



پروژه پایانی درس پردازش تصویر رقمی  
ارائه راهکاری مبتنی بر پردازش تصویر در جهت تشخیص و شناسایی تابلوهای  
راهنمایی

اعضای گروه

حسین پرتو هفشجانی

علی صفرپور دهکردی

محمد مظفری

زهرا یوسفی

استاد درس

خانم دکتر کسائی

تدریس‌یار همراه

آقای آرش چایچی

تابستان ۱۴۰۱

## فهرست مطالب

۲	چکیده
۳	مقدمه
۳	تعریف مسئله
۴	معرفی برخی راهبردها
۵	راهبرد پیشنهادی
۵	استخراج تابلوهای راهنمایی و رانندگی
۶	آستانه‌گیری از مقادیر پیکسل‌ها
۶	بستن مورفولاجیکال و تشخیص لبه
۷	تشخیص تابلوها
۸	خطاهای روش‌های تشخیص علائم راهنمایی و رانندگی
۱۰	مجموعه دادگان
۱۱	دسته‌بندی تابلوها
۱۴	مات شدگی و رفع مات شدگی
۱۴	کلیات
۱۹	بررسی یک روش تماماً چشم پوشیده
۲۰	بررسی روش‌های مبتنی بر اعمال فیلتر از قبیل فیلر وینر و فیلتر شارپ کننده
۲۵	روش‌های تکرار شونده
۲۷	ارزیابی
۳۳	کارهای آینده
۳۳	جمع‌بندی
۳۴	منابع
۳۶	سرواژه‌ها
۳۶	معادل‌ها

## چکیده

با پیشرفت خودروهای خودران، نیاز به سامانه‌هایی که به این خودروها در تشخیص تابلوهای راهنمایی و رانندگی کمک کنند، رو به افزایش است. این سامانه‌ها باید توانایی تشخیص تابلوهای راهنمایی و رانندگی که در مسیر خودرو قرار دارد را داشته باشند اما حرکت خودرو باعث مات شدگی تصویر دریافتی خودرو خواهد شد که تشخیص این تابلوها را سخت‌تر می‌کند. علاوه بر این چالش، تابلوهای راهنمایی و رانندگی شکل‌ها و رنگ‌های مختلفی دارند که استخراج آن‌ها را از تصویر ورودی چنین سامانه‌ای پیچیده‌تر می‌کند. در نهایت تابلوهای استخراج شده از تصویر ورودی این سامانه باید در دسته‌بندی شوند تا معنی آن‌ها تفسیر شود. در این پژوهش تلاش کردیم تا راه‌حلی برای رفع این چالش‌ها ارائه دهیم.

## مقدمه

امروزه در بسیاری از زمینه‌ها رد پای هوش مصنوعی دیده می‌شود و پردازش تصویر به عنوان یکی از شاخه‌های مهم هوش مصنوعی، توانسته است در مسائل مختلفی ایفای نقش نماید و موفقیت چشم‌گیر آن بر کسی پوشیده نیست.

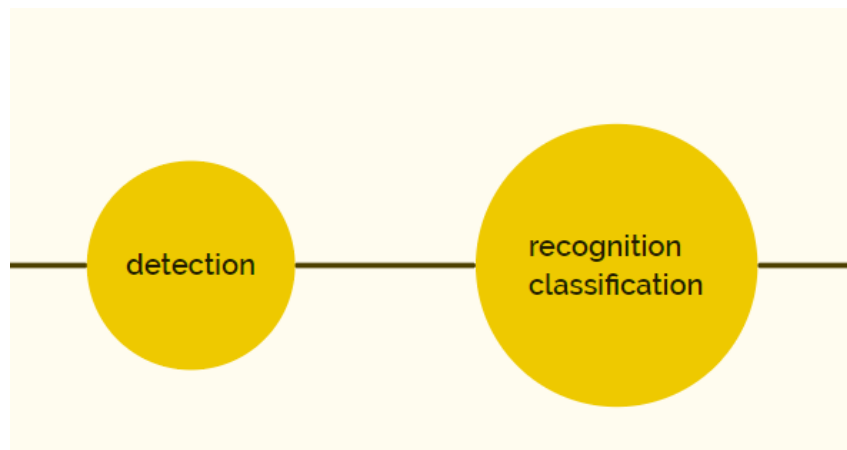
از زمانی که ماشین‌ها ساخته شدند، این مسئله که برای ایجاد نظم و قوانین باید کاری کرد مطرح گردید [1]. بخشی از این کار به ایجاد راهبردی جهت انتقال نکات و پیام‌های مهم به رانندگان معطوف گردیده است و در این راستا تابلوهای راهنمایی و رانندگی به عنوان یک راه‌حل مقبول و همگانی مورد استفاده قرار گرفته‌اند. این تابلوها در جهت ارائه اطلاعات مختلف و هشدار دادن به رانندگان قوانین تهیه گردیده‌اند [2]. در این بین بی‌توجهی رانندگان به این علائم می‌تواند مخاطراتی داشته باشد و از آن مهم‌تر، با ایجاد ماشین‌های خودران، نیاز به تشخیص و شناسایی تابلوها در جهت درک پیام آن تبدیل به امری بسیار حیاتی شده است. در نتیجه، در سال‌های اخیر پژوهش‌های مختلفی با تمرکز بر تشخیص تابلوهای راهنمایی و رانندگی انجام شده است و توجه به این موضوع روز به روز افزایش می‌یابد [1-5]. در همین راستا، از میان روش‌های مختلفی که برای حل این مسئله مطرح شده‌اند، یادگیری عمیق به عنوان یکی از موفق‌ترین راهبردهای هوش مصنوعی در این زمینه عمل کرده است. اما محدودیت زمانی و سخت‌افزاری باعث شده است که در ایجاد سامانه‌های کاربردی از مدل‌های حجیم و نیازمند به پردازش زیاد چشم پوشی گردد.

## تعریف مسئله

سامانه‌ی طراحی شده برای حل این مسئله باید توانایی دریافت و پردازش مجموعه‌ای از تصاویر ورودی جهت استخراج و تشخیص تابلوهای راهنمایی و رانندگی را داشته باشد. همچنین، این سامانه باید توانایی دسته‌بندی تابلوهای راهنمایی و رانندگی استخراج شده از تصاویر مختلف را داشته باشد.

در راستای نیل به این هدف سامانه‌های مختلفی ارائه شده است. در این دست سامانه‌ها، جهت تشخیص محل قرار گرفتن تابلوهای راهنمایی و رانندگی از دو ویژگی مهم این تابلوها استفاده می‌گردد. این دو ویژگی عبارت‌اند از رنگ و شکل تابلوها. تابلوهای راهنمایی و رانندگی رنگ‌های به خصوصی مانند قرمز، سفید، سیاه، آبی و زرد دارند که توجه رانندگان را به خود جلب می‌کنند. این تابلوها همچنین به شکل‌های هندسی ساده مانند دایره، مثلث و مستطیل هستند [5]. در موارد دیگر اشکال مشخصی وجود دارند که الگوریتم‌های تشخیص این اشکال می‌تواند

چاره گشا باشد [4]. پس به طور کلی راهبرد ارائه شده در این پژوهش، شامل دو فاز اصلی است که در شکل ۱ قابل مشاهده است. ابتدا باید تابلوها تشخیص داده شده و سپس دسته‌بندی شوند.



شکل ۱ - مسیر عملکرد کلی سامانه

همچنین در این زمینه چالش‌های مختلفی وجود دارد. وجود انواع پس زمینه‌ها، خرابی و آسیب تابلوها، پوشیده شدن بخشی از تابلو، وجود نویز در تصاویر، کیفیت پایین تصاویر یا تارشدگی، تغییر میزان نور در ساعات مختلف و تغییرات آب و هوایی از جمله این موارد هستند [3]. از دیگر چالش‌ها نیز می‌توان محدودیت‌های تصویر برداری، وجود تعداد متغیری از تابلوها در هر تصویر، محدودیت مجموعه داده‌های در دسترس به ویژه برای روش‌های یادگیری عمیق و وجود داده‌های محدود برچسب‌گذاری شده برای روش‌های ساده‌تر و مدت زمان پردازش محدود را نام برد [2].

## معرفی برخی راهبردها

در این راستا ویش و همکاران [1] با استفاده از هیستوگرام شیب‌های جهت دار و تحلیل رنگ‌ها به تشخیص شکل‌ها می‌پردازند و سپس به کمک روش ماشین بردار پشتیبانی به تفکیک تصاویر حاوی تابلو و بدون تابلو می‌پردازند و در نهایت با روش‌های مبتنی بر یادگیری عمیق به شناسایی تابلوها می‌پردازند. در مسائل مختلف ممکن است شرایط مسئله تفاوت داشته باشد. برای مثال، در مقاله والی و همکاران [2]، به وجود داده‌های متوالی اشاره شده است و استفاده از امکان دنبال کردن شکل‌ها فراهم است. در این مقاله به شرح روش‌های مختلف تشخیص علائم مبتنی بر رنگ، مبتنی بر تصویر و راهبردهای ترکیبی پرداخته شده است. به عنوان مثال، یکی از روش‌های پردازش مبتنی بر رنگ، آستانه‌گیری یا قراردادن حدنصاب است که براساس آن تصویر اولیه به تصویری دودویی تبدیل می‌شود. گسترش نواحی رنگی که منجر به ایجاد نواحی گردد نیز قابل ذکر است. همچنین، در

راستای بهره‌بری از اشکال، روش‌های متعددی وجود دارند. از جمله این روش‌ها به؛ تبدیل هاف، مقایسه دو تصویر، استفاده از لبه‌ها، شیب لبه‌ها و ... اشاره شده است. در مرحله دسته‌بندی نیز روش‌های متعددی وجود دارند که هریک مزایا و معایب مختص به خود را دارد. از جمله این موارد می‌توان به درخت تصمیم‌گیری، الگوریتم ژنتیک، تطبیق الگو، ماشین بردار پشتیبانی و شبکه‌های عصبی عمیق اشاره کرد.

کاوا و همکاران [3] نیز روشی مبتنی بر تغییر فضای رنگی از RGB به HSV ارائه کرده‌اند که امکان بهره‌بری از اطلاعات رنگی تصویر را فراهم می‌کند. گرچه هریک از این فضاها رنگی محدودیت‌ها و مزایای متعلق به خود را دارند اما این تفاوت‌ها فرصت بهره‌بری حداکثری از اطلاعات رنگی را فراهم آورده است. این مقاله در ادامه تلاش می‌کند تا با قراردادن حدنصاب، تصاویر اولیه را به تصویری دودویی تبدیل کرده و سپس، با پر کردن حفره‌های تصویر، نواحی نامزد تابلوها را مشخص کند. در ادامه، دسته‌بندی و شناسایی تابلو براساس مدل‌های یادگیری عمیق اتفاق می‌افتد. در نمونه دیگر، سوگیهارتو و همکاران [4]، از فضای رنگی HSI استفاده کرده‌اند. در این روش به وجود پس زمینه‌ها و امکان رخداد خطا در این مورد توجه خاصی شده است و در شبکه در حال تعلیم نوعی دسته بندی بین نواحی مورد علاقه درست و غلط یا همان عدم وجود تابلو ایجاد گردیده است. سپس بین تصاویری که برچسب مثبت خورده باشند از روش‌هایی مانند ماشین بردار پشتیبان چند دسته‌ای و روش  $k$  نزدیک‌ترین همسایه استفاده گردیده است. در این مقاله، استفاده از شبکه‌های مبتنی بر یادگیری عمیق از نوع ضرب پیچشی مطرح شده است.

تانگ و همکاران در پژوهشی دیگر [5]، با مطرح کردن استفاده از فضای رنگی YUV توانسته‌اند به نتایج مطلوب خود دست یابند. در این مقاله برای حد نصاب نیز پیشنهاد رنگی مربوط به این فضا ارائه گردیده است و به وجود رنگ‌های مشخص در تابلوها توجه شایانی شده است. در این مقاله نیز هیستوگرام شیب‌های جهت‌دار جهت تشخیص و ماشین بردار پشتیبانی در جهت شناسایی تابلوها استفاده گردیده است.

## راهبرد پیشنهادی

### استخراج تابلوهای راهنمایی و رانندگی

روش استفاده شده برای استخراج علائم راهنمایی و رانندگی از تصاویر چهار مرحله دارد. در مرحله اول، ابتدا تمامی پیکسل‌های قرمز تصویر با استفاده از آستانه‌گیری مقادیر پیکسل‌ها در فضای HSV استخراج می‌شوند. سپس لبه‌های تصویر حاصل با استفاده از یک روش تشخیص لبه استخراج می‌شوند تا با استفاده از توابع تشخیص

شکل‌های هندسی مانند دایره و مربع، علائم راهنمایی رانندگی استخراج شوند. در ادامه به جزئیات این روش می‌پردازیم.

## آستانه‌گیری از مقادیر پیکسل‌ها

فضای رنگی HSI بر پایه‌ی درک چشم انسان از رنگ‌هاست و در نتیجه، به راحتی می‌توانیم اطلاعات رنگ‌ها را از آن استخراج کنیم، بدون اینکه نگران نور و شدت اشباع رنگ باشیم. اما تصاویر ارائه شده به سامانه در فضای رنگی RGB هستند و پیچیدگی محاسباتی تبدیل فضای رنگی RGB به HSI بالاست. پس به جای HSI از HSV استفاده می‌شود که مشابه HSI است ولی استفاده از آن، پیچیدگی محاسباتی کمتری به همراه دارد. با آستانه‌گیری از مقادیر پیکسل‌ها در این فضای رنگی دو تصویر باینری به دست می‌آید. تصویر اول حاوی پیکسل‌هایی است که قرمزی آن‌ها در آستانه‌ی تعیین شده است و تصویر دوم حاوی پیکسل‌هایی است که شدت رنگ آبی در آن‌ها در آستانه‌ی تعیین شده است. دلیل آستانه‌گیری روی مقادیر این دو رنگ، رایج بودن آن‌ها در تابلوهای راهنمایی و رانندگی است. در ادامه با استفاده از عملیات OR منطقی، دو تصویر باینری به دست آمده را با هم ترکیب می‌کنیم.

## بستن مورفولاجیکال و تشخیص لبه

با توجه به اینکه ممکن است بخشی از مرزی تابلوها توسط دیگر اجسام حاضر در تصویر پوشانده شده باشد، جهت تکمیل بهتر این مرزها که در مراحل بعدی جهت تشخیص تابلو استفاده می‌شوند، از عملیات بستن مورفولاجیکال استفاده می‌کنیم. پس از این مرحله، از الگوریتم‌های تشخیص لبه جهت استخراج مرزهای اجسام حاضر در تصویر استفاده می‌کنیم. اوپراتورهای مختلفی برای این مرحله قابل استفاده هستند که ما از اوپراتور Canny استفاده می‌کنیم. با توجه به باینری بودن تصویر ورودی این اپراتور، حد آستانه‌ی آن اهمیتی ندارد.

```
hsv = cv2.cvtColor(image, cv2.COLOR_RGB2HSV)
r = np.logical_and(np.logical_or(hsv[:, :, 0] <= 10, hsv[:, :, 0] >= 170), hsv[:, :, 1] >= 100)
b = np.logical_and(np.logical_and(hsv[:, :, 0] <= 135, hsv[:, :, 0] >= 105), hsv[:, :, 1] >= 100)
rb = np.logical_or(r, b)
rb = (rb * 1.0).astype('uint8')

"""closing"""
rb = cv2.morphologyEx(rb, cv2.MORPH_CLOSE, np.ones((3,3), np.uint8))
```

شکل ۲ - برنامه‌ی استخراج پیکسل‌های قرمز و آبی و بستن مورفولاجیکال

## تشخیص تابلوها

جهت تشخیص تابلوها از روش‌های مختلفی برای هر نوع تابلو استفاده می‌کنیم. در ادامه به جزئیات این روش‌ها می‌پردازیم.

جهت تشخیص تابلوهای دایره‌ای شکل از متد HoughCircles پکیج OpenCV استفاده می‌کنیم. این متد تصویر خروجی الگوریتم تشخیص لبه را به عنوان ورودی دریافت کرده و با بررسی مرزهای تصویر، دایره‌های حاضر در تصویر را پیدا می‌کند. پس از تشخیص دایره‌ها با استفاده از این متد، تراکم پیکسل‌های رنگ‌های آبی، قرمز و سفید را در آن دایره بررسی می‌کنیم. هدف از این کار تشخیص دایره‌های متعلق به علائم راهنمایی رانندگی از سایر دایره‌های تصویر است. علاوه بر بررسی تراکم پیکسل‌های سفید، آبی و قرمز به کل پیکسل‌های دایره، مساحت دایره‌ها را نیز بررسی می‌کنیم تا در نهایت دایره‌های متعلق به تابلوها را از سایر دایره‌ها جدا کنیم.

```
def detect_circular_signs(image, edge, rb, image_num, save_path):

    circles = cv2.HoughCircles(edge,
                                cv2.HOUGH_GRADIENT,
                                1,
                                20,
                                param1 = 10,
                                param2 = 20,
                                minRadius = 0, # TODO: find the right parameters
                                maxRadius = 0)

    h, w = image.shape[:2]
    cop = image.copy()
    final_signs = []
    if circles is not None:

        circles = np.round(circles[0, :]).astype("int")

        for (x, y, r) in circles:
            sub_area = cop[y-r:y+r, x-r:x+r, :]

            if sub_area.shape[0] * sub_area.shape[1] == 0:
                continue

            total, whites, reds, blues = count_pixels(sub_area)

            area = 3.141592 * r * r
            margin = int(0.1 * r)

            if (blues/total > 0.3) and (whites/total > 0.15) and (area > 500) and (area < 6000):
                cv2.circle(cop, (x, y), r, (0, 255, 0), 2)
                ymin = max(0, y-r-margin)
                ymax = min(h, y+r+margin)
                xmin = max(0, x-r-margin)
                xmax = min(w, x+r+margin)
                x = image[ymin:ymax, xmin:xmax]
                final_signs.append(x)

            elif (reds/total > 0.2) and (whites/total > 0.3) and (area > 500) and (area < 6000):
                cv2.circle(cop, (x, y), r, (0, 255, 0), 2)
                ymin = max(0, y-r-margin)
                ymax = min(h, y+r+margin)
                xmin = max(0, x-r-margin)
                xmax = min(w, x+r+margin)
                x = image[ymin:ymax, xmin:xmax]
                final_signs.append(x)
```

شکل ۳ - متد تشخیص تابلوهای دایره‌ای شکل



جهت تشخیص تابلوهای مثلثی شکل از تعداد گوشه‌های کانتورهای حاضر در تصویر استفاده می‌کنیم. برای این کار، از متد `findContours` پکیج `OpenCV` استفاده کرده و تعداد گوشه‌های کانتورهای به دست آمده را با استفاده از متد `approxPolyDP` می‌شماریم. در ادامه با روشی مشابه تشخیص تابلوهای دایره‌ای شکل، مساحت مثلث‌های به دست آمده را بررسی کرده و مثلث‌های متعلق به تابلوها را استخراج می‌کنیم.

```
def detect_triangular_signs(image, edge, rb, iamge_num, save_path):

    ROI_number = 0
    cnts = cv2.findContours(rb, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    cnts = cnts[0] if len(cnts) == 2 else cnts[1]
    final_signs = []

    height, width = image.shape[:2]

    for cnt in cnts:
        approx = cv2.approxPolyDP(cnt, 0.25*cv2.arcLength(cnt, True), True)
        if len(approx) == 3:
            if cv2.contourArea(cnt) >= 500:
                x,y,w,h = cv2.boundingRect(cnt)
                margin = int(0.2 * w)
                ymin = max(0, y-margin)
                ymax = min(height, y+h+margin)
                xmin = max(0, x-margin)
                xmax = min(width, x+w+margin)
                final_signs.append(image[ymin:ymax, xmin:xmax])

    for i, s in enumerate(final_signs):
        cv2.imwrite(f'{save_path}/{iamge_num}-tri-{i}.png', s[:, :, ::-1])
    return final_signs
```

شکل ۴ - متد تشخیص تابلوهای مثلثی شکل

## خطاهای روش‌های تشخیص علائم راهنمایی و رانندگی

چالش‌های زیادی در سر راه تشخیص این علائم با روش‌های پردازش تصویر وجود دارد. از جمله‌ی این چالش‌ها می‌توان به پنهان شدن قسمتی از مرز تابلوها توسط اجسام خارجی اشاره کرد. پوشیده شدن قسمتی از مرز تابلو در روند تشخیص شکل مرز آن که بخش مهمی از عملیات استخراج تابلو از تصویر است، مشکل‌ساز است. همانطور که ذکر شد برای حل این مشکل می‌توان از بستن مورفولاجیکال استفاده کرد. اما استفاده از این روش باعث ایجاد مشکل در روند تشخیص تابلوهایی می‌شود که در کنار یکدیگر قرار گرفته‌اند. شکل ۵ نمونه‌ای از این تابلوها را نمایش می‌دهد.

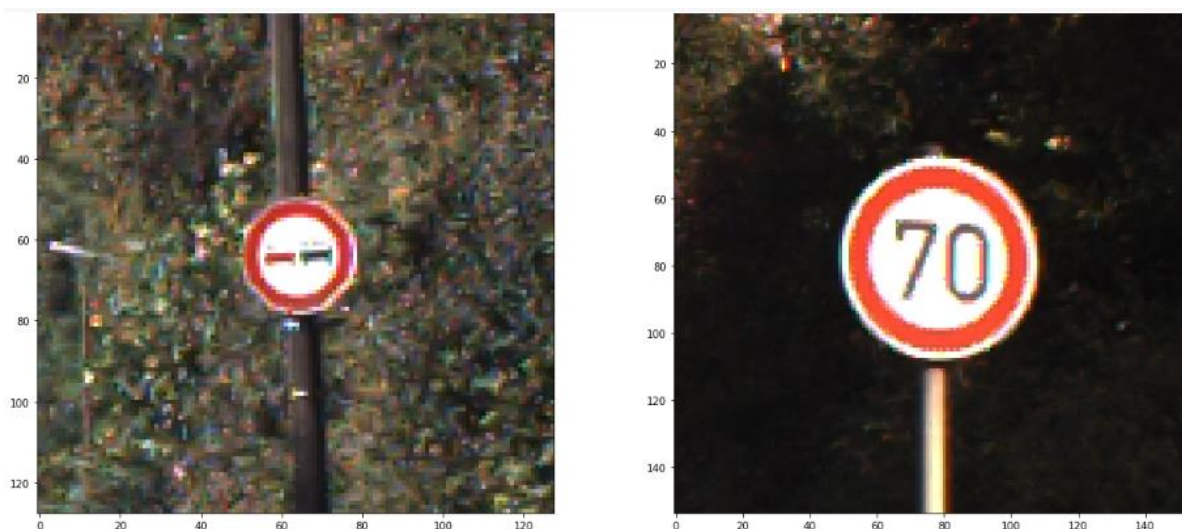


شکل ۵ - نمونه‌ای از قرار گرفتن تابلوهای راهنمایی و رانندگی در نزدیکی یکدیگر

اگر الگوریتم استخراج تابلوهای راهنمایی و رانندگی که در قسمت قبل ذکر شد را بر این تصویر اعمال کنیم، پس از انجام عملیات بستن مورفولاجیکال، مرزهای تابلوی مثلثی و تابلوی دایره‌ای شکل در هم فرو رفته که در روند تشخیص جداگانه‌ی این تابلوها مشکل‌ساز است. به همین دلیل، در این پژوهش، از تشخیص تابلوهایی که در کنار یکدیگر قرار گرفته‌اند، صرف نظر کردیم.

از دیگر چالش‌ها، پیچیدگی تشخیص مرز تابلوهای شش ضلعی است. با توجه به اینکه روند تشخیص شکل مرز اجسام حاضر در تصویر با شمردن گوشه‌های کانتورها انجام می‌شود، و با توجه به اینکه تعداد بالایی از اجسامی که مرز آن‌ها تعداد گوشه‌های زیادی دارد در تصویر حاضر هستند، تشخیص این تابلوها با خطای بالایی همراه است. اشکال مختلف از ترکیب مرزهای اجسام حاضر در جسم به وجود می‌آیند. از جمله تاثیرگذارترین این اجسام، شاخ و برگ درختان کنار جاده‌ها هستند که ممکن است کانتورهایی با تعداد گوشه‌های مختلف تولید کنند. این موضوع سبب بالا رفتن تعداد قسمت‌هایی از تصویر است که به اشتباه شش ضلعی تشخیص داده می‌شوند. به همین دلیل، در این پژوهش، از تشخیص تابلوهای شش ضلعی صرف نظر کردیم.

این چالش‌ها از جمله دلایلی بودند که بسیاری از تحقیقات انجام شده بر این مسئله مانند [6-8] تنها بر دسته‌های خاصی از تابلوهای راهنمایی و رانندگی مانند تابلوهای دایره‌ای شکل تمرکز کرده‌اند.



شکل ۶ - خروجی الگوریتم استخراج تابلوهای راهنمایی و رانندگی

## مجموعه دادگان

تابلوهای راهنمایی و رانندگی در کشورهای مختلف شکل‌های متفاوتی دارند. طراحی این تابلوها در کشورهای اتحادیه‌ی اروپا بر پایه‌ی طراحی‌های ساده داخل شکل‌های هندسی با رنگ پایه (آبی، قرمز و ...) است. از این ایده در کشورهای آمریکای شمالی هم استفاده می‌شود، با این تفاوت که علائم راهنمایی رانندگی این کشورهای حاوی متن بیشتری نسبت به کشورهای اتحادیه‌ی اروپا است که تفسیر این تابلوها را برای انسان‌ها ساده‌تر و برای کامپیوترها پیچیده‌تر می‌کند. شکل ۷ چند نمونه از تابلوهای مختلف مورد استفاده در اتحادیه‌ی اروپا را با معادل آن‌ها در آمریکا مقایسه می‌کند.

	EU	US
Warning		
Regulatory		
Obligatory		

شکل ۷ - مقایسه‌ی تابلوهای راهنمایی و رانندگی اروپا و آمریکای شمالی [9]

در این پروژه از مجموعه دادگان علائم راهنمایی و رانندگی آلمان استفاده می‌کنیم. این مجموعه دادگان حاوی ۹۰۰ تصویر دیجیتال است که هر کدام حاوی ۰ تا ۶ تا از علائم راهنمایی و رانندگی هستند، می‌باشد. تصاویر به فرمت ppm ذخیره شده‌اند و ابعاد آن‌ها از  $16 \times 16$  تا  $128 \times 128$  متغیر است. در این تصاویر، تابلوهای راهنمایی و رانندگی از زوایای مختلف و در نورهای متفاوت دیده می‌شوند.

تابلوهای راهنمایی و رانندگی در این مجموعه دادگان به شکل‌های مختلفی در کنار یکدیگر قرار می‌گیرند. ممکنه است یک تابلو به تنهایی و یا در کنار تابلوهای دیگر قرار گرفته باشد. نحوه قرارگیری تابلوها در کنار یکدیگر میتواند در روند تشخیص آن‌ها تاثیرگذار باشد.

## دسته‌بندی تابلوها

زمانی که خروجی مرحله قبل ایجاد گردد نیاز به سامانه‌ای است که تشخیص دهد تابلو پیدا شده از چه دسته‌ای از تابلوها می‌باشد. در این راستا نیاز به تفکیک داده‌هایی که وجود تابلو درست تشخیص نداده‌اند، حس می‌شود. چنین پالایشی برای بهبود دقت سامانه و مقاومت بیشتر در برابر خطای احتمالی مرحله قبل بسیار مطلوب است. در نتیجه در هنگام تشخیص نوع تابلو دسته‌ای به این منظور تعریف گردیده است.

در ادامه به بررسی این بخش از سامانه می‌پردازیم. تابلوها هر یک شامل ویژگی‌های منحصر به فردی هستند که می‌توانند در سنجش دسته هر تابلو مورد استفاده قرار گیرند. بخشی از این دسته‌ها مانند آنچه در بخش قبلی گفته شد می‌باشند و از طرفی بخش دیگری از ویژگی‌ها که مبتنی بر حوزه فرکانسی هستند نیز معرفی شده‌اند. طبق آنچه در مطالعات مقالات مختلف انجام شد، حوزه زمان بسیار محبوب‌تر است اما با این وجود حوزه فرکانسی نیز می‌تواند کمک کننده باشد. به ویژه این موضوع در تشخیص تابلو از غیر تابلو در عمل موفقیت آمیز بوده است چرا که با توجه به زمینه ساده تابلوها و این که اغلب خطا در فضاهای شلوغ مثلاً به خاطر وجود درختان رخ داده است، می‌توان انتظار وجود سیگنال‌های ریزتر در مواردی که تابلویی وجود ندارد را داشته باشیم. طبق مشاهداتی که داشتیم خروجی حاصل از تبدیل تصویر به فضای HSV توانست نتایج بهتری داشته باشد که علت آن نیز مشابه توضیحات قبلی می‌باشد.

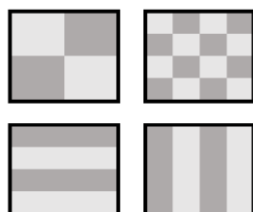
در حوزه فرکانسی ابتدا از ماسکی مانند شکل ۸ که هر رنگ نماینده یک ناحیه است استفاده گردید. این مورد این مزیت را ایجاد میکند تا فرکانس‌هایی که رفتار مشابهی دارند در یک دسته قرار بگیرند. این مورد در حوزه زمان به صورت مشبک‌های مستطیلی شکل انجام می‌گیرد.



شکل ۸ - ماسک فرکانسی

همچنین در حوزه زمان نیز ویژگی‌های مختلفی استخراج شده است. در این مورد ابتدا تصویر بر اساس یک ترشولد به صورت دودویی تبدیل می‌شود. سپس تابعی در جهت محاسبه اندازه میانگین از نظر عددی پیکسل‌های هر ناحیه ایجاد شده است. برای این منظور به طرق مختلفی ناحیه بندی انجام می‌شود.

- تقسیم تصویر به ۱۶ ناحیه به صورت  $4 \times 4$
- تقسیم تصویر به ۴ ناحیه به صورت  $2 \times 2$
- تقسیم تصویر به ۴ ناحیه به صورت  $4 \times 1$
- تقسیم تصویر به ۴ ناحیه به صورت  $1 \times 4$
- خروجی اکتشاف لبه‌های افقی
- خروجی اکتشاف لبه‌های عمودی
- خروجی اکتشاف لبه‌های کج چپ
- خروجی اکتشاف لبه‌های کج راست



شکل ۹ - روش‌های مختلف ناحیه بندی

توجه شود همه این موارد به صورت نرمالایز شده هستند و در بازه  $[0,1]$  قرار دارند. همچنین در صورت وجود کانال‌های مختلف کانال‌ها تفکیک شده و هر کانال به صورت مستقل بررسی می‌گردد. این مورد در ویژگی‌های حوزه فرکانس هم وجود دارد. همچنین خود پیکسل‌ها نیز به صورت به خط شده اضافه گردیده است. برای ایجاد دسته بند از ویژگی‌های استخراج شده، روش‌های مختلفی وجود دارد که از جمله موارد ساده آن می‌توان به ام ال پی ساده، دسته بند درخت تصمیم و دسته‌بندی با KNN اشاره کرد که مورد آخر در حالی که کمترین پیچیدگی زمانی را داشت، از نظر عملکردی بهترین نتایج را حاصل نمود. بر اساس فیچرهای ذکر شده، کلاس بندی را بر روی دیتاستی آماده با استفاده از KNN انجام دادیم که دقت  $0.98$  به دست آمد. در مرحله ی دوم داده های استخراج شده را با داده های از پیش آماده مخلوط کردیم و نتایج را بر اساس فیچر های مختلف مقایسه کردیم . که در جدول زیر قابل مشاهده است.

	Frequency features HSV	Frequency features	HSV features	Hog features	Accuracy
Method 1					۰.۹۱۵
Method 2					۰.۹۱۴
Method 3					۰.۹۱۱

۰.۹۴۱					Method 4
۰.۹۷۰					Method 5

البته از آن جایی که داده‌های تست و آموزش به صورت تصادفی استفاده می‌شود و اختلاف کم بین دقت‌ها در این مقایسه چندان قابل استناد نیست. به علاوه برای استفاده به صورت عمومی نیاز است که بر اساس یک سری داده‌ی از پیش آماده شده آموزش انجام شود و با ورود هر داده‌ی جدیدی سامانه بتواند آن را برچسب گذاری نماید و نمی‌توان هر بار بر اساس تابلوهای استخراج شده آموزش را انجام داد به همین دلیل با استفاده از داده‌هایی که از پیش داشتیم آموزش انجام شد و حالا تست بر روی تابلوهای استخراج شده می‌باشد. اگر بخواهیم جدولی مانند جدول فوق بر این اساس رسم کنیم نتیجه‌ی زیر حاصل می‌شود. داده‌های استفاده شده در این بخش، شامل داده‌های اضافی مرحله‌ی تعلیم در [این پوشه](#) موجود است.

Accuracy	Hog features	HSV features	Frequency features	Frequency features HSV	
۰.۹۱۵					Method 1
۰.۹۲۴					Method 2
۰.۹۳۳					Method 3
۰.۹۳۳					Method 4
۰.۹۴۳					Method 5

## مات شدگی و رفع مات شدگی

### کلیات

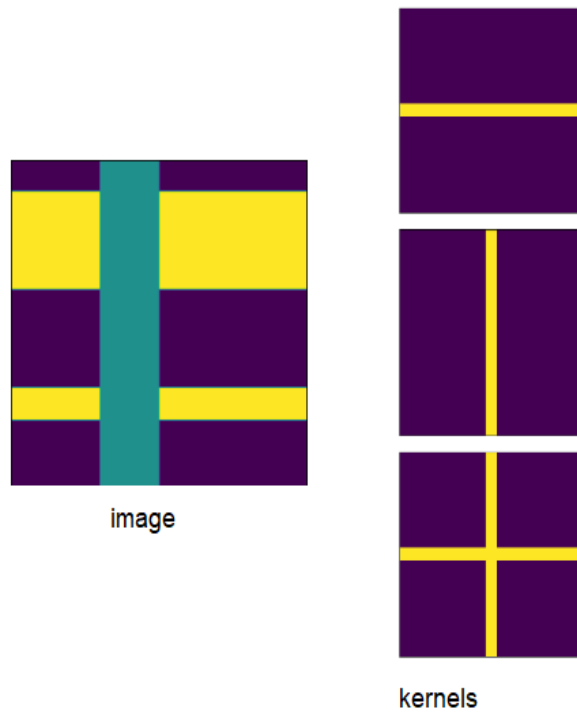
مات شدگی و رفع آن معادل با کلمات blurring و deblurring هستند. اصطلاح مات شدگی یک از چالش‌های مهم تصویربرداری و پردازش تصویر است. در این دست موارد دقت و نکات ریز تصویر تخریب شده و تصویر دچار ابهام می‌گردد. از جمله مهم‌ترین تخریب‌های این مشکل تخریب مرزها می‌باشد به نحوی که به صورت ساده در شکل ۱۰ مرز سمت راست به مرز سمت چپ تبدیل گردیده است.



شکل ۱۰ - تخریب مرزها

این مورد می‌تواند علل مختلفی داشته باشد. برای نمونه حرکت دوربین و جسم نسبت به هم (حرکت دوربین، جسم و یا هردو) می‌تواند علت این ماتی گردد چرا که دریچه دریافت نور دوربین در یک بازه زمانی فعال خواهد بود و این حرکت می‌تواند مؤثر باشد. در این صورت هرچه حرکت نسبی بیشتر باشد یا مدت زمان دریافت نور دوربین بیشتر باشد تصویر ماتی بیشتری خواهد داشت.

برای ایجاد شهود بهتر به شکل ۱۱ دقت فرمایید. در این تصویر در سمت چپ نمونه تصویر ساده و در سمت راست سه کرنل برای ایجاد ماتی در دو جهت افقی و عمودی و هر دو وجود دارد که با کمک کانوولوشن در تصویر، ماتی ساده ایجاد می‌کنند که شبیه ساز حرکت ماشین براساس تغییرات جاده می‌توانند باشند. برای اعمال این فیلتر توابع مختلفی در کتابخانه‌های مختلف پایتون پیاده سازی شده است که آن‌ها را در شکل ۱۲ می‌بینید. در هر حال با انتخاب یکی از این موارد و اعمال کرنل‌های فوق تصاویری مشابه شکل ۱۳ خواهیم داشت که به ترتیب از چپ به راست مات شدگی افقی، عمودی و کلی را نمایش می‌دهند.



شکل ۱۱ - تاثیر کرنل‌های مختلف در ایجاد ماتی در تصویر

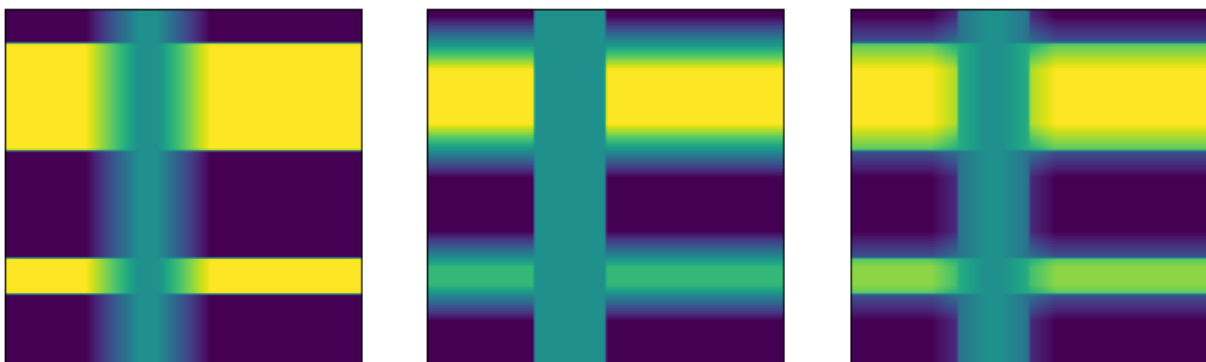
```
from scipy.ndimage import convolve
img = convolve(im,psf)

from scipy.signal import convolve2d
img = convolve2d(im,psf,'same')

img = cv2.filter2D(src=image, ddepth=-1, kernel=psf)

from PIL import Image, ImageFilter
im = Image.open(path)
img = im.filter(ImageFilter.Kernel((3, 3),
(1, 1, 1, 1, 1, 1, 1, 1, 1), 1, 0))
```

شکل ۱۲ - پکیج‌های حاوی پیاده‌سازی‌های کرنل‌های ماتی در پایتون



شکل ۱۳ - انواع مات شدگی با توجه به جهت کرنل. به ترتیب از راست به چپ: مات شدگی کلی، عمودی و افقی



یکی دیگر از عوامل مهم مات شدگی زوم کردن بر روی تصویر است. برای نمونه شکل ۱۴ به وضوح مات شدگی و کاهش کیفیت در هنگام زوم بر روی تصویر سمت چپ، در تصویر سمت راست قابل مشاهده است.



شکل ۱۴- تاثیر زوم کردن بر مات شدگی

البته موارد دیگری نیز می‌تواند منجر به ماتی گردد. مثلاً فشردن سازی تصویر، کیفیت پایین تصویر برداری و شرایط محیطی. از طرفی، این مشکل همراه با وجود نویز می‌باشد که دلایلی مختلفی می‌تواند داشته باشد. از جمله این دلایل کیفیت دوربین، شرایط محیطی مانند ذرات معلق در هوا و کمبود نور را می‌توان برشمرد. در راستای بهبود تصاویر از نظر کاربر نیاز به رفع چنین مشکلی بدیهی است. از طرفی در پردازش تصویر نیز این مورد اهمیت بسیاری دارد به نحوی که می‌تواند در عملکرد سیستم تاثیر بسیاری داشته باشد.

آنچه به نظر بدیهی می‌رسد اکتشاف کرنل مات کننده و برطرف کردن آن است. مثلاً اعمال کرنل معکوس برای بازبازی تصویر. اما چنین چیزی در عمل امکان پذیر نیست. در این زمینه ممکن است بخشی از اطلاعات به کلی از دست رفته باشند مانند حذف فرکانس‌های بالا در فشردن سازی تصویر و یا عدم ثبت جزئیات ریز در هنگام تصویر برداری که به ویژه در مورد زوم کردن می‌تواند چالش جدی باشد. این مشکل در مواردی که نیاز به پردازش لبه و اکتشاف اشکال است نمود زیادی دارد و می‌تواند عملکرد سیستم را تحت تأثیر قرار دهد. اما آیا امکان انجام ایده مطرح شده در اول پاراگراف ممکن است؟ جدای از اینکه الزاماً همه اطلاعات در دسترس نیست، پیچیدگی کرنلی که تصویر را مات کرده است در نمونه‌های عملی بسیار زیاد است به جدی که برای انجام تقریب خوب از آن نیاز است تا محاسبات پیچیده‌ای انجام شود یا در صورت استفاده از الگوریتم‌های تخمین بیشینه شباهت با تصویر اصلی، نیاز به محاسبات بسیاری است به نحوی که در انجام یک نمونه تست براساس کاهش گرادیان، برای یک کرنل گوسی، برای رسیدن به دقت بالای ۹۸ درصدی در اجرای ۴۰ نمونه مختلف با کرنل‌های با ابعاد رندم در هر بار بیش از ۱۰۰ دور اجرا و به روز رسانی کرنل تخمین زده شده مورد نیاز بود. در این مورد روش مدنظر با

الگو گرفتن از مرجع [12] پیاده سازی شده است و به طور کلی شامل مواردی است که در شکل ۱۵ مشاهده می کنید.

```

"""Loss function:
difference between conv(deblurredimage, kernel) and blurred_image

Gradient function:
gradient descent function for conv(deblurred_image,kernel) and blurred"""

deblurred_image = blurred_image.copy()
while(converge):
    | deblurred_image -= learning_rate*gradient(deblurred_image)

```

شکل ۱۵

روش های رفع ماتی تصویر به دو دسته چشم باز و چشم پوشیده هستند. در مورد روش های چشم باز، الگوریتم با در دست داشتن تصویر سالم تلاش میکند تا کرنلی را تخمین بزند که بتواند تصویر مات را به تصویر سالم تبدیل کند و در صورت یافتن چنین موردی به نظر می رسد مسئله حل شده است. در این زمینه روش های مبتنی بر یادگیری عمیق توانسته اند موفقیت در خوری بدست آورند و البته روش های ساده تری نیز وجود دارند که تلاش کرده اند به نوعی این مسئله را حل کنند. برای نمونه یک ایده ساده این است که اگر تصویر سالم را  $i$  بنامیم و کرنل را  $k$  بنامیم. در صورتی که تصویر سالم با اعمال کرنل مات شده باشد، تصویر مات شده را  $b$  بنامیم. همچنین تصویر بازیابی شده را  $\tilde{I}$  بنامیم. پس داریم:

$$\begin{aligned}
 I &= FFT(i) \\
 B &= FFT(b) \\
 K &= FFT(k) \\
 assume \otimes &= convolution\_function \\
 b &= i \otimes k \\
 \Rightarrow B &= I * K \\
 \Rightarrow I &= K^{-1} * B
 \end{aligned}$$

اما وجود نویز  $\eta$  مشکلی ایجاد می کند:

$$\begin{aligned}
b &= i \circledast k + \eta \\
\Lambda &= FFT(\eta) \\
\Rightarrow B &= I \circledast K + \Lambda \\
\Rightarrow I &= K^{-1} \circledast (B - \Lambda) \\
\Rightarrow \begin{cases} \text{kernel\_inverse\_estimation : } \tilde{K} = K^{-1} = I \circledast (B - \Lambda)^{-1} \simeq I \circledast B^{-1} \\ \text{restored\_image : } \tilde{I} \simeq \tilde{K} \circledast B \end{cases}
\end{aligned}$$

این در حالی است که وجود نویز از تابع خاصی پیروی نمی کند و از طرفی در تبدیل فوریه با توجه به جزئیات ریز اثرگذار است.

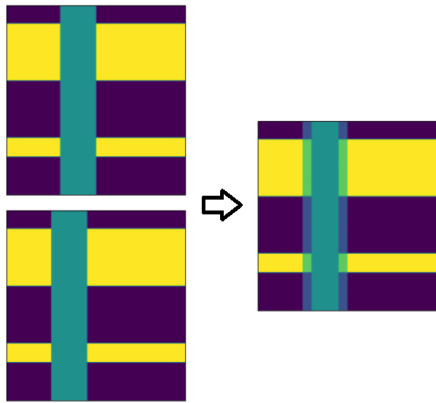
در هر حال گرچه می توان به تقریب تصویر را بازیابی کرد اما تقریب دقت کافی نخواهد داشت و برای نویز نیز محاسبه تابع و تبدیل فوریه آن با توجه به رفتار اغلب تصادفی عملاً میسر نیست.

از طرفی چنین فرضی در صورت ثابت بودن کرنل مات کننده در نمونه های مختلف اسیت. در صورتی که این فرض برقرار باشد می توان ادعا کرد به نحوی  $\tilde{K}$  محاسبه شده است و می توان داده های تست را به کمک آن بازیابی کرد. در هر حال مسئله پیش رو ما با توجه به تغییر شرایط جاده و حرکت ماشین با کرنلی متغیر رو به رو است و همچنین محاسبه تبدیل فوریه ها و کرنل واروون همیشه ساده نیست و حتی در صورتی که در محاسبات در حوزه زمان باشیم در مواردی محاسبه  $\tilde{K} = K^{-1}$  ناممکن است.

در مورد دیگر برخلاف روش های چشم باز، روش های چشم بسته وجود دارد که امکان محاسباتی مانند مورد فوق ممکن نخواهد بود. در مسئله ما با توجه به اینکه داده تعلیم وجود دارد و عامل مات کنندگی و نویز تصادفی را شبیه سازی نموده ایم می توانیم به روش های چشم باز بپردازیم اما در مسئله اصلی عملاً تصاویر موجود همان داده های مات هستند و محدودیت به مراتب بیشتری وجود خواهد داشت.

در ادامه به بررسی برخی راهبردها اشاره گردیده است. ابتدا یک روش تماماً چشم بسته مورد بررسی و ارزیابی قرار گرفت. سپس در ادامه تلاش شد تا تخمینی از کرنل بازیابی تصویر ایجاد نماییم. سپس نمونه هایی از روش های کاهش نویز و استفاده از فیلتر وینر مورد بررسی قرار خواهد گرفت. برای ارزیابی یک کرنل عمومی چند معیار جهت بررسی اثرگذاری مورد بررسی قرار خواهد گرفت. با بررسی عوامل فوق تلاش شد تا مانند موردی که پیشتر معرفی شد روشی مبتنی بر تکرار ارائه گردد. همچنین در این روش برای بهبود عملکرد و تطابق با مسئله مدنظر این پژوهش، از برخی موارد قبلی کمک گرفته شده است.

## بررسی یک روش تماماً چشم پوشیده



شکل ۱۶ - ترکیب شبه فریم‌های متوالی

در صورتی که بخواهیم تصویری واضح از تصویر مات شده ارائه دهیم، یک ایده این است که داده‌های افزونه حذف گردد.

برای نمونه فرض کنید شکل ۱۶ دو شبه فریم متوالی است که بر اثر حرکت ایجاد شده اند و هردو این موارد در یک تصویر ثبت شده اند و از ترکیب این دو تصویری مات مانند تصویر سمت راست بوجود آمده است.

پس عملاً نوعی افزونگی وجود دارد و در صورتی که بتوان به نوعی ماتریسی بازیابی کرد که حداقل وابستگی بین داده‌ها را داشته باشد می‌توان امیدوار بود که این مات شدگی از بین برود. برطبق مقاله [11] یک روش استفاده از تجزیه مقادارهای منفرد می‌باشد. در این مقاله با کمک راهبرد تجزیه مقدار منفرد کوتاه شده می‌باشد. معادل دو عبارت فوق به اختصار SVD و TSVD می‌باشد. در پژوهش انجام شده در [11] به این نکته توجه گردیده است که روش‌هایی مانند فیلتر وینر مبتنی بر وجود داده‌های چشم باز می‌باشند و روشی که در ادامه توضیح داده می‌شود مستقل از داده‌های چشم‌باز است چراکه معیار موفقیت آن براساس محاسبات جبری است و نیازی به قیاس با داده‌های سالم ندارد. در تجزیه مقدار تکین عبارتی به صورت زیر داریم:

$$A = U\Sigma U^T = \sum u_i \sigma_i v_i^T$$

برطبق محاسبات تجزیه مقدار تکین کوتاه شده و با در نظر گرفتن اینکه  $K = A$  باشد با توجه به علامت گذاری که در ابتدای بحث معرفی گردید می‌توانیم بنویسیم:

$$\tilde{i} = K^{-1}g = A^{-1}g = \sum \frac{u_i^T g}{\sigma_i} v_i$$

کتابخانه‌های پایتون توابع مورد نیاز برای محاسبات فوق را ارائه کرده اند و هرچند روش فوق به وضوح دارای تقریب است اما نتایج حاصل شده بسیار ضعیف‌تر از انتظار بود. برای نمونه در شکل ۱۷ از چپ به راست تصویر سالم، تصویر مات شده و تصویر بازیابی شده توسط این روش قابل مشاهده است.



شکل ۱۷ - به ترتیب از چپ به راست: تصویر سالم، تصویر مات و تصویر بازیابی شده با استفاده از روش ارائه شده در پژوهش [11]

## بررسی روش‌های مبتنی بر اعمال فیلتر از قبیل فیلر وینر و فیلتر شارپ کننده

پیش‌تر گفته شد تلاش برای ارائه یک فیلتر برای تبدیل از تصویر مات به تصویر اصلی می‌تواند ایده آل باشد. در ادامه تلاش شده است تا فیلتری عمومی برای این هدف ارائه گردد.


اولین ایده استفاده از فیلترهای شارپ کننده است. این دست روش‌ها در زمان وجود نویز، باعث تقویت نویز می‌شوند که مطلوب نیست و از طرفی با توجه به اینکه هدف اصلی مسئله حل مشکل ماتی است، استفاده از فیلترهای پایین‌گذر برای حذف نویز مطلوب نیست. در شکل ۱۸ یک تصویر سالم و تصویر مات شده پس از اعمال فیلتر میانگین‌گیر  $3 \times 3$  ساده می‌باشد.



شکل ۱۸ - تصویر سالم و مات شده با استفاده از فیلتر میانگین‌گیر  $3 \times 3$

برای ساخت داده‌های مورد بررسی، نیاز به اعمال کرنلی برای مات کردن تصویر است. همچنین در ادامه نویزی رندم برای پیکسل‌ها اعمال می‌شود تا در شرایط مناسب‌تری مسئله بررسی گردد. شکل ۱۹ خروجی این روش را نشان می‌دهد.

$$\text{Blur Filter} : \frac{1}{9} \begin{bmatrix} 1, 1, 1 \\ 1, 1, 1 \\ 1, 1, 1 \end{bmatrix}$$

$$\text{Sharp Filter} : \begin{bmatrix} -1, -1, -1 \\ -1, 9, -1 \\ -1, -1, -1 \end{bmatrix}$$


شکل ۱۹ - از راست به چپ: خروجی کرنل مات کردن تصویر و کرنل‌ها

این مورد نتوانست به تنهایی نتایج مدنظر را بدست آورد. در این باره در بخش معرفی راهبردهای ارزیابی بیشتر توضیح داده خواهد شد. شکل ۲۰ وجود نویز تصادفی را در تصویر بررسی می‌کند.

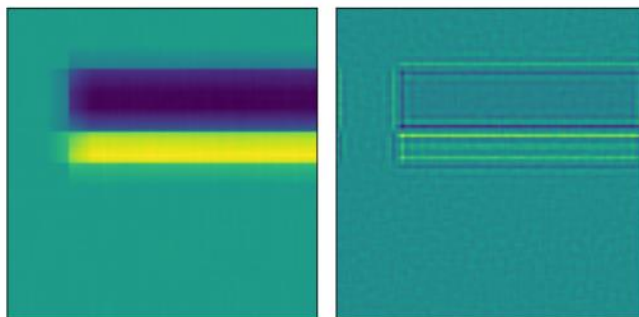


شکل ۲۰ - نویز تصادفی در تصویر

می‌توان نویزهای تقویت شده که در تصویر مات چندان محسوس نبودند را در این شکل به وضوح دید. حال فرض کنید این نویز بر روی تصویر خروجی بعد از مات شدگی نشسته باشد (مثلا خطای سیستم) در این صورت با شرایط به مراتب بدتری رو به رو خواهیم بود. این مورد به ویژه در زمانی که روش‌هایی مبتنی بر تشخیص شکل و به ویژه خط باشند بسیار نمود پیدا می‌کند. در روش معرفی شده در این پژوهش نیز به ویژه برای تشخیص اشکال چنین عاملی می‌تواند نقش بسیار مخربی ایفا نماید.

با بررسی مقالات مختلف به این نتیجه رسیدیم که بهترین گزینه مطرح شده که نیازی به شبکه‌های عصبی عمیق هم نداشته باشد، استفاده از روش فیلتر وینر برای مدیریت نویز می‌باشد [10-12].

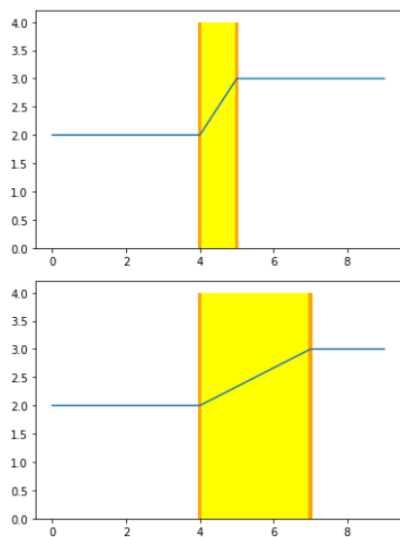
در این روش می‌توان کاهش اثر نویز را مشاهده کرد و اثر این فیلتر نیز پس از معرفی راهبردهای ارزیابی این روش‌ها، بررسی خواهد شد. در استفاده از فیلتر وینر اما یک مشکل وجود دارد و آن کاهش کیفیت مرزها نسبت به حالت قبل است که رخداد چنین موردی با توجه به خاصیت فیلتر وینر گرچه نسبت به یک فیلتر پایین گذر بسیار کمتر است اما به هر حال باید مورد توجه قرار بگیرد.



شکل ۲۱ - کاهش اثر نویز و کیفیت مرزها در خروجی فیلتر وینر

در این راستا یک سوال مطرح می‌شود. آیا در صورتی که مرزها تخریب شده باشند ولی نویزها مدیریت شده باشند این روش ارزش دارد؟ پاسخ به این مورد بله مشروط است. در خط لوله مدنظر این پژوهش نویزها می‌توانند بسیار مخرب باشند و همچنین مرزها اهمیت زیادی دارند. پس شارپ نمودن تصویر و مدیریت نویز باید انجام گیرد و فیلتر وینر در این بخش به ما کمک شایانی می‌کند. این فیلتر با مدیریت نویز امکان دریافت یک تصویر شارپ بهتر را فراهم می‌کند. اما هنوز بحث ضعف مرزها بسیار مهم است.

ایده: در خط لوله این پژوهش در گام بعدی، نیاز است تا در تصویر خطوط و اشکال استخراج گردند تا بتوان به کمک آن تابلوها را بررسی نمود. پس مرزهای اصلی می‌توانند بسیار مهم باشند. الگوریتم canny درحالی می‌تواند مرزهای قوی را تشخیص دهد که وجود نویزهای ریز چندان مشکل را نمی‌باشد، چراکه در این دست نویزها، عملاً اگر مرزی تشخیص داده شود نیز به عنوان مرز ضعیف حذف می‌گردد. پس ما یک مرز قوی و نسبتاً سالم می‌توانیم داشته باشیم. اگر این مرز در ابتدا استخراج گردد و در نهایت به تصویر افزوده شود در حالی که شاید تا حدودی از تصویر اصلی دور شویم اما بدون شک به افزایش بهره‌وری ادامه خط لوله خواهد انجامید. مورد دیگری که توجه



شکل ۲۲ - تاثیر ماتی بر تشخیص مرز

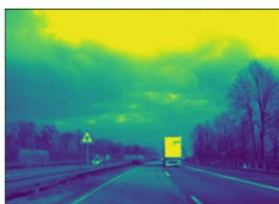
به آن خالی از لطف نیست، فاز نهایی خط لوله این پژوهش است، در آن مرحله از استخراج ویژگی‌های سیگنالی استفاده می‌گردد. بدیهی است که تقویت مرزهای اصلی و کاهش نویز در کنار هم به منزله تقویت ویژگی‌هایی که به مرزهای اصلی مرتبط می‌شوند و کاهش خطاهای فرکانس‌های بالا که ناشی از نویز است می‌انجامد.

در مورد مرزها یک مشکل اساسی وجود دارد. در صورتی که تصویر دچار مات شدگی شدید باشد تشخیص مرز یک چالش جدی خواهد بود به نحوی که مطابق شکل ۲۲، تمامی ناحیه زرد رنگ به نحوی می‌تواند مرز

باشد. حال هرچه این شیب کمتر باشد ناحیه زرد بیشتر و گزینه‌های مرز بیشتر وجود خواهد داشت و تشخیص مرز نادقیق‌تر خواهیم داشت.

در ادامه از نمونه‌های مجموعه داده اصلی استفاده خواهد شد. در این راستا یک کرنل مات کننده و یک عامل نویز تصادفی برای ایجاد نمونه مات شده استفاده گردیده است که عامل و خروجی آن را در شکل ۲۳ می‌بینید.

```
psf = np.ones((5,5)) / 5*5
im2 = convolve2d(im,psf,'same')
im3 = im2 + 0.02 * im2.std() * np.random.standard_normal(im2.shape)
```



شکل ۲۳ - عامل نویز تصادفی برای ایجاد تصویر مات و خروجی آن

شکل ۲۴ تنها نویز تصادفی حاضر در تصاویر شکل ۲۳ را نمایش می‌دهد.



شکل ۲۴ - نویز تصادفی حاضر در تصاویر شکل ۲۱



در این پژوهش برای فیلتر وینر از کتابخانه scikit استفاده شده است که نحوه استفاده از آن را در شکل ۲۵ می‌بینید.

```
from skimage import restoration
deconvolved_img = restoration.wiener(img, psf, 1100)
```

شکل ۲۵ - فیلتر وینر کتابخانه scikit

همچنین روش دیگر به صورت شکل ۲۶ می‌باشد.

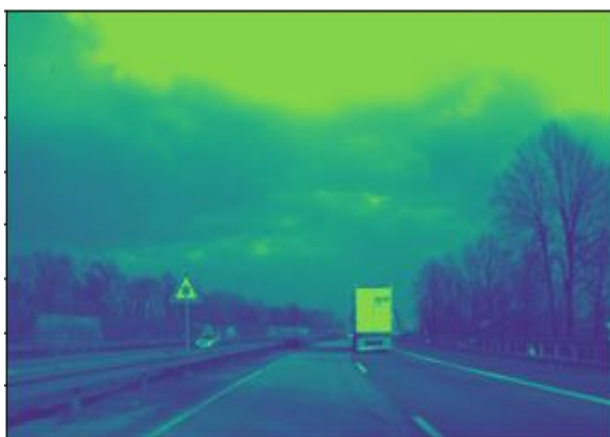
```
deconvolved, _ = restoration.unsupervised_wiener(img, psf)
```

شکل ۲۶

در محاسبات مد نظر ما روش دوم مورد استفاده قرار گرفته است. تفاوت این دو روش در این است که مدل دوم به صورت خودکار بحث منظم‌سازی را انجام می‌دهد. فیلتر وینر در حالی که نویز را تضعیف میکند که سعی میکند ساختار اصلی حفظ گردد. از طرفی این فیلتر به نحوی یک ضد کانوولوشن است و با توجه اینکه چه کرنلی به عنوان ورودی آن مورد استفاده قرار گیرد این مشکل را نیز خنثی می‌کند و در نتیجه به بهبود میزان ماتی تصویر کمک می‌کند.

در نمونه کدی که در بالا آورده شده است پارامتر سوم یک پارامتر جهت منظم سازی مقادیر پارامترها می‌باشد که به تعادل بین بهبود جهت بازیابی سیگنال اصلی و حفظ حالت قبلی یا همان پیشگیری از ورود نویز است می‌انجامد.

در نمونه تصاویری که در شکل ۲۳ آورده شد با اعمال این فیلتر شکل ۲۷ حاصل می‌گردد. با دقت در تصویر می‌توان کاهش میزان ماتی را مشاهده کرد.

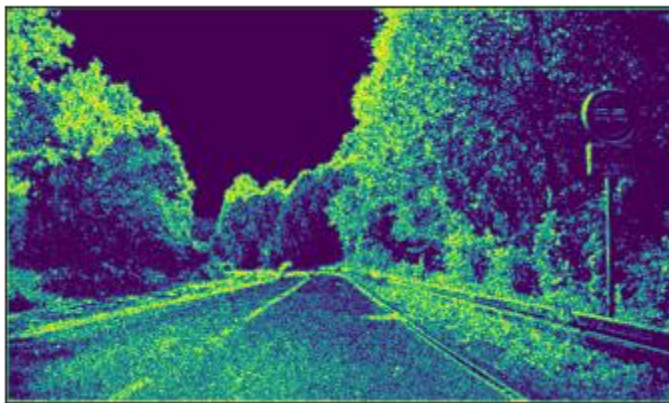


شکل ۲۷ - تصویر بازسازی شده از تصویر مات

## روش‌های تکرار شونده

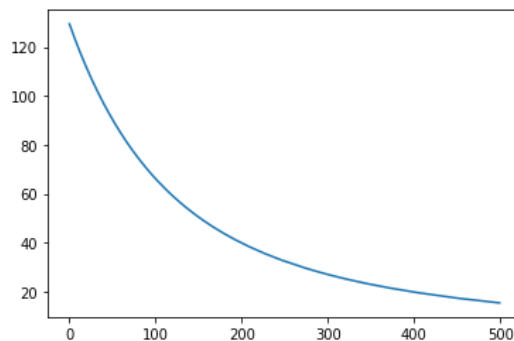
برخی روش‌هایی که می‌توان برای حل مسئله به کار گرفت به جای یک بار حل کامل به صورت مرحله‌های تکرار شونده حل مسئله می‌کنند. در این صورت به طور کلی می‌توان گفت که به مرور زمان الگوریتم بهبود می‌یابد. در این دسته به دو روش اشاره می‌نماییم. روش اول همان موردی است که به صورت کلی در قسمت ابتدایی همین بخش یعنی بخش مات شدگی و رفع آن آورده شده است. این روش با محاسبه یک گرادیان سعی می‌کند تصویری را بسازد که با اعمال کرنل به تصویر مات شده بیشترین شباهت را داشته باشد. در این صورت می‌توان امید داشت که تصویر اولیه را تخمین خواهد زد. این روش به صورت چشم بسته عمل می‌کند و از این نظر مورد توجه ما می‌باشد. اما از طرفی این روش در صورت دانش اولیه از حدود کرنل عملکرد به مراتب مطلوب‌تری خواهد داشت ولی در این مسئله با توجه به عدم دانش از این مورد ترجیح داده شد کرنل کوچک  $3 \times 3$  در نظر گرفته شود تا بتواند جزئیات بیشتری را نمایش دهد. در این راستا تلاش شد تا برخی موارد ترکیب گردد اما هیچ یک علارغم افزایش بار محاسباتی و پیچیدگی زمانی، نتوانست تفاوت محسوسی در خروجی ایجاد نماید. شاید یک دلیل آن استفاده از داده‌های کمکی عمومی بود که در مقابل وجود گرادیان که رفتار محلی تری دارد انعطاف کافی نداشته است.

از جمله این روش‌ها استفاده از تصویر کمکی از مرزهای استخراج شده قوی‌تر در هر مرحله از تعلیم و بررسی اثر آن که در صورت اثر مثبت اعمال گردد را می‌توان نام برد. همچنین استفاده از نوع دیگری از محاسباتی تابع هزینه، استفاده از خروجی فیلتر وینر به عنوان تصویر اولیه و تست تقویت مرزهای قوی تصویر را می‌توان برشمرد. شکل ۲۸ اختلاف خروجی و تصویر اولیه که مرزها به وضوح نتوانسته‌اند به خوبی بازیابی شوند را نمایش می‌دهد.



شکل ۲۸ - اختلاف خروجی و تصویر اولیه

همچنین شکل ۲۹ کاهش میزان شکست در تکرارهای متوالی را نمایش می‌دهد.



شکل ۲۹ - کاهش میزان شکست در تکرارهای متوالی

در ادامه روش دوم یعنی روش‌های مبتنی بر یادگیری عمیق استفاده شد. استفاده از MLP در اینجا چندان خوشایند نیست چراکه کرنل ماتریس وزن شبکه با توجه به سائز تصویر بسیار زیاد می‌شود. پس روش مبتنی بر ضرب پیچشی به کار گرفته شد به طوری که ورودی و خروجی هر دو تصویر باشند و شبکه باید بیاموزد تا از تصویر مات به تصویر سالم برسد. سپس در زمان استفاده شبکه با دریافت تصاویر مات، تصاویر بهبود یافته را در خروجی خواهد داشت. در این مورد برای صرفه جویی در اندازه مدل و زمان پردازش با توجه به اهمیت آن در عمل، دو نوع شبکه ساده به کار گرفته شد که نوع اول متشکل از تنها یک لایه و نوع دوم تنها شامل دولایه از ضرب پیچشی خواهند بود. گرچه مرحله تعلیم می‌تواند با داده‌های بسیار مختلفی تست گردد و با اعمال فیلترهای دستی، امکان آگیومننتیشن فراوانی وجود دارد اما این مورد به پژوهش‌های آینده منتقل شد. در اینجا در مرحله تعلیم در روش ساده‌تر هر بار اجرا ۲۰ ثانیه زمان می‌برد که ۶۰ دور اجرا می‌گردد. در مدل پیچیده‌تر ۶۰ ثانیه زمان اجرا در سخت افزار مورد استفاده ما بود که با توجه به تمایل به تست در زمان مساوی به ۲۰ دور بسنده شد. البته باید توجه شود داده مورد تست بسیار محدود بود و تعلیم بیشتر در گرو وجود داده‌های بیشتر بود. در هر حال اگر چنین سامانه‌ای بتواند حالات مختلفی از مات شدگی را ببیند می‌توان امید داشت در زمان تست خروجی مطلوبی مشاهده گردد. بررسی خروجی‌های این روش‌ها نیز در قسمت ارزیابی همین بخش آورده شده است. شکل ۳۰ به ترتیب مدل‌های دولایه ضرب پیچشی، مدل تک لایه ضرب پیچشی و سایر اطلاعات الگوریتم را نمایش می‌دهد.

```
filter_size = 3

model2 = Sequential([
    Conv2D(4, filter_size, input_shape=(800, 1360, 1), padding="SAME"),
    Conv2D(1, filter_size, input_shape=(800, 1360, 1), padding="SAME"),
])
```

(الف)

```
model = Sequential([
    Conv2D(1, filter_size, input_shape=(800, 1360, 1), padding="SAME"),
])
```

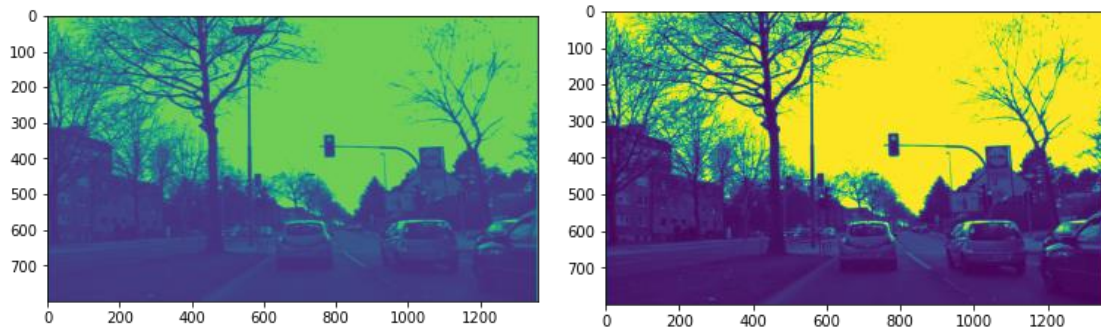
(ب)

```
model.compile(
    optimizer="adam", loss="mean_squared_error"
)
```

(ج)

شکل ۳۰ - الف) مدل دولایه ضرب پیچشی ب) مدل تک لایه ضرب پیچشی ج) سایر اطلاعات الگوریتم

شکل ۳۱ نیز یک نمونه از تصویر اصلی و خروجی سامانه را نمایش می‌دهد.



شکل ۳۱ - به ترتیب از راست به چپ: نمونه تصویر اولیه و خروجی

## ارزیابی

تاکنون تلاش شد تا روشی ارائه گردد تا یک راهبرد عمومی برای تصاویر وجود داشته باشد. در ادامه شرح مواردی است که به عنوان معیار ارزیابی می‌توانند استفاده گردند.

در بررسی میزان ماتی تصویر می‌توان یک قاعده کلی را عنوان کرد که با مات شدن تصویر واریانس پیکسل‌های تصویر کاهش می‌یابد که چنین چیزی به ویژه با تجسم فیلتر میانگین‌گیر بدیهی است.

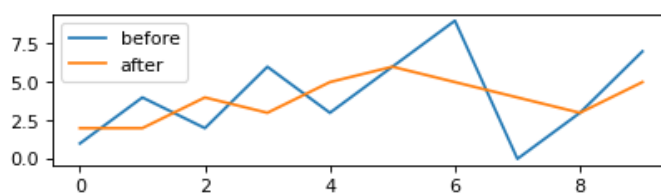
در یک نمونه ساده تک بعدی در نظر بگیرید که کرنل  $[1/3, 1/3, 1/3]$  را داشته باشیم و آرایه شامل ۱۰ عدد تصادفی و تک رقمی زیر باشد:

$[1, 4, 2, 6, 3, 6, 9, 0, 3, 7]$

خروجی کرنل برابر خواهد شد با:

$[2, 2, 4, 3, 5, 6, 5, 4, 3, 5]$

نمودار رسم شده در شکل ۳۲ نیز همین نکته را نشان می‌دهد. در هر حال واریانس حاصل شده به ترتیب ۷.۳ و ۱.۷ می‌شود.

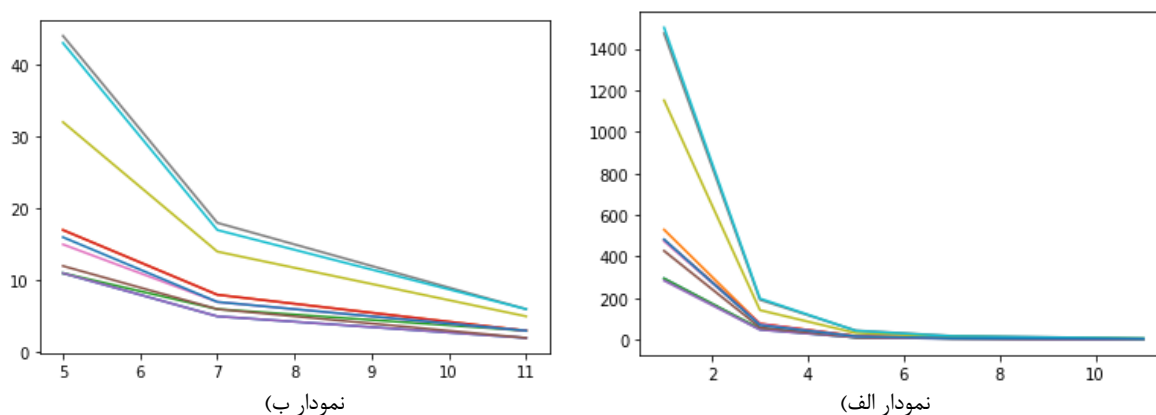


شکل ۳۲ - تغییرات واریانس

پس این حدس ایجاد می‌شود که نمونه داده‌ها نیز بر اساس میزان مات شدگی تحت تاثیر کرنل‌های مختلفی قرار گرفته اند و در نتیجه ایده ساده این است که با تست نمونه‌های زیادی از خروجی‌ها کرنل‌های مختلف به نوعی برای هر کرنل یک ترشولد میانگین بدست آوریم.

این روش ساده است اما با توجه به اینکه تصویر چه شرایطی دارد خیلی متغیر هست، به عبارتی بیش از تحت تاثیر تغییر رنگ‌ها در نواحی مختلف است. پس بهتر است به نحوی همسایگی‌ها اعمال گردد. در این راستا دو ایده به ذهن می‌رسد. یکی اینکه از مشتق آن تصویر استفاده نماییم. برای این هدف از واریانس لاپلاسین تصویر استفاده می‌شود. این روش گرچه ساده است و منطقی به نظر می‌رسد اما یک چالش مهم دارد و آن هم تنوع داده‌ها به خاطر وجود تفاوت در میزان تغییرات در داده اصلی است. به عبارتی برای یک تصویر می‌توان انتظار داشت که با بزرگ شدن کرنل مات کننده و افزایش ماتی تصویر، میزان واریانس کاهش یابد. اما این معیار به عنوان ترشولد نمی‌تواند برای چند تصویر مختلف هم زمان کار کند. برای نمونه نمودار الف در شکل ۳۳ از چند تصویر مختلف تهیه شده است که اندازه کرنل‌های ۱، ۳، ۵، ۷، ۹، ۱۱ مورد بررسی قرار گرفته که در محور افقی می‌باشد و میزان واریانس در محور عمودی مشخص است. در اینجا به عنوان نمونه، نمونه‌ی متناظر با خط آبی رنگ در کرنل اندازه

۵ مقدار بیشتری دارد نسبت به نمونه متناظر با خط قهوه‌ای رنگ در کرنل سایز ۳. تصویر دوم با زوم بیشتر و گویاتر می‌باشد.

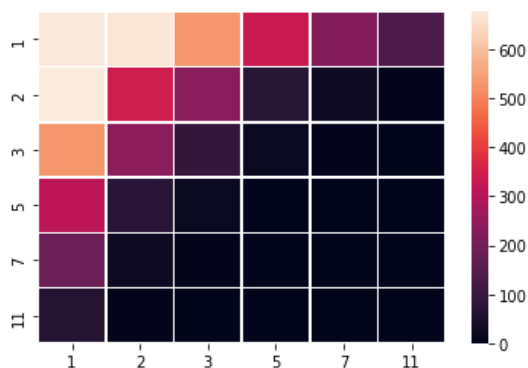


شکل ۳۳ - تغییرات واریانس با تغییر اندازه‌ی کرنل

شکل ۳۴ نحوه محاسبه این مورد را نشان می‌دهد.

```
cv2.Laplacian(img, cv2.CV_64F).var()
```

شکل ۳۴ - نحوه محاسبه‌ی لاپلاسیان



شکل ۳۵ - میزان ماتی نسبت به کرنل‌های مختلف

توجه شود کرنل‌های مختلفی می‌توانند باعث مات شدن تصویر شوند. ابعاد کرنل، تمایل به افقی یا عمودی بودن کرنل و خود نوع کرنل مثلاً توزیع یکنواخت یا توزیع گوسی می‌توانند عوامل این تفاوت‌ها باشند که هر یک به نحوی منجر به خروجی متفاوتی می‌گردد. پیشتر یک نمونه ساده از این کرنل‌ها به صورت افقی و عمودی آن قیاس شد. برای نمونه تاثیر اندازه کرنل بر روی واریانس نمودار شکل ۳۵ محاسبه شده است. توجه شود کاهش واریانس با میزان ماتی رابطه معکوس دارد.

از جمله معیارهای دیگری که می‌توان عنوان کرد بررسی میزان تفاوت بین عکس سالم و عکس بازیابی شده است که هرچه این تفاوت کمتر باشد انتظار خروجی بهتری را می‌توان داشت. اما این مورد در زمانی که طبق روش پیشنهادی تلاش شود مرزها بیشتر تقویت شوند عملاً نتایج ضعیف‌تری را نشان خواهد داد و این مورد می‌تواند منجر به نتایج غیر کاربردی تر گردد. از طرفی پس از اعمال فیلتر مات کننده مانند کرنل پایین‌گذر، بدون کاهش

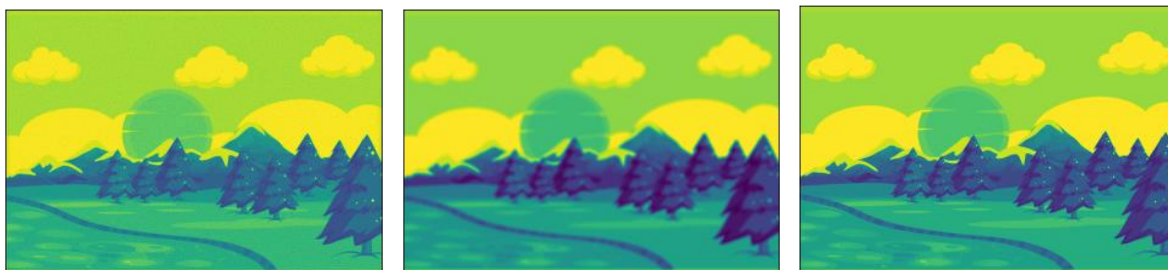
ماتی می‌تواند در مواردی خروجی بهتری داشته باشد تا روش شارپ شده و با نویز کاهش یافته اما مرزهای پر رنگ شده نمی‌تواند در برابر تصویر مات نویزی ورودی رقابت کند. در هر حال این مورد در روش‌هایی که تلاش می‌شود تا کرنل اکتشاف شود و مسئله به صورت چشم باز باشد مورد استفاده قرار می‌گیرد. این چنین مواردی در روش‌های مبتنی بر یادگیری عمیق نمونه‌های فراوانی دارند. برای نمونه مقاله‌ی پیدونگ و همکاران [13].

شکل ۳۶ نحوه‌ی محاسبه تفاوت را نمایش می‌دهد.

```
def differenceComputation(im1,im2):
    return round(np.abs(im1-im2).sum()/(im1.shape[0]*im1.shape[1]),3)
```

شکل ۳۶ - نحوه‌ی محاسبه‌ی تفاوت

یک نمونه برای اعمال وینر در شکل ۳۷ آورده شده است. به ترتیب از راست به چپ تصویر سالم، تصویر مات شده دارای نویز و تصویر بازسازی شده به کمک فیلتر وینر قابل مشاهده است:



ج) تصویر بازسازی شده با فیلتر وینر

ب) تصویر مات

الف) تصویر اولیه

شکل ۳۷ - بازسازی تصویر مات با فیلتر وینر

با توجه به اینکه گفته شد مرزها و تخریب شدگی آن‌ها اهمیت زیادی دارند، یک معیار دیگر معرفی می‌نماییم. در این مورد ابتدا مرزهای اساسی تصویر استخراج می‌گردد و سپس این مرزها مقایسه خواهند شد و درصد مطابقت محاسبه می‌گردد. برای یافتن مرزها از اپراتور canny استفاده می‌کنیم. همچنین برای مقایسه شباهت با توجه به تغییر خروجی به اعداد ۰ و ۱ پس نیاز به تابع شباهت جدید است که در شکل ۳۸ آورده شده است.

```
def difrencecomputation2(im1,im2):
    return round((im1==im2).sum()/(im1.shape[0]*im1.shape[1]),3)
```

شکل ۳۸ - نحوه‌ی مقایسه‌ی شباهت

شکل ۳۹، نمونه‌هایی از خروجی مرزها را در این سه حالت نمایش می‌دهد.



الف) تصویر اولیه

ب) تصویر مات

ج) تصویر بازسازی شده با فیلتر وینر

شکل ۳۹ - نمونه‌هایی از مرزهای به دست آمده در تصویر اصلی، مات و بازسازی شده

البته اگر پارامتر سیگما مقدار دهی نمی‌شد مرزهای بیشتری کشف می‌شد اما ترجیح دادیم مرزهای اصلی را جدا نماییم. در نهایت با تجمیع مرزها با تصویر خروجی وینر، به خروجی به مراتب بهتری دست یافته‌ایم که میزان شباهت آن بیشتر شده است.

در ادامه بر اساس تست تعداد بیشتری نمونه تلاش شد یک میانگین خروجی از موارد فوق و بر اساس پارامترهای مطرح شده ارائه گردد. لازم به ذکر است در این مورد اطلاعات گزارش شده براساس بهترین پارامترها است که بخشی از آن به صورت تجربی بدست آمده است.

معیار برتری		بیشتر بودن مقادیر		بیشترین شباهت با تصویر اولیه			امتیاز	
معیار	شباهت مرزها با تصویر اولیه	شباهت خود تصویرها با تصویر اولیه	واریانس تصویر (میزان شباهت)	در صد معادل	واریانس لاپلاسین تصویر (میزان شباهت)	نسبت	امتیاز کلی جمع	امتیاز کلی ضرب (نرمالایز شده)
تصویر اولیه			۰.۱۳۱		۰.۰۱۴			
مات بدون وینر	%۹۶.۷	%۱.۴۵	۰.۱۳	%۹۹	۰.۰۰۲	۷/۱ برابر	۴۹.۵	۰.۵۵
مات با وینر	%۹۶.۴۵	%۱۷.۰۲	۰.۰۳۲	%۲۴	۰.۰۰۵	۳/۱ برابر	۴۷	۴۶.۹
فیلتر شارپ کننده	%۹۵.۹	%۲۵.۱۷۳	۰.۰۰۸	%۶	۰.۰۰۳	۵/۱ برابر	۴۳	۲.۲۲
ادغام وینر و کنی	%۹۵.۴	%۱.۳۶۸	۰.۱۳۷	%۹۵	۰.۰۷۲	۵ برابر	۵۵	۴۷
ادغام وینر با کنی با ضریب ۰.۵	%۹۵.۷۳۴	%۱.۰۷۵	۰.۱۳۴	%۹۷.۷	۰.۰۴۵	۳ برابر	۶۱	۱۰۷

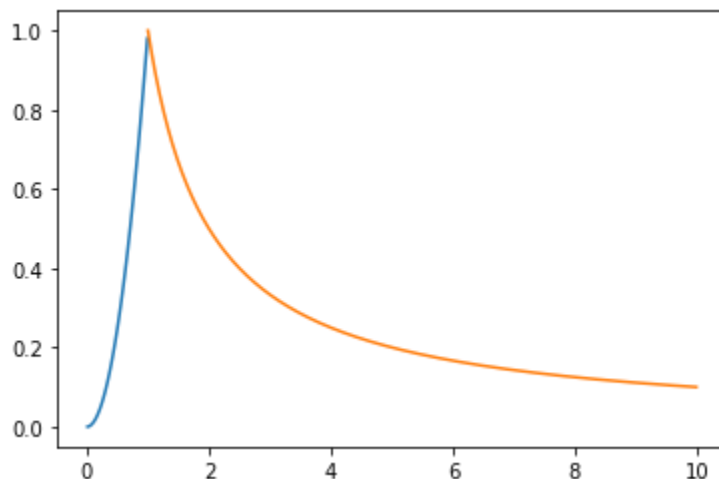


روش مکرر	۴.۶۹۵	۰.۱۴۶	۰.۰۰۵	%۴	۰.۰۰۱	۱۴/۱ برابر	۲.۶	۳
روش مکرر + لبه در هر مرحله	۴.۹۱	۰.۱۴	۰.۰۱	%۸	۰.۰۰۲	۷/۱ برابر	۳.۷	۰
روش مکرر + وینر به عنوان اولیه	۴.۱	۰.۱۴	۰.۰۰۵	%۴	۰.۰۰۱	۱۴/۱ برابر	۲.۳	۲.۵
روش مکرر + افزودن مرز به تصویر اولیه با ضریب کم	۴	۰.۱۰۸	۰.۰۰۵	%۴	۰.۰۰۱	۱۴/۱ برابر	۲.۳	۲.۵
عمیق تک لایه ضرب پیچشی	۹۴.۷۸۲	۲.۲۹۹	۰.۱۱	%۸۴	۰.۰۱۴	۱	۸۶	۱۷۰۰
عمیق دو لایه ضرب پیچشی	۹۱.۷۶۴	۲.۴۵	۰.۱۲۱	%۹۳	۰.۰۱۱	۱۴/۱۱ برابر	۷۰	۷۳۰

با وزن دهی به شباهت‌ها براساس اهمیتی که مطرح شد و توجه به این نکته که درصدهای مطرح شده در ستون آخر، می‌دانیم در صورت بیشتر بودن، عملکرد بهتری باید در نظر گرفته شود پس به طور کلی با استفاده از رابطه‌ی زیر داریم:

فرمول: میانگین وزن دار همه ستون‌ها با در نظر گرفتن ضرایب به ترتیب از راست به چپ ۰.۴ و ۰.۱ و ۰.۱ و ۰.۴ باشند. در مورد ستون آخر با این فرمول حساب گردیده است، نمودار آن نیز قابل مشاهده است:

$$\begin{cases} \text{if } n \geq 1 : \text{output} = \frac{1}{n} \\ \text{else} : \text{output} = n^2 \end{cases}$$



همچنین در ستون سمت چپ همین اعداد میانگین حساسی شده اند و مواردی که ضریب ۰.۴ دارند تبدیل به توان ۲ شده‌اند. با توجه به همه موارد فوق بهترین انتخاب ادغام وینر با کنی با ضریب ۱/۲ است.

## کارهای آینده

با توجه به کار انجام شده و مطالعاتی که انجام شد گزینه‌های مختلفی مطرح می‌گردد که می‌توانند در آینده در بهبود این سامانه اثر گذار باشند. در صورتی که داده‌هایی با ویژگی دنباله زمانی در دسترس باشد می‌توان از دنبال کردن اشیا در این سامانه استفاده کرد که کار تشخیص و دسته‌بندی تابلوها را آسان می‌کند. از طرفی گرچه تلاش شد با توجه به عدله ارائه شده از یادگیری عمیق تا حد امکان چشم پوشی شود اما در صورت وجود داده‌های متنوع و کافی برای تعلیم چنین سامانه‌ای به صورت مبتنی بر یادگیری عمیق گزینه مطلوبی است که می‌تواند مورد بررسی قرار گیرد. در این مورد امکان سنجی شامل تخمینی از فرصت و مدت اجرای پیشبینی شبکه و میزان حافظه در دسترس اهمیت خواهد داشت. یک گزینه دیگر در رابطه با مشکلات مات شدگی است که می‌توان یک سامانه فازی تعریف نمود که دامنه کرنل مات کننده و شدت نویز را در شرایط مختلف داشته باشد و پیشنهاد کرنل ضد مات کنندگی بر طبق آن باشد. خصوصی سازی سامانه برای تابلوهای خاص یک کشور یا منطقه نیز می‌تواند مورد توجه قرار بگیرد گرچه باید توجه شود در این صورت سامانه گرچه عملکرد بهتری خواهد داشت اما محدودیت مکانی استفاده از سامانه تشدید می‌شود.

## جمع‌بندی

در این پژوهش تلاش شد تا سامانه‌ای برای تشخیص تابلوهای ترافیک و دسته بندی آن ارائه گردد. این سامانه به طور جامع با انواع مختلفی از داده‌های ورودی رو به رو شده و با تشخیص تابلوها و سپس دسته‌بندی آن به انواع مختلفی از تابلوهای ممکن کار میکند. در این راستا پژوهش‌های مختلفی بررسی شد و تلاش شد با ایده گرفتن از آن‌ها و ایده‌های تیم، سامانه در جهت بهبود عملکرد توسعه یابد. در این مسئله، چالش‌های مختلفی وجود دارد که در هر بخش مورد بررسی قرار گرفته‌اند. از جمله این چالش‌ها نویز و مات شدگی تصاویر ورودی است که تلاش شد جداگانه مورد بررسی قرار بگیرند. از طرفی این مسئله که سامانه باید حداقل پیچیدگی فضایی و زمانی را داشته باشد مد نظر بوده و همین مورد باعث شد تا از سامانه‌های عمیق پیچیده چشم پوشی گردد.

خط لوله این سامانه شامل سه فاز متوالی است که عبارت اند از کاهش مات شدگی، تشخیص تابلوها و دسته‌بندی تابلوها. در این زمینه ایده‌های مختلفی مورد بررسی قرار گرفت که شامل نوآوری‌های مختلفی نیز بوده است. از

جمله این نوآوری‌ها می‌توان به ترکیب فیلتر وینر و یال‌های قوی استخراج شده، ارائه‌ی سامانه‌ی تکرار شونده با ورودی بهبود یافته و طراحی معیارهای جدید کاهش میزان مات شدگی جهت بهبود مات شدگی اشاره کرد. همچنین جهت بهبود دسته‌بندی تابلوها می‌توان از ویژگی‌های فرکانسی تصاویر و ماسک‌های فرکانسی حلقوی استفاده کرد.

## منابع

- [1] Vashisth, Sharda, and Sumeet Saurav. "**Histogram of oriented gradients based reduced feature for traffic sign recognition.**" *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2018.
- [2] Wali, Safat B., et al. "**Vision-based traffic sign detection and recognition systems: Current trends and challenges.**" *Sensors* 19.9 (2019): 2093.
- [3] Cao, Jingwei, et al. "**Improved traffic sign detection and recognition algorithm for intelligent vehicles.**" *Sensors* 19.18 (2019): 4021.
- [4] Sugiharto, Aris, and Agus Harjoko. "**Traffic sign detection based on HOG and PHOG using binary SVM and k-NN.**" *2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*. IEEE, 2016.
- [5] Tang, Jialin, et al. "**Traffic sign recognition based on HOG feature and SVM.**" *Proceedings of the 2020 4th International Conference on Electronic Information Technology and Computer Engineering*. 2020.
- [6] Wang, Gangyi, Guanghui Ren, and Taifan Quan. "**A traffic sign detection method with high accuracy and efficiency.**" *Conference of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*. Atlantis Press, 2013.
- [7] Soetedjo, Aryunto, and I. Komang Somawirata. "**An efficient algorithm for implementing traffic sign detection on low cost embedded system.**" *AN EFFICIENT ALGORITHM FOR IMPLEMENTING TRAFFIC SIGN DETECTION ON LOW COST EMBEDDED SYSTEM* 14.1 (2018): 1-14.
- [8] Zeng, Yiliang, et al. "**Restoration of motion-blurred image based on border deformation detection: A traffic sign restoration model.**" *PLoS one* 10.4 (2015): e0120885.
- [9] Horak, Karel, Pavel Cip, and Daniel Davidek. "**Automatic traffic sign detection and recognition using colour segmentation and shape identification.**" *MATEC Web of Conferences*. Vol. 68. EDP Sciences, 2016.

- [10] Yngve Moe, "**Deblurring an image,**" Website URL: "<https://stackoverflow.com/questions/53743561/deblurring-an-image>", last checked: 2022, Aug,18, last edition: 2022, Jun, 23.
- [11] Bojarczak, Piotr, and Zbigniew Lukasik. "**Image deblurring–Wiener filter versus TSVD approach.**" *Advances in Electrical and Electronic Engineering* 6.2 (2011): 86-89.
- [12] Biswas, Prodip, Abu Sufian Sarkar, and Mohammed Mynuddin. "Deblurring images using a Wiener filter." *International Journal of Computer Applications* 109.7 (2015): 36-38.
- [13] Liu, Peidong, et al. "**Self-supervised linear motion deblurring.**" *IEEE Robotics and Automation Letters* 5.2 (2020): 2475-2482.

## سرواژه‌ها

- Traffic sign recognition (TSR)
- Traffic sign detection and recognition (TSDR)
- Regions of interest (ROIs)
- Singular value decomposition (SVD)
- Truncated singular value decomposition (TSVD)

## معادل‌ها

- تشخیص: Detection
- شناسایی: Recognition
- دسته بندی: Classification
- هیستوگرام شیب های جهت دار: Histogram of Oriented Gradients
- ماشین بردار پشتیبانی: Support-vector machine
- نواحی مورد علاقه: Regions of interest
- تشخیص علامت‌های راهنمایی رانندگی: Traffic sign recognition
- مات کردن: Blurring
- رفع مات شدگی: Deblurring
- مات: Blur
- چوش پوشیده: Blind
- چشم باز: non Blind