ตรวจจับใบหน้าโดยใช้ Haar Cascades

เป้าหมาย

ในเซสชั่นนี้,

* เราจะเห็นพื้นฐานของการตรวจหาใบหน้าโดยใช้ตัวเก็บประจุแบบแฮร์แครช
* เราจะขยายเหมือนกันสำหรับการตรวจจับตาเป็นต้น

พื้นฐาน

การตรวจหาวัตถุโดยใช้ตัวเก็บประจุแบบแฮคที่ใช้ Haar เป็นวิธีการตรวจจับวัตถุที่มีประสิทธิภาพซึ่งเสนอโดยพอลวิโอลาและไมเคิลโจนส์ในเอกสาร "การตรวจหาวัตถุอย่างรวดเร็วโดยใช้ Cascade ที่เพิ่มขึ้นของคุณลักษณะอย่างง่าย" ในปี 2544 เป็นวิธีการเรียนรู้ด้วยเครื่อง ฟังก์ชั่นน้ำตกได้รับการฝึกฝนจากภาพลักษณ์ที่ดีและไม่ดี จากนั้นจะใช้เพื่อตรวจจับวัตถุในภาพอื่น ๆ

ที่นี่เราจะทำงานร่วมกับการตรวจหาใบหน้า ขั้นตอนแรกอัลกอริทึมต้องการภาพที่เป็นบวก (ภาพใบหน้า) และภาพเชิงลบ (ภาพที่ไม่มีใบหน้า) ในการฝึกอบรมตัวแบ่งประเภท จากนั้นเราจำเป็นต้องดึงข้อมูลออกจากคุณลักษณะ สำหรับคุณลักษณะนี้จะใช้คุณลักษณะด้านล่างที่แสดงในภาพด้านล่าง พวกเขาเป็นเหมือนเมล็ดของเรา convolutional คุณลักษณะแต่ละอย่างเป็นค่าเดียวที่ได้จากการลบผลรวมของพิกเซลภายใต้สี่เหลี่ยมผืนผ้าสีขาวจากผลรวมของพิกเซลภายใต้สี่เหลี่ยมสีดำ



ตอนนี้ทุกขนาดที่เป็นไปได้และตำแหน่งของเคอร์เนลแต่ละตัวจะถูกใช้เพื่อคำนวณคุณสมบัติมากมาย แม้เพียงหน้าต่าง 24x24 มากกว่า 160000 คุณลักษณะ) สำหรับการคำนวณแต่ละคุณลักษณะเราจำเป็นต้องหาจำนวนพิกเซลที่อยู่ในสี่เหลี่ยมสีขาวและสีดำ เพื่อแก้ปัญหานี้พวกเขานำเสนอภาพรวม ช่วยให้การคำนวณผลรวมของพิกเซลง่ายขึ้นขนาดของพิกเซลอาจสูงถึงสี่เท่า ดีใช่มั้ย? มันทำให้สิ่งต่างๆเร็วมาก

แต่ในบรรดาคุณลักษณะทั้งหมดที่เราคำนวณส่วนใหญ่ไม่เกี่ยวข้อง ตัวอย่างเช่นพิจารณาภาพด้านล่าง แถวบนสุดแสดงคุณลักษณะที่ดีสองประการ คุณลักษณะแรกที่เลือกดูเหมือนจะเน้นที่บริเวณที่ดวงตามักมืดกว่าบริเวณจมูกและแก้ม คุณลักษณะที่สองเลือกขึ้นอยู่กับคุณสมบัติที่ดวงตามืดกว่าสะพานจมูก แต่หน้าต่างเดียวกันกับแก้มหรือที่อื่นใดที่ไม่เกี่ยวข้อง ดังนั้นเราจะเลือกคุณสมบัติที่ดีที่สุดจากคุณลักษณะ 160000 + ได้อย่างไร มันจะทำได้โดยAdaBoost



สำหรับตอนนี้เราใช้คุณลักษณะแต่ละอย่างในภาพการฝึกอบรมทั้งหมด สำหรับแต่ละคุณลักษณะจะพบเกณฑ์ที่ดีที่สุดซึ่งจะแบ่งหน้าเป็นบวกและลบ แต่ชัดจะมีข้อผิดพลาดหรือ misclassifications เราเลือกคุณลักษณะที่มีอัตราความผิดพลาดต่ำสุดซึ่งหมายความว่าคุณลักษณะเหล่านี้เป็นคุณสมบัติที่สามารถแยกแยะภาพใบหน้าและภาพที่ไม่ใช่ใบหน้าได้ดีที่สุด (กระบวนการนี้ไม่ง่ายเท่านี้ภาพแต่ละภาพจะได้รับน้ำหนักที่เท่ากันในตอนเริ่มต้นหลังจากแต่ละการจำแนกแล้วน้ำหนักของรูปภาพที่มีการจัดประเภทผิดพลาดจะเพิ่มขึ้นจากนั้นทำซ้ำอีกขั้นตอนเดียวกันอัตราการผิดพลาดใหม่จะถูกคำนวณรวมทั้งน้ำหนักใหม่ด้วย กระบวนการดำเนินต่อไปจนกว่าจะต้องมีความถูกต้องหรืออัตราความผิดพลาดที่ได้รับหรือต้องมีจำนวนคุณลักษณะที่พบ)

ตัวจัดจำแนกครั้งสุดท้ายคือผลรวมถ่วงน้ำหนักของตัวจำแนกประเภทที่อ่อนแอเหล่านี้ มันเรียกว่าอ่อนแอเพราะคนเดียวไม่สามารถจำแนกภาพ แต่ร่วมกับคนอื่น ๆ เป็นตัวจำแนกประเภทที่แข็งแกร่ง กระดาษกล่าวว่าแม้แต่ 200 คุณสมบัติให้การตรวจสอบมีความถูกต้อง 95% การตั้งค่าขั้นสุดท้ายของพวกเขามีคุณลักษณะประมาณ 6000 รายการ (ลองนึกภาพลดจาก 160000 + คุณสมบัติเป็น 6000 คุณสมบัตินั่นคือกำไรใหญ่)

ตอนนี้คุณถ่ายภาพแล้ว ใช้หน้าต่างแต่ละขนาด 24x24 ใช้คุณสมบัติ 6000 ไปเลย ตรวจสอบว่ามีใบหน้าหรือไม่ ว้าว .. ว้าว .. มันไม่น้อยไม่มีประสิทธิภาพและเสียเวลา? ใช่แล้ว. ผู้เขียนมีทางออกที่ดีสำหรับเรื่องนี้

ในภาพส่วนใหญ่ของพื้นที่ภาพเป็นพื้นที่ที่ไม่ใช่ใบหน้า ดังนั้นจึงเป็นความคิดที่ดีกว่าที่จะมีวิธีง่ายๆในการตรวจสอบว่าหน้าต่างไม่ใช่พื้นที่หน้าหรือไม่ หากไม่เป็นเช่นนั้นให้ทิ้งในภาพเดียว อย่าประมวลผลอีกครั้ง แทนที่จะมุ่งเน้นไปที่พื้นที่ที่สามารถเผชิญได้ ด้วยวิธีนี้เราสามารถหาเวลาเพิ่มเติมในการตรวจสอบพื้นที่ใบหน้าที่เป็นไปได้

สำหรับพวกเขานำแนวคิดของCascade ของลักษณนาม แทนที่จะใช้คุณลักษณะทั้งหมด 6000 ในหน้าต่างให้จัดกลุ่มคุณลักษณะไว้ในขั้นตอนต่างๆของตัวแบ่งประเภทและใช้ทีละแบบ (โดยปกติขั้นตอนแรก ๆ จะมีคุณลักษณะน้อยมาก) ถ้าหน้าต่างล้มเหลวในขั้นตอนแรกให้ละทิ้ง เราไม่พิจารณาคุณลักษณะที่เหลืออยู่ ถ้าผ่านไปใช้ขั้นตอนที่สองของคุณลักษณะและดำเนินกระบวนการต่อ หน้าต่างที่ผ่านทุกขั้นตอนคือบริเวณใบหน้า แผนการอย่างไร!

เครื่องตรวจจับของ Authors มีคุณสมบัติ 6000+ พร้อมด้วย 38 ขั้นตอนพร้อมคุณสมบัติ 1, 10, 25, 25 และ 50 ในห้าขั้นตอนแรก (สองคุณสมบัติในภาพข้างต้นได้รับจริงเป็นสองคุณสมบัติที่ดีที่สุดจาก Adaboost) ตามที่ผู้เขียนได้รับการประเมินโดยเฉลี่ย 10 คุณลักษณะจาก 6000+ ต่อหน้าต่างย่อย

นี่เป็นคำอธิบายที่เข้าใจได้ง่ายว่าการตรวจจับใบหน้าของ Viola-Jones ทำงานได้ดีเพียงใด อ่านกระดาษสำหรับรายละเอียดเพิ่มเติมหรือดูข้อมูลอ้างอิงในส่วนแหล่งข้อมูลเพิ่มเติม

Haar-cascade Detection ใน OpenCV

OpenCV มาพร้อมกับผู้ฝึกสอนและเครื่องตรวจจับ ถ้าคุณต้องการฝึกตัวจำแนกของคุณเองสำหรับวัตถุใด ๆ เช่นรถยนต์เครื่องบินเป็นต้นคุณสามารถใช้ OpenCV เพื่อสร้าง รายละเอียดทั้งหมดจะได้รับที่นี่: [Cascade Classifier Training](http://docs.opencv.org/doc/user_guide/ug_traincascade.html)

ที่นี่เราจะจัดการกับการตรวจจับ OpenCV มีตัวจำแนกประเภทที่ได้รับการฝึกมาก่อนล่วงหน้าหลายรูปแบบสำหรับใบหน้าตายิ้ม ฯลฯ ไฟล์ XML เหล่านี้จะถูกเก็บไว้ในโฟลเดอร์/ opencv / data / haarcascades / ลองสร้างเครื่องตรวจจับใบหน้าและตาด้วย OpenCV

ขั้นแรกเราต้องโหลดตัวแบ่งประเภท XML ที่จำเป็น จากนั้นโหลดภาพนำเข้า (หรือวิดีโอ) ในโหมดสีเทา

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

**import** **cv2**

face\_cascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

eye\_cascade = cv2.CascadeClassifier('haarcascade\_eye.xml')

img = cv2.imread('sachin.jpg')

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

ตอนนี้เราพบใบหน้าในภาพแล้ว หากพบใบหน้าจะส่งกลับตำแหน่งของใบหน้าที่ตรวจพบเป็น Rect (x, y, w, h) เมื่อเราได้ตำแหน่งเหล่านี้แล้วเราสามารถสร้าง ROI สำหรับใบหน้าและใช้การตรวจจับตากับ ROI (เนื่องจากตาอยู่บนใบหน้าเสมอ !!!)

faces = face\_cascade.detectMultiScale(gray, 1.3, 5)

**for** (x,y,w,h) **in** faces:

cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)

roi\_gray = gray[y:y+h, x:x+w]

roi\_color = img[y:y+h, x:x+w]

eyes = eye\_cascade.detectMultiScale(roi\_gray)

**for** (ex,ey,ew,eh) **in** eyes:

cv2.rectangle(roi\_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)

cv2.imshow('img',img)

cv2.waitKey(0)

cv2.destroyAllWindows()

ผลลัพธ์มีลักษณะดังนี้:

