localhost

jupyter Deepfake_detection Last Checkpoint: last year

File Edit View Run Kernel Settings Help

Code

Trusted

JupyterLab Python 3 (ipykernel)

Import Libraries

[1]:
```
import gradio as gr
import torch
import torch.nn.functional as F
from facenet_pytorch import MTCNN, InceptionResnetV1
import numpy as np
from PIL import Image
import cv2
from pytorch_grad_cam import GradCAM
from pytorch_grad_cam.utils.model_targets import ClassifierOutputTarget
from pytorch_grad_cam.utils.image import show_cam_on_image
import warnings
warnings.filterwarnings("ignore")
```

```
C:\kandikits\deepfake-detection\deepfake-detection-env\lib\site-packages\gradio_client\documentation.py:103: UserWarning: Could not get documentation gro
up for <class 'gradio.mix.Parallel'>: No known documentation group for module 'gradio.mix'
  warnings.warn(f"Could not get documentation group for {cls}: {exc}")
C:\kandikits\deepfake-detection\deepfake-detection-env\lib\site-packages\gradio_client\documentation.py:103: UserWarning: Could not get documentation gro
up for <class 'gradio.mix.Series'>: No known documentation group for module 'gradio.mix'
  warnings.warn(f"Could not get documentation group for {cls}: {exc}")
```

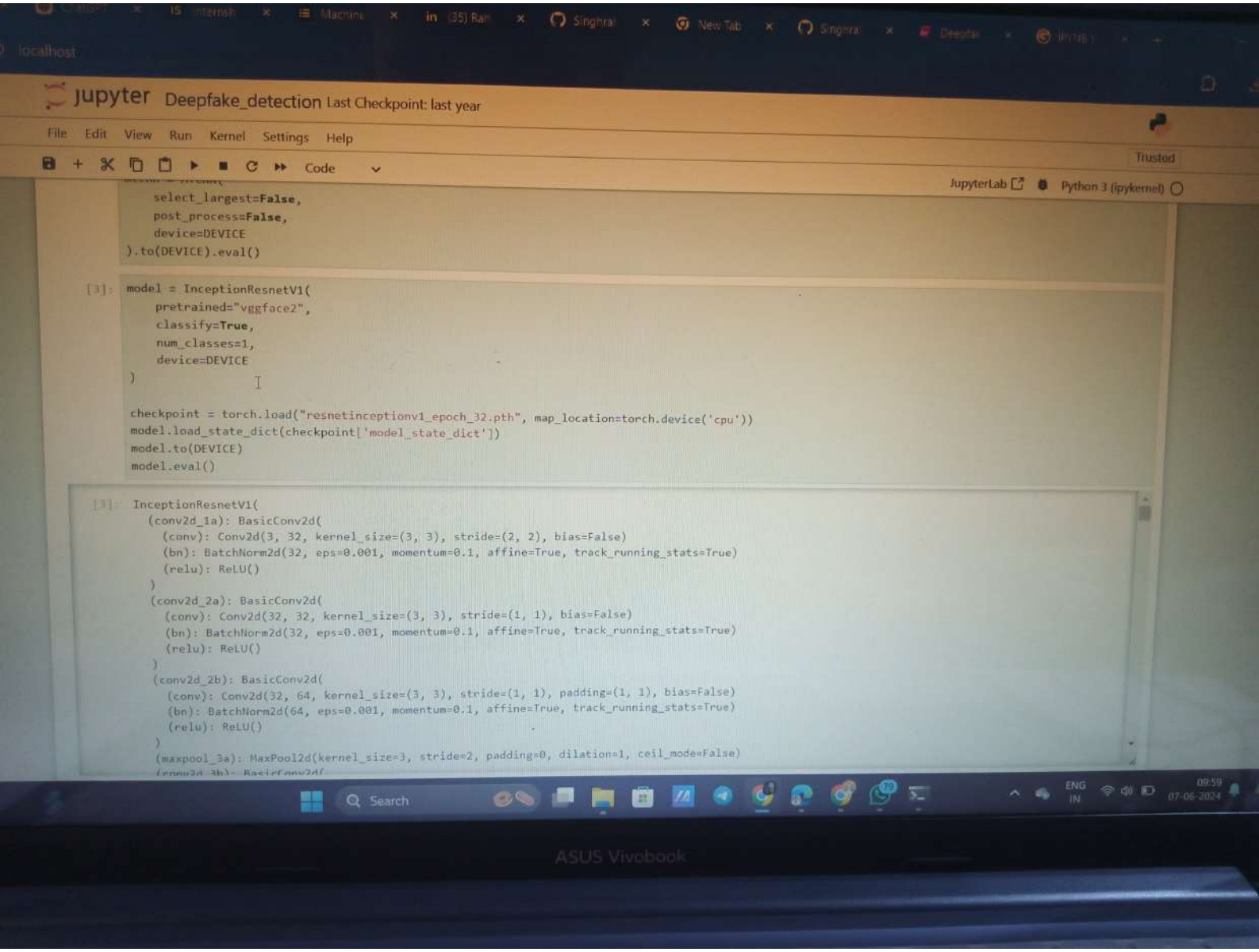
Download and Load Model

[2]:
```
DEVICE = 'cuda:0' if torch.cuda.is_available() else 'cpu'

mtcnn = MTCNN(
    select_largest=False,
    post_process=False,
```

Search

ENG IN 09:59 07-06-2024

ASUS Vivobook


jupyter Deepfake_detection Last Checkpoint: last year
File Edit View Run Kernel Settings Help
Code
Trusted
JupyterLab
Python 3 (ipykernel)
select_largest=False,
post_process=False,
device=DEVICE
).to(DEVICE).eval()
[3]:
model = InceptionResnetV1(
pretrained="vggface2",
classify=True,
num_classes=1,
device=DEVICE
)
checkpoint = torch.load("resnetinceptionv1_epoch_32.pth", map_location=torch.device('cpu'))
model.load_state_dict(checkpoint['model_state_dict'])
model.to(DEVICE)
model.eval()
[3]:
InceptionResnetV1(
(conv2d_1a): BasicConv2d(
(conv): Conv2d(3, 32, kernel_size=(3, 3), stride=(2, 2), bias=False)
(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU()
)
(conv2d_2a): BasicConv2d(
(conv): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), bias=False)
(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU()
)
(conv2d_2b): BasicConv2d(
(conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU()
)
(maxpool_3a): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
Search
ENG
IN
09:59
07-06-2024
ASUS Vivobook

localhost

jupyter Deepfake_detection Last Checkpoint: last year

File Edit View Run Kernel Settings Help

Code

Trusted

JupyterLab Python 3 (ipykernel)

Model Inference

[4]:
```
def predict(input_image:Image.Image):
    """Predict the label of the input_image"""
    face = mtcnn(input_image)
    if face is None:
        raise Exception('No face detected')
    face = face.unsqueeze(0) # add the batch dimension
    face = F.interpolate(face, size=(256, 256), mode='bilinear', align_corners=False)

    # convert the face into a numpy array to be able to plot it
    prev_face = face.squeeze(0).permute(1, 2, 0).cpu().detach().int().numpy()
    prev_face = prev_face.astype('uint8')

    face = face.to(DEVICE)
    face = face.to(torch.float32)
    face = face / 255.0
    face_image_to_plot = face.squeeze(0).permute(1, 2, 0).cpu().detach().int().numpy()

    target_layers=[model.block8.branch1[-1]]
    use_cuda = True if torch.cuda.is_available() else False
    cam = GradCAM(model=model, target_layers=target_layers, use_cuda=use_cuda)
    targets = [ClassifierOutputTarget(0)]

    grayscale_cam = cam(input_tensor=face, targets=targets, eigen_smooth=True)
    grayscale_cam = grayscale_cam[0, :]
    visualization = show_cam_on_image(face_image_to_plot, grayscale_cam, use_rgb=True)
    face_with_mask = cv2.addWeighted(prev_face, 1, visualization, 0.5, 0)

    with torch.no_grad():
        output = torch.sigmoid(model(face).squeeze(0))
        prediction = "real" if output.item() < 0.5 else "fake"
```

Search

ENG IN 09:59 07-06-2024

ASUS Vivobook

localhost

jupyter Deepfake_detection Last Checkpoint: last year

File Edit View Run Kernel Settings Help

Code

Trusted

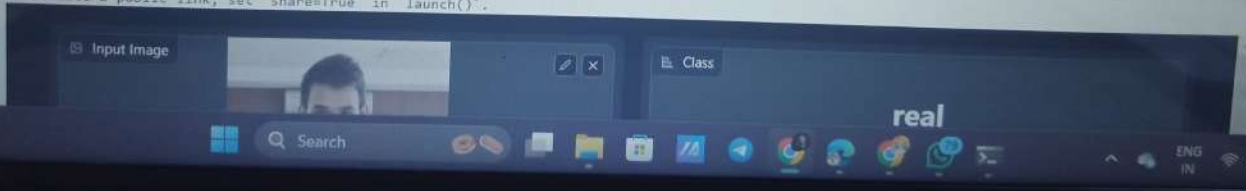
JupyterLab Python 3 (ipykernel)

```
        real_prediction = 1 - output.item()
        fake_prediction = output.item()

        confidences = {
            'real': real_prediction,
            'fake': fake_prediction
        }
    return confidences, face_with_mask
```

Gradio Interface ¶

[5]:
```
interface = gr.Interface(
    fn=predict,
    inputs=[
        gr.inputs.Image(label="Input Image", type="pil")
    ],
    outputs=[
        gr.outputs.Label(label="Class"),
        gr.outputs.Image(label="Face with Explainability", type="pil")
    ],
).launch()
```

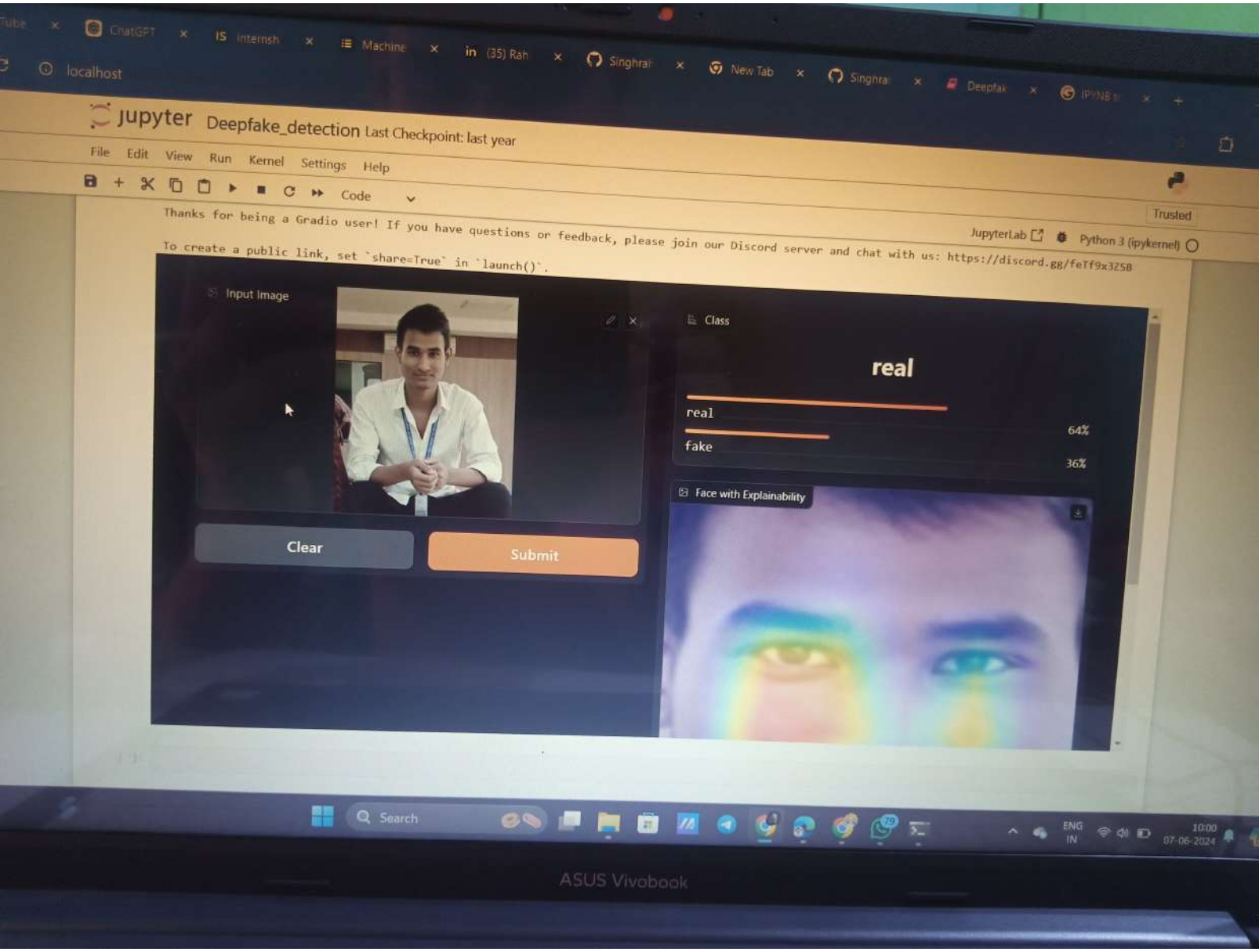
```
Running on local URL:  http://127.0.0.1:7860

Thanks for being a Gradio user! If you have questions or feedback, please join our Discord server and chat with us: https://discord.gg/feTf9x3ZSB

To create a public link, set `share=True` in `launch()`.
```



Search

ENG IN

ASUS Vivobook


localhost
jupyter Deepfake_detection Last Checkpoint: last year
File Edit View Run Kernel Settings Help
Code
Trusted
JupyterLab
Python 3 (ipykernel)
Thanks for being a Gradio user! If you have questions or feedback, please join our Discord server and chat with us: https://discord.gg/feTf9x3ZSB
To create a public link, set `share=True` in `launch()`.
Input Image
Clear
Submit
Class
real
real
64%
fake
36%
Face with Explainability
Search
ENG
IN
10:09
07-06-2024
ASUS Vivobook