**Import Libraries**

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")

# Download and Load Model

DEVICE = 'cuda:0' if torch.cuda.is\_available() else 'cpu'

mtcnn = MTCNN(

select\_largest=False,

post\_process=False,

device=DEVICE

).to(DEVICE).eval()

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()

**#output cell**

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)

(conv2d\_3b): BasicConv2d(

(conv): Conv2d(64, 80, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(80, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(conv2d\_4a): BasicConv2d(

(conv): Conv2d(80, 192, kernel\_size=(3, 3), stride=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(conv2d\_4b): BasicConv2d(

(conv): Conv2d(192, 256, kernel\_size=(3, 3), stride=(2, 2), bias=False)

(bn): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(repeat\_1): Sequential(

(0): Block35(

(branch0): BasicConv2d(

(conv): Conv2d(256, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(256, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(32, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(branch2): Sequential(

(0): BasicConv2d(

(conv): Conv2d(256, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(32, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(32, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(96, 256, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

(1): Block35(

(branch0): BasicConv2d(

(conv): Conv2d(256, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(256, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(32, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(branch2): Sequential(

(0): BasicConv2d(

(conv): Conv2d(256, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(32, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(32, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(96, 256, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

(2): Block35(

(branch0): BasicConv2d(

(conv): Conv2d(256, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(256, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(32, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(branch2): Sequential(

(0): BasicConv2d(

(conv): Conv2d(256, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(32, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(32, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(96, 256, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

(3): Block35(

(branch0): BasicConv2d(

(conv): Conv2d(256, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(256, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(32, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(branch2): Sequential(

(0): BasicConv2d(

(conv): Conv2d(256, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(32, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(32, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(96, 256, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

(4): Block35(

(branch0): BasicConv2d(

(conv): Conv2d(256, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(256, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(32, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(branch2): Sequential(

(0): BasicConv2d(

(conv): Conv2d(256, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(32, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(32, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(32, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(96, 256, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

)

(mixed\_6a): Mixed\_6a(

(branch0): BasicConv2d(

(conv): Conv2d(256, 384, kernel\_size=(3, 3), stride=(2, 2), bias=False)

(bn): BatchNorm2d(384, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(256, 192, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(192, 192, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(192, 256, kernel\_size=(3, 3), stride=(2, 2), bias=False)

(bn): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(branch2): MaxPool2d(kernel\_size=3, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(repeat\_2): Sequential(

(0): Block17(

(branch0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(1, 7), stride=(1, 1), padding=(0, 3), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(7, 1), stride=(1, 1), padding=(3, 0), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(256, 896, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

(1): Block17(

(branch0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(1, 7), stride=(1, 1), padding=(0, 3), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(7, 1), stride=(1, 1), padding=(3, 0), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(256, 896, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

(2): Block17(

(branch0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(1, 7), stride=(1, 1), padding=(0, 3), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(7, 1), stride=(1, 1), padding=(3, 0), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(256, 896, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

(3): Block17(

(branch0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(1, 7), stride=(1, 1), padding=(0, 3), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(7, 1), stride=(1, 1), padding=(3, 0), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(256, 896, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

(4): Block17(

(branch0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(1, 7), stride=(1, 1), padding=(0, 3), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(7, 1), stride=(1, 1), padding=(3, 0), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(256, 896, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

(5): Block17(

(branch0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(1, 7), stride=(1, 1), padding=(0, 3), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(7, 1), stride=(1, 1), padding=(3, 0), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(256, 896, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

(6): Block17(

(branch0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(1, 7), stride=(1, 1), padding=(0, 3), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(7, 1), stride=(1, 1), padding=(3, 0), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(256, 896, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

(7): Block17(

(branch0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(1, 7), stride=(1, 1), padding=(0, 3), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(7, 1), stride=(1, 1), padding=(3, 0), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(256, 896, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

(8): Block17(

(branch0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(1, 7), stride=(1, 1), padding=(0, 3), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(7, 1), stride=(1, 1), padding=(3, 0), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(256, 896, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

(9): Block17(

(branch0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(896, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(1, 7), stride=(1, 1), padding=(0, 3), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(128, 128, kernel\_size=(7, 1), stride=(1, 1), padding=(3, 0), bias=False)

(bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(256, 896, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

)

(mixed\_7a): Mixed\_7a(

(branch0): Sequential(

(0): BasicConv2d(

(conv): Conv2d(896, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(256, 384, kernel\_size=(3, 3), stride=(2, 2), bias=False)

(bn): BatchNorm2d(384, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(896, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(256, 256, kernel\_size=(3, 3), stride=(2, 2), bias=False)

(bn): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(branch2): Sequential(

(0): BasicConv2d(

(conv): Conv2d(896, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(256, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(bn): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(256, 256, kernel\_size=(3, 3), stride=(2, 2), bias=False)

(bn): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(branch3): MaxPool2d(kernel\_size=3, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(repeat\_3): Sequential(

(0): Block8(

(branch0): BasicConv2d(

(conv): Conv2d(1792, 192, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(1792, 192, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(192, 192, kernel\_size=(1, 3), stride=(1, 1), padding=(0, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(192, 192, kernel\_size=(3, 1), stride=(1, 1), padding=(1, 0), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(384, 1792, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

(1): Block8(

(branch0): BasicConv2d(

(conv): Conv2d(1792, 192, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(1792, 192, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(192, 192, kernel\_size=(1, 3), stride=(1, 1), padding=(0, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(192, 192, kernel\_size=(3, 1), stride=(1, 1), padding=(1, 0), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(384, 1792, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

(2): Block8(

(branch0): BasicConv2d(

(conv): Conv2d(1792, 192, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(1792, 192, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(192, 192, kernel\_size=(1, 3), stride=(1, 1), padding=(0, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(192, 192, kernel\_size=(3, 1), stride=(1, 1), padding=(1, 0), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(384, 1792, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

(3): Block8(

(branch0): BasicConv2d(

(conv): Conv2d(1792, 192, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(1792, 192, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(192, 192, kernel\_size=(1, 3), stride=(1, 1), padding=(0, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(192, 192, kernel\_size=(3, 1), stride=(1, 1), padding=(1, 0), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(384, 1792, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

(4): Block8(

(branch0): BasicConv2d(

(conv): Conv2d(1792, 192, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(1792, 192, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(192, 192, kernel\_size=(1, 3), stride=(1, 1), padding=(0, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(192, 192, kernel\_size=(3, 1), stride=(1, 1), padding=(1, 0), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(384, 1792, kernel\_size=(1, 1), stride=(1, 1))

(relu): ReLU()

)

)

(block8): Block8(

(branch0): BasicConv2d(

(conv): Conv2d(1792, 192, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(branch1): Sequential(

(0): BasicConv2d(

(conv): Conv2d(1792, 192, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(1): BasicConv2d(

(conv): Conv2d(192, 192, kernel\_size=(1, 3), stride=(1, 1), padding=(0, 1), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

(2): BasicConv2d(

(conv): Conv2d(192, 192, kernel\_size=(3, 1), stride=(1, 1), padding=(1, 0), bias=False)

(bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(relu): ReLU()

)

)

(conv2d): Conv2d(384, 1792, kernel\_size=(1, 1), stride=(1, 1))

)

(avgpool\_1a): AdaptiveAvgPool2d(output\_size=1)

(dropout): Dropout(p=0.6, inplace=False)

(last\_linear): Linear(in\_features=1792, out\_features=512, bias=False)

(last\_bn): BatchNorm1d(512, eps=0.001, momentum=0.1, affine=True, track\_running\_stats=True)

(logits): Linear(in\_features=512, out\_features=1, bias=True)

)

# Model Inference

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"

real\_prediction = 1 - output.item()

fake\_prediction = output.item()

confidences = {

'real': real\_prediction,

'fake': fake\_prediction

}

return confidences, face\_with\_mask

# Gradio Interface

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()

**#output cell**

Running on local URL: [http://127.0.0.1:7860](http://127.0.0.1:7860/)

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

