High-Performance DeepFake Video Classifier

Help people to identify Deepfake videos with 87 % accuracy

Yunting Chiu

Related Works

Deepfake Representation with Multilinear Regression

- The authors use SVM classification to complete the video classifier. The accuracy is around 82 %

Protecting World Leaders Against Deep Fakes

- The authors use the OpenFace2 toolkit to classify various facial features such as the mouth, nose, and lip. The partial accuracy is over 90%

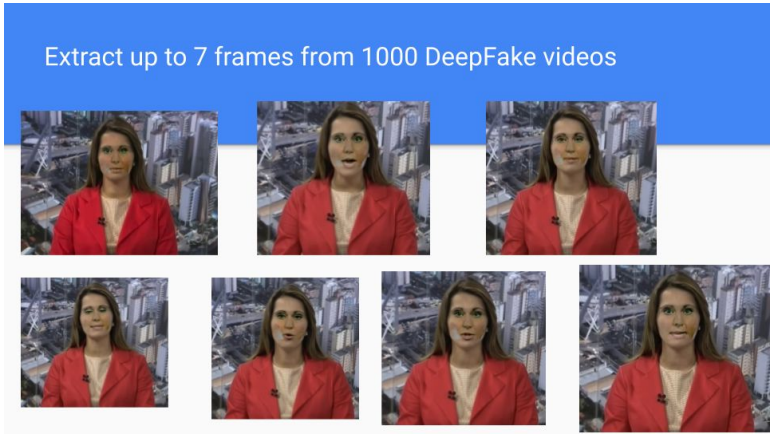
[1] Abdali, S., Vasilescu, M. A. O., & Papalexakis, E. E. (2021). Deepfake Representation with Multilinear Regression. *arXiv preprint arXiv:2108.06702*.

[2] Agarwal, S., Farid, H., Gu, Y., He, M., Nagano, K., & Li, H. (2019, June). Protecting World Leaders Against Deep Fakes. In *CVPR workshops (Vol. 1)*.

Dataset

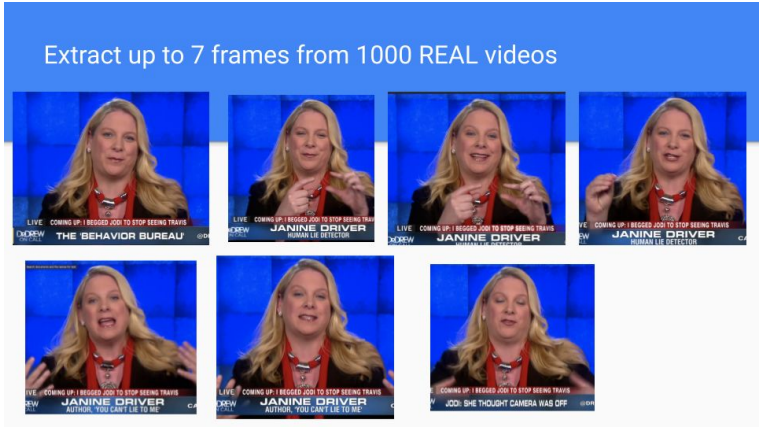
- Download FaceForensics++ from GitHub
- 1000 real videos and 1000 Deepfake videos
- TV reporters and journalists, YouTube
- Various sexes, ages, and races.

Data Preprocessing



Extract Video Frames and Save to Images

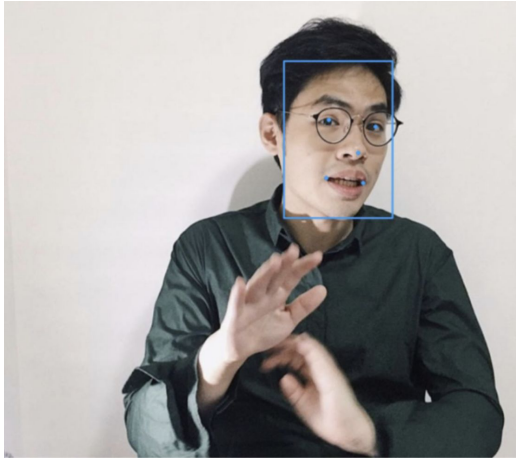
- Capture 1 frame in every 30 frames
- Different facial expressions
- Extract up to 7 frame in a single video
- 7000 images from the Deepfake set
- 7000 images from the Real set



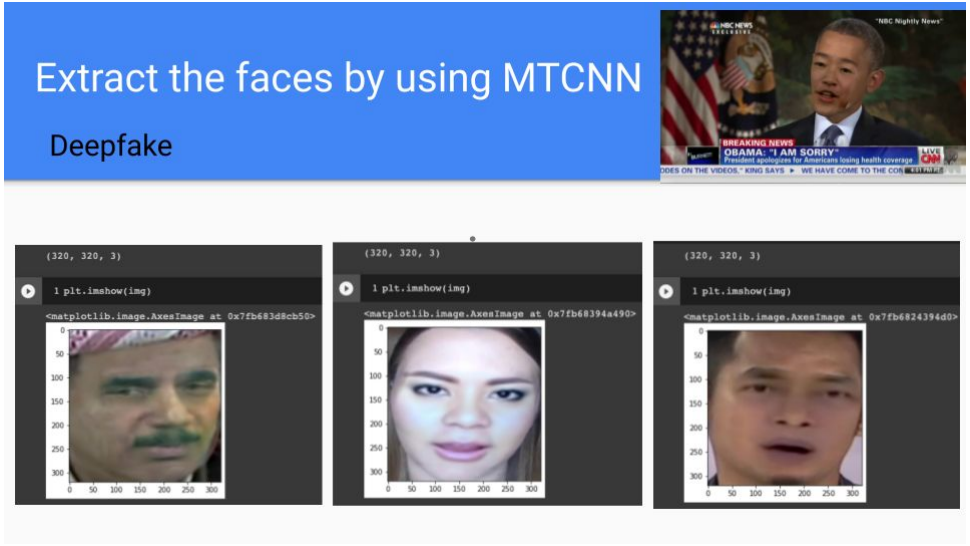
Data Preprocessing

Facial Extraction using pre-trained MTCNN model

- Provide exact pixel positions to keep face boundaries intact
- If MTCNN cannot detect the face, remove the image
- After MTCNN, there are 6984 images in the Deepfake set
- After MTCNN, there are 7000 images in the Real set
- The number of observations: 13984 images



Deepfake set



Real set

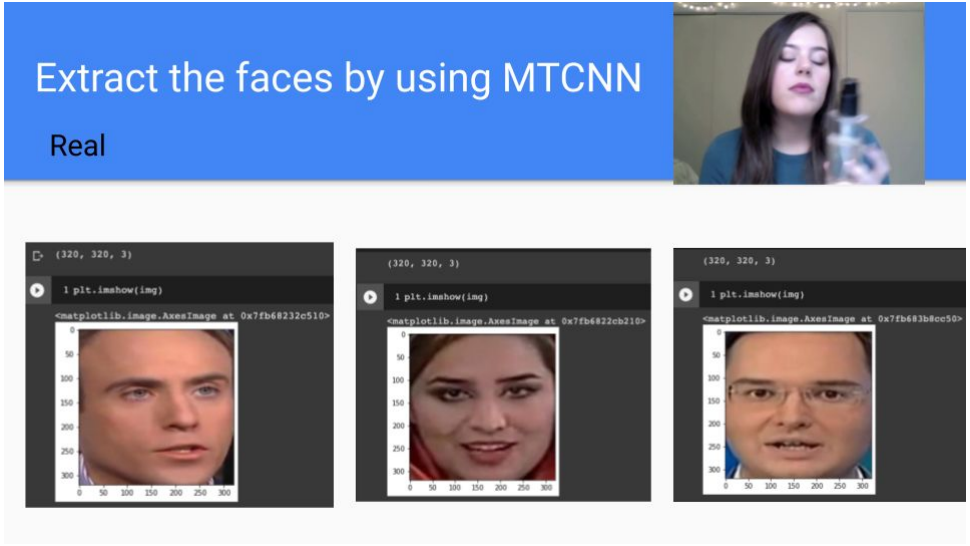
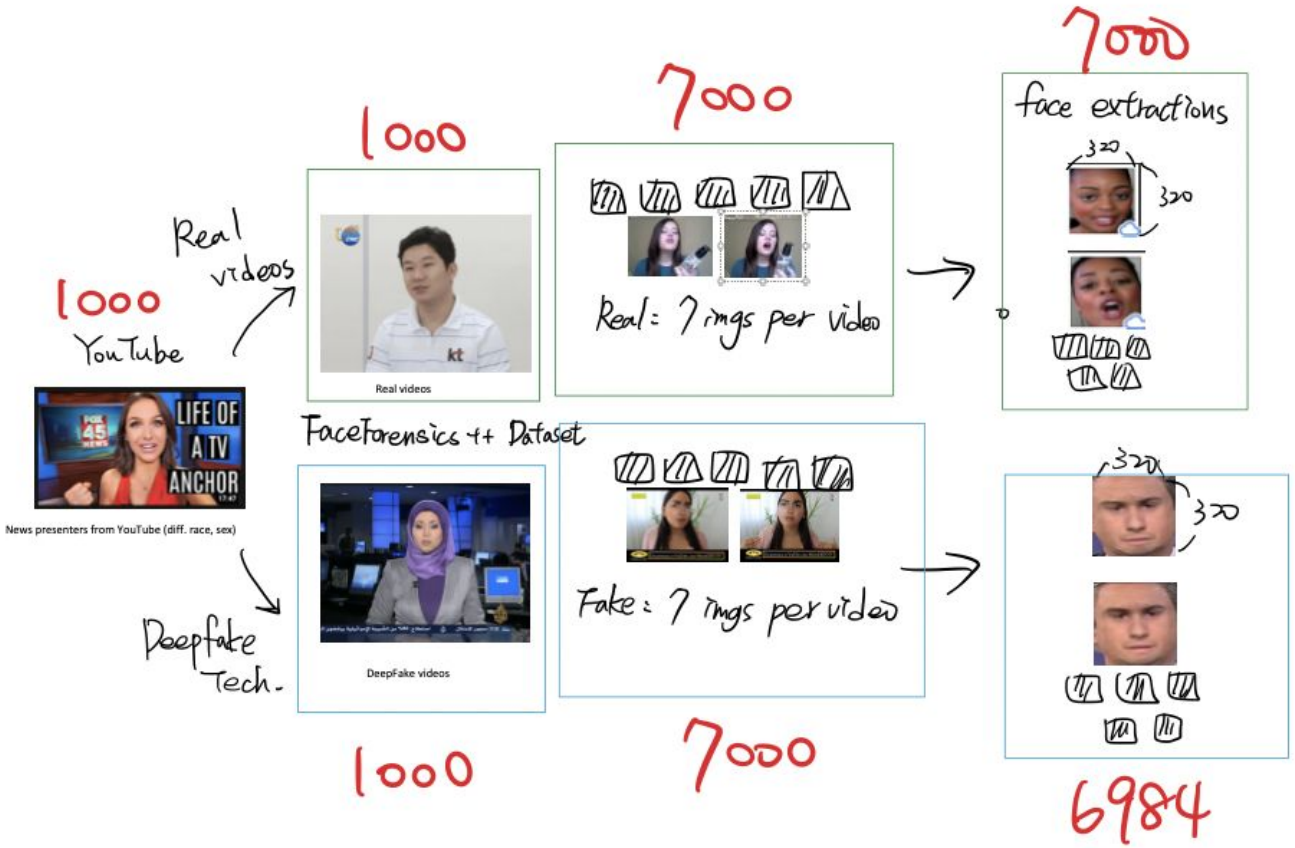

Extract the faces by using MTCNN
Real
(320, 320, 3)
1 plt.imshow(img)
<matplotlib.image.AxesImage at 0x7fb68232c510>
(320, 320, 3)
1 plt.imshow(img)
<matplotlib.image.AxesImage at 0x7fb6822cb210>
(320, 320, 3)
1 plt.imshow(img)
<matplotlib.image.AxesImage at 0x7fb683b8cc50>


image_872.jpg
image_873.jpg
image_874.jpg
image_875.jpg
image_876.jpg
image_877.jpg
image_878.jpg
image_879.jpg
image_880.jpg
image_881.jpg
image_882.jpg
image_883.jpg
image_884.jpg
image_885.jpg
image_886.jpg
image_887.jpg
image_888.jpg
image_889.jpg
image_890.jpg
image_891.jpg
image_892.jpg
image_893.jpg
image_894.jpg
image_895.jpg
image_896.jpg
image_897.jpg
image_898.jpg
image_899.jpg
image_900.jpg
image_901.jpg
image_902.jpg
image_903.jpg
image_904.jpg
image_905.jpg
image_906.jpg
image_907.jpg
image_908.jpg
image_909.jpg
image_910.jpg
image_911.jpg

Workflow


1000
YouTube
Real videos
LIFE OF A TV ANCHOR
News presenters from YouTube (diff. race, sex)
Deepfake Tech.
1000
Real videos
FaceForensics++ Dataset
DeepFake videos
7000
Real: 7 imgs per video
7000
Fake: 7 imgs per video
7000
face extractions
320
320
320
320
6984

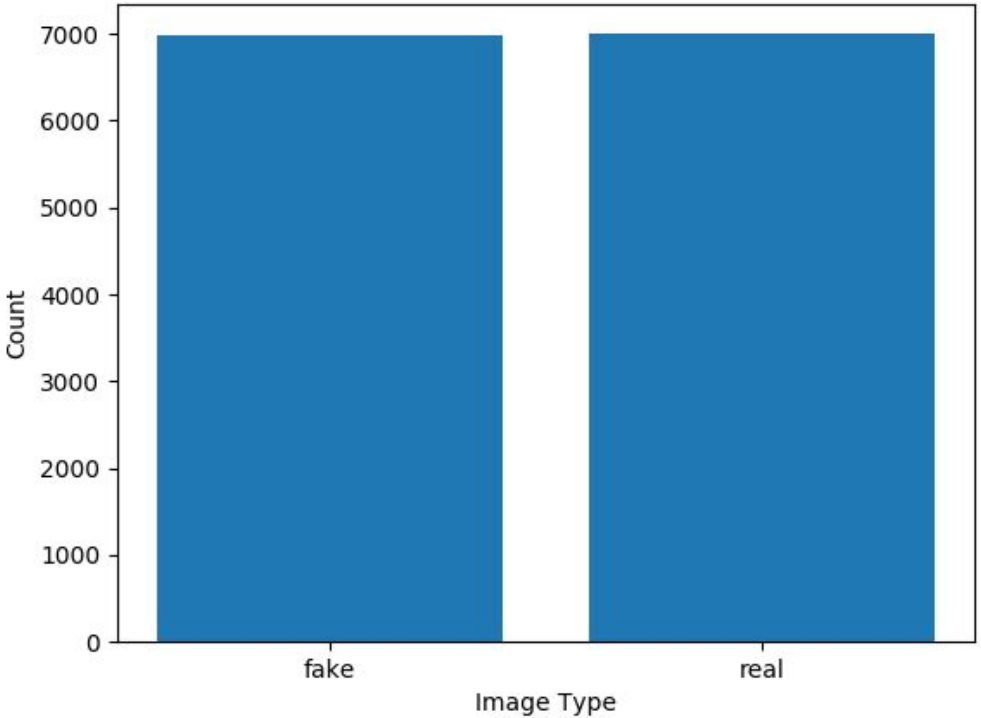
Data Preprocessing

Construct X Features and y Labels

- Flatten 320 x 320 x 3 to 1 x 307,200 as X features
- Label “fake” and “real” as y labels
- Data dimension: 13984 rows and 307201 columns
- Supervised Learning

```
[[array([134, 131, 116, ...,  68,  60,  71], dtype=uint8) 'fake']
 [array([133, 130, 115, ...,  71,  59,  71], dtype=uint8) 'fake']
 [array([117, 113, 112, ...,  43,  31,  45], dtype=uint8) 'fake']
 ...
 [array([ 33,  20,  66, ..., 188, 155, 172], dtype=uint8) 'real']
 [array([ 51,  28,  46, ..., 116,  52,  50], dtype=uint8) 'real']
 [array([174, 140, 102, ...,  23,  39,  98], dtype=uint8) 'real']]
```

Data Visualization

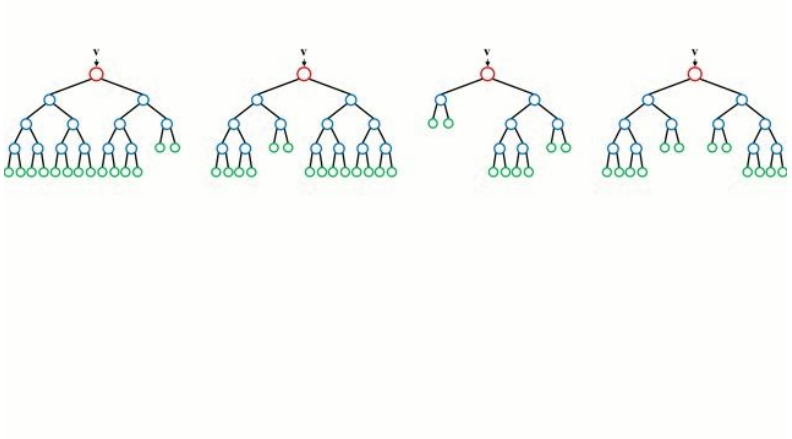


Model Selection

Random Forest Model

- It can handle large features efficiently
- It is not necessary to normalize data points before running the model
- It can reduce computation time

pc: wikimedia.org



Model Optimization

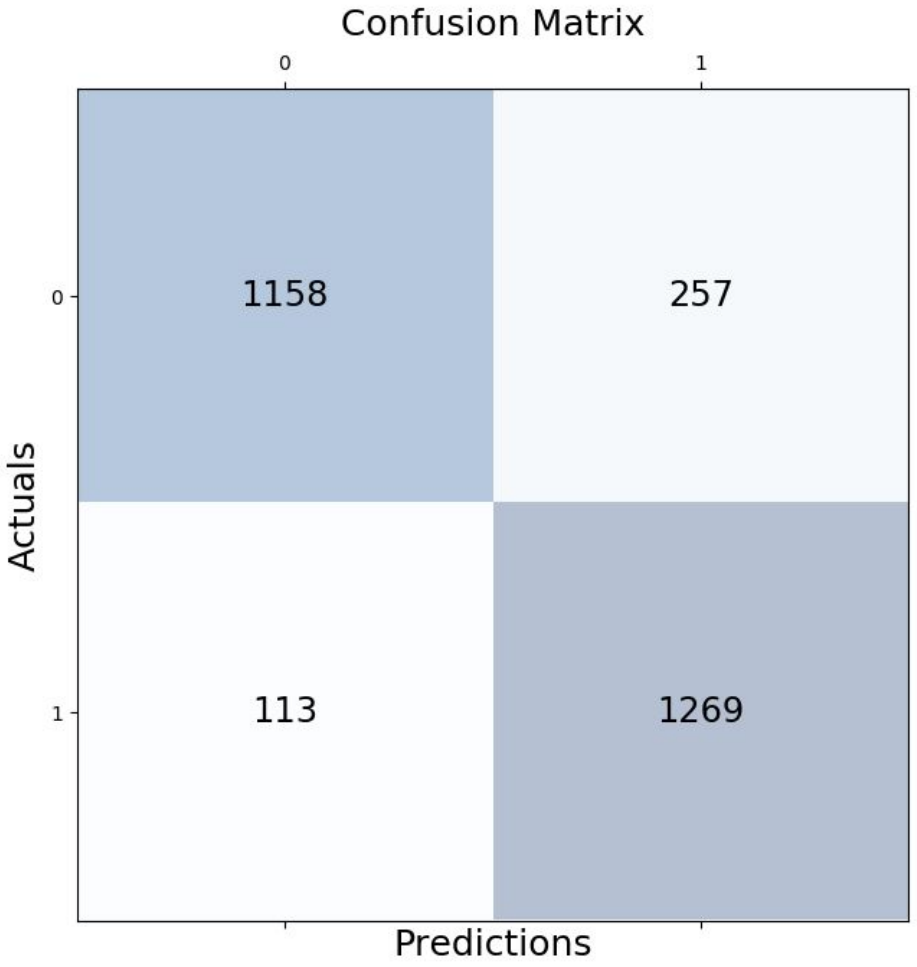
Grid Search and Cross-Validation

- 80 % training, 20 % testing
- Grid search for number of trees: [100, 500, 1000, 2000]
- 5-fold cross validation + bootstrap

```
Best Model: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                    max_depth=None, max_features='auto', max_leaf_nodes=None,
                    min_impurity_decrease=0.0, min_impurity_split=None,
                    min_samples_leaf=1, min_samples_split=2,
                    min_weight_fraction_leaf=0.0, n_estimators=2000,
                    n_jobs=None, oob_score=False, random_state=42, verbose=0,
                    warm_start=False)
```

Confusion Matrix

- 0 represents fake
- 1 represents real



Classification Report

	precision	recall	f1-score	support
fake	0.91	0.82	0.86	1415
real	0.83	0.92	0.87	1382
accuracy			0.87	2797
macro avg	0.87	0.87	0.87	2797
weighted avg	0.87	0.87	0.87	2797

Improvements

- Require more HPC resources
  - HPC would automatically cancel the submission if computation time more than 2 days
  - Add more numbers of grid search and cross-validation
  - Tuning: max_leaf_nodes, min_samples_leaf instead of n_estimators only
- Deep Learning Task
  - Construct a neural network architecture
  - Include activation functions in the hidden layers
  - Grid search to determine the optimal number of epochs and batch size

Computing Environment

- At least Python 3.6.9
- Google Colab free version - Python notebook files
- High Performance Computing - Python script files
- Source Code: ***twyunting/Deepfake_Video_Classifier*** - *GitHub*

pc: GitHub & Google

References

1. Abdali, S., Vasilescu, M. A. O., & Papalexakis, E. E. (2021). Deepfake Representation with Multilinear Regression. *arXiv preprint arXiv:2108.06702*.
2. Agarwal, S., Farid, H., Gu, Y., He, M., Nagano, K., & Li, H. (2019, June). Protecting World Leaders Against Deep Fakes. In *CVPR workshops (Vol. 1)*.
3. Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 1-11).
4. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. I*EEE Signal Processing Letters*, *23*(10), 1499-1503.

THANKS!

Any questions?