity=NPV
Character=numeric
Optotype=2
Angle=0.0
Size=L
Distortion=low
Augmentation=None

Acuity=W
Character=wingding
Optotype=bird
Angle=0.0
Size=L
Distortion=low
Augmentation=Gray.Png

2.1 Testing Dataset

[ ]:
```
fig, ax = plt.subplots(4, 4, figsize=(12, 19))

for a in ax.ravel():
    test_random_index(testing_dataset, a)

plt.show()
```