



دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

پروژه درس آزمایشگاه سخت افزار

عنوان:

**مجموعه مستندات آینه هوشمند**

نگارندگان:

آریا جلالی، محمد هجری، امیرحسین باقری

استاد گرامی:

جناب آقای دکتر اجلالی – جناب آقای دکتر فصحتی

تابستان 1401

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

## فهرست

4	مقدمه
4	دیتاشیت محصول
6	معماری سیستم
7	طراحی و پیاده سازی سخت افزار
7	LCD
8	دوربین
9	نکاتی درباره دوربین
9	پیاده سازی کد سرور در رزبری پای
10	مشکلات ضبط و استریم با هم
11	راهکار و پیاده سازی نهایی
12	دلایل عدم استفاده از پروتکل های بالا به صورت کلی
13	مقایسه راهکار های گفته شده
14	طراحی و پیاده سازی اپلیکیشن موبایل
18	امکانات اپلیکیشن نوشته شده
18	ذخیره ی اتوماتیک:
18	ذخیره ی استریم به مدت زمان های پیش فرض:
19	اجرای سرور در رزبری پای
19	قیمت:
20	نماهایی از محصول
26	جمع بندی:

# فصل 1

## مقدمه

هدف از این پروژه، ارائه محصولی جهت حفظ امنیت خودرو و اشتراک گذاری محتوا در شبکه‌های اجتماعی می‌باشد. پیاده‌سازی این پروژه به صورت کلی و توانایی استفاده از آن در انواع مختلفی از ماشین‌ها وجود دارد.

محصول توانایی پخش تصویر پشت ماشین به منظور کمک به راننده در حین رانندگی و اشتراک گذاری ویدیوهای گرفته شده از پشت ماشین در هنگام سفر یا مقاصد دیگری که نیاز به اشتراک گذاری ویدئو در شبکه‌های اجتماعی دارند را دارد، همچنین در پیشگیری از مخاطرات ناشی از عدم دید کافی راننده به پشت خود می‌تواند موثر باشد.

ویدئوهای ضبط شده توسط این محصول در یک سرور جانبی ذخیره می‌گردند، و کاربر می‌تواند به مقاصد مختلفی از آن‌ها استفاده کند.

# فصل 2

## دیتاشیت محصول

کیفیت دوربین	5 مگاپیکسل
فریم بر ثانیه	30 فریم در کیفیت (1080p Full HD) 60 فریم در کیفیت (720p HD) 90 فریم در کیفیت 480p
بهترین کیفیت عکس	2592 × 1944 پیکسل
جرم دوربین به گرم	9.97
ابعاد دوربین به اینچ	1.57×1.18×0.79
فرکانس پردازنده به مگاهرتز	300 تا 400
ابعاد پردازنده به میلیمتر	85× 56× 17
جریان ورودی	1.5 آمپر
ولتاژ ورودی	5 ولت
جرم پردازنده به گرم	50
تعداد پورت USB	4
تعداد Ethernet port	1
WIFI	✓
Bluetooth	✓
کابل HDMI	✓
سیستم عامل	Raspbian OS
LCD	✗

جدول 1 دیتاشیت سیستم

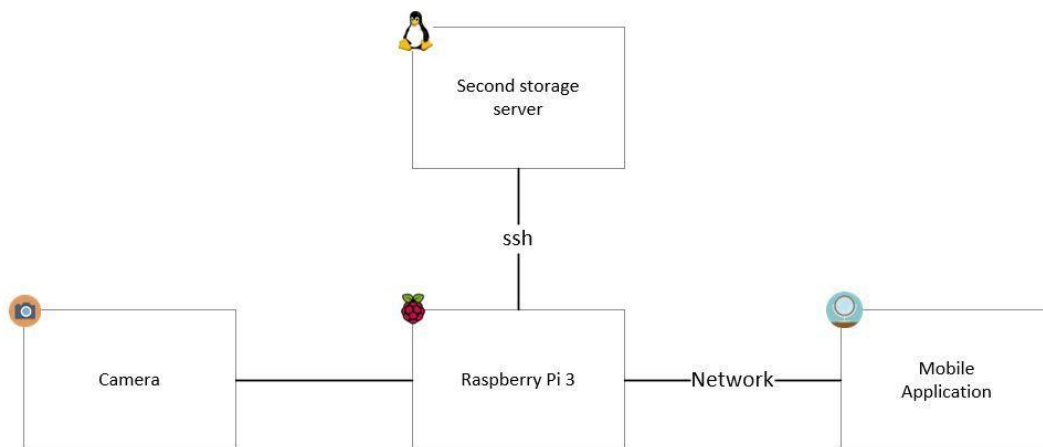
- وزن محصول بسته به شرایط و موارد مورد استفاده در تولید جعبه‌ی آن متغیر است.

## فصل 3

### معماری سیستم

سیستم طراحی شده توسط ما از دو قسمت سخت افزاری و نرم افزاری تشکیل شده است. قسمت سخت افزاری شامل یک رزبری پای و دوربین جهت دریافت داده ها و قسمت نرم افزاری ما شامل اپلیکیشن موبایل جهت استفاده از دوربین و یک استریم سرور در رزبری پای است.

معماری سیستم در شکل زیر قابل مشاهده است.



شکل 1 معماری کلی سیستم طراحی شده

## طراحی و پیاده سازی سخت افزار

پیاده سازی قسمت سخت افزاری با استفاده از قطعات زیر

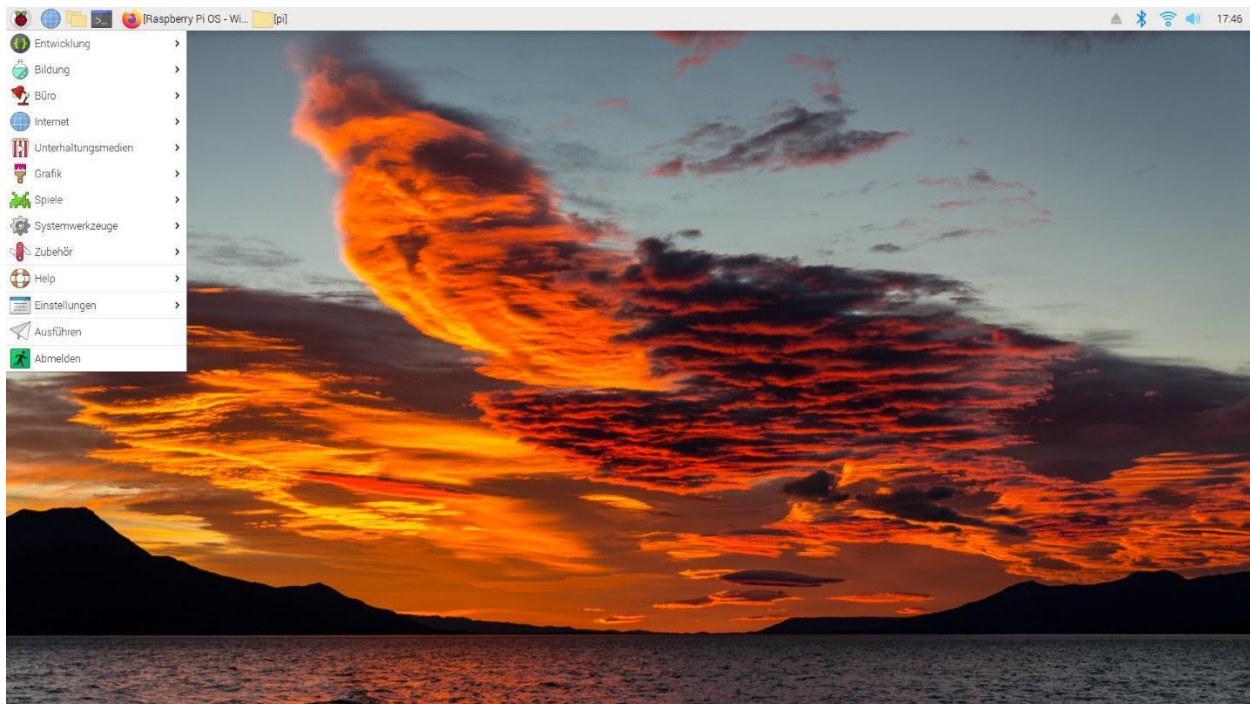
- برد رزبری پای
- دوربین 5 مگاپیکسلی رزبری پای
- LCD

### LCD

برای استفاده از LCD مشکلات زیر پیش آمد و بنابراین از استفاده از آن صرف نظر کردیم.

- ماژول LCD نیاز به نصب درایور دارد که توضیحات آن در این [لینک](#) آمده است.

با نصب این درایور و اتصال LCD به رزبری پای باید صفحه سیستم عامل رزبری پای را در LCD-3.5 inch مشاهده کنید.



شکل 2 صفحه‌ی سیستم عامل رزبری پای

اما اگر LCD مورد استفاده چینی باشد ممکن است با مشکلی مواجه شوید که صفحه آن سفید می ماند و یا پس از صبر طولانی صفحه می آید اما کار تصویر نویز دارد و حتی کارکرد آن نیز رندم است و ممکن است دفعه بعدی حتی با صبر بسیار نیز صفحه بالا نیاید.

پیشنهاد ما به شما این است که از خیر آن بگذرید و زمان بر روی آن نگذارید. تقریباً ۳ روز تلاش برای درست کردن مشکل بی ثمر بوده و در نهایت ما نیز به پاسخ هایی برخوردیم که اگر مدل شما چینی است و بار اول درست کار نکرد دیگر درست نخواهد شد.

## دوربین

ماژول دوربین را می توانید مطابق [لینک](#) متصل کنید.



شکل 3 ماژول دوربین 5 مگاپیکسلی رزبری پای



## نکاتی درباره دوربین

- دوربین را می‌توانید با دستور **raspivid** کنترل کنید. استفاده از لایبری **cameraPi** برای کنترل دوربین در کد پایتون نیز مهیا شده است.

پیاده سازی کد سرور در رزبری پای

برای پیاده سازی کد سرور راهکار های متفاوتی وجود دارد که به راهکار های تست شده می‌پردازیم و مزایا و معایب آن‌ها را بررسی می‌کنیم. راهکار های پیش رو شامل دو دسته استفاده از برنامه های موجود در PATH سیستم عامل و استفاده از کد پایتون خواهند بود و بررسی هر کدام به شرح زیر است.

- استفاده از **netcat** , **raspivid**

```
raspivid -t {time} -w {w} -h {h} -hf -fps 20 -o - | nc {ip} {port}
```

برای اجرای دستور بالا ابتدا در کامپیوتر مقصد دستور زیر را اجرا کنید.

```
nc -l {port} | mplayer -fps 200 -demuxer h264es -
```

سپس با اجرای دستور اول می‌توان استریم را در کامپیوتر اول مشاهده کرد.

از مزایای این روش می‌توان به ساده بودن و دلیلی فوق العاده کم در استریم آن نام برد.

- استفاده از **libcamera** , **vlc**

برای استفاده از این روش تنها نیاز است که VLC بر روی کامپیوتر مقصد نصب شده باشد.

با استفاده از دو دستور زیر می‌توانید استریم را مشاهده کنید.

```
libcamera-vid -t 0 --inline -o - | cvlc stream:///dev/stdin --sout '#rtp{sdp=rtsp://:8554/stream1}' :demux=h264
```

و در کامپیوتر مقصد دستور

```
vlc rtsp://<ip-addr-of-server>:8554/stream1
```

می‌توانید استریم را مشاهده کنید.

دقت کنید که روش بالا برای پروتکل های مختلفی پیاده سازی دارد  
(*http,tcp,udp,rtsp*).

از مشکلات این روش به دلیلی بالا آن می‌توان اشاره کرد اما توانایی رکورد آن توسط *vlc* وجود دارد.

مشکلات ضبط و استریم با هم

استفاده از روش های بالا در عین سادگی امکان ضبط همزمان و استریم کردن را نمی‌دهد مگر اینکه با استفاده از ۲ پردازش ضبط و استریم را انجام دهیم برای ضبط کردن می‌توان از کد زیر در پایتون استفاده کرد و به عنوان یک پردازش جدا آن را کرد. اما مشکلات آن را در زیر بررسی می‌کنیم.

```
def save_recording(data,name,conn):  
    print("create Picamera")  
    camera = PiCamera()  
    fileName = name  
    camera.start_recording(fileName)  
    camera.wait_recording(int(data["time"]))  
    camera.stop_recording()  
    camera = None  
    n = gc.collect()  
  
    print("video-recorded")  
    try:  
        f = open(fileName, "rb")  
    except:
```

```

print("Error: file not found")
return
print("start sending vide")
conn.send(fileName.encode())
print(conn.recv(1024))
video = f.read(1024)
while (video):
    conn.send(str(len(video)).encode())
    conn.recv(1024)
    conn.send(video)
    video = f.read(1024)
    conn.recv(1024)
conn.send("0".encode())
print(conn.recv(1024))
print("video sent")

```

- چون تنها یک پردازش توانایی برداشت اطلاعات از بافر دوربین را دارد این کار سبب می‌شود که هم استریم هم ضبط دچار وقفه‌های طولانی شود و عملاً فائزکنشالیتی برنامه از بین می‌رود.
- ماژول رزبری‌پای بر اثر افزایش دما ریووت می‌شود و ران کردن دو پردازش که دیتای سنگین فیلم و ویدئو را استریم و ذخیره می‌کند موجب داغ شدن و ریووت شدن سیستم می‌شود ( این فرایند تست شده است)

راهکار و پیاده‌سازی نهایی

به همین منظور که مشکلات ذکر شده پیش نیاید تصمیم بر آن شد که ریکورد استریم توسط اپلیکیشن موبایل انجام شود. بنابراین رزبری‌پای یک استریم سرور است که بر روی پورت ۸۰۰۰ آن بالا می‌آید.

پیاده‌سازی سرور با استفاده از **socket server** , **web server** , **stream server** پایتون پیاده‌شده‌اند.

## دلایل عدم استفاده از پروتکل‌های بالا به صورت کلی

به دلیل درخواست استاد گرامی مبنا بر نوشتن اپلیکیشن موبایل بر بستر پلتفرم اندروید، نمی‌توانستیم از بسیاری از پروتکل‌های ذکر شده برای نمایش استریم دوربین استفاده کنیم. همانطور که بالاتر نیز به آن اشاره شده است netcat بهترین ابزار برای تماشای استریم را برای ما فراهم می‌کرد، ولی استفاده از آن روی سیستم عامل اندروید غیرممکن بود و مجبور به انتخاب پروتکل‌های کندتر و ساده‌تر شدیم تا بتوانیم نمایش استریم بر روی گوشی را ممکن سازیم.

می‌توان به جای استفاده از موبایل، از یک کامپیوتر شخصی برای نمایش و ضبط استریم استفاده کرد. این کار یک استریم بدون لگ را برای ما مهیا می‌کند ولی کاربردی نخواهد بود، زیرا استفاده از کامپیوتر شخصی برای راننده‌ی یک ماشین امری بسیار دشوار یا غیر ممکن خواهد بود.

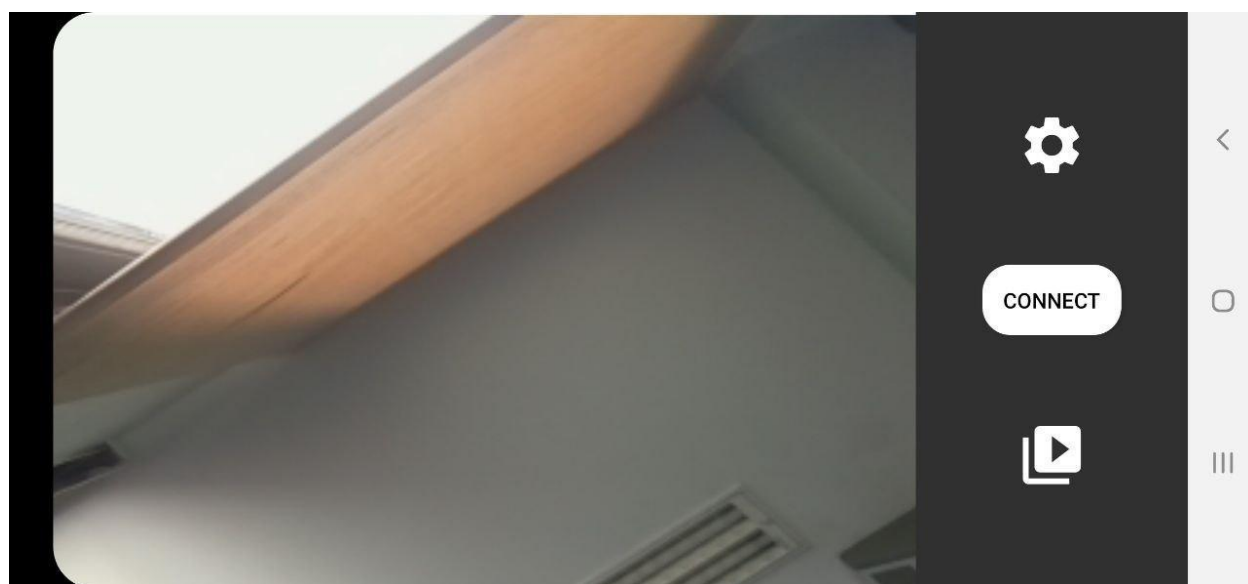
## مقایسه راهکار های گفته شده

توانایی ضبط و پخش	میانگین فریم بر ثانیه استریم(تاخیر)	روش
stream only	60 fps	raspivid + netcat
stream only	12 fps	raspivid + Network protocol (http,udp,tcp,rtsp)
stream only recording with vlc player	25 fps	libcamera + VLC (http,udp,tcp,rtsp)
stream and record (but needs second server)	60 fps	raw server on raspberry using raspivid
stream and record	60 fps	web server on raspberry using PiCamera

تاخیر گفته شده در جدول بالا به این صورت است که در حالت عادی استریم با ۳۰ فریم بر ثانیه باید انجام شود. هنگامی که لگ در استریم پیش می‌آید به این معناست که در فریم بر ثانیه پخش تصاویر تاخیر آمده است و برای محاسبه تاخیر ما میانگین fps سمت کلاینت را در طول یک دقیقه در نظر گرفته ایم. و این بدان معنا نیست که تصویر به عنوان مثال ۱۲ فریم بر ثانیه پخش میشود بلکه بدان معناست که تعداد فریم های دریافتی در یک دقیقه که باید معادل ۳۶۰۰ فریم باشد به عدد میانگین رسیده است.

## طراحی و پیاده سازی اپلیکیشن موبایل

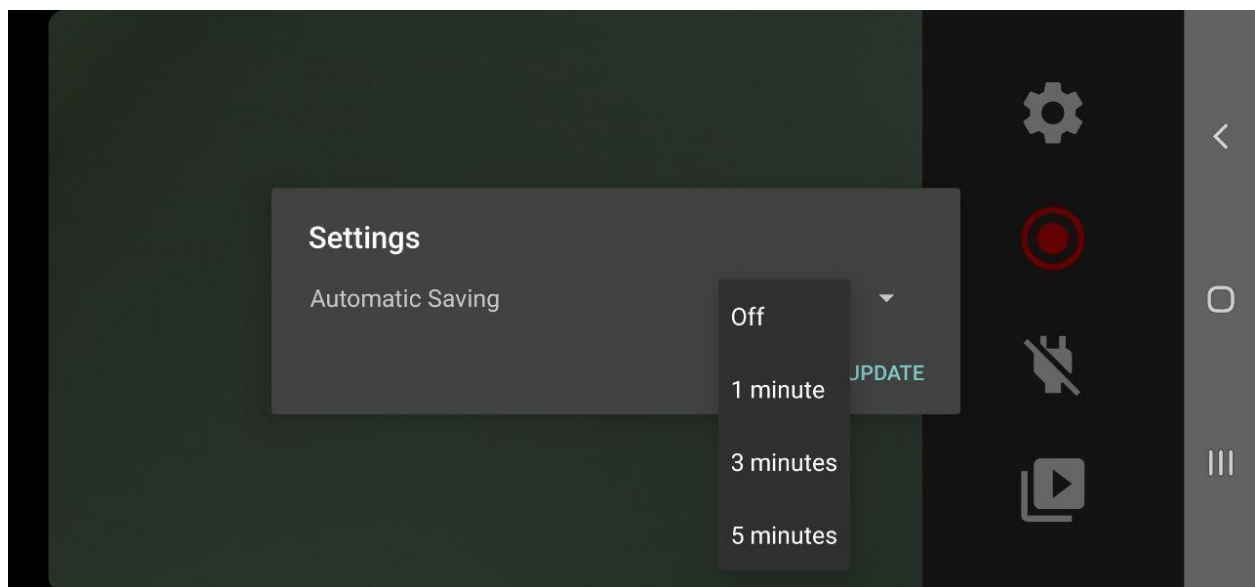
نمای کلی برنامه به صورت مقابل است:



شکل 4 نمای کلی اپلیکیشن

برنامه دارای 4 دکمه‌ی اصلی است که در سمت راست تصویر بالا مشخص است. در ادامه به شرح وظایف هرکدام از این دکمه‌ها می‌پردازیم:

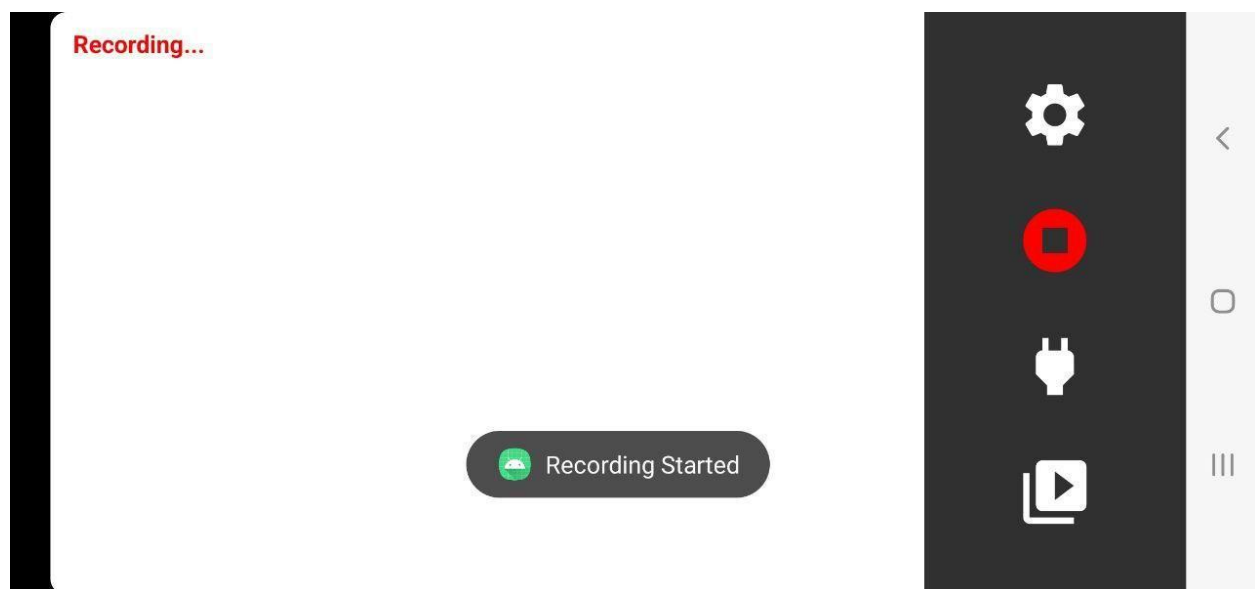
- دکمه‌ی تنظیمات :



شکل 5 انتخاب زمان ضبط از منوی تنظیمات

کاربر پس از فشار دادن این دکمه می‌تواند زمانی که می‌خواهیم رکورد کنیم را از بین گزینه‌های 1، 3 و 5 دقیقه انتخاب کند، و اگر کاربر بخواهد می‌تواند با انتخاب گزینه‌ی Off به مقدار دلخواه صفحه را رکورد کند.

- دکمه‌ی رکورد:

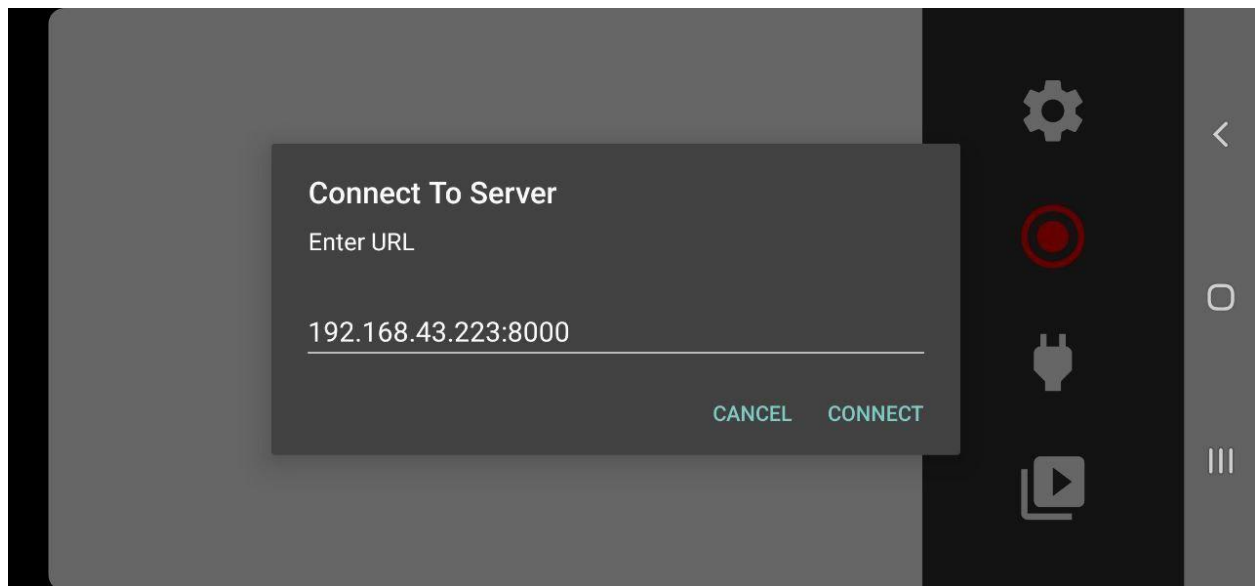


شکل 6 شروع ضبط صفحه در اپلیکیشن

پس از فشار دادن این دکمه یک Toast مبنی بر شروع شدن ضبط نشان داده می‌شود و تکست "Recording..." در بالای صفحه به نمایش در می‌آید. اگر کاربر زمان خاصی را انتخاب کرده باشد، رکوردینگ پس از زمان مشخص شده پایان می‌یابد و فایل در حافظه‌ی گوشی ذخیره می‌شود. دقت کنید در هر دو حالت (انتخاب زمان پیشفرض و عدم انتخاب زمان خاص) کاربر می‌تواند در صورت نیاز دوباره دکمه‌ی رکورد را بزند و رکوردینگ را به پایان برساند.



- دکمه‌ی Connect:



شکل 7 وارد کردن IP سرور رزبری در بخش اتصال

هنگام فشردن شدن دکمه‌ی Connect، یک Popup برای وارد کردن آدرس سرور به کاربر نمایش داده می‌شود. این IP در هر وصل شدن عوض می‌شود، دقت کنید در صورت عرضه‌ی این پروژه این حقیقت مشکل‌چندانی برای ما ایجاد نمی‌کند، زیرا در صورت عرضه شدن این پروژه سرور به ندرت خاموش می‌شود و IP برای مدت‌های بسیار طولانی ثابت می‌ماند، و یا حتی می‌توانیم یک IP ثابت برای هر کاربر با دریافت مبلغی اضافه تهیه کنیم.

## • دکمه‌ی Files:

هنگام فشردن دکمه‌ی Files، فولدری که فیلم‌های ضبط شده در آن جا ذخیره شده‌اند برای کاربر باز می‌شود تا بتواند فایل مورد نظر خود را تماشا کند.

## امکانات اپلیکیشن نوشته شده

### ذخیره‌ی اتوماتیک:

اگر در هنگام ذخیره‌ی استریم، کاربر به هر دلیلی (زنگ زدن گوشی، قفل شدن اشتباهی) مجبور به خروج از برنامه شود، ضبط متوقف می‌شود و پس از ورود کاربر ادامه می‌یابد.

### ذخیره‌ی استریم به مدت زمان‌های پیشفرض:

همانطور که در بخش توضیحات بخش‌های مختلف اپلیکیشن به آن پرداختیم، کاربر می‌تواند زمان‌های ضبط را برابر با 1، 3 و 5 دقیقه قرار دهد و یا می‌تواند کلاً این گزینه را نادیده بگیرد و ضبط را به صورت دستی قطع کرده و نتیجه‌ی آن را ذخیره کند.

این قابلیت در بسیاری از موارد می‌تواند کمک حال راننده باشد؛ فرض کنید راننده در حال رانندگی قصد ضبط کردن را دارد و نمی‌تواند به مدت زیادی با گوشی تعامل داشته باشد. او می‌تواند با انتخاب یکی از زمان‌های پیشفرض دکمه‌ی ضبط را بزند و با خیال راحت به رانندگی خود ادامه دهد و مطمئن باشد که ویدئو ضبط شده پس از گذر مدت مشخص شده در گوشی او ذخیره می‌شود.

### عدم وجود دلیلی و لگ در استریم و ضبط

یکی از مشکلاتی که در پیاده‌سازی پروژه به آن برخورد کردیم، وجود لگ و دلیلی در استریم به دلیل ویژگی‌های پروتکل‌های مختلف بود؛ به عنوان مثال پروتکل tcp به دلیل بررسی هر packet فرستاده شده به آن باعث بروز دلیلی بسیار زیاد در استریم می‌شد. پس از

بررسی پروتکل‌های مختلف به این نتیجه رسیدیم که استفاده از پروتکل HTTP بهترین نتیجه را دارد و می‌توان آن را بر روی برنامه‌های اندروید به راحتی نمایش داد.

مشکل بزرگ دیگری که زمان زیادی را صرف حل آن کردیم، عدم توانایی ضبط و استریم همزمان در رزبری پای بود. سیستم رزبری پای به دلیل داشتن یک بافر برای بایت‌های فرستاده شده توسط دوربین و داشتن یک Thread توانایی استریم و ضبط همزمان را به صورت lag free نداشت و هنگامی که می‌خواستیم استریم را ضبط کنیم، با دلیلی بسیار زیادی مواجه می‌شدیم. در نهایت توانستیم با ضبط استریم فرستاده شده توسط گوشی سربار ضبط و استریم همزمان را از دوش رزبری پای برداریم و بین پردازنده و گوشی کاربر تقسیم کنیم، که نتیجه‌ی آن یک استریم روان و بدون لگ است که می‌توان آن را به مدت زمان دلخواه ضبط کرد.

## اجرای سرور در رزبری پای

```
192.168.43.186 - - [23/Aug/2022 15:17:54] "GET /index.html HTTP/1.1" 200 -
192.168.43.186 - - [23/Aug/2022 15:17:54] "GET /stream.mjpg HTTP/1.1" 200 -
192.168.43.186 - - [23/Aug/2022 15:17:54] code 404, message Not Found
192.168.43.186 - - [23/Aug/2022 15:17:54] "GET /favicon.ico HTTP/1.1" 404 -
WARNING:root:Removed streaming client ('192.168.43.186', 45358): [Errno 32] Broken pipe
192.168.43.186 - - [23/Aug/2022 15:19:13] "GET /index.html HTTP/1.1" 200 -
192.168.43.186 - - [23/Aug/2022 15:19:13] "GET /stream.mjpg HTTP/1.1" 200 -
```

شکل 8 شروع سرور از طریق اجرای کد پایتون

سرور اجرا شده در رزبری پای اطلاعات گرفته شده از دوربین رزبری پای را در بافر کرنل قرار می‌دهد و در ادامه یک کلاس Streamer این اطلاعات را از بافر می‌خواند و در قالب یک فایل HTML که دارای یک img tag (که source آن بافر ذکر شده است) قرار می‌دهد.

**قیمت:**

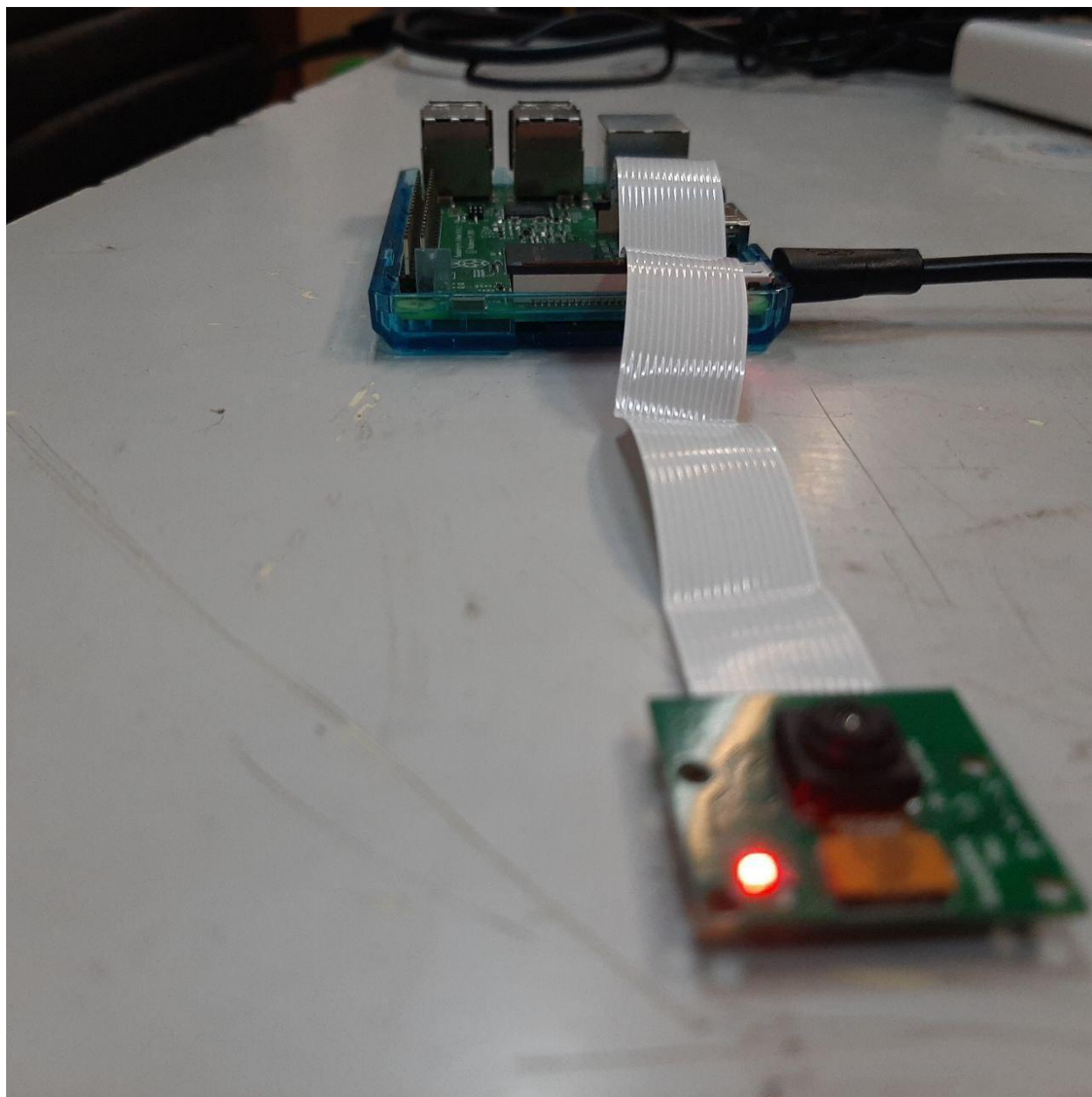
ردیف	قطعه	قیمت
1	رزبری پای 3	4,150,000 تومان
2	ماژول دوربین 5 مگاپیکسلی	159,500 تومان
3	جمع	4,309,500 تومان

جدول 2 قیمت قطعات و محصول نهایی

نماهایی از محصول



شکل 9 نمایی از محصول



شکل 10 نمایی از محصول



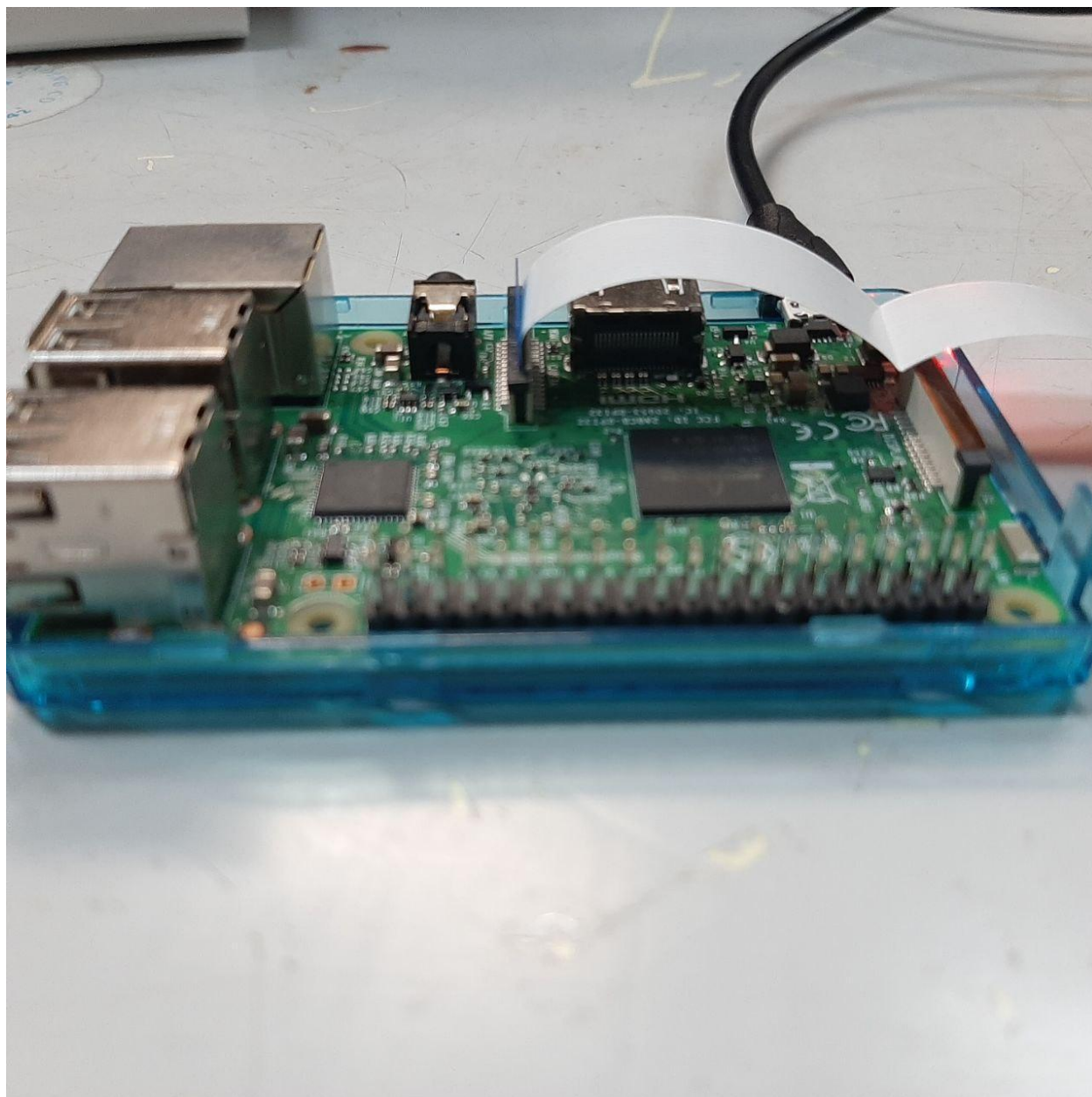


شکل 11 نمایی از محصول



شکل 12 نمایی از محصول





شکل 13 نمایی از محصول

## فصل 4:

### جمع‌بندی:

در این پروژه به پیاده‌سازی یک سیستم هوشمند آینه‌ی پشت ماشین پرداختیم. این سیستم با استفاده از یک دوربین و یک سرور در رزبری پای و یک اپلیکیشن موبایل می‌تواند به راننده در حین رانندگی کمک کند تا در هنگام وقایع غیر مترقبه مانند مسدود شدن دید پشت توسط سرنشینان پشت یا صندوق عقب ماشین به پشت خود دید داشته باشد.

در کنار این محصول، یک اپلیکیشن موبایل فراهم گشته تا با استفاده از آن بتوانید از استریم فیلم تهیه کرده و آن را در شبکه‌های اجتماعی برای مخاطبان خود به اشتراک بذارید.

می‌توان با استفاده از ابزارهای یادگیری ماشین بر روی رزبری پای یک مدل تشخیص اشیاء تا بتواند به راننده هشدارهای لازم جهت جلوگیری از وقوع حادثه را دهد.