بسم الله الرحمن الرحیم

درس بینایی ماشین

دکتر صفابخش

گزارش پروژه تشخیص چهره

موعد تحویل: ۱۳۹۸,۱۱,۱۹

دانشجو: حمیدرضا فهیمی

توضیح:

در گزارش حاضر قصد داریم روند طی شده برای برنامه تشخیص چهره را شرح دهیم. هدف نهایی در این پروژه آن است که سیستمی طراحی کنیم که این توانایی را داشته باشد که با دریافت عکس جدید از افراد، در خصوص افرادی که از قبل چهره ی آنها به برنامه شناسانده شده است، تشخیص چهره را انجام داده و علاوه بر شناسایی محل صورت در تصویر، هویت شخص را مطابق با داده آموزش داده شده تعیین نماید.

مراحل روند مذکور در یک نگاه کلی عبارت است از:

تهیه مجموعه داده (dataset) آزمایش

آموزش (training) لازم برای شخیص جای صورت افراد در تصویر

اجرای فرایند استخراج قاب چهره افراد از تصویر

تطبیق چهره ها با داده های طبقه بندی شده از قبل

صحت سنجی نتایج

در ادامه به شرح هر مرحله می پردازیم.

تهیه مجموعه داده (dataset) آزمایش

با یک جستجوی ساده مجموعه داده های متعددی برای برنامه های تشخیص چهره می توان یافت. در این پروژه از مجموعه داده تحت عنوان Frontal Face Database تهیه شده توسط Markus Weber از دانشگاه Caltech در سال ۱۹۹۹ استفاده شده است. آدرس اینترنتی در زیر آورده شده است:

http://www.vision.caltech.edu/html-files/archive.html/

این مجموعه شامل ۴۵۰ تصویر رو به رو از ۲۶ شخص مختلف می باشد که از هر نفر چند نسخه تصویر با شرایط حالت چهره، نورپردازی و ... متفاوت در آن موجود است. در زیر چند نمونه از تصاویر افراد آورده شده است:

آموزش (training) لازم برای شخیص جای صورت افراد در تصویر

برای استخراج قاب چهره افراد مجموعه داده آموزش داده شده ای برای برنامه نویسی OpenCV موجود است که HaarCascade نامیده می شود. تشخیص اجسام با استفاده از الگوریتمHaar Cascade ، یک روش بسیار موثر است که اولین بار توسط Paul Viola و Michael Jones در سال ۲۰۰۱ مطرح شد. این الگوریتم بر پایه یادگیری ماشین بوده و بدین صورت عمل می کند که با استفاده از تصاویری که چهره در آنها وجود دارد (تصاویر

مثبت) و تصاویری که چهره در آنها وجود ندارد (تصاویر منفی) تابعی را آموزش می دهد تا بتواند مولفه هایی مانند وجود فاصله بین چشم ها را پیدا کند.

این الگوریتم منحصر به تشخیص چهره نیست؛ و بیشتر یک آموزش دهنده است و می توان با آموزش دادن تابع دلخواه خودتان هر شی مانند ماشین، میز، مداد و ... را شناسایی کنید OpenCV .مجموعه ای از توابع از پیش آموزش دیده را برای تشخیص چهره، چشمان، لبخند و ... در خود دارد که می توان به سادگی از آنها استفاده کرد.

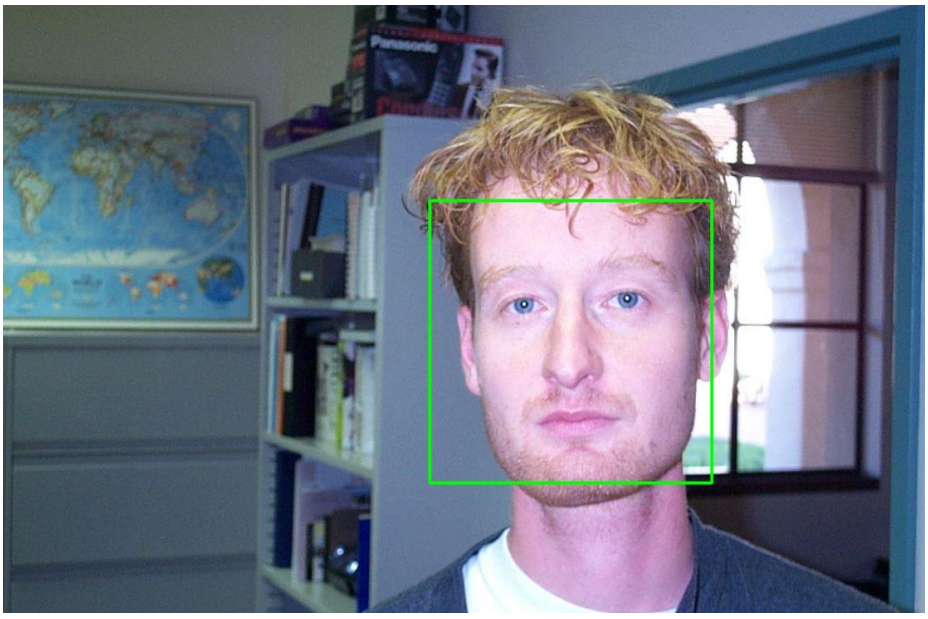
در پروژه حاضر از این امکان برای استخراج قاب اولیه چهره افراد استفاده کرده ایم. برای اطلاعات بیشتر راجع به این کتابخانه به لینک زیر مراجعه شود:

https://docs.opencv.org/3.4/d7/d8b/tutorial_py_face_detection.html

روش استفاده به این صورت است که یک فایل xml. در پوشه حاوی برنامه اصلی قرار دارد. دو خط زیر از برنامه، شیء مورد استفاده در کد برای استخراج قاب چهره را می سازد:

```
cascPath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascPath)
```

در ادامه پس از دریافت تصویر ورودی توسط برنامه، تابع detectMultiScale از شیء فوق را فراخوانی نموده و در خروجی تصویری به صورت زیر دریافت می کنیم که یک مربع دور چهره در آن کشیده شده است.



اجرای فرایند استخراج قاب چهره افراد از تصویر

با توجه به آنچه توضیح داده شد، مشخصا استخراج ناحیه موردنظر برای فرایند تطبیق در هر عکس به سادگی قابل انجام است. خط های زیر از برنامه، برای استخراج مستطیل رسم شده استفاده می شوند.

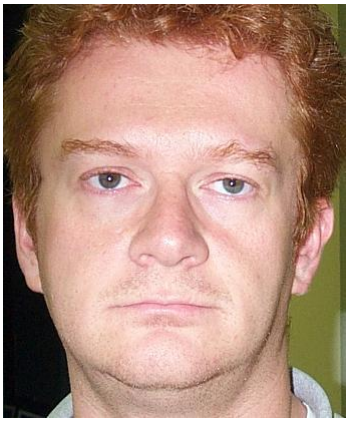
```
fac = faceCascade.detectMultiScale(frame, scaleFactor=1.1, minNeighbors=5,  minSize=(120, 120), flags = cv2.CASCADE_SCALE_IMAGE)

face = np.zeros((۱۰،۱۰،۳))

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

face = frame[y:y+h,x:x+w]

num = "face " + str(cnt)
```

به این ترتیب ROI هر تصویر را به دست آورده و به مراحل بعدی الگوریتم می رویم.

تطبیق چهره ها با داده های طبقه بندی شده از قبل

در این مرحله، ابتدا به برنامه تصویر چهره ی هر شخص به همراه هویت وی را می دهیم. هویت، در حقیقت در این جا همان نام تصاویر می باشد. داده های زیر در یک آدرس معین برای سیستم به عنوان معیار مقایسه تعیین شده اند. این تعداد، نمونه ای از تصویر ۲۶ نفر به همراه هویت آنها می باشند.

نام : karl



نام: john

نام: nina

و اما برای تطبیق هر عکس ورودی با تصاویر مشابه فوق که کلیشه محسوب می شوند، از الگوریتم SURF استفاده می کنیم. به این ترتیب که ابتدا شیء از کلاس xfeatures2d و با فراخوانی تابع SURF_create ساخته می شود.

```
detector = cv2.xfeatures2d.SURF_create()
```

در ادامه به وسیله SURF نقاط کلیدی را برای هر کلیشه و هر تصویر ورودی به دست خواهیم آورد. سپس در ادامه به وسیله تطبیقگر FlannBasedMatcher عملیات تطبیق را انجام می دهیم. کد اولیه تعریف به صورت زیر است:

```
index_params = dict(algorithm=0, trees=5)

search_params = dict()

flann = cv2.FlannBasedMatcher(index_params, search_params)
```

الگوریتم اصلی به شرح زیر است:

ابتدا تصویر ورودی که می خواهیم فرد را در آن شناسایی کنیم به برنامه می دهیم. سپس برنامه به ازای هر کلیشه در پوشه کلیشه ها، ابتدا با SURF نقاط کلیدی و توصیفگرها را شناخته و عملیات تطبیق را بین کلیشه فعلی حلقه با هر مستطیل شناخته شده در تصویر انجام می دهد:

```
matches = flann.knnMatch(desc_image, desc_gframe, k=2)
```

سپس تطبیق هایی که از ۰,۷ کمتر اختلاف دارند انتخاب می شوند:

```
goods[] =
for m, n in matches:
   if m.distance < 0.7*n.distance:
      goods.append(m)
```

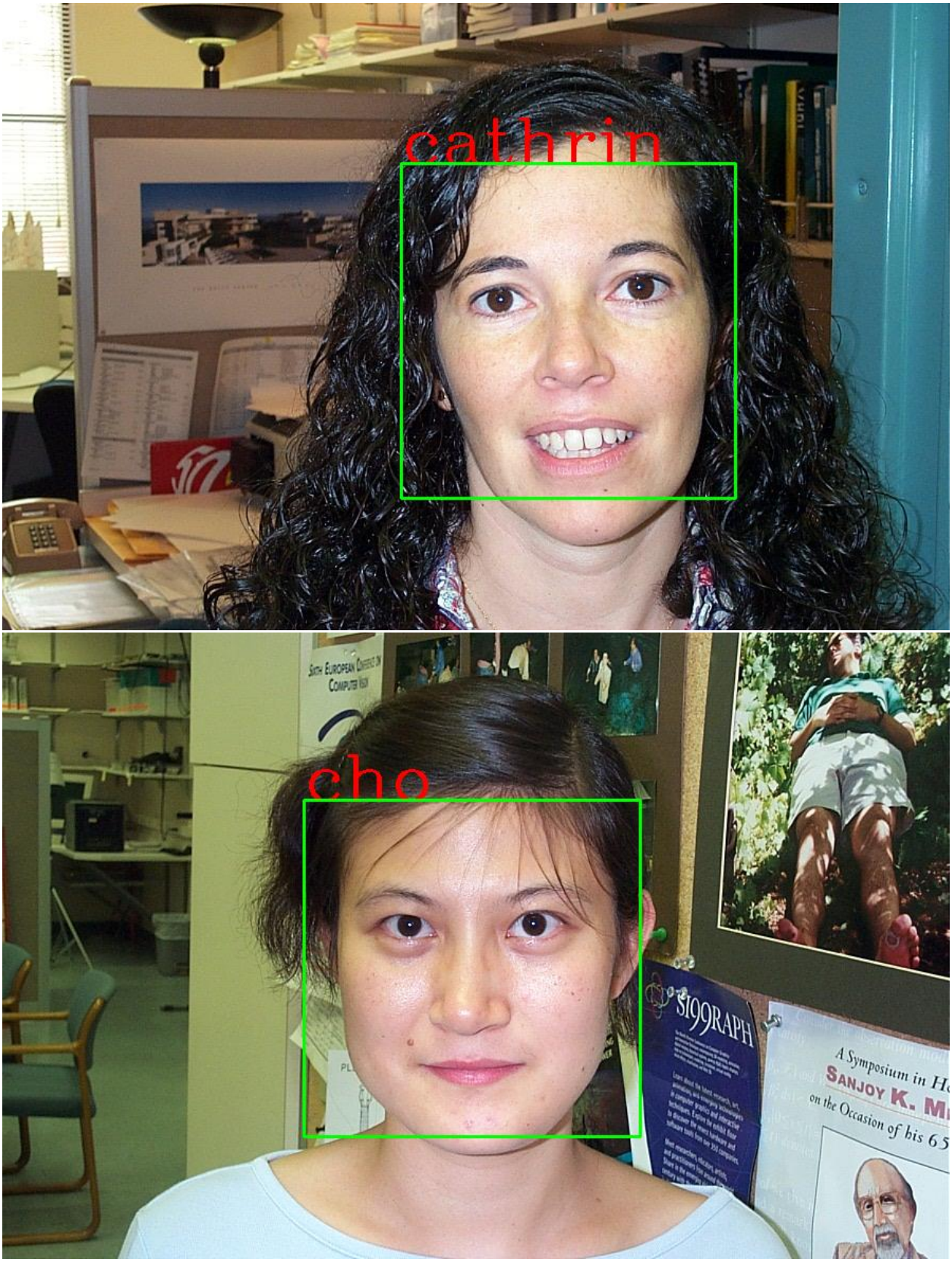
در ادامه اگر تعداد تطبیق های درست از آستانه ۴۰ عدد بیشتر باشد، الگوریتم تایید می کند که چهره کلیشه با مستطیل یافت شده منطبق است. در این صورت اسم شخص از نام فایل کلیشه روی تصویر نوشته می شود:

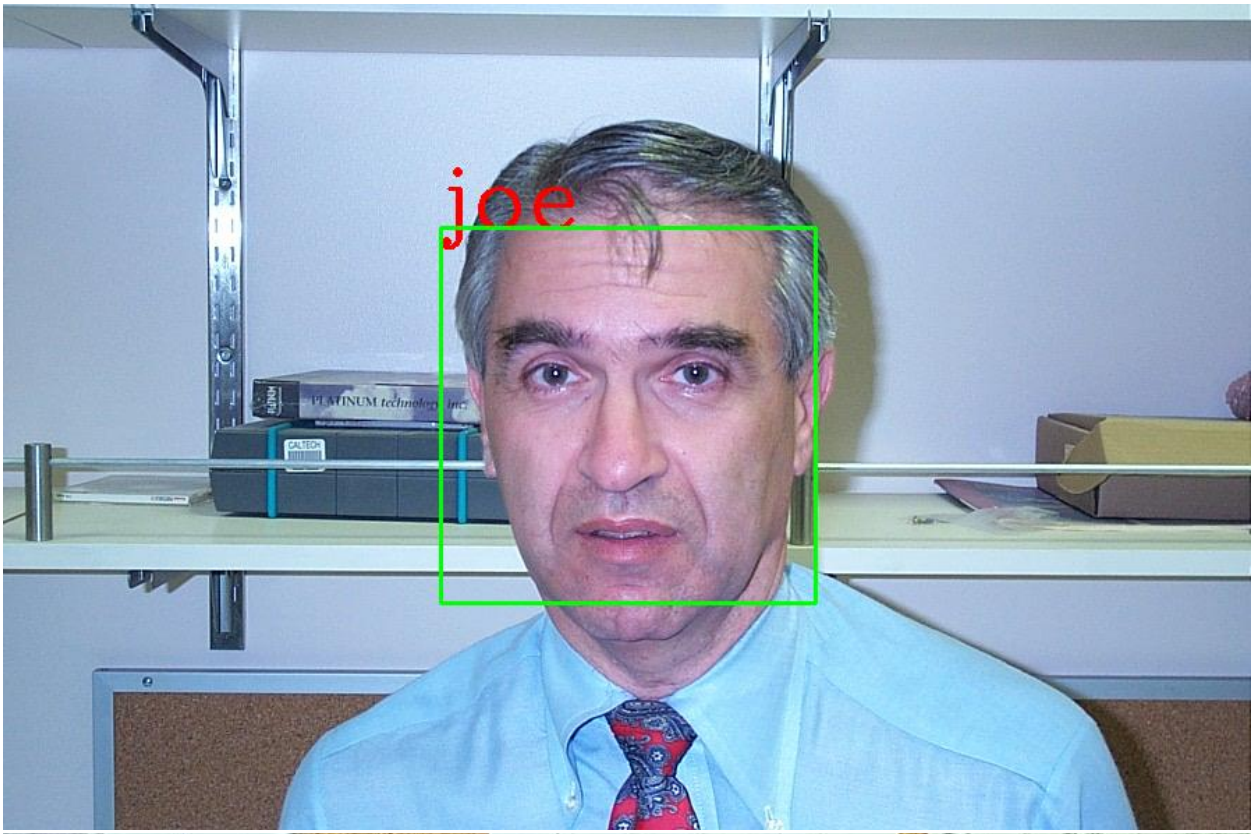
```
  print len(goods)
if len(goods) > 40:
   cv2.putText(oframe,name,(x,y),cv2.FONT_HERSHEY_COMPLEX,2,(0,0,255),2)
else: break
```

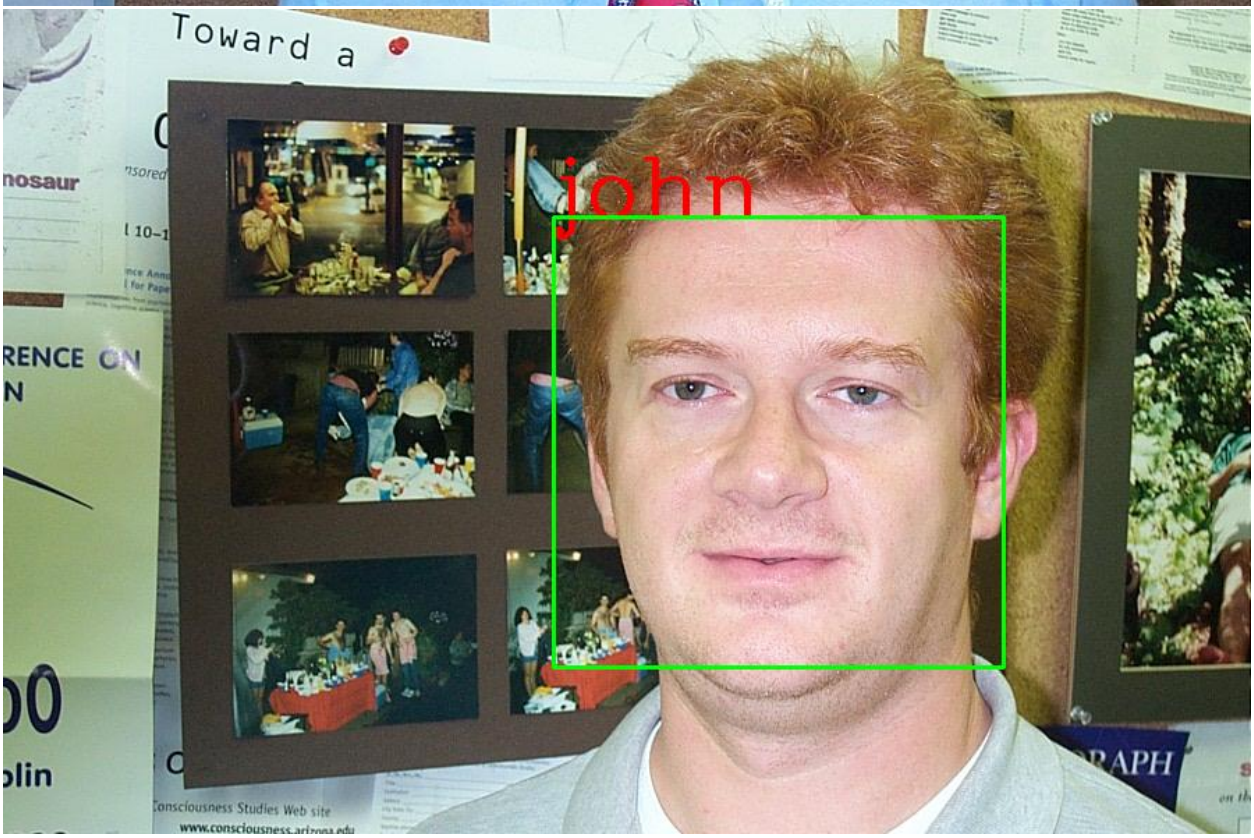
به این ترتیب چهره افراد تنها در صورت صدق آستانه معین تایید شده و برنامه متوقف می شود. این الگوریتم قابلیت اجرا برای تصویر حاوی چند چهره را نیز با همین کد دارا می باشد. البته در مجموعه داده آزمایش فعلی، همه تصاویر دارای تنها یک چهره هستند.

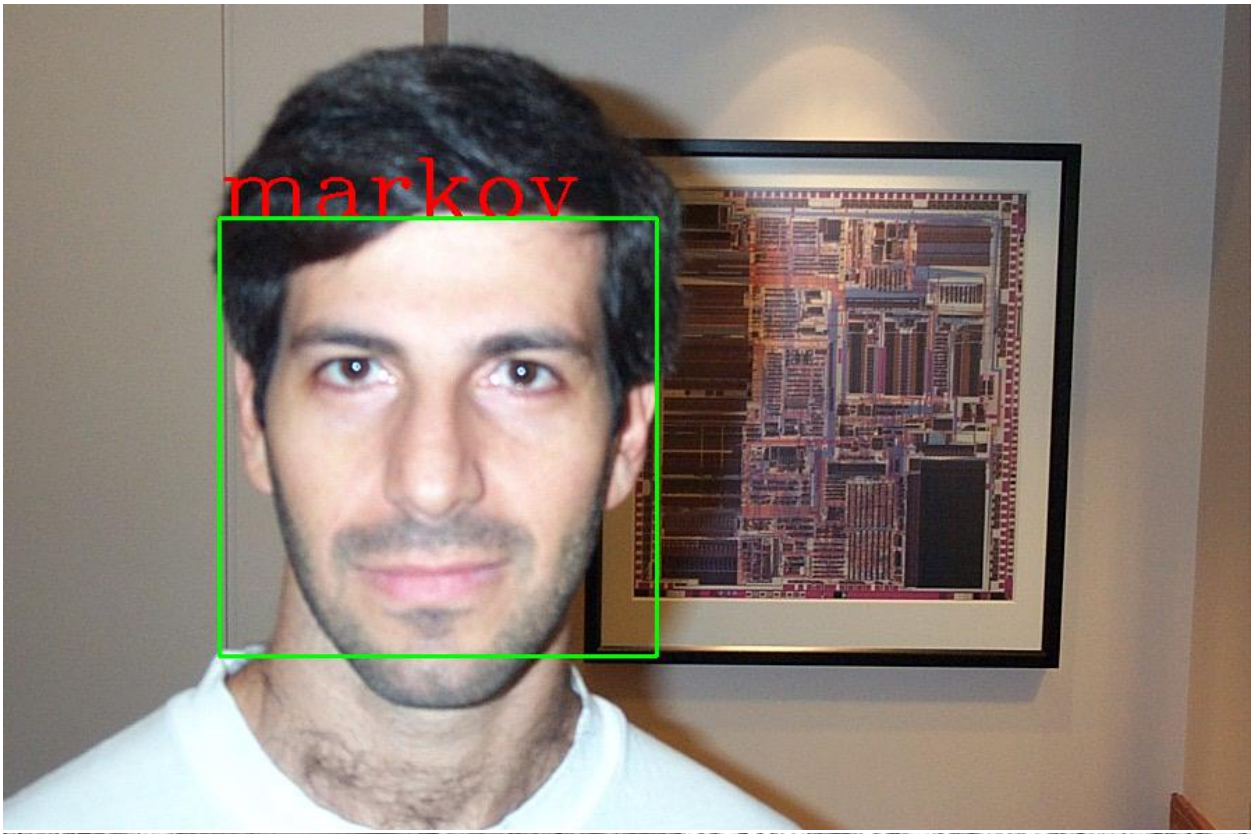
صحت سنجی نتایج

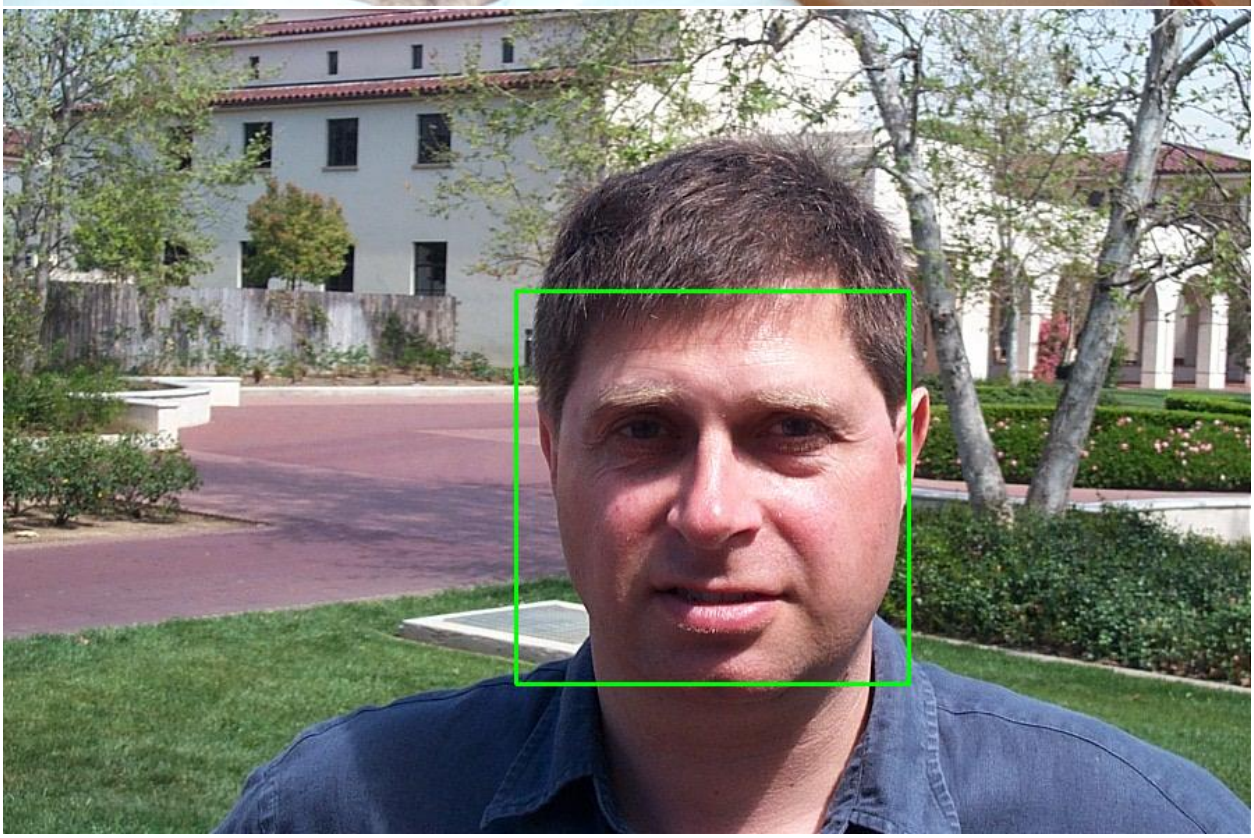
می توان گفت که عملکرد الگوریتم برای مجموعه داده موجود فوق العاده استفاده. تقریبا در ۹۰ درصد موارد الگوریتم به درستی تشخیص را انجام می دهد. موارد عدم تشخیص، شامل حالاتی است که چهره شخص حالت بسیار متفاوتی نسبت وضع عادی آن گرفته است. در زیر تصویر خروجی الگوریتم برای کلیشه های نشان داده شده در قسمت های قبلی مشهود است:

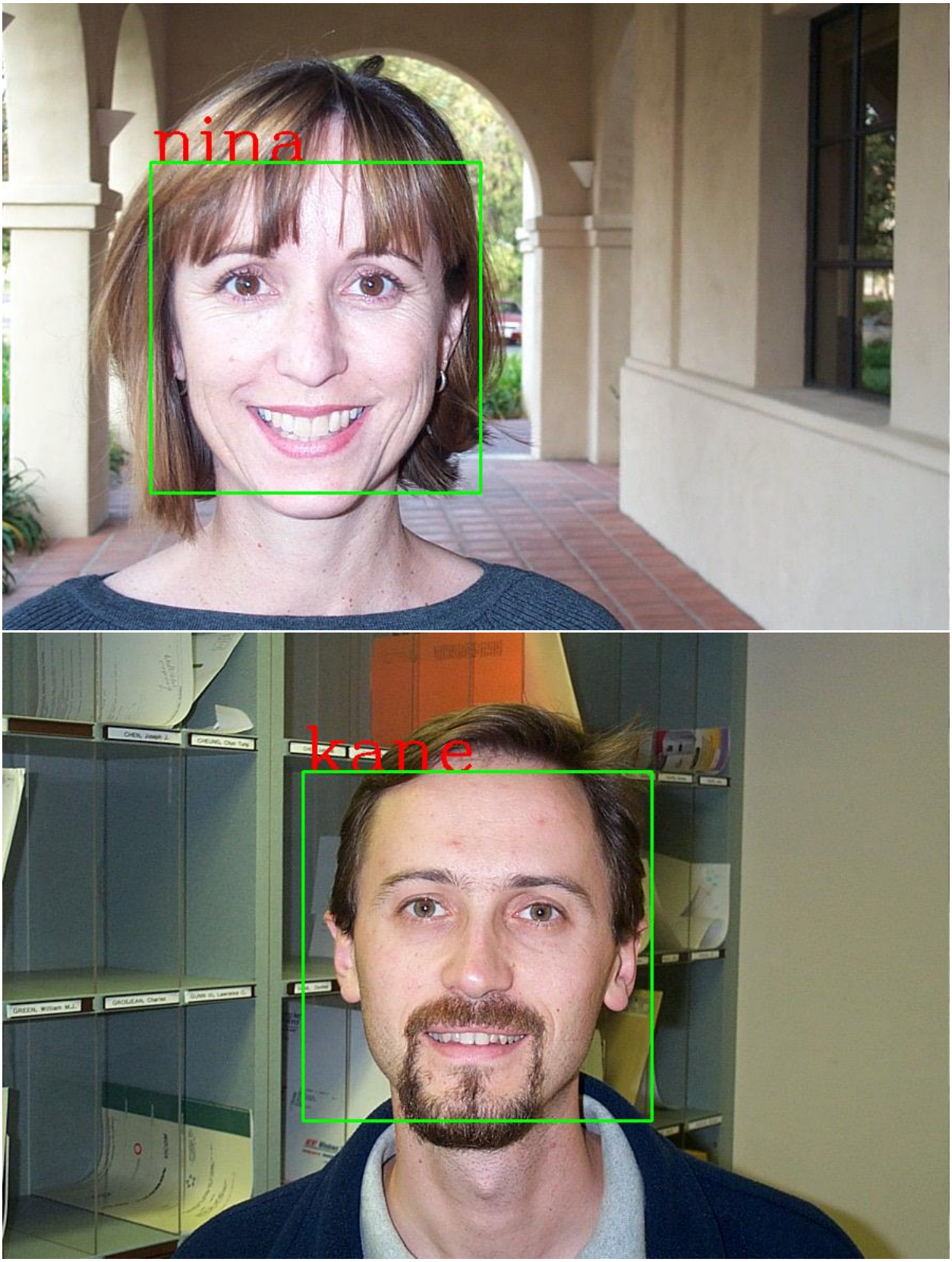

cathrin
cho
SIGGRAPH
A Symposium in H
SANJOY K. M
on the Occasion of his 65


joe


Toward a
john
Consciousness Studies Web site


markov




nina
kane