## خطوات بناء مشروع

### ****1. فكرة المشروع****

المشروع يهدف إلى:

* **تحليل المشاعر** من الصور أو الكاميرا مباشرة.
* **التفرقة بين الوجوه الحقيقية والمزيفة** (التي تم إنشاؤها باستخدام الذكاء الاصطناعي).

### ****2. الأدوات المستخدمة****

* **لغة البرمجة:** Python
* **محرر الأكواد :** VS Code
* **المكتبات المستخدمة:**
  + os: لإدارة الملفات والمجلدات.
  + cv2 (OpenCV): لمعالجة الصور والفيديو.
  + numpy: للتعامل مع المصفوفات.
  + seaborn و matplotlib: لرسم الرسوم البيانية.
  + tensorflow.keras: لتحميل الصور وتحويلها إلى مصفوفات.
  + deepface: لتحليل الوجوه والمشاعر.
  + random: لاختيار عينات عشوائية.

### ****3. خطوات بناء المشروع****

#### ****الخطوة 1: تجميع البيانات****

* استخدام **مجموعة بيانات** من موقع **Kaggle**، مثل **Real vs Fake Faces Dataset**

<https://www.kaggle.com/code/martin1234567890/real-vs-fake-face/input>

* البيانات مقسمة إلى:
  + **Real Faces** (صور وجوه حقيقية).
  + **Fake Faces** (صور وجوه مزيفة تم إنشاؤها بالذكاء الاصطناعي).
* تحميل البيانات وفك الضغط عنها بحيث تكون المجلدات كما يلي:

Datas/

├── training\_real/ # صور الوجوه الحقيقية

├── training\_fake/ # صور الوجوه المزيفة

#### ****الخطوة 2: تحضير البيانات****

* **قراءة الصور وتحويلها إلى مصفوفات عددية:**

import os

from tensorflow.keras.preprocessing.image import load\_img, img\_to\_array

data\_dir = r'.\Datas'

real\_data = os.listdir(os.path.join(data\_dir, 'training\_real'))

fake\_data = os.listdir(os.path.join(data\_dir, 'training\_fake'))

X = [] # حفظ الصور

Y = [] # حفظ التسميات (1 للوجوه الحقيقية، 0 للوجوه المزيفة)

# تحميل الصور الحقيقية

for img in real\_data:

img\_path = os.path.join(data\_dir, 'training\_real', img)

img = load\_img(img\_path, target\_size=(170, 170))

X.append(img\_to\_array(img) / 255.0)

Y.append(1)

# تحميل الصور المزيفة

for img in fake\_data:

img\_path = os.path.join(data\_dir, 'training\_fake', img)

img = load\_img(img\_path, target\_size=(170, 170))

X.append(img\_to\_array(img) / 255.0)

Y.append(0)

#### ****الخطوة 3: معالجة البيانات****

* **تطبيع الصور** بحيث تكون القيم بين 0 و 1 لتحسين أداء النموذج.
* **تحويل التسميات إلى One-Hot Encoding**:

from tensorflow.keras.utils import to\_categorical

Y = to\_categorical(Y, 2)

#### ****الخطوة 4: تقسيم البيانات****

* **80% للتدريب، 20% للاختبار**:

from sklearn.model\_selection import train\_test\_split

X\_train, X\_val, Y\_train, Y\_val = train\_test\_split(X, Y, test\_size=0.2, random\_state=5)

#### ****الخطوة 5: عرض عينات من البيانات****

* عرض بعض الصور مع التسمية الخاصة بها:

import matplotlib.pyplot as plt

import random

import numpy as np

num\_samples = 10

random\_indices = random.sample(range(len(X\_val)), num\_samples)

for i in random\_indices:

plt.imshow(X\_val[i])

plt.axis('off')

label = "حقيقي" if np.argmax(Y\_val[i]) == 1 else "مزيف"

plt.title(label, fontsize=14)

plt.show()

#### ****الخطوة 6: تحليل البيانات****

* **رسم توزيع البيانات** بين الصور الحقيقية والمزيفة:

import seaborn as sns

plt.figure(figsize=(8, 6))

sns.countplot(x=[np.argmax(label) for label in Y], palette="coolwarm", edgecolor='black')

plt.title("توزيع الوجوه: حقيقية vs مزيفة")

plt.xlabel("نوع الوجه (0 = مزيف, 1 = حقيقي)")

plt.ylabel("عدد الصور")

plt.show()

#### ****الخطوة 7: تحليل المشاعر باستخدام DeepFace****

* تحليل المشاعر للصور الحقيقية باستخدام **DeepFace**:

from deepface import DeepFace

import cv2

for i in random\_indices:

if np.argmax(Y\_val[i]) == 1: # فقط للوجوه الحقيقية

try:

img = (X\_val[i] \* 255).astype("uint8") # إعادة الصورة إلى القيم الأصلية

img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

analysis = DeepFace.analyze(img, actions=['emotion'], enforce\_detection=False)

emotion = analysis[0]['dominant\_emotion']

except Exception as e:

emotion = "غير معروف"

plt.imshow(X\_val[i])

plt.title(f"المشاعر: {emotion}", fontsize=14, color='blue')

plt.axis('off')

plt.show()

#### ****الخطوة 8: إنشاء واجهة المستخدم البسيطة****

* يمكن للمستخدم اختيار تحليل المشاعر من الكاميرا أو تحليل الصور من المجلد:

print("اختر خيارًا:")

print("1. تحليل المشاعر من الكاميرا")

print("2. تحليل الصور من المجلد")

choice = input("اختيارك (1 أو 2): ")

### ****4. كيفية تشغيل المشروع****

1. **تحميل البيانات** إلى المجلد Datas/.
2. **تثبيت المكتبات المطلوبة**:

pip install opencv-python numpy seaborn matplotlib tensorflow keras deepface

1. **تشغيل الكود في VS Code**.
2. **اختيار الوظيفة**: تحليل المشاعر من الكاميرا أو تحليل الصور من المجلد.

### ****5. شرح الكود بشكل عام****

* **تحميل البيانات** من Kaggle وتنظيمها في مجلدات.
* **تحضير البيانات** عبر تحويل الصور إلى مصفوفات عددية وتطبيع القيم.
* **تحليل البيانات** عبر عرض بعض الصور وتوزيع البيانات.
* **تحليل المشاعر** باستخدام DeepFace.
* **إنشاء واجهة تفاعلية** لاختيار الوظيفة المطلوبة.

### ****6. نصائح لتقديم المشروع****

✅  **ابدأ بشرح الفكرة العامة**: كيف يمكن للذكاء الاصطناعي تحليل الوجوه والمشاعر.  
 ✅ **وضح الأدوات المستخدمة**: Python، VS Code، المكتبات المختلفة.  
✅  **استعرض النتائج بصريًا**: استخدم رسوم بيانية وصور توضيحية.