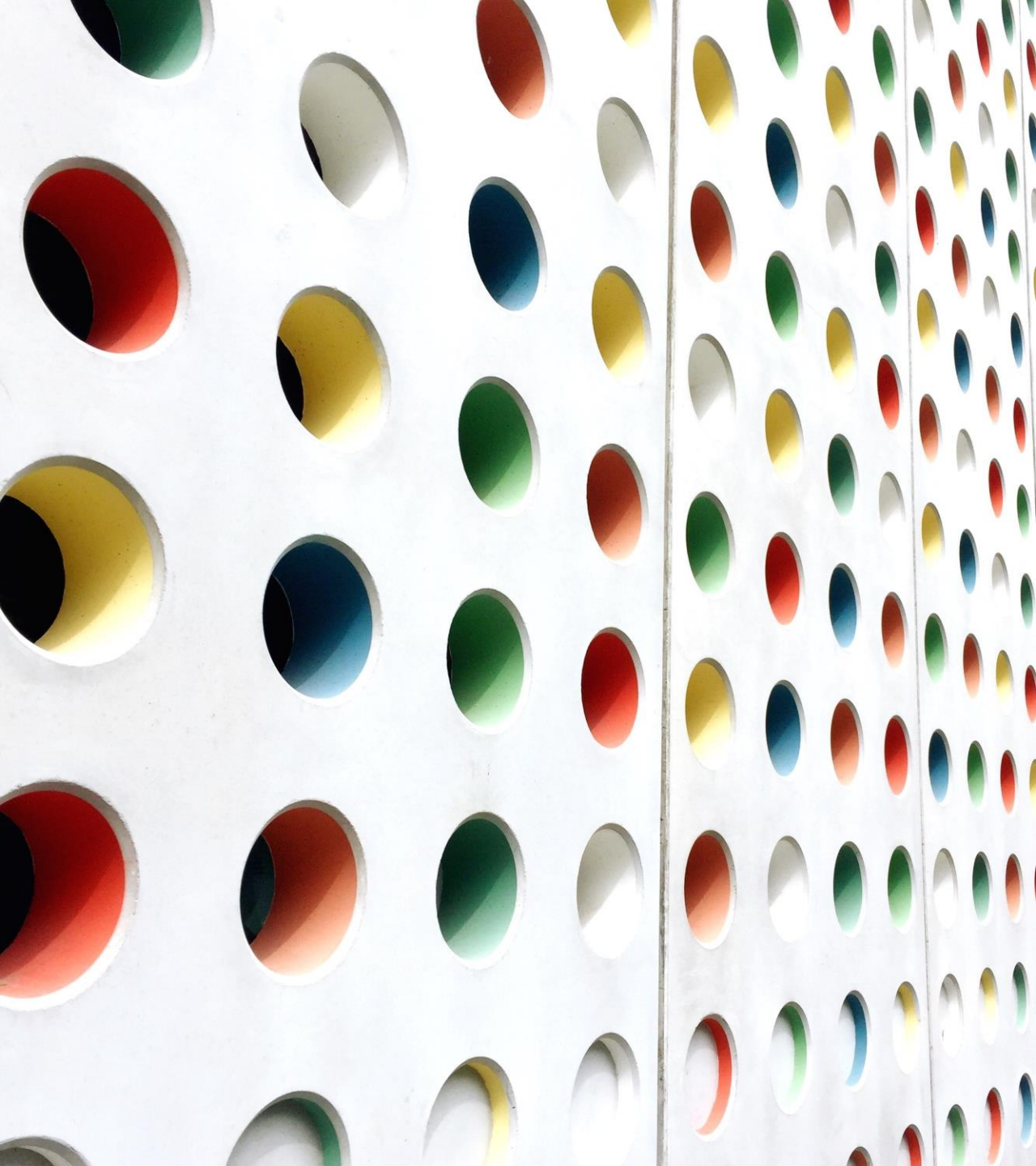




# AI- Generated Images Detector

---

Faculty :- Mrs.R.Syed Ali Fathima



## TEAM MEMBER

Rahul kumar-99220042041  
Deepak kumar -99220042123  
Amit raj-99220042023  
Sathwik-99220041883

---

# Introduction:

---

This project focuses on addressing the specific challenge of detecting and classifying real and fake faces within digital images. By leveraging state-of-the-art deep learning techniques, we aim to develop a robust system capable of accurately distinguishing between genuine and manipulated faces.

The motivation behind this project stems from the need for reliable tools and methods to combat the spread of misinformation and deepfakes. By developing an effective face detection and classification system, we hope to contribute to the efforts aimed at preserving the integrity and trustworthiness of digital media platforms.

# Abstract:

---

The advancement of deep learning techniques has led to significant progress in the field of computer vision, particularly in tasks such as face detection, recognition, and classification. In this project, we explore the application of deep learning models for detecting and classifying real and fake faces within digital images.

Our approach involves leveraging two key deep learning architectures: the Multi-task Cascaded Convolutional Networks (MTCNN) for face detection and the InceptionResnetV1 model pretrained on the VGGFace2 dataset for face recognition.

These models are integrated into an interactive system using the Gradio library, allowing users to input images and receive predictions regarding the authenticity of detected faces.

# Problem Statement:

---

In today's digital landscape, the proliferation of manipulated images and videos poses significant challenges for online content moderation, journalism, and cybersecurity. The rise of deepfake technology, fueled by advances in artificial intelligence and machine learning, has made it increasingly difficult to discern between authentic and manipulated media.

One of the key challenges in this domain is the detection and classification of real and fake faces within digital images and videos.

Detecting these manipulated faces requires sophisticated algorithms capable of identifying subtle visual cues and inconsistencies that may indicate digital tampering. Furthermore, accurately classifying these faces as either real or fake is crucial for mitigating the spread of misinformation and maintaining trust in digital media platforms.

# Objectives:

---

## 1. Develop a Face Detection and Classification System:

Description: The primary objective of this project is to develop a robust face detection and classification system capable of accurately distinguishing between real and fake faces within digital images.

Components: The system will incorporate state-of-the-art deep learning models such as MTCNN (Multi-task Cascaded Convolutional Networks) for face detection and InceptionResnetV1 for face recognition.

## 2. Integrate Explainability Methods for Transparency:

Description: In addition to accurate classification, transparency and interpretability are essential aspects of the system. Therefore, the project aims to integrate explainability methods such as Grad-CAM (Gradient-weighted Class Activation Mapping) to provide visual explanations of the model's predictions.

Enhanced Understanding: These explanations will enhance the users' understanding of the model's decision-making process and increase trust in the system's reliability.

# Literature survey:

---

## Face Detection Techniques:

Viola-Jones Algorithm: One of the earliest and widely used techniques for face detection, which uses Haar-like features and cascade classifiers to detect faces in images.

Deep Learning Approaches: Recent advancements in deep learning have led to the development of highly accurate face detection models, such as MTCNN (Multi-task Cascaded Convolutional Networks) and SSD (Single Shot MultiBox Detector), which leverage convolutional neural networks (CNNs) for improved performance.

## Explainability Techniques:

Gradient-based Methods: Techniques such as Gradient-weighted Class Activation Mapping (Grad-CAM) generate visual explanations by highlighting the regions of input images that contribute most to the model's predictions.

# Existing system:

---

## Limitations of Current Approaches:

**Lack of Transparency:** Many existing face detection and classification systems operate as black boxes, providing little to no insight into the factors influencing their decisions.

**Limited Explainability:** Existing systems often lack explainability mechanisms, making it challenging for users to understand why certain predictions are made.

**Vulnerability to Adversarial Attacks:** Current systems may be susceptible to adversarial attacks, where small perturbations to input images can lead to incorrect predictions.

**Scalability Issues:** Some face detection and classification systems may face scalability issues when dealing with large-scale datasets or real-time processing requirements.



# Existing system:

---

## Challenges in Real-world Scenarios:

**Variability in Data:** Real-world scenarios often involve diverse datasets with variations in lighting conditions, poses, facial expressions, and occlusions, posing challenges for accurate face detection and classification.

**Performance Degradation:** Existing systems may experience performance degradation when applied to real-world scenarios with complex backgrounds or noisy images.

**Interpretability Requirements:** In applications where trust and transparency are paramount, the lack of interpretability in existing systems can hinder their adoption and acceptance.

# Proposed system:

---

## System Architecture:

**Overview:** The proposed system aims to address the limitations of existing face detection and classification approaches by integrating state-of-the-art deep learning models and explainability techniques.

## Components:

**The system comprises two main components:** face detection using MTCNN and face classification using InceptionResnetV1, with Grad-CAM for explainability.

# Proposed system:

---

## Face Detection with MTCNN:

**Role:** MTCNN (Multi-task Cascaded Convolutional Networks) is employed for accurate and efficient face detection within input images.

**Functionality:** MTCNN detects faces by first generating candidate regions using a series of convolutional networks and then refining these regions through bounding box regression and non-maximum suppression.

## Face Classification with InceptionResnetV1:

**Role:** InceptionResnetV1, pre-trained on the VGGFace2 dataset, is utilized for face recognition and classification.

**Functionality:** The model extracts high-level features from detected faces and maps them to a feature space for classification into real or fake categories.

# Methodology

---

## 1. Multi-task Cascaded Convolutional Networks (MTCNN) for Face Detection :

Overview: MTCNN is a widely used deep learning architecture for face detection, capable of detecting faces with high accuracy even under challenging conditions such as occlusion and varying lighting conditions.

## 2. InceptionResnetV1 for Face Recognition:

Overview: InceptionResnetV1 is a deep convolutional neural network architecture pretrained on the VGGFace2 dataset, designed for face recognition tasks. It extracts high-level features from facial images and maps them to a feature space for classification.

# Code Demonstration:

---

## Gradio Interface:

**Interactive Testing:** We have integrated our face detection and classification system into an interactive interface using the Gradio library. This interface allows users to upload images containing faces and receive real-time predictions regarding the authenticity of the detected faces.

**User-Friendly Interface:** The Gradio interface provides a user-friendly experience, enabling individuals with varying levels of technical expertise to easily interact with the system.

**Sample Input and Predictions:** Below are screenshots demonstrating the Gradio interface with sample input images and corresponding predictions:

Screenshot 1: Input Image with Detected Faces

Screenshot 2: Predictions for Detected Faces

# code

---

## Import Libraries

```
[14]: 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

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

mtcnn = MTCNN(
    select_largest=False,
    post_process=False,
    device=DEVICE
).to(DEVICE).eval()

[16]: 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()
```

# code

---

```
(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(
```

## Model Inference

```
[17]: 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)
```

# code

---

```
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()
```

Running on local URL: <http://127.0.0.1:7862>

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

---




# Sample Output 2

---

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

Input Image



✎ ✕

Clear

Submit

Class

fake


fake

100%

real

0%

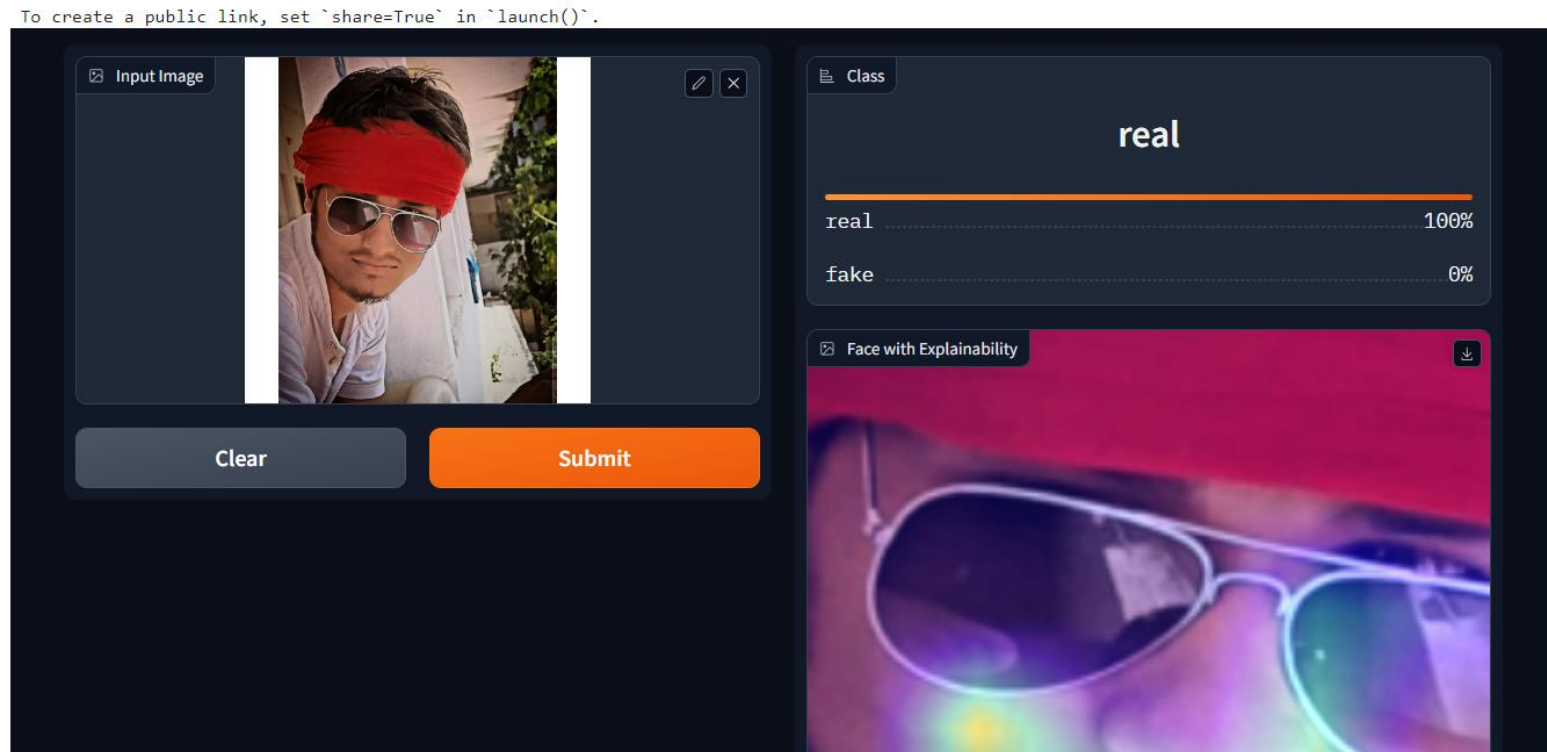
Face with Explainability



⬇

# Sample Output 1

---



# Result:

---

Sample Predictions:

Real Faces: Examples of input images containing genuine faces correctly classified as real by the system, along with their corresponding confidence scores.

Fake Faces: Instances of manipulated or fake faces detected and classified as such by the system, with explanations provided through Grad-CAM visualizations.

Grad-CAM Visualizations:

Interpretability: Grad-CAM heatmaps are generated to provide visual explanations of the model's predictions, highlighting the regions of input images that contribute most to the classification decision.

# Conclusion:

---

## Key Findings:

Our project focused on developing a face detection and classification system capable of accurately distinguishing between real and fake faces within digital images.

Leveraging deep learning models such as MTCNN for face detection, InceptionResnetV1 for face recognition, and Grad-CAM for explainability, we achieved robust performance in detecting and classifying faces with high accuracy.

## Contributions:

Our system provides a valuable tool for combating the spread of misinformation and deepfakes by offering reliable and interpretable solutions for verifying the authenticity of digital media.

The integration of Grad-CAM visualizations enhances the transparency and interpretability of the system, providing valuable insights into the model's decision-making process.