

```
In [ ]: from feat import Detector
        from skimage import io, img_as_ubyte
        import numpy as np
        import pandas as pd
        import os
```

```
In [ ]: detector = Detector(
        face_model="retinaface",
        landmark_model="mobilefacenet",
        au_model='xgb',
        emotion_model="resmasknet",
        facepose_model="img2pose",
    )
```

Angry

```
In [ ]: angry_images_dir = 'Images/Angry'
        angry_image_files = os.listdir(angry_images_dir)
        angry_image_paths = [os.path.join(angry_images_dir, file)
                               for file in angry_image_files if file.endswith(('.jpg'))]
```

```
In [ ]: features_dict = {
        'label': [],
        'FaceRectX': [],
        'FaceRectY': [],
        'FaceRectWidth': [],
        'FaceRectHeight': [],
        'FaceScore': [],
        'Pitch': [],
        'Roll': [],
        'Yaw': [],
    }
```

```
In [ ]: for i, image_path in enumerate(angry_image_paths, 1):
        image = img_as_ubyte(io.imread(image_path))

        single_face_prediction = detector.detect_image(image_path)

        faceboxes = single_face_prediction.faceboxes

        faceboxes_x = faceboxes['FaceRectX']
        faceboxes_x_mean = round(np.mean(faceboxes_x), 5)

        faceboxes_y = faceboxes['FaceRectY']
```

```

faceboxes_y_mean = round(np.mean(faceboxes_y), 5)

faceboxes_w = faceboxes['FaceRectWidth']
faceboxes_w_mean = round(np.mean(faceboxes_w), 5)

faceboxes_h = faceboxes['FaceRectHeight']
faceboxes_h_mean = round(np.mean(faceboxes_h), 5)

faceboxes_s = faceboxes['FaceScore']
faceboxes_s_mean = round(np.mean(faceboxes_s), 5)

aus = single_face_prediction.aus
aus_mean = np.mean(aus, axis=0)

for j, au in enumerate(aus_mean):
    feature_name = f'AUs{j+1}'
    if feature_name not in features_dict:
        features_dict[feature_name] = []
    features_dict[feature_name].append(round(au,5))

landmarks = single_face_prediction.landmarks
landmarks_mean = np.mean(landmarks, axis=0)
num_landmarks = len(landmarks_mean) // 2
for j in range(num_landmarks):
    feature_name_x = f'x_{j}'
    feature_name_y = f'y_{j}'
    if feature_name_x not in features_dict:
        features_dict[feature_name_x] = []
    if feature_name_y not in features_dict:
        features_dict[feature_name_y] = []
    features_dict[feature_name_x].append(round(landmarks_mean[2 * j],5))
    features_dict[feature_name_y].append(round(landmarks_mean[2 * j + 1],5))

poses = single_face_prediction.poses

poses_pitch = poses['Pitch']
poses_pitch_mean = round(np.mean(poses_pitch), 5)
poses_roll = poses['Roll']
poses_roll_mean = round(np.mean(poses_roll), 5)

poses_yaw = poses['Yaw']
poses_yaw_mean = round(np.mean(poses_yaw), 5)

features_dict['FaceRectX'].append(faceboxes_x_mean)
features_dict['FaceRectY'].append(faceboxes_y_mean)
features_dict['FaceRectWidth'].append(faceboxes_w_mean)
features_dict['FaceRectHeight'].append(faceboxes_h_mean)
features_dict['FaceScore'].append(faceboxes_s_mean)

```

```
features_dict['Pitch'].append(poses_pitch_mean)
features_dict['Roll'].append(poses_roll_mean)
features_dict['Yaw'].append(poses_yaw_mean)
```

```
features_dict['label'].append(f'Angry')
```

```
features = pd.DataFrame(features_dict)
features.to_csv("features.csv", index=False)
print(features)
```

```
100%|██████████| 1/1 [00:02<00:00, 2.65s/it]
100%|██████████| 1/1 [00:01<00:00, 1.39s/it]
100%|██████████| 1/1 [00:01<00:00, 1.26s/it]
100%|██████████| 1/1 [00:01<00:00, 1.26s/it]
100%|██████████| 1/1 [00:01<00:00, 1.26s/it]
100%|██████████| 1/1 [00:01<00:00, 1.27s/it]
100%|██████████| 1/1 [00:01<00:00, 1.27s/it]
100%|██████████| 1/1 [00:01<00:00, 1.24s/it]
100%|██████████| 1/1 [00:01<00:00, 1.25s/it]
100%|██████████| 1/1 [00:01<00:00, 1.23s/it]
100%|██████████| 1/1 [00:01<00:00, 1.24s/it]
100%|██████████| 1/1 [00:01<00:00, 1.26s/it]
100%|██████████| 1/1 [00:01<00:00, 1.23s/it]
100%|██████████| 1/1 [00:01<00:00, 1.24s/it]
100%|██████████| 1/1 [00:01<00:00, 1.25s/it]
100%|██████████| 1/1 [00:01<00:00, 1.25s/it]
100%|██████████| 1/1 [00:01<00:00, 1.30s/it]
100%|██████████| 1/1 [00:01<00:00, 1.23s/it]
100%|██████████| 1/1 [00:01<00:00, 1.25s/it]
100%|██████████| 1/1 [00:01<00:00, 1.25s/it]
100%|██████████| 1/1 [00:01<00:00, 1.24s/it]
100%|██████████| 1/1 [00:01<00:00, 1.24s/it]
100%|██████████| 1/1 [00:01<00:00, 1.25s/it]
100%|██████████| 1/1 [00:01<00:00, 1.26s/it]
100%|██████████| 1/1 [00:01<00:00, 1.26s/it]
```

	label	FaceRectX	FaceRectY	FaceRectWidth	FaceRectHeight	FaceScore	\
0	Angry	-0.31715	-0.29662	36.38660	47.85770	0.99359	
1	Angry	3.02009	-1.33738	42.36711	50.01836	0.95564	
2	Angry	0.98461	-1.29084	35.94039	47.50459	0.73074	
3	Angry	5.36803	-0.93098	32.32775	44.01271	0.98712	
4	Angry	2.61129	-0.59984	42.42387	48.76449	0.98044	
5	Angry	4.24402	-0.15019	37.94941	49.34007	0.65269	
6	Angry	3.88712	-1.28848	40.27287	48.47514	0.98098	
7	Angry	8.69832	5.24490	37.02285	44.35610	0.91126	
8	Angry	4.63346	1.33216	40.24331	46.02695	0.88391	
9	Angry	6.43628	3.32476	35.02141	41.43999	0.99732	
10	Angry	7.24802	1.73065	27.49158	37.13423	0.97093	
11	Angry	7.52906	0.23670	34.44239	42.73464	0.99285	
12	Angry	1.54413	0.80602	42.22838	48.80264	0.78201	
13	Angry	3.52362	0.41539	38.94727	50.97354	0.99138	
14	Angry	3.76737	0.57034	41.66933	47.60431	0.97011	
15	Angry	4.47786	1.44926	38.66001	48.08429	0.65821	
16	Angry	13.29078	5.04290	28.60094	36.88139	0.97645	
17	Angry	12.01175	16.92690	22.96819	28.01377	0.99249	
18	Angry	8.33563	2.12213	34.33430	42.01813	0.87734	
19	Angry	6.45628	0.11393	37.56089	45.77484	0.99698	
20	Angry	11.19520	8.35673	30.18993	36.90687	0.99765	
21	Angry	-0.88786	1.56034	40.29081	42.89886	0.75447	
22	Angry	5.03740	0.35370	38.57442	48.43821	0.98809	
23	Angry	4.78840	1.70670	35.95501	44.68795	0.97432	
24	Angry	6.64979	-0.07352	36.60447	44.95799	0.98797	

	Pitch	Roll	Yaw	AUs1	...	x_63	y_63	x_64	\
0	-9.59011	12.72161	-28.42508	0.29705	...	51.34946	49.90415	47.72598	
1	6.27818	1.67929	-0.37270	0.31780	...	57.55396	54.86165	50.38312	
2	-64.88136	-61.70474	70.54467	0.77023	...	48.56225	47.52279	46.07655	
3	-8.59568	-12.81338	0.20218	0.33113	...	45.59402	43.52846	39.25653	
4	6.68759	7.24873	-4.51140	0.36227	...	47.19149	46.04761	44.87098	
5	3.89649	7.73664	-1.51762	0.41161	...	43.64235	42.97847	41.74977	
6	-11.20104	-17.62932	1.40134	0.35615	...	51.79514	51.34820	49.76644	
7	14.95990	4.91897	9.62562	0.78737	...	46.60971	45.98521	44.99949	
8	-2.42957	-0.39625	1.98246	0.28441	...	48.75859	47.73665	46.17806	
9	-16.35832	8.01615	-1.86663	0.31079	...	39.46017	38.11435	36.12073	
10	4.80803	-5.63905	-0.16728	0.32574	...	38.64256	37.23975	33.52383	
11	-5.16738	-1.35247	-6.14091	0.44408	...	45.46536	42.95829	38.41932	
12	2.44478	-1.00455	-6.45561	0.27606	...	49.24510	48.05542	45.57366	
13	-7.68378	1.84850	1.59962	0.41540	...	54.41248	53.16613	51.56366	
14	6.75912	-7.27790	1.40129	0.33000	...	47.64507	46.96100	45.62711	
15	6.83713	4.05712	-3.39196	0.52369	...	51.21891	48.59579	44.27207	
16	4.92096	5.16002	13.15131	0.29374	...	35.98997	35.36502	34.56941	
17	-15.20225	0.04250	4.02826	0.47835	...	44.89959	44.02676	43.06398	
18	-22.83625	-7.64782	2.60944	0.41871	...	45.51139	42.94252	38.65132	
19	-10.26012	-2.79798	-2.61570	0.36370	...	46.79287	44.88459	41.73058	

20	-9.37936	9.33075	-5.07646	0.25355	...	41.24019	40.03034	38.23332
21	3.26512	-27.88666	-10.02206	0.52776	...	45.21367	44.21977	40.01105
22	-6.16340	-1.69232	0.95592	0.32357	...	50.25191	49.37100	47.67476
23	7.21778	-5.13591	-1.68029	0.64227	...	41.26955	41.15530	40.47839
24	7.17466	9.49693	8.65224	0.43612	...	42.12746	41.24167	39.85636

	y_64	x_65	y_65	x_66	y_66	x_67	y_67
0	47.68770	47.96588	47.76735	48.01969	48.09647	48.30843	48.04583
1	49.71835	50.09810	49.95040	51.19756	53.98301	54.36479	53.80264
2	45.22688	45.06554	44.50404	44.63741	45.66631	46.05116	46.15090
3	37.68587	37.52354	36.71322	36.54344	41.71583	42.67867	42.73752
4	44.79318	45.30472	45.38649	46.53598	45.56435	45.38419	44.92821
5	40.88410	41.30558	41.24106	42.75518	41.76462	41.76067	41.38189
6	48.65210	48.27772	47.19686	45.29603	47.28954	48.26158	48.64084
7	44.22962	44.40324	44.20072	45.05677	44.97989	45.04771	44.86577
8	45.45605	45.78980	45.58985	46.15471	45.85681	45.94684	45.65118
9	36.85123	37.22140	37.04634	36.77770	37.40551	37.53893	37.18070
10	29.08769	28.76539	28.24817	30.55471	36.11993	36.93891	37.05071
11	36.19891	36.29893	35.88897	37.47193	42.50791	43.11184	42.85235
12	44.40465	44.45625	43.92182	43.83859	46.19634	46.73145	46.69814
13	51.31637	51.73897	51.59297	52.07527	51.42591	51.52890	51.17169
14	44.00383	43.92711	43.45686	43.65153	43.93018	44.36738	44.44405
15	42.84952	43.22702	43.21779	45.31366	47.99311	48.11837	47.56531
16	34.61540	34.85642	34.79008	35.01005	34.89838	34.91278	34.67344
17	43.31449	43.60100	43.51150	43.36295	43.47693	43.54460	43.31999
18	37.40900	37.59622	37.19710	37.95054	42.94607	43.46535	43.09045
19	41.39934	41.59860	41.04564	40.60460	43.51867	44.06621	43.80541
20	38.83849	39.11050	38.98660	38.73775	39.08933	39.20126	38.96998
21	34.70711	33.68788	32.41457	32.57708	39.87747	41.64611	42.59338
22	46.96706	47.11603	46.77174	46.82745	47.42244	47.75433	47.62721
23	39.38225	39.44393	39.21420	39.83225	39.46615	39.62092	39.60027
24	39.63242	40.00556	40.15368	41.48679	40.30880	40.13307	39.77514

[25 rows x 165 columns]

Disgusted

```
In [ ]: disgusted_images_dir = 'Images/Disgusted'
disgusted_image_files = os.listdir(disgusted_images_dir)
disgusted_image_paths = [os.path.join(disgusted_images_dir, file) for file in disgusted_image_files if file.endswith((''.jpg'))]
```

```
In [ ]: features_dict = {
    'label': [],
    'FaceRectX': [],
    'FaceRectY': [],
    'FaceRectWidth': [],
    'FaceRectHeight': [],
```

```

'FaceScore': [],
'Pitch': [],
'Roll': [],
'Yaw': [],
}

```

```

In [ ]: for i, image_path in enumerate(disgusted_image_paths, 1):
        image = img_as_ubyte(io.imread(image_path))

        single_face_prediction = detector.detect_image(image_path)

        faceboxes = single_face_prediction.faceboxes

        faceboxes_x = faceboxes['FaceRectX']
        faceboxes_x_mean = round(np.mean(faceboxes_x), 5)

        faceboxes_y = faceboxes['FaceRectY']
        faceboxes_y_mean = round(np.mean(faceboxes_y), 5)

        faceboxes_w = faceboxes['FaceRectWidth']
        faceboxes_w_mean = round(np.mean(faceboxes_w), 5)

        faceboxes_h = faceboxes['FaceRectHeight']
        faceboxes_h_mean = round(np.mean(faceboxes_h), 5)

        faceboxes_s = faceboxes['FaceScore']
        faceboxes_s_mean = round(np.mean(faceboxes_s), 5)

        aus = single_face_prediction.aus
        aus_mean = np.mean(aus, axis=0)

        for j, au in enumerate(aus_mean):
            feature_name = f'AUs{j+1}'
            if feature_name not in features_dict:
                features_dict[feature_name] = []
            features_dict[feature_name].append(round(au,5))

        landmarks = single_face_prediction.landmarks
        landmarks_mean = np.mean(landmarks, axis=0)
        num_landmarks = len(landmarks_mean) // 2
        for j in range(num_landmarks):
            feature_name_x = f'x_{j}'
            feature_name_y = f'y_{j}'
            if feature_name_x not in features_dict:
                features_dict[feature_name_x] = []
            if feature_name_y not in features_dict:
                features_dict[feature_name_y] = []

```

```

        features_dict[feature_name_x].append(round(landmarks_mean[2 * j],5))
        features_dict[feature_name_y].append(round(landmarks_mean[2 * j + 1],5))

poses = single_face_prediction.poses

poses_pitch = poses['Pitch']
poses_pitch_mean = round(np.mean(poses_pitch), 5)
poses_roll = poses['Roll']
poses_roll_mean = round(np.mean(poses_roll), 5)

poses_yaw = poses['Yaw']
poses_yaw_mean = round(np.mean(poses_yaw), 5)

features_dict['FaceRectX'].append(faceboxes_x_mean)
features_dict['FaceRectY'].append(faceboxes_y_mean)
features_dict['FaceRectWidth'].append(faceboxes_w_mean)
features_dict['FaceRectHeight'].append(faceboxes_h_mean)
features_dict['FaceScore'].append(faceboxes_s_mean)
features_dict['Pitch'].append(poses_pitch_mean)
features_dict['Roll'].append(poses_roll_mean)
features_dict['Yaw'].append(poses_yaw_mean)

features_dict['label'].append(f'Disgusted')

new_features = pd.DataFrame(features_dict)
features = pd.read_csv('features.csv')

updated_features = pd.concat([features, new_features], ignore_index=True)
updated_features.to_csv('features.csv', index=False)

print(new_features)

```

100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.27s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.25s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.24s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.39s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.23s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.23s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.24s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.23s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.25s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.24s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.28s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.35s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.85s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.45s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.42s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.56s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.49s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.31s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.30s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.38s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.42s/it]

	label	FaceRectX	FaceRectY	FaceRectWidth	FaceRectHeight	FaceScore	\
0	Disgusted	3.67244	-2.86092	39.78965	53.03853	0.97566	
1	Disgusted	1.46437	-0.53303	44.98455	44.81738	0.92433	
2	Disgusted	6.72449	1.07800	35.41726	45.39867	0.99173	
3	Disgusted	1.77770	-1.85181	39.63944	54.59731	0.77397	
4	Disgusted	2.66361	-3.00449	39.54651	52.66926	0.99147	
5	Disgusted	-1.11882	3.62277	38.65945	45.04558	0.84483	
6	Disgusted	1.28284	1.06802	41.44248	47.35446	0.96012	
7	Disgusted	6.77747	2.19426	38.46776	47.18093	0.99443	
8	Disgusted	9.77997	0.38685	39.63224	46.93843	0.91829	
9	Disgusted	4.40676	0.61368	39.15677	48.03072	0.98048	
10	Disgusted	2.60978	-1.16448	41.69537	50.11109	0.97394	
11	Disgusted	8.53950	2.48090	36.57729	46.05894	0.90283	
12	Disgusted	2.11371	-2.12910	41.59076	54.22998	0.96881	
13	Disgusted	5.57215	-2.22912	39.41005	50.92660	0.98581	
14	Disgusted	1.57236	-3.21281	39.91403	50.88123	0.97252	
15	Disgusted	5.30713	-1.34392	41.72202	51.51230	0.92616	
16	Disgusted	3.64915	2.75790	38.11914	48.91744	0.71055	
17	Disgusted	5.58605	-0.73406	38.33005	48.67450	0.95435	
18	Disgusted	6.41792	1.47051	35.76351	43.61426	0.99532	
19	Disgusted	5.65244	-0.64167	35.68320	46.38793	0.99680	
20	Disgusted	5.40614	4.71897	32.80795	43.03106	0.96264	
21	Disgusted	2.80908	-0.80217	40.53120	51.72324	0.98793	
22	Disgusted	3.95236	-1.87348	39.91628	51.28874	0.98365	
23	Disgusted	3.32160	-1.52109	37.75612	49.17039	0.99386	
24	Disgusted	4.29430	0.58010	37.35761	45.86472	0.99602	

	Pitch	Roll	Yaw	AUs1	...	x_63	y_63	x_64	\
0	-9.56717	-3.83185	7.01300	0.32893	...	54.63247	52.61866	50.08278	
1	-35.39299	3.88627	9.58490	0.44983	...	47.71506	44.04174	39.87969	
2	-21.22395	-5.94628	8.62887	0.49565	...	48.00812	46.16761	43.60552	
3	-3.36627	2.17326	-6.60581	0.26388	...	60.86683	59.25714	57.08025	
4	14.07332	9.81538	-10.49145	0.57245	...	58.31362	54.78094	49.03667	
5	-4.92903	7.90453	-19.92973	0.36113	...	46.06714	44.93354	43.19956	
6	3.33375	-7.49320	1.06179	0.43633	...	53.23091	51.47567	47.78760	
7	3.86848	3.81918	7.18356	0.49720	...	47.18563	45.73075	43.88363	
8	-17.61274	-19.45959	10.98470	0.52427	...	49.56175	48.41494	46.53475	
9	-3.55883	-5.84968	2.94416	0.30137	...	52.32405	50.45175	47.49661	
10	-9.13035	-0.92223	2.91008	0.47588	...	54.65781	52.95679	50.48003	
11	-4.71994	-1.05876	-0.38149	0.46257	...	47.77144	46.32894	44.34144	
12	-6.35164	-9.41243	-2.58276	0.38655	...	61.98087	60.35726	56.71263	
13	-12.68827	-5.33434	10.19908	0.42035	...	55.16019	53.37796	50.26382	
14	-12.27277	-4.55980	-3.63104	0.19605	...	50.59738	49.45359	47.83793	
15	-14.75054	-11.70720	15.84461	0.53401	...	57.11906	55.55451	52.61066	
16	-4.06410	-1.39119	0.75153	0.30048	...	50.61016	48.97461	46.80119	
17	-15.39749	-4.88372	2.15868	0.23818	...	52.01534	50.48860	47.43021	
18	-4.47842	-0.32286	2.22066	0.50922	...	44.55564	43.29892	41.56894	
19	-13.69244	0.05627	2.22984	0.24492	...	48.64807	46.07461	42.35325	

20	4.58323	-7.17616	-4.95071	0.42461	...	44.73516	43.90304	41.93166
21	4.00553	-1.02608	-4.36244	0.38479	...	58.00818	55.01254	50.10465
22	-6.27671	-1.10728	1.91704	0.33580	...	54.94853	53.69062	51.66825
23	-2.14452	-4.23696	-3.18468	0.25046	...	50.72744	50.11065	48.85313
24	-22.80353	-4.01400	2.57413	0.53372	...	49.32125	46.71643	42.65278

	y_64	x_65	y_65	x_66	y_66	x_67	y_67
0	50.82630	51.26859	50.88686	50.12389	51.54451	51.83020	51.39631
1	42.79729	43.63680	43.53824	42.05979	46.24931	46.26375	45.33977
2	43.77835	43.93521	43.32473	42.02476	44.87944	45.44489	45.21264
3	57.53056	57.92608	57.41670	56.71110	57.64825	58.09306	57.70312
4	45.80219	46.35056	46.67089	51.22629	55.69970	55.72060	54.91704
5	42.66475	42.99156	42.88852	43.56658	43.56974	43.67398	43.32839
6	45.35225	45.14753	44.48555	44.76022	48.87741	49.76339	49.90611
7	43.59676	44.17129	44.35864	45.97696	44.62664	44.41416	43.90664
8	45.80120	45.66706	44.79437	43.10606	45.20729	46.03979	46.16679
9	46.64107	46.74819	46.14229	45.70141	48.72582	49.37171	49.25478
10	49.96996	50.22907	49.60991	49.21259	51.40433	51.97934	51.74382
11	44.86177	45.22577	44.86463	44.34845	45.12980	45.36852	45.09068
12	54.99483	54.63396	53.44136	51.68475	56.19640	57.50535	57.81184
13	49.14427	49.23290	48.62138	48.54645	51.37122	52.01908	51.91241
14	47.56574	47.76640	47.23009	46.56303	47.49473	47.97154	47.80766
15	51.47498	51.45906	50.80854	50.48178	52.54563	53.26040	53.23861
16	46.56318	46.76534	46.25821	45.69884	46.99125	47.44562	47.27286
17	46.90592	47.02370	46.65177	46.62023	48.04613	48.48865	48.32466
18	41.17984	41.40099	41.00825	41.03264	41.87635	42.20820	41.98115
19	42.82450	43.19354	42.79855	42.27372	45.71173	46.16253	45.75944
20	39.91131	39.80682	39.41985	40.19553	41.65201	42.06970	42.12818
21	48.70172	49.03253	48.78572	50.24473	55.09487	55.57557	55.14788
22	50.87680	51.03202	50.59124	50.44925	51.23684	51.67542	51.57384
23	47.77114	47.67240	46.94133	46.01764	47.10246	47.75823	47.87715
24	42.35615	42.58532	42.02150	41.74629	46.71186	47.32343	46.96868

[25 rows x 165 columns]

Fear

```
In [ ]: fear_images_dir = 'Images/Fear'
fear_image_files = os.listdir(fear_images_dir)
fear_image_paths = [os.path.join(fear_images_dir, file) for file in fear_image_files if file.endswith(('.jpg'))]
```

```
In [ ]: features_dict = {
    'label': [],
    'FaceRectX': [],
    'FaceRectY': [],
    'FaceRectWidth': [],
    'FaceRectHeight': [],
```

```

'FaceScore': [],
'Pitch': [],
'Roll': [],
'Yaw': [],
}

```

```

In [ ]: for i, image_path in enumerate(fear_image_paths, 1):
        image = img_as_ubyte(io.imread(image_path))

        single_face_prediction = detector.detect_image(image_path)

        faceboxes = single_face_prediction.faceboxes

        faceboxes_x = faceboxes['FaceRectX']
        faceboxes_x_mean = round(np.mean(faceboxes_x), 5)

        faceboxes_y = faceboxes['FaceRectY']
        faceboxes_y_mean = round(np.mean(faceboxes_y), 5)

        faceboxes_w = faceboxes['FaceRectWidth']
        faceboxes_w_mean = round(np.mean(faceboxes_w), 5)

        faceboxes_h = faceboxes['FaceRectHeight']
        faceboxes_h_mean = round(np.mean(faceboxes_h), 5)

        faceboxes_s = faceboxes['FaceScore']
        faceboxes_s_mean = round(np.mean(faceboxes_s), 5)

        aus = single_face_prediction.aus
        aus_mean = np.mean(aus, axis=0)

        for j, au in enumerate(aus_mean):
            feature_name = f'AUs{j+1}'
            if feature_name not in features_dict:
                features_dict[feature_name] = []
            features_dict[feature_name].append(round(au,5))

        landmarks = single_face_prediction.landmarks
        landmarks_mean = np.mean(landmarks, axis=0)
        num_landmarks = len(landmarks_mean) // 2
        for j in range(num_landmarks):
            feature_name_x = f'x_{j}'
            feature_name_y = f'y_{j}'
            if feature_name_x not in features_dict:
                features_dict[feature_name_x] = []
            if feature_name_y not in features_dict:
                features_dict[feature_name_y] = []

```

```

        features_dict[feature_name_x].append(round(landmarks_mean[2 * j],5))
        features_dict[feature_name_y].append(round(landmarks_mean[2 * j + 1],5))

    poses = single_face_prediction.poses

    poses_pitch = poses['Pitch']
    poses_pitch_mean = round(np.mean(poses_pitch), 5)
    poses_roll = poses['Roll']
    poses_roll_mean = round(np.mean(poses_roll), 5)

    poses_yaw = poses['Yaw']
    poses_yaw_mean = round(np.mean(poses_yaw), 5)

    features_dict['FaceRectX'].append(faceboxes_x_mean)
    features_dict['FaceRectY'].append(faceboxes_y_mean)
    features_dict['FaceRectWidth'].append(faceboxes_w_mean)
    features_dict['FaceRectHeight'].append(faceboxes_h_mean)
    features_dict['FaceScore'].append(faceboxes_s_mean)
    features_dict['Pitch'].append(poses_pitch_mean)
    features_dict['Roll'].append(poses_roll_mean)
    features_dict['Yaw'].append(poses_yaw_mean)

    features_dict['label'].append('Fear')

new_features = pd.DataFrame(features_dict)
features = pd.read_csv('features.csv')

updated_features = pd.concat([features, new_features], ignore_index=True)
updated_features.to_csv('features.csv', index=False)

print(new_features)

```

100%	<div></div>	1/1	[00:01<00:00,	1.36s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.31s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.30s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.29s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.30s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.30s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.25s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.25s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.24s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.25s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.32s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.27s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.25s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.37s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.27s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.25s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.24s/it]

	label	FaceRectX	FaceRectY	FaceRectWidth	FaceRectHeight	FaceScore	\
0	Fear	8.23405	3.43177	32.38232	40.28793	0.50958	
1	Fear	3.87996	0.02295	37.05399	49.72417	0.97093	
2	Fear	4.65876	-0.06087	40.66630	52.80577	0.68076	
3	Fear	1.01737	-1.27954	40.76603	50.86104	0.98982	
4	Fear	7.03450	0.81813	39.50064	46.25839	0.97211	
5	Fear	5.73775	-0.32898	34.83064	46.66260	0.98311	
6	Fear	6.89291	0.85074	38.05181	46.41454	0.97719	
7	Fear	3.80139	5.02888	33.16196	42.17642	0.94821	
8	Fear	4.51768	1.12561	35.73444	45.44044	0.99678	
9	Fear	1.90162	-0.38750	39.02562	49.84409	0.91000	
10	Fear	5.82308	-0.13716	38.88869	49.91063	0.99165	
11	Fear	2.39117	3.99527	36.98512	43.69399	0.91534	
12	Fear	0.38291	-0.17189	40.95583	47.48382	0.72719	
13	Fear	3.75234	-1.17039	42.02843	51.00419	0.91369	
14	Fear	2.46056	3.99993	40.38418	45.24587	0.55232	
15	Fear	7.20896	-1.13686	35.07062	44.31019	0.99166	
16	Fear	5.73994	-2.46244	40.67371	49.36174	0.99260	
17	Fear	-1.62481	-0.99770	42.85218	52.34444	0.98920	
18	Fear	7.73189	-0.74424	39.77528	46.33296	0.98282	
19	Fear	0.25097	0.02925	44.09817	48.88225	0.89628	
20	Fear	1.85115	-3.04463	42.85645	51.58608	0.95794	
21	Fear	3.99982	2.73485	38.19728	45.54285	0.99070	
22	Fear	10.63512	1.84967	36.12809	49.44750	0.98086	
23	Fear	3.39383	-2.00745	37.54149	49.35254	0.99751	
24	Fear	3.77559	-1.18630	38.36117	50.55214	0.96136	

	Pitch	Roll	Yaw	AUs1	...	x_63	y_63	x_64	\
0	-32.01462	-3.93380	1.97570	0.46146	...	40.95165	39.37061	36.86421	
1	1.22263	-0.12969	-11.22055	0.52335	...	54.85016	52.05235	47.44178	
2	0.19911	22.21268	5.36190	0.55052	...	54.75975	51.94575	48.86390	
3	2.96028	-1.94094	-9.07431	0.39763	...	53.40504	51.64485	48.41354	
4	-30.11337	-14.81665	18.02552	0.43739	...	47.99613	46.54035	44.47854	
5	-15.55777	-6.89316	-1.85987	0.71073	...	52.84832	51.56180	49.34932	
6	4.79232	-8.70563	15.45703	0.62455	...	49.33986	47.33384	43.71724	
7	-17.28552	-3.04267	17.67790	0.36385	...	44.34537	40.95079	37.43131	
8	-12.64820	2.41659	-3.07084	0.32901	...	48.77940	47.29832	45.25061	
9	1.57973	-4.35437	-3.75432	0.65703	...	58.59060	56.45291	52.48024	
10	1.39220	-4.21572	4.11580	0.44233	...	55.84596	53.39634	48.76723	
11	-7.54340	-7.65682	-10.04008	0.12529	...	47.00023	46.43975	45.24791	
12	3.25621	-6.76058	-29.16274	0.52023	...	46.12815	45.05192	41.67628	
13	-5.97736	3.76671	15.03077	0.49604	...	52.66714	50.55493	47.70594	
14	-15.54787	-9.63907	20.46483	0.45137	...	51.63849	49.79434	47.45398	
15	-1.68760	-2.69072	-5.62694	0.45942	...	48.50394	44.71327	38.10675	
16	7.83712	-4.61717	3.14730	0.77857	...	47.01358	46.05429	43.91478	
17	2.01177	-1.98949	-14.67377	0.27331	...	52.20196	51.15559	49.58959	
18	-8.72250	-18.23906	24.59081	0.63778	...	46.83942	46.01411	44.35730	
19	-0.23737	-1.80130	-2.90825	0.38359	...	57.07247	53.94987	48.71681	

20	-17.06499	4.78722	-7.25720	0.15584	...	53.74362	52.17507	50.14602
21	1.61652	-16.76449	-2.53922	0.50285	...	52.14043	50.04681	45.53837
22	7.54304	-8.56089	21.74920	0.62386	...	52.05343	51.52720	50.09843
23	5.86926	0.87837	-13.14791	0.66441	...	48.27398	47.16356	45.30027
24	-4.23063	-0.83021	0.71344	0.40691	...	56.10745	55.03292	53.56274

	y_64	x_65	y_65	x_66	y_66	x_67	y_67
0	37.92036	38.10317	37.60634	35.91333	38.11862	38.59736	38.40099
1	47.04236	47.43054	47.17856	47.76388	51.82141	52.22516	51.75454
2	50.77482	52.15790	52.91707	55.27203	52.92989	52.11535	50.77568
3	46.18839	46.20218	45.69854	46.35702	50.04139	50.68799	50.61642
4	44.59326	44.79778	44.29348	43.28744	44.42711	44.88246	44.69338
5	49.08266	49.19074	48.49259	47.45752	49.22489	49.80544	49.71641
6	42.37609	42.28596	41.57026	41.12783	46.36020	47.10725	47.05917
7	37.03533	37.60673	37.68488	39.28066	43.10565	42.96252	42.02432
8	45.26064	45.55306	45.36754	45.53994	45.77673	45.93426	45.57038
9	50.58848	50.57516	50.01865	50.38948	54.27223	55.00584	54.91567
10	46.95773	46.94323	46.25749	46.44905	51.58387	52.45869	52.39976
11	44.75575	44.62131	43.88939	42.36145	43.66055	44.27631	44.39646
12	38.15856	37.71466	36.86448	37.37862	41.73991	42.88505	43.38183
13	48.34499	48.88487	48.76034	48.84874	49.29286	49.34311	48.80627
14	48.36823	48.65904	48.02137	45.84360	48.37215	48.87844	48.61969
15	33.95164	33.94455	33.35725	36.21852	45.62520	46.71334	46.36695
16	41.33053	41.12755	40.74812	42.03906	42.49560	42.94998	43.07793
17	48.80692	48.97954	48.67979	48.96466	48.71257	48.96965	48.80337
18	44.27390	44.20754	43.62993	42.11081	43.42701	43.97949	44.09488
19	46.17109	46.16961	45.59698	46.77351	53.62147	54.52324	54.30551
20	51.06822	51.53179	51.09817	50.27788	51.15877	51.50142	51.04786
21	42.93316	42.48782	41.61291	41.50610	47.24938	48.40173	48.63549
22	48.91066	48.67026	48.12302	47.75928	48.15310	48.63210	48.84265
23	45.35453	45.66715	45.58930	45.82991	46.03570	46.06408	45.77416
24	53.73774	54.01996	53.66930	53.16931	53.31864	53.58670	53.29666

[25 rows x 165 columns]

Happy

```
In [ ]: happy_images_dir = 'Images/Happy'
happy_image_files = os.listdir(happy_images_dir)
happy_image_paths = [os.path.join(happy_images_dir, file) for file in happy_image_files if file.endswith(('.jpg'))]
```

```
In [ ]: features_dict = {
    'label': [],
    'FaceRectX': [],
    'FaceRectY': [],
    'FaceRectWidth': [],
    'FaceRectHeight': [],
```

```

'FaceScore': [],
'Pitch': [],
'Roll': [],
'Yaw': [],
}

```

```

In [ ]: for i, image_path in enumerate(happy_image_paths, 1):
        image = img_as_ubyte(io.imread(image_path))

        single_face_prediction = detector.detect_image(image_path)

        faceboxes = single_face_prediction.faceboxes

        faceboxes_x = faceboxes['FaceRectX']
        faceboxes_x_mean = round(np.mean(faceboxes_x), 5)

        faceboxes_y = faceboxes['FaceRectY']
        faceboxes_y_mean = round(np.mean(faceboxes_y), 5)

        faceboxes_w = faceboxes['FaceRectWidth']
        faceboxes_w_mean = round(np.mean(faceboxes_w), 5)

        faceboxes_h = faceboxes['FaceRectHeight']
        faceboxes_h_mean = round(np.mean(faceboxes_h), 5)

        faceboxes_s = faceboxes['FaceScore']
        faceboxes_s_mean = round(np.mean(faceboxes_s), 5)

        aus = single_face_prediction.aus
        aus_mean = np.mean(aus, axis=0)

        for j, au in enumerate(aus_mean):
            feature_name = f'AUs{j+1}'
            if feature_name not in features_dict:
                features_dict[feature_name] = []
            features_dict[feature_name].append(round(au,5))

        landmarks = single_face_prediction.landmarks
        landmarks_mean = np.mean(landmarks, axis=0)
        num_landmarks = len(landmarks_mean) // 2
        for j in range(num_landmarks):
            feature_name_x = f'x_{j}'
            feature_name_y = f'y_{j}'
            if feature_name_x not in features_dict:
                features_dict[feature_name_x] = []
            if feature_name_y not in features_dict:
                features_dict[feature_name_y] = []

```



```

        features_dict[feature_name_x].append(round(landmarks_mean[2 * j],5))
        features_dict[feature_name_y].append(round(landmarks_mean[2 * j + 1],5))

poses = single_face_prediction.poses

poses_pitch = poses['Pitch']
poses_pitch_mean = round(np.mean(poses_pitch), 5)
poses_roll = poses['Roll']
poses_roll_mean = round(np.mean(poses_roll), 5)

poses_yaw = poses['Yaw']
poses_yaw_mean = round(np.mean(poses_yaw), 5)

features_dict['FaceRectX'].append(faceboxes_x_mean)
features_dict['FaceRectY'].append(faceboxes_y_mean)
features_dict['FaceRectWidth'].append(faceboxes_w_mean)
features_dict['FaceRectHeight'].append(faceboxes_h_mean)
features_dict['FaceScore'].append(faceboxes_s_mean)
features_dict['Pitch'].append(poses_pitch_mean)
features_dict['Roll'].append(poses_roll_mean)
features_dict['Yaw'].append(poses_yaw_mean)

features_dict['label'].append('Happy')

new_features = pd.DataFrame(features_dict)
features = pd.read_csv('features.csv')

updated_features = pd.concat([features, new_features], ignore_index=True)
updated_features.to_csv('features.csv', index=False)

print(new_features)

```

100%	<div></div>	1/1	[00:01<00:00,	1.27s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.25s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.28s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.27s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.25s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.25s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.27s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.27s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.27s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.28s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.25s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.25s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.27s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.27s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.24s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.24s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.30s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.27s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.24s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.29s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]

	label	FaceRectX	FaceRectY	FaceRectWidth	FaceRectHeight	FaceScore	\
0	Happy	2.82097	3.78955	39.74320	43.62240	0.98112	
1	Happy	3.23014	-0.29557	42.10837	49.77184	0.94763	
2	Happy	4.69849	-0.14742	39.84939	45.73217	0.99287	
3	Happy	7.57921	-1.57234	39.32342	48.18084	0.97288	
4	Happy	3.16656	1.79263	39.50128	46.37878	0.99305	
5	Happy	4.15738	-2.35682	42.59965	52.89632	0.96608	
6	Happy	6.02322	1.58543	37.68077	42.42455	0.89286	
7	Happy	3.94852	1.88991	38.59476	43.26874	0.99515	
8	Happy	3.66724	0.88557	38.95014	47.64102	0.99518	
9	Happy	3.99610	-1.33586	38.69261	50.45693	0.99387	
10	Happy	9.64921	8.21111	34.32427	36.75671	0.96240	
11	Happy	5.79213	0.45323	36.92946	43.88050	0.99157	
12	Happy	5.45202	1.57513	36.47587	45.17902	0.92232	
13	Happy	5.01016	0.07781	35.64139	45.35544	0.99495	
14	Happy	1.81655	-1.63820	42.13181	51.52717	0.97138	
15	Happy	4.96268	-0.10993	38.51628	48.40302	0.95817	
16	Happy	2.40669	0.49175	44.05666	50.68521	0.92489	
17	Happy	4.43747	1.28141	38.03958	46.59628	0.98965	
18	Happy	3.08279	-2.42656	37.94578	49.93173	0.99208	
19	Happy	3.95994	1.74537	36.57555	42.92633	0.99826	
20	Happy	4.46482	1.11405	41.45819	48.18375	0.85899	
21	Happy	1.69778	1.78035	37.50077	45.67955	0.76619	
22	Happy	3.97010	0.02399	37.63667	47.11597	0.99378	
23	Happy	7.62186	1.30965	37.43205	48.57163	0.91889	
24	Happy	3.05185	1.14349	40.23302	47.51925	0.97397	

	Pitch	Roll	Yaw	AUs1	...	x_63	y_63	x_64	\
0	15.08875	-1.87233	-1.80143	0.49686	...	46.30667	44.77561	41.63152	
1	8.68991	-4.74037	-2.07803	0.38939	...	53.82524	51.06531	46.15846	
2	-4.07303	-0.70963	3.18776	0.22413	...	48.95612	45.76986	40.67724	
3	-17.05611	13.47042	-10.30932	0.41449	...	47.93712	43.69226	37.59790	
4	-0.02138	-6.69636	-3.63754	0.70264	...	51.72627	50.21758	47.30727	
5	1.89144	-4.62587	5.50992	0.41942	...	56.97996	55.93875	53.83454	
6	-2.15578	9.23142	-15.47425	0.58114	...	42.79360	39.41914	34.10875	
7	9.21588	-11.41871	0.95147	0.37553	...	44.23717	43.37234	41.45018	
8	-5.94514	3.25595	-7.05029	0.46087	...	50.37490	48.83183	46.11720	
9	4.08458	-7.87066	-2.84602	0.30700	...	53.08860	51.05185	46.86125	
10	-0.89431	10.26508	-2.25499	0.35872	...	38.33857	36.12856	33.09094	
11	-22.87022	5.72191	-13.16278	0.50100	...	42.57123	41.15204	39.04689	
12	7.76491	1.23898	7.95112	0.37495	...	43.83552	41.92769	38.93730	
13	12.37430	7.00153	6.23926	0.79621	...	44.61380	42.02794	38.17739	
14	0.85071	11.11807	-1.18093	0.25831	...	54.72459	50.94437	45.55588	
15	0.37190	4.73909	-1.96053	0.36392	...	52.12419	48.95626	43.93544	
16	4.87804	-6.69648	-3.00431	0.45675	...	55.64644	53.41529	49.51105	
17	-10.00780	-0.70327	-4.95415	0.41308	...	52.07696	48.11954	41.83155	
18	-10.52668	-5.60003	-10.54262	0.21663	...	52.42464	50.35475	46.01841	
19	-20.99732	5.25816	-4.13804	0.35842	...	42.85005	40.10775	35.96405	

20	1.34047	10.56039	0.87612	0.46834	...	50.27929	46.66583	41.62180
21	9.43273	-20.68390	2.93207	0.36137	...	46.77518	46.01100	43.30177
22	-15.06336	5.68271	-6.96117	0.31325	...	46.54940	44.15996	40.31841
23	3.68432	-2.92377	2.58004	0.41867	...	52.36897	49.65683	44.68474
24	-8.65364	24.05696	-9.86292	0.37980	...	47.11368	42.70005	37.09408

	y_64	x_65	y_65	x_66	y_66	x_67	y_67
0	38.49018	38.43617	38.09942	39.94118	43.55960	44.04960	44.00475
1	44.21084	44.29877	43.79446	44.54732	50.71019	51.36227	51.14982
2	41.79972	42.16398	41.65920	40.55078	45.72862	46.31057	45.85010
3	38.66105	39.45535	39.33904	40.21974	46.03108	46.27361	45.28226
4	45.79863	45.77318	45.16614	45.07966	47.93336	48.65853	48.66433
5	53.72312	53.79496	53.10837	51.76094	53.09620	53.64813	53.60696
6	35.08200	35.76941	35.73456	36.44846	40.89662	41.05634	40.35957
7	39.32391	39.12495	38.65139	38.97894	41.06940	41.60314	41.78529
8	46.20744	46.35679	45.99175	45.62565	47.54698	47.95356	47.79850
9	44.73199	44.61454	43.83918	43.81229	49.38868	50.31952	50.46571
10	34.56457	35.16667	35.20734	35.44461	37.44477	37.37629	36.76062
11	39.51891	39.73815	39.43600	39.00496	39.63232	39.95024	39.73686
12	40.10273	40.45738	40.13712	39.04692	41.84377	42.08030	41.69536
13	37.86938	38.46090	38.50748	40.28210	43.12795	43.00025	42.31291
14	47.41405	48.44222	48.70130	49.87336	52.71777	52.45635	51.35612
15	43.86474	44.34221	44.14518	44.90519	49.29279	49.62406	49.07657
16	48.60837	48.52417	47.49061	45.88626	51.25419	52.34941	52.42459
17	42.09288	42.43404	41.92697	41.63927	49.05340	49.77919	49.25198
18	45.68994	45.52095	44.67091	43.14203	47.76962	48.82484	49.04423
19	37.69289	38.09201	37.74611	36.65119	39.95871	40.36253	39.99110
20	43.40842	44.36005	44.76417	46.42092	48.52895	48.19809	47.11606
21	39.95320	39.36743	38.22024	36.90350	42.18980	43.42737	44.08463
22	42.08740	42.48712	42.16755	41.21792	44.41563	44.77450	44.40310
23	43.87067	43.93998	43.35482	43.32564	48.48156	49.22055	49.04821
24	40.05084	41.43855	42.05831	43.86137	46.72093	46.09278	44.56653

[25 rows x 165 columns]

Neutral

```
In [ ]: neutral_images_dir = 'Images/Neutral'
neutral_image_files = os.listdir(neutral_images_dir)
neutral_image_paths = [os.path.join(neutral_images_dir, file) for file in neutral_image_files if file.endswith(('.jpg'))]
```

```
In [ ]: features_dict = {
    'label': [],
    'FaceRectX': [],
    'FaceRectY': [],
    'FaceRectWidth': [],
    'FaceRectHeight': [],
```

```

'FaceScore': [],
'Pitch': [],
'Roll': [],
'Yaw': [],
}

```

```

In [ ]: for i, image_path in enumerate(neutral_image_paths, 1):
        image = img_as_ubyte(io.imread(image_path))

        single_face_prediction = detector.detect_image(image_path)

        faceboxes = single_face_prediction.faceboxes

        faceboxes_x = faceboxes['FaceRectX']
        faceboxes_x_mean = round(np.mean(faceboxes_x), 5)

        faceboxes_y = faceboxes['FaceRectY']
        faceboxes_y_mean = round(np.mean(faceboxes_y), 5)

        faceboxes_w = faceboxes['FaceRectWidth']
        faceboxes_w_mean = round(np.mean(faceboxes_w), 5)

        faceboxes_h = faceboxes['FaceRectHeight']
        faceboxes_h_mean = round(np.mean(faceboxes_h), 5)

        faceboxes_s = faceboxes['FaceScore']
        faceboxes_s_mean = round(np.mean(faceboxes_s), 5)

        aus = single_face_prediction.aus
        aus_mean = np.mean(aus, axis=0)

        for j, au in enumerate(aus_mean):
            feature_name = f'AUs{j+1}'
            if feature_name not in features_dict:
                features_dict[feature_name] = []
            features_dict[feature_name].append(round(au,5))

        landmarks = single_face_prediction.landmarks
        landmarks_mean = np.mean(landmarks, axis=0)
        num_landmarks = len(landmarks_mean) // 2
        for j in range(num_landmarks):
            feature_name_x = f'x_{j}'
            feature_name_y = f'y_{j}'
            if feature_name_x not in features_dict:
                features_dict[feature_name_x] = []
            if feature_name_y not in features_dict:
                features_dict[feature_name_y] = []

```

```

        features_dict[feature_name_x].append(round(landmarks_mean[2 * j],5))
        features_dict[feature_name_y].append(round(landmarks_mean[2 * j + 1],5))

poses = single_face_prediction.poses

poses_pitch = poses['Pitch']
poses_pitch_mean = round(np.mean(poses_pitch), 5)
poses_roll = poses['Roll']
poses_roll_mean = round(np.mean(poses_roll), 5)

poses_yaw = poses['Yaw']
poses_yaw_mean = round(np.mean(poses_yaw), 5)

features_dict['FaceRectX'].append(faceboxes_x_mean)
features_dict['FaceRectY'].append(faceboxes_y_mean)
features_dict['FaceRectWidth'].append(faceboxes_w_mean)
features_dict['FaceRectHeight'].append(faceboxes_h_mean)
features_dict['FaceScore'].append(faceboxes_s_mean)
features_dict['Pitch'].append(poses_pitch_mean)
features_dict['Roll'].append(poses_roll_mean)
features_dict['Yaw'].append(poses_yaw_mean)

features_dict['label'].append('Neutral')

new_features = pd.DataFrame(features_dict)
features = pd.read_csv('features.csv')

updated_features = pd.concat([features, new_features], ignore_index=True)
updated_features.to_csv('features.csv', index=False)

print(new_features)

```

100%	<div></div>	1/1	[00:01<00:00, 1.28s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.27s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.27s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.26s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.27s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.27s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.26s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.25s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.25s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.25s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.29s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.26s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.24s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.23s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.26s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.25s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.25s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.26s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.26s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.26s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.26s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.28s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.28s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.24s/it]
100%	<div></div>	1/1	[00:01<00:00, 1.25s/it]

	label	FaceRectX	FaceRectY	FaceRectWidth	FaceRectHeight	FaceScore	\
0	Neutral	5.12296	-0.30444	39.05001	46.40080	0.98309	
1	Neutral	2.38735	0.62791	37.21551	47.37532	0.98691	
2	Neutral	7.01911	5.80385	29.38776	41.31418	0.99419	
3	Neutral	8.28344	6.21075	34.56144	40.44554	0.99077	
4	Neutral	4.74867	3.54630	37.56432	45.03683	0.99676	
5	Neutral	3.81859	0.75255	43.08570	47.26272	0.97010	
6	Neutral	11.23246	6.42700	34.82922	40.94336	0.97664	
7	Neutral	7.43279	0.02592	37.70141	49.46546	0.98796	
8	Neutral	3.17755	0.84387	41.33131	51.11934	0.87945	
9	Neutral	3.87168	0.84788	39.24878	48.17735	0.99635	
10	Neutral	0.45171	-0.23094	44.43656	46.83657	0.96983	
11	Neutral	5.24938	1.23601	38.96188	47.61682	0.98883	
12	Neutral	5.95535	4.15890	38.61210	45.52596	0.99670	
13	Neutral	4.01815	-0.95156	41.05607	50.13072	0.99345	
14	Neutral	0.92972	-2.48820	39.94264	55.67538	0.71077	
15	Neutral	3.86304	-1.28702	40.13023	50.42874	0.98394	
16	Neutral	3.68297	-0.18276	39.09629	48.29828	0.99464	
17	Neutral	3.53848	0.54512	37.15710	46.49200	0.99365	
18	Neutral	2.38795	-0.62694	41.40369	48.82926	0.98452	
19	Neutral	2.15994	-0.93478	41.12747	50.59358	0.98870	
20	Neutral	7.09346	8.40723	30.81046	38.59147	0.99273	
21	Neutral	5.55060	3.02723	37.81081	43.46281	0.98714	
22	Neutral	4.13600	0.85621	39.20039	46.88092	0.98707	
23	Neutral	4.74239	-0.03515	37.70703	49.24160	0.98065	
24	Neutral	7.14332	3.20548	38.69984	45.60378	0.95855	

	Pitch	Roll	Yaw	AUs1	...	x_63	y_63	x_64	\
0	6.42400	-4.91268	2.36535	0.36796	...	48.27386	47.69605	46.08908	
1	12.26465	-1.83989	-5.04030	0.66306	...	45.05735	44.81588	43.91887	
2	8.20208	-1.01178	0.46535	0.81137	...	43.37622	42.47787	40.91398	
3	-16.86372	15.20968	-10.40641	0.16129	...	42.70514	41.00150	38.82017	
4	1.78971	-5.44917	-1.29288	0.64638	...	44.93648	44.11098	42.54761	
5	5.36933	-3.23071	1.24262	0.42043	...	47.16610	46.50276	45.42894	
6	-8.83935	13.29156	-17.57981	0.28505	...	41.93151	40.70309	39.22682	
7	1.44042	3.92352	11.47178	0.44005	...	54.38008	52.67367	50.79238	
8	6.72385	-1.96199	0.16932	0.40368	...	54.56207	53.70149	51.94819	
9	1.05866	-1.59459	0.74687	0.21882	...	48.24162	47.67757	46.51974	
10	1.87652	1.18745	-5.66928	0.35544	...	45.43796	43.72301	40.87709	
11	-11.95926	13.80248	-5.29345	0.34049	...	51.06564	48.92968	46.05479	
12	8.45718	4.14268	-0.52913	0.41953	...	45.42094	44.57067	43.20325	
13	-9.37175	0.64500	7.17097	0.49217	...	55.04662	53.75579	51.85122	
14	-4.51063	6.06995	-13.31240	0.27433	...	56.54810	54.81457	52.11776	
15	2.46094	1.84015	-6.04077	0.21779	...	53.14006	51.64221	49.05653	
16	10.62012	-2.40166	-5.43012	0.41395	...	48.20251	47.65140	46.49792	
17	-13.53817	-7.96352	-4.44027	0.26596	...	46.83051	46.01952	44.46784	
18	-5.07218	-13.90758	3.31806	0.42394	...	50.94018	50.63263	49.44104	
19	-2.65634	10.58117	-3.50799	0.29759	...	51.39670	49.40016	46.99575	

20	1.47458	2.42140	0.34913	0.32000	...	43.70448	42.63113	40.98257
21	-18.51394	-6.02253	-3.55383	0.33114	...	44.13678	42.50808	39.29478
22	16.34243	-3.66871	-0.81944	0.67659	...	48.13968	47.77460	46.11453
23	3.06935	-0.95230	-3.68523	0.53325	...	50.66766	49.31951	47.00747
24	5.60897	1.09422	0.19764	0.43280	...	46.73997	45.67160	44.01676

	y_64	x_65	y_65	x_66	y_66	x_67	y_67
0	45.77065	45.80264	45.30564	44.67556	45.34523	45.75200	45.70833
1	42.41459	42.46275	42.18698	43.03843	42.51444	42.69321	42.61747
2	40.57873	40.72669	40.42232	40.37163	41.12937	41.36862	41.21712
3	40.26228	40.75826	40.80255	40.90574	40.99811	40.92735	40.43369
4	42.14140	42.17126	41.70488	41.31453	41.67616	42.08624	42.06306
5	44.27705	44.34196	43.91291	44.14062	44.17844	44.56666	44.50711
6	40.55245	41.00917	40.98136	40.96790	40.97642	40.94281	40.44883
7	50.54234	51.04289	51.04828	51.91372	51.42384	51.39193	50.86915
8	51.30830	51.58006	51.32241	51.83131	51.18445	51.40278	51.17362
9	45.88113	46.07153	45.84180	46.06678	45.44794	45.62138	45.55425
10	40.67244	40.99529	40.79559	41.18289	42.80703	43.00957	42.70049
11	46.69785	47.36472	47.42910	48.16720	48.20467	48.14230	47.50277
12	42.41946	42.67960	42.67676	43.72115	43.01721	42.97691	42.75080
13	51.34280	51.61662	51.37612	51.71437	51.77593	51.98688	51.71628
14	52.06722	52.59490	52.57654	53.37031	53.02233	52.95984	52.39896
15	49.75550	50.07971	49.86246	49.73758	50.21231	50.43631	49.98529
16	44.73857	44.78997	44.60316	45.75766	45.08423	45.25344	45.13425
17	44.13312	43.96402	43.14702	41.51867	43.17824	43.93044	44.07826
18	48.26133	47.95976	47.10761	45.52162	46.83652	47.61254	47.93912
19	47.61775	48.19376	48.09728	48.51566	48.50512	48.55731	48.05194
20	41.47471	41.72830	41.47275	41.06625	41.54993	41.73468	41.51052
21	39.76973	39.73923	39.07216	37.38964	40.78160	41.53047	41.62299
22	43.09711	42.83652	42.40988	43.76270	43.97990	44.46645	44.63642
23	46.58020	46.79930	46.56330	46.84871	46.67543	46.88945	46.68680
24	43.07869	43.27793	43.09941	43.99034	43.96173	44.15800	43.93379

[25 rows x 165 columns]

Sad

```
In [ ]: sad_images_dir = 'Images/Sad'
sad_image_files = os.listdir(sad_images_dir)
sad_image_paths = [os.path.join(sad_images_dir, file) for file in sad_image_files if file.endswith((''.jpg'))]
```

```
In [ ]: features_dict = {
    'label': [],
    'FaceRectX': [],
    'FaceRectY': [],
    'FaceRectWidth': [],
    'FaceRectHeight': [],
```

```

'FaceScore': [],
'Pitch': [],
'Roll': [],
'Yaw': [],
}

```

```

In [ ]: for i, image_path in enumerate(sad_image_paths, 1):
        image = img_as_ubyte(io.imread(image_path))

        single_face_prediction = detector.detect_image(image_path)

        faceboxes = single_face_prediction.faceboxes

        faceboxes_x = faceboxes['FaceRectX']
        faceboxes_x_mean = round(np.mean(faceboxes_x), 5)

        faceboxes_y = faceboxes['FaceRectY']
        faceboxes_y_mean = round(np.mean(faceboxes_y), 5)

        faceboxes_w = faceboxes['FaceRectWidth']
        faceboxes_w_mean = round(np.mean(faceboxes_w), 5)

        faceboxes_h = faceboxes['FaceRectHeight']
        faceboxes_h_mean = round(np.mean(faceboxes_h), 5)

        faceboxes_s = faceboxes['FaceScore']
        faceboxes_s_mean = round(np.mean(faceboxes_s), 5)

        aus = single_face_prediction.aus
        aus_mean = np.mean(aus, axis=0)

        for j, au in enumerate(aus_mean):
            feature_name = f'AUs{j+1}'
            if feature_name not in features_dict:
                features_dict[feature_name] = []
            features_dict[feature_name].append(round(au,5))

        landmarks = single_face_prediction.landmarks
        landmarks_mean = np.mean(landmarks, axis=0)
        num_landmarks = len(landmarks_mean) // 2
        for j in range(num_landmarks):
            feature_name_x = f'x_{j}'
            feature_name_y = f'y_{j}'
            if feature_name_x not in features_dict:
                features_dict[feature_name_x] = []
            if feature_name_y not in features_dict:
                features_dict[feature_name_y] = []

```

```

        features_dict[feature_name_x].append(round(landmarks_mean[2 * j],5))
        features_dict[feature_name_y].append(round(landmarks_mean[2 * j + 1],5))

    poses = single_face_prediction.poses

    poses_pitch = poses['Pitch']
    poses_pitch_mean = round(np.mean(poses_pitch), 5)
    poses_roll = poses['Roll']
    poses_roll_mean = round(np.mean(poses_roll), 5)

    poses_yaw = poses['Yaw']
    poses_yaw_mean = round(np.mean(poses_yaw), 5)

    features_dict['FaceRectX'].append(faceboxes_x_mean)
    features_dict['FaceRectY'].append(faceboxes_y_mean)
    features_dict['FaceRectWidth'].append(faceboxes_w_mean)
    features_dict['FaceRectHeight'].append(faceboxes_h_mean)
    features_dict['FaceScore'].append(faceboxes_s_mean)
    features_dict['Pitch'].append(poses_pitch_mean)
    features_dict['Roll'].append(poses_roll_mean)
    features_dict['Yaw'].append(poses_yaw_mean)

    features_dict['label'].append('Sad')

new_features = pd.DataFrame(features_dict)
features = pd.read_csv('features.csv')

updated_features = pd.concat([features, new_features], ignore_index=True)
updated_features.to_csv('features.csv', index=False)

print(new_features)

```

100%	<div></div>	1/1	[00:01<00:00,	1.29s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.31s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.28s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.27s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.27s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.27s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.24s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.23s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.28s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.26s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.25s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.37s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.30s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.37s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.34s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.36s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.46s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.55s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.37s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.51s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.33s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.39s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.30s/it]

	label	FaceRectX	FaceRectY	FaceRectWidth	FaceRectHeight	FaceScore	\
0	Sad	7.53780	3.87589	33.24858	39.17407	0.98998	
1	Sad	10.96405	6.34983	29.47453	39.84032	0.99806	
2	Sad	0.39538	-2.85294	41.29798	51.02794	0.96916	
3	Sad	7.65502	0.85312	37.87595	45.31879	0.98893	
4	Sad	1.17937	0.36617	39.33860	48.42120	0.98940	
5	Sad	13.46831	7.95348	30.47258	36.35251	0.99254	
6	Sad	1.41937	-0.10874	39.73501	48.40682	0.92352	
7	Sad	1.73833	-1.21788	39.59742	49.00056	0.98927	
8	Sad	3.89306	-0.58001	38.62445	47.62249	0.97202	
9	Sad	1.72152	-2.37958	43.16897	55.71462	0.92960	
10	Sad	0.24087	-3.87537	47.24090	56.53215	0.84470	
11	Sad	4.81807	4.67305	36.78684	45.93368	0.98747	
12	Sad	7.33867	-0.21640	30.38550	42.79407	0.99484	
13	Sad	7.87501	5.13824	27.79170	36.24727	0.99166	
14	Sad	5.72289	-0.01628	41.67873	51.41524	0.98030	
15	Sad	5.71447	3.19029	36.98966	44.17866	0.89008	
16	Sad	6.80969	3.36700	36.66109	44.28494	0.99083	
17	Sad	4.38371	-2.15060	39.81011	50.01125	0.99671	
18	Sad	4.59525	6.45232	32.42309	40.43469	0.95170	
19	Sad	3.00185	-1.61533	42.77150	50.65828	0.97092	
20	Sad	5.11320	-2.16159	42.49845	50.96217	0.99168	
21	Sad	5.81421	-2.14076	37.93638	51.30288	0.97664	
22	Sad	4.59780	-2.93436	41.67180	50.39603	0.98160	
23	Sad	10.31256	4.96529	28.97951	38.53129	0.99741	
24	Sad	3.33401	-2.30286	37.72202	53.93241	0.96593	

	Pitch	Roll	Yaw	AUs1	...	x_63	y_63	x_64	\
0	10.24327	5.43862	11.18917	0.54406	...	40.56133	38.64050	36.13033	
1	-5.23047	-2.21136	-3.51795	0.19655	...	42.60119	41.98713	40.61989	
2	10.57769	-7.79054	-6.58551	0.29396	...	47.15438	47.03681	46.61432	
3	-11.49605	-4.95856	12.29341	0.51270	...	46.05916	44.59823	42.69650	
4	-4.10269	-7.82821	-12.41051	0.32753	...	46.90407	46.77980	46.07404	
5	0.97710	4.26345	-13.84788	0.31937	...	39.74305	39.09499	37.95573	
6	-15.94155	9.27026	-7.59606	0.28502	...	51.83492	49.98031	47.39202	
7	-22.19714	2.12403	-1.47638	0.38323	...	52.27042	50.98517	48.93990	
8	-18.84207	7.25292	0.04148	0.27266	...	47.70125	46.42895	44.85609	
9	0.02674	-2.68523	-8.51392	0.46586	...	59.34657	58.61427	57.16934	
10	-5.09349	11.58688	-15.23274	0.60279	...	63.17103	60.90053	58.24123	
11	-20.26378	-12.22352	4.01318	0.34891	...	51.08208	49.89527	48.46374	
12	-4.69857	1.02687	-7.11857	0.35764	...	37.24043	36.72396	36.07200	
13	1.11449	5.81946	9.17469	0.24497	...	37.08649	36.29909	35.32185	
14	-19.82804	-4.04417	15.72971	0.48735	...	58.79071	56.87554	54.48090	
15	-19.54540	-20.09093	2.47700	0.28651	...	48.03215	46.95301	45.25067	
16	1.45682	8.24574	-0.16780	0.40468	...	44.55950	43.02935	41.14306	
17	-8.50593	-1.92358	-1.16709	0.41866	...	51.39872	49.94447	47.89358	
18	5.77822	3.04223	-12.51159	0.57456	...	43.95384	42.68459	40.15538	
19	4.15781	-6.98430	4.33729	0.46311	...	52.99506	51.66657	48.58490	

20	-1.55907	6.29630	-7.75514	0.36140	...	52.43357	50.95522	49.19928
21	-6.86349	3.06454	8.90828	0.48751	...	52.08789	50.84790	48.95712
22	-7.66782	-2.77751	6.19045	0.42397	...	49.89786	48.49133	46.27412
23	-18.05486	3.04387	-0.63903	0.42945	...	40.45858	39.54468	38.17121
24	-8.92933	8.34399	-0.84262	0.35557	...	53.62977	51.80286	49.66234

	y_64	x_65	y_65	x_66	y_66	x_67	y_67
0	35.44202	35.84029	35.90862	37.41380	38.90569	38.87905	38.35896
1	40.08147	40.12359	39.80898	39.71585	40.08003	40.39828	40.36083
2	44.51411	44.47495	44.11817	44.57600	44.40130	44.73534	44.84665
3	43.23921	43.55671	43.20244	42.48184	43.26100	43.58409	43.30719
4	44.66737	44.45091	43.63409	42.71589	43.76459	44.51694	44.75502
5	37.42999	37.52834	37.40015	37.69151	37.80735	37.93616	37.86248
6	48.35810	48.96734	48.79339	48.47609	48.75543	48.82396	48.24032
7	49.51253	49.77243	49.29131	48.42193	49.64442	50.06171	49.75142
8	45.37781	45.87247	45.79267	46.09485	45.96227	45.97638	45.48660
9	56.33266	56.45761	55.97205	55.58129	55.58764	55.96208	55.90729
10	58.39479	59.22364	59.01997	59.52633	59.37680	59.46509	58.65172
11	48.48750	48.50955	47.92293	46.53748	48.04935	48.56681	48.56074
12	35.36210	35.56767	35.37889	35.78249	35.67919	35.85637	35.68707
13	35.56044	35.82659	35.69562	35.80700	35.98088	36.06186	35.80882
14	54.74792	55.24104	54.94475	54.30681	55.16272	55.36161	54.90011
15	45.29853	45.09032	44.10462	41.45849	43.96359	44.92646	45.17388
16	42.12574	42.76878	42.82879	43.39547	42.97149	42.80671	42.18878
17	47.91010	48.19336	47.77378	47.16899	47.68922	48.01798	47.85030
18	38.59595	38.84877	38.85157	40.54797	42.01401	42.07963	41.84088
19	45.58759	45.39827	44.99114	46.55684	49.29721	49.84162	49.90697
20	49.14579	49.70517	49.63540	50.59852	49.89389	49.89066	49.37630
21	49.12470	49.56733	49.37220	49.74243	49.33366	49.46723	49.09930
22	46.30393	46.57959	46.19358	45.75508	46.41349	46.72730	46.47720
23	38.63954	38.85524	38.56648	38.03217	38.62845	38.87805	38.69240
24	50.40893	51.18011	51.26369	52.41575	51.54439	51.33292	50.60988

[25 rows x 165 columns]

Surprised

```
In [ ]: surprised_images_dir = 'Images/Surprised'
surprised_image_files = os.listdir(surprised_images_dir)
surprised_image_paths = [os.path.join(surprised_images_dir, file) for file in surprised_image_files if file.endswith((''.jpg'))]
```

```
In [ ]: features_dict = {
    'label': [],
    'FaceRectX': [],
    'FaceRectY': [],
    'FaceRectWidth': [],
    'FaceRectHeight': [],
```

```

'FaceScore': [],
'Pitch': [],
'Roll': [],
'Yaw': [],
}

```

```

In [ ]: for i, image_path in enumerate(surprised_image_paths, 1):
        image = img_as_ubyte(io.imread(image_path))

        single_face_prediction = detector.detect_image(image_path)

        faceboxes = single_face_prediction.faceboxes

        faceboxes_x = faceboxes['FaceRectX']
        faceboxes_x_mean = round(np.mean(faceboxes_x), 5)

        faceboxes_y = faceboxes['FaceRectY']
        faceboxes_y_mean = round(np.mean(faceboxes_y), 5)

        faceboxes_w = faceboxes['FaceRectWidth']
        faceboxes_w_mean = round(np.mean(faceboxes_w), 5)

        faceboxes_h = faceboxes['FaceRectHeight']
        faceboxes_h_mean = round(np.mean(faceboxes_h), 5)

        faceboxes_s = faceboxes['FaceScore']
        faceboxes_s_mean = round(np.mean(faceboxes_s), 5)

        aus = single_face_prediction.aus
        aus_mean = np.mean(aus, axis=0)

        for j, au in enumerate(aus_mean):
            feature_name = f'AUs{j+1}'
            if feature_name not in features_dict:
                features_dict[feature_name] = []
            features_dict[feature_name].append(round(au,5))

        landmarks = single_face_prediction.landmarks
        landmarks_mean = np.mean(landmarks, axis=0)
        num_landmarks = len(landmarks_mean) // 2
        for j in range(num_landmarks):
            feature_name_x = f'x_{j}'
            feature_name_y = f'y_{j}'
            if feature_name_x not in features_dict:
                features_dict[feature_name_x] = []
            if feature_name_y not in features_dict:
                features_dict[feature_name_y] = []

```

```

        features_dict[feature_name_x].append(round(landmarks_mean[2 * j],5))
        features_dict[feature_name_y].append(round(landmarks_mean[2 * j + 1],5))

poses = single_face_prediction.poses

poses_pitch = poses['Pitch']
poses_pitch_mean = round(np.mean(poses_pitch), 5)
poses_roll = poses['Roll']
poses_roll_mean = round(np.mean(poses_roll), 5)

poses_yaw = poses['Yaw']
poses_yaw_mean = round(np.mean(poses_yaw), 5)

features_dict['FaceRectX'].append(faceboxes_x_mean)
features_dict['FaceRectY'].append(faceboxes_y_mean)
features_dict['FaceRectWidth'].append(faceboxes_w_mean)
features_dict['FaceRectHeight'].append(faceboxes_h_mean)
features_dict['FaceScore'].append(faceboxes_s_mean)
features_dict['Pitch'].append(poses_pitch_mean)
features_dict['Roll'].append(poses_roll_mean)
features_dict['Yaw'].append(poses_yaw_mean)

features_dict['label'].append('Surprised')

new_features = pd.DataFrame(features_dict)
features = pd.read_csv('features.csv')

updated_features = pd.concat([features, new_features], ignore_index=True)
updated_features.to_csv('features.csv', index=False)

print(new_features)

```


100%	<div></div>	1/1	[00:01<00:00,	1.31s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.27s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.43s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.35s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.36s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.43s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.28s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.35s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.33s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.35s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.38s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.31s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.40s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.33s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.33s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.32s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.34s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.32s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.32s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.32s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.33s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.33s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.34s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.30s/it]
100%	<div></div>	1/1	[00:01<00:00,	1.34s/it]

	label	FaceRectX	FaceRectY	FaceRectWidth	FaceRectHeight	FaceScore	\
0	Surprised	2.85344	-0.63587	37.69870	48.91405	0.99063	
1	Surprised	7.08851	-1.89367	38.95039	52.40803	0.96038	
2	Surprised	1.96710	1.22613	36.15473	49.31109	0.97304	
3	Surprised	5.38411	1.16287	37.78231	47.32035	0.99785	
4	Surprised	6.76245	-0.09300	34.85153	44.37422	0.98046	
5	Surprised	5.65651	-0.57452	38.35103	47.75775	0.99293	
6	Surprised	0.20780	-2.38630	37.84409	52.60358	0.94736	
7	Surprised	7.78728	2.90390	31.55254	41.14660	0.99440	
8	Surprised	3.19911	-2.59622	41.69751	50.29013	0.98997	
9	Surprised	10.80361	3.55572	26.96743	37.30335	0.91059	
10	Surprised	5.89765	0.52622	39.30468	49.03660	0.94070	
11	Surprised	2.84932	-2.63776	40.60037	47.58816	0.87628	
12	Surprised	6.88963	0.70618	36.14800	45.21003	0.94435	
13	Surprised	6.07010	4.08327	33.12854	40.70073	0.88755	
14	Surprised	7.29605	3.90375	38.87454	45.73373	0.86557	
15	Surprised	3.63083	-1.34386	39.87326	51.45263	0.97675	
16	Surprised	2.43313	2.49545	34.45116	47.38824	0.96149	
17	Surprised	1.87659	-2.87243	41.63138	53.33953	0.97055	
18	Surprised	6.07671	2.55890	35.88535	44.66297	0.93439	
19	Surprised	6.61739	-0.72695	42.00081	51.13622	0.98462	
20	Surprised	1.41105	-3.86620	39.01684	56.90246	0.78381	
21	Surprised	2.42433	1.27497	40.74375	45.96572	0.99458	
22	Surprised	5.01765	-1.34560	38.09922	48.96662	0.76567	
23	Surprised	4.75763	0.19341	41.06581	50.28328	0.97140	
24	Surprised	2.27412	-2.18565	36.29708	50.41788	0.99378	

	Pitch	Roll	Yaw	AUs1	...	x_63	y_63	x_64	\
0	-8.42949	-0.78876	-0.87662	0.55920	...	54.69026	52.01212	47.49632	
1	-4.18444	-5.87023	5.45812	0.63479	...	61.54065	58.64709	53.75302	
2	4.82960	8.43769	-17.23248	0.61477	...	55.88972	52.05290	45.99653	
3	-1.55782	-4.09472	5.08892	0.59031	...	48.13669	47.12003	45.19974	
4	-8.14116	-3.91831	-4.85173	0.24114	...	45.17700	43.24011	39.36210	
5	-7.40716	3.99059	-1.51505	0.48638	...	51.11903	48.49466	44.14674	
6	-14.77022	1.20327	-16.12848	0.37499	...	54.48764	53.22102	50.40618	
7	-4.91590	1.10797	-0.78815	0.89323	...	47.58698	43.89794	38.16065	
8	-1.60531	-9.92052	-1.03365	0.64890	...	53.44074	52.03423	48.60287	
9	17.69979	-1.24832	-0.04820	0.82422	...	36.43653	35.09786	32.44774	
10	3.70738	1.35310	6.73733	0.42533	...	50.43806	49.96265	48.69224	
11	3.28901	-5.57443	-2.99043	0.38820	...	51.05585	49.14245	45.55431	
12	-11.69090	-3.59276	-2.43997	0.41174	...	49.52544	47.64045	43.74761	
13	-2.79504	-3.25991	-4.93606	0.42062	...	46.06212	42.95131	37.65702	
14	-3.03500	3.45620	10.50409	0.58757	...	51.36513	49.26195	45.74891	
15	-1.19320	-0.46504	3.16119	0.56372	...	55.34923	54.00650	51.97891	
16	1.97670	-21.83607	23.35004	0.76664	...	50.94696	50.21058	48.47399	
17	-1.15681	3.60307	-7.38734	0.38510	...	56.90596	54.94726	51.96329	
18	-11.89833	11.82170	-6.96498	0.38592	...	47.49746	45.95893	43.75467	
19	8.75426	-2.45750	5.94624	0.52879	...	57.65994	55.80597	51.91113	

20	-2.94365	-10.65624	-3.52717	0.66397	...	66.41776	64.10250	59.25031
21	3.07901	-5.45736	-2.37579	0.68636	...	49.46564	48.23407	45.67922
22	-9.41676	-2.76792	5.20498	0.59595	...	51.07666	48.65738	44.50696
23	-10.58146	-3.18401	4.54324	0.40091	...	55.24992	53.56933	50.93834
24	3.90216	-4.74276	-7.52598	0.60697	...	55.69708	53.66169	49.48473

	y_64	x_65	y_65	x_66	y_66	x_67	y_67
0	46.27273	46.40865	46.00894	46.60680	50.75081	51.32491	51.05518
1	51.52741	51.49037	50.96785	51.85685	56.96234	57.72325	57.56163
2	45.12809	45.53594	45.24871	46.46969	52.72993	53.38613	52.86687
3	43.50664	43.55758	43.30449	44.39903	44.52084	44.76995	44.70442
4	37.92926	38.00440	37.64405	38.52496	42.30593	42.80628	42.65378
5	44.41958	44.80170	44.49074	44.53502	48.44263	48.82518	48.43996
6	49.16213	49.17820	48.72146	49.05509	50.42355	50.94271	50.93750
7	36.69713	36.84042	36.37741	37.65301	44.65634	45.46573	45.01655
8	46.56854	46.30600	45.57546	45.20967	49.20493	50.03883	50.30930
9	30.53996	30.53202	30.34129	31.65026	33.93624	34.24808	34.14170
10	47.96085	48.09454	47.90096	48.27900	47.64490	47.77638	47.70977
11	42.00516	41.87609	41.45588	43.41926	46.94834	47.59470	47.50583
12	42.05176	41.99336	41.54742	42.44148	45.96036	46.56089	46.45398
13	36.21613	36.31800	35.98825	37.20504	43.15082	43.73235	43.38222
14	44.22952	44.49087	44.23931	45.56267	48.54833	48.88363	48.53484
15	52.52243	52.92773	52.46838	51.96843	52.84522	53.12783	52.73565
16	46.18891	45.88105	45.45810	46.26224	48.10161	48.54254	48.72901
17	51.07078	51.56541	51.69044	53.40136	53.24985	53.17471	52.64225
18	44.41694	44.93060	44.99136	45.43933	45.18035	45.09390	44.62331
19	47.84491	47.64039	47.15610	49.27141	54.18355	54.91880	54.88247
20	55.72463	55.40406	54.66655	55.65281	60.50422	61.52358	61.60235
21	42.61480	42.51348	42.25055	44.07388	45.73324	46.14311	46.11675
22	43.27282	43.44377	43.02880	43.59578	47.56894	48.06598	47.81011
23	50.30189	50.49053	50.23031	50.68097	52.02252	52.28159	51.98402
24	45.76706	45.73327	45.46433	48.08038	51.69023	52.23615	52.10125

[25 rows x 165 columns]