National College of Ireland

Configuration Manual

Research Project

Msc in Data Analytics

Ajay Kumar Kommalapati
Student ID: 20168829

School of Computing
National College of Ireland

Supervisor: Arghir Nicolae Moldovan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Ajay Kumar Kommalapati

Student ID: 20168829

Programme: MSc in data anlytics **Year:** 2021

Module: Research Project

Lecturer: Arghir Nicolae Moldovan

Submission Due Date: 31-1-2022

Project Title: Comparative study of state of the art deepfake detection models

Word Count: 1246 **Page Count:** 14

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Ajay Kumar Kommalapati

Date: 16-12-2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Ajay Kumar Kommalapati
20168829

1 State of art: DefakeHop

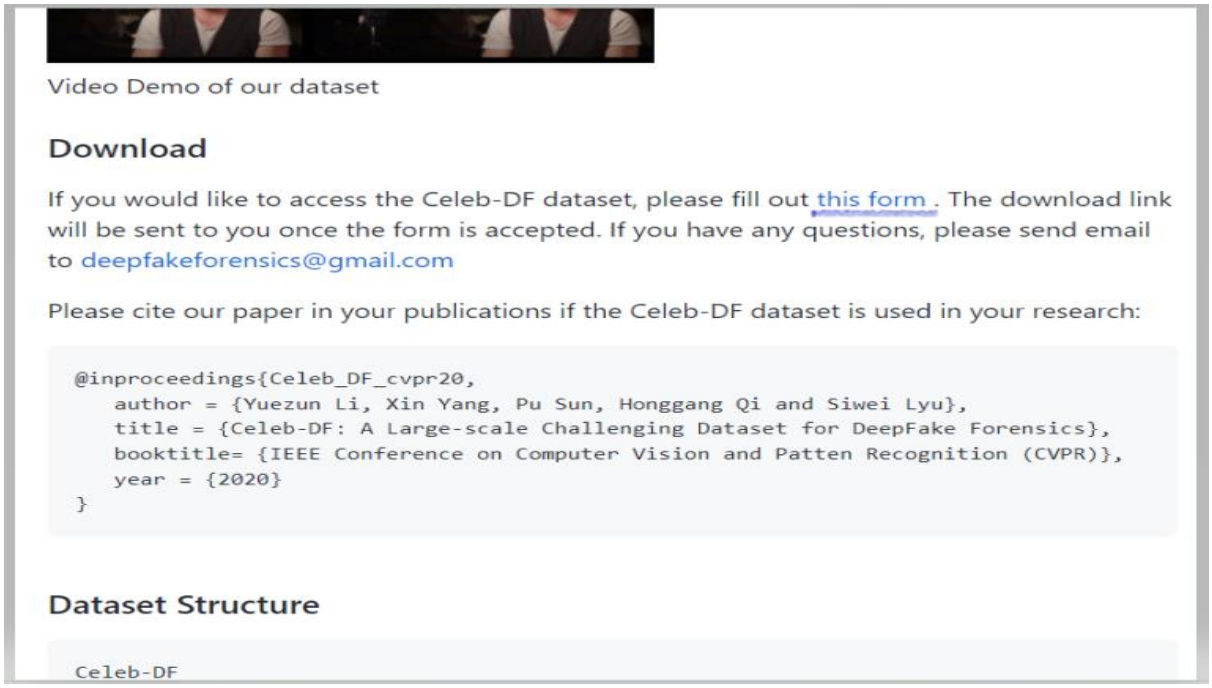
In this manual I'm going to list and note down the steps taken to install , setup and build the state of art which is used for deepfake detection and about the two online scanners.
DefakeHop[1] setup:
DefakeHop is available in github in this link: https://github.com/hongshuochen/DefakeHop
Required packages to install.

```
pip install opencv-python
pip install scikit-image
pip install matplotlib
pip install scikit-learn
pip install pandas
pip install xgboost
```

In this study I have been using the celeb df v1 as a dataset. For that we need to do steps

Link: https://github.com/yuezunli/celeb-deepfakeforensics/tree/master/Celeb-DF-v1

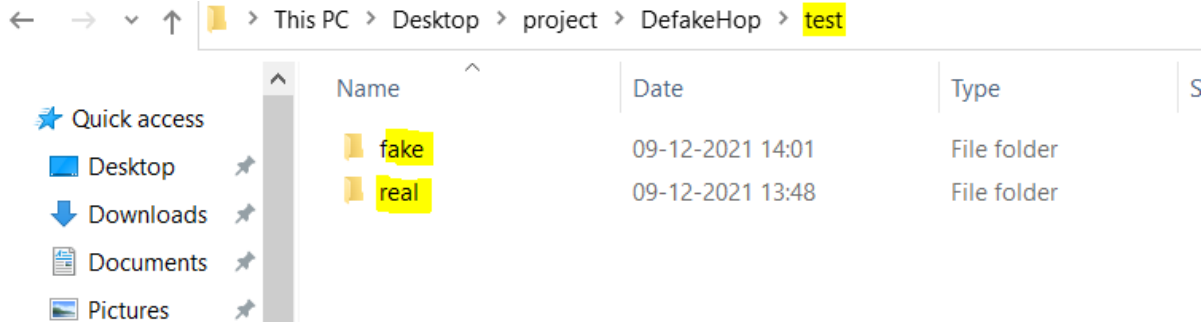
Need to visit the link in that can find the form link



After filling few required filed it will navigate to the drive where we can download the dataset.

__pycache__	01-12-2021 01:30	File folder	
data	15-12-2021 20:17	File folder	
img	25-11-2021 10:13	File folder	
openface	13-12-2021 14:08	File folder	
test	09-12-2021 14:01	File folder	
train	09-12-2021 14:23	File folder	
data	11-12-2021 12:06	PY File	3 KB
defakeHop	22-11-2021 15:20	PY File	8 KB
face_aligner	22-11-2021 15:20	PY File	4 KB
landmark_extractor	01-12-2021 01:26	PY File	2 KB
model	15-12-2021 13:45	PY File	9 KB
multi_cwSaab	22-11-2021 15:20	PY File	7 KB
patch_extractor	15-12-2021 10:23	PY File	4 KB
README.md	22-11-2021 15:20	MD File	3 KB
saab	22-11-2021 15:20	PY File	6 KB
utils	22-11-2021 15:20	PY File	2 KB

This is the folder structure of defakeHop when we download. The dataset is arranged in way such train video and test video, again there nested folders like fake and real.



This is the test folder this is the way train is also arranged.
The dataset is the breakdown into test and train with the help of the a document file called *List_of_testing_videos ,* which provides the list of test videos. Based on this file I have divided.
And also there is another installation which is *openface* [2] which is used to extract the faces from the video.
Link: https://github.com/TadasBaltrusaitis/OpenFace
After downloading there other supporting which need to download from links based 32 bit or 64 bit operating system.

- https://github.com/TadasBaltrusaitis/OpenFace/releases/download/OpenFace_2.2.0/OpenFace_v2.2.0_win_x86.zip
- https://github.com/TadasBaltrusaitis/OpenFace/releases/download/OpenFace_2.2.0/OpenFace_v2.2.0_win_x64.zip

After downloading those files need to place in the folder structure of open/patch_experts

Preprocessing of DefakeHop:

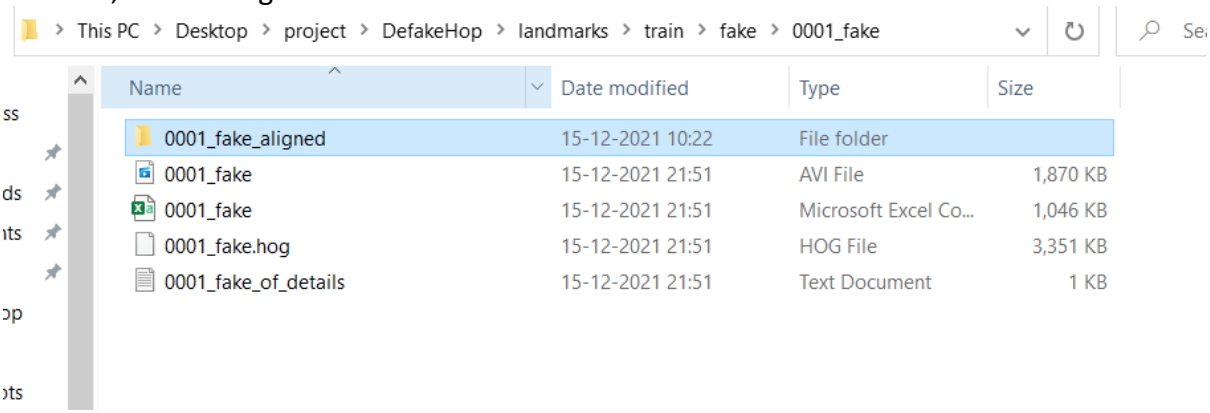
Here the preprocessing takes place initially with the landmark_extractor.

```
(deepfakes_venv) C:\Users\ragha\Desktop\project\DefakeHop>python landmark_extractor.py
Input:  C:\Users\ragha\Desktop\project\DefakeHop\train\fake\0000_fake.mp4
OpenFace\FeatureExtraction -f C:\Users\ragha\Desktop\project\DefakeHop\train\fake\0000_fake.mp4 -out_dir C:\Users\ragha
Desktop\project\DefakeHop\landmarks\train\fake\0000_fake
Input:  C:\Users\ragha\Desktop\project\DefakeHop\train\fake\0001_fake.mp4
OpenFace\FeatureExtraction -f C:\Users\ragha\Desktop\project\DefakeHop\train\fake\0001_fake.mp4 -out_dir C:\Users\ragha
Desktop\project\DefakeHop\landmarks\train\fake\0001_fake
Input:  C:\Users\ragha\Desktop\project\DefakeHop\train\real\0000.mp4
OpenFace\FeatureExtraction -f C:\Users\ragha\Desktop\project\DefakeHop\train\real\0000.mp4 -out_dir C:\Users\ragha\Desk
op\project\DefakeHop\landmarks\train\real\0000
Input:  C:\Users\ragha\Desktop\project\DefakeHop\train\real\0001.mp4
OpenFace\FeatureExtraction -f C:\Users\ragha\Desktop\project\DefakeHop\train\real\0001.mp4 -out_dir C:\Users\ragha\Desk
op\project\DefakeHop\landmarks\train\real\0001
Input:  C:\Users\ragha\Desktop\project\DefakeHop\test\real\0039.mp4
OpenFace\FeatureExtraction -f C:\Users\ragha\Desktop\project\DefakeHop\test\real\0039.mp4 -out_dir C:\Users\ragha\Deskt
p\project\DefakeHop\landmarks\test\real\0039
Input:  C:\Users\ragha\Desktop\project\DefakeHop\test\real\0040.mp4
OpenFace\FeatureExtraction -f C:\Users\ragha\Desktop\project\DefakeHop\test\real\0040.mp4 -out_dir C:\Users\ragha\Deskt
p\project\DefakeHop\landmarks\test\real\0040
Input:  C:\Users\ragha\Desktop\project\DefakeHop\test\fake\0039_fake.mp4
OpenFace\FeatureExtraction -f C:\Users\ragha\Desktop\project\DefakeHop\test\fake\0039_fake.mp4 -out_dir C:\Users\ragha\D
esktop\project\DefakeHop\landmarks\test\fake\0039_fake
Input:  C:\Users\ragha\Desktop\project\DefakeHop\test\fake\0040_fake.mp4
OpenFace\FeatureExtraction -f C:\Users\ragha\Desktop\project\DefakeHop\test\fake\0040_fake.mp4 -out_dir C:\Users\ragha\D
esktop\project\DefakeHop\landmarks\test\fake\0040_fake
Output: C:\Users\ragha\Desktop\project\DefakeHop\landmarks\test\real\0039
Output: C:\Users\ragha\Desktop\project\DefakeHop\landmarks\test\fake\0039_fake
Output: C:\Users\ragha\Desktop\project\DefakeHop\landmarks\train\fake\0001_fake
```

In this particular step I have executed the patch_extractor in this code there executing the openface.

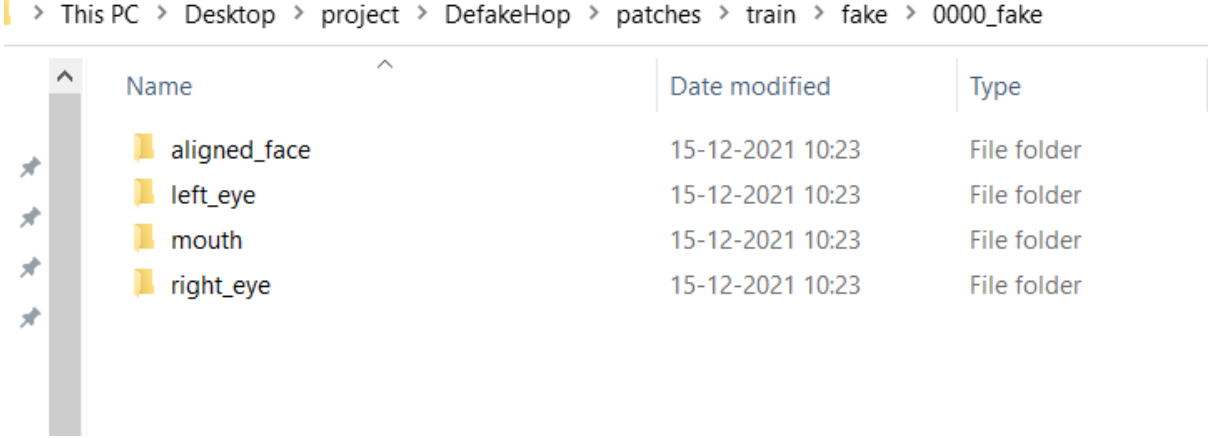
```
print(r"OpenFace\FeatureExtraction -f {input} -out_dir {output}".format(input = file_path, output = output_dir))
stream = os.popen(r"OpenFace\FeatureExtraction -f {input} -out_dir {output}"
        .format(input = file_path, output = output_dir))
output = stream.read()
print("Output:", output_dir)
```

In this openface input is given as video and output is the landmark folder with video id as folder name.in that folder there will be the frames which cropped from the video and other files csv , AVI and log file



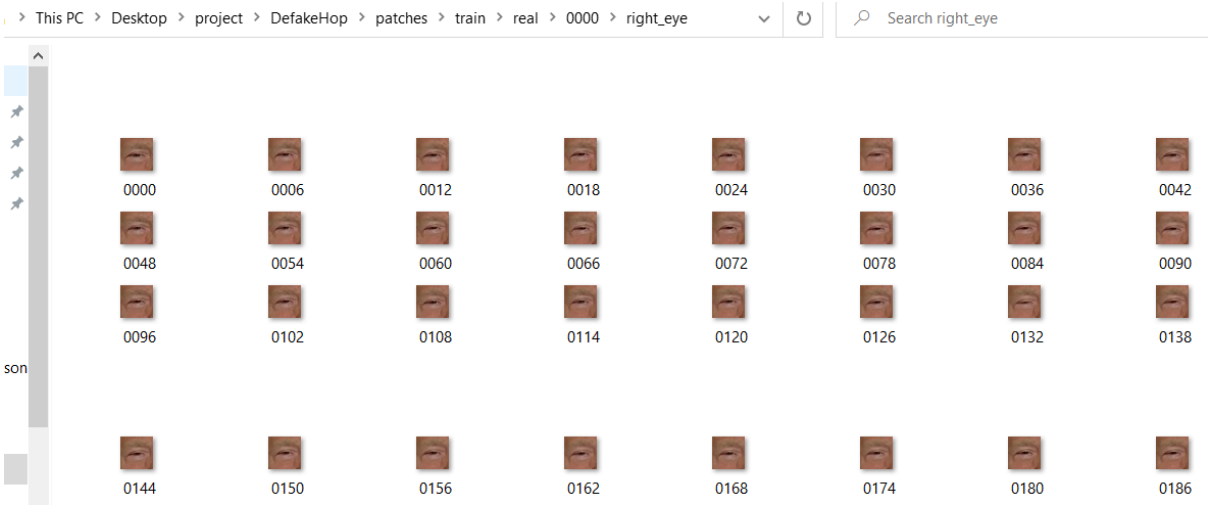
Next step is executing the *patch_extractor.py*. take input as the video and created the patch file and inside it created with test ot train and fake or real.

```
(deepfakes_venv) C:\Users\ragha\Desktop\project\DefakeHop>python patch_extractor.py
Input:  train/fake\0000_fake.mp4
Output: patches\train/fake\0000_fake
Input:  train/fake\0001_fake.mp4
Output: patches\train/fake\0001_fake
Input:  train/real\0000.mp4
Output: patches\train/real\0000
Input:  train/real\0001.mp4
Output: patches\train/real\0001
Input:  test/real\0039.mp4
Output: patches\test/real\0039
Input:  test/real\0040.mp4
Output: patches\test/real\0040
Input:  test/fake\0039_fake.mp4
Output: patches\test/fake\0039_fake
Input:  test/fake\0040_fake.mp4
Output: patches\test/fake\0040_fake
```

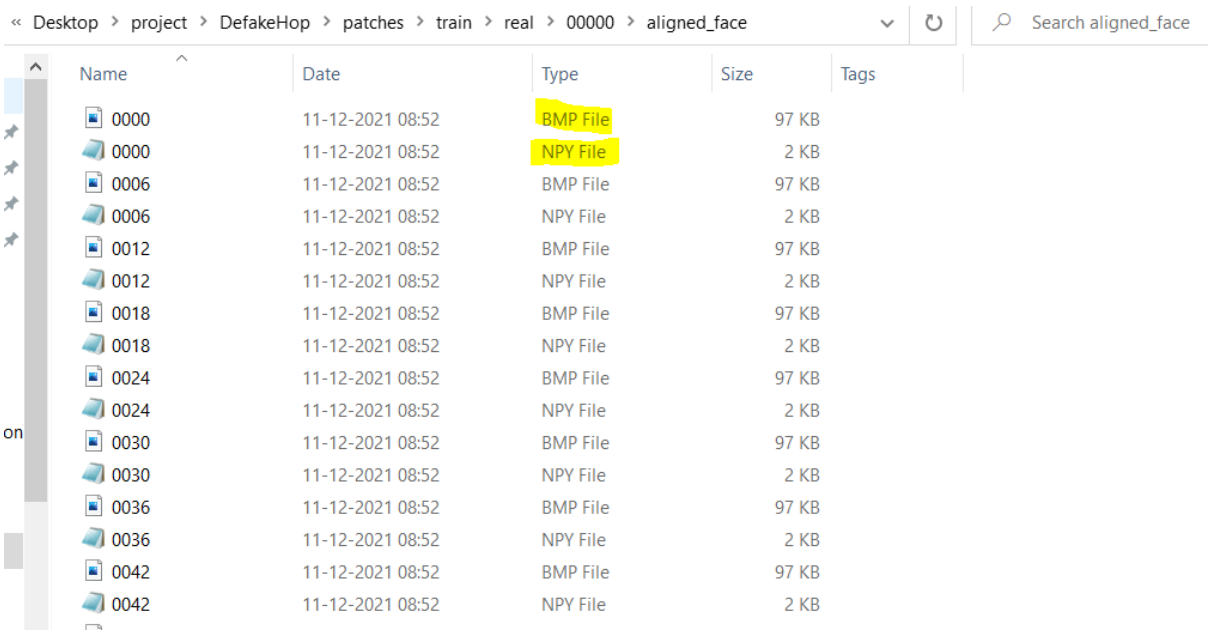


After executing the patch_extractor it created the 4 folders.

Again inside that patch is extractor as per the regions like left_eye , right_eye and mouth.
But inside the aligned face along with .bmp file it also created the .npy file



This is right eye folder



This is the aligned face folder.

After preprocessing now need to created the .npz by using the data.py file.
It's the input as patch files and produces the .npz files.

```
(deepfakes_venv) C:\Users\ragha\Desktop\project\DefakeHop>python data.py
left_eye
right_eye
mouth
left_eye
right_eye
mouth
```

It produces the .npz files for test and train separately. These are created based on the regions provided in data.py.

After this the model.py file.
It produces the AUC values of the frames and videos separate.

```
Output shape: 13
Input shape: (3, 3) 9
Output shape: 5
===============Soft Classifiers===============
Output shape: (75235, 30)
===============Concatenation===============
===============Prediction===============
Features shape: (69676, 360)
===============Training Results===============
Frame AUC 0.9892520269128884
Video AUC 0.9976770768039527
===============Testing Results===============
Frame AUC 0.9396865235569321
Video AUC 0.9465895140235302
```

Online scanners:

There are few computational power limitations while running the DefakeHop. Still generating the npz file which is the last but final step it was executed but it took a lot of time to finish but in the final step it took around 10 hours for the first time and later on it took a day and another time it struck in between at final attempt program get terminated and system get switched off. For a trail I have install in the MacOS and try to executed it, after few minutes it took full space in the RAM and started occupying main memory it about to occupy 25gb and immediately it get terminated by itself due to less space. I even gave try in the colab and same thing happened.

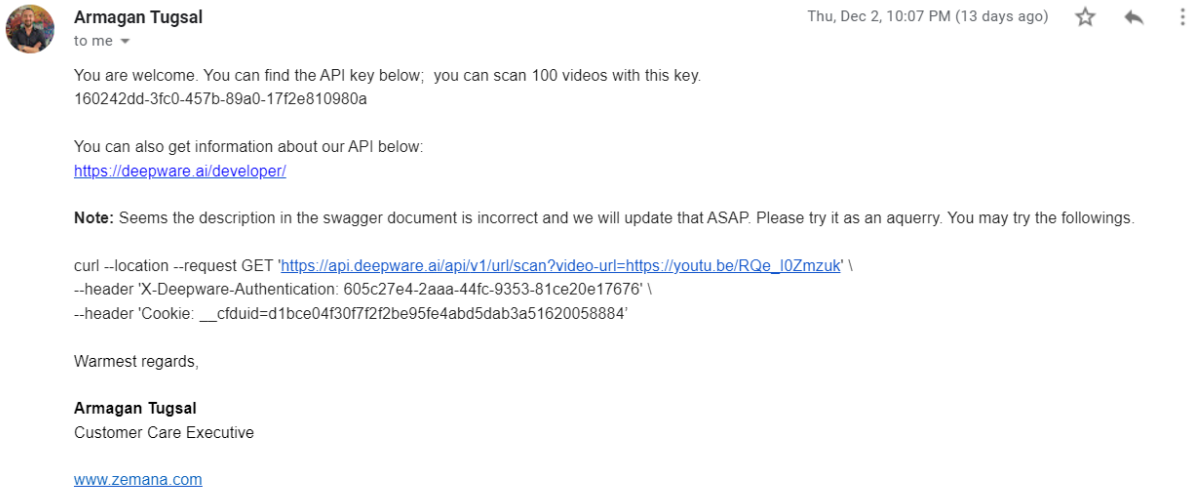
After a less than a minute It get termiated by itslef.

I have been used to online scanner which detect the deep fake in videos.

1. Deepware [3]
2. WeVerify [4]

2 Deepware

This portal allows us to upload the video and can check the deepfake in video. Regarding the access of deepware I have mailed and here is their reply.



Armagan Tugsal Thu, Dec 2, 10:07 PM (13 days ago)

to me

You are welcome. You can find the API key below; you can scan 100 videos with this key.
160242dd-3fc0-457b-89a0-17f2e810980a

You can also get information about our API below:
https://deepware.ai/developer/

Note: Seems the description in the swagger document is incorrect and we will update that ASAP. Please try it as an aquerry. You may try the followings.

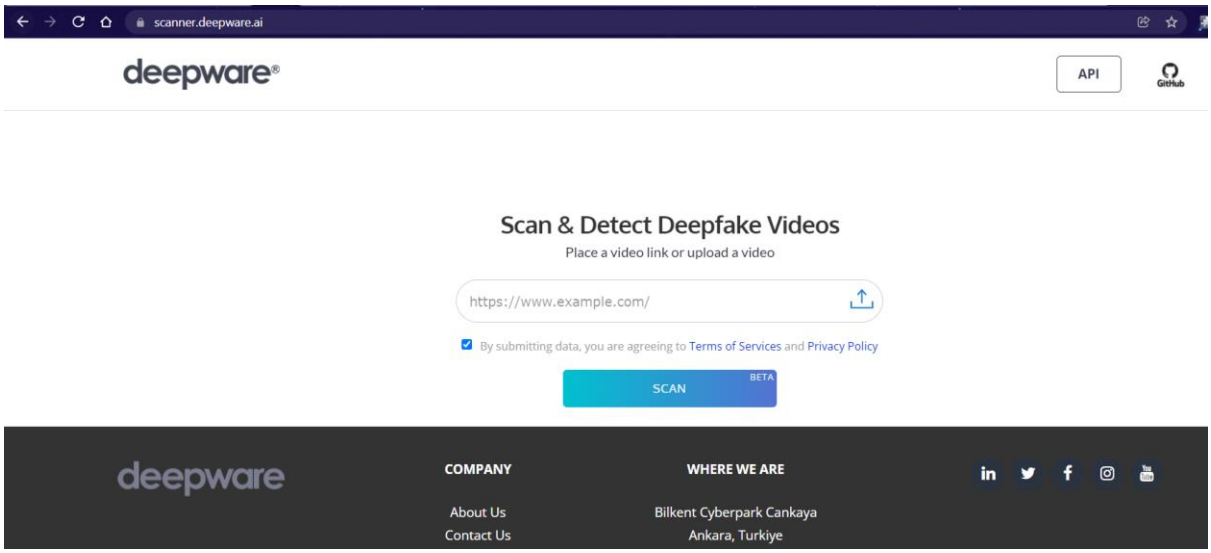
curl --location --request GET 'https://api.deepware.ai/api/v1/url/scan?video-url=https://youtu.be/RQe_l0Zmzuk' \
--header 'X-Deepware-Authentication: 605c27e4-2aaa-44fc-9353-81ce20e17676' \
--header 'Cookie: __cfduid=d1bce04f30f7f2f2be95fe4abd5dab3a51620058884'

Warmest regards,

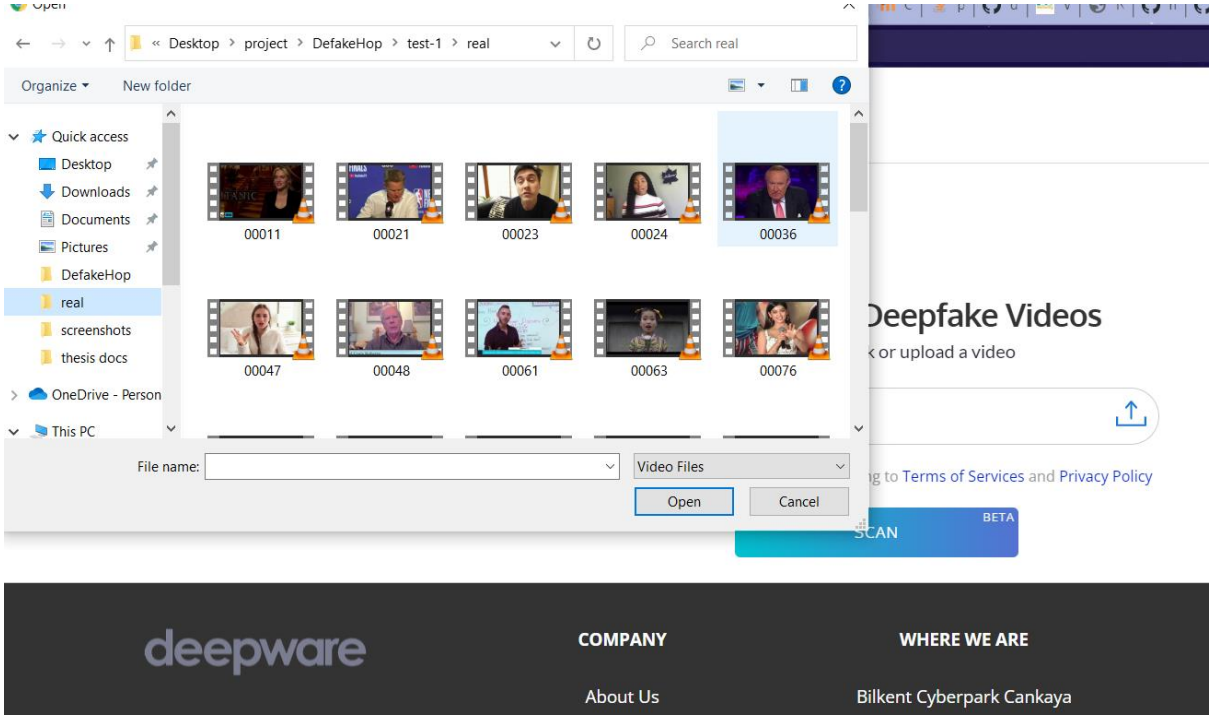
Armagan Tugsal
Customer Care Executive

www.zemana.com

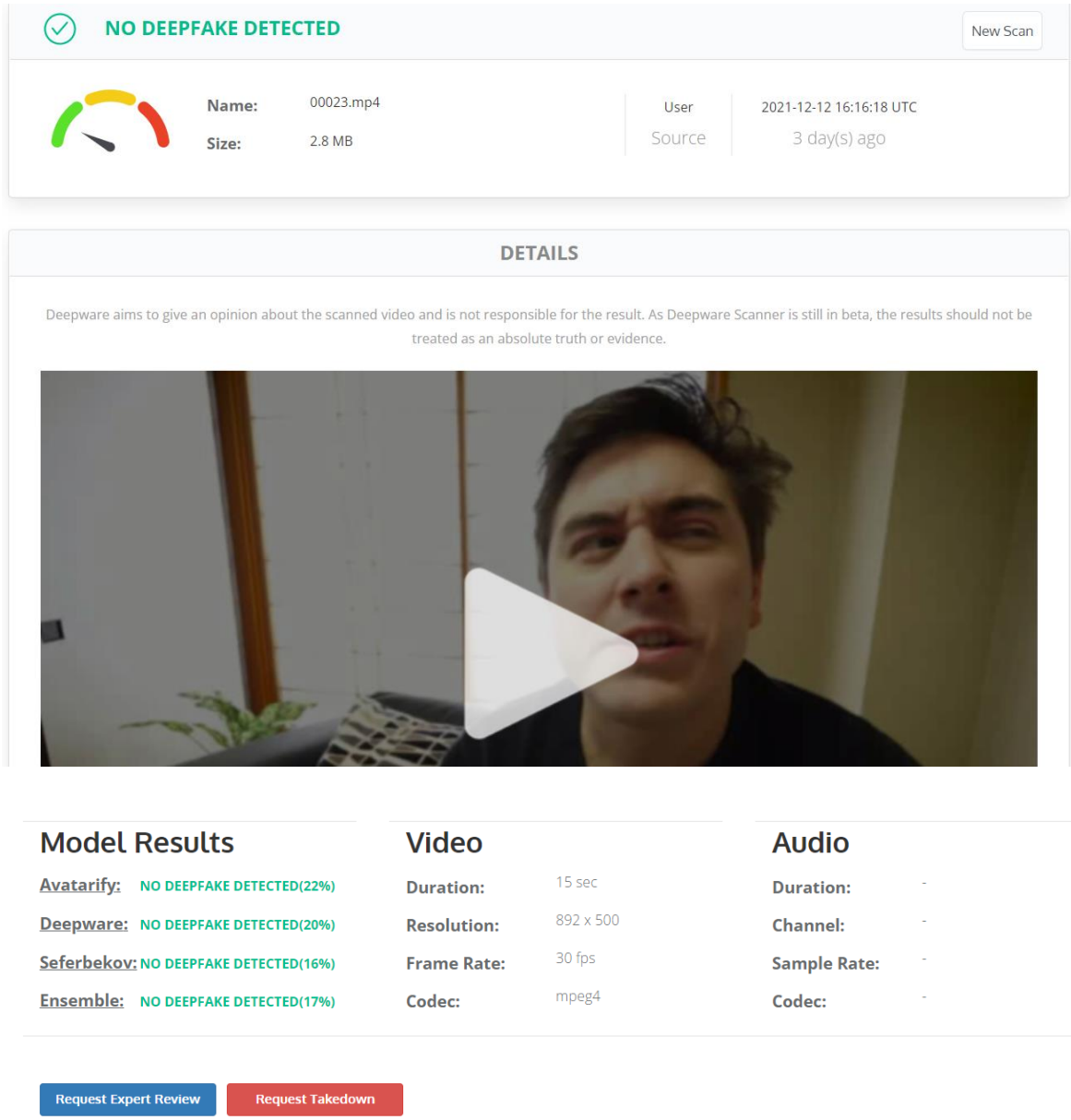
They gave access key for the 100 videos through and there is another way to find the get check deepfakes of video.
Deepware having the portal which can upload and check.
Link: https://scanner.deepware.ai/



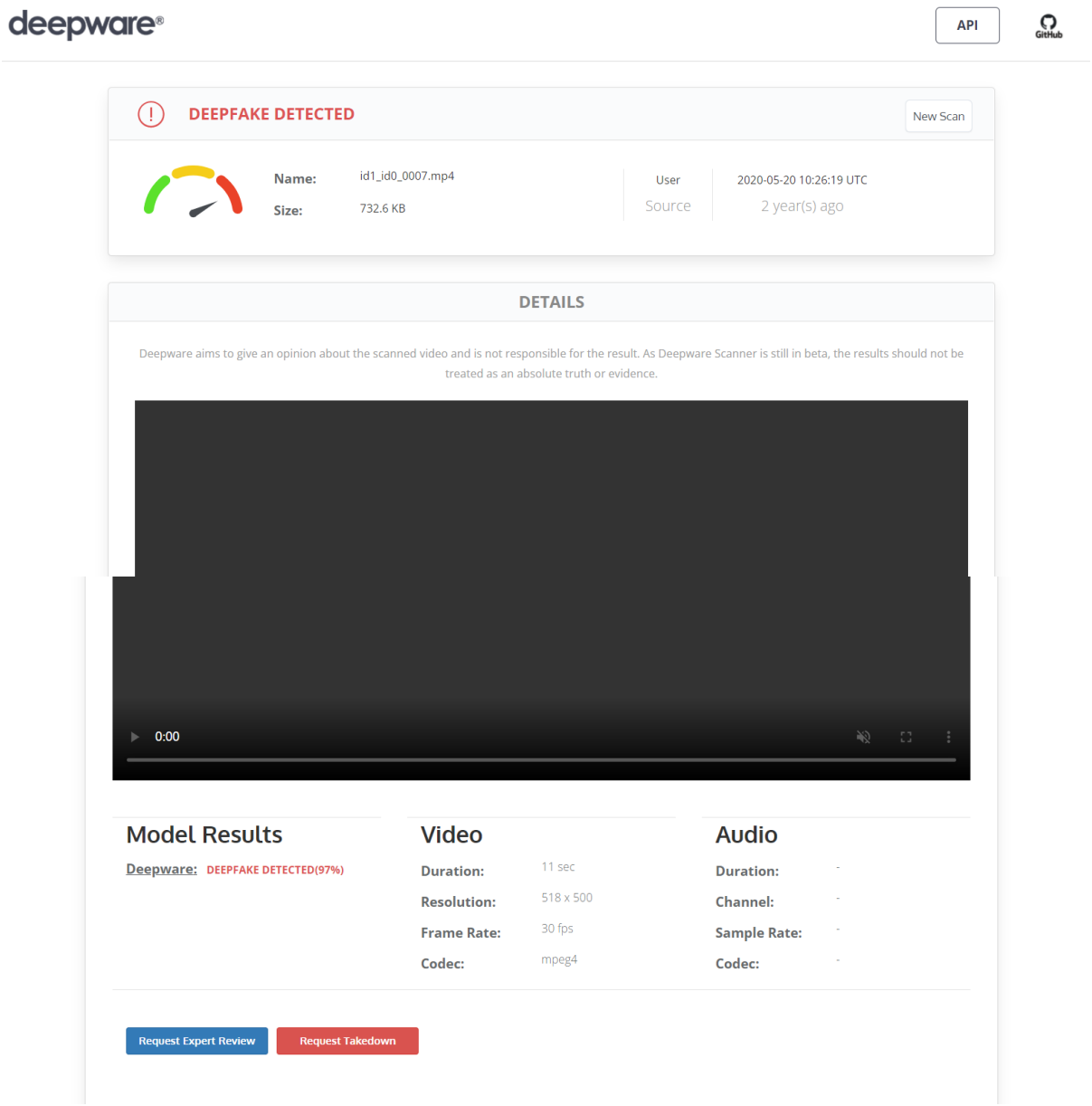
This is the website of deepware.



In this I have been uploading the video to scan



In the top screen can observe the details of video and where deepfake is detected or not. And in the model result can see 4 different reuslts. This results are for real video.



The above screenshot can show that deepfake is detected. In model result can see the results.

3 WeVerify

This is the another online scanner that can give us the video result. But in this there is no portal like deepware scanner so I have mailed them all the test videos so they have provided me the results of the videos with the probablity .

---------- Forwarded message ---------
From: **Spiros Baxevanakis** <spirosbax@iti.gr>
Date: Mon 13 Dec 2021 at 9:01 a.m.
Subject: Re: Requesting benchmarking results
To: <gdevesan@gmail.com>
Cc: Symeon (Akis) Papadopoulos <papadop@iti.gr>

Dear Devesan,

I'm a colleague of Symeon and the lead developer of our DeepFake Detection pipeline. We evaluated our service against the videos you provided. Please find attached a CSV file with the prediction score for each video. Overall, our system achieved an accuracy of 75% , an ROC-AUC of 85.18%, an F1 score of 80.31% and a log-loss of 0.4517 .

Kind Regards,
Spiros (Spyridon) Baxevanakis,
Researcher,
Information Technologies Institute,
Centre for Research and Technology Hellas,
spirosbax@iti.gr

In this mail they also mentioned the overall accuracy , AUC score , f1 score and log -loss.
In deepfake I have changed few lines of code generate and produce the probabilities of each testing videos. After producing the results from the state of art and other two scanners and I made a excel .

	video id	Category	Gender	TrueLabel	DeepwareProb	DeepwareLabel	DeepwareCorrec	WeverifyProb	WeverifyLabel	WeverifyCorri	DeFakeHopPr(	DeFakeHopL	DeFakeHopCorrect
8	00048.mp4	youtube-real	male	1	29	1	1	58	0	0	10.88325278	1	1
9	00061.mp4	youtube-real	male	1	4	1	1	56	0	0	5.916283363	1	1
13	00092.mp4	youtube-real	male	1	1	1	1	10	1	1	3.122856766	1	1
14	00095.mp4	youtube-real	male	1	16	1	1	6	1	1	0.773385328	1	1
15	00106.mp4	youtube-real	male	1	18	1	1	28	1	1	18.06858672	1	1
16	00119.mp4	youtube-real	male	1	34	1	1	29	1	1	28.13626027	1	1
17	00133.mp4	youtube-real	male	1	16	1	1	56	0	0	4.910687077	1	1
18	00138.mp4	youtube-real	male	1	2	1	1	26	1	1	60.8880346	0	0
19	00168.mp4	youtube-real	male	1	4	1	1	40	1	1	30.28787011	1	1
20	00170.mp4	youtube-real	male	1	24	1	1	73	0	0	1.292474452	1	1
23	00194.mp4	youtube-real	male	1	1	1	1	23	1	1	0.806297102	1	1
27	00208.mp4	youtube-real	male	1	1	1	1	2	1	1	35.99185413	1	1
28	00213.mp4	youtube-real	male	1	0	1	1	34	1	1	34.73326173	1	1
30	00236.mp4	youtube-real	male	1	0	1	1	43	1	1	30.02521416	1	1
32	id1_0007.mp4	celeb-real	male	1	4	1	1	13	1	1	80.9775672	0	0
33	id1_id0_0007.mp4	celeb-fake	male	0	97	0	1	67	0	1	97.54257107	0	1
34	id1_id16_0007.mp4	celeb-fake	male	0	98	0	1	64	0	1	98.61787868	0	1
35	id1_id17_0007.mp4	celeb-fake	male	0	98	0	1	73	0	1	99.38028395	0	1
36	id1_id2_0007.mp4	celeb-fake	male	0	98	0	1	99	0	1	99.12077427	0	1
37	id1_id3_0007.mp4	celeb-fake	male	0	98	0	1	97	0	1	99.1450969	0	1
38	id1_id4_0007.mp4	celeb-fake	male	0	98	0	1	83	0	1	96.41936088	0	1
39	id1_id6_0007.mp4	celeb-fake	male	0	98	0	1	88	0	1	98.39176929	0	1
40	id1_id9_0007.mp4	celeb-fake	male	0	98	0	1	91	0	1	99.51299417	0	1
51	id16_0011.mp4	celeb-real	male	1	49	1	1	72	0	0	79.2913008	0	0
52	id16_id0_0011.mp4	celeb-fake	male	0	96	0	1	89	0	1	95.43869626	0	1
53	id16_id1_0011.mp4	celeb-fake	male	0	50	0	1	99	0	1	94.25810096	0	1
54	id16_id17_0011.mp4	celeb-fake	male	0	87	0	1	96	0	1	98.73892374	0	1

This is the final excel sheet where I populated whole results and prodced accuracy on different types of subets like male vs female , celeb-real vs celeb-fake vs youtube real.

Along with this there is another code snippet which is used to calculate the other metrics. To verify that taken the results of the WeVerify as base. In that code snippet calculated the AUC, f1 scores and accuracies of the defakeHop and other two online scanners.

References

1. H. -S. Chen, M. Rouhsedaghat, H. Ghani, S. Hu, S. You and C. -C. Jay Kuo, "DefakeHop: A Light-Weight High-Performance Deepfake Detector," 2021 IEEE International Conference on Multimedia and Expo (ICME), 2021, pp. 1-6, doi: 10.1109/ICME51207.2021.9428361.

2.T. Baltrušaitis, P. Robinson and L. Morency, "OpenFace: An open source facial behavior analysis toolkit," *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1-10, doi: 10.1109/WACV.2016.7477553.

3."Weverify" , https://weverify.eu/tools/deepfake-detector/

4. "Deepware", https://scanner.deepware.ai/