کد شما با استفاده از کتابخانه‌های **MediaPipe**، **OpenCV**، و **Face Recognition** برای اسکن چهره و ذخیره‌ی انکدینگ‌ها در یک فایل .pkl عمل می‌کند. فرآیند اصلی در دو مرحله‌ی ثبت چهره (Enrollment) و شناسایی چهره (Recognition) انجام می‌شود. در ادامه، جزئیات هر مرحله را توضیح می‌دهم:

**1. ساخت فایل saved\_face.pkl (مرحله ثبت چهره)**

این مرحله با اجرای تابع capture\_and\_store\_encodings انجام می‌شود و فرآیند آن به شرح زیر است:

1. **دسترسی به دوربین:** کد با استفاده از cv2.VideoCapture(0) به دوربین دسترسی پیدا می‌کند.
2. **شناسایی و انکدینگ چهره:**
   * با استفاده از face\_recognition.face\_locations و face\_recognition.face\_encodings، مکان و ویژگی‌های چهره (face encodings) استخراج می‌شود.
   * هر انکدینگ چهره یک بردار عددی است که ویژگی‌های منحصربه‌فرد چهره را توصیف می‌کند. این انکدینگ‌ها توسط الگوریتم‌های یادگیری عمیق از داده‌های آموزشی بدست آمده‌اند.
3. **رسم و تحلیل مش صورت (Face Mesh):**
   * با استفاده از **MediaPipe Face Mesh**، یک شبکه‌ی مش روی صورت کاربر رسم می‌شود که شامل نقاط مشخصی از صورت (مانند چشم‌ها، بینی و دهان) است.
   * این نقاط برای تعیین وضعیت و موقعیت چهره استفاده می‌شوند. به عنوان مثال، کد از نقاط مش برای تشخیص اینکه چهره از زوایای مختلف (مانند چپ، راست، بالا و پایین) اسکن شده است استفاده می‌کند.
4. **ذخیره داده‌ها:**
   * وقتی که کاربر سر خود را به اندازه کافی چرخاند و تعداد زوایای اسکن‌شده کافی بود، انکدینگ‌های ثبت‌شده ذخیره می‌شوند.
   * این انکدینگ‌ها در یک فایل saved\_face.pkl با استفاده از pickle ذخیره می‌شوند. از میان انکدینگ‌های ثبت‌شده، میانگین (average encoding) محاسبه می‌شود تا یک نمای کلی و عمومی از چهره‌ی کاربر به عنوان داده‌ی مرجع ثبت شود.

**2. شناسایی چهره (مرحله Recognition)**

این مرحله با اجرای تابع recognize\_face انجام می‌شود و فرآیند آن به شرح زیر است:

1. **بارگذاری انکدینگ ذخیره‌شده:**
   * فایل saved\_face.pkl بارگذاری شده و انکدینگ ذخیره‌شده به عنوان مرجع (reference encoding) استفاده می‌شود.
2. **شناسایی چهره جدید:**
   * با استفاده از face\_recognition.face\_encodings، انکدینگ‌های چهره‌های شناسایی‌شده در دوربین محاسبه می‌شوند.
   * این انکدینگ‌ها با انکدینگ مرجع مقایسه می‌شوند. فاصله‌ی بین انکدینگ‌ها با استفاده از face\_recognition.face\_distance محاسبه می‌شود:
     + اگر فاصله‌ی بین انکدینگ‌ها کمتر از یک مقدار آستانه (مثلاً 0.3) باشد، چهره شناسایی می‌شود.
     + این مقدار آستانه نشان‌دهنده‌ی میزان شباهت است: هرچه مقدار کوچک‌تر باشد، شباهت بیشتر است.
3. **رسم وضعیت و تطبیق:**
   * کد روی تصویر ورودی وضعیت تطبیق (Matched یا Unknown) را نمایش می‌دهد.
   * اگر چهره شناسایی شود، پیام موفقیت (Matched) روی تصویر نمایش داده می‌شود و کد تا 10 ثانیه پیام را نشان می‌دهد.
4. **کنترل زمان تطبیق:**
   * اگر چهره‌ای شناسایی نشود و بیش از 30 ثانیه گذشته باشد، فرآیند خاتمه می‌یابد.

**نکات کلیدی:**

1. **فایل saved\_face.pkl:**
   * این فایل شامل یک آرایه numpy است که نماینده‌ی انکدینگ میانگین چهره کاربر است. هر انکدینگ به طول 128 عدد است که ویژگی‌های چهره را با استفاده از مدل یادگیری عمیق توصیف می‌کند.
2. **مش صورت (Face Mesh):**
   * استفاده از **MediaPipe Face Mesh** در این کد برای بهبود تجربه‌ی کاربر و کمک به ثبت زوایای مختلف چهره است.
   * هدف از استفاده از مش صورت، هدایت کاربر برای حرکت دادن سر به زوایای مختلف است تا انکدینگ‌های متنوع‌تری ثبت شود.
3. **مقدار آستانه تطبیق:**
   * مقدار 0.3 به عنوان حد شباهت در تابع recognize\_face استفاده می‌شود. این مقدار می‌تواند بر اساس نیاز تغییر کند.

این کد ترکیبی از شناسایی دقیق (انکدینگ چهره) و تعامل با کاربر (با استفاده از MediaPipe) را ارائه می‌دهد که باعث می‌شود فرآیند ثبت و شناسایی چهره کاربر تجربه‌ای تعاملی و کارآمد باشد.