
That's it, You're ready to go!

Datasets

Half of the dataset used in this project is from the FaceForensics deepfake detection dataset. .

To download this data, please make sure to fill out the google form to request access to the data.

For the dataset that we collected from Youtube, it is accessible on S3 for download.

To automatically download and restructure both datasets, please execute:

```
1 bash restructure_data.sh faceforensics_download.py
```

Note: You need to have received the download script from FaceForensics++ people before executing the restructure script.

Note2: We created the `restructure_data.sh` to do a split that replicates our exact experiments available in the UI above, please feel free to change the splits as you wish.

Walkthrough

Before starting to train/evaluate models, we should first create the docker image that we will be running our experiments with. To do so, we already prepared a dockerfile to do that inside `custom_docker_image`. To create the docker image, execute the following commands in terminal:

```
1 cd custom_docker_image
2 nvidia-docker build . -t atlas_ff
```

Note: if you change the image name, please make sure you also modify line 16 of `job.config.yaml` to match the docker image name.

Inside `job.config.yaml`, please modify the data path on host from `/media/biggie2/FaceForensics/datasets/` to the absolute path of your `datasets` folder.

The folder containing your datasets should have the following structure:

```
1 datasets|—
2   augment_deepfake      (2) |
3   |— fake|
4   |   |— frames|
5   |— real|
6   |   |— frames|
7   |— val|
```

8	fake	
9	real	
10	base_deepfake	(1)
11	fake	
12	frames	
13	real	
14	frames	
15	val	
16	fake	
17	real	
18	both_deepfake	(3)
19	fake	
20	frames	
21	real	
22	frames	
23	val	
24	fake	
25	real	
26	precomputed	(4)
27	T_deepfake	(0)
28	manipulated_sequences	
29	DeepFakeDetection	
30	Deepfakes	
31	Face2Face	
32	FaceSwap	
33	NeuralTextures	
34	original_sequences	
35	actors	
36	youtube	

Notes: * (0) is the dataset downloaded using the FaceForensics repo scripts * (1) is a reshaped version of FaceForensics data to match the expected structure by the codebase. subfolders called `frames` contain frames collected using `ffmpeg` * (2) is the augmented dataset, collected from youtube, available on s3. * (3) is the combination of both base and augmented datasets. * (4) precomputed will be automatically created during training. It holds cached cropped frames.

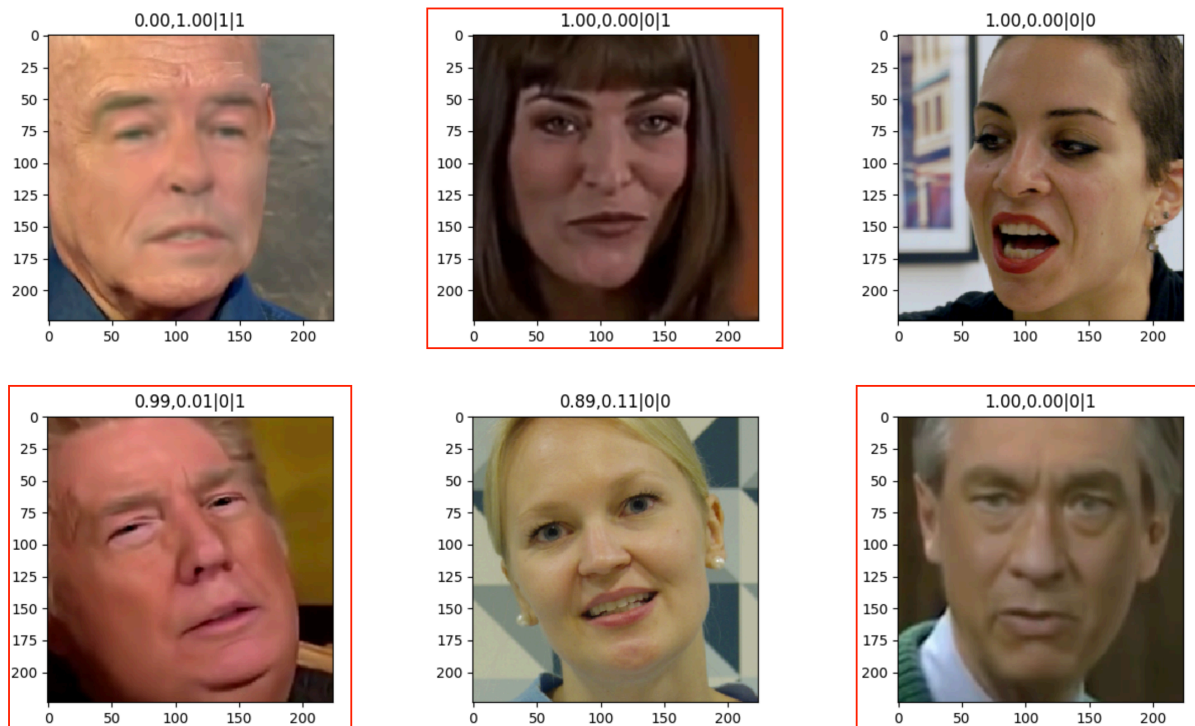
Then, to run all the experiments we will show in the article to come, you can launch the script `hparams_search.py` using:

```
1 python hparams_search.py
```

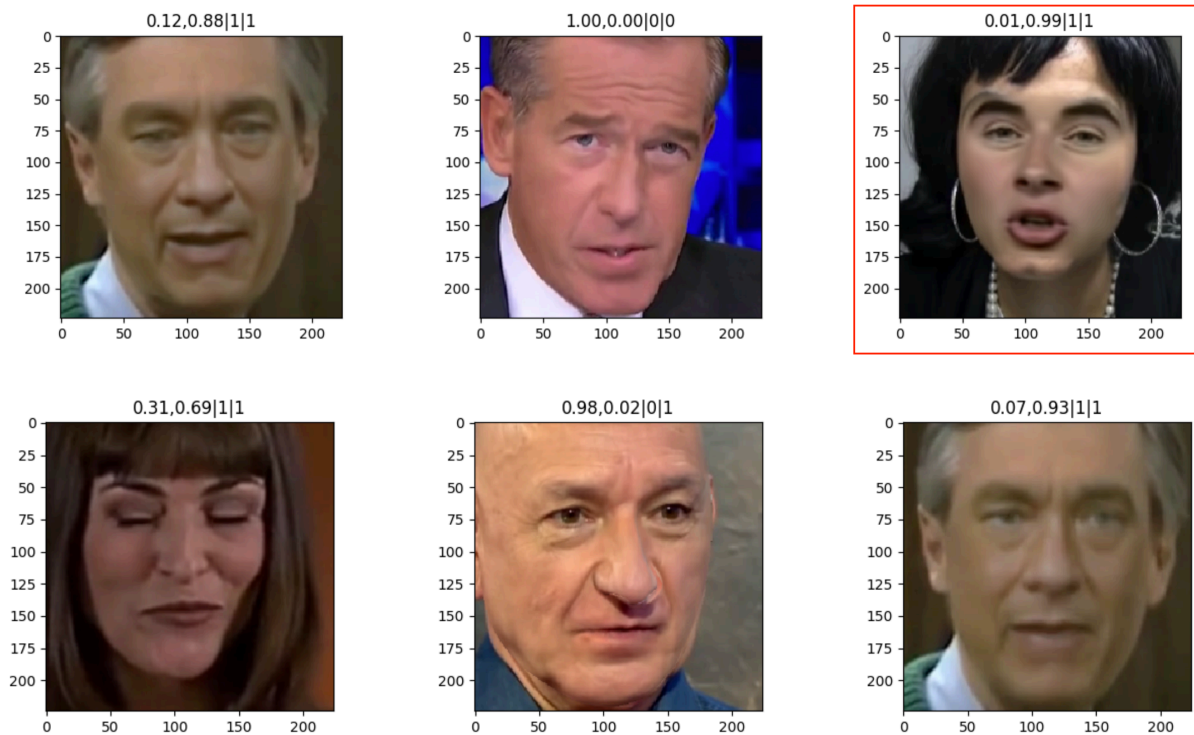
Results

In the following pictures, the title for each subplot is in the form `real_prob, fake_prob | prediction | label`.

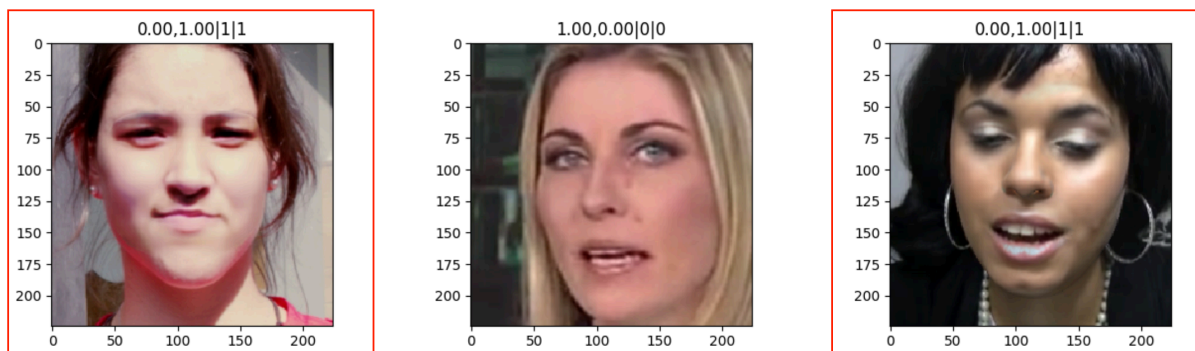
Model trained on FaceForensics++ dataset For models trained on the paper dataset alone, we notice that the model only learns to detect the manipulation techniques mentioned in the paper and misses all the manipulations in real world data (from data)

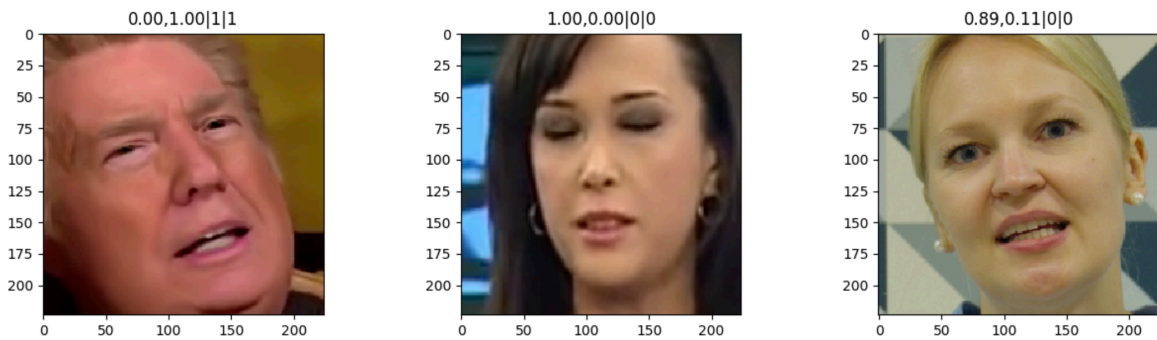


Model trained on Youtube dataset Models trained on the youtube data alone learn to detect real world deepfakes, but also learn to detect easy deepfakes in the paper dataset as well. These models however fail to detect any other type of manipulation (such as NeuralTextures).



Model trained on Paper + Youtube dataset Finally, models trained on the combination of both datasets together, learns to detect both real world manipulation techniques as well as the other methods mentioned in FaceForensics++ paper.





for a more in depth explanation of these results, please refer to the article we published. More results can be seen in the interactive UI

Help improve this technology

Please feel free to fork this work and keep pushing on it.

If you also want to help improving the deepfake detection datasets, please share your real/forged samples at foundations@dezza.com.

LICENSE

© 2020 Square, Inc. ATLAS, DESSA, the Dessa Logo, and others are trademarks of Square, Inc. All third party names and trademarks are properties of their respective owners and are used for identification purposes only.