بررسی ، شناخت و پیشبینی اثر انگشت

مقدمه

این طرح ابتدا با سوال «چگونه تابع یک به یک بین اثر انگشت و افراد ایجاد کنیم» شروع شد . برای پاسخ به این سوال نیاز بود تا شکل هر انگشت را تبدیل به دیتای مشخصی کرده و در نمودار نمایش دهیم . اما از آنجایی که اثر انگشت شامل نقش و نگاری خاص و یکتا است و شکل گیری نامعلومی دارد تبدیل آن به یک دیتا با همین خصوصیات در نگاه اول پیچیده بنظر میرسید . در ادامه دو ایده برای انجام اینکار ارائه شده است.

بعد از نمایش نمودار ، ایده پیدا کردن الگو از روی سهمی نمودار بوجود آمد و استفاده از علم ماشین لرنینگ اولین پیشنهاد بود ، بعد از این با چینش دیگر نمودار های مرتبط و یافتن الگو هایی نو کم کم طرح پیشبینی اثر انگشت به چشم رسید

این طرح به گونه ای تعریف شده که ابتدا با گرفتن تصویر اسکن شده اثر انگشت افراد آن را به دیتایی خاص با ویژگی های ذکر شده تبدیل و در یک دیتابیس به همراه تمامی مشخصات ذخیره شود سپس با تمامی مقادیر مرتبط با ایجاد اثر انگشت در زمان تشکیل بررسی و الگوهایی یافت شود .

با کنار هم قرار دادن همه این اطلاعات با گرفتن مشخصات فرد جدید تا حدی درست اثر انگشت آن تحویل داده میشود و بلعکس (با گرفتن اسکن اثر انگشت

بدون داشتن دیتای ثبت شده قبلی مشخصات آن داده میشود) و در تطابق دو اثر انگشت و فرد در صورت وجود اطلاعات قبلی نقش بسزایی دارد .

شروعی برای اثر انگشت :

یک اثر انگشت عبارت است از شیارها و لبه های موجود در نوک انگشتان که منحصر به فرد بوده و یک ویژگی بسیار خوب برای شناسایی افراد به کار میرود .

شکل گیری این لبه ها و شیارها در دوران جنینی و به صورت کاملا اتفاقی و

رندوم و جدای از فاکتورهای وراثتی صورت میگیرد درنتیجه کاملا متفاوت بوده و حتی در انگشتان یک فرد یا دوقلوهای همسان یکسان نیست و تا پایان عمر پایدار بوده و تغییری نمیکند .

در ادامه، به برخی از دسته بندی ها که برای اثرانگشت تعریف شده میپردازیم :

انواع الگو های اثر انگشت :

الگو های حلقه ای (Loop Patterns) :

Loop Pattern

الگوهای حلقه ای در اثر انگشت، شامل خطوطی است که از یک نقطه شروع شده و به دور انگشت حلقه میزنند.

این الگوها میتوانند دو نوع حلقه باز (Open Loop) و حلقه بسته (Closed Loop) را شامل شوند.

الگو های گرد (Whorl Patterns) :


Whorl Pattern

الگوهای گرد در اثر انگشت، شامل خطوطی است که به شکل گرد یا حلقه به دور یک نقطه مرکزی می پیچند.
. این الگوها به عنوان الگوهای گرد (Whorl) شناخته میشوند و میتوانند دارای یک نقطه مرکزی (Central Pocket) یا بدون نقطه مرکزی (Plain Whorl) باشند.

الگو های قوسی(Arch Patterns) :


Arch Pattern

حاوی خطوطی هستند که به شکل مستقیم و بدون پیچش به ارتفاع بالا یا پایین حرکت میکنند. این الگوها به شکل همواره از یک سو به دیگر میپیچند و بدون ایجاد حلقه یا گردی به دور یک نقطه مرکزی میباشند . الگوهای تاشو به دو نوع مهم معکوس (Ulnar Loop) و شعاعی (Radial Loop) تقسیم میشوند که هرکدام خصوصیات خاص خود را دارند.

مفهوم("minutiae"):

در اثر انگشت نگاری، مفهوم "minutia" به جزئیات کوچک و مهمی اشاره دارد که در الگوی اثر انگشت قابل تشخیص هستند. این جزئیات میتوانند شامل نقاط تاشو، حلقه ها ، اتصالات بین خطوط، و نقاط پایانی خطوط باشند.

	Termination
	Bifurcation
	Lake
	Independent ridge
	Point or island
	Spur
	Crossover

روش های تبدیل شکل هندسی اثر انگشت به کد یکتا

برای ساخت تابع یک به یک بین افراد نیاز به شکلی عددی از اثر انگشت خواهیم داشت ، این درصورتی است که اثر انگشت به شکل هندسی ثبت شده است در ادامه روش هایی برای این تبدیل ارائه میشود

۱–روش نسبت گیری(ابداعی):

در این روش سعی میشود با بدست آوردن نسبت پیکسل های سیاه به پیکسل های سفید در تصویر اسکن شده اثر انگشت ، عددی متمایز با دیگر اثر انگشت ها بدست آورد . مراحل برای اجرای این طرح به شرح زیر است :

۱-برای ثبت اثر انگشت ها نیاز به تصویر اسکن شده آن ها داریم

+ سایز تصویر های اسکن شده همگی باید برابر باشند .

+ تصاویر بعد از اسکن باید با فیلتر یکسان سیاه و سفید شوند

اسکن

ثبت اثر انگشت

۲-تصاویر اسکن شده به برنامه داده شده و مقادیر زیر را محاسبه میشود

۱- محاسبه تعداد پیکسل ها

۲- محاسبه تعداد پیکسل مشکی

۳- محاسبه تعداد پیکسل سفید

۴- محاسبه نسبت پیکسل سفید به مشکی

```python
from PIL import Image
from tabulate import tabulate

def count_pixels(image_path):
    # باز کردن تصویر با استفاده از کتابخانه PIL
    image = Image.open(image_path).convert('L')  # تبدیل به تصویر سیاه و سفید
    # ابعاد تصویر
    width, height = image.size
    # تعداد کل پیکسل‌ها
    total_pixels = width * height
    # تعداد پیکسل‌های سیاه
    black_pixels = sum(1 for pixel in image.getdata() if pixel < 128)
    # تعداد پیکسل‌های سفید
    white_pixels = sum(1 for pixel in image.getdata() if pixel >= 128)
    # نسبت پیکسل سفید به سیاه
    white_black = white_pixels / black_pixel
    return {
        'آدرس تصویر': image_path,
        'تعداد کل پیکسل‌ها': total_pixels,
        'تعداد پیکسل‌های سیاه': black_pixels,
        'تعداد پیکسل‌های سفید': white_pixels,
        'نسبت پیکسل سفید به سیاه' : white_black
    }
```

مثال :

ورودی

```
آدرس تصویر: /content/101_2.jpg

تعداد کل پیکسل‌ها: 307200

تعداد پیکسل‌های سیاه: 40559

تعداد پیکسل‌های سفید: 266641

نسبت پیکسل سفید به سیاه: 6.574151236470327
```

خروجی

ایجاد دیتایس برای اثر انگشت ها :

```python
# لیستی برای ذخیره اطلاعات هر عکس
image_info_list = []

while True:
    # آدرس تصویر را وارد کنید
    image_path = input(": (برای خروج عبارت exit را وارد کنید) آدرس تصویر را وارد کنید")

    if image_path.lower() == 'exit':
        break

    # محاسبه اطلاعات تصویر
    image_info = count_pixels(image_path)

    # اضافه کردن اطلاعات به لیست
    image_info_list.append(image_info)

    # نمایش اطلاعات
    print(f"\nاطلاعات تصویر '{image_path}':")
    for key, value in image_info.items():
        print(f"{key}: {value}\n")

# نمایش اطلاعات تمام تصاویر
print("\nاطلاعات تمام تصاویر:")
for index, image_info in enumerate(image_info_list, start=1):
    print(f"تصویر شماره {index}:")
    for key, value in image_info.items():
        print(f"{key}: {value}")
    print("\n")

# نمایش اطلاعات تمام تصاویر در یک جدول دو ستونی
table_headers = ["نسبت پیکسل سفید به سیاه", "آدرس تصویر"]
table_data = [(info['نسبت پیکسل سفید به سیاه'], info['آدرس تصویر']) for info in image_info_list]
table = tabulate(table_data, headers=table_headers, tablefmt="pretty")

# چاپ جدول
print(table)
```

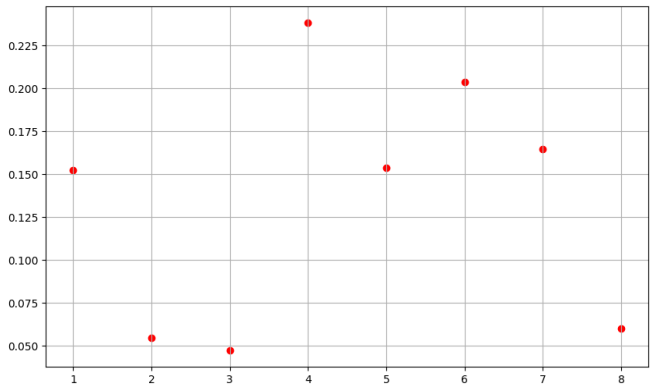
برای نمایش رابطه بین اثر انگشت و نسبت پیکسل ها نیاز به دیتابیس برای ذخیره اطلاعات کلیه اثر انگشت ها داریم

مثال دیتابیس ایجاد شده :

نسبت پیکسل سفید به سیاه	آدرس تصویر
6.574151236470327	/content/101_2.jpg
4.196826417200954	/content/102_2.jpg
6.501648311396547	/content/102_3.jpg
4.910875086585084	/content/102_4.jpg
6.083890605543513	/content/102_5.jpg

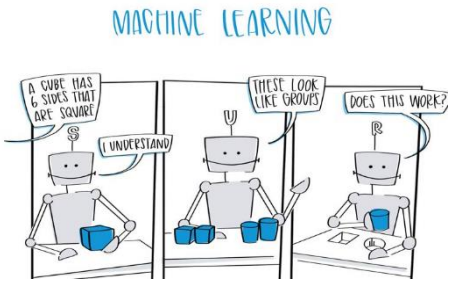
اصل ماجرا : نمودار

در ادامه با استفاده از ماشین لرنینگ سعی در رسیدن به پیشبینی اثرانگشت هستیم



نمودار بین نسبت و اثر انگشت

تعریف : ماشین لرنینگ یک شاخه از هوش مصنوعی است که به الگوریتم ها و مدل هایی اطلاق میشود که از داده ها یاد میگیرند و توانایی پیشبینی و اتخاذ تصمیمات را دارند. در اینجا، از ماشین لرنینگ برای تحلیل نمودار نسبت پیکسل سفید به سیاه به وسیله کد برنامه استفاده شده است. این مدل با استفاده از الگوریتم رگرسیون خطی به داده هایی آموزشی آموزش دیده و سپس عملکرد خود را با داده هایی آزمون ارزیابی کرده است



۱- استخراج ویژگی ها:

در این مرحله، ویژگی های مهم از داده۱ استخراج میشود. در اینجا، شماره تصویر و نسبت پیکسل سفید به سیاه به عنوان ویژگی های ورودی مورد نظر برای ماشین لرنینگ میتوانند مورد استفاده قرار گیرند

۲- آموزش مدل :

از الگوریتم های ماشین لرنینگ مانند رگرسیون خطی یا رگرسیون غیرخطی برای آموزش مدل استفاده میشود در این مرحله، مدل با استفاده از داده‌ای آموزشی آموزش میبیند

```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# اختصاص شماره به هر آدرس تصویر
for i, image_info in enumerate(image_info_list, start=1):
    image_info['شماره تصویر'] = i

# آموزش مدل ماشین لرنینگ
X = [[info['شماره تصویر'], info['نسبت پیکسل سفید به سیاه']] for info in image_info_list]
y = [اثر_انگشت]  # جایگزین کنید
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)

# ارزیابی مدل
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')

# پیشبینی اثر انگشت برای یک نمونه جدید
new_sample = [[شماره_تصویر, نسبت_پیکسل_سفید_به_سیاه]]
predicted_fingerprint = model.predict(new_sample)
print(f'پیشبینی اثر انگشت: {predicted_fingerprint[0]}')
```

۳- ارزیابی مدل :

پس از آموزش مدل، نیاز است که عملکرد مدل با استفاده از داده‌ای آزمون ارزیابی شود. معیارهایی یا خطا (accuracy) مانند دقت میتوانند برای ارزیابی مدل استفاده شوند..

۴- پیشبینی اثر انگشت :

پس از ارزیابی موفقیت مدل، میتوان از آن برای پیشبینی اثر انگشت استفاده کرد. بر اساس شماره تصویر و نسبت پیکسل سفید به سیاه، مدل میتواند پیشبینی کند .که آیا این تصویر به چه انگشتی تعلق دارد

"ادامه دارد ..."

۲–ایجاد ماتریس :

در این روش با دادن مقادیر ۰ و ۱ برای پیکسل های سیاه و سفید سعی در ایجاد ماتریس و تبدیل آن به کد یکتا داریم ، مراحل اجرای این طرح به شیوه زیر است :

۱- ثبت تصاویر اسکن شده از اثر انگشت ها

۲- تبدیل تصویر به آرایه در کتابخانه numpy

۳- دادن مقادیر ۰ برای پیکسل سفید و ۱ برای پیکسل سیاه :در تابع where رنگ هر پیکسل از ۰ تا ۲۵۶ شماره گذاری شده و شماره های ۱۲۸ به بالا را سفید (مقدار ۰) و ۰ تا ۱۲۸ را سیاه (مقدار ۱) شناسایی میشود .

```python
from PIL import Image
import numpy as np

def image_to_matrix(image_path):
    # باز کردن تصویر با استفاده از کتابخانه PIL
    image = Image.open(image_path).convert('L')  # تبدیل به تصویر سیاه و سفید

    # تبدیل تصویر به یک آرایه numpy
    image_array = np.array(image)

    # ایجاد یک ماتریس با ابعاد مشابه تصویر و تعیین مقادیر به مقدار 1 برای پیکسل‌های سیاه
    binary_matrix = np.where(image_array < 128, 1, 0)

    return binary_matrix

# آدرس تصویر را وارد کنید
image_path = "p.png"
result_matrix = image_to_matrix(image_path)

# نمایش ماتریس حاصل
print(result_matrix)
```

۴- و در ادامه تبدیل ماتریس به کد ، نمایش و پیدا کردن ضابطه....