

4_det_curve_multinet_analysis

February 6, 2020

1 Generate DET Curves specific to the different subgroups of the Balance Faces in the Wild (BFW) dataset.

Uses the data in data/bfw-datatable.pkl to evaluate DET curves of different attributes.

```
[16]: import pathlib
path_package=f'../'
import sys
if path_package not in sys.path:
    sys.path.append(path_package)
```

```
[17]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(font_scale=1.1)
# Load out custom tool for loading and processing the data
from facebias.iotools import load_bfw_datatable, mkdir
from facebias.visualization import draw_det_curve
from facebias.metrics import calculate_det_curves
%matplotlib inline
```

1.1 Load the data

Read in the data as a pandas.DataFrame and show the first few rows.

```
[18]: # datatable (See Load Data)
dir_data = '../data/bfw-data/bfw/'
dir_features = f'{dir_data}features/sphereface/'
f_datatable = f'{dir_data}meta/bfw-v0.1.5-datatable.pkl'
use_feature = 'sphereface'

dir_results = f"../results/{use_feature}/"
mkdir(dir_results)

data = load_bfw_datatable(f_datatable)
data['score'] = data[use_feature]
data.head()
```

```
/Users/jrobby/WORK/src/facebias/code/facebias/iotools.py:20: UserWarning:
Directory ../../results/sphereface/ exists
warnings.warn(f"Directory {din} exists")
```

```
[18]: fold
0    1  asian_females/n000009/0010_01.jpg ... 0.392526 0.392526
1    1  asian_females/n000009/0010_01.jpg ... 0.354262 0.354262
2    1  asian_females/n000009/0010_01.jpg ... 0.302028 0.302028
3    1  asian_females/n000009/0010_01.jpg ... -0.009217 -0.009217
4    1  asian_females/n000009/0010_01.jpg ... 0.132534 0.132534
```

```
[5 rows x 19 columns]
```

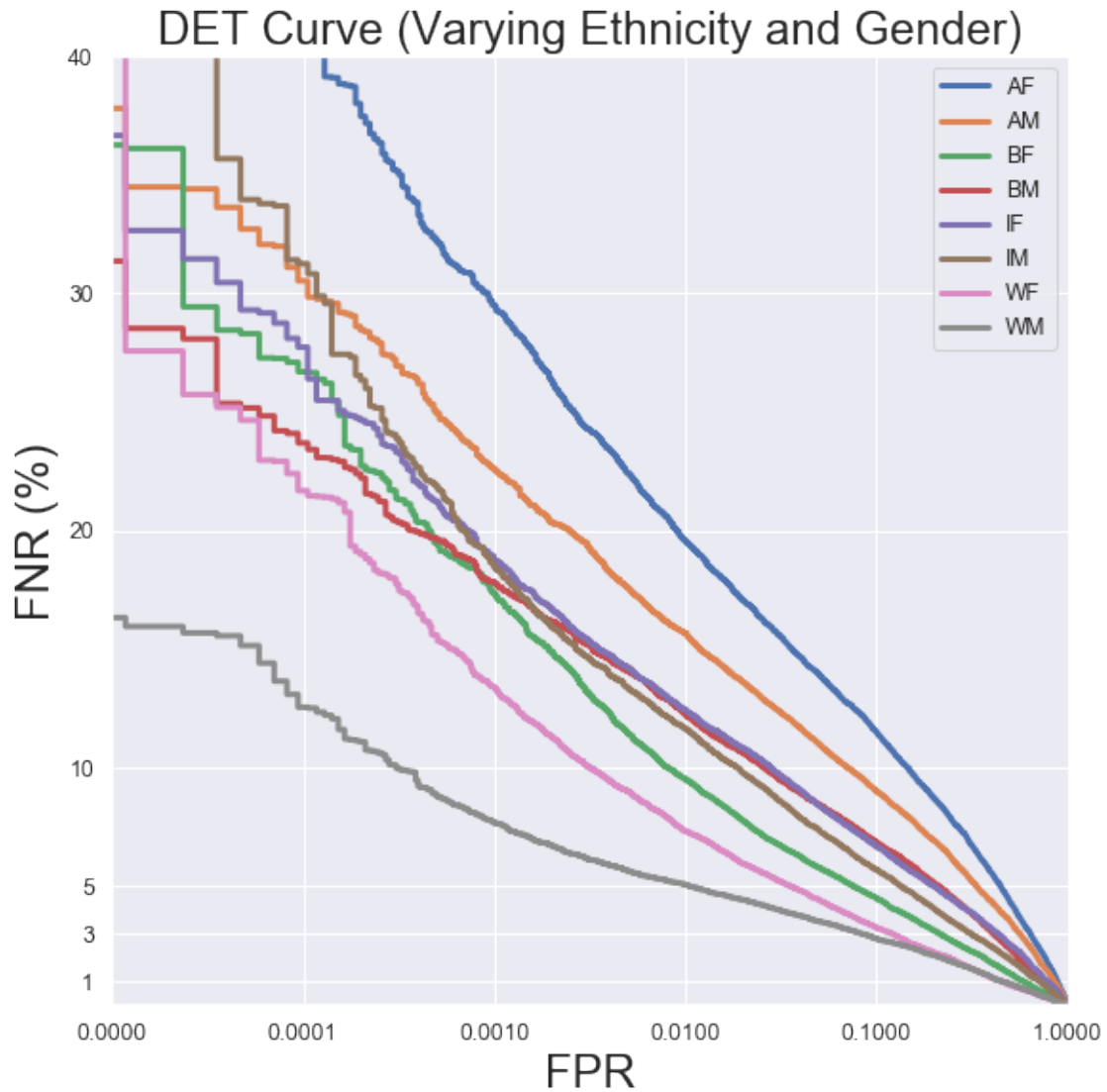
```
[19]: subgroups = data.groupby('a1')
```

```
[29]: rates = {}
li_subgroups = subgroups.groups
fig = plt.figure(figsize=(8, 8), constrained_layout=True)
gs = fig.add_gridspec(1, 3)

ax1 = fig.add_subplot(gs[0, :])
for i, subgroup in enumerate(li_subgroups):
    # for each subgroup
    df = subgroups.get_group(subgroup)
    print('{} pairs'.format(len(df)), subgroup)
    labels, scores = df['label'].values.astype(int), df['score'].values
    det_data = calculate_det_curves(labels, scores)

    # df2 = pd.DataFrame([[x[0], x[1], x[2], [j]] for j, x in
    ↪ enumerate(det_data)])
    ax1 = draw_det_curve(det_data[0], det_data[1], ax=ax1, label=subgroup,
    ↪ title='DET Curve (Varying Ethnicity and Gender)')
plt.savefig(f"{dir_results}det_subgroups.pdf")
```

```
115455 pairs AF
115542 pairs AM
115364 pairs BF
115459 pairs BM
115595 pairs IF
115458 pairs IM
115536 pairs WF
115489 pairs WM
```



```
[34]: import numpy as np
from matplotlib.ticker import ScalarFormatter

subgroups = data.groupby('g1')
li_subgroups = subgroups.groups
scale=1
fig, ax2 = plt.subplots(1,1)
label_x="FPR"
label_y="FNR (%)"
ticks_to_use_x=(1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1e-0)
ticks_to_use_y=(0.01, 0.03, 0.05, 0.10, 0.20, 0.30, 0.40, 0.50)
fontsize=24
title='DET Curve (Varying Gender)'
```

```

for i, subgroup in enumerate(li_subgroups):
    # per gender
    df = subgroups.get_group(subgroup)

    scores, labels = df['senet50'], df['label'].values.astype(int)

    det_data = calculate_det_curves(labels, scores)
    # rates[subgroup] = pd.DataFrame([[x[0], x[1], x[2]] for x in det_data])
    print('Completed {} \n'.format(subgroup))

    ax2.plot(det_data[0], det_data[1] * scale, label=subgroup, linewidth=3)
    ax2.set_xscale("log")
    ax2.get_xaxis().set_major_formatter(ScalarFormatter())
    ax2.get_yaxis().set_major_formatter(ScalarFormatter())
    ax2.set_xticks(ticks_to_use_x)
    ax2.set_yticks(scale * np.array(ticks_to_use_y))

    # add 10% to upper ylimit
    ax2.set_ylim(0.00, scale * np.max(ticks_to_use_y))
    ax2.set_xlim(np.min(ticks_to_use_x), np.max(ticks_to_use_x))
    ax2.set_xlabel(label_x, fontsize=fontsize)
    ax2.set_ylabel(label_y, fontsize=fontsize)

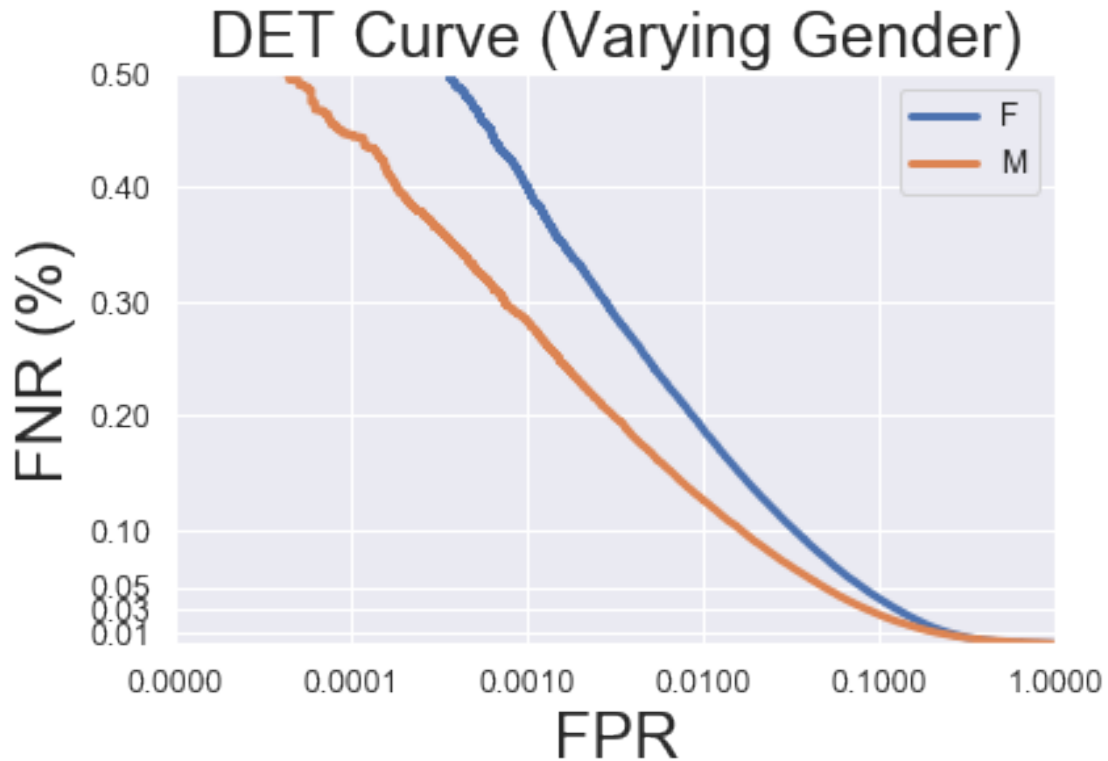
    ax2.legend(loc="best")
    ax2.set_title(title, fontsize=fontsize)
    # for i, cat in enumerate(np.unique(li_subgroups)):
    #     ax2 = draw_det_curve(df[0].values, df[1].values, ax=ax2, label=cat,
    ↪ set_axis_log_x=True, title='DET Curve (Varying Gender)')

plt.savefig(f"{dir_results}det_genders.pdf")

```

Completed F

Completed M



```
[40]: rates = {}
subgroups = data.groupby('a1')
li_subgroups = subgroups.groups
fig, axs = plt.subplots(1, 3, figsize=(22, 7), constrained_layout=True,
    ↳facecolor='white', sharey=True)
# gs = fig.add_gridspec(1, 3)

# ax1 = fig.add_subplot(gs[0, :])
models = ['vgg16', 'resnet50', 'senet50']

obin=dir_results + "det_{}.pkl"
#obin="inter/{}/{}_ax.pkl"
for i, subgroup in enumerate(li_subgroups):
    # for each subgroup

    df = subgroups.get_group(subgroup)
    labels = df['label'].values.astype(int)
    print('{} pairs'.format(len(df)))
    for k, model in enumerate(models):
        print(f"processing {model}")
        scores = df[model].values
        det_data = calculate_det_curves(labels, scores)
```

```

        pd.to_pickle(det_data, obin.format(model, subgroup))
        #         df2 = pd.DataFrame([[x[0], x[1], x[2], y_formatted[j]] for j,
↪x in enumerate(det_data)])
        axs[k] = draw_det_curve(det_data[0], det_data[1], ax=axs[k],
↪label=subgroup, title=model)

fig.savefig('SDMs_supplemental.png', facecolor=fig.get_facecolor(),
↪edgecolor='none', transparent=True)
fig.savefig('SDMs_supplemental.pdf', facecolor=fig.get_facecolor(),
↪edgecolor='none', transparent=True)

```

```

115455 pairs
processing vgg16
processing resnet50
processing senet50
115442 pairs
processing vgg16
processing resnet50
processing senet50
115364 pairs
processing vgg16
processing resnet50
processing senet50
115459 pairs
processing vgg16
processing resnet50
processing senet50
115595 pairs
processing vgg16
processing resnet50
processing senet50
115458 pairs
processing vgg16
processing resnet50
processing senet50
115536 pairs
processing vgg16
processing resnet50
processing senet50
115489 pairs
processing vgg16
processing resnet50
processing senet50

```

