اولین کنفرانس ملی
انجمن علمی پارک‌های علم و فناوری و مراکز رشد ایران

سیستم تشخیص چهره مبتنی بر یادگیری ژرف مورد استفاده در زمان همه گیری کرونا

محمدپویا ملک[۱]، علی غفرانی[۲*]

۱- کارشناس ارشد، برق مخابرات، صدا وسیما، تهران
۲- کارشناس ارشد، مهندسی برق مخابرات، صدا و سیما، تهران
* تهران، صندوق پستی ۱۶۳۴۹۱۳۷۱۴، Mopoma1995@gmail.com

چکیده

در این پژوهش سعی شده است تا با ارایه مدلی مبتنی بر یادگیری ژرف و با حداقل تعداد پارامترها به ارائه مدلی پرداخته شد که توانایی اجرا در سیستم‌های توکار را داشته و همچنین از دقت قابل قبولی برخوردار باشد. علاوه بر این در دوران همه‌گیری بیماری کرونا و الزام استفاده از ماسک که اکثر افراد صورت خود را با ماسک پوشانیده‌اند که باعث ایجاد اختلال در سیستم‌های سنتی و حتی بسیاری از سیستم‌های مبتنی بر یادگیری ژرف گردیده است. بنابراین بیش از هر زمان دیگری نیازمند توسعه و پیاده سازی سیستم‌های تشخیص چهره‌ی مقاومی هستیم که توانایی تشخیص چهره افراد حتی با پوشش‌هایی همچون ماسک، شال و ... را داشته باشند. بر این اساس ما سیستمی ارائه نمودیم تا علاوه بر فائق آمدن بر مشکلات شناسایی چهره‌ی ماسک زده مدلی ارائه کنیم که با حجم کم و قابلیت پیاده سازی بر روی سیستم‌های با قدرت پردازش بالا بتوان به آنالیز بلادرنگ تصویر رسید. این مهم با استفاده از مجموعه دادگان وسیعی همچون WiderFace و دادگان منبع باز Kaggle انجام گرفته است که به دقتی در حدود ۹۹ درصد بر روی مجموعه دادگان تست دست یافتیم.

کلیدواژگان

تشخیص چهره، یادگیری عمیق، همه_گیری_ویروس_کرونا

Deep learning-based face Detection System Used During Corona Virus epidemic

MohammadPooya Malek[1], Ali Ghofrani[2*]

1- Department of Electrical Engineering, IRIBU, Tehran, Iran.
2- Department of Electrical Engineering, IRIBU, Tehran, Iran
* P.O.B. 1634913714 Tehran, Iran, Mopoma1995@gmail.com

Abstract

In this research, an attempt has been made to provide a model based on Deep learning with a minimum number of parameters that can be implemented in built-in systems and also has acceptable accuracy. In addition, during the epidemic of coronary Virus disease and the need to use a mask, most people have covered their faces with masks, which has disrupted traditional systems and even many systems based on deep learning. Therefore, more than ever, we need to develop and implement durable face Detection systems that have the ability to recognize people's faces, even with masks such as masks, scarves, etc. Based on this, we presented a system that, in addition to overcoming the problems of masked face recognition, provides a model that can achieve real-time image analysis with low volume and the ability to implement on high-processing systems. This has been done using extensive datasets such as WiderFace and Kaggle open source datasets, which we achieved with about 99% accuracy on test datasets.

Keywords:

Face_Detection, Deep_Learning, Corona_Virus_Pandemic

۱- مقدمه

تشخیص چهره که در سال‌های گذشته توجه زیادی را به خود جلب نموده است به تکنیکی جهت تعیین موقعیت مکانی چهره در یک فریم تصویر و یا در بخشی از ویدیو گفته می‌شود و به عنوان یکی از چالش‌های روز در حوزه‌ی هوش‌مصنوعی و بینایی‌ماشین مطرح گردیده است. سیستم‌های تشخیص چهره در دنیای امروز از اهمیت بالایی برخوردارند به این دلیل که می‌توانند به عنوان بخش اولیه‌ای از طیف وسیعی از کاربردها از جمله شمارش تعداد افراد حاضر در کنفرانس، تخمین سن، تخمین جنسیت فرد از روی چهره، اصالت‌سنجی چهره، شناسایی افراد جهت تردد، تشخیص احساسات و ... مورد استفاده قرار بگیرند.

الگوریتم‌های کلاسیک موجود جهت تشخیص چهره از عملکرد خوبی در مقایسه با روش‌های یادگیری ژرف برخوردار نبودند. بنابراین در دهه‌ی اخیر جای خود را به رویکردهای مبتنی بر یادگیری ژرف دادند. البته این افزایش دقت و عملکرد که در سیستم‌های مبتنی بر پادگیری ژرف به وجود آمده به سبب افزایش شدید تعداد پارامتر های آموزش پذیر است که بار محاسباتی زیادی را به زیرساخت های سخت افزاری اعمال می‌نماید و در عمل نیازمند پردازنده‌هایی قوی برای پردازش عملیات‌های مورد نیاز در این حوزه است.

در این مقاله سعی شده است تا با ارایه مدلی مبتنی بر یادگیری ژرف و با حداقل تعداد پارامترها به ارائه مدلی پرداخته شود که توانایی اجرا در سیستم‌های توکار را داشته و همچنین از دقت قابل قبولی برخوردار باشد. علاوه بر این در دوران همه‌گیری بیماری کرونا و الزام استفاده از ماسک که اکثر افراد صورت خود را با ماسک پوشانیده اند، باعث ایجاد اختلال در سیستم‌های سنتی و حتی بسیاری از سیستم‌های مبتنی بر یادگیری ژرف گردیده است. بنابراین بیش از هر زمان دیگری نیازمند توسعه و پیاده سازی سیستم‌های تشخیص چهره‌ی مقاومی هستیم که توانایی تشخیص چهره افراد حتی با پوشش‌هایی همچون ماسک، شال و ... را داشته باشند.

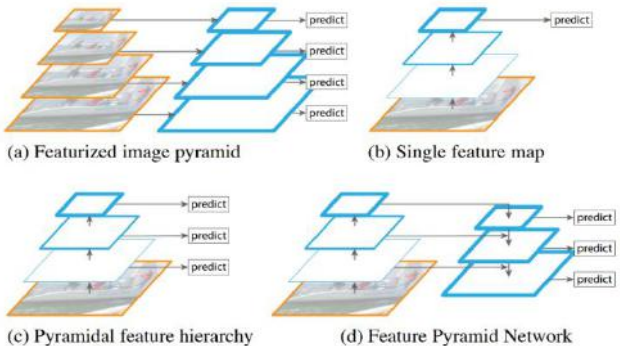
۲- کارهای گذشته

بسیاری از الگوریتم‌ها برای تشخیص نقاط کلیدی در شناسایی چهره به کار گرفته شدند. بعضی از آن‌ها با استفاده از تکنیک هایی مانند پوشانیدن و یا ماسک کردن بخشی از صورت بطور مثال بینی، چشم ها، دهان، گونه‌ها و ... سعی در تشخیص مهمترین نقاط کلیدی نمودند. در [1] که به آنالیز نقاط کلیدی مختلف بر اساس ویژگی‌های Gabor پرداخته بود، مشخص شد که پوشانیدن دهان بیش از بینی در سیستم تشخیص اثر گذار است و با پوشانیدن دهان عملکرد سیستم کاهش بیشتری پیدا می‌کند. البته در سالهای اخیر و با رشد شبکه‌های عصبی کانولوشنی این مشکلات تا حد قابل قبولی به کمک سیستم‌های یادگیری عمیق کاهش یافته است اما هنوز موارد زیادی وجود دارد که نیاز به بهبود دارند. از این رو در این مقاله بر آن شدیم تا روشی نوین در جهت تشخیص چهره در تصویر ارائه دهیم تا با استفاده از چهره‌ی بدست آمده و همچنین شرایط محیطی موجود در تصویر به آنالیز و شناسایی چهره‌های ماسک دار یا بدون ماسک در تصویر بپردازیم.

برای تشخیص چهره در تصاویر الگوریتم‌های متعددی ارائه گردیده است. اولین موردی که در سال ۲۰۱۷ با دقت قابل قبولی به شناسایی چهره پرداخت، الگوریتمی به نام صورت‌های کوچک [۱]بود[2]. این الگوریتم پایه‌ی بسیاری از تحقیقات شد و باب جدیدی را در این حوزه باز نمود. اگرچه در شناسایی اشیا گام‌های بلندی برداشته شده است، اما یکی از چالش‌های باز باقی مانده کشف اشیا کوچک در تصاویر است. در Tiny سه جنبه از مسئله را در زمینه یافتن چهره‌های کوچک مورد بررسی قرار دادند: نقش **تغییر مقیاس**، **وضوح تصویر** و **استدلال محتوی**. بعد از Tiny تا مدتها سراغ روش‌های مبتنی بر هرم به دلیل حجم محاسبات و حافظه زیاد نرفتند، تا اینکه الگوریتم Feature Pyramid Networks معرفی شد[3]. Feature pyramids یک جز اساسی در سیستم‌های تشخیص اشیا با در نظر گرفتن مقیاس‌های مختلف است. در آن از ساختار سلسله مراتبی هرمی چندگانه ذاتی شبکه‌های کانولوشنی عمیق برای ساخت هرم‌های مشخصه با بار محاسباتی اضافی حاشیه ای رونمایی کردند. یک معماری از بالا به پایین با اتصالات جانبی برای ساختن نقشه‌های

[۱] Tiny Faces (Tiny به اختصار)

ویژگی[1] معنایی سطح بالا در همه مقیاس‌ها ایجاد کردند. این معماری که Feature Pyramid Network (FPN) نامیده می‌شود، پیشرفت چشمگیری را به عنوان یک ویژگی اضافی در چندین رقابت بین المللی از خود نشان داده است. این روش می‌تواند با سرعت ۵ فریم در ثانیه بر روی یک GPU اجرا شود و بنابراین یک راه حل عملی و دقیق برای تشخیص



شکل ۱ (a) استفاده از هرم تصویر برای ساخت هرم ویژگی. ویژگی‌ها در هر یک از مقیاس های تصویر به طور مستقل محاسبه می شوند، که کند است. (b) سیستم های اخیر تشخیص این را انتخاب کرده اند که فقط از ویژگی های تک مقیاس برای تشخیص سریعتر استفاده کنند. (c) یک گزینه جایگزین استفاده مجدد از سلسله مراتب ویژگی های هرمی محاسبه شده توسط ConvNet است به گونه ای که هرم تصویری برجسته ای باشد. (d) Feature Pyramid Network سریع مانند (b) و (c) است، اما دقیق تر است. در این شکل، مستطیل های آبی پر رنگ نشان گر درک معنایی بهتر در مقابل سایر لایه‌ها میباشد.

شی چند مقیاسی است.

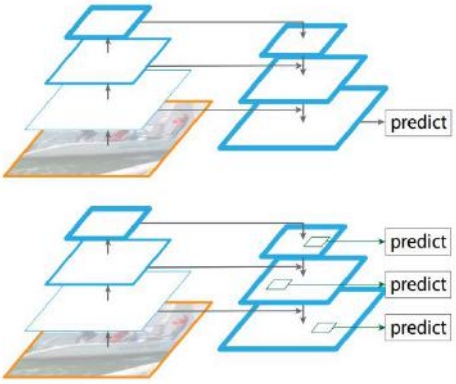
شناخت اشیا در مقیاس‌های کاملاً متفاوت یک چالش اساسی در بینایی رایانه است. ساختار هرمی اساس یک راه حل استاندارد را تشکیل میدهند. در شکل ۱ (a) این اهرام مقیاس ناپذیر هستند به این معنا که تغییر مقیاس یک جسم با تغییر سطح آن در هرم جبران می‌شود. این ویژگی یک مدل را قادر می‌سازد تا اشیا را در طیف وسیعی از مقیاس‌ها با اسکن‌های مختلف مدل از هر دو موقعیت و سطح هرم تشخیص دهد. از اهرام تصویر برجسته به شدت در عصر استخراج دستی ویژگی استفاده می‌شد. طوری که ردیاب‌های جسمی مانند DPM برای دستیابی به نتایج خوب به نمونه برداری متراکم در مقیاس نیاز داشتند. برای کارهای تشخیصی، مهندسی ویژگی تا حد زیادی با ویژگی‌های محاسبه شده توسط شبکه‌های کانولوشن عمیق (ConvNets) جایگزین شده اند. جدا از توانایی نشان دادن معناشناسی سطح بالاتر، ConvNets همچنین در مقیاس از واریانس قوی تری بهره می‌برد و بنابراین تشخیص ویژگی‌های محاسبه شده در یک مقیاس ورودی یکسان را بهبود می‌بخشد. در شکل ۱ (b) اما حتی با وجود این استحکام، به اهرام نیز برای دستیابی به دقیق ترین نتایج نیاز است. مزیت اصلی برجسته سازی هر سطح از هرم تصویر این است که یک نمایش ویژگی چند مقیاس را تولید می‌کند که در آن تمام سطوح از نظر محتوی و ویژگی قوی هستند، از جمله در سطوح با وضوح بالا. با این وجود، برجسته سازی هر سطح از هرم تصویر دارای محدودیت هایی است. زمان استنباط به طور قابل توجهی افزایش می‌یابد (به عنوان مثال ، چهار برابر می‌شود)، و این روش را برای کارهای آنی دنیای واقعی[2] غیر عملی می‌کند. علاوه بر این، آموزش شبکه‌های عمیق از انتها به انتها بر روی هرم تصویر از نظر حافظه غیرقابل اجرا است و بنابراین در صورت بهره برداری از هرم‌های تصویر فقط در زمان تست استفاده می‌شود.

با این حال، اهرام تصویری تنها راه محاسبه بازنمایی ویژگی چند مقیاسی نیستند. یک ConvNet عمیق سلسله مراتب ویژگی را لایه به لایه محاسبه می‌کند و با لایه‌های زیر نمونه، در حقیقت به طور ذاتی یک سلسله مراتب ویژگی شکل هرمی چند مقیاس است. این سلسله مراتب ویژگی در شبکه نقشه‌های ویژگی مشخصی از تفکیک‌های فضایی مختلف را تولید می‌کند، البته اختلافات محتویی و ویژگی ایجاد شده که به دلیل عمق‌های مختلف ایجاد می‌شوند. نقشه‌های با وضوح بالا دارای ویژگی‌های سطح پایینی هستند که به ظرفیت بازیابی آنها برای تشخیص شی آسیب می‌رسانند. شناساگر تک شات (SSD) یکی از اولین موارد است که سعی در استفاده از سلسله مراتب ویژگی‌های هرمی ConvNet دارد و

1 Feature Maps
2 Real time and Real world

همانند هرم تصویری برجسته است. شکل ۱(c) در حالت ایده آل، هرم به سبک SSD را نشان می‌دهد که از نقشه ویژگی‌های چند مقیاس از لایه‌های مختلف محاسبه شده، در حرکت به جلو[1] استفاده مجدد می‌کند و بنابراین بدون اضافه کردن بار محاسباتی زیاد به الگوریتمی قابل قبول دست می‌یابد. اما برای جلوگیری از استفاده از حذف ویژگی‌های سطح پایین، SSD از استفاده مجدد از لایه‌های محاسبه شده چشم پوشی می‌کند و به جای آن هرم را از بالا در شبکه ایجاد می‌کند (به عنوان مثال، conv4 از شبکه‌های VGG) و سپس با افزودن چندین لایه جدید باعث بهبود قدرت درک شبکه می‌گردد. بنابراین فرصت استفاده مجدد از نقشه‌های با وضوح بالاتر در سلسله مراتب ویژگی‌ها را از دست می‌دهد. در FPN نشان داده شد که این موارد برای تشخیص اشیا کوچک مهم هستند. هدف اصلی در FPN استفاده از شکل هرمی سلسله مراتب ویژگی‌های ConvNet است در حالی که از یک هرم ویژگی که دارای محتوی و ویژگی‌های قوی در همه مقیاس‌ها باشد، استفاده می‌کند. برای دستیابی به این هدف، آن‌ها به معماریی تکیه کردند که از طریق یک رویکرد از بالا به پایین با اتصالات جانبی و از ترکیب نقشه‌های ویژگی با وضوح نسبتا پایین که ویژگی‌های قوی و معتبری دارند با نقشه‌های ویژگی وضوح بالا که ویژگی‌های ضعیفی داخل خود دارند به دقت بسیار قایل قبول و بار محاسباتی مناسبی می‌انجامد همانطور که در شکل ۱ (d) نشان داده شده است، بهره بردند. نتیجه یک هرم ویژگی است که دارای معناشناسی غنی در تمام سطوح است و با سرعت بسیار مناسبی از یک تصویر تک مقیاس ورودی ساخته شده است. به عبارت دیگر، نشان دادند که چگونه اهرام ویژگی درون شبکه ای ایجاد کنیم که بتواند بدون جایگزینی قدرت نمایش، سرعت یا حافظه، هرم‌های تصویر برجسته را جایگزین کند و از آنها بهره ببرد. در تحقیقات مختلف معماری‌های مشابهی که از بالا به پایین و جستار استفاده می‌کنندکه اهداف آنها تولید یک نقشه ویژگی سطح بالا با وضوح خوب است که قرار است پیش بینی‌ها روی آن انجام شود همانند قسمت بالای شکل ۲. برعکس آن در روش FPN از معماری به عنوان یک هرم ویژگی استفاده می‌کند که در آن پیش بینی‌ها (به عنوان مثال تشخیص چهره) به طور مستقل در هر سطح انجام می‌شود که در شکل ۲ قسمت پایین شکل نشان داده شده است.

بنابراین در FPN هدف، استفاده از سلسله مراتب ویژگی‌های هرمی ConvNet است که دارای ویژگی‌های محتوایی از سطح پایین به سطح بالا است و ساختن یک هرم ویژگی با معناشناسی سطح بالا در کل تصویر را نشان دادند.



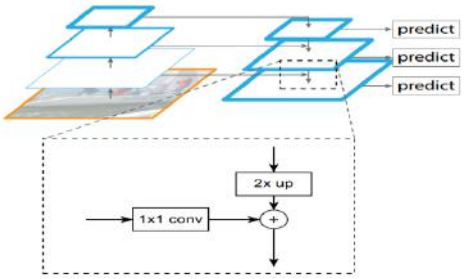
شکل۲ بخش بالایی: معماری از بالا به پایین با اتصالات قبلی، جایی که پیش بینی‌ها در بهترین سطح با قوی ترین ویژگی‌ها انجام می شود. بخش پایینی: مدل FPN دارای ساختاری مشابه است اما از آن به عنوان یک هرم ویژگی استفاده می کنند، که پیش بینی‌ها به طور مستقل در همه سطوح انجام میگردد.

مسیر از پایین به بالا محاسبه رو به جلوی اجزای اصلی شبکه‌ی کانولوشنی است که یک سلسله مراتب ویژگی متشکل از نقشه ویژگی را در چندین مقیاس با درجه ای از ۲ (توانی از دو) محاسبه می‌کند. اغلب لایه‌های زیادی وجود دارد که نقشه‌های خروجی با همان اندازه تولید می‌کنند و می‌گویند این لایه‌ها در همان مرحله از شبکه قرار دارند. برای هرم ویژگی و برای هر مرحله یک سطح هرم تعریف می‌شود. که خروجی آخرین لایه هر مرحله را به عنوان مجموعه مرجع نقشه‌های ویژگی انتخاب می‌کند، که باعث غنی تر شدن هرم ویژگی‌های و درنتیجه انتخاب و استخراج ویژگی می‌گردد.

1 Feed Forward

این یک انتخاب طبیعی است زیرا عمیق ترین لایه هر مرحله از قوی ترین ویژگی‌ها برخوردار است که باعث بهبود عملکرد الگوریتم می‌شود. به طور خاص ، برای ResNets از فعال سازی ویژگی‌های تولید شده توسط آخرین بلوک باقی مانده هر مرحله استفاده می‌شود. که در FPN خروجی این آخرین بلوک‌های باقیمانده را به عنوان { C2، C3، C4، C5} برای خروجی‌های conv2، conv3، conv4 و conv5 نشان داده اند و با توجه به اینکه لایه‌ی اول کانولوشن به دلیل استفاده از حافظه‌ی زیاد داخل هرم قرار نگرفته است که موجب سرعت بخشیدن و کاهش حافظه‌ی مصرفی می‌گردد.

در مسیرهای بالا به پایین و اتصالات جانبی می‌توان به این نکته پرداخت که با نمونه برداری از نقشه‌های ویژگی‌های درشت، اما از لحاظ محتویِ قوی تر، از سطوح بالاتر هرم، ویژگی‌های با وضوح بالاتر را بیان می‌کنند و به استخراج اینگونه ویژگی‌ها می‌پردازند. این ویژگی‌ها سپس با ویژگی هایی از مسیر پایین به بالا از طریق اتصالات جانبی بهبود می‌یابند. هر اتصال جانبی نقشه‌های ویژگی همان اندازه فضایی را از مسیر پایین به بالا را با مسیر بالا به پایین ادغام می‌کند. نقشه ویژگی پایین به بالا از ویژگی‌های محتوایی سطح پایین است، اما فعال سازی‌های آن با دقت بیشتری انجام می‌شود زیرا چندین بار کمتر نمونه برداری صورت گرفته است. شکل ۳ بلوک ساختاری را نشان می‌دهد که نقشه‌های ویژگی‌های بالا به پایین را می‌سازد.



شکل۳ یک بلوک ساختاری که اتصالات جانبی را نشان می دهد و مسیر از بالا به پایین، با اضافه شدن نقشه ویژگی‌ها از مسیر پایین به بالا ادغام شده است.

در نقشه ویژگی‌های با وضوح پایین تر نمونه برداری با دقت دو برابری انجام می‌گردد یا اصطلاحا برونیابی با توجه به نزدیکترین همسایگی انجام می‌شود. حال این نقشه‌ی ویژگی‌های دو برابر شده با نقشه‌ی متناسب در مسیر پایین به بالا ترکیب می‌گردد. این فرایند تا زمانی تکرار می‌شود که بهترین نقشه با وضوح متناسب تولید شود. به این ترتیب به ویژگی‌های مقاومی دست پیدا کرده اند که می‌تواند به عنوان عناصر اصلی باشند. برای تکرار، به سادگی یک لایه کانولوشن ۱*۱ روی C5 کانوالو می‌گردد تا نقشه با وضوح درشت تولید شود. سرانجام، برای تولید نقشه مشخصه ویژگی نهایی، یک عملیات کانولوشنی ۳*۳ بر روی هر کدام از نقشه‌های ویژگی ترکیب شده اعمال می‌گردد تا اثر درهم روندگی[1] حاصل از برونیابی را از بین ببرد و نقشه ویژگی نهایی خروجی مناسب را تولید نماید. این مجموعه نهایی از نقشه‌های ویژگی {P2، P3، P4، P5}، به ترتیب مربوط به { C2، C3، C4، C5} است که از همان اندازه‌های مکانی هستند.

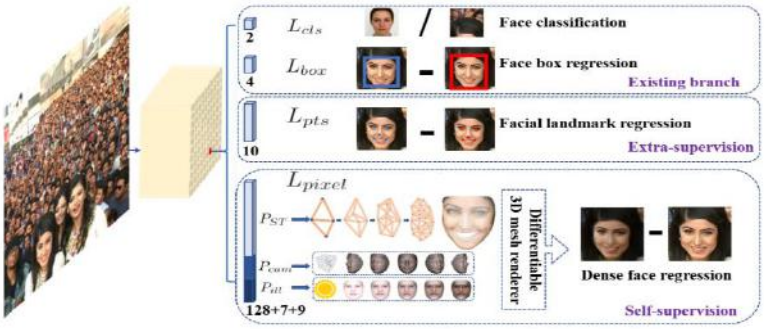
بنابراین در FPN یک الگوریتم مفید برای کاهش بار محاسباتی و همچنین افزایش سرعت معرفی شده است که از ساختار هرمی تبعیت می‌نماید بدین صورت که به جای تغییر مقیاس تصویر اصلی و محاسبه‌ی ویژگی‌ها در هر لایه از تصویر تغییر مقیاس داده شده، ابتدا نقشه‌های ویژگی را در تصویر اصلی تهیه می‌نماید و با تغییر مقیاس این نقشه‌های ویژگی در لایه‌های مختلف به دنبال ویژگی‌های مقاوم و پایدار می‌گردد و با استخراج آنها در نهایت به دسته بندی و رگرسیون می‌پردازد. این کار باعث افزایش دقت، کاهش بار محاسباتی و افزایش سرعت محاسبات گردیده است.

بعد از FPN مهمترین شبکه ای که معرفی گردید شبکه‌ی Retina Face بود [4] که با استفاده از مزایای یادگیری چند منظوره مشترک تحت نظارت و یادگیری خود نظارتی، محلی سازی پیکسل را در مقیاس‌های مختلف صورت انجام می‌دهد.

1 Aliasing

۳- الگوریتم ارائه شده

به طور خاص، Retina دو مرحله‌ی اساسی را اعمال می‌کند: ۱) به طور دستی پنج نشانه از چهره را در مجموعه داده WIDER FACE اضافه کرده اند و با کمک این سیگنال نظارتیِ اضافی، بهبود قابل توجهی در تشخیص چهره‌هایی با امکان شناسایی سخت، ایجاد نمودند. ۲) همچنین یک شاخه کدگشایِ مشِ[1] خود نظارتی برای پیش بینی اطلاعات سه



شکل ۴ روش مکانیابی و شناسایی چهره‌ی "یک مرحله ای" پیگسلیِ ارائه شده توسط Retina Face، با یادگیری های فوق نظارتی و خود نظارتی که به صورت موازی با شاخه های طبقه بندی مستطیلی و رگرسیون قرار گرفته است و به استخراج و انتخاب بهتر ویژگی کمک شایانی می نماید. هر Anchor مثبت (کاندید انتخاب چهره در آن بخش از تصویر) باید چهار خروجی زیر را نشان دهد: (۱) امتیاز و درصد احتمال وجود چهره (۲) مستطیل دربردارنده‌ی چهره، (۳) پنج علامت مشخصه چهره و (۴) رئوس متراکم سه بعدی چهره که در صفحه تصویر پیش‌بینی می شود.

بعدی صورت به شکل پیکسلی به موازات شاخه‌های نظارت شده موجود اضافه کردند. هر دوی این موارد با استفاده از شبکه‌های سبک وزن عصبی عمیق باعث بهبود عملکرد و البته دقت و سرعت اجرا گردیده است. RetinaFace می‌تواند بر روی یک هسته CPU به صورت بلادرنگ برای یک تصویر با وضوح VGA با دقت قابل قبولی اجرا گردد.

همانطور که گفته شد مکانیابی چهره به صورت خودکار قدم اولیه برای آنالیز پردازش‌های چهره است. در روشهای سنتی مکانیابی چهره عمدتا به تخمین مستطیل‌های محدود کننده چهره (مرز چهره) بدون هیچ مقیاس و موقعیت قبلی پرداخته می‌شد. در Retina به تعریف وسیع تری از مکانیابی چهره اشاره شده است که شامل تشخیص چهره، ترازبندی چهره، تجزیه چهره به صورت پیکسلی و رگرسیون متناسب با تراکم سه بعدی چهره می‌باشد. این نوع مکانیابی متراکم چهره اطلاعات دقیقی از وضعیت صورت برای همه مقیاس‌های مختلفِ چهره را فراهم می‌کند. برخلاف شناسایی اشیاء، تشخیص چهره دارای تغییرات نسبتی کوچکتری است (از ۱:۱ به ۱:۱٫۵) اما دارای تغییرات مقیاسی بسیار بزرگتری (از چندین پیکسل تا هزار پیکسل) است. جدیدترین روشهای پیشرفته بر روی طراحی تک مرحله ای تمرکز دارند که به طور متراکم از مکان‌ها و مقیاس‌های هرم ویژگی نمونه برداری می‌کند که عملکرد بهتر و سرعت بیشتری نسبت به روشهای دو مرحله‌ای دارند. Retina به پیروی از این الگو، چارچوب تشخیص چهره تک مرحله ای را ارائه دادند که با بهره گیری از عملیات‌های چند وظیفگی که از سیگنالهای فوق نظارتی و خود نظارتی به مکانیابی چهره که در شکل ۴ نشان داده شده است، پرداخته اند.

همانطور که گفته شد در این الگوریتم ابتدا چهره شناسایی شده و سپس پنج نشانه‌ی مشخص که شامل دو نشانه برای چشم‌ها، یک نشانه برای بینی و دو نشانه برای دهان در نظر گرفته شده است. در الگوریتم Retina با الهام از Mask R-CNN با افزودن شاخه ای برای پیش بینی ماسک جسم به موازات شاخه موجود برای محدود کردن مستطیل تشخیص چهره و رگرسیون، عملکرد تشخیصی را به طرز چشمگیری بهبود دادند. Retina همچنین با ایده ارائه شده در کدگشایِ مش که با بهره گیری از گرافهای پیچشی[2] روی شکل و بافت اتصالات، به سرعت بسیار بالایی حتی بالاتر از پردازش بلادرنگ در رزولوشن VGA دست یافتند در حقیقت کدگشای مش یک روش تجزیه نمودار بر اساس فیلتراسیون طیفی سریع موضعی است که البته استفاده از کدگشای مش که به ساخت تصاویر سه بعدی از تصویر ورودی کمک می‌کند در

1 Mesh
2 Graph Convolutional

نهایت باعث افزایش دقت سیستم نیز می‌گردد. با این حال، استفاده از کد گشای مش تک مرحله ای با یکسری چالش همراه هست: ۱) ارزیابی دقیق پارامترهای دوربین دشوار است ۲) استفاده از آشکال پنهان متصل و بافت‌های پیشبینی شده از یک بردار ویژگی تنها (مثلا کانولوشن ۱*۱ روی هرم ویژگی) به جای ویژگیهای RoI[1]، که نشان دهنده‌ی خطر تغییر ویژگی است.

روش‌های دو مرحله ای از مکانیزم "پیشنهاد و پالایش" استفاده می کنند بدین معنی که ابتدا مواردی که به عنوان کاندید از وجود تصویر هستند انتخاب شده و سپس در مرحله‌ی دوم به حذف بعضی از این کاندیدا بر اساس معیارهای مشخص پرداخته می‌شود بنابراین این روشها دارای دقت بالایی در مکانیابی چهره هستند. در مقابل، روشهای تک مرحله ای از مکانها و مقیاسهای صورت به طور متراکم نمونه برداری شده‌اند، که منجر به ایجاد نمونه‌های مثبت و منفی بسیار نامتعادل در حین آموزش می‌شوند. برای رسیدگی به این عدم تعادل، روشهای نمونه برداری و وزنی به طور گسترده ای مورد استفاده قرار گرفتند. در مقایسه با روشهای دو مرحله‌ای، روشهای یک مرحله ای کارآمدتر هستند و سرعت فراخوانی بالاتری دارند اما در معرض خطر دستیابی به نرخ مثبت کاذب[2] بالاتر و دقت مکانیابی چهره پایین‌تر هستند.

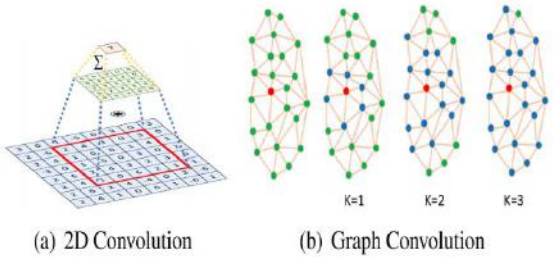
از آنجا که فرایند بهینه سازی یک مدل با کمینه کردن خطا شروع می‌شود و در مدل‌های هوش مصنوعی برای محاسبه‌ی خطا از عبارت "تلفات"[3] استفاده می‌شود بنابراین برای بهینه سازی مدل باید مدل به طرزی طراحس شود که کمینه‌ی تلفات را داشته باشیم. در Retina برای هر Anchor تلفاتی به شکل زیر محاسبه شده است.

$$L = L_{cls}(P_i, P_i^*) + \lambda_1 P_i^* L_{box}(t_i, t_i^*) + \lambda_2 P_i^* L_{pts}(l_i, l_i^*) + \lambda_3 P_i^* L_{pixel}$$ معادله ۱

همانطور که در شکل ۴ ملاحظه می‌نمایید و در فرمول بالا موجود است اولین تلفات مربوط به تلفات دسته بندی چهره است (L_{cls}) که در آن از تلفات Softmax برای تشخیص وجود چهره یا عدم وجود (باینری) استفاده شده است. دومین تلفات که هم در شکل و هم در فرمول موجود است، تلفات مربوط به رگرسیون است (L_{Box}) که در حقیقت نشان دهنده‌ی مختصات همان مستطیلی است که بر روی بخش کاندید وجود چهره قرار می‌گیرد. سومین تلفات مربوط به نقاط کلیدی چهره است (L_{pts}) که پنج نقطه مهم چهره را شامل می‌شود که پیشتر توضیح داده شد و شامل دو چشم، بینی، و گوشه‌ی لب‌ها می‌باشد. چهارمین و آخرین تلفاتی که در Retina در نظر گرفته شده است تلفات مرتبط با رگرسیون تراکم هست (L_{Pixel}) که در پایین ترین بخش شکل ۴ نشان داده شده است که بر اثر تغییرات زاویه‌ی دید سه بعدی یا تغییرات شدت روشنایی و مواردی از این دست محاسبه خواهد شد و بر اساس ترکیب تمام این موارد همانطور که در فرمول بالا نشان داده شده است سعی در کمینه کردن تلفات کلی سیستم نمودند. تمامی این موارد در شکل ۳-۲۹ به صورت کلی در سه بخش دسته بندی گردیده است بخش اول که شاخه‌ی موجود نام گرفته شامل تلفات دسته بندی و تلفات مستطیلی می‌باشد. بخش دوم که با استفاده از یادگیری فوق نظارتی (نظارت شده) ایجاد گردیده است و شامل نقاط کلیدی چهره است (پنج نقطه‌ی مهم) و در بخش سوم نیز از الگوریتم خود نظارتی (خودآموز) جهت کاهش تلفات حاصل از سه بعدی سازی متراکم چهره و تغییرات شدت روشنایی و زاویه دید می‌باشد.

در خصوص مفهوم نمودارهای کانولوشنی[4] معرفی شده که در شکل ۵ نیز نشان داده شده است، می‌توان این طور توضیح داد که یک عملیات کانولشن دوبعدی در حقیقت مجموع همسایگی‌های هسته(فیلتر)ی وزن داری در میدان یک شبکه اقلیدسی است. در نمودارهای کانولوشن نیز همان مفهوم را نشان می‌دهد با این تفاوت که فاصله همسایه با شمارش حداقل تعداد بر روی نقاط نمودار محاسبه می‌شود که در شکل ۵ (b) نشان داده شده است.

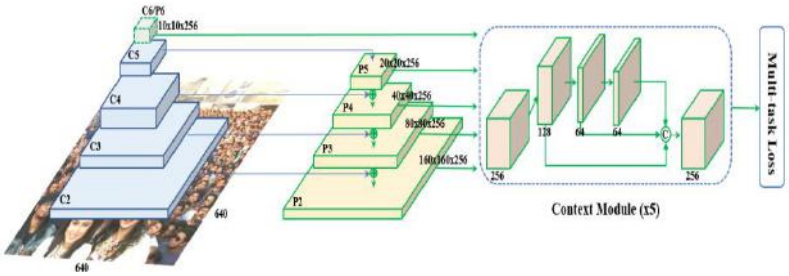
1 Region of interest
2 False Positive
3 Loss
4 Graph Convolution



شکل ۵ (a) کانولوشن دو بعدی حاصل جمع همسایگی های هسته در میدان شبکه اقلیدسی است. (b) نمودارهای کانولوشن نیز به صورت مجموع همسایگی هسته ای است، اما فاصله همسایه با شمارش حداقل تعداد لبه های اتصال دو راس بر روی نمودار محاسبه می شود.

در حقیقت Retina بعد از محاسبه‌ی اشکال و بافت پارامترهای مرتبط با چهره یک بازبینی با مضمون تغییر در شدت روشنایی چهره، تغییر در زاویه‌ی دید یا زاویه‌ی دوربین و مواردی از این دست که به صورت خود نظارتی است اعمال می‌گردد تا دقت الگوریتم تا حد بسیار قابل قبولی بالا رود. همانطور که در شکل ۶ ملاحظه می‌نمایید، RetinaFace برپایه‌ی ساختار هرمی ساخته شده که هر کدام دارای ماژول‌های محتوایی مختص به خود می‌باشند و از معماری ResNet بهره می‌برد که به علت ساختار تقریبا یکسان با الگوریتم معرفی شده‌ی FPN به توضیحات داده شده بسنده می‌شود. بنابراین Retina با استفاده از نمودارهای کانولشنی و معرفی یک الگوریتم تک مرحله ای به همراه تلفات مختلف و در نهایت یکپارچه سازی تمامی موارد توضیح داده شده یک الگوریتم با دقت بالا را معرفی نموده است که می‌تواند پایه‌ی سیستم تشخیص حالات چهره‌ی ما در بخش تشخیص چهره باشد. البته در ادامه سیستمی نوآورانه برای بهبود عملکرد سیستم معرفی می‌شود اما می‌توان گفت پایه‌های الگوریتمی که در ادامه ارائه شده است مواردی است که تا کنون به جزئیات بیان گردیده است و در ادامه تنها به کلیت اشاره خواهد شد. البته لازم به ذکر است که در الگوریتم Retina پیدا کردن نقاط کلیدی در ابتدا به صورت دستی بر روی دیتاست widerFace انجام گردیده است که در شکل ۷ نشان داده شده است.

در پروژه‌ی مورد نظر، ما ابتدا با استفاده از معماری ResNet که در بخش‌های قبل توضیح داده شده است یک پایه برای



شکل ۶ تصویری کلی از روش مکانیابی چهره متراکم تک مرحله ای معرفی شده توسط RetinaFace که بر اساس اهرام ویژگی با ماژول های زمینه مستقل طراحی شده است. به دنبال ماژول های زمینه، یک تلفات چند کاره را برای هر Anchor محاسبه می کند.

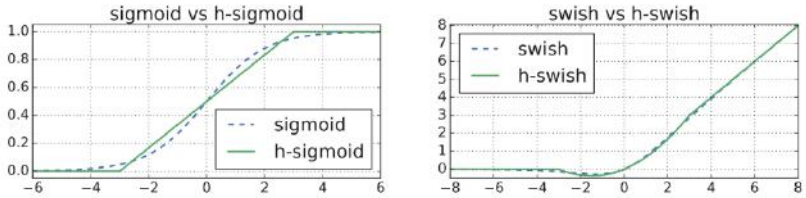
الگوریتم تعریف می‌نماییم. در این پژوه با ایده گرفتن از روش‌های RetinaFace و FPN در ابتدا همانند شکل ۵ ابتدا تصویر ورودی داده شده به شبکه را با استفاده از الگوریتم ارائه شده در FPN و محاسبه‌ی نقشه‌ی ویژگی از تصویر اصلی محاسبه می‌نماییم. سپس و در ادامه نقشه‌ی ویژگی بدست آمده از تصویر اصلی را با ضرایب مختلف از ۰,۰۱۵۵، ۰,۰۳۱۲، ۰,۰۶۲۵، ۰,۱۲۵ ، ۰,۲۵، ۰,۵ تقسیم می‌نماییم تا نقشه‌های ویژگی متناسب برای هر لایه از تصویر اصلی بدست آید. با این روش ما ویژگی‌های با ابعاد مختلف از هر مقیاس را استخراج نمودیم. یعنی ۶ نوع مقیاس مختلف از نقشه‌های ویژگی تصویر اصلی داریم. در ادامه همانند روشی که در FPN معرفی شد برای مسیر بالا به پایین و بدست آوردن نقشه‌های ویژگی مقاوم تر خروجی‌های مسیر پایین به بالا را نیز که با همان اندازه هستند به مدل اضافه می‌نماییم. نشان داده شد که این کار باعث افزایش دقت و در نتیجه خروجی بهتر خواهد گردید. از آنجا که مدل معرفی شده توسط Retina دارای ۵ لایه‌ی C2 تا C6 بود و اینجا ما ۶ لایه‌ی C2 تا C7 داریم دو لایه‌ی آخر را که دارای ویژگی‌های غنی تر و مقاوم تری هستند را نگاه داشته و سایر لایه‌ها را نیز همانند الگوریتم FPN در مسیر از بالا با پایین استفاده می‌نماییم. عملا تا اینجا ما یک استخراج کننده‌ی ویژگی ۶ لایه‌ای با دقت بالا ساخته‌ایم که می‌تواند ویژگی‌های قوی و مقاوم را از دلِ نقشه‌های

ویژگی مربوط به مقیاس‌های مختلف استخراج نماید و البته در ادامه می‌توان از آن جهت انتخاب بهترین کاندید مورد نظر استفاده نمود. در این مرحله پس از استخراج ویژگی، دو لایه‌ی آخر نقشه‌های ویژگی حاصل شده از مرحله‌ی قبل را مستقیما و بدون دخالت در مسیر بالا به پایین به بلوک‌های شبکه عصبی که در اینجا ما بلوک‌های Inception را انتخاب کردیم می‌دهیم تا شبکه از این ویژگی‌های قوی و مقاومی که در دولایه‌ی آخر هرم وجود دارد، جهت آموزش نهایت بهره را ببرد. از طرف دیگر خروجی‌های لایه‌های ترکیبی C2 تا C5 که در مسیرهای بالا به پایین نیز مشارک داشته و با لایه‌های هم اندازه‌ی خود ترکیب شده اند را نیز به صورت مجزا و لایه به لایه به شبکه‌ی عصبی Inception خود می دهیم، تا از این نقشه‌های ویژگی و البته ویژگی‌های استخراج شده‌ای که تقویت شده اند نیز بهره ببرد. حال در این مرحله بر خلاف آنچه که در الگوریتم Retina اتفاق می افتاد، یعنی داخل شدن تمامی این ویژگی‌ها و نقشه‌های ویژگی به یک بلوک واحد با ماژول‌های مختلف (همانند ماژول‌های تغییر روشنایی و تغییر زاویه دید)، ما از بلوک‌های مختلف با ماژول‌های یکسان داخل همه‌ی بلوک‌ها استفاده کردیم. بدین صورت که خروجی هر لایه Inception به صورت مجزا به بلوک‌های جداگانه‌ای وارد می‌شوند که داخل هر کدام از این بلوک‌ها می‌تواند بسته به شماره‌ی لایه‌ی ورودی به Inception عملیات‌های متفاوتی انجام گردد. به طور مثال می‌توان برای لایه‌های آخر هرم که اطلاعات و ویژگی‌های مهم و مقاومی دارند، تمامی روش‌های ارائه شده در الگوریتم RetinaFace را استفاده نمود(روش‌هایی همچون نمودارهای کانولوشنی، تغییرات شدت روشنایی، تغییرات زاویه دید و ...) در حالیکه برای لایه‌های پایین‌تر هرم مثلا C4 ممکن است بخواهیم تنها از پنج مشخصه‌ی کلیدی برای تشخیص چهره بهره ببریم و هیچکدام از مواردی را که برای لایه‌های C6 یا C7 هرم اعمال کردیم، استفاده ننماییم. بنابراین ما به الگوریتم امکان تغییرات دینامیکی را دادیم. به این طریق هم می‌توان با هزینه‌کَرد از بار محاسباتی شبکه، به دقت بسیار قابل قبولی رسید و هم با استفاده از هزینه‌کَرد از کاهش پیچیدگی باعث

شکل۷ تصاویر حاوی نقاط کلیدی و اضافی از پنج علامت مشخصه که از مجموعه های آموزش WIDER FACE و اعتبار سنجی حاصل شده

سرعت بخشیدن به فرایند تشخیص چهره گردید. عملا یک بِده و بِستانی می‌تواند صورت پذیرد که با افزایش بار محاسباتی شبکه باعث افزایش دقت مدل گردیم و هم می‌توان با کاهش پیچیدگی که نتیجتاً منجر به کاهش بار محاسباتی شبکه می‌شود، به افزایش سرعت مدل در کاربردهای بلادرنگ دنیای واقعی دست یافت. لازم به ذکر است که با توجه به اینکه هر بلوک Inception از شبکه‌ی عصبی می‌تواند دارای برخی ویژگی‌های مشخص از چهره باشد، به جای الگوریتم پیشنهادی در Retina، ما از IOU استفاده می‌نماییم به این صورت که برای هر مستطیل مشخص کننده در هر بلوک Inception با توجه به امکان وجود همپوشانی بین هر کدام از این بلوک‌ها در تشخیص نهایی معیار مشخصی برای پیدا کردن مناسب‌ترین مستطیل مشخص‌کننده‌ی بهترین کاندید اعمال می‌نماییم تا حالت بهینه رخ دهد.

بنابراین به طور خلاصه می‌توان گفت الگوریتم ارائه شده در این پژوهش ابتداً از ۶ لایه‌ی شبکه‌ی هرمی برای استخراج ویژگی بهره می‌برد و در ادامه با استفاده از لایه‌های Inception، طوری به آموزش الگوریتم می‌پردازد که هر کدام از بلوک‌های Inception بتوانند از روش‌های مختلفی در تشخیص چهره استفاده گردند. به این ترتیب پویایی بسیار بالایی به مدل داده می‌شود و شبکه می‌تواند از آن هم در جهت افزایش دقت و هم در جهت افزایش سرعت بهره برد. لازم به ذکر است که برای این کار ما از مجموعه دادگان WiderFace جهت آموزش و تست شبکه استفاده نمودیم، تا هم جامعیت شناسایی قابل قبول باشد و هم دقت بالایی در شناسایی‌های دنیای واقعی بدست آید.



شکل .Error! No text of specified style in document۸ تفاوت بین Swish وSigmoid

الگوریتم موبایل نت معرفی شده در [5] که به کم حجم بودن و قابلیت اجرا بر روی دستگاه‌های با قدرت پردازشی پایین همانند رزبری‌پای و تلفن‌های همراه معروف است، می‌تواند به عنوان بخشی از پژوهش ما مورد استفاده قرار بگیرد. موبایل نت نسخه‌های متفاوتی دارد که جدیدترین آنها نسخه‌ی سوم آن است. در ابتدا به معرفی ویژگی‌های کلیدی آن می‌پردازیم و در نهایت نحوه‌ی استفاده از آن را در مقاله‌مان مورد بررسی قرار می‌دهیم. همانطور که می‌دانید پیشرفت‌های شگرفی در حوزه‌ی هوش مصنوعی صورت گرفته است، امروزه داخل بسیاری از دستگاه‌ها از پردازنده‌های مخصوص به هوش مصنوعی استفاده می‌گردد. نکته ای که باید به آن توجه شود آن است که با رشد فناوری، حجم دادگان موجود و تبادل شده بین دستگاه‌های مختلف رشد شگرفی کرده است و نیاز به سیستم‌ها آنالیز با سرعت بالا و حجم پایین که بتواند در زمان کم با دقت بالا به شناسایی بپردازد، بسیار حس می‌شود. موبایل نت در نسخه‌های مختلف خود سعی در کاهش حجم و افزایش دقت نموده است. در نسخه‌های اول و دوم این معماری از تفکیک‌های عمقی کانولوشنی و بلوک‌های باقیمانده [1]استفاده شده است. در نسخه‌ی سوم این معماری دو مهم انجام شده است: ۱) حذف یا بازنگری استفاده از لایه‌های با بار محاسباتی بالا ۲) استفاده از یک تابع فعالساز غیرخطی به نام Swish که بهبود قابل توجهی نسبت به فعالسازهای مرسومی همچون سیگموید ایجاد نمود.

در اکثر شبکه‌های عصبی بعضی از لایه‌های آخر شبکه و برخی از لایه‌های قبل‌تر نیز بار محاسباتی بالایی نسبت به سایر لایه‌ها دارند. البته باید به این نکته توجه نمود که ممکن است با حذف این لایه‌ها باعث کاهش تاخیر در شبکه شویم ولی در عین حال باید به دقت شبکه در شناسایی نیز آسیبی وارد نگردد. برای اینکه به هر دو مورد توجه شود آنها در برخی از لایه‌های آخر شبکه به جای استفاده از کانولوشن با هسته‌ی ۷*۷ از یک فیلتر با سایز ۱*۱ استفاده کردند. نتیجه‌ی این کار را می‌توان به کاهش بار محاسباتی ویژگی‌ها به نزدیک صفر اشاره نمود. بدین ترتیب علاوه بر کاهش محاسبات به تاخیر بسیار کم دست یافتند. یکی دیگر از لایه‌های با بار محاسباتی بالا، مجموعه اولیه فیلترهاست. مدل‌های مرسوم فعلی تمایل دارند که از ۳۲ فیلتر در یک ترکیب ۳ * ۳ کامل برای ساخت بانکهای فیلتر اولیه برای تشخیص لبه استفاده کنند. اغلب این فیلترها تصاویر آینه ای از یکدیگر هستند و مشابهت زیادی با یکدیگر دارند. با کاهش تعداد فیلترها و استفاده از توابع فعالساز غیرخطی مختلف می‌توانیم بیش از پیش باعث کاهش بار محاسباتی شود.آنها تعداد فیلترها را به ۱۶ فیلتر کاهش دادند در حالی که دقت ۳۲ فیلتر را نیز تقریبا حفظ کردند. با این کار ۲ میلی ثانیه و میلیونها پارامتر صرفه جویی گردیده است. البته این کار با استفاده از جایگزینی تابع فعالساز ReLU با Swish اتفاق افتاده است. در شکل ۸ تفاوت تابع Swish با سیگموید نشان داده شده است. فرمول محاسبه‌ی سوییش به صورت روبروست.

$$Swish\ x = x.\sigma(x)$$ معادله ۲

البته آنچه که در نسخه‌ی سوم موبایل نت استفاده شد، نسخه‌ی بهبود یافته‌ی سوییش بود که با جایگزینی سیگموید با ReLU اتفاق افتاده است و نحوه‌ی محاسبه‌ی آن به صورت زیر است

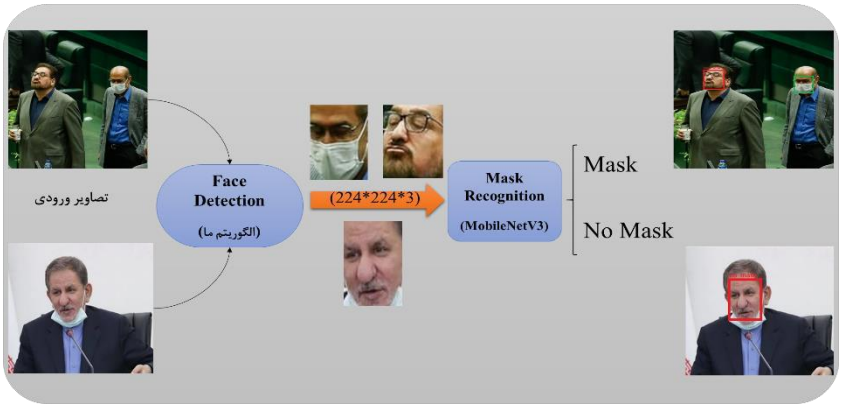
$$h - Swish[x] = x\ \frac{Relu6(x+3)}{6}$$ معادله ۳

1 Residual Blocks

حال با توجه به توضیحات ارائه شده و ترکیب الگوریتم معرفی شده‌ی مدل پیشنهادی ما و نسخه‌ی سوم شبکه موبایل نت فرایندی را جهت شناسایی چهره و همچنین تشخیص وجود ماسک بر روی چهره‌های شناسایی شده ارائه می‌نماییم، تا به مدلی با دقت بالا در عین حجم کم و بار محاسباتی پایین دست پیدا کنیم. به این صورت که در ابتدا تصاویر ورودی به الگوریتم ارائه شده‌ی ما داده می‌شود تا تشخیص چهره بر روی ان انجام پذیرد. در ادامه پس از شناسایی چهره‌های موجود در تصویر، فاز دوم کار یعنی شناسایی چهره‌های همراه با ماسک و بدون ماسک همانند شکل ۹ انجام می‌گردد.

۴- نحوه پیاده سازی

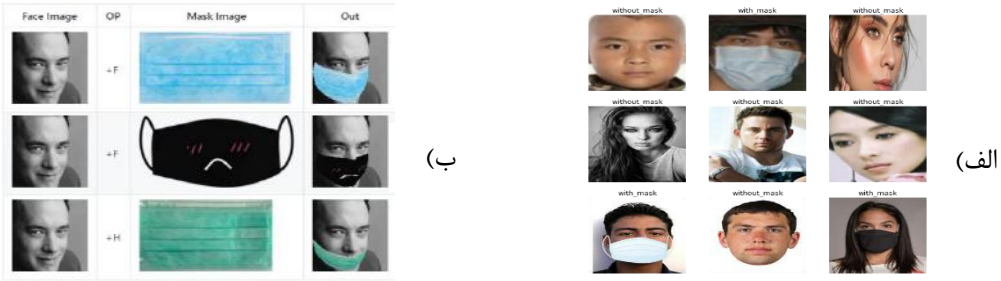
در گام نخست با استفاده از معماری نوین ارائه شده، نواحی از تصویر که دارای چهره می‌باشد تشخیص[1] داده می‌شود. سپس ناحیه تشخیص داده شده متناسب با ورودی معماری موبایل نت، تبدیل شده (۲۲۴*۲۲۴*۳) و به این معماری



شکل۹ فرایند تسخیص چهره قابل استفاده در دوران همه گیری کرونا

که وظیفه‌ی دسته بندی دو کلاسه[2] را دارد، داده می‌شود تا نهایتا تصویر خروجی برچسب‌گذاری شود.

در خصوص آموزش شبکه موبایل نت برای تشخیص با ماسک یا بدون ماسک بودن از مجموعه دادگان متن باز چالش Kaggle [6] استفاده شده است. که دارای ۲,۵ گیگابایت حجم می‌باشد و شامل حدود ۳۰هزار تصویر چهره در دو دسته بدون ماسک و با ماسک می‌باشد. از این مجموعه تصاویر، حدود ۶۰ درصد برای آموزش، ۲۰درصد برای ارزیابی و ۲۰درصد نیز برای تست مورد استفاده قرار گرفته است. شایان توجه است به منظور تعمیم پذیری مدل با انواع گوناگون ماسک در این مجموعه دادگان، در کلاس "با ماسک" هم تصاویر حقیقی چهره با ماسک وجود دارد و هم تصاویری از کلاس "بدون ماسک" که متناسب با landmark چهره، انواع مختلف ماسک به آن‌ها به صورت واقعیت افزوده[3]اضافه گردیده است که در شکل ۱۰ نمایش داده شده است.

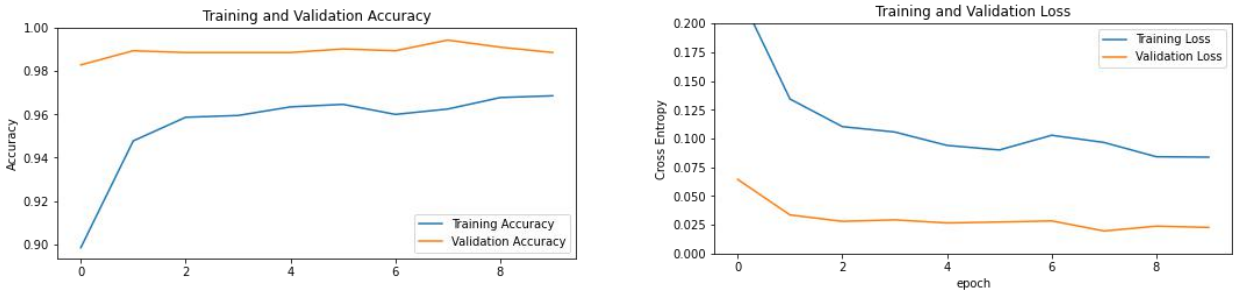


ب) الف)

شکل ۱۰ الف) نمونه هایی از مجموعه دادگان مورد استفاده برای آموزش موبایل نت موجود در سایت Kaggle ب)نحوه اضافه کردن ماسک به مجموعه دادگان

1 Detection Phase
2 Binary Classification
3 Augmented Reality

معماری ارائه شده تنها برای ۱۰ ایپاک آموزش داده شده است و نمودار صحت[1] و مقادیر تابع تلفات بر مبنای binary cross entropy گزارش شده است. شایان ذکر است که به منظور جلوگیری از بیش برازش[2] و تعمیم پذیری مدل بر روی مجموعه دادگان از تکنیک drop out با ضریب ۰,۵ استفاده شده است. پیاده سازی این فرایند با استفاده از زبان برنامه نویسی پایتون، کتابخانه‌ی تنسورفلو نسخه‌ی ۲,۳ و بر روی کارت گرافیک NVIDIA GPU 1080ti به همراه پردازنده اصلی هشت هسته ای Core i7 7700HQ و ۳۲ گیگابایت رم انجام گرفته است.



شکل ۱۱ نمودارهای صحت و تابع تلفات

در جدول ۱ نیز میزان صحت به درصد بیان گردیده است

جدول 1 میزان صحت آموزش، تست، توسعه

	Training	Validation	Test
Accuracy	96.7	99.72	99.54

۵- نتیجه گیری

در این مقاله ما با استفاده از مدل ارائه شده جهت تشخیص چهره و همچنین استفاده از نسخه‌ی سوم الگوریتم موبایل نت به تشخیص وجود ماسک بر روی چهره پرداختیم. در این مسیر از مجموعه دادگان WiderFaces **و مجموعه دادگان** Kaggle **استفاده نمودیم تا به دقتی بسیار قابل قبول در شناسایی چهره‌های با ماسک و بدون ماسک دست یافتیم.**

۶- منابع

[1]Han, J. and Ma, K.K., 2007. Rotation-invariant and scale-invariant Gabor features for texture image retrieval. Image and vision computing, 25(9), pp.1474-1481.
[2] Hu, P. and Ramanan, D., 2017. Finding tiny faces. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 951-959).
[3] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S., 2017. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).
[4] Deng, J., Guo, J., Zhou, Y., Yu, J., Kotsia, I. and Zafeiriou, S., 2019. Retinaface: Single-stage dense face localisation in the wild. arXiv preprint arXiv:1905.00641.
[5] Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V. and Le, Q.V., 2019. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 1314-1324).
[6] https://www.kaggle.com/wobotintelligence/face-mask-detection-dataset

1 Accuracy
2 OverFitting