



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

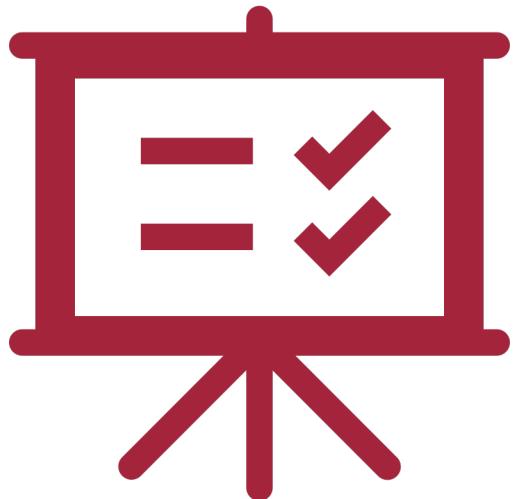
پیاده‌سازی قفل هوشمند با استفاده از روش شناسایی چهره مبتنی بر هوش مصنوعی

ارائه دهنده: امیرحسین علی‌بخشی

استاد درس: دکتر حمیدرضا زرندی

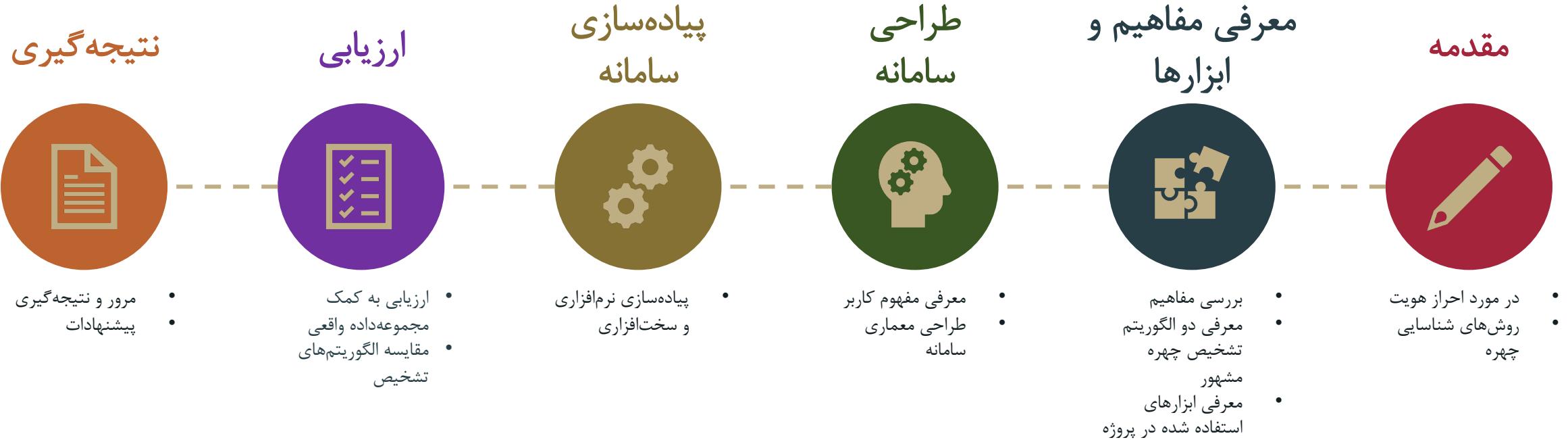
بهار ۱۴۰۲

هدف



- بررسی اهمیت احراز هویت
- معرفی روش‌ها و ابزارهای استفاده شده در پروژه
- طراحی و پیاده‌سازی پروژه
- ارزیابی پروژه به کمک مجموعه داده‌های چهره‌های واقعی
- نتیجه‌گیری و ارائه پیشنهادات

≡ سیر ارائه



مقدمه

در مورد احراز هویت

روش‌های شناسایی چهره



مقدمه



مقدمه

مفاهیم

طراحی

پیاده‌سازی

ارزیابی

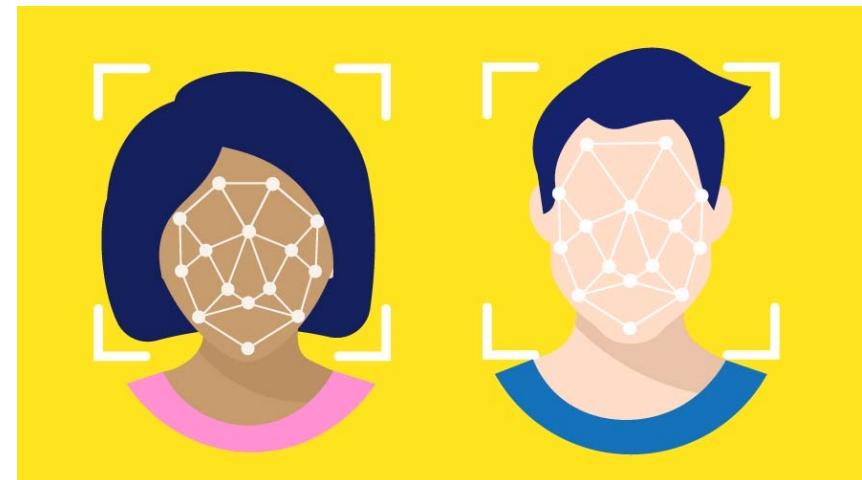
نتیجه



Name:

Password:

[Continue](#)



۷ / مقدمه

• روش‌های ارائه شده برای تشخیص چهره

• شبکه عصبی چند لایه Perceptron

• شبکه عصبی انتشار معکوس

★ • شبکه عصبی پیچشی

• ماشین بردار پشتیبان

• مدل‌های ترکیبی شبکه عصبی و Adaboost

★ • هیستوگرام شبکه‌های جهت‌دار

مقدمه

مفاهیم

طراحی

پیاده‌سازی

ارزیابی

نتیجه



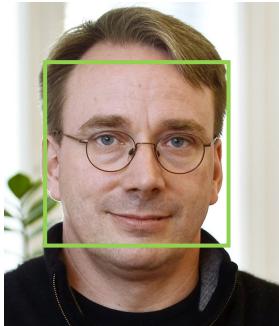
معرفی مفاهیم و ابزارها

بررسی مفاهیم

معرفی دو الگوریتم تشخیص چهره مشهور

معرفی ابزارهای استفاده شده در پروژه

معرفی مفاهیم پایه



مقدمه

مفاهیم

طراحی

پیاده‌سازی

ارزیابی

نتیجه

• الگوریتم تشخیص چهره

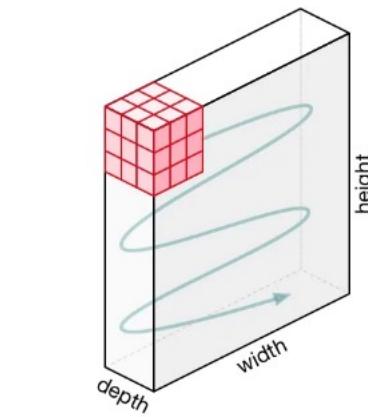
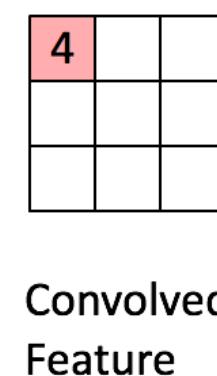
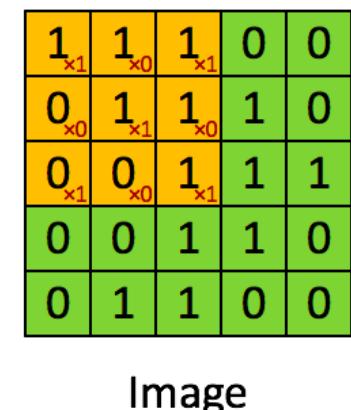
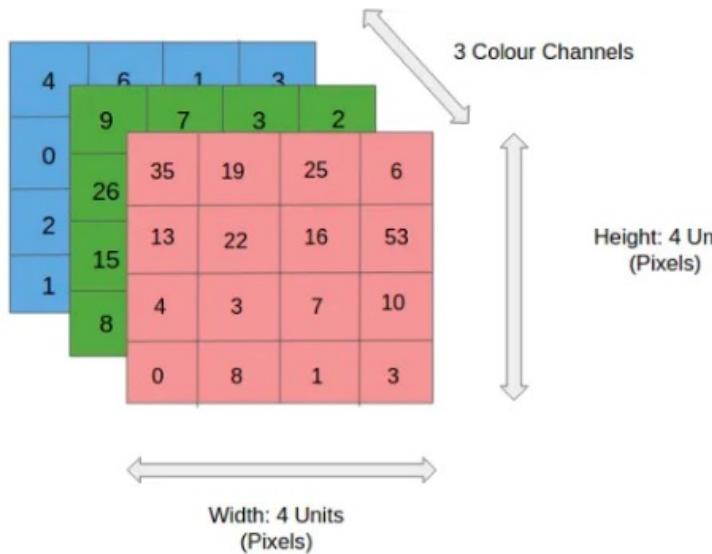
- الگوریتمی که مشخص می‌کند در تصویر یا ویدیوی ورودی خود، چهره‌ای وجود دارد یا خیر. فرقی نمی‌کند که چهره متعلق به چه کسی است و تنها هدف بررسی وجود چهره است.

• الگوریتم شناسایی چهره

- الگوریتمی که امکان شناسایی چهره‌های تشخیص‌داده شده توسط الگوریتم تشخیص چهره را بر اساس شاخص‌های بیومتریک یکتای موجود در چهره افراد را فراهم می‌کند.

معرفی دو الگوریتم مشهور تشخیص چهره

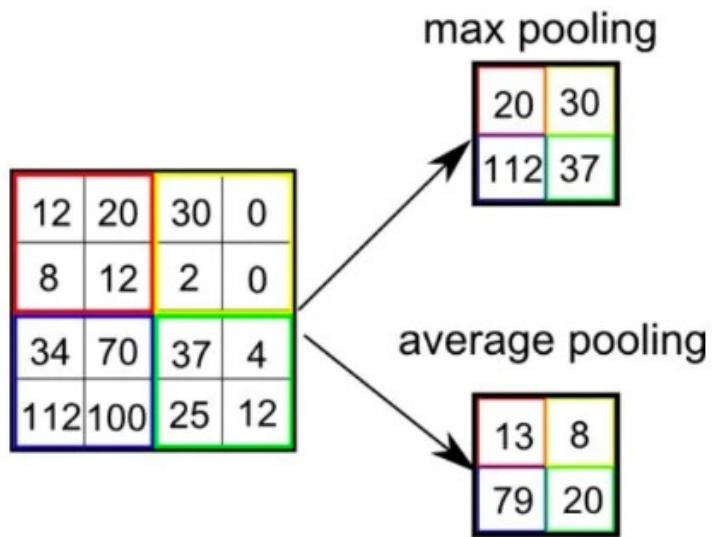
- مقدمه
- مفاهیم
- طراحی
- پیادهسازی
- ارزیابی
- نتیجه



شبکه عصبی پیچشی (CNN) •

- تقسیم عکس ورودی
- لایه پیچشی
- ضرب ماتریس هسته در پیکسل‌ها

معرفی دو الگوریتم مشهور تشخیص چهره



شبکه عصبی پیچشی (CNN)

یادآمد یک شبکه عصبی پیچشی

- لایه کاملا متصل: یادگیری ترکیب‌های خطی به کمک perceptronها و توابع فعال‌سازی

- تقسیم عکس ورودی

- لایه پیچشی

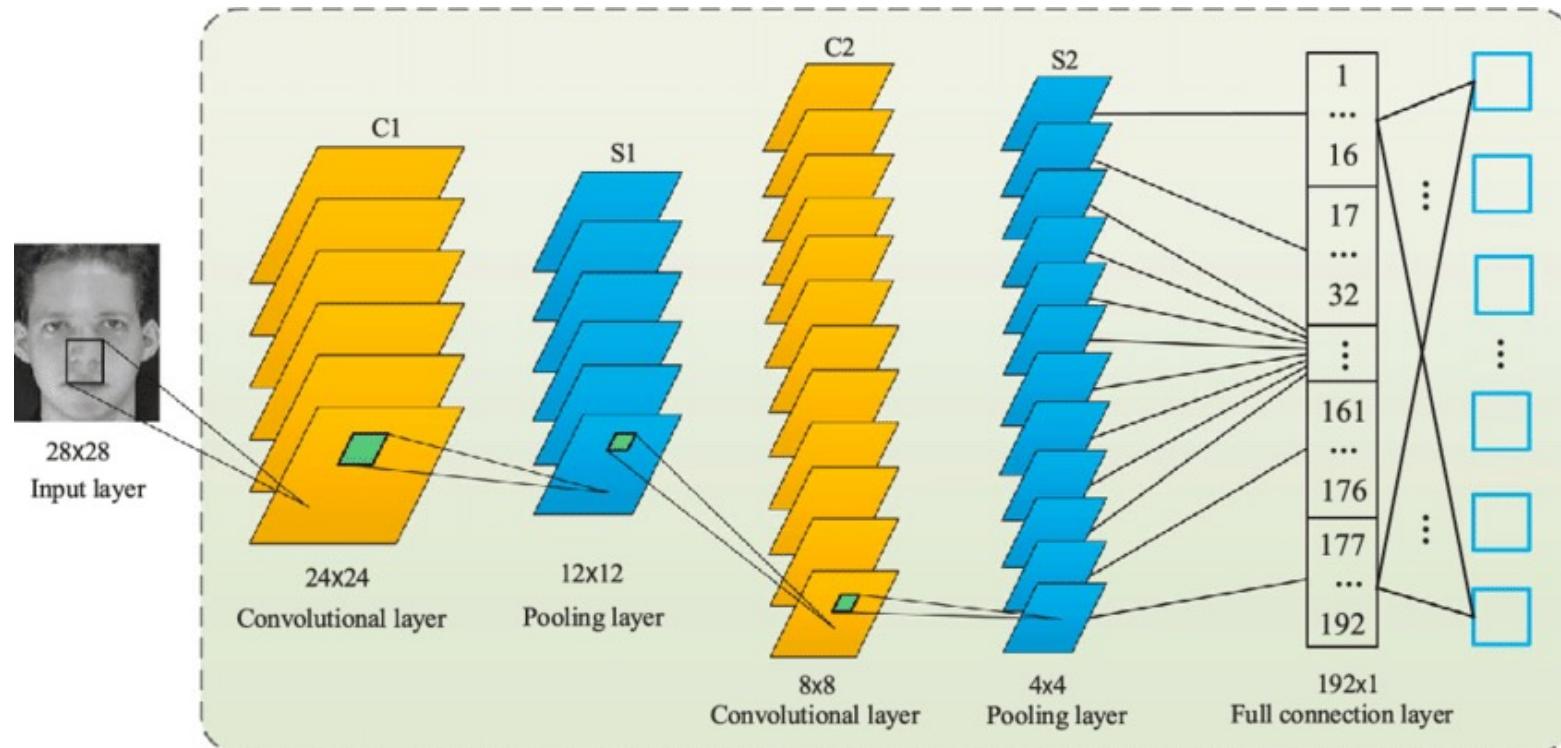
- ضرب ماتریس هسته در پیکسل‌ها

- ادغام حداقلی و میانگین

- ادغام حداقلی: مقاوم‌تر در برابر نویز

معرفی دو الگوریتم مشهور تشخیص چهره

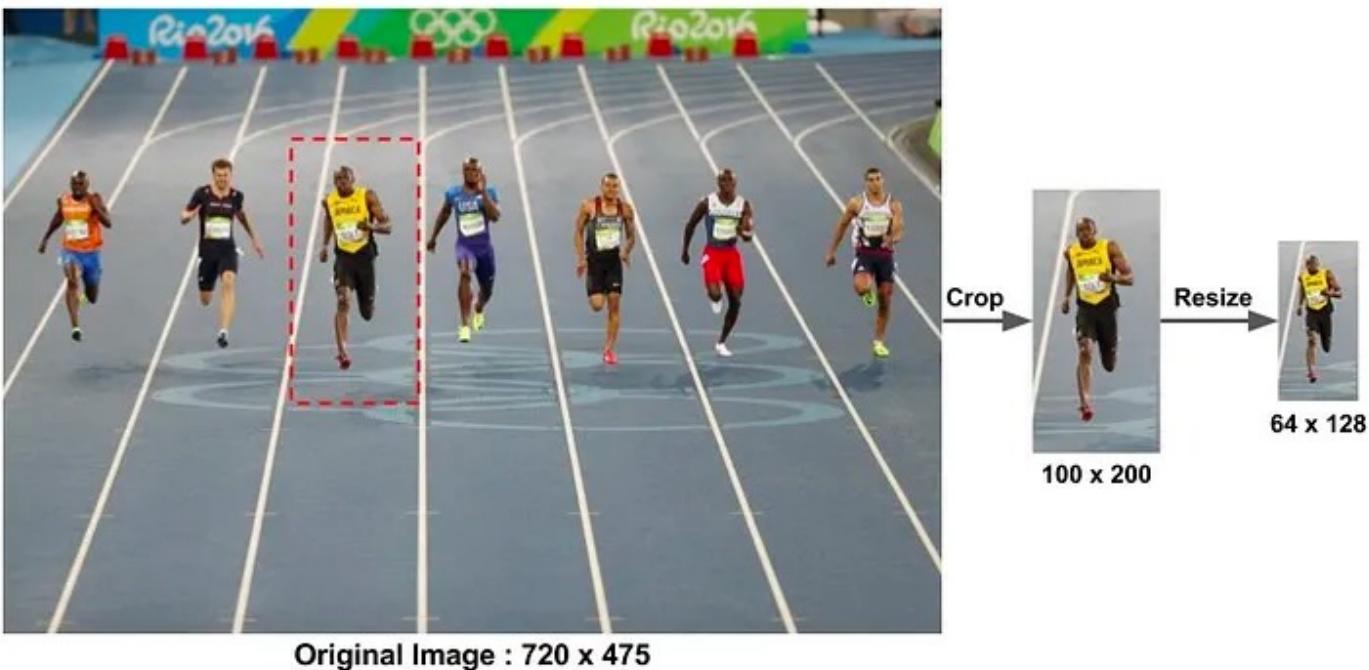
• شبکه عصبی پیچشی (CNN)



معرفی دو الگوریتم مشهور تشخیص چهره

• هیستوگرام شبیه‌های جهت‌دار (HOG)

• پیش‌پردازش



معرفی دو الگوریتم مشهور تشخیص چهره

$$G_x(r, c) = I(r, c + 1) - I(r, c - 1)$$

$$G_y(r, c) = I(r - 1, c) - I(r + 1, c)$$

$$Magnitude(\mu) = \sqrt{G_x^2 + G_y^2}$$

$$Angle(\theta) = \left| \tan^{-1} \left(\frac{G_y}{G_x} \right) \right|$$

مقدمه

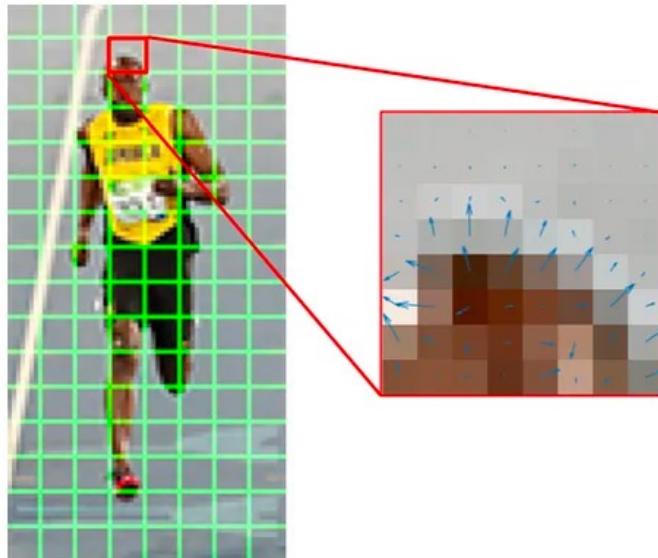
مفاهیم

طراحی

پیادهسازی

ارزیابی

نتیجه



• هیستوگرام شیب‌های جهت‌دار (HOG)

• پیش‌پردازش

• محاسبه شیب

• ساخت هیستوگرام از شیب

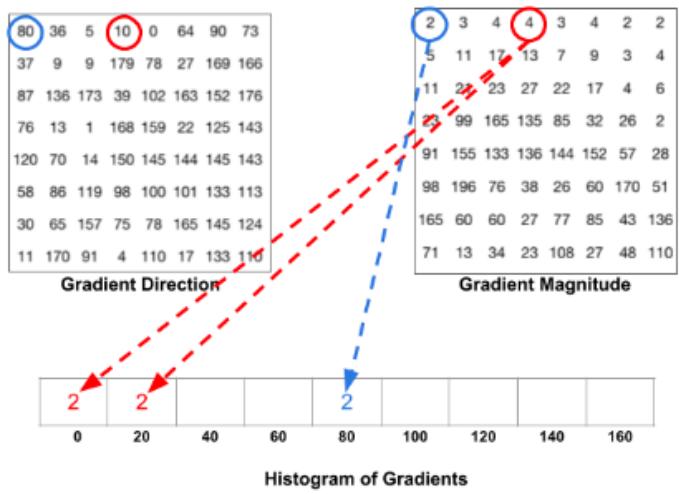
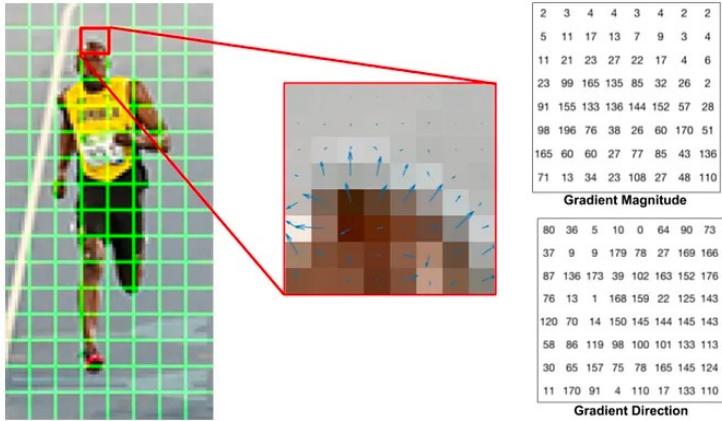
2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

معرفی دو الگوریتم مشهور تشخیص چهره

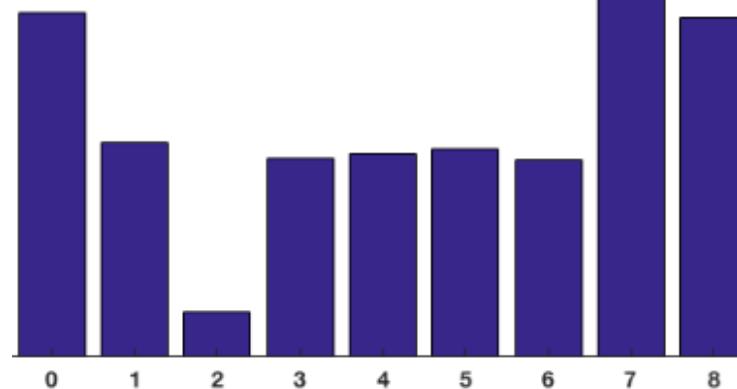


• هیستوگرام شیب‌های جهت‌دار (HOG)

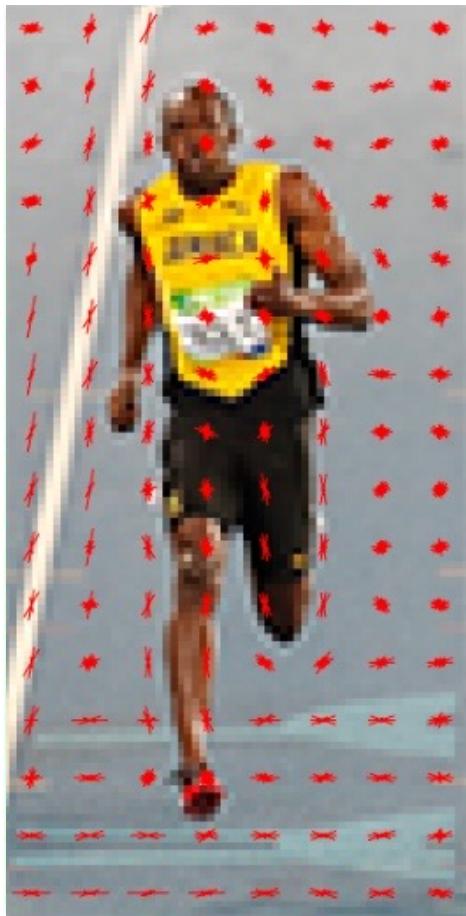
- پیش‌پردازش

- محاسبه شیب

- ساخت هیستوگرام از شیب



معرفی دو الگوریتم مشهور تشخیص چهره



• هیستوگرام شیب‌های جهت‌دار (HOG)

- مقدمه
- مفاهیم
- طراحی
- پیاده‌سازی
- ارزیابی
- نتیجه

- پیش‌پردازش
- محاسبه شیب
- ساخت هیستوگرام از شیب
- نرمال‌سازی بلوک‌ها
- محاسبه بردار ویژگی

معرفی ابزارها و فناوری‌های استفاده شده در پروژه

• ابزارهای سخت‌افزاری

- بورد **Raspberry Pi** که یکی از SBC‌های پر قدرت و پرفروش بازار است و در این پروژه از نسخه 4B با RAM 1GB آن استفاده شده است.



• ماژول دوربین



• قفل solenoid



معرفی ابزارها و فناوری‌های استفاده شده در پروژه



• ابزارهای سخت‌افزاری

- صفحه کلید عددی
- ماژول **SIM900** که برای ارسال پیامک استفاده می‌شود
- آژیر
- دیود نشرده‌نده نور (LED)

معرفی ابزارها و فناوری‌های استفاده شده در پروژه

• ابزارهای نرم‌افزاری

- زبان برنامه‌نویسی پایتون: زبانی شیء‌گرا و خوانا دارای کتابخانه‌های مناسب کار در حوزه هوش مصنوعی



• کتابخانه‌های مورد استفاده:

• OpenCV: اعمال تغییرات بر روی تصویر ثبت شده از دوربین سامانه



• dlib: ابزار قوی برای کاربردی‌هایی از جمله پردازش تصویر، یادگیری ماشین و جبر خطی



• NumPy: مجموعه‌ای از توابع ریاضی سطح بالا

• matplotlib: رسم نمودار



• RPi.GPIO: کنترل پایه‌های RPi.GPIO

• Flask: پیاده‌سازی تابلوی کنترلی مدیران سامانه



طراحی سامانه

- معرفی مفهوم کاربر
- طراحی معماری سامانه



معرفی مفهوم کاربر

• تعریف

- هر شخص که قرار است مقابله دوربین سامانه قرار بگیرد و شناسایی شود

• اطلاعات هر کاربر

- شناسه کاربری
- نام (اختیاری)
- تصاویر ذخیره شده

مقدمه

مفاهیم

طراحی

پیادهسازی

ارزیابی

نتیجه

معرفی مفهوم کاربر

• انواع کاربر

- **کاربر ناشناس:** کاربری که چهره او در سامانه تعریف نشده است
- **کاربر عادی:** کاربری که می‌تواند به کمک چهره خود، قفل سامانه را باز کند
- **کاربر مدیر:** کاربری که علاوه بر باز کردن قفل، می‌تواند کاربر جدید به سامانه اضافه کند و یا کاربری را از آن حذف نماید

مقدمه

مفاهیم

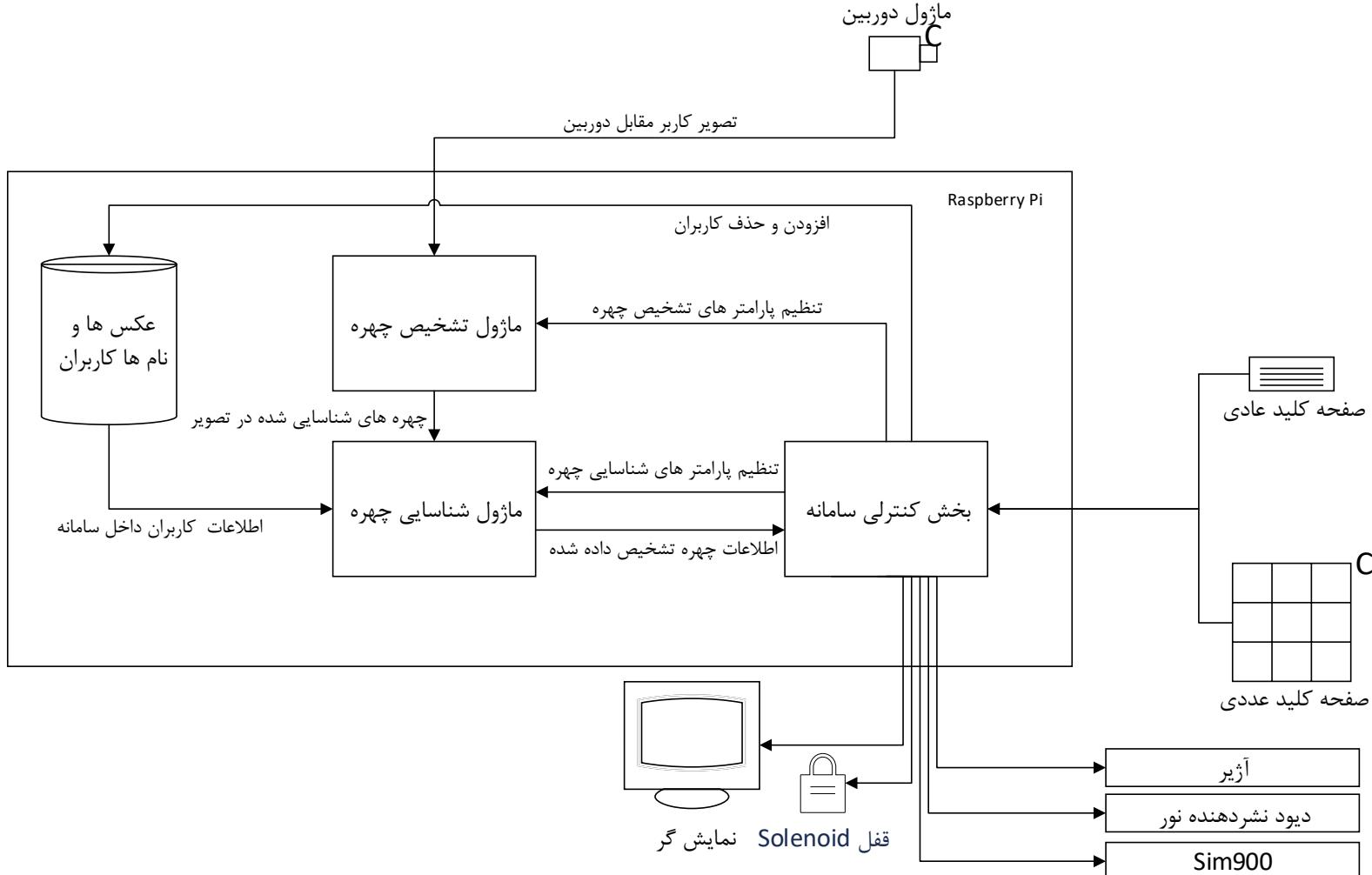
طراحی

پیاده‌سازی

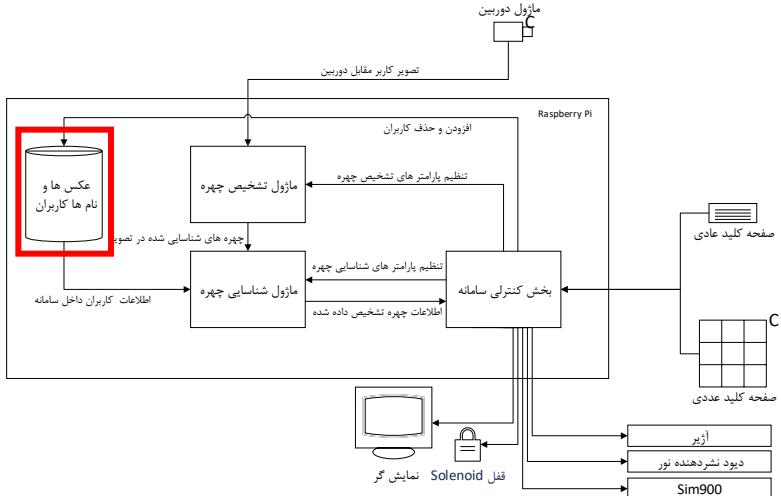
ارزیابی

نتیجه

طراحی معماری سامانه



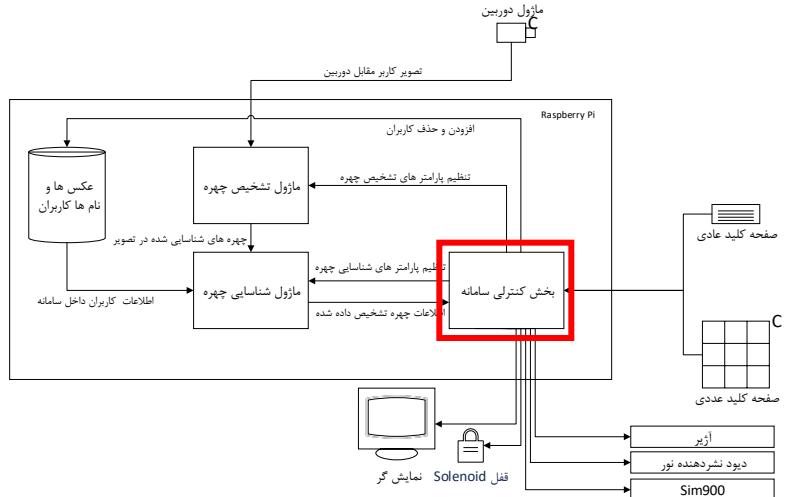
طراحی معماری سامانه



• مجموعه داده‌ها

- عکس و نام کاربران
- در اختیار قرار دادن این اطلاعات به بخش شناسایی چهره
- تغییر داده‌ها توسط بخش کنترلی سامانه

طراحی معماری سامانه



• بخش کنترلی سامانه

- تغییر پارامترهای سامانه به کمک فایل پیکربندی
- احراز هویت با کمک ماژول شناسایی چهره
- گرفتن ورودی از کاربر
- کنترل قطعات سخت افزاری

طراحی معماری سامانه

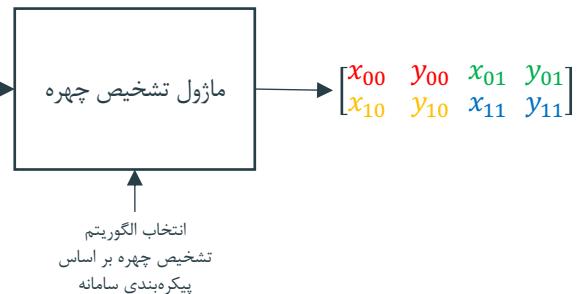
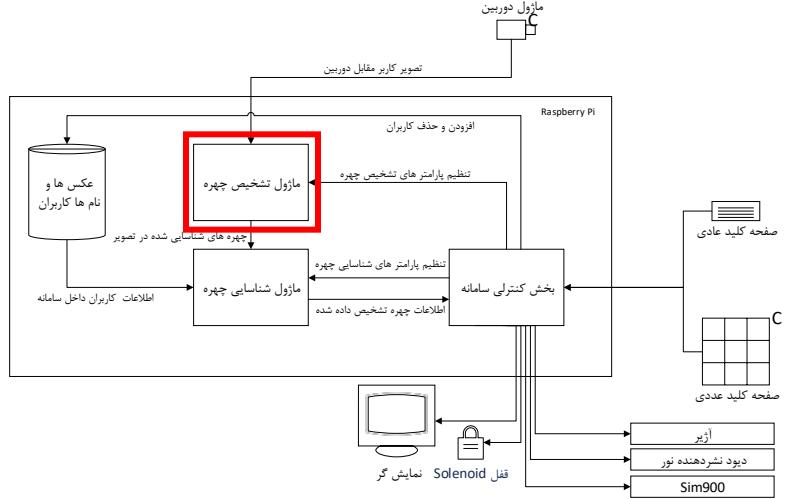
بخش تشخیص چهره

- استفاده از یکی از دو الگوریتم معروفی شده

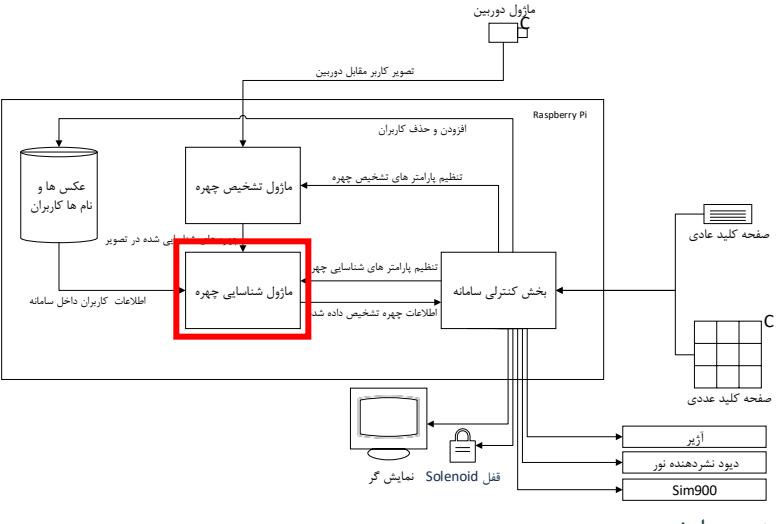
- شبکه عصبی در هم تنیده

- هیستوگرام شبکه‌های جهت‌دار

- خروجی: مختصات دو گوشه مستطیل در برگیرنده چهره‌ها



طراحی معماری سامانه

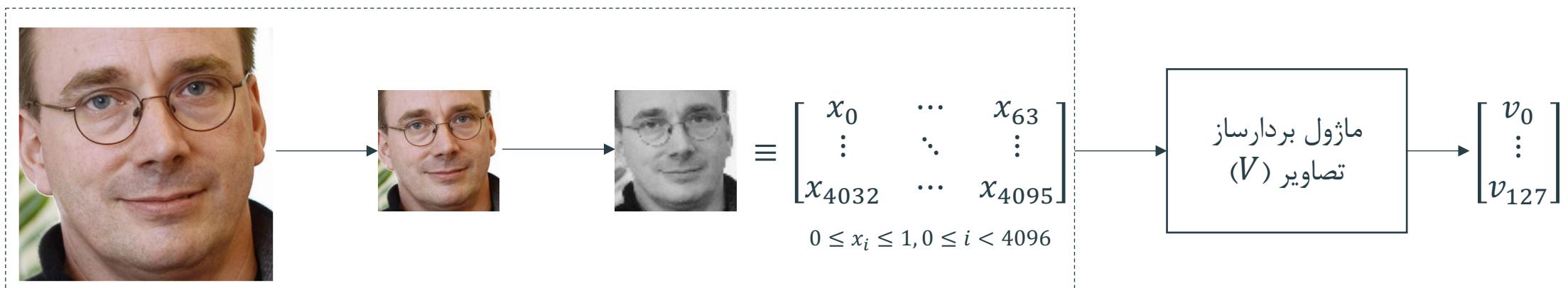


بخش شناسایی چهره

پیش پردازش تصاویر

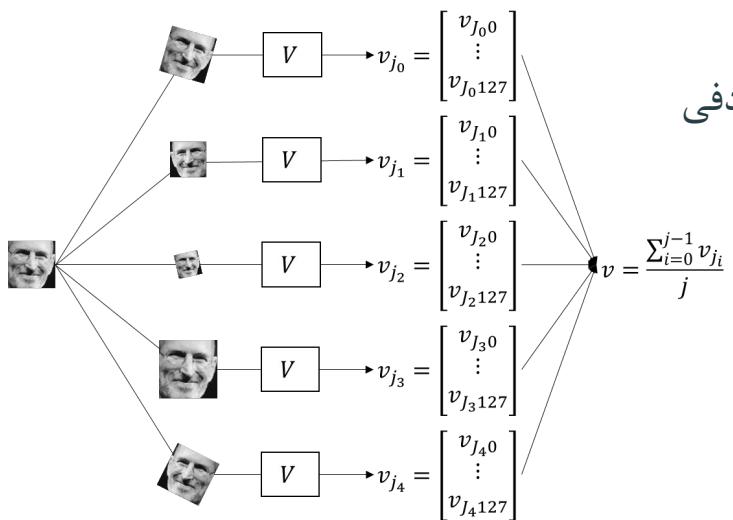
- حذف مولفه رنگ و تغییر ابعاد عکسها

- تبدیل ماتریس چهره پیش پردازش شده به بردار ($V(img)$)



طراحی معماری سامانه

$$v' = \frac{\sum_{i=0}^{j-1} (R(V(img)))}{j}$$



• بخش شناسایی چهره

• نمونهبرداری مجدد از تصویر برای بهبود کیفیت شناسایی چهره

• استفاده از پارامتر j

• اعمال چرخش، جابهجایی، برگردانده شدن و بزرگنمایی به صورت تصادفی

(به کمک تابع R) روی تصویر ورودی به تعداد j دفعه

• میانگینگیری از بردارهای خروجی هر عکس به عنوان بردار نهایی

طراحی معماری سامانه

• بخش شناسایی چهره

$$v' = V(P(img, \gamma))$$

مقدمه

مفاهیم

طراحی

پیادهسازی

ارزیابی

نتیجه

• شناسایی چهره‌های کوچک‌تر

• ضعف مدل‌های شناسایی چهره در تشخیص چهره‌های کوچک

• راه حل: بزرگ‌نمایی

• استفاده از پارامتر γ

• بزرگ‌نمایی عکس ورودی به کمک تابع P به اندازه γ

• تشخیص مجدد چهره‌ها از روی عکس بزرگ‌نمایی شده

طراحی معماری سامانه

• بخش شناسایی چهره

- معیار اختلاف میان دو چهره: فاصله بردارهای دو چهره

- محاسبه نرم-1 اختلاف بردارها به کمک تابع D

- معیار تطابق دو چهره

- استفاده از پارامتر t

- حد آستانه فاصله بردارهای متناظر با چهرهها

- $isRecognized$ شناسایی به کمک تابع

$$\|W\|_1 = \sum_i^n |w_i|$$
$$D(X, Y) = \|X - Y\|$$

$$isRecognized(i) = \begin{cases} u, & \text{if } \min_{u \in U} D(V(i), V_u) < t \\ none, & \text{if } \min_{u \in U} D(V(i), V_u) > t \end{cases}$$

خلاصه فرایند شناسایی چهره

• خلاصه فرایند شناسایی چهره



مقدمه

مفاهیم

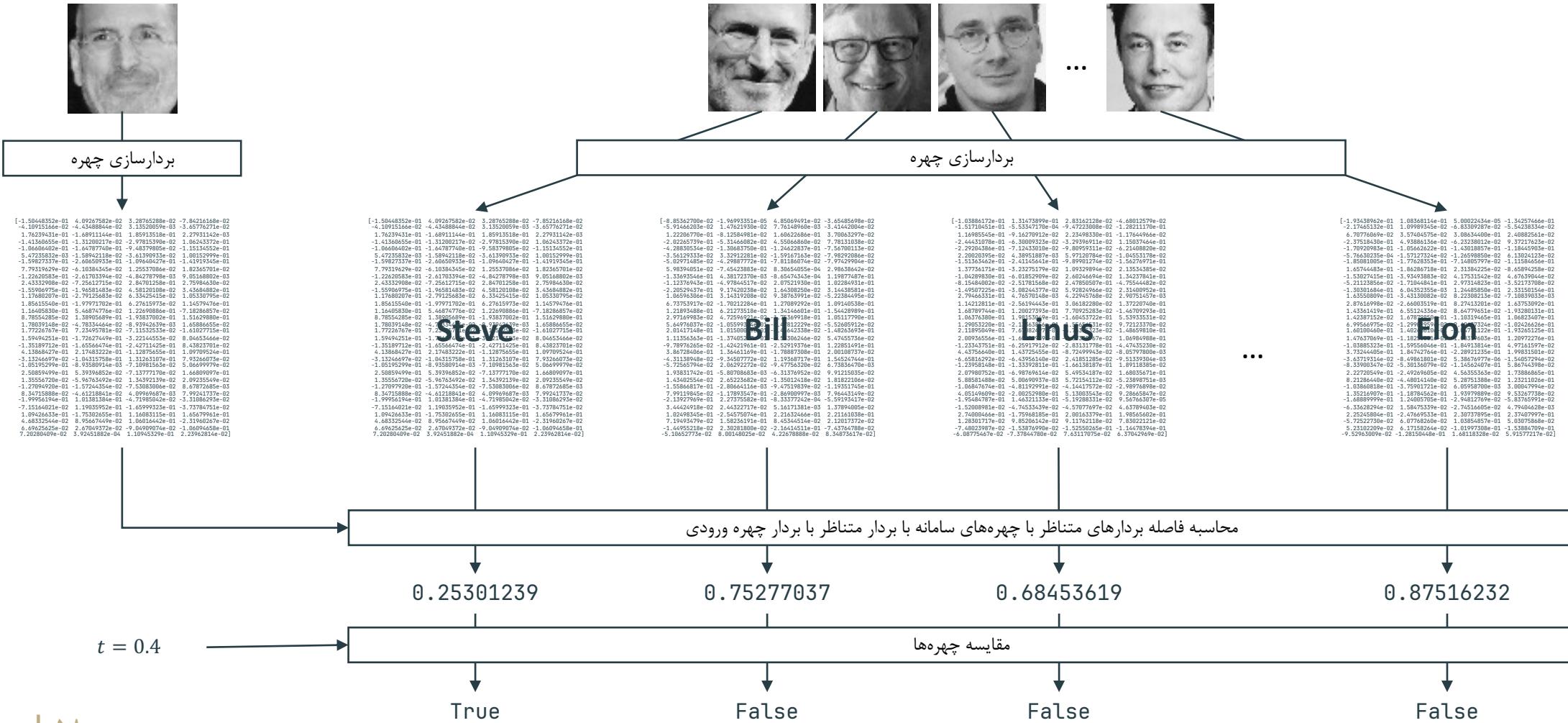
طراحی

پیادهسازی

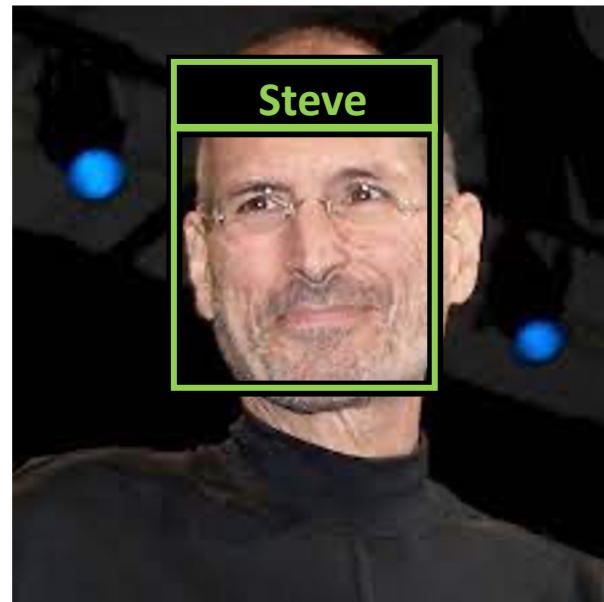
ارزیابی

نتیجه

خلاصه فرایند شناسایی چهره



خلاصه فرایند شناسایی چهره



مقدمه

مفاهیم

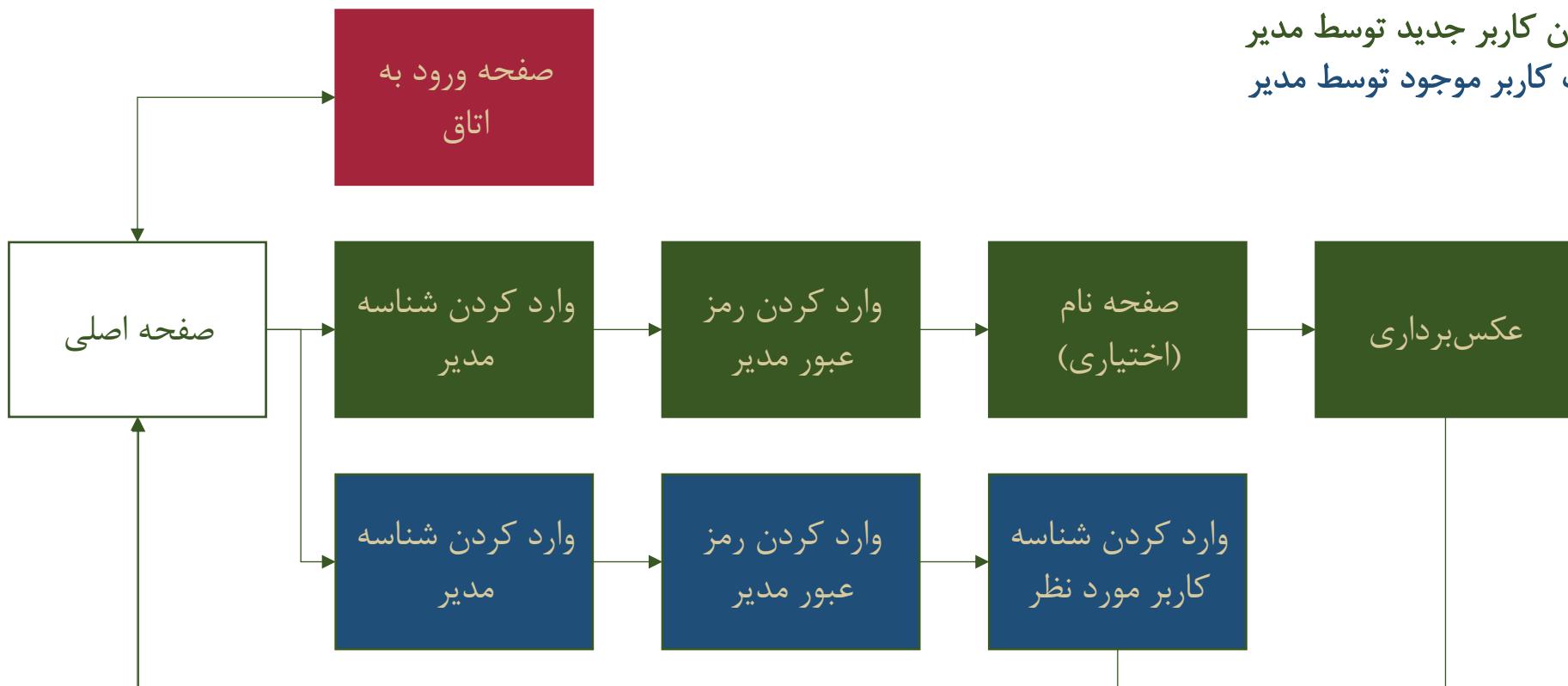
طراحی

پیادهسازی

ارزیابی

نتیجه

طراحی جریان اجرایی



- سناریوی ورود
- سناریوی افزودن کاربر جدید توسط مدیر
- سناریوی حذف کاربر موجود توسط مدیر

مقدمه

مفهوم

طراحی

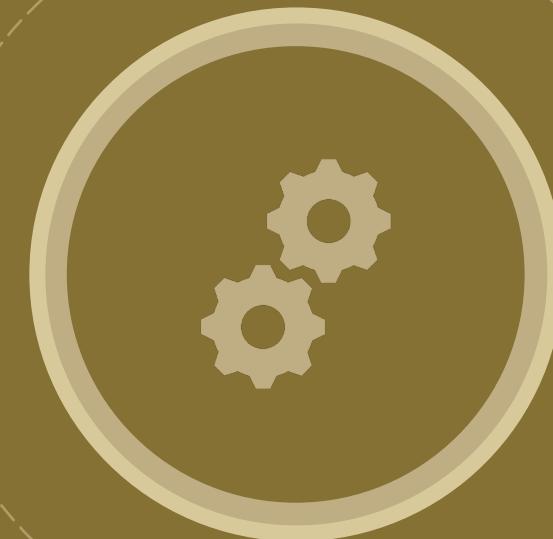
پیاده‌سازی

ارزیابی

نتیجه

پیاده‌سازی

پیاده‌سازی نرم‌افزاری و سخت‌افزاری



پیاده‌سازی

• پیاده‌سازی بخش تشخیص چهره

- امکان استفاده از دو الگوریتم با توجه به فایل پیکربندی

HOG: مدل از پیش تمرین داده شدهی haarcascade_frontalface_default.xml توسط OpenCV

CNN: کتابخانه MTCNN که به کمک کتابخانه TensorFlow امکان ساخت شبکه عصبی پیچشی را فراهم می‌سازد

• پیاده‌سازی بخش شناسایی چهره

- پیاده‌سازی کد تبدیل چهره‌ها به بردار

استفاده از تابع "compute_face_descriptior" از کتابخانه dlib

```
def face_encodings(face_image, known_face_locations=None, num_jitters=1):
    raw_landmarks = _raw_face_landmarks(face_image, known_face_locations)
    return [
        np.array(
            face_encoder.compute_face_descriptor(face_image, raw_landmark_set, num_jitters)
        ) for raw_landmark_set in raw_landmarks
    ]
```

پیاده‌سازی

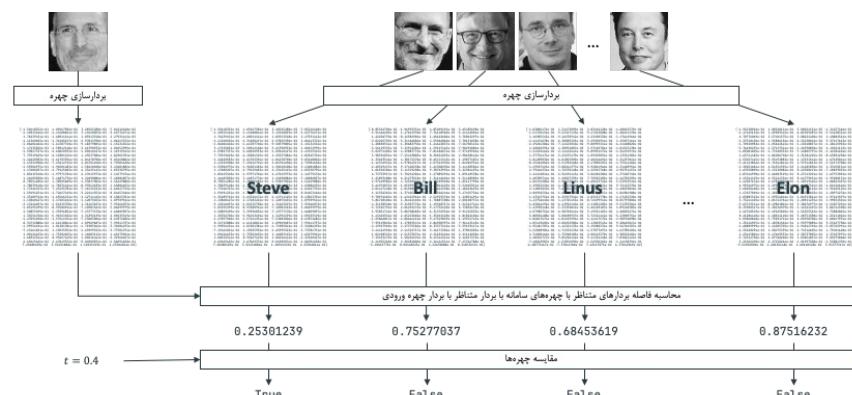
• پیاده‌سازی بخش شناسایی چهره

```
def face_distance(_face_encodings, face_to_compare):  
  
    if len(_face_encodings) == 0:  
  
        return np.empty(0)  
  
    return np.linalg.norm(_face_encodings - face_to_compare, axis=1)
```

- محاسبه فاصله چهره‌ها

```
def compare_faces(known_face_encodings, face_encoding_to_check, tolerance=0.6):  
  
    return list(face_distance(known_face_encodings, face_encoding_to_check) <= tolerance)
```

- مقایسه چهره‌ها



پیاده‌سازی

• پیاده‌سازی جریان اجرایی

- مقدمه
- مفاهیم
- طراحی
- پیاده‌سازی
- ارزیابی
- نتیجه

```
python3 main.py
26 encoding images found.
100% | 26/26 [00:00<00:00, 46.68it/s]
Model was trained in 0.5807s.
```



پیاده‌سازی

• پیاده‌سازی جریان اجرایی

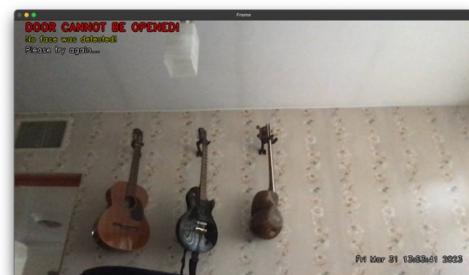
• ورود به اتاق



ورود موفق



ورود ناموفق (چهره ناشناس)



ورود ناموفق (عدم تشخیص چهره)

مقدمه

مفاهیم

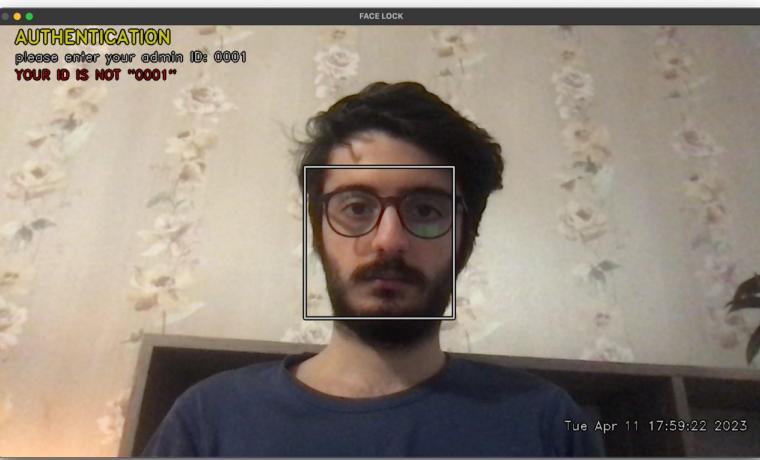
طراحی

پیاده‌سازی

ارزیابی

نتیجه

پیاده‌سازی



- پیاده‌سازی جریان اجرایی

- افزودن کاربر جدید

- وارد کردن شناسه مدیر

پیاده‌سازی



وارد کردن رمز صحیح



وارد کردن رمز نادرست

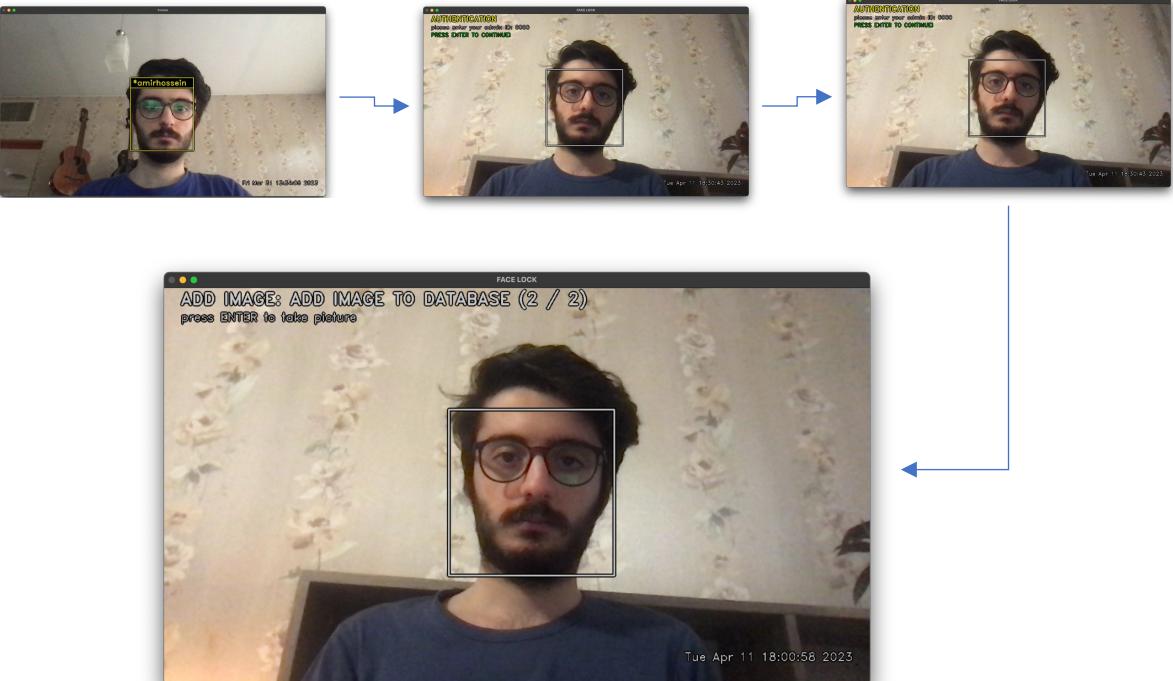
• پیاده‌سازی جریان اجرایی

• افزودن کاربر جدید

• وارد کردن شناسه مدیر

• وارد کردن رمز عبور

پیاده‌سازی



مقدمه

مفاهیم

طراحی

پیاده‌سازی

ارزیابی

نتیجه

• پیاده‌سازی جریان اجرایی

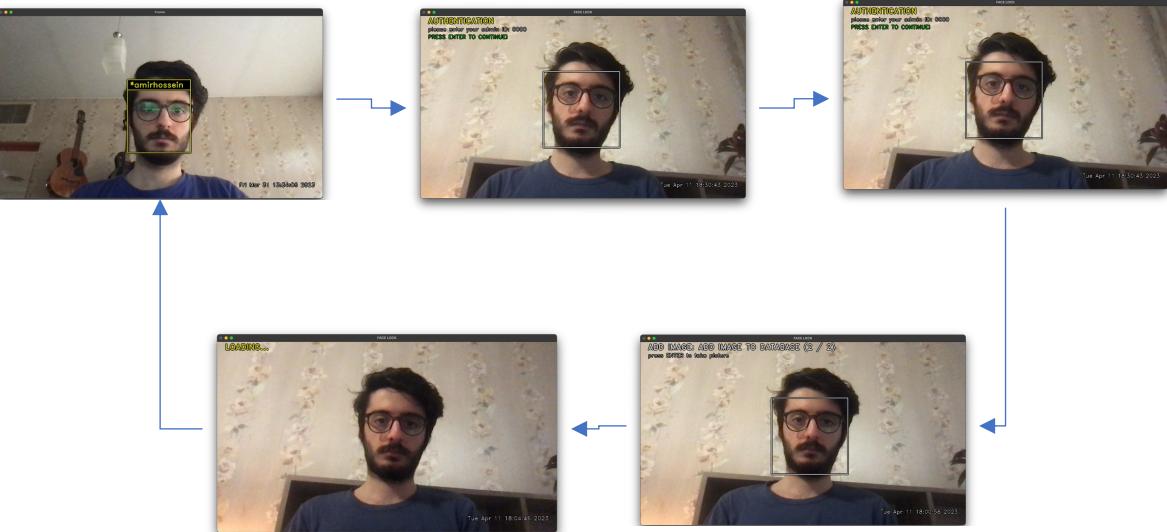
• افزودن کاربر جدید

• وارد کردن شناسه مدیر

• وارد کردن رمز عبور

• عکسبرداری از کاربر جدید

پیاده‌سازی



• پیاده‌سازی جریان اجرایی

• افزودن کاربر جدید

- وارد کردن شناسه مدیر

- وارد کردن رمز عبور

- عکسبرداری از کاربر جدید

- ساخت کاربر جدید و تمرین مدل

شناسایی چهره

پیاده‌سازی

• پیاده‌سازی جریان اجرایی

- حذف کاربر موجود
- افزودن کاربر جدید
- وارد کردن شناسه مدیر
- وارد کردن رمز عبور
- وارد کردن رمز عبور
- عکسبرداری از کاربر جدید
- ساخت کاربر جدید و تمرین مدل
- شناسایی چهره
- مدل شناسایی چهره
- حذف کاربر موجود و تمرین
- وارد کردن شناسه کاربر
- وارد کردن رمز عبور
- حذف کاربر موجود

مقدمه

مفاهیم

طراحی

پیاده‌سازی

ارزیابی

نتیجه

پیاده‌سازی

• ثبت و قایع برنامه به کمک ذخیره

رخدادها

```
1681029714: ADMIN ENTERED USER ID "0000"  
1681029714: FACE FOR ID "0000" WAS NOT DETECTED  
1681029718: ADMIN ENTERED USER ID "0000"  
1681029718: FACE FOR ID "0000" WAS NOT DETECTED  
1681029721: ADMIN ENTERED USER ID "0000"  
1681029721: FACE FOR ID "0000" WAS NOT DETECTED  
1681029773: ADDING USER...  
1681031304: ADDING USER...
```

- برای پیگیری راحت‌تر ورود و خروج و همچنین افزودن و حذف کاربران توسط مدیرها، می‌توان از ذخیره رخداد برای تمام اتفاق‌ها مربوط به سامانه استفاده کرد.
- باز کردن موفق در توسط کاربر شناخته شده توسط سامانه
- باز نشدن در پس از اقدام ناموفق کاربر ناشناس
- اقدام برای ورود به سیستم به عنوان مدیر
- وارد کردن رمز عبور اشتباه توسط مدیر
- افزودن کاربر توسط مدیر
- حذف کاربر توسط مدیر

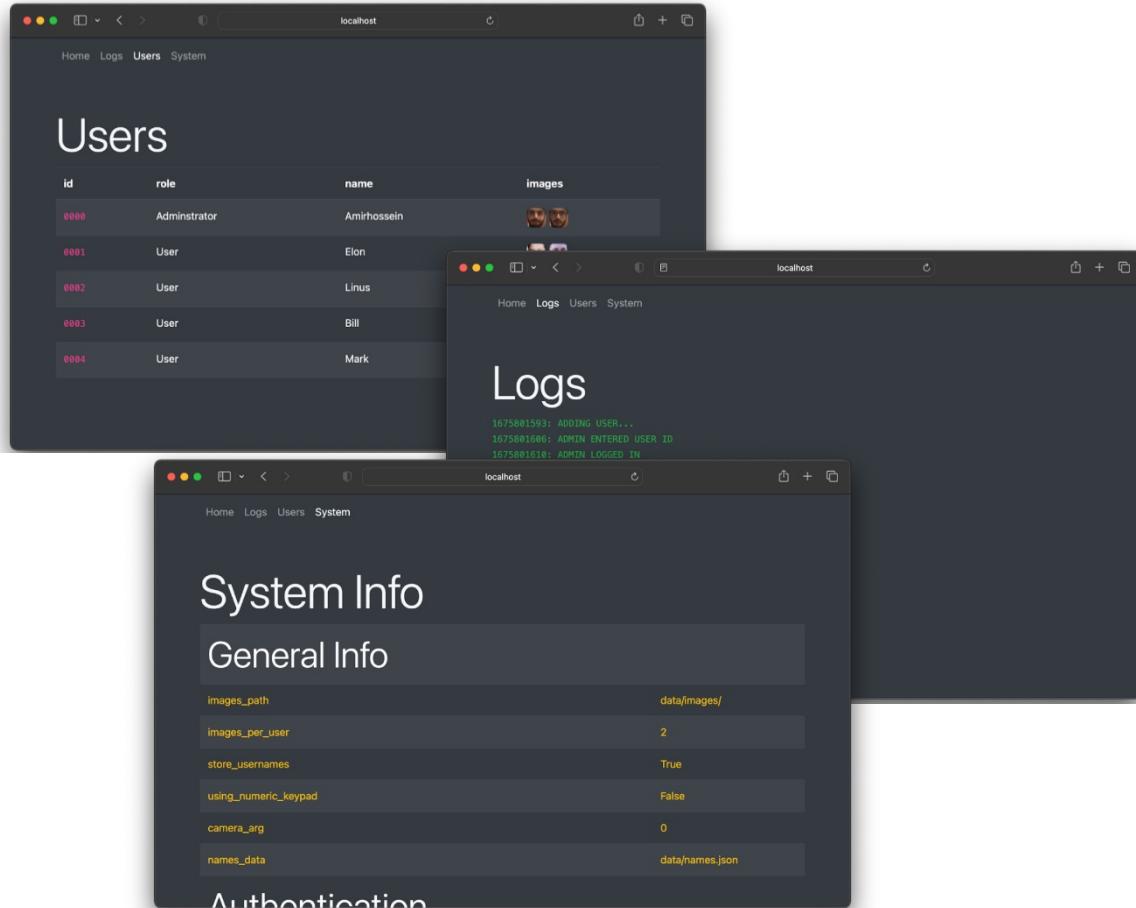
پیاده‌سازی

• پیکربندی سامانه



- مقدمه
- مفاهیم
- طراحی
- پیاده‌سازی
- ارزیابی
- نتیجه

پیاده‌سازی



• توسعه تابلوی کنترلی مدیر

- مشاهده راحت‌تر اطلاعات موجود در

سامانه

• اجرا به کمک دستور flask run

(آی‌پی ۱۲۷.۰.۰.۱ و درگاه ۵۰۰۰)

• مشاهده‌ی:

لیست کاربران

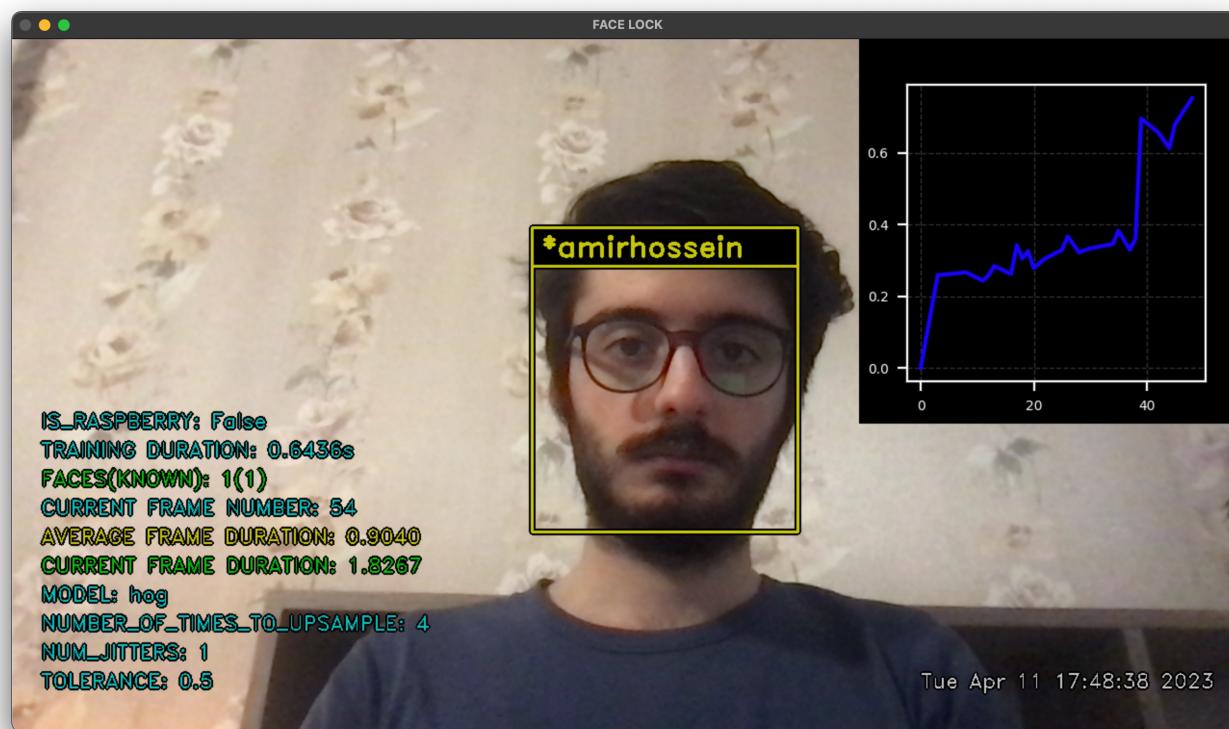
رخدادهای سامانه

تنظیمات پیکربندی

پیاده‌سازی

• حالت اشکال‌زدایی

python main.py -debug true •



مقدمه

مفاهیم

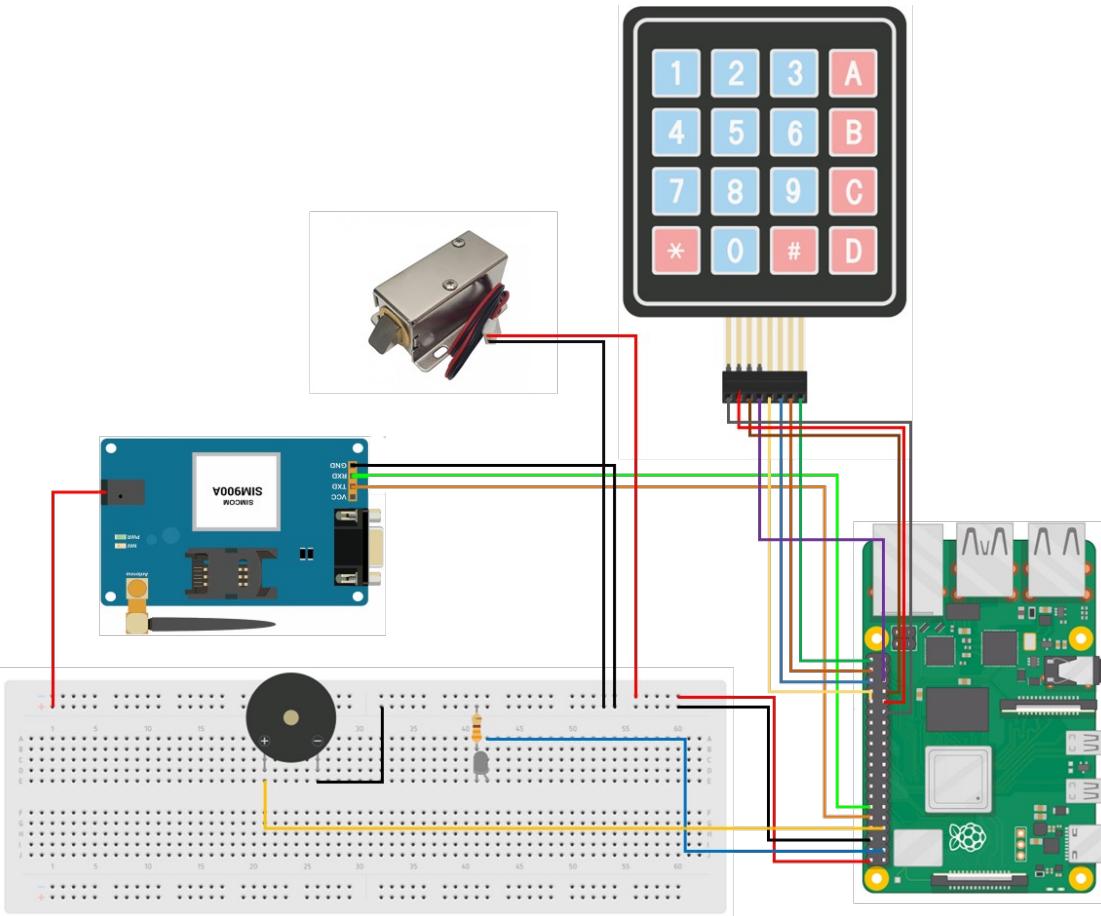
طراحی

پیاده‌سازی

ارزیابی

نتیجه

پیاده‌سازی



• پیاده‌سازی سخت‌افزار

- مدار پروژه
- راهاندازی سیستم عامل Raspbian
- کنترل قطعات به کمک کتابخانه RPi.GPIO

ارزیابی

- ارزیابی به کمک مجموعه داده واقعی
- مقایسه الگوریتم های تشخیص



ارزیابی

- استفاده از تحلیل مولفه‌های اصلی (PCA) برای بررسی

کیفیت شناسایی چهره

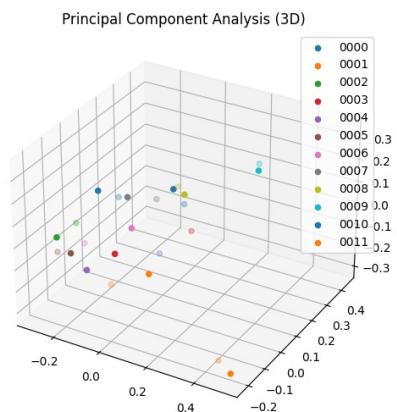
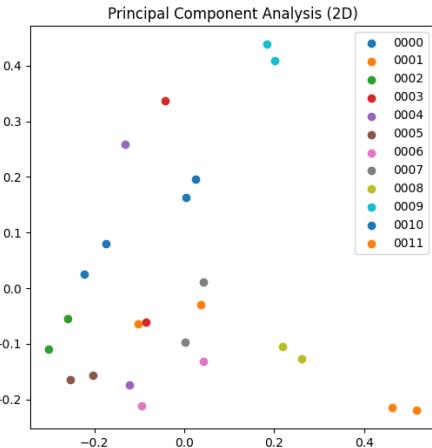
- تبديلی در فضای برداری به منظور کاهش بعد برای تحلیل و تفسیر

داده‌های بزرگ و با تعداد بعد بالا با حفظ حداقل اطلاعات

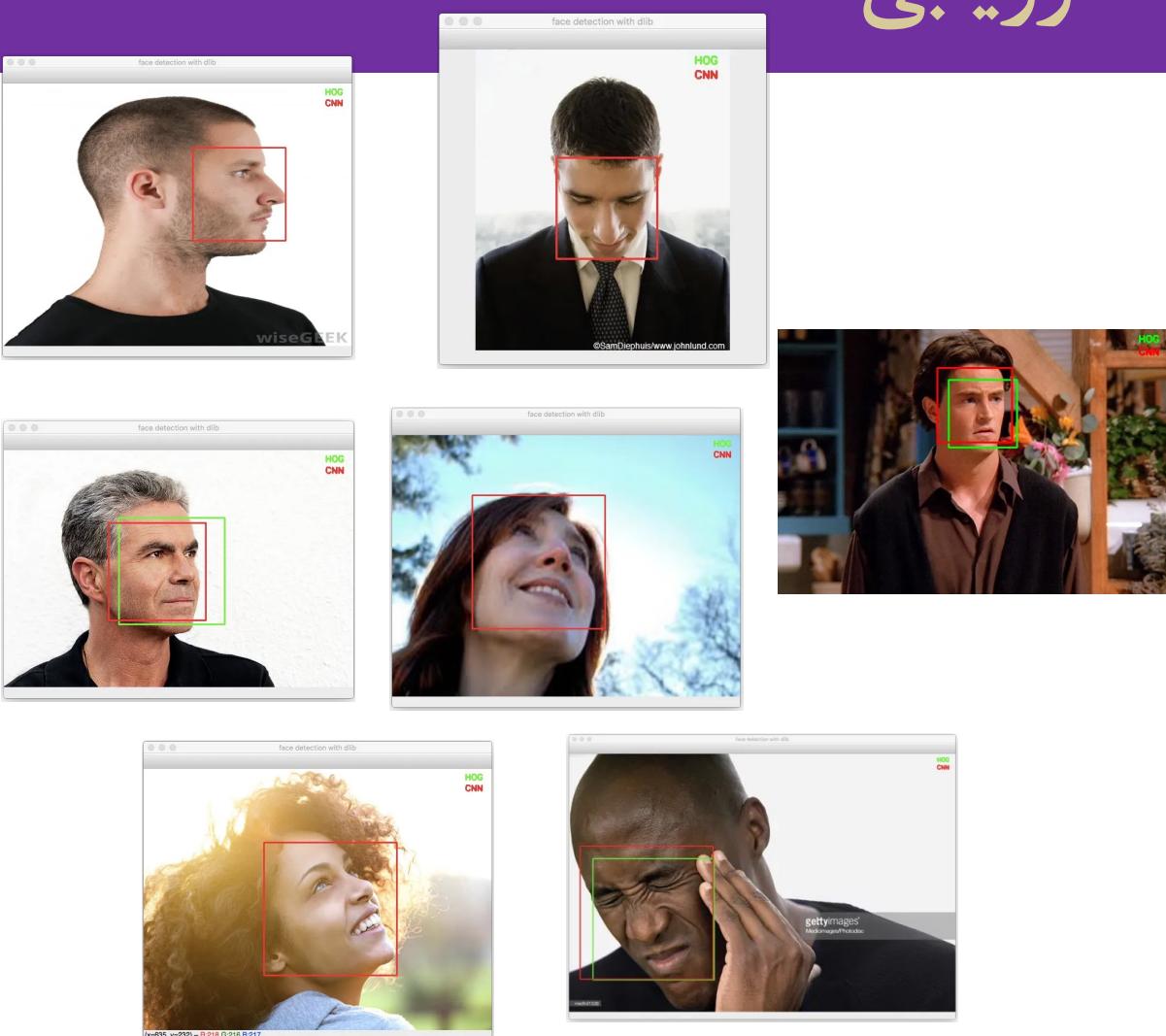
- هدف استفاده: مصورسازی بردارهای متناظر با چهره‌های سامانه

- نگاشت بردارها از فضای ۱۲۸بعدی به فضای سه‌بعدی

چهره‌های یک شخص: نقاط نزدیک هم در این فضا



ارزیابی



• مقایسه CNN و HOG •

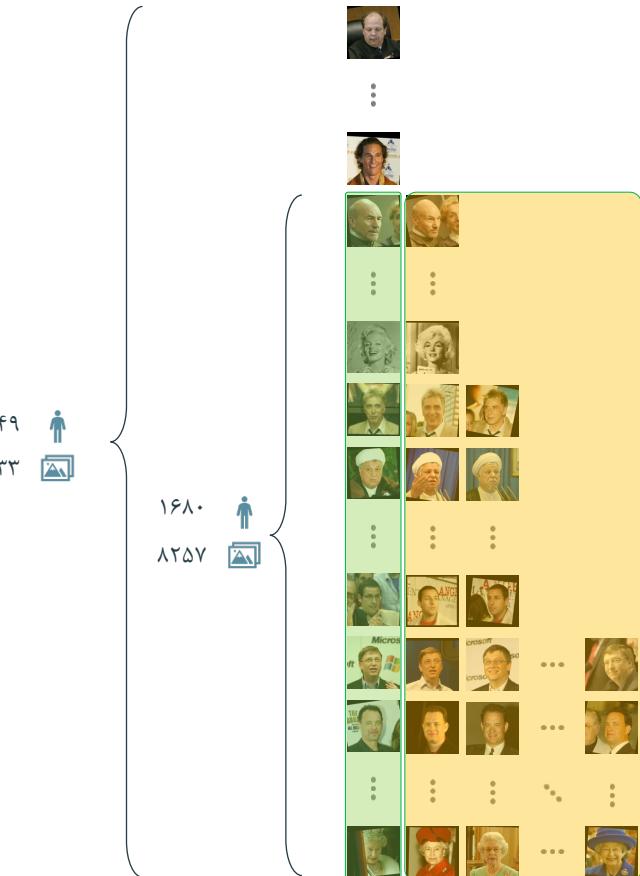
CNN •

- تشخیص چهره قوی‌تر در حالتی که چهره دقیقاً رو به دوربین نیست
- الگوریتم سنگین‌تر

HOG •

- مناسب تشخیص چهره‌هایی که چهره رو به دوربین است
- الگوریتم سبک‌تر ← مناسب‌تر برای اجرا روی Raspberry Pi

ارزیابی



• بررسی دقیق شناسایی چهره

- مجموعه داده: "چهره‌های برچسب دار در دنیا واقعی"

- ارائه شده توسط آزمایشگاه بینایی دانشگاه ماساچوست

- مجموعه تمرین: ۱۶۸۰ چهره متعلق به ۱۶۸۰ شخص متمایز

- مجموعه تست: ۶۵۷۷ تصویر جدید از چهره‌های موجود در مجموعه تمرین

- فرایند آزمایش:

- تمرین مدل شناسایی چهره به تصاویر مجموعه تمرین (استفاده از HOG، t , $\gamma = 1$ و $j = 1$, $\gamma = 0.6$)

- محاسبه خروجی مدل برای تصاویر مجموعه تست و مقایسه با مالک واقعی عکس

ارزیابی

- بررسی دقت شناسایی چهره

- ماتریس سردرگمی (confusion matrix)

		خروچی مدل شناسایی چهره		
		6577	$P'(6346)$	$N'(231)$
متاد بنویسی	$P(6566)$	$TP(6341)$	$FN(225)$	
	$N(11)$	$FP(5)$	$TN(6)$	

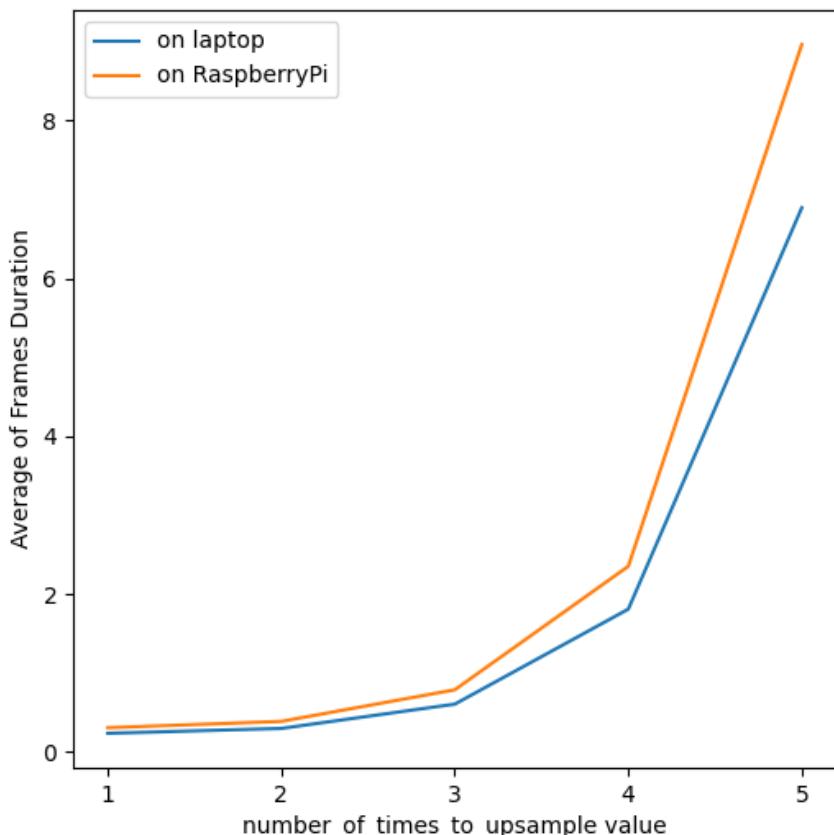
$$Accuracy = \frac{TP + TN}{P + N} = \frac{6341 + 6}{6577} = \frac{6347}{6577} = 96.502\%$$

$$Precision = \frac{TP}{TP + FP} = \frac{6341}{6341 + 5} = \frac{6341}{6346} = 99.921\%$$

$$F1 Score = \frac{2TP}{2TP + FP + FN} = \frac{2 \times 6341}{2 \times 6341 + 5 + 225} = \frac{12682}{12312} = 98.218\%$$

ارزیابی

Effect of number_of_times_to_upsample on Frame Duration



- تاثیر مقدار γ در مدت زمان هر قاب

- افزایش γ :

- تشخیص بهتر چهره‌های کوچکتر

- مشاهده پرش در تصویر به علت افزایش زمان هر قاب

- پیشنهاد می‌شود این مقدار در فایل پیکربندی بیشتر

- از عدد "۲" نباشد

نتیجه‌گیری

مرورو و نتیجه‌گیری

پیشنهادات



نتیجه

• جمع‌بندی و نتیجه‌گیری

• بررسی دو الگوریتم تشخیص چهره:

- CNN: دقت بالا در تشخیص چهره‌هایی که با زاویه مقابله دارند + سربار محاسباتی زیاد
- HOG: تشخیص مناسب چهره‌هایی که مستقیماً به دوربین نگاه می‌کنند + سربار کم و مناسب بورد Raspberry Pi
- ارزیابی شناسایی چهره به کمک مجموعه داده "چهره‌های برچسب دار در دنیای واقعی"
- صحت: ۹۶/۵۰۲ درصد
- دقت: ۹۹/۹۲۱ درصد
- امتیاز F1: ۹۸/۲۱۸ درصد
- تنظیم پارامترها
- زیاد کردن پارامترهایی که به دقت سامانه کمک می‌کنند → پرش بیشتر در تصویر
- باید از زیاد کردن بیش از حد آن‌ها جلوگیری کرد

نتیجه

- پیشنهادات و کارهای آینده

- شناسایی FP

- بهینه‌سازی فرایند حذف و افزودن کاربران
- عدم بررسی تمام عکس‌ها که منجر به سریع‌تر شدن فرایند تمرین مدل شناسایی چهره می‌شود
- تغییر تنظیمات پیکربندی در تابلوی کنترلی مدیر
- اجرای سامانه در صورتی که شخصی مقابله آن قرار دارد

- احراز هویت به کمک عکس شخص عضو سامانه در تلفن همراه

- راه حل پیشنهادی: فاصله + محدودیت در اندازه قاب تشخیص داده شده

- استفاده از سایر مدل‌های Raspberry Pi

- در این پروژه از RAM 1GB استفاده شده که در صورت زیاد شدن تصاویر سامانه ممکن است فرایند کند شود.

مراجع

- [1] B. T. V. Bhandiwad, "Face recognition and detection using neural networks," International Conference on Trends in Electronics and Informatics (ICEI), p. 879–882, 2017.
- [2] M. J. S. L.-J. L. S. S. Farfade, "Multi-view Face Detection Using Deep Convolutional Neural Networks," Shanghai China, 2015, p. 643–650.
- [3] H. A. B. M. H. Ahmad Fadzil, "Human face recognition using neural networks," 1st International Conference on Image Processing, vol. 3, p. 936–939, 1994.
- [4] D.-S. H.-Y. S. Z.-Q. Zhao, "Human face recognition based on multi-features using neural networks committee," Pattern Recognition Letters, vol. 25, p. 1351–1358, 2004.
- [5] T. H. Le, "Applying Artificial Neural Networks for Face Recognition," Advances in Artificial Neural Systems, 2011.
- [6] P. Latha, "Face Recognition using Neural Network," Signal Processing, p. 8.
- [7] T. i, "Face Detection vs. Facial Recognition – What's the Difference?," Two i, 2022. [Online]. Available: <https://www.two-i.com/blog/face-detection-vs-facial-recognition-whats-the-difference>. [Accessed 2023].
- [8] NEC, "Face Detection vs Facial Recognition – what's the difference?," NEC, 1 June 2022. [Online]. Available: <https://www.nec.co.nz/market-leadership/publications-media/face-detection-vs-facial-recognition-whats-the-difference/>. [Accessed 2023].
- [9] IBM, "Introduction to Neural Networks," IBM, 2021. [Online]. Available: <https://www.ibm.com/topics/neural-networks>. [Accessed 2023].
- [10] A. Rosebrock, "Face detection with dlib," pyimagesearch, 2021 April 2021. [Online]. Available: <https://pyimagesearch.com/2021/04/19/face-detection-with-dlib-hog-and-cnn/>. [Accessed 2023].

مراجع

- [11] R. Pi, "Raspberry Pi Documentation," Raspberry Pi, 2022. [Online]. Available: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>. [Accessed 2023].
- [12] R. Pi, "About the Camera Modules," Raspberry Pi, 2022. [Online]. Available: <https://www.raspberrypi.com/documentation/accessories/camera.html>. [Accessed 2023].
- [13] A. B. Downey, Think Python: How to Think Like a Computer Scientist, O'Reilly, 2020.
- [14] OpenCV, "OpenCV website," OpenCV, [Online]. Available: <https://opencv.org/>. [Accessed 2023].
- [15] Dlib, "Dlib Web Page," Dlib, [Online]. Available: <http://dlib.net>. [Accessed 2023].
- [16] NumPy, "NumPy Webpage," NumPy, [Online]. Available: <https://numpy.org>. [Accessed 2023].
- [17] MatPlotLib, "MatPlotLib Web Page," MatPlotLib, [Online]. Available: <https://matplotlib.org>. [Accessed 2023].
- [18] Flask, "Flask Web Framework," Flask, [Online]. Available: <https://Flask.palletsprojects.com/en/2.2.x>. [Accessed 2023].
- [19] AWS, "What is an IDE?," Amazon, [Online]. Available: <https://aws.amazon.com/what-is/ide/>. [Accessed 2023].
- [20] "PyCharm Guide," RealPython, [Online]. Available: <https://realpython.com/pycharm-guide/>. [Accessed 2023].
- [21] OpenCV, "HOG pre-trained model," OpenCV, [Online]. Available: <https://github.com/opencv/opencv/tree/master/data/haarcascades>. [Accessed 2023].
- [22] Z. Z. Z. L. Y. Q. K Zhang, "Face Detection and alignment using multitask cascaded convolutional networks," *IEEE*, 2016.
- [23] U. o. Massachusetts, "Labeled Faces in the Wild," University of Massachusetts, 2018. [Online]. Available: <http://vis-www.cs.umass.edu/lfw/>. [Accessed 2023].

با تشکر از توجه شما

✉ amirhosseinalibakhshi@gmail.com