



جامعة دمشق

كلية الهندسة المعلوماتية

السنة الخامسة - قسم الذكاء الصناعي

الرؤية الحاسوبية

Gestures Photo Editor

إعداد:

رائد محمد زهير السبيناتي

سدره فراس ميرخان

عبد الرحمن محمد بشار الارمشي

عبد العليم يحيى السيد

إشراف:

م. نور الحكيم

2022-12-21

الفهرس

3	اكتشاف اليد.....
3	اكتشاف الجلد.....
3	طرح الأطر.....
3	إيجاد contours اليد.....
4	تحديد ال Posture
4	تحديد مركز اليد.....
4	إيجاد ال defects.....
4	عد الأصابع.....
5	تحديد الوضعية النهائية.....
7	Class Queue.....
7	الواجهة وعمليات التعديل على الصورة.....
9	ربط ال postures بعمليات التعديل على الصورة.....
10	المصادر.....

اكتشاف اليد

اكتشاف الجلد

تم باستخدام skin histogram العلاقات التالية:

$$\begin{aligned} 137 < Cr < 177; \\ 77 < Cb < 127; \\ 190 < Cb + 0.6 Cr < 215. \end{aligned}$$

- $R > 95$ AND $G > 40$ AND $B > 20$ AND $\max\{R, G, B\} - \min\{R, G, B\} > 15$ AND RGB components must not be close together grayness eliminate

$|R - G| > 15$ AND R and G components must not be close together, otherwise we are not dealing within the fair complexion

$R > G$ AND $R > B$ (i.e., R component must be the greatest component)

ضمن نظام YCbCr

ضمن نظام RGB

ووفقاً لتجاربنا فإن نظام YCbCr أعطى نتائج أفضل.

طرح الأطر

لتمييز اليد عن وجه الشخص فإنه سيطلب من المستخدم أن يقوم بمعايرة الخلفية التي ستظل ثابتة بشكل نسبي بما في ذلك وجهه وجسمه باستثناء يده، وعند الضغط على الحرف Z من لوحة المفاتيح سيتم اتخاذ إطار مرجعي يمثل الخلفية، ولأجل كل إطار جديد يتم طرحه من الإطار المرجعي.

كنتيجة لذلك فإنه في حال ثبات المستخدم والخلفية بشكل نسبي فإن ذلك سيؤدي لتمييز فقط الأجسام المتحركة التي لها لون الجلد؛ والتي ستكون يد المستخدم في حالتنا.

إيجاد contours اليد

باستخدام cv2.findContours سيتم إيجاد جميع ال contours الناتجة عن ال mask المركب من المرحلتين السابقتين؛ ولتحديد ال contour الأكثر احتمالاً لكونه يداً تم الاعتماد على المعايير التالية بشكل تجريبي بما يوافق ال postures المستخدمة لاحقاً:

- نسبة عرض ال contour إلى الطول أكبر من 0.4 وأصغر من 1.7

- مساحة ال contour أكبر من 7000

بعد فلترة ال contours حسب المعايير السابقة يتم اعتبار ال contour الأكبر مساحة هو الذي يمثل يد المستخدم.

تحديد ال Posture

تحديد مركز اليد

وذلك باستخدام cv2.moments والتي تعد سمة أساسية كما سنرى أهميتها لاحقاً.

إيجاد ال defects

والتي ستمثل في حالتنا الفراغات بين الأصابع، تم التحقيق عن طريق المقارنة بين ال contour والمغلف المحدب الخاص به باستخدام cv2.convexityDefects.

عد الأصابع

لأجل كل defect نقوم بحساب مساحة المثلث الذي يشكله باستخدام Heron's Formula:

$$s = \frac{a + b + c}{2}, \text{ Area} = \sqrt{s(s - a)(s - b)(s - c)}$$

يتم اعتبار ال defect يشكل فراغاً بين الأصابع في حال تحقق:

- $\frac{2 \text{ Area}}{a} > 30$ ، حيث a هو الضلع المقابل للفراغ
- الزاوية التي يشكلها الفراغ أصغر أو تساوي 90 درجة

ويمكن اعتبار أن عدد الأصابع يساوي عدد الفراغات بين الأصابع + 1

ولتمييز حالة اليد المغلقة عن حالة اليد ذات الاصبع الواحد فيتم باستخدام العلاقة:

$$\frac{\text{contour area}}{\text{hull area}} > 0.85$$

حيث اليد المغلقة تشغل مساحة كبيرة من المغلف المحدب المحيط بها على العكس اليد ذات الاصبع الواحد.

تحديد الوضعية النهائية

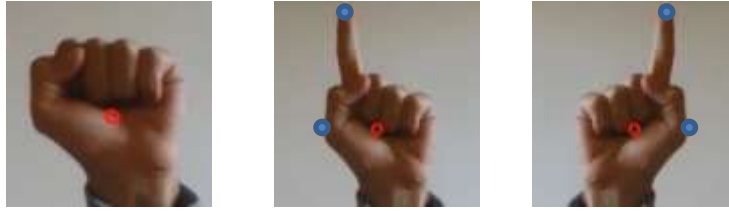
تم وضع قواعد لتحديد 15 وضعية بناء على المعلومات المستخرجة سابقاً بعضها يميز فيما إذا كانت الوضعية منفذة باستخدام اليد اليمنى أو اليسرى وبعضها لا يميز، فيما يلي استعراض لهذه القواعد:

a- في حال عدم وجود أي فراغ:

1. Zero: بناءً على النسبة المذكورة في الفقرة السابقة.

2. ONE_RIGHT: عندما تكون السبابة أقرب إلى يسار اليد من يمينها.

3. ONE_LEFT: عندما تكون السبابة أقرب إلى يمين اليد من يسارها.



b- في حال وجود فراغ واحد:

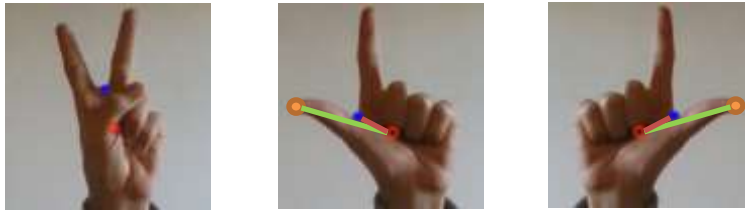
4. TWO_MIDDLE: إذا كان الفراغ له تقريبا نفس قيمة y التي يملكها مركز اليد.

5. TWO_RIGHT: إذا كان الشعاع بين المركز والفراغ يملك \cos similarity موجب مع

الشعاع بين المركز ويسار اليد.

6. TWO_LEFT: إذا كان الشعاع بين المركز والفراغ يملك \cos similarity موجب مع

الشعاع بين المركز ويمين اليد.



c- في حال وجود فراغين:

7. THREE_MIDDLE: في حال كان أحد الفراغين له قيمة x أكبر من x المركز، والفراغ

الآخر له قيمة x أصغر من x المركز.

8. THREE_RIGHT: في حال كان كلا الفراغين له قيمة x أصغر من x المركز.

9. THREE_LEFT: في حال كان كلا الفراغين له قيمة x أكبر من x المركز.



-d في حال وجود ثلاث فراغات:

10. FOUR_RIGHT: في حال كان الضلع الأيسر للمثلث المتشكل من ال defect في

المنتصف أطول من الضلع الأيمن لنفس المثلث.

11. FOUR_LEFT: في حال كان الضلع الأيسر للمثلث المتشكل من ال defect في

المنتصف أقصر من الضلع الأيمن لنفس المثلث.



-e في حال وجود أربع فراغات:

12. FIVE_RIGHT: إذا كان أخفض فراغ يقع على يسار المركز وبقية الفراغات تتوزع بشكل

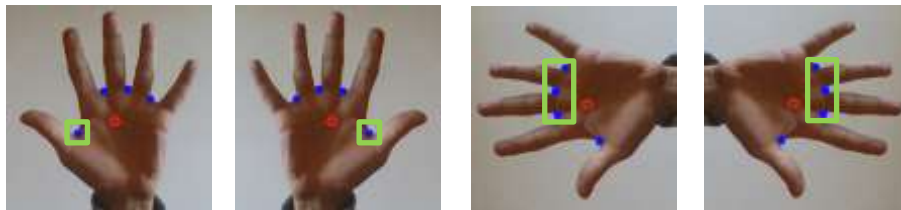
مختلف بين يمين ويسار المركز.

13. FIVE_LEFT: إذا كان أخفض فراغ يقع على يمين المركز وبقية الفراغات تتوزع بشكل

مختلف بين يمين ويسار المركز.

14. FIVE_RIGHT_SIDE: إذا وقع ثلاث فراغات على يسار المركز.

15. FIVE_LEFT_SIDE: إذا وقع ثلاث فراغات على يمين المركز.



Class Queue

يقوم هذا الصف بتمثيل بنية ال queue المعلومة، بالإضافة لتحديد الحجم الأعظمي ، كما يوجد تابع يعيد القيمة الأكثر تكراراً وآخر يعيد أبعد قيمتين (أول وآخر قيمة)

يفيد هذا الصف في :

1- Tracking : عن طريق طرح النقط المتتبعه من اليد نستطيع معرفة جهة الحركة

(up,down,left,right) حيث يوجد تابع

2- في عملية undo : ولذلك لاحتوائه على تابع pop الذي يقوم بإرجاع آخر قيمة مضافة

(top) والتي تحوي الصورة ما قبل التعديل.

3- لزيادة استقرار تحديد posture الملتقط.

Tracking:

بعد التقاط اليد ، يتم تحديد موضعها وعن طريق ملاحقة أعلى نقطة بال contour الخاص باليد ويتم

تحديد جهة حركة اليد بمساعدة ال Queue

الواجهة وعمليات التعديل على الصورة

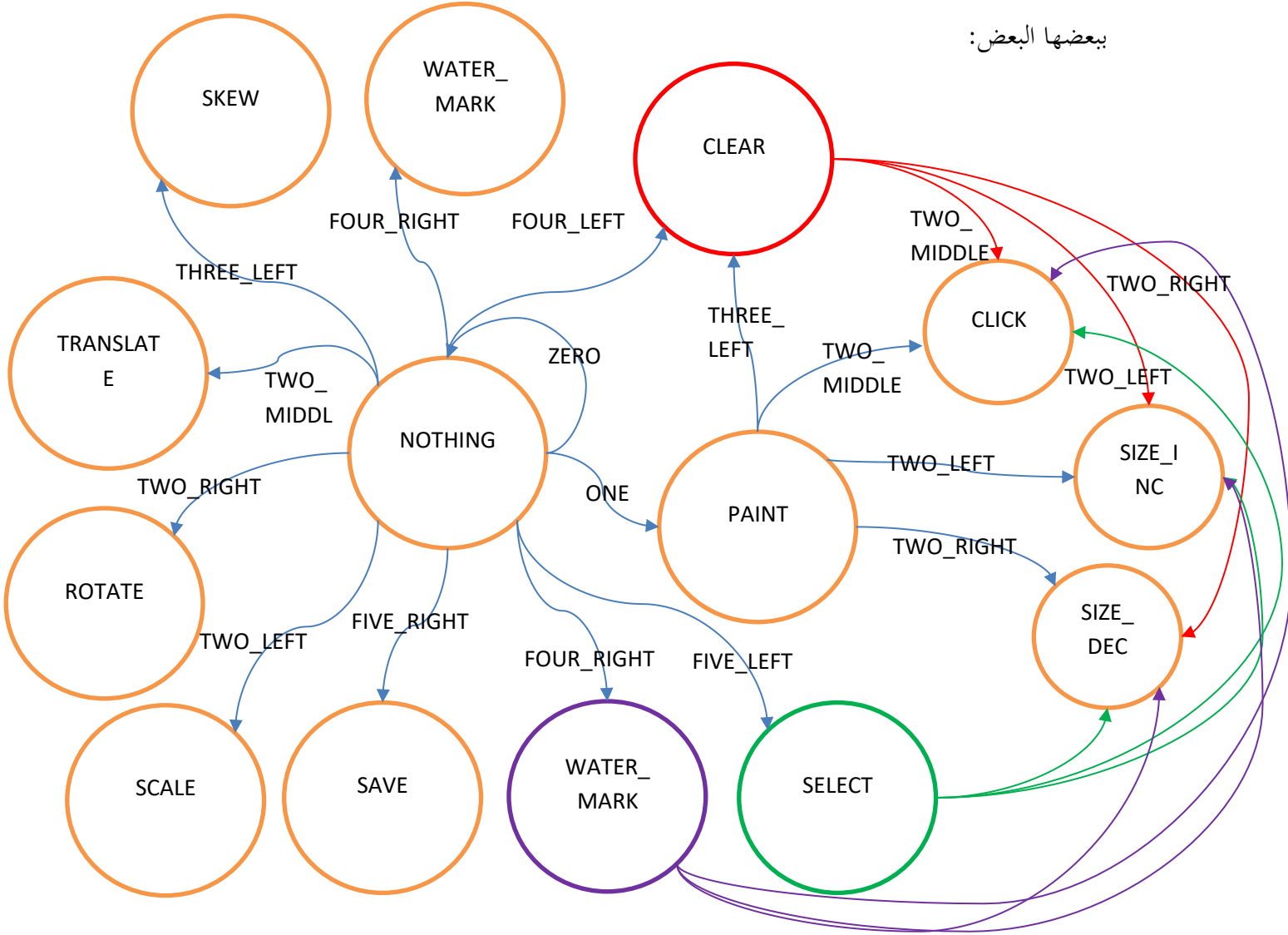
تم استخدام Tkinter لإنشاء واجهة التطبيق، ومكتبة Pillow للقيام بالعمليات على الصورة



2. اعملية Translate وعن طريق Tracking لحركة اليد يتم تحديد الجهة المراد الانسحاب عليها (left, right, top, down).
3. اعملية Rotate وعن طريق Tracking لحركة اليد يتم تحديد جهة الدوران.
4. اعملية Scale وعن طريق Tracking لحركة اليد يتم تحديد العملية تصغير أم تكبير.
5. اعملية Skew وعن طريق Tracking لحركة اليد يتم تحديد الزوايا.
6. إمكانية الرسم بأداة الفرشاة بالإضافة إلى إمكانية مسح الرسم وذلك عن طريق محاكاة حركة الفأرة، وتغيير حجم الفرشاة والممحاة وتغيير لون الفرشاة وذلك عن طريق إيماءات مخصصة لكل عملية.
7. إمكانية إضافة Image Watermark للصورة، وذلك بنفس طريقة اختيار الصورة.
8. إمكانية حفظ الصورة الناتجة.
9. إمكانية التراجع عن أي عملية من العمليات السابقة.

ربط ال postures بعمليات التعديل على الصورة

لتحقيق عملية الربط تم إنشاء class يمثل DFA بحيث كل عملية تمثل node وكل posture يمثل input لل dfa ينقله من حالة إلى أخرى ، في المخطط التالي سيتم توضيح علاقة العمليات والإيماءات ببعضها البعض:



- جميع الحالات تعود إلى NOTHING إذا كان ال posture هو ZERO
- الحالات الثلاث الصغيرة هي حالات آنية فمثلا إذا كنا عند الحالة paint وانتقلنا إلى الحالة CLICK تعود مباشرة إلى PAINT دون انتظار posture جديد من المستخدم .

- [1] G. Mohammed, H. Majeed, Aldujaili, Hassan, Abdul-Jabbar and H. Majeed - IMAGE SEGMENTATION FOR SKIN DETECTION - JOURNAL OF SOUTHWEST JIAOTONG UNIVERSITY - Vol. 55 No. 1 - Feb. 2020