



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر
زبان و ساختار کامپیوتر ترم ۳ کارشناسی

عنوان:

پروژه اصلی ۴ پردازش تصویر با استفاده از رزبری پای

نگارش

۱۰ گروه

استاد درس

دکتر اسدی

۱۴۰۳ پاییز

چکیده: تحت عنوان پروژه درس ساختار و زیان کامپیوتر، با هدف یادگیری مقدمات پردازش تصویر و هوش مصنوعی ما این پروژه را به ثمر رساندیم. هدف کلی پروژه کار با سیستم عامل رزبری پای، تشخیص معیار پس از آموزش مدل، محاسبه مساحت سطوح بسته است. به دلیل محدودیت های مالی کشور ما قادر به تهیه برد رزبری پای نبوده و تمام پروژه بر روی شبیه ساز اجرا شده.

واژه‌های کلیدی: رزبری پای، پردازش تصویر، هوش مصنوعی

۱ شرح صورت پروژه

در این بخش ابتدا به شرح صورت پروژه و اهداف آن می‌پردازیم. این موضوع برای درک بهتر ارزش و روش کار ما ضروری است

۱-۱ مقدمه

پروژه "محاسبه مساحت" یک سیستم ساده است که از یک دوربین، ال‌سی‌دی و برد رزبری پای برای شناسایی و محاسبه مساحت اشکال بسته بر روی یک صفحه سفید استفاده می‌کند. این سیستم می‌تواند مساحت هر کدام از اشکال را با توجه به یک علامت معیار از پیش تعیین شده محاسبه کند. با استفاده از دکمه‌ای برای جابجایی بین اشکال مختلف، سیستم به طور خودکار اندازه‌گیری‌های لازم را انجام داده و نتیجه را بر روی نمایشگر ال‌سی‌دی نمایش می‌دهد. این پروژه نه تنها کاربردی است، بلکه زمینه‌ای برای یادگیری پردازش تصویر، شناسایی اشکال هندسی و کار با سخت‌افزارهای رزبری پای فراهم می‌کند. اهداف پروژه به شرح زیر اند:

- پیاده سازی مدار و اتصال دوربین و LCD به رزبری پای
- تشخیص علامت معیار برای محاسبات بعدی
- طراحی سیستم: توسعه سیستم شناسایی و محاسبه مساحت اشکال و کار با دوربین
- نمایش بی‌درنگ: نمایش مساحت به صورت زنده بر روی ال‌سی‌دی.
- امکان جابجایی بین اشکال با استفاده از سوییچ

۲-۱ صورت داک پروژه

در این پروژه می‌خواهیم سیستم ساده‌ای بسازیم که مساحت اشکال بسته درون یک صفحه سفید را با توجه به یک علامت معیار از پیش تعیین شده و با اندازه قراردادی محاسبه کند. در ابتدا دوربین مقابله علامت معیار قرار گرفته و بعد از آن، اندازه معیار شناسایی می‌شود. سپس باید اشکال رسم شوند و با استفاده از دوربین، مساحت این اشکال محاسبه و اعلام شود. دقیق شود که در پیاده‌سازی خود، می‌توانید از OpenCV یا الگوریتم پیاده‌سازی شده توسط خودتان استفاده کنید. سیستم محاسبه مساحت باید ویژگی‌های زیر را داشته باشد:

- در ابتدا با رسم معیار، به سامانه اندازه معیار را توضیح دهید.
- سپس با رسم یک شکل روی یک صفحه سفید، باید بتواند مساحت آن را محاسبه کند.
- باید امکان جابجایی میان اشکال رسم شده روی صفحه سفید با استفاده از سوییچ وجود داشته باشد.
- به صورت بی‌درنگ، مساحت یکی از اشکال روی صفحه ال‌سی‌دی نمایش داده شود.
- در صورت نگهداشتن سوییچ به مدت ۴ ثانیه، باید معیار فراموش شود و معیار جدیدی انتخاب شود.
- باید امکان ادغام کردن دو شکل با وصل کردن آن دو به هم وجود داشته باشد. در این صورت باید این شکل بسته جدید به عنوان شکل شناسایی شود.

۳-۱ وسائل مورد نیاز

- یک برد رزبری پای
- یک ال‌سی‌دی
- یک دوربین
- یک سوییچ

۴-۱ گام‌های پروژه

۱. اتصال رزبری پای، دوربین، صفحه ال‌سی‌دی و سوییچ و ارتباط میان آن‌ها
۲. نوشتن برنامه تشخیص معیار با استفاده از تصویر مشاهده شده در دوربین
۳. نوشتن برنامه محاسبه مساحت شکل بسته با معیار تعریف شده
۴. برنامه‌ریزی ال‌سی‌دی برای نشان دادن مساحت یکی از اشکال
۵. استفاده از سوییچ برای جابجایی میان اشکال و عوض کردن معیار
۶. تست و اطمینان از نحوه پیاده‌سازی

۲ گزارش کار پروژه

در قسمت قبل با اهداف و شرح صورت کار آشنا شدید. در این بخش مرحله به مرحله گزارش گام‌های پروژه و نتایج به دست آمده را شرح می‌دهیم.

```
1 import cv2
2 # Replace with your camera's IP address and port number
3 ip_camera_url = 'http://192.168.1.8:8080/video'
4 # Create a VideoCapture object
5 cap = cv2.VideoCapture(filename=ip_camera_url)
6 if not cap.isOpened():
7     print("Error: Could not open video stream from the camera.")
8 else:
9     print("Successfully connected to the camera.")
10 # Read and display frames
11 while True:
12     ret, frame = cap.read()
13     if not ret:
14         print("Error: Could not read frame.")
15         break
16     cv2.imshow(winname='IP Camera Stream', mat=frame)
17     # Exit the loop when 'q' key is pressed
18     if cv2.waitKey(delay=1) & 0xFF == ord('q'):
19         break
20
21 # Release the capture and close windows
22 cap.release()
23 cv2.destroyAllWindows()
```

شکل ۱: camera connecting

۱-۲ گام اول: اتصالات

هدف کلی این بخش مسئله اتصالات lcd و برد رزبری پای است. از آنجا که دانشگده هیچ یک را در اختیار ما قرار نداد تمامی مراحل به صورت شبیه سازی شده بر روی کامپیوتر انجام شده.

۱-۱-۲ اتصال دوربین

همانطور که در شکل ۱ میبینید در ابتدا کتابخانه ۲۵۷ را ایمپورت میکنیم و سپس با استفاده از نرم افزار Camera ip که صرفا برای اتصال دوربین گوشی به برنامه و نمایش تصویر روی لپتاپ استفاده میشود، ایپی دستگاه متصل به لپتاپ را در این قسمت وارد میکنیم تا دستگاه به لپتاپ متصل شود. سپس در خط ۵ یک ابجکت ویدیو کپچر میسازیم که در واقع به صورت زنده بتوانیم هر لحظه فریم های گرفته شده توسط دوربین را کپچر کنیم. سپس چک میکنیم که استریم به درستی متصل شده یا نه. و در نهایت در یک حلقه نامتناهی در هر لحظه یک فریم را در ورودی میخوانیمو در خط ۱۶ انرا نمایش میدهیم. خط ۱۸ و ۱۹ هم برای ست کردن کلید q برای خروج از برنامه هست. در خط ۲۲ و ۲۳ نیز تمامی پنجره های اضافی را بسته و در خط ۲۲ فریم ها را میبندیم.

۲-۱-۲ اتصال رزبری پای و نمایشگر LCD

به دلیل محدودیت های سخت افزاری این مورد تحت تاثیر کمبودات قرار گرفت و مجبور شدیم با استفاده از شبیه ساز ها سیستم عامل رزبری پای را شبیه سازی کنیم. KASRA بقیه اش رو کامل کن عکس بزار نمی خواهد خیلی توضیح بدی

۲-۲ گام دوم: برنامه تشخیص معیار

در این قسمت باید برنامه ای پیاده سازی می کردیم که معیاری را ببیند و آن را در محاسبات بعدی تشخیص دهد. به دلیل گوناگونی معیار ها و وجود اشکال شبیه معیار، تعیین کردیم که معیار باید قرمز رنگ باشد و در تصاویر بعدی هم اشکال قرمز را با معیار مقایسه می کند. چالش اصلی اینجا تشخیص مشخصات معیار، و استفاده از این مشخصات در مراحل بعد برای تشخیص و بازیابی دوباره آن است.

۳-۲ گام سوم: نوشتمن برنامه محاسبه مساحت

این قسمت به لطف کتابخانه قدرمند ۲۵۷ بسیار ساده گذشت. تنها پیچیدگی که لحاظ کردیم مسئله تشخیص مساحت و اینکه کدام یک ارزشمند هستند بود. در واقع این کتابخانه همه گوشه هارا حساب می کرد و این باعث می شد داده پرت زیاد داشته باشیم، مثلا کوچکترین سایه ای هم مساحتی دارد

در صفحه. که با تعیین حد پایین منطقی توانستیم این داده های پرت را حذف کنیم. از آنجا که سعی می کردیم طراحی برنامه با اصول برنامه نویسی همخوانی داشته باشد، این مورد را به صورت مازولات و تحت یک تابع پیاده سازی کردیم.

```

1 import cv2
2 import numpy as np
3
4 # Global variable to hold the last detected red object as a template
5 last_template = None
6 def calculate_area(frame, scale) -> list:
7     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
8     ret, thresh = cv2.threshold(gray, 150, 255, cv2.THRESH_BINARY_INV)
9     contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
10    areas = []
11    cnts = []
12    for cnt in contours:
13        area = cv2.contourArea(contour=cnt)
14        if area > 1000: # Threshold for minimum area
15            areas.append(area / (scale ** 2)) # Calculate the area relative to the scale
16            cnts.append(cnt)
17            cv2.drawContours(frame, [cnt], -1, (0, 255, 0), 2)
18
19    return areas
20

```

area surface :۲

در خطوط آغازین همه کنتور ها (گوشه ها) را شناسایی می کند و در حلقه به محاسبه مساحت با استفاده از توابع آماده می پردازد. اگر مساحت آن بر حسب پیکسل از حد پایین معقولی پایین تر بود به عنوان داده پرت، کنار گذاشته می شد و برگردانده نمی شد. همینطور بر اساس آینده نگری ای که داشتیم از همینجا scale را تعریف کردیم، در این گام هر وقت تابع را صدا می کردیم اسکیل را یک می دادیم ولی طبیعتا در مراحل بعدی نیاز های ما با این ابتکار برآورده می شد و نیازی به تغییر دوباره نبود.

۴-۲ گام چهارم: برنامه نویسی LCD

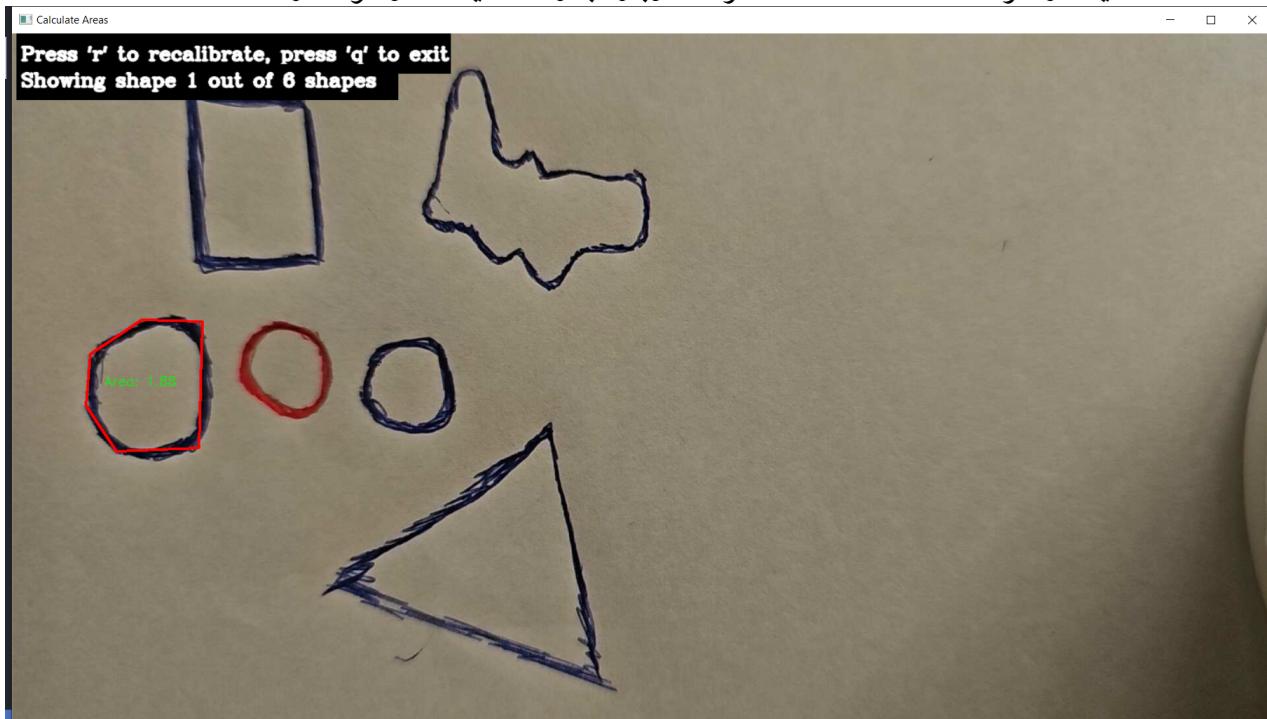
این مورد هنگامی که دانشکده قطعه ای تحویل ما نداد کاملا به مشکل خورد چون ما توان و بنیه تهیه صفحه جدید نداشتیم و هرچه داشتیم مانند صفحات نمایشگر بود که فرقی صفحه لپتاپ نداشت. این مورد در شبیه ساز اجرا شده و نمایشگر آن خو صفحه لپتاپ است که در اختیار سیستم عامل رزبری پای قرار می گیرد. برای عکس های بیشتر می توانید به گام آخر مراجعه کنید.

۵-۲ گام پنجم: استفاده از سوییچ

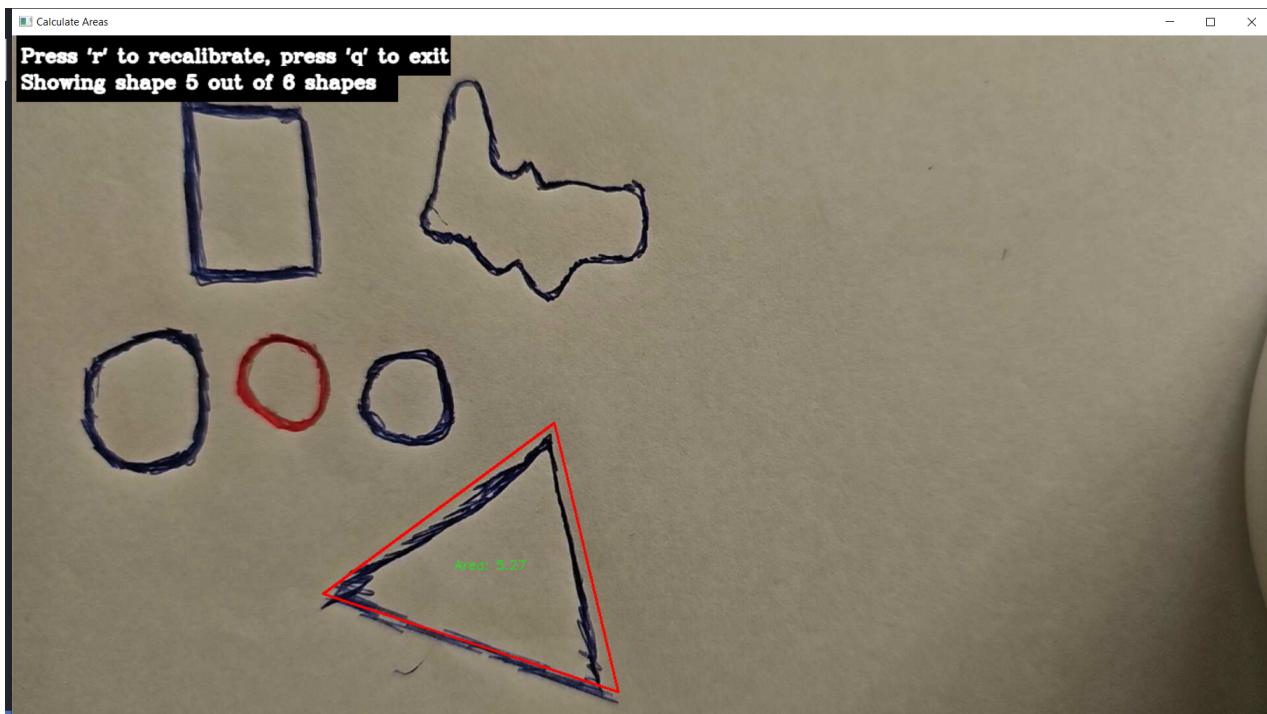
این مورد هم هرچند توسط دانشکده فراهم نشد ولی ما با استفاده از شبیه ساز و کیبورد لپتاپ و دکمه های d و R سوییچ کردن را پیاده سازی کردیم که عکس های نتایج آن را در گام آخر می توانید بینید.

۶-۲ گام آخر: تست پروژه و بررسی نتایج

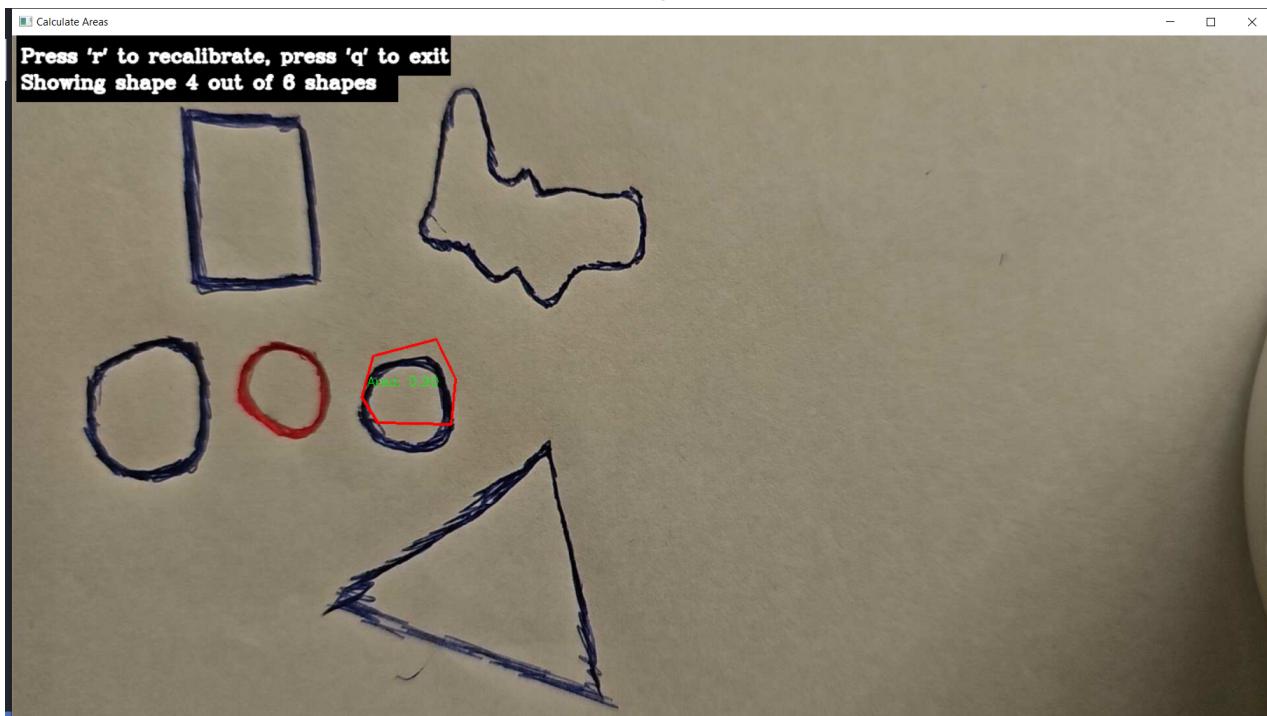
مطابق عکس های زیر برای تست نرم افزار بایستی در ابتدا پس از اجرای برنامه یک شکل بسته به رنگ قرمز روی کاغذ کشید و با زدن کلید R تمام مساحت اشکال ها بمحاسبه این مبدأ scale خواهند شد. سپس پس از یافتن مبدأ با زدن کلید r میتوانیم مساحت هر کدام از اشکال را محاسبه کنیم و با استفاده از a ، d بین اشکال جابجا شویم و مساحت هر کدام را بمحاسبه مبدأ مشاهده کنیم.
دقت کنید در هر لحظه امکان scale کردن دوباره با زدن کلید R وجود دارد.



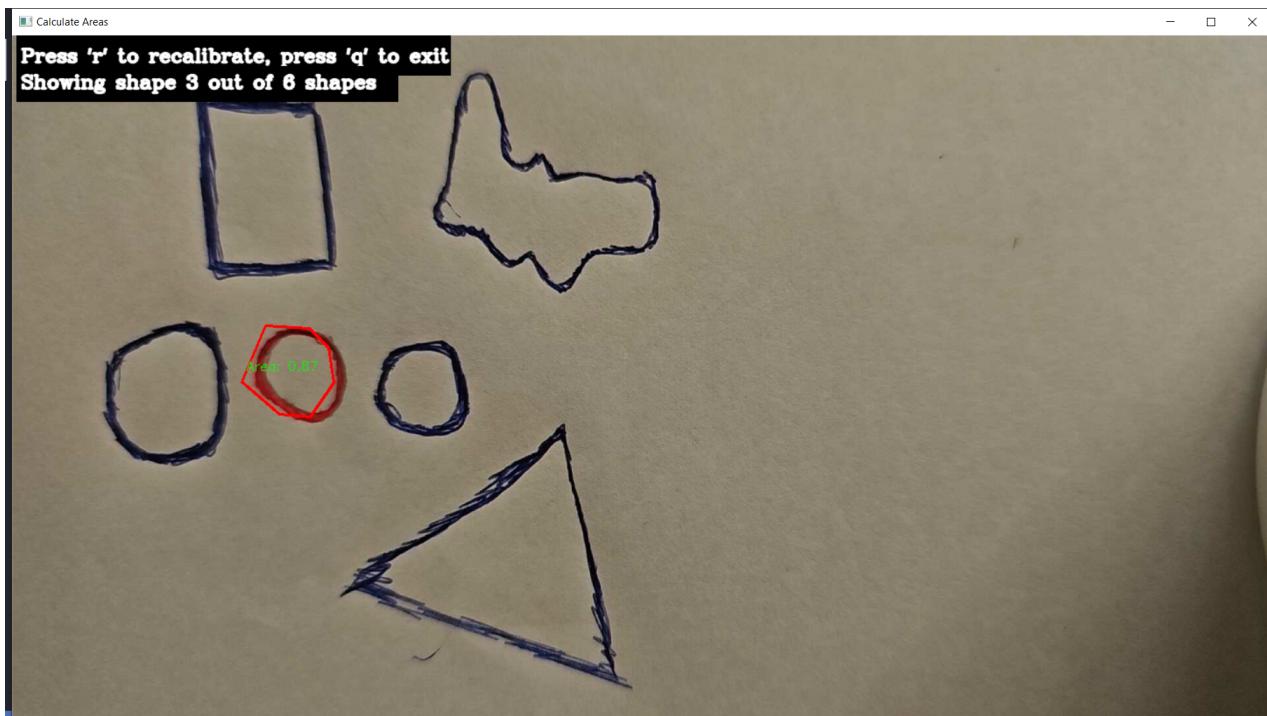
شکل ۳: First



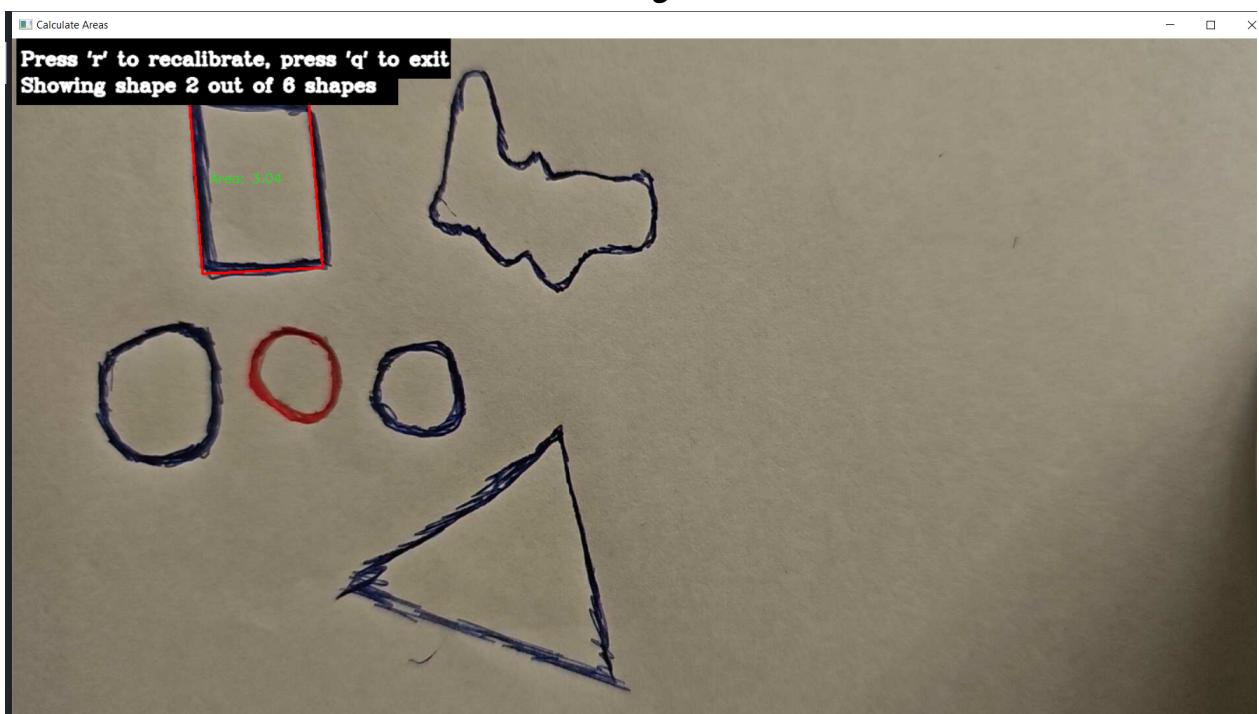
شكل :٤



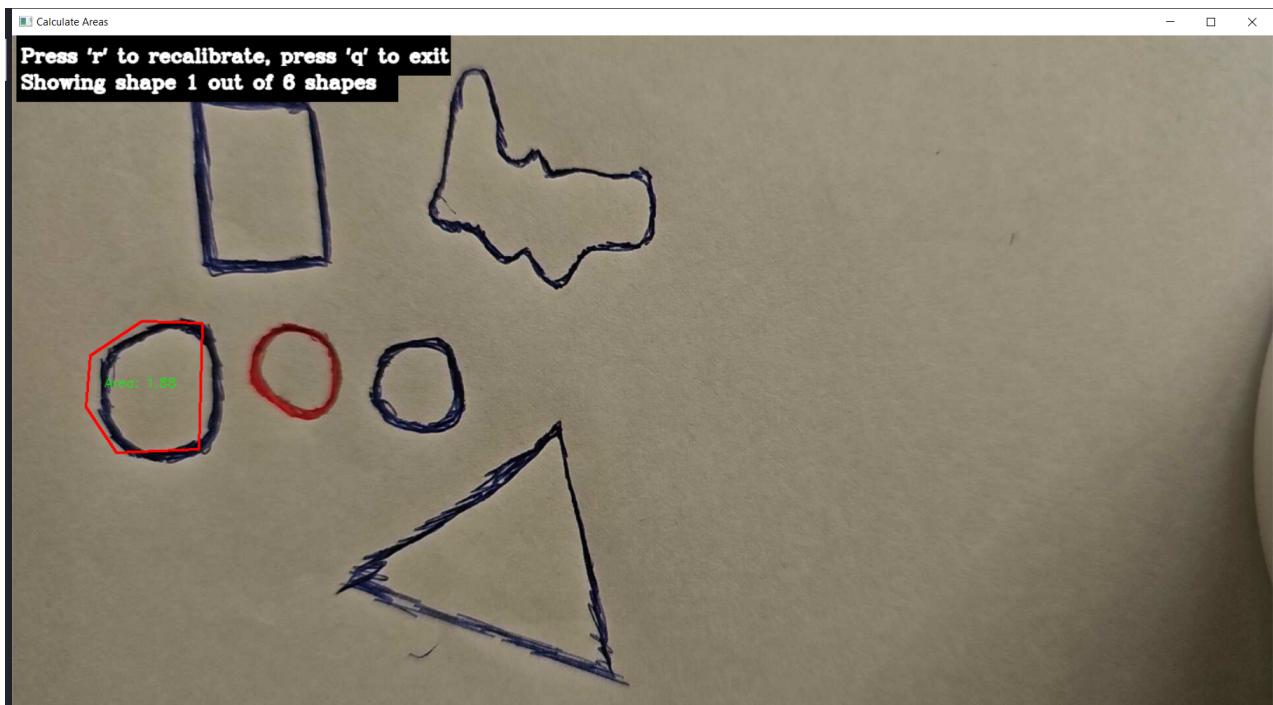
شكل :٥



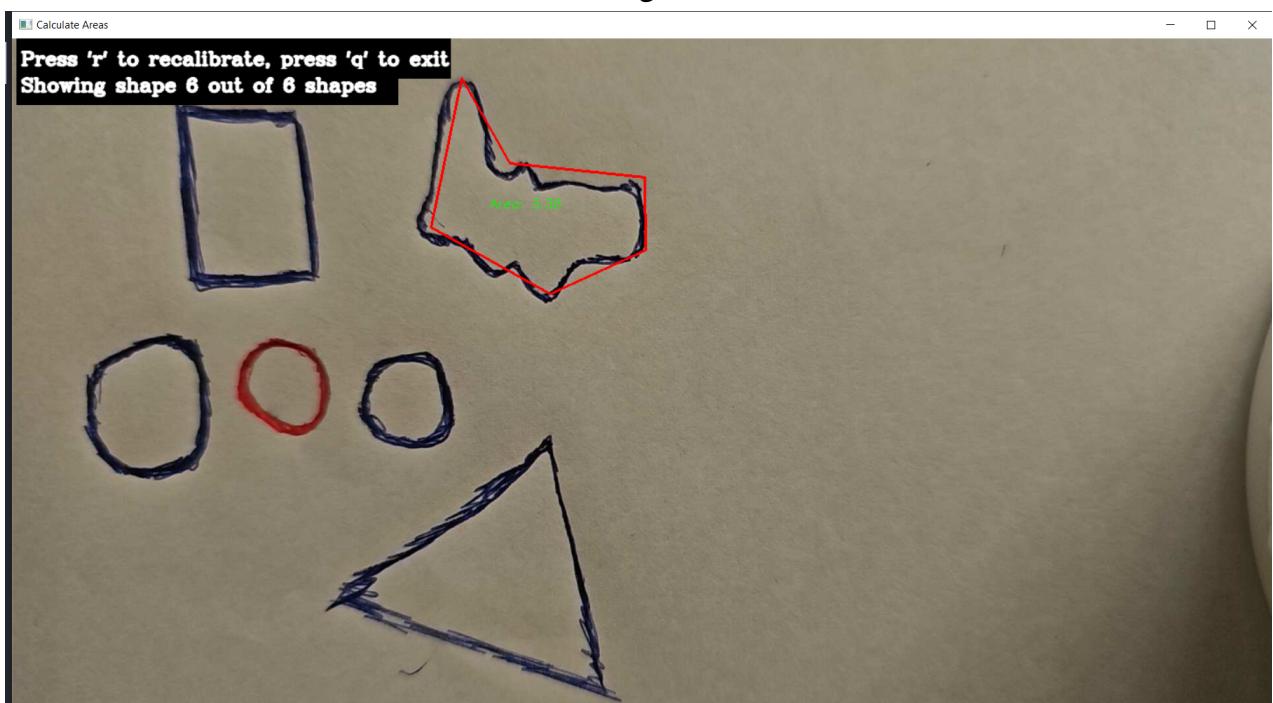
شكل :٦



شكل :٧



شکل :۸ Sixth :



شکل :۹ Seventh :

عکس اول شروع پروژه و ۷ آخرين شكل با سويچ کردن است

۳ نتیجه گیری

با وجود تمام محدودیت های مادی، ما با علم و سعی و تلاش خود توانستیم به بهترین نحو ممکن پروژه را به ثمر برسانیم. در این راستا در این بخش چند مورد از چالش ها، نکات جدیدی که یاد گرفتیم، نتایجی که دریافت کردیم و همینطور تحلیل آنها را بررسی می کنیم.

۱-۳ چالش ها و نیازمندی ها

صرف نظر از چالش های مادی مانند نبود برد، نمایشگر و حتی سوییچ ساده، ما با چالش های زیادی در بخش علمی و عملی پروژه مواجه بودیم صرف نظر از چالش های مادی مانند نبود قطعات ما در مواجه با این مفهوم جدید، پردازش تصویر، با چالش هایی نظیر چگونگی تطابق و استفاده از کتابخانه ها در جهت اهداف گفته شده در صورت پروژه مواجه بودیم. مورد اولی که چالش ایجاد کرد ماهیت دید کامپیوتر بود، با وجود توابع بسیار برای مساحت و طول، ما باید تصویر هارا ابتدا پردازش می کردیم، با فیلتر ها به تصاویر قابل فهم تر و دقیق تری تبدیل می کردیم که رایانه بتواند شکل هارا درست تشخیص دهد و داده های پرت حذف شوند. پیدا کردن ضرایب مناسب برای ترشولد ها یا فیلتر ها خود چند روز زمان برد.

مورد بعدی مسئله محاسبه معیار بود. توابع با پیکسل کار می کردند و باید اعداد را به واحد های قابل فهم تبدیل می کردیم. همچنین متوجه شدیم به لحاظ علمی برخی توابع می توانند خروجی های بسیار غیر قابل انتظاری برگردانند که این خود چالشی عظیم بود که فقط با کنار گذاشتن آنها و ساده تر کردن مدل قابل حل بود

سوییچ کردن بین شکل ها به تنها یی مشکل بزرگی نبود ولی از حیث چگونگی نوشتار و رسم خطوط برای زیبایی بیشتر خود چالشی زیبا بود.

چالش های برنامه نویسی چندانی هرچند نداشتیم ولی خود این چالش را پذیرفتیم که برنامه ماژولار باشد و در ادامه هم بتوان از قسمت هایی یا کل آن استفاده کرد یا حتی توسعه داد و داکیومنت نویسی خوبی داشته باشیم

۲-۳ نتایج و تحلیل

نتایج که با استفاده از کد ها بدست می آمد، حاصل ساده سازی ها، پاکسازی داده ها و ... بود. این نتایج هرچند توسط برنامه تعیین می شوند و نمونه آن را در تصاویر گام آخر مشاهده می کنید، نتایج جالبی دارند که در خورد تحلیل است ابتدا با وجود سرعت بالا در پردازش و بی درنگ بودن، با اندکی تغییر زاویه دوربین نتایج می توانند به طور شگرفی تغییر کنند. یا به دلیل محاسبه پیکسل به پیکس خطای می تواند غیر قابل پیشگیری شود. این مورد را وقتی بعد از کالیبره کردن دوربین مساحت هارا پیدا می کنیم می توان دید، که خود معیار مساحت دقیقا ۱ نخواهد داشت. همچنین قطر و ضیحامت خود اصلاح تعیین کننده است و چون توابع نمی توانند خط صاف را بینند و فقط ضیحامت را می توانند تشخیص دهنند شناسایی اشکال خود نیازمند نورپردازی مناسب و قطور سازی اصلاح است که خود بر مساحت می تواند تاثیر بگذارد.

۳-۳ نتیجه گیری نهایی و سخن آخر

پردازش تصویر پروژه ای به یادماندنی با بار آموزشی بسیار برای ما بود که بتوانیم با دنیای هوش مصنوعی و پردازش تصویر آشنا شویم. به خاطر محدودیت ها از شبیه ساز ها استفاده کردیم و مزه استفاده از قطعات حقیقی را نچشیدیم. ساده سازی های آماری داده ها جزء پیش بینی نشده ای بود که از پس آن برآمدیم. برنامه ما دونوع خطای داشت: یکی شامل نوع دید دوربین و نورپردازی و دیگری خود ریاضیات توابع که پیکسلی کار می کردند و روابط پیچیده و ظریفی داشتند. با تلاش بسیار خطای داده های پرت را کم کردیم. برنامه ای قابل توسعه بیشتر و مازولار طراحی و مهندسی کردیم و یروژه پردازش تصاویر شامل محاسبه مساحت اشکال و نشان دادن آن روی صفحه LCD را به اتمام رساندیم.

References