**การแบ่งส่วนภาพกับอัลกอริทึมลุ่มน้ำ**

เป้าหมาย

ในบทนี้,

* เราจะเรียนรู้การใช้การแบ่งส่วนภาพตามลำดับชั้นโดยใช้ขั้นตอนวิธีลุ่มน้ำ
* เราจะเห็น: **cv2.watershed ()**

ทฤษฎี

ภาพระดับสีเทาใด ๆ ที่สามารถมองได้ว่าเป็นพื้นผิวที่มีความเข้มสูงหมายถึงยอดเขาและเนินเขาในขณะที่ความเข้มต่ำจะบ่งบอกถึงหุบเขา คุณเริ่มต้นกรอกทุกหุบเขาที่แยก (minima ท้องถิ่น) ด้วยน้ำสีที่แตกต่างกัน (ป้าย) ขึ้นอยู่กับยอดเขา (gradients) ในบริเวณใกล้เคียงน้ำจากหุบเขาที่แตกต่างกันอย่างชัดแจ้งด้วยสีที่แตกต่างกันจะเริ่มผสาน เพื่อหลีกเลี่ยงปัญหานี้คุณจะสร้างอุปสรรคในบริเวณที่น้ำผสาน คุณยังคงทำงานของการเติมน้ำและสร้างอุปสรรคจนยอดทั้งหมดอยู่ภายใต้น้ำ จากนั้นอุปสรรคที่คุณสร้างขึ้นจะช่วยให้คุณได้ผลการแบ่งส่วน นี่คือ "ปรัชญา" เบื้องหลังลุ่มน้ำ คุณสามารถไปที่[หน้าเว็บ CMM ในลุ่มน้ำ](http://cmm.ensmp.fr/~beucher/wtshed.html)เพื่อทำความเข้าใจกับภาพเคลื่อนไหวบางอย่าง

แต่วิธีนี้ทำให้คุณได้รับผลกระทบจากการรบกวนเนื่องจากมีเสียงรบกวนหรือความผิดปกติอื่น ๆ ในภาพ OpenCV จึงใช้อัลกอริทึมลุ่มน้ำตามเครื่องหมายที่คุณระบุว่าเป็นจุดหุบเขาทั้งหมดที่จะรวมเข้าด้วยกันและที่ไม่ได้อยู่ เป็นการแบ่งส่วนภาพแบบโต้ตอบ สิ่งที่เราทำคือให้ป้ายชื่อต่างๆสำหรับวัตถุที่เรารู้จัก ตั้งชื่อบริเวณที่เรามั่นใจว่าเป็นเบื้องหน้าหรือวัตถุที่มีสีเดียว (หรือความเข้ม) ตั้งชื่อบริเวณที่เรามั่นใจว่าเป็นพื้นหลังหรือไม่ใช่วัตถุด้วยสีอื่นและในที่สุดพื้นที่ที่เราไม่แน่ใจว่ามีอะไร ป้ายชื่อด้วยเครื่องหมาย 0. นั่นคือเครื่องหมายของเรา จากนั้นใช้อัลกอริทึมลุ่มน้ำ เครื่องหมายของเราจะได้รับการอัปเดตด้วยป้ายกำกับที่เราให้และขอบเขตของวัตถุจะมีค่า -1

รหัส

ด้านล่างเราจะเห็นตัวอย่างเกี่ยวกับวิธีใช้ Conversion ระยะทางพร้อมกับลุ่มน้ำเพื่อแบ่งวัตถุที่แตะด้วยกัน

พิจารณาเหรียญภาพด้านล่างเหรียญจะสัมผัสกัน แม้ว่าคุณจะเกณฑ์เกณฑ์ก็จะได้สัมผัสกัน



เราเริ่มต้นด้วยการหาประมาณการโดยประมาณของเหรียญ สำหรับที่เราสามารถใช้ binary การ Otsu ของ

**import** **numpy** **as** **np**

**import** **cv2**

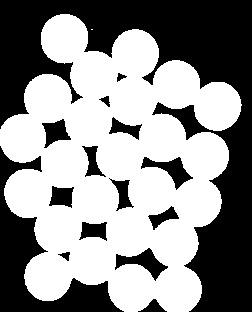
**from** **matplotlib** **import** pyplot **as** plt

img = cv2.imread('coins.png')

gray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

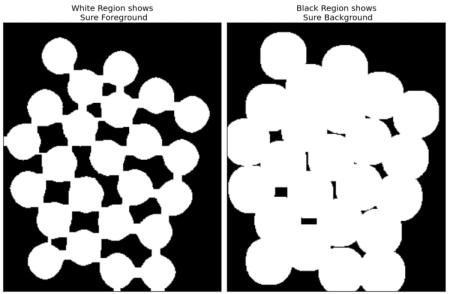
ret, thresh = cv2.threshold(gray,0,255,cv2.THRESH\_BINARY\_INV+cv2.THRESH\_OTSU)

ผล:



ตอนนี้เราจำเป็นต้องลบเสียงสีขาวเล็ก ๆ ในภาพ สำหรับการที่เราสามารถใช้การเปิดทางสัณฐานวิทยา เพื่อขจัดหลุมเล็ก ๆ ในวัตถุเราสามารถใช้การปิดก้านทางสัณฐานวิทยา ดังนั้นตอนนี้เรารู้แน่ว่าบริเวณใกล้กับจุดศูนย์กลางของวัตถุอยู่เบื้องหน้าและพื้นที่ห่างจากวัตถุมากเป็นพื้นหลัง เฉพาะภูมิภาคที่เราไม่แน่ใจว่าเป็นเขตแดนของเหรียญ

ดังนั้นเราจำเป็นต้องแยกพื้นที่ที่เรามั่นใจว่าเป็นเหรียญ การกร่อนจะเอาพิกเซลขอบเขตออก ดังนั้นสิ่งที่เหลือเราสามารถตรวจสอบว่าเป็นเหรียญ ที่จะทำงานถ้าวัตถุไม่ได้สัมผัสกัน แต่เนื่องจากพวกเขาได้สัมผัสกันและกันตัวเลือกอื่นที่ดีก็คือการหาการเปลี่ยนแปลงระยะทางและใช้เกณฑ์ที่เหมาะสม ต่อไปเราต้องหาพื้นที่ที่เรามั่นใจว่าไม่ใช่เหรียญ สำหรับการที่เราขยายผล การขยายตัวจะเพิ่มขอบเขตของวัตถุต่อพื้นหลัง ด้วยวิธีนี้เราสามารถตรวจสอบให้แน่ใจว่าพื้นที่ใดในพื้นหลังในผลที่ได้คือพื้นหลังจริงๆเนื่องจากขอบเขตถูกลบออก ดูภาพด้านล่าง



ส่วนที่เหลือคือพื้นที่ที่เราไม่มีความคิดไม่ว่าจะเป็นเหรียญหรือพื้นหลัง ขั้นตอนวิธีลุ่มน้ำควรหาได้ พื้นที่เหล่านี้เป็นปกติรอบขอบเขตของเหรียญที่เบื้องหน้าและพื้นหลังตอบสนอง (หรือแม้กระทั่งสองเหรียญที่แตกต่างกันตอบสนอง) เราเรียกว่าเส้นขอบ สามารถหาได้จากการลบพื้นที่ sure\_fg จากพื้นที่ sure\_bg

*# noise removal*

kernel = np.ones((3,3),np.uint8)

opening = cv2.morphologyEx(thresh,cv2.MORPH\_OPEN,kernel, iterations = 2)

*# sure background area*

sure\_bg = cv2.dilate(opening,kernel,iterations=3)

*# Finding sure foreground area*

dist\_transform = cv2.distanceTransform(opening,cv2.DIST\_L2,5)

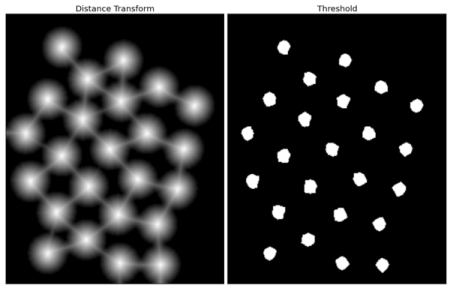
ret, sure\_fg = cv2.threshold(dist\_transform,0.7\*dist\_transform.max(),255,0)

*# Finding unknown region*

sure\_fg = np.uint8(sure\_fg)

unknown = cv2.subtract(sure\_bg,sure\_fg)

ดูผลลัพธ์ ในภาพที่มีเกณฑ์เราได้รับบางส่วนของเหรียญที่เรามั่นใจในเหรียญและพวกเขาจะออกตอนนี้ (ในบางกรณีคุณอาจจะสนใจในการแบ่งส่วนเบื้องหน้าเท่านั้นไม่ได้ในการแยกวัตถุที่สัมผัสร่วมกัน. ในกรณีที่คุณไม่จำเป็นต้องใช้ระยะเปลี่ยนเพียงแค่การกัดเซาะจะเพียงพอ. การกัดเซาะเป็นเพียงวิธีอื่นที่จะดึงแน่ใจว่าพื้นที่เบื้องหน้าที่ ทั้งหมด.)



ตอนนี้เรารู้แน่นอนว่าเป็นพื้นที่ของเหรียญซึ่งเป็นพื้นหลังและทั้งหมด ดังนั้นเราจึงสร้างเครื่องหมาย (เป็นอาร์เรย์ที่มีขนาดเดียวกับภาพต้นฉบับ แต่มีประเภทข้อมูล int32) และกำหนดป้ายกำกับพื้นที่ภายใน ภูมิภาคที่เรารู้จักอย่างแน่นอน (ไม่ว่าจะเป็นเบื้องหลังหรือพื้นหลัง) จะมีป้ายกำกับว่ามีจำนวนเต็มบวกใด ๆ แต่ตัวเลขที่แตกต่างกันและพื้นที่ที่เราไม่ทราบว่าจะเหลือเพียงศูนย์เท่านั้น สำหรับวันนี้เราใช้cv2.connectedComponents () ป้ายชื่อพื้นหลังของภาพด้วย 0 จากนั้นวัตถุอื่น ๆ จะมีข้อความจำนวนเต็มเริ่มต้นจาก 1

แต่เรารู้ว่าถ้าพื้นหลังถูกทำเครื่องหมายด้วย 0 ลุ่มน้ำจะพิจารณาว่าเป็นพื้นที่ที่ไม่รู้จัก ดังนั้นเราต้องการทำเครื่องหมายด้วยจำนวนเต็มต่างกัน แต่เราจะทำเครื่องหมายพื้นที่ที่ไม่รู้จักซึ่งกำหนดโดยไม่ทราบด้วย 0

*# Marker labelling*

ret, markers = cv2.connectedComponents(sure\_fg)

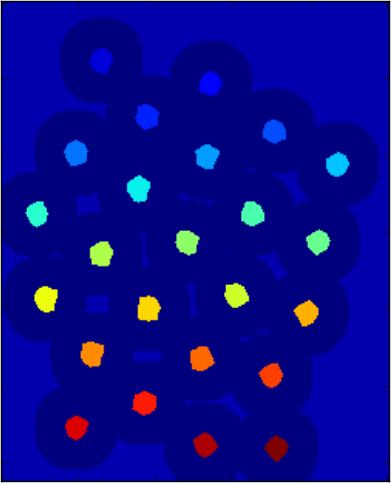
*# Add one to all labels so that sure background is not 0, but 1*

markers = markers+1

*# Now, mark the region of unknown with zero*

markers[unknown==255] = 0

ดูผลลัพธ์ที่แสดงในรูปแบบสี JET บริเวณสีน้ำเงินเข้มแสดงพื้นที่ที่ไม่รู้จัก เหรียญที่แน่นอนมีสีที่มีค่าต่างกัน พื้นที่ที่เหลือซึ่งเป็นพื้นหลังที่แน่นอนจะแสดงเป็นสีน้ำเงินที่มีน้ำหนักเบาเมื่อเทียบกับพื้นที่ที่ไม่รู้จัก



ตอนนี้เครื่องหมายของเราพร้อมแล้ว ถึงเวลาแล้วที่ขั้นตอนสุดท้ายให้ใช้ลุ่มน้ำ ภาพเครื่องหมายจะถูกแก้ไข เขตแดนจะถูกทำเครื่องหมายด้วย -1

markers = cv2.watershed(img,markers)

img[markers == -1] = [255,0,0]

ดูผลลัพธ์ด้านล่าง สำหรับเหรียญบางพื้นที่ที่พวกเขาสัมผัสจะถูกแบ่งส่วนอย่างถูกต้องและสำหรับบางพวกเขาไม่ได้

