

مقایسه شبکه های عصبی YOLOv3 ، YOLOv5s و MobileNet-SSD V2 برای تشخیص ماسک صورت در زمان واقعی

مریم فردی، کوروش داداش تبار احمدی*
دانشکده مهندسی کامپیوتر هوش مصنوعی، دانشگاه صنعتی مالک اشتر، تهران، ایران.

چکیده

امروزه، یادگیری عمیق با استفاده از موارد متعدد پتانسیل واقعی خود را نشان می دهد و در حوزه های مختلف تکنولوژیکی اهمیت دارد. یکی از پرطرفدارترین کاربردهای یادگیری عمیق، تشخیص و ردیابی اشیاء است. محصولات اخیر نتایج امیدوارکننده ای را در همین رابطه نشان داده است. این مقاله رویکردهای سیستماتیک مختلفی را مورد بحث و مقایسه قرار می دهد که تصاویر را تجزیه و تحلیل نموده و مشخص می کند که آیا فرد ماسک صورت را به درستی، نادرست و یا اصلاً استفاده نموده است یا خیر. تشخیص ماسک بر روی تصاویر، ویدئوها و سیستمهای نظارتی در زمان واقعی، با استفاده از سه الگوریتم یادگیری ماشینی پرکاربرد انجام می شود Yolon3، Yolon5 و MobileNet-SSD V2 هرمدل وجود ماسک را روی صورت افراد تشخیص می دهد که بر اساس دقت آن ها و میزان سرعت پردازش ویدیو قضاوت می شود. نتایج عملکرد سه الگوریتم برای تشخیص وجود ماسک صورت بر روی یک فرد در زمان واقعی بر حسب FPS تعیین میگردد.

کلمات کلیدی: Yolon3، Yolon5، MobileNet-SSD، mAP، FPS.

تاریخچه مقاله:

تاریخ ارسال: ۱۴۰۰/۱۲/۲۸

تاریخ اصلاحات: ۱۴۰۱/۰۱/۱۵

تاریخ پذیرش: ۱۴۰۱/۰۴/۱۷

تاریخ انتشار: ۱۴۰۱/۰۵/۱۱

Keywords:

Yolon3, Yolon5,
MobileNet-SSD,
mAP, FPS

*ایمیل نویسنده مسئول:

Faranak.emtehani@yahoo.com

Compare neural networks YOLOv3 ، YOLOv5s، MobileNet-SSD V2 to detect face masks in real time

Kurosh Dadashtabar Ahmadi*, Maryam fardi

Department of Computer Science, Malek-Ashtar University of Technology, Tehran, Iran.

Abstract

Surveys, evaluations using multiple cases show their true potential and there are important technologies in various fields. One of the most popular applications of concepts is object recognition and tracking. Recent products show promising results in this regard. These systems can be compared and compared, which analyzes the images and determines whether the person has used the mask correctly, incorrectly or not at all. Mask detection is performed on real-time video and surveillance systems using three widely used machine image algorithms: Yolon3, Yolon5 and MobileNet-SSD V1. Recognized. Will be judged. The performance results of the three algorithms for detecting the presence of a face mask on a person in real time are determined in terms of FPS..

Keywords: Yolon3, Yolon5, MobileNet-SSD, mAP, FPS.

۱- مقدمه

در ماه های پایانی سال ۲۰۱۹، ویروس کشنده جدیدی در ووهان چین کشف شد که جهان را درگیر نمود و زندگی و اقتصاد بی شماری را ویران کرد. دولت ها در سراسر جهان شروع به کار بر روی استراتژی های مختلفی مانند فاصله گذاری اجتماعی، استفاده از ضد عفونی کننده ها و کیت های PPE برای جلوگیری از انتشار ویروس نمودند. بیماری کرونا ویروس ۲۰۱۹ (کووید-۱۹) به دلیل اثرات مضر آن مانند مرگ و میر، عفونت های حاد تنفسی و بحران مالی یک موضوع جدی بهداشت عمومی و اقتصادی است. ویروس به سرعت در محیط های شلوغ و در تماس نزدیک با میزان بالا پخش می شود. پوشیدن ماسک اولین خط دفاع شخصی در برابر این ویروس اعلام شد و توسط همه دولت ها در سراسر جهان اجباری شده است. با این حال، درصد قابل توجهی از مردم در جهان نیز وجود دارد که از رعایت آنها سر باز می زنند. برای اجرای زدن ماسک صورت، مقامات برای افرادی که در اماکن عمومی از ماسک استفاده نمی کنند شروع به اعمال جریمه نمودند. هدف اصلی این مقاله تشخیص و ردیابی وجود یک صورت ماسکدار در کنار سایر شرایط است، مانند اینکه آیا ماسک به درستی استفاده شده یا اصلاً استفاده نشده است. با توجه به ورودی، خواه یک تصویر یا یک ویدیوی در زمان واقعی، از چهره نقابدار در خروجی بر اساس MobileNet-v5s، YOLOv3 و SSD V2 نشان داده شده است. تشخیص اشیاء به دلیل طیف گسترده ای از موارد استفاده ممکن، جنبه عمیق تری از بینایی کامپیوتر را بدست آورده است. بیشتر تکنیک های تشخیص ماسک صورت، مانند الگوریتم های یادگیری ماشین سنتی بر روی تشخیص و ساخت چهره تمرکز دارند. تمرکز این مقاله صرفاً شناسایی و ردیابی افرادی است که ماسک های صورت را نادرست استفاده مینمایند و یا نمی پوشند و نتایج را بر اساس دقت آن ها و میزان سرعت آرام پردازش ویدیو (FPS) با پارامتر تصمیم گیری بیشتر برای تشخیص زمان واقعی هم مقایسه می کند.

۲- الگوریتم های موجود

الگوریتم های موجود تشخیص شی از دو بخش تشکیل شده است: محلی سازی و طبقه بندی^۱. شروع تشخیص با استخراج ویژگی های انتخابی Haar, HOG, Convolutional (layer) صورت میگردد و سپس از محلی سازی یا طبقه بندی

کننده برای طبقه بندی شی استفاده می شود. به طور کلی، این بومی سازی و طبقه بندی کننده بر روی یک تصویر بر اساس رویکرد^۲ پیشنهاد منطقه یا تکنیک پنجره کشویی^۳ روی تصویر اجرا می شود.

روش هایی مانند، مدل های قطعات قابل تغییر شکل^۴ (DPM) الگوی رویکرد پنجره کشویی هستند و روش هایی مانند-R CNN از رویکرد پیشنهادی منطقه^۵ برای تولید جعبه های مرزی استفاده می کنند و پس از آن، یک طبقه بندی را بر روی جعبه های مرزی^۶ توضیح داده شده اجرا می کنند، تا پس از پردازش، جعبه های تکراری فیلتر شوند. ماهیت مسیرمورد استفاده در این روش ها بسیار پیچیده و دشوار است، زیرا در چنین سیستم هایی هر جزء به طور جداگانه آموزش داده می شود. اما سیستم هایی مانند Yolo تشخیص شی را به عنوان یک رگرسیون تک مرحله ای، در یک شبکه واحد انجام میدهد. بنابراین، در چنین سیستم هایی مانند Yolo، الگوریتم محاسباتی را بر روی یک تصویر انجام می دهد تا مکان آنها را پیش بینی نموده و آن اشیاء را طبقه بندی نماید.

همچنین MobileNetV2 با مدل های latency و توان کم دقت خوبی از خود نشان داده است. در این مقاله سیستم های YOLOv3، YOLOv5s و MobileNet-SSD V2 مقایسه شده اند تا بهترین الگوریتم مناسب برای سیستم انتشار ماسک شناسایی شود.

۱-۲-YOLOv3: در این روش بومی سازی و طبقه بندی را تنها با کمک یک شبکه عصبی انجام می دهد، از این رو این شبکه را به یکی از بهترین ها برای سیستم های تشخیص بلا درنگ تبدیل می نماید، بنابراین YOLOv3 در مقایسه با سایر الگوریتم ها سریع تر می باشند. در مقایسه با سایر الگوریتم ها این ویژگی به YOLOv3 اجازه می دهد تا، ورودی های در زمان واقعی آموزش ببیند و تشخیص ها را تقریباً دقیق انجام دهد. سرعت YOLOv3 از ۴۵ فریم بر ثانیه (fps) تا ۱۵۵ فریم در ثانیه می رسد [۷].

1 localization and classification
2 region proposal
3 in-sliding window
4 Deformable Parts Models
5 region proposal
6 bounding boxes

لایه استفاده کرده است، برای شبکه های عصبی عمیق، افزایش تعداد لایه ها به معنای افزایش دقت است. با این حال، تصویر ورودی هنگام ارسال به لایه های عمیق تر، نمونه برداری^۸ می شود که منجر به از دست دادن ویژگی های ریز دانه^۹ می شود. به همین دلیل است که YOLOv2 اغلب با تشخیص اشیاء کوچک دچار چالش بوده است [۴].

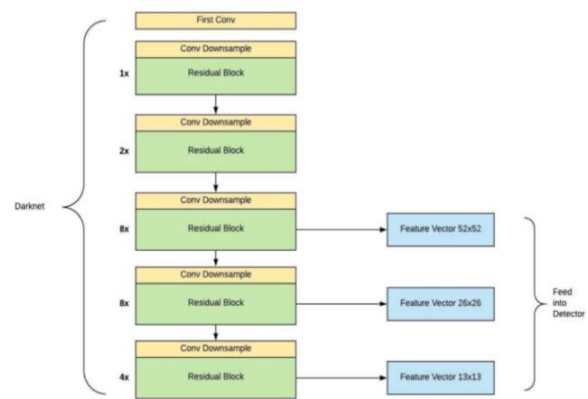
۲-۲- YOLOv5: برای مدت طولانی بر حوزه خود تسلط داشته، و در می ۲۰۲۰ پیشرفت بزرگی حاصل شد. دو نسخه به روز و بهتر از YOLO یکی پس از دیگری معرفی شدند. یکی YOLOv4 بود که توسط نویسندگان معمولی جوزف ردمن و الکسی بوچکوسکی توسعه یافت [۴]، دیگری YOLOv5 که توسط گلن جوچر منتشر گردید. نسخه جدید که نویسنده معمولی سری YOLO نیست، با بحث هایی همراه بود، اما با گذشتن از آن، مدل YOLOv5 عملکرد قابل توجهی نسبت به نسخه های قبلی خود نشان داده است. با این حال، YOLOv5 دارای مزایای زیادی در مهندسی بوده است. استفاده از زبان پایتون به جای C را میتوان نام برد که این امر نصب و ادغام با دستگاه های IoT را بسیار آسان تر می نماید. علاوه بر این، جامعه PyTorch نیز از جامعه Darknet بزرگتر است، به این معنی که PyTorch پتانسیل رشد زیادی در آینده خواهد داشت و مشارکت های بیشتری دریافت خواهد نمود و همراه با توسعه YOLO در سال ۲۰۱۶، بسیاری از الگوریتم های تشخیص اشیاء با رویکردهای مختلف به دستاوردهای قابل توجهی نیز دست یافته اند. پیشرفت ها دو مفهوم از معماری را فرموله کرده است تشخیص شیء: آشکارساز یک مرحله ای و آشکارساز دو مرحله ای. شبکه YOLOv5 از سه بخش اصلی تشکیل شده است.

ستون فقرات (backbone): یک لایه CNN دارای ویژگی های تصویر در مقیاس های مختلف است.

گردن (Neck): مجموعه ای از لایه ها برای ترکیب ویژگی های تصویر و انتقال آنها به پیش بینی (prediction).

سر (Head): ویژگی ها را از گردن می گیرد، محلی سازی و طبقه بندی را انجام می دهد.

این روش، از یک رویکرد ترکیبی از YOLOv2 و Darknet-19 استفاده مینماید، که در آن تصویر یک بار در طول تشخیص ارسال شده و خروجی در مرحله بعد تولید می گردد، که باعث می شود YOLOv3 سریع تر از R-CNN باشد. از آنجایی که YOLOv3 بر روی تصاویر کامل بدون استخراج نکات سوخت^۷ و با استفاده از روش آموزشی چند مقیاسی آموزش داده شده است، تصویر ورودی در ابتدا به شبکه هایی تقسیم می شود و شبکه، جعبه های مرزی را به همراه عرض و ارتفاع آنها پیش بینی می کند. گاهی اوقات ممکن است شبکه ها حاوی هیچ شیئی نباشند که منجر به نمره شیئی صفر می باشد. موقعیت هایی مانند این ممکن است بی ثباتی ایجاد کند و به مدل خطا اضافه نماید، اما می توان با افزایش تعداد مختصات جعبه های مرزی و در نتیجه کاهش فضای خالی در یک تصویر، با آن مقابله نمود [۷].



(شکل-۱): برای آشکارساز چند مقیاسی به شبکه کناری اضافه شد تا ۳ بار در ۳ مقیاس مختلف را تشخیص دهد

این امکان، کارایی شناسایی شیء در تصویر را بهبود می بخشد. در یک سلول شبکه ای خاص، YOLOv3 قادر به پیش بینی و تجزیه و تحلیل جعبه های محدود کننده متعدد است. یکی از ویژگی های اصلی YOLOv3 این است که اشیاء را در سه مقیاس همانطور که در شکل ۱ نشان داده شده است تشخیص می دهد. به طور خاص، YOLOv3 پیش بینی هایی را در لایه های ۸۲، ۹۴ و ۱۰۶ انجام می دهد که دقیقاً با گام شبکه مدل، ارائه می شوند که به ترتیب ۸۰۳۲، ۱۶ و ۳۰ هستند [۴]. YOLOv2 از ۳۰ لایه کانولوشنال استفاده مینماید که معماری آن Darknet می باشد، همچنین تعداد لایه های آن بیشتر از YOLOv1 می باشد که از ۱۱

⁷ hard negative

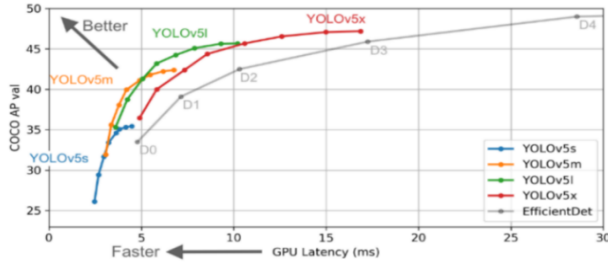
⁸ downsampled

⁹ fine-grained

۳- yolov5-l نسخه بزرگ

۴- yolov5-x نسخه فوق العاده بزرگ

تجزیه و تحلیل عملکرد همه این مدل ها طبق Glenn Jocher در شکل ۳ در زیر ارائه شده است.

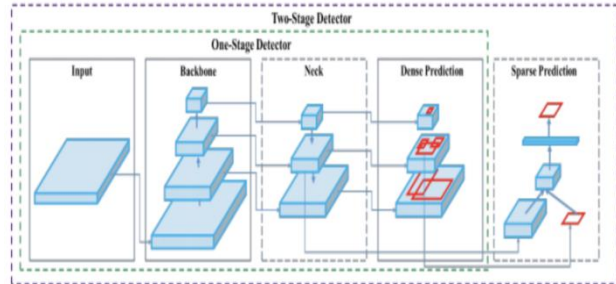


(شکل-۳): مقایسه ای از خانواده YOLOv5

۲-۲- MobileNet-SSD V2: روند کلی مشاهده شده بدین صورت میباشد که مدل های بینایی کامپیوتری عمیق تر و پیچیده تر می شوند تا به دقت بیشتری دست یابند. با این حال، این پیشرفت ها در حال افزایش اندازه و تاخیر^{۱۴} هستند و نمی توانند در سیستم های محاسباتی ناتوان^{۱۵} استفاده شوند. در چنین مواردی، MobileNet مفید است. این مدلی است که به طور خاص برای موبایل و برنامه های جاسازی شده^{۱۶} که به سرعت بالا نیاز دارند طراحی شده است.

نسخه اول آن (MobileNetV1) دارای یک پیچیدگی عمیق قابل تفکیک بود (depthwise separable convolution) که اندازه مدل و هزینه پیچیدگی شبکه را به سطح مناسبی کاهش داد تا آن را برای برنامه های کاربردی کم پردازش قابل استفاده نماید. پس از آن در ویرایش دوم خانواده MobileNet، یک ساختار ماژولار بودن^{۱۷} بسیار بهتر ارائه شده است و این نسخه MobileNetV2 نامگذاری گردید. این امر به حذف غیرخطی ها^{۱۸} در لایه های باریک کمک کرده است که منجر به عملکرد پیشرفته cutting-edge برای کاربردهای فوق که در بالا اشاره گردید می شود [۵].

Detection Neck (Neck) به عنوان یک جمع کننده ویژگی کار می کند و وظیفه دارد ویژگی های تشکیل شده در Backbone را ترکیب کند تا برای مرحله بعد در Detection Head یا (Head) آماده شود [۳].



(شکل-۲): دو مفهوم تشخیص اشیاء معماری [۶]

نقطه مشترک همه معماری های تشخیص اشیاء این است که ویژگی های تصویر ورودی از طریق یک استخراج کننده ویژگی (Backbone) فشرده می شود و سپس به آشکارساز شی^{۱۰} ارسال می شود. (شکل ۲) تفاوت در اینجا این است که Head مسئول تشخیص همراه با طبقه بندی و محلی سازی برای هر جعبه مرزی است. آشکارساز دو مرحله ای این ۲ کار را جداگانه اجرا می کند و نتایج آنها را بعداً ترکیب مینماید. (تشخیص پراکنده Sparse Detection)، در حالی که آشکارساز یک مرحله ای آن را به طور همزمان اجرا می کند (تشخیص متراکم Dense Detection) در شکل ۲ نشان داده شده است.

Yolov یک آشکارساز تک مرحله ای است، بنابراین، شما فقط یک بار نگاه می کنید. در مورد آشکارساز یک مرحله ای، عملکرد Head انجام پیش بینی های Dense متراکم است. پیش بینی متراکم، پیش بینی نهایی است که از امتیاز اطمینان پیش بینی^{۱۱}، کلاس های احتمال^{۱۲} و بردار حاوی مختصات جعبه مرزی پیش بینی شده (مرکز، ارتفاع، عرض) تشکیل شده است. Yolov5 دارای Head یکسانی با Yolov3 برای شناسایی اشیاء میباشد. مراحل تشخیص مبتنی بر لنگر^{۱۳}، و دارای سه سطح دانه بندی جهت تشخیص میباشد.

Yolov5 دارای نسخه های مختلفی است که هر کدام ویژگی منحصر به فرد خود را دارند. این نسخه ها عبارتند از:

۱- yolov5-s نسخه کوچک

۲- yolov5-m نسخه متوسط

¹⁰Detection Neck and Detection Head

¹¹confidence score

¹²probability classes

¹³anchor-based

¹⁴latency

¹⁵computationally handicapped

¹⁶applications embedded

¹⁷residual structure

¹⁸non-linearities

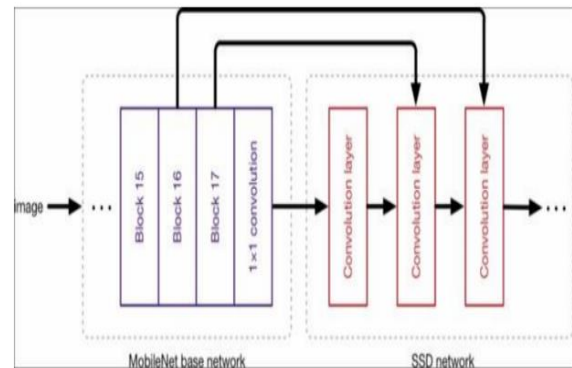
شده ذخیره شدند. هنگامی که تصاویر به خوبی برچسب گذاری شدند، همانطور که در شکل ۵ نشان داده شده است به مجموعه های آموزشی، اعتبار سنجی و آزمایش تقسیم شدند. اعمال چنین تقسیم بندی ضروری نیست، اما مجموعه اعتبارسنجی به بررسی تشخیص های اشتباه کمک می کند.



۲-۳- آموزش: داده ها به صورت دستی از این مدل ها با استفاده از پلتفرم Google Cloud بر روی پردازنده گرافیکی Nvidia Tesla T4 با ۱۶ گیگابایت رم آموزش داده شدند. نرخ یادگیری برای همه مدل ها ۰.۰۰۱ اولیه شد و هر ۱۰۰۰ مرحله آن را کاهش داده، هر سه مدل برای ۹۰۰ دوره آموزش داده شد و از آنجایی که مجموعه داده شامل تقریباً ۱۰۰۰ تصویر بود و اندازه دسته روی ۱۰ تنظیم گردید، بنابراین، آموزش تقریباً ۹۰۰۰۰ مرحله ادامه یافت. نمودار loss plot برای هر سه قبل از علامت پایان به نقطه ثابت رسید، با مدل های Yolo در حدود ۴۰۰۰۰ گام و در SSD ۵۰۰۰۰ گام تثبیت گردید. همانطور که قبلاً ذکر شد، از کل مجموعه داده، ۱۰۰۰ تصویر برای آزمایش استفاده شد و ۴۵۰ تصویر برای اعتبارسنجی و آزمایش از هم جدا شدند.

تقریباً در زمانی که اولین نسخه MobileNet معرفی شد، گوگل آشکار ساز تک شات (SSD) را برای برنامه هایی که به شدت به سرعت و دقت به طور مساوی وابسته هستند، منتشر کرد. همانطور که از نام خود پیداست، SSD اساساً چندین شی را در یک تصویر با استفاده از یک عکس شناسایی می کند.

MobileNet مدلی است که سرعت مناسبی دارد و تنها نقطه ضعف آن دقت آن است. SSD واقعاً برای مدل model since مفید بود زیرا ابزاری برای بهبود دقت و حفظ سرعت مدل ها داشت. الگوریتم SSD به گونه ای طراحی شده است که بتوان آن را با شبکه های مختلفی مانند MobileNet, Yolo و معماری VGG ادغام کرد. بنابراین MobileNet برای عملکرد برتر با SSD ادغام شد و MobileNet-SSD نامیده شد. این معماری یکپارچه در شکل ۴ نشان داده شده است.



(شکل-۴): بلوک دیاگرام معماری MobileNet-SSD [۸]

۴- نتایج و بحث

مقایسه هر سه مدل پس از تکمیل فرآیند آموزش انجام شد. این مدل ها با استفاده از سه دستگاه به نام های Jetso Nano n, Nvidia GTX 1660 Ti و Nvidia Tesla T4 شده اند. برای تعیین عملکرد آنها در یک واحد پردازش سطح بالا، سطح متوسط و سطح پایین انجام شده است. حال با توجه به عواملی که برای تشخیص این سه مدل باید مورد توجه قرار گیرند، از دو پارامتر تأثیرگذار استفاده شد، که می تواند مدلی را که برای موارد استفاده خاص مناسب است تعیین کند. میانگین دقت متوسط (mAP) و قابلیت پردازش مدل اندازه گیری شده از

۳- روش

۳-۱- مجموعه داده: داده ها به صورت دستی از تصاویر google images از کلاس های خاص، جمع آوری شده است. مجموعه ای از تصاویر دارای زوایای متعدد از انسان با ماسک استفاده گردیده است. پس از جمع آوری تمام تصاویر مناسب، به دلیل تغییرات ابعادی برای ارائه برش داده شدند.

تصاویر با وضوح ۴۱۶x۴۱۶ برای یک فرآیند آموزشی نسبتاً سریع برش داده شده اند. با افزایش این وضوح، زمان آموزش مورد نیاز برای مدل ها نیز افزایش می یابد. پس از پیش پردازش، تصاویر برچسب گذاری شدند و این حاشیه نویسی ها (شامل مکان جعبه های محدود کننده) بسته به نیاز مدل ها در قالب های کدگذاری

(جدول-۱): تحلیل عملکرد سه مدل

Model	mAP(%)	FPS		
		TeslaT4	1660 Ti	Jetson Nano
YOLOV3	54.3	80	21	8
YOLOV5s	37.6	100	28	15
MobileNet-SSD V2	33.7	94	26	15

۵- نتیجه گیری

در این مقاله، سه مدل مقایسه شده است و هر یک از آنها ویژگی های منحصر به فرد خود را نشان می دهند. هر یک از مدل ها در کاربرد مورد نیاز تشخیص ماسک موفق بودند، با YOLOV5s به دلیل ترکیب سرعت و دقت، بهینه ترین مدل برای استقرار در زمان واقعی است. دو مدل دیگر نیز مدل های بسیار مناسبی هستند که سناریوهای مورد استفاده متفاوتی دارند. می توان نتیجه گرفت که استفاده از برنامه های ببنایی کامپیوتری در زمان واقعی امکان پذیر است و همه این مدل ها برای تبدیل به محصولات قابل فروش مناسب هستند.

۶- منابع

- [1] Deepa, R., et al. "Comparison of Yolo, SSD, Faster RCNN for Real Time Tennis Ball Tracking for Action Decision Networks." 2019 International Conference on Advances in Computing and Communication Engineering (ICACCE). IEEE, 2019.
- [2] Liu, Yifan, et al. "Research on the Use of YOLOv5 Object Detection Algorithm in Mask Wearing Recognition." World Scientific Research Journal 6.11 (2020): 276-284.
- [3] Jocher, G., Stoken, A., Borovec, J., Changyu, L., & Hogan, A. (2020). ultralytics/yolov5: v3. 0. Zenodo.
- [4] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).
- [5] Chiu, Yu-Chen, et al. "Mobilenet-SSDv2: An improved object detection model for embedded systems." 2020 International Conference on System Science and Engineering (ICSSE). IEEE, 2020.
- [6] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.
- [7] R. Deepa, E. Tamilselvan, E. S. Abrar and S. Sampath, "Comparison of Yolo, SSD, Faster RCNN for Real Time Tennis Ball Tracking for

طریق فریم در ثانیه (fps) محاسبه گردید. خروجی مدل های آموزش دیده در شکل ۶ نشان داده شده است.

اغلب هنگام اعتبارسنجی عملکرد یک مدل، دقت بر سرعت در اولویت قرار می گیرد و فرض بر این است که GPU های کالیبر بالا به وفور در دسترس هستند و از این رو ضریب سرعت قابل تحمل خواهد بود، در حالی که برای سیستم های بلادرنگ، سرعت یک عامل مهمتر محسوب میگردد و چنین مدل هایی معمولاً وجود دارند.



(شکل-۶): خروجیهای معماری YOLOV5s (الف) ماسک استفاده نادرست تشخیص داده شده است (ب) ماسک به درستی پوشیده شده تشخیص داده شود (ج) هیچ ماسکی شناسایی نشده است.

برای اینکه تشخیص به صورت بلادرنگ در نظر گرفته شود، مقدار کلی قابل قبول عملکرد هر یک از مدل های آموزش دیده در جدول ۱ ارائه شده است. با در نظر گرفتن نکات فوق الذکر، می توان استنباط کرد که YOLOV5s بهترین مدل مناسب برای موقعیت های بلادرنگ با مقادیر بهینه دقت accuracy و fps است. می توان بحث کرد که MobileNet-SSD V2 تا حدودی سرعتی مشابه با YOLOV5s ارائه می دهد، اما در بخش دقت accuracy کمتر از آن است. برای اهداف بلادرنگ، سرعت یک عامل تعیین کننده است، اما دقت مدل نیز برای عملکرد روان^{۱۹} ضروری است. با آمدن به YOLOV3، این مدل از دقت بسیار خوبی برخوردار است، اما به سخت افزار محاسباتی فشرده نیاز دارد. اگر چنین وسیله ای در دسترس باشد، این مدل برای سرعت مورد نیاز کافی خواهد بود. بنابراین می توان گفت که بسته به نیاز کاربردهای مختلف می توان هر کدام از مدل ها را انتخاب نمود.

[20]Demirkol, A., Z. Demir, and E. Emre. *Lecture notes in computer science*. 2005. SPRINGER-VERLAG BERLIN.
[21]Din, N.U., et al., *A novel GAN-based network for unmasking of masked face*. *IEEE Access*, 2020. 8: p. 44276-44287.
[22]Long, X., et al., *PP-YOLO: An effective and efficient implementation of object detector*. *arXiv preprint arXiv:2007.12099*, 2020.



کورش داداش تبار احمدی مدرک کارشناسی ارشد خود را در سال ۱۳۸۳ از دانشگاه صنعتی مالک اشتر در رشته برق مخابرات و کارشناسی ارشد در سال ۱۳۸۹ در رشته مهندسی فناوری

اطلاعات از دانشگاه تربیت مدرس و مدرک دکتری تخصصی را در سال ۱۳۹۴ از دانشگاه صنعتی مالک اشتر در رشته مهندسی کامپیوتر گرایش هوش مصنوعی و در حال حاضر به عنوان استادیار و عضو هیئت علمی دانشگاه صنعتی مالک اشتر مشغول به کار میباشند. زمینه پژوهشی مورد علاقه ایشان عبارتند از: هوش مصنوعی، مخابرات ماهواره ای، فرماندهی و کنترل، شبکه های نرم افزار محور، پردازش تصویر، یادگیری عمیق



مریم فردی مدرک کارشناسی ارشد خود را در سال ۱۴۰۰ از دانشگاه صنعتی مالک اشتر در رشته مهندسی کامپیوتر گرایش هوش مصنوعی اخذ نموده اند

روش ارجاع به مقاله م.فردی، ک.داداش تبار احمدی. مقایسه شبکه های عصبی YOLOv3، YOLOv5s و MobileNet-SSD V2 برای تشخیص ماسک صورت در زمان واقعی. دو فصلنامه محاسبات و سامانه های توزیع شده. سال پنجم، شماره اول، شماره پیاپی ۹، صفحه ۴۹ تا ۵۵، سال ۱۴۰۱.

How to cite: Kurosh Dadashtabar Ahmadi, Maryam fardi. Compare neural networks, MobileNet-SSD V2 to YOLOv5s, YOLOv3 detect face masks in real time *Journal of Distributed Computing and Systems (JDACS)*, Vol 5, Issue 1, Page 49-55, 2022.

Action Decision Networks," 2019 *International Conference on Advances in Computing and Communication Engineering (ICACCE)*, 2019, pp. 1-4, doi: 10.1109/ICACCE46606.2019.9079965.
[8] Hollemans, Matthijs. (2018, April 22). "MobileNet Version 2." <https://machinethink.net/blog/mobilenetv2/>
[9] Chen, W., et al., *YOLO-face: a real-time face detector*. *The Visual Computer*, 2020: p. 1-9. *fluent functioning*.

[10]Barthakur, M. and K.K. Sarma. *Semantic Segmentation using K-means Clustering and Deep Learning in Satellite Image*. in *2019 2nd International Conference on Innovations in Electronics, Signal Processing and Communication (IESC)*. 2019. IEEE.
[11]Loey, M., et al., *Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection*. *Sustainable cities and society*, 2021. 102600.
[12]Loey, M., M. Naman, and H. Zayed, *Deep transfer learning in diagnosing leukemia in blood cells*. *Computers*, 2020. 9(2): p. 29.
[13]Redmon, J., et al. *You only look once: Unified, real-time object detection*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
[14]Yang, W. and Z. Jiachun. *Real-time face detection based on YOLO*. in *2018 1st IEEE international conference on knowledge innovation and invention (ICKI)*. 2018. IEEE.
[15]Redmon, J. and A. Farhadi. *YOLO9000: better, faster, stronger*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
[16]Mao, Q.-C., et al., *Mini-YOLOv3: real-time object detector for embedded applications*. *IEEE Access*, 2019. 7: p. 133529-133533.
[17]Redmon, J. and A. Farhadi, *Yolov3: An incremental improvement*. *arXiv preprint arXiv:1804.02767*, 2018.
[18]Bochkovskiy, A., C.-Y. Wang, and H.-Y.M. Liao, *Yolov4: Optimal speed and accuracy of object detection*. *arXiv preprint arXiv:2004.10934*, 2020.
[19]Degadwala, S., et al. *Yolo-v4 Deep Learning Model for Medical Face Mask Detection*. in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*. 2021. IEEE.