



دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

دانشکده مهندسی کامپیوتر

پایان‌نامه کارشناسی

سامانه‌ی پیر کردن چهره با مدل‌های مولد

نگارش

امیرحسین رجب‌پور

استاد راهنما

جناب آقای دکتر محمد رحمتی

خرداد ۱۴۰۲

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ



دانشگاه صنعتی امیرکبیر
(پلی‌تکنیک تهران)

به نام خدا

تعهدنامه اصالت اثر

تاریخ: خرداد ۱۴۰۲

اینجانب امیرحسین رجب‌پور متعهد می‌شوم که مطالب مندرج در این پایان‌نامه حاصل کار پژوهشی اینجانب تحت نظرارت و راهنمایی استادی دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مأخذ ذکر گردیده است. این پایان‌نامه قبل‌اً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است. در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان‌نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس از این پایان‌نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مأخذ بلامانع است.

امیرحسین رجب‌پور

امضا

تقدیم به پدر و مادر عزیز و مهربانم که در نختی ها و دشواری های زندگی همواره یاوری
دلوز و فداکار و پشتیبانی محکم و مطمئن برایم بوده اند.

سپاسگزاری

از استاد دلسوز و محترم؛ جناب آقای دکتر محمد رحمتی که با صبر و حوصله، از هیچ کمکی در مسیر انجام این پروژه و نوشتمن این پایان‌نامه از من دریغ ننمودند؛ کمال تشکر و قدردانی را دارم.

امیرحسین رجب‌پور
خرداد ۱۴۰۲

چکیده

بینایی ماشین^۱ زیرمجموعه‌ای از تحقیقات در حوزه‌ی هوش مصنوعی^۲ می‌باشد که بسیار پرکاربرد است و در سال‌های اخیر پیشرفت‌های چشمگیری را در این زمینه شاهد بوده‌ایم. یکی از مواردی که در این زمینه بسیار جای کار و تحقیق دارد بحث تحلیل ویژگی‌های چهره‌ی انسان و ایجاد تغییرات در آن به نحوی مناسب و طبیعی می‌باشد. یکی از این موارد پیر کردن چهره^۳ می‌باشد که کاربردهای زیادی در زمینه‌های سرگرمی، تجاری و یا جرم‌شناسی دارد. محققان پیش‌تر از وجود مدل‌های یادگیری عمیق^۴ و مدل‌های مولد^۵ سعی کرده بودند تا بتوانند از روش‌های دیگری عمل پیر کردن چهره را با ایجاد یکسری ویژگی‌های مشخص شده به صورت دستی و یا کمک گرفتن از تصاویر چهره‌های پیر شده انجام دهند اما با ظهور شبکه‌های عصبی عمیق^۶ و مدل‌های یادگیری عمیق و مدل‌های مولد جایگاه ویژه‌ای در این زمینه تحقیقاتی به وجود آمده است. این مدل‌های جدید دارای قابلیت‌های بیشتر و بهتری می‌باشند و قادر هستند تا عمل پیر کردن چهره را با ظرافت و دقیقت بیشتری انجام دهند. هدف این پروژه بررسی مدل‌های متفاوت برای انجام عمل پیر کردن چهره و سپس استفاده از دو رویکرد برتر آن‌ها و بررسی عمیق‌تر ساختار و معماری آن‌ها می‌باشد. همچنین بخش دیگری از پروژه به پیاده‌سازی یک سامانه‌ی تحت وب با فناوری‌های نوین با بهره‌بردن از معماری مایکروسرویس^۷ می‌باشد که قابلیت این را داشته باشد که چندین مدل سنگین هوش مصنوعی را در خود جای دهد و بتوان به سادگی از آن برای بهره‌بردن از مدل‌های مختلف هوش مصنوعی استفاده کرد.

واژه‌های کلیدی:

پیر کردن چهره، یادگیری عمیق، مدل‌های مولد، شبکه‌های عصبی عمیق، معماری مایکروسرویس

¹Vision

²Artificial Intelligence

³Face Aging

⁴Deep Learning

⁵Generative Models

⁶Deep Neural Networks

⁷Microservice

فهرست مطالب

صفحه

عنوان

۱	مقدمه	۱
۱-۱	اهداف پژوهش	۳
۲-۱	چالش‌های پژوهش	۳
۳-۱	ساختار پایان‌نامه	۵
۲	روش‌های مختلف برای پیر کردن چهره	۶
۱-۲	مقدمه	۷
۲-۲	روش‌های قابل انجام برای پیر کردن چهره	۷
۱-۲-۲	روش‌های سنتی	۷
۲-۲-۲	روش‌های مبتنی بر داده	۸
۱-۲-۲-۲	روش‌های مبتنی بر یادگیری عمیق	۸
۲-۲-۲-۲	روش‌های مبتنی بر مدل‌های مولد	۱۰
۳-۲	تولید شرطی و تولید قابل کنترل	۱۲
۱-۳-۲	تولید قابل کنترل	۱۲
۲-۳-۲	تولید شرطی	۱۳
۴-۲	مدل‌های انتخاب شده‌ی نهایی	۱۳
۱-۴-۲	مدل گلو [۱]	۱۳
۲-۴-۲	مدل سایکل‌گن [۲]	۱۴
۵-۲	خلاصه	۱۴
۳	مدل‌های مولد انتخاب شده	۱۵
۱-۳	مقدمه	۱۶
۲-۳	مدل گلو [۱]	۱۶
۱-۲-۳	مقدمه‌ای بر مدل‌های مبتنی بر جریان	۱۶
۲-۲-۳	معماری مدل گلو	۲۰
۱-۲-۲-۳	لایه‌ی اکترونرم	۲۱
۲-۲-۲-۳	کانولوشن معکوس پذیر ^{۱*}	۲۲
۳-۲-۲-۳	لایه‌ی جفتی خطی	۲۳
۳-۳	مدل سایکل‌گن [۲]	۲۴
۱-۳-۳	مقدمه‌ی ترجمه‌ی تصویر به تصویر	۲۴
۲-۳-۳	مقدمه‌ی مدل سایکل‌گن	۲۵
۳-۳-۳	ساختار کلی مدل سایکل‌گن	۲۵
۱-۳-۳-۳	سازگاری چرخه	۲۶

۲۶	زیان حداقل مربعات	۲-۳-۳-۳
۲۸	زیان هویت	۳-۳-۳-۳
۲۹	معایب سایکل گن	۴-۳-۳
۳۰	خلاصه و جمع‌بندی	۵-۳-۳
۳۱	خلاصه	۴-۳
۳۲	ارزیابی و چالش‌ها	۴
۳۳	مقدمه	۱-۴
۳۳	تخمین سن	۲-۴
۳۳	مقدمه	۱-۲-۴
۳۳	ساختار کلی مدل تخمین سن [۴] [۳]	۲-۲-۴
۳۴	چالش‌های عمل تخمین سن	۳-۲-۴
۳۴	پایگاه‌داده‌های بکار گرفته شده برای عمل تخمین سن	۴-۲-۴
۳۵	مجموعه دادگان [۵] IMDB-WIKI	۱-۴-۲-۴
۳۵	مجموعه دادگان CACD	۲-۴-۲-۴
۳۵	مجموعه دادگان WebFace	۳-۴-۲-۴
۳۵	مجموعه دادگان MORPH	۴-۴-۲-۴
۳۶	مجموعه دادگان UTKFace	۵-۴-۲-۴
۳۶	مجموعه دادگان FGNET	۶-۴-۲-۴
۳۶	مدل استفاده شده برای عمل تخمین سن	۵-۲-۴
۳۷	چالش‌های ارزیابی کمی مدل‌های پیرکننده‌ی چهره	۳-۴
۳۷	استفاده از مجموعه دادگان دارای برچسب سن	۱-۳-۴
۳۸	استفاده از یک نگاشت از مقدار آلفا بر سنین و حالات مختلف	۲-۳-۴
۳۹	استفاده از مدل‌های تخمین سن مختلف	۳-۳-۴
۴۰	استفاده از یک مدل تشخیص واقعی بودن تصویر	۴-۳-۴
۴۰	ارزیابی کیفی مدل‌های پیرکننده‌ی چهره	۴-۴
۴۶	پیشنهادها و کارهای آتی	۱-۴-۴
۴۸	خلاصه	۵-۴
۵۰	پلتفرم ایجاد شده با چارچوب جنگو بر پایه‌ی معماری مایکروسرویس	۵
۵۱	مقدمه	۱-۵
۵۳	ساختار کلی سامانه	۲-۵
۵۳	پیاده‌سازی سامانه به صورت یکپارچه	۱-۲-۵
۵۳	پیاده‌سازی سامانه به صورت معماری مایکروسرویس	۲-۲-۵
۵۴	مشکلات معماری مایکروسرویس	۱-۲-۲-۵

۵۵	۳-۵ جنگو
۵۵	۱-۳-۵ مقدمه
۵۵	۲-۳-۵ ساختار برنامه‌ها در جنگو
۵۶	۳-۳-۵ ساختار فایل‌های اولیه‌ی پروژه جنگو
۵۷	۴-۳-۵ مدل
۵۸	۵-۳-۵ ویو
۵۸	۶-۳-۵ نگاشت رابطه‌ای شیئی
۵۹	۷-۳-۵ تمپلیت
۶۰	۸-۳-۵ فرم
۶۱	۹-۳-۵ پنل ادمین
۶۲	۱۰-۳-۵ جنگو رست
۶۳	۱۱-۳-۵ میان‌افزار
۶۴	۴-۵ تشریح سرویس‌ها
۶۴	۱-۴-۵ داکر
۶۵	۲-۴-۵ پیاده‌سازی سرویس‌ها
۶۶	۱-۲-۴-۵ سرویس سرور اصلی
۶۶	۲-۲-۴-۵ سرویس مدل گلو
۶۷	۳-۲-۴-۵ سرویس مدل سایکل گن
۶۸	۴-۲-۴-۵ سرویس مدل تخمین سن
۶۸	۵-۵ داکر کامپوز
۷۰	۶-۵ رابط کاربری گرافیکی
۷۱	۷-۵ خلاصه
۷۲	۶ جمع‌بندی و نتیجه‌گیری
۷۴	منابع و مراجع

فهرست شکل‌ها

صفحه

شکل

۱-۲	نمونه‌ای از تغییر یک ویژگی تصویر با تولید قابل کنترل	۱۲
۱-۳	سمت چپ: نمونه‌ای از یک جریان در مدل گلو ، سمت راست: معماری چند مقیاسی که در آن عمق جریان مقدار K و تعداد سطوح‌های آن L می‌باشد [۱]	۲۱
۲-۳	نمونه‌ای از ترجمه‌ی تصویر به تصویر جفت و غیر جفت [۲]	۲۵
۳-۳	شمایی از نحوه کارکرد سازگاری چرخه [۲]	۲۶
۴-۳	هزینه‌ی حداقل مربعات محاسبه شده برای سه نقطه‌ی A، B و C	۲۷
۵-۳	تأثیر نگرفتن طرح و استایل تصویر ورودی از گروه اول با اعمال شبکه‌ی مولد تقابلی گروه مخالف	۲۸
۶-۳	عملکرد بد مدل سایکل گن بر روی تصویر آزمون با تغییرات ساختاری زیاد [۲]	۲۹
۷-۳	عملکرد بد مدل سایکل گن بر روی تصویر آزمون ناشناخته [۲]	۲۹
۸-۳	ساختار کلی مدل سایکل گن [۲]	۳۰
۱-۴	شمای کلی آموزش یک مدل تخمین سن [۳] [۴]	۳۴
۲-۴	رونده کلی مدل DEX [۵]	۳۷
۳-۴	تصویر خروجی بهم ریخته مدل گلو از چهره‌ی یک فرد سیاهپوست	۳۹
۴-۴	خروجی مدل گلو با آلفاهای ۱.۲ (بالا سمت چپ) تا ۲ (پایین سمت راست)	۴۱
۵-۴	خروجی مدل گلو با آلفای ۵	۴۱
۶-۴	مقایسه‌ی خروجی مدل سایکل گن با مدل گلو با آلفا ۱.۷	۴۲
۷-۴	مقایسه‌ی دوم خروجی مدل سایکل گن با مدل گلو با آلفا ۱.۷	۴۳
۸-۴	مقایسه‌ی سوم خروجی مدل سایکل گن با مدل گلو با آلفا ۱.۵	۴۴
۹-۴	مقایسه‌ی چهارم خروجی مدل سایکل گن با مدل گلو با آلفا ۱.۹	۴۵
۱۰-۴	مقایسه‌ی پنجم خروجی مدل سایکل گن با مدل گلو با آلفا ۱.۵	۴۶
۱۱-۴	مقایسه‌ی ششم خروجی مدل سایکل گن با مدل گلو با آلفا ۱.۵	۴۷
۱۲-۴	مقایسه‌ی هفتم خروجی مدل سایکل گن با مدل گلو با آلفا ۱.۵	۴۸
۱-۵	صفحه‌ی اول سامانه	۵۱
۲-۵	صفحه‌ی نتایج در حالت پیر شدن چهره	۵۲
۳-۵	صفحه‌ی نتایج در حالت جوان شدن چهره	۵۲
۴-۵	مايكروسرويس‌های تشکيل‌دهنده‌ی سامانه	۵۴
۵-۵	معماری ساختار MVT	۵۵
۶-۵	ساختار فایل‌های اوليه‌ی پروژه جنگو	۵۶
۷-۵	شمایی از نگاشت رابطه‌ای شیئی	۵۹

فهرست شکل‌ها

۸۰	۸-۵ نمونه‌ای از فرم ثبت‌نام
۶۱	۹-۵ نمونه‌ای از مدل‌فرم
۶۴	۱۰-۵ ترتیب قرارگیری میان افزارها میان کاربر و ویو
۶۵	۱۱-۵ تفاوت ساختاری میان داکر و ماشین مجازی
۶۶	۱۲-۵ داکرفایل سرویس اصلی
۶۷	۱۳-۵ داکرفایل سرویس مدل گلو
۶۸	۱۴-۵ داکلرفایل سرویس مدل سایکل گن
۶۹	۱۵-۵ داکلرفایل سرویس مدل تخمین سن

فهرست نمادها

نماد مفهوم

β ضریب جابجایی

γ ضریب مقیاس

فصل اول

مقدمه

پیر کردن چهره یک حوزه تحقیقاتی جذاب در زمینه‌ی بینایی ماشین است که بر روی تبدیل تصاویر چهره به چهره‌ی پیر شده با شبیه‌سازی اثرات پیری بر روی صورت تمکز دارد. این حوزه کاربردهای عملی در زمینه‌های مختلف از جمله سرگرمی، جرم‌شناسی و بازاریابی شخصی دارد. در این پروژه، ما با استفاده از دو مدل مولد قدرتمند گلو^۱ و سایکل گن^۲ عمل پیر کردن چهره را انجام می‌دهیم. علاوه بر این، این پروژه بر روی یک سامانه با معماری مایکروسرویس^۳ پیاده‌سازی شده است که امکان پیاده‌سازی ساده‌تر مدل‌های استفاده شده، مقیاس‌پذیری، مازولاریته بودن و عیب‌یابی بهتر و کارآمدتر را برای ما فراهم می‌کند.

۱-۱ اهداف پروژه

هدف اصلی این پروژه استفاده از مدل‌های گلو و سایکل گن در تولید چهره‌های پیرشده‌ی واقع‌گرایانه در بستر یک سامانه‌ی تحت وب می‌باشد. گلو یک مدل مولد بر اساس معماری‌های مبتنی بر جریان است که چارچوبی انعطاف‌پذیر برای مدل‌سازی توزیع‌های پیچیده داده‌ها ارائه می‌دهد. از طرف دیگر، سایکل گن یک شبکه مولد تقابلی^۴ مبتنی بر رقابت چرخه‌ای است که نتایج قابل قبولی در وظایف ترجمه تصویر به تصویر^۵ ارائه کرده است.

۲-۱ چالش‌های پروژه

این دو مدل، گلو و سایکل گن، به طور خاص مدل‌های مولد شرطی^۶ نیستند. با استفاده از مدل‌های مولد شرطی نیز می‌توان وظیفه پیری چهره را انجام داد؛ در آن روش، مدل بر روی چهره در سن‌های مختلف آموزش داده می‌شود و سن‌های مختلف را به عنوان برچسب در اختیار مدل می‌گذاریم. با این حال، در این پروژه تصمیم گرفته شده از مدل‌های غیر شرطی استفاده شود. این انتخاب مبتنی بر محدودیت‌های خاصی است که در استفاده از مدل‌های مولد شرطی وجود دارد. به عنوان مثال، کاملاً واضح است که این مدل‌ها برای آموزش دیدن نیاز به مجموعه دادگانی دارند که به راحتی قابل دستیابی نیست. در دادگان مورد نیاز، باید چهره‌ی افراد زیادی در سنین مختلف موجود باشد که جمع‌آوری و ایجاد چنین مجموعه دادگانی کاری بسیار دشوار است.

¹Glow

²CycleGAN

³Microservice

⁴Image to image translation

⁵Conditional Generative Models

یکی از دلایل فنی عدم استفاده از یک مدل مولد شرطی در این پروژه می‌تواند عدم کارایی مناسب در حوزه تولید تصاویر باشد. مدل‌های بدون شرط مانند گلو و سایکل گن به دلیل طراحی و ساختار خود، قابلیت تولید تصاویر واقع‌گرایانه و بدون نیاز به برچسب را دارند. همچنین، در برخی موارد، ممکن است برچسب‌های شرط یا داده‌های مربوط به پیری چهره در دسترس نباشند یا کمیت و کیفیت آن‌ها ناکافی باشد. در نتیجه، استفاده از مدل‌های بدون شرط، به دلیل این محدودیت‌ها و پتانسیل بالایی که در تولید تصاویر پیر شامل می‌شود، مناسب و منطقی است.

چالش دیگری که با آن در این پروژه روبرو بودیم پیاده‌سازی این پروژه و مدل‌ها بر روی یک سامانه یکپارچه^۱ بود زیرا مدل‌های مختلف استفاده شده در این پروژه هر کدام دارای نیازمندی‌های متفاوت و دارای مغایرت با یکدیگر بودند که کار پیاً سازی این سامانه را چالش‌برانگیر می‌کرد. در نتیجه پیاده‌سازی این پروژه بر روی یک سامانه با معماری مایکروسرویس انجام شده است. چنین معماری‌ای برای توسعه سیستم‌های پیچیده با اجزای مستقل مناسب است. در این معماری، هر بخش سیستم به عنوان یک مایکروسرویس در نظر گرفته شده تا امکان توسعه به صورت مستقل اجزاء، مقیاس‌پذیری و ادغام آسان فراهم شود. مایکروسرویس‌ها از طریق رابطه‌های برنامه‌نویسی^۲ کارآمدی با یکدیگر ارتباط برقرار می‌کنند که ارتباط و جریان اطلاعات میان آن‌ها را ممکن می‌سازند. دلیل اصلی پیاده‌سازی این پروژه با کمک معماری مایکروسرویس داشتن نیازمندی‌های متفاوت و دارای مغایرت مدل‌های هوش‌مصنوعی‌ای می‌باشد که در این پروژه استفاده شده‌اند. همچنین از دلایل مهم دیگر آن قابلیت آزمون و عیب‌یابی ساده‌تر برنامه با کمک این معماری است.

به صورت خلاصه کارهایی که در این پروژه انجام شده است در ادامه آورده شده است. اولاً، کاربرد دو مدل مولد پیشرفت‌هه گلو و سایکل گن را برای پیری چهره بررسی می‌شود با هدف آن که تصاویری با کیفیت و واقع‌گرا حاصل شود. همچنین با کمک یک مدل رگرسیونی تشخیص سن، سن چهره‌ای که این دو مدل خروجی می‌دهند با چهره‌ی ورودی تخمین زده شده و با یکدیگر مقایسه می‌شوند. علاوه بر این، استفاده از معماری مایکروسرویس انعطاف‌پذیری و قابلیت مقیاس‌پذیری را فراهم می‌کند. طبیعت مژو‌لار این معماری همچنین امکاناتی را برای توسعه و ادغام آینده با سایر اجزا یا مدل‌های دیگر فراهم می‌کند.

¹Monolithic

²API: Application Programming Interface

۳-۱ ساختار پایان‌نامه

ساختار این گزارش بدین صورت می‌باشد که در فصل دو به توضیح راهکارهایی که برای انجام عمل پیر کردن چهره وجود دارد پرداخته شده و چرایی انتخاب دو مدل استفاده شده در این پروژه بیان می‌شود. سپس در فصل سه به توضیح مدل‌های مولد استفاده شده یعنی گلو و سایکل گن می‌پردازیم. در فصل چهار مدل تخمین سن و نکات قابل توجه در این حوزه را شرح می‌دهیم. سپس چالش‌های تحلیل و ارزیابی این نوع مدل‌ها مورد بحث قرار می‌گیرد. در فصل پنجم توضیح کاملی از ساختار سامانه و فناوری‌های متفاوت پیاده‌سازی‌شان و همچنین دلیل استفاده از آن‌ها تشریح می‌شود. در نهایت در فصل آخر نتیجه‌گیری و خلاصه‌ای از این پروژه را ارائه می‌کنیم.

فصل دوم

روش‌های مختلف برای پیر کردن چهره

۱-۲ مقدمه

فرایند پیر کردن چهره که شامل تبدیل تصویر چهره داده شده به ظاهر همان فرد در سنین بالاتر است، توجه قابل ملاحظه‌ای در حوزه بینایی ماشین و پردازش تصویر به دست آورده است. این فرایند توانایی‌های بالقوه‌ی زیادی برای کاربردهای مختلف مانند سرگرمی، تحقیقات کیفری و پژوهشکی شخصی دارد. هدف این فصل بررسی عمیق روش‌های مختلفی است که می‌تواند برای عمل پیر کردن چهره استفاده شوند. این روش‌ها به طور خلاصه به روش‌های سنتی که شامل تعریف قواعدی مشخص برای انجام عمل پیر کردن چهره می‌باشند، روش‌های متکی بر دادگان که خود شامل دو گروه مدل‌های یادگیری عمیق و در نهایت روش‌هایی که بر مبنای مدل‌های مولد تقابلی دسته‌بندی می‌شوند می‌باشند. سپس به معایب و مزایای آن‌ها می‌پردازیم و در نهایت به این نقطه می‌رسیم که چرا مدل‌هایی که در این پروژه برای انجام این عمل خاص یعنی پیر کردن چهره انتخاب شده‌اند، مدل‌های مناسبی هستند.

۲-۱ روش‌های قابل انجام برای پیر کردن چهره

۲-۱-۱ روش‌های سنتی

این روش‌ها عموماً به دو دسته‌ی زیر تقسیم می‌شوند [۷] [۸]:

- روش‌های مبتنی بر قوانین^۱: این روش‌ها از قوانین و قاعده‌های از پیش‌ تعیین شده استفاده می‌کنند تا فرآیند پیر شدن را شبیه‌سازی کنند. به طور معمول، این روش‌ها شامل تعریف دستی تغییرات مربوط به سن در چهره و اعمال آن‌ها به تصویر ورودی می‌شوند.

- تکنیک‌های تغییر شکل تصویر^۲: این روش‌ها شامل تغییر شکل تصویر ورودی با یک تصویر قالبی که سن مطلوب را نشان می‌دهد است. این روش‌ها از تکنیک‌هایی مانند تغییر شکل مبتنی بر شکل و ترکیب بافت برای ایجاد تصویر میانی که به شکل پیر شده به نظر برسد، استفاده می‌کنند.

اگرچه روش‌های سنتی در گذشته به طور گسترده استفاده می‌شدند، اما اغلب محدودیت‌هایی در تولید تصاویر پیر شده واقع گرایانه و طبیعی داشتند. آن‌ها به طور قابل توجهی به تنظیمات دستی و قوانین از پیش تعیین شده وابسته هستند که ممکن است جزئیات و تنوع پیچیده مرتبط با پیر شدن را به خوبی دربرنداشته باشند. به عبارتی، تصاویر تولید شده ممکن است واقع گرایی کمتری داشته باشند و نتوانند فرآیند پیر شدن را به طور دقیق نشان دهند. اما مزیتی که این مدل‌ها دارند این است که به نسبت مدل‌های مبتنی بر یادگیری عمیق و مدل‌های مولد سبک‌تر و سریع‌تر می‌باشند. در نتیجه در موقعي که

¹Rule-based methods

²Image Morphing Techniques

سرعت و سبکی مدل در اولویت توسعه دهنده‌گان باشد، این مدل‌ها می‌توانند عملکرد قابل قبول‌تری را از خود نشان دهند.

اما برای غلبه بر محدودیت‌های این مدل‌ها، پژوهشگران به روش‌های پیشرفته‌تری [۷] [۸] [۹] ، به‌ویژه رویکردهای مبتنی بر یادگیری عمیق و مدل‌های مولد روی آورده‌اند که به طور قابل توجهی در تولید تصاویر پیر شده واقع‌گرا و جذاب‌تر پیشرفته است. این روش‌ها از قدرت شبکه‌های عصبی عمیق استفاده می‌کنند تا الگوها و تبدیلات زیربنایی مرتبط با پیر شدن را از مجموعه داده‌های بزرگ یاد بگیرند و دیگر نیازی ندارند تا توسعه دهنده‌گان به صورت دستی یک سری ویژگی‌ها را طراحی کنند و آن‌ها را در تصاویر استخراج و یا ایجاد کنند.

۲-۲-۲ روش‌های مبتنی بر داده

حال، تمرکز بر روش‌های مبتنی بر داده است [۸] [۹] که در این روش‌ها از الگوریتم‌های یادگیری ماشین برای یادگیری الگوهای پیر شدن از مجموعه داده‌های بزرگ استفاده می‌شود. این بحث شامل کاربرد تکنیک‌های یادگیری عمیق مانند روش‌های مبتنی بر یادگیری عمیق و شبکه‌های مولد در حوزه پیر شدن چهره است. این روش‌ها پیشرفته چشمگیری را در درک و ضبط ویژگی‌های پیچیده چهره و تولید تصاویر بازنمایی پیر شده واقع‌گرا نشان داده‌اند.

۱-۲-۲-۲ روش‌های مبتنی بر یادگیری عمیق

در زمینه پیر شدن چهره، تکنیک‌های یادگیری عمیق به عنوان ابزارهای قدرتمندی در این حوزه به کار گرفته شده‌اند [۸] [۹]. شبکه‌های عصبی پیچشی^۱ نقش بسیار حیاتی را در این زمینه ایفا کرده‌اند. شبکه‌های عصبی پیچشی برای پردازش و استخراج ویژگی‌های معنادار از تصاویر طراحی شده‌اند. آن‌ها شامل چندین لایه از نورون‌های پیوسته هستند که عمل پیچش را انجام می‌دهند، به گونه‌ای که قادر به شناسایی الگوهای محلی می‌باشند و می‌توانند در هر لایه نسبت به لایه‌ی قبل الگوهای پیچیده‌تری را یاد بگیرند. با استفاده از شبکه‌های عصبی پیچشی بر روی تصاویر چهره‌ها، این شبکه‌ها می‌توانند به طور خودکار ویژگی‌های چهره را که با بالا رفتن سن تغییر می‌کنند مانند چروک‌های روی صورت، تغییرات پوست و تغییرات در تراکم چهره را، یاد گرفته و نمایش دهند.

از طرف دیگر، شبکه‌های عصبی بازگشتی^۲ نشان داده‌اند که قابلیت‌های خوبی در ضبط وابستگی‌های زمانی در داده‌های دارای توالی دارند. در مورد پیر شدن چهره، می‌توان از شبکه‌های عصبی بازگشتی برای مدل‌سازی رشد و تغییرات سن در طول زمان استفاده کرد. با در نظر گرفتن توالی تصاویر چهره در سینه مختلف، شبکه‌های عصبی بازگشتی قادر به ضبط تغییرات و تحولات جزئی هستند که در فرآیند پیر شدن افراد در چهره‌ی آن‌ها رخ می‌دهد. درنتیجه این امکان را به ما می‌دهند تا چهره‌های پیر شده‌ای

¹Convolutional Neural Network

²Recurrent Neural Network

توسط مدل تولید شوند که دارای تغییرات واقع‌گرایانه‌ای با در نظر گرفتن توسعه و پیر شدن تدریجی چهره هستند.

اگرچه این مدل‌ها به نسبت قبل دارای نقاط قوت بسیار بهتر و بیشتری هستند اما عاری از مشکل نمی‌باشند. از جمله مشکلات این مدل‌ها می‌توان به موارد زیر اشاره کرد:

- کمبود داده‌های آموزشی به مقدار کافی: مدل‌های یادگیری عمیق به مجموعه داده‌های بزرگی برای آموزش نیاز دارند تا تنوع وسیعی از تغییرات چهره مرتبط با پیر شدن را دربرگیرند و بر روی آن‌ها آموزش ببینند. با این حال، به دست آوردن مجموعه داده‌های باکیفیت و با تنوع بالا با داشتن برچسب سن ممکن است چالش‌هایی داشته باشد و منجر به بایاس و تعمیم محدود شود.
- کمبود قابلیت تفسیر مدل‌ها: مدل‌های یادگیری عمیق، به خصوص آن‌هایی که دارای ساختارهای پیچیده هستند مانند شبکه‌های عصبی عمیق، به سختی قابل تفسیر می‌باشند. این کمبود قابلیت تفسیر باعث می‌شود تحلیل و توضیح ویژگی‌ها و تبدیلات خاصی که مدل در فرآیند پیر شدن یاد می‌گیرد، دشوار باشد و در صورت نیاز به ایجاد تغییرات مشخص در مدل، این کار یا نشدنی یا بسیار دشوار شود.
- بیش‌برازش و تعمیم‌دهی: مدل‌های یادگیری عمیق در معرض بیش‌برازش قرار می‌گیرند که در آن عملکرد خوبی روی داده‌های آموزشی ارائه می‌دهند اما در تعمیم به داده‌های دیده نشده یا کمی متفاوت دچار مشکل می‌شوند. این موضوع ممکن است منجر به پیش‌بینی‌های غیر واقع‌بینانه یا نادقيق از چهره‌های پیر شده در سناریوهای واقعی شود.
- بایاس‌ها و پایین بودن دقت: مدل‌های یادگیری عمیق تحت تأثیر بایاس‌ها و محدودیت‌های موجود در داده‌های آموزش قرار می‌گیرند. این مورد می‌تواند منجر به پیش‌بینی‌های دارای بایاس یا عدم دقت کافی شود، به خصوص زمانی که قصد ثبت الگوهای پیری متنوع در جنسیت‌ها، قومیت‌ها یا افراد با پیش‌زمینه‌های فرهنگی مختلف را داشته باشیم. به عنوان مثال اگر مجموعه دادگان ما تنها دارای چهره‌های مردان باشد، آن‌گاه مدل‌های ما تنها ویژگی‌های چهره‌های مردان را آموزش می‌بینند. در نتیجه هنگامی که در مرحله‌ی آزمون برنامه، مدل چهره‌ی یک زن را به عنوان ورودی بگیرد نمی‌تواند عمل پیر کردن چهره را با توجه به ویژگی‌های زنانه چهره انجام دهد و با شکست مواجه می‌شود.
- نیازهای محاسباتی: مدل‌های یادگیری عمیق، به ویژه آن‌هایی که ساختارهای پیچیده و بزرگی دارند، نیاز به منابع محاسباتی قابل توجهی برای آموزش و استنتاج دارند. این می‌تواند چالش‌هایی در زمینه زمان، حافظه و قدرت محاسباتی ایجاد کند، به ویژه برای پژوهشگران یا افرادی که دسترسی محدودی به زیرساخت محاسباتی با عملکرد بالا دارند.
- حفظ نشدن هویت چهره‌ی ورودی: این مشکل را نمی‌توان به عنوان یک مشکل کلی برای این مدل‌ها ذکر کرد اما در این پروژه یک مشکل بسیار مهم تلقی می‌شود. در عمل پیر کردن چهره،

ما توقع داریم تا برنامه خروجی‌ای که به ما می‌دهد، فارغ از نحوه‌ی پیر شدن و کیفیت آن، چهره‌ی همان شخصی باشد که در ورودی گرفته است. اما در این مدل‌ها احتمال اینکه این اتفاق بیافتد که چهره‌ی خروجی با چهره‌ی ورودی یکی نباشد زیاد است زیرا هیچ راهکاری برای جلوگیری از این اتفاق در این مدل‌ها تعییه نشده است.

حال که با مشکلات این دسته از مدل‌ها آشنا شدیم به سراغ گروه نهایی یعنی مدل‌های مبتنی بر مدل‌های مولد می‌رویم. البته این نکته حائز اهمیت است که روش‌های مبتنی بر مدل‌های مولد قادر نیستند تا تمام مشکلات ذکر شده در دو گروه قبلی را حل کنند و از همه نظر بی‌عیب و نقص باشند. اما توانسته‌اند تا بخشی از مشکلات مهم را حل کنند که در نتیجه به انتخاب نهایی برای این پژوهه تبدیل شده‌اند.

۲-۲-۲ روش‌های مبتنی بر مدل‌های مولد

این گروه خود شامل دو گروه از مدل‌ها می‌باشد که خودکدگذارها^۱ و شبکه‌های مولد تقابلی هستند [۸] [۹] [۶] که به طور گسترده برای عمل پیر کردن چهره استفاده شده‌اند. در این بخش به توضیح مختصری از این دو گروه مدل‌ها می‌پردازیم و سپس در بخش بعد به توضیح دقیق مدل‌های استفاده شده که زیرمجموعه‌ی همین گروه می‌باشند می‌پردازیم.

- روش‌های مبتنی بر خودکدگذارها: به صورت خلاصه، خودکدگذارها دارای معماری‌ای تشکیل شده از شبکه‌های عصبی هستند که شامل یک کدگذار^۲ و یک کدگشا^۳ می‌باشند. روش کار آن‌ها به این صورت است که برای بازسازی داده‌های ورودی، در این حالت تصاویر چهره، آموزش داده می‌شوند و می‌توانند برای هدف پیر کردن چهره تنظیم شوند. با تبدیل یک تصویر از چهره‌ی ورودی به یک بازنمایی پنهان^۴ و ایجاد تغییرات در بازنمایی پنهان و سپس برگرداندن آن به یک تصویر خروجی، خودکدگذارها می‌توانند الگوهای پیر شدن را بر روی تصویر ایجاد کنند و تصاویر پیر شده را تولید کنند.

- روش‌های مبتنی بر شبکه‌های مولد تقابلی: [۶] از سوی دیگر، شبکه‌های مولد تقابلی در گروه مدل‌های مولد می‌باشند که به صورت کلی از دو شبکه عصبی تشکیل شده‌اند: یک شبکه مولد^۵ و یک شبکه‌ی تمايزگر^۶. شبکه‌ی مولد یاد می‌گیرد تا تصاویر تا حد امکان نزدیک به واقعیت چهره‌ی پیر شده را از یکسری نویز تصادفی تولید کند، در حالی که هدف شبکه‌ی تمايزگر

¹Auto-encoder

²Encoder

³Decoder

⁴Latent Representation

⁵Generator

⁶Discriminator

آن است تا بین تصاویر واقعی و تصاویر تولید شده توسط شبکه‌ی مولد تمایز قائل شود. توسط فرآیند آموزش رقابتی^۱، شبکه‌ی مولد توانایی خود را در تولید تصاویر پیر شده قانع‌کننده‌تر و بهتر می‌کند، در حالی که شبکه‌ی تمایزگر در تمیزدھی بین تصاویر واقعی و تولید شده بهتر و موثرتر می‌شود. این آموزش رقابتی باعث می‌شود تا شبکه‌ی مولد تصاویر پیر شده‌ای را تولید کند که به طور زیادی واقع‌گرایانه و قانع‌کننده‌تر می‌باشند و همچنین شبکه‌ی تمایزگر توانایی‌اش در تشخیص تصاویر واقعی و تولید شده زیاد می‌شود. در نتیجه نکته‌ی مهم در آموزش این مدل‌ها این است که این دو شبکه باید با هم پیشرفت کنند و اگر یکی از این دو سرعت پیشرفت‌ش از دیگری بیشتر شود آنگاه در یادگیری مدل خلال وارد می‌شود.

به طور خلاصه هر دو دسته‌ی خودکدارها و شبکه‌های مولد تقابلی [۶] تکنیک‌های قدرتمندی برای پیر شدن چهره ارائه می‌دهند. خودکدارها در سرعت عمل و نیاز به محاسبات کمتر برتری دارند، در حالی که شبکه‌های مولد تقابلی بر تولید تصاویر پیر شده بسیار واقع‌گرایانه و جذاب از نظر بصری تمرکز دارند. نقاط قوت تکمیلی این روش‌ها آن‌ها را به ابزارهای ارزشمندی در زمینه پیر شدن چهره تبدیل می‌کند که به محققان و عملگران ابزارهای موثری برای تولید تصاویر پیر شده واقع‌گرایانه و شخصی شده را ارائه می‌دهند.

در نهایت باید به این نکته اشاره کرد که همانطور که گفته شد مدل‌های مولد قادر نیستند تا تمام ایرادات مدل‌های قبلی را بر طرف کنند. به عنوان مثال موارد زیر از ایرادات این مدل‌ها هستند که همچنان پابرجا می‌باشند:

- تفسیرپذیری پایین: متاسفانه مشکل تفسیرپذیری نیز همچنان وجود دارد و حل نشده است و این مدل‌ها نیز چون عموماً دارای ساختارهای پیچیده و شبکه‌های عصبی می‌باشند به خوبی تفسیرپذیر نیستند.
- نیاز به داده‌ی زیاد: این مدل‌ها نیز همچون قبل نیاز به داده‌ی زیاد برای آموزش دیدن دارند تا بتوانند عملکرد خوبی بر روی داده‌هایی که قبلاً ندیده‌اند و داده‌های آزمون داشته باشند.
- وجود بایاس: این مشکل همچنان در این مدل‌ها نیز پابرجاست و اگر در مجموعه دادگان بایاس وجود داشته باشد مدل‌های آموزش دیده شده توسط این دادگان نیز دارای بایاس خواهند بود.
- نیازهای محاسباتی: در این مورد مدل‌های خودکدار نیاز به محاسبات و سخت‌افزارهای سنگین جهت اجرا ندارند اما این مشکل در شبکه‌های مولد تقابلی نه تنها حل نشده بلکه شاید حتی وحیم‌تر نیز باشد زیرا این شبکه‌ها عموماً خود از چندین شبکه‌ی پیچیده تشکیل شده‌اند که آموزش آن‌ها نیز شرایط خاصی می‌طلبد که در نتیجه این مدل‌ها را از لحاظ نیازهای محاسباتی در شرایط دشواری قرار می‌دهد.

¹Adversarial

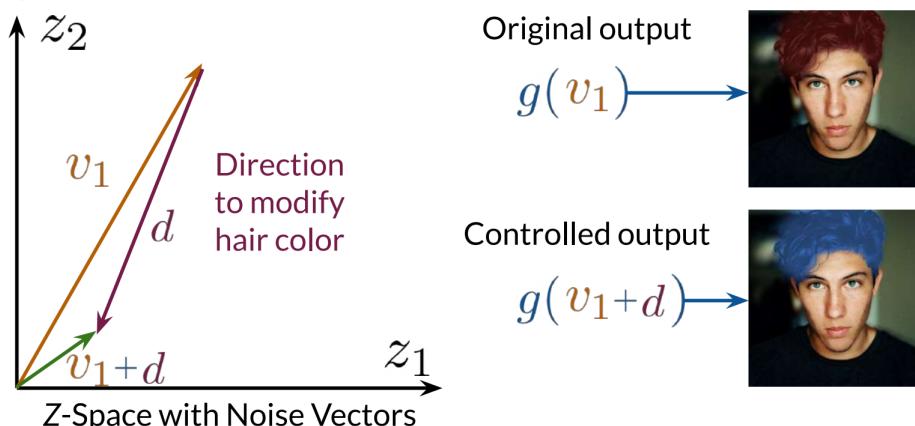
۳-۲ تولید شرطی و تولید قابل کنترل

در برخی از کاربردهای مدل‌های مولد با بحث تولید شرطی^۱ و تولید قابل کنترل^۲ روبرو هستیم. تولید قابل کنترل و شرطی، تکنیک‌هایی هستند که در مدل‌های مولد استفاده می‌شوند تا با ارائه اطلاعات اضافی یا محدودیت‌ها در طول فرایند تولید، خروجی‌های خاص و مطلوب را به دست آورده شوند.

۱-۳-۲ تولید قابل کنترل

تولید قابل کنترل به توانایی تلاش بر روی ویژگی‌ها یا جنبه‌های خاص خروجی تولید شده اشاره دارد. این توانایی به توسعه‌دهندگان امکان کنترل دقیق بر محتوای تولید شده را با تنظیم پارامترها یا ویژگی‌های خاص می‌دهد. به عنوان مثال، در زمینه تولید چهره، تولید قابل کنترل به کاربران امکان می‌دهد سن، جنسیت، حالت صورت یا ویژگی‌های دیگر چهره‌های تولید شده را کنترل کنند. این روش بیشتر در مدل‌های خودکدگذار قابل اعمال است زیرا در این مدل‌ها ورودی به یک فضای پنهان میانی نگاشت می‌شود و سپس بر روی فضای میانی تغییرات دلخواه داده می‌شود و با کمک کدگشا به فضای قبلی بازمی‌گردد. با این توصیفات باید عمل پیر کردن چهره با کمک این روش و مدل‌های خودکدگذار ساده باشد. تنها نیاز است تا فضای میانی را با کمک ویژگی‌های مربوط به پیری چهره، که در مدل به حالت یک برداری از اعداد می‌باشد دستکاری کنیم. مانند تصویر زیر که با کمک برداری که مربوط به رنگ مو می‌باشد، تصویر ورودی دچار تغییر شده و تنها رنگ موی آن تغییر کرده است.

Z-Space and Controllable Generation



شکل ۲-۱ نمونه‌ای از تغییر یک ویژگی تصویر با تولید قابل کنترل

¹Conditional Generation

²Controllable Generation

این روش شاید بسیار ساده به نظر برسد اما به این راحتی قابل انجام نمی‌باشد. زیرا یافتن بردارهایی که تنها یک ویژگی از چهره را بتوانند تغییر بدنهند کار ساده‌ای نیست و بردارها عموماً ترکیبی از چندین ویژگی هستند. در نتیجه تصویر خروجی آن‌ها کیفیت خوبی ندارد و ویژگی مدنظر را به طور دلخواه تغییر نمی‌دهد.

۲-۳-۲ تولید شرطی

تولید شرطی، به عکس، شامل تولید خروجی‌ها بر اساس شرایط یا ورودی‌های خاص برای تولید نمونه در تنها یک کلاس خاص است. این فرایند شامل ارائه اطلاعات اضافی یا زمینه‌ای است که فرایند تولید را به جهت خاصی و مشخصی هدایت می‌کند. به عنوان مثال، در وظیفه تولید چهره، تولید شرطی ممکن است شامل ارائه تصویر هدف یا توصیف متنی به عنوان شرط باشد و مدل تولیدی یک چهره تولید می‌کند که با آن شرط هماهنگ است. به عنوان مثال این شرط می‌تواند سن افراد باشد و خروجی نهایی چهره‌ی افراد در آن سن یا بازه‌ی سنی مشخص باشد. اما مشکلی که این روش دارد این است که نیاز به دادگان برچسب‌گذاری شده‌ای دارد که فراهم کردن آن کار دشواری می‌باشد و نمی‌توان به سادگی مجموعه دادگانی درست کرد که در آن چهره‌ی تعداد زیادی از افراد در سنین مختلف جمع‌آوری شده باشد.

هر دو تکنیک تولید قابل کنترل و شرطی کنترل بیشتر و سفارشی‌سازی بیشتری بر روی خروجی‌های تولیدی ارائه می‌دهند و به کاربران امکان می‌دهند محتوای خاص و مطلوبی تولید کنند که نیازها و ترجیحاتشان را برآورده می‌کند. این تکنیک‌ها کاربردهای گسترده‌ای در حوزه‌های مختلف از جمله ترکیب تصاویر، تولید متن و واقعیت مجازی دارند.

۴-۲ مدل‌های انتخاب شده‌ی نهایی

حال که مزایا و معایب مدل‌های مختلف در بحث پیر کردن چهره را دیدیم در این بخش می‌خواهیم توضیح دهیم که چرا مدل‌هایی که ما برای انجام این پروژه انتخاب کردہ‌ایم مدل‌های مناسبی هستند.

۱-۴-۲ مدل گلو [۱]

مدل اولی که برای عمل پیر کردن چهره انتخاب شده است مدل گلو می‌باشد. در این پروژه ما با استفاده از مدل گلو دقیقاً تولید قابل کنترل را انجام داده‌ایم و تنها ویژگی‌هایی در صورت را که مربوط به بحث پیری چهره می‌باشد تغییر داده‌ایم. این کار به لطف ساختار کمی متفاوت گلو امکان‌پذیر بود که در فصل بعد به طور دقیق به آن می‌پردازیم. به طور خلاصه مدل گلو تصویر ورودی را به یک فضای پنهان بسیار بزرگ نگاشت می‌کند که همین امر باعث می‌شود تا بتواند تصویر بازسازی شده‌ی بسیار با کیفیتی را ارائه کند و ویژگی‌های مدنظر را به خوبی دستکاری کند.

۲-۴-۲ مدل سایکل گن [۲]

مدل سایکل گن یک مدل از شبکه‌های مولد تقابلی می‌باشد که در دل خود دو شبکه‌ی تقابلی مولد را جا داده است و با یک شبکه ورودی را به حالت دلخواه، و با شبکه‌ی دیگر ورودی را از مقصد به حالت اولیه بر می‌گرداند. مزیت بزرگی که این مدل دارد این است که قابلیت حفظ هویت چهره‌ی فرد در تصویر ورودی را دارد و با کمک توابع زیانی که دارد از یکسان بودن هویت چهره اطمینان حاصل می‌کند.

۵-۲ خلاصه

در این فصل به طور خلاصه مروری کردیم بر روشن‌های متفاوتی که موجود بود برای انجام عمل پیر کردن چهره و سپس به مقایسه‌ی این روشن‌ها و مدل‌ها پرداختیم و به مزايا و معایب هر گروه از این مدل‌ها اشاره کردیم. در نهایت با توضیح مدل‌های مولد دیدیم که چرا این مدل‌ها برای انجام عمل پیر کردن چهره مناسب می‌باشند. سپس به معرفی بسیار کوتاهی از دو مدل انتخاب شده برای این عمل یعنی مدل گلو [۱] و سایکل گن [۲] پرداختیم.

فصل سوم

مدل‌های مولد انتخاب شده

۱-۳ مقدمه

در این بخش به توضیح مفصل‌تر مدل‌های گلو [۱] و سایکل گن [۲] و مقاله‌های آن‌ها می‌پردازیم. در بخش قبل مقایسه‌ای بر مدل‌های مولد داشتیم و در این بخش نیز با بررسی مدل‌های مبتنی بر جریان به مقایسه‌ی دوباره‌ی این مدل‌ها می‌پردازیم. به طور خلاصه باید گفت که مدل‌های انتخاب شده‌ی نهایی ما از چند جهت برتری‌هایی بر مدل‌های دیکر برای انجام عمل پیر کردن چهره دارند. به عنوان مثال مدل سایکل گن که یک مدل تشکیل شده از دو شبکه‌ی مولد تقابلی است با داشتن توابع زیان متفاوتی که در خود دارد این اطمینان را حاصل می‌کند که چهره‌ی فرد در تصویر ورودی با تغییر هویت مواجه نشود و تنها طرح و استایل تصویر برای عمل پیر کردن چهره مورد تغییر قرار گیرد. همچنین مدل گلو که یک مدل مبتنی بر جریان است با کمک یکسری از تغییراتی که بر روی توزیع داده ایجاد می‌کند می‌تواند تا یک نگاشت کاملاً برگشت‌پذیر را به وجود آورد و درنتیجه‌ی آن بتوانیم به صورت دقیق و ظرفی تغییراتی بر روی چهره ایجاد کنیم که مرتبط با عمل پیر کردن چهره می‌باشند.

۲-۳ مدل گلو [۱]

۱-۲-۳ مقدمه‌ای بر مدل‌های مبتنی بر جریان

مدل‌های مبتنی بر جریان [۱۰] [۱۱]، نوعی از مدل‌های مولد هستند که یکسری نگاشت‌ها و تبدیلاتی از داده‌ها را یاد می‌گیرند و در نتیجه می‌توانند با کمک این نگاشت‌های یادگرفته شده به تولید داده‌های جدید بپردازنند. به عنوان مثال فرض کنید که یک تابعی وجود دارد که می‌تواند تصویری ساده مانند نویز تصادفی را دریافت کرده و آن را به یک تصویر پیچیده‌تر مانند یک چهره واقع‌گرایانه تبدیل کند و بر عکس. حال این تابع می‌تواند تا به تولید داده‌ها به طریق موثری بپردازد. مدل‌های مبتنی بر جریان به همین روش عمل می‌کنند.

در عوض استفاده از یک تابع پیچیده، مدل‌های مبتنی بر جریان از یک سری عملیات ریاضی برای تبدیل داده‌های ساده به داده‌های پیچیده استفاده می‌کنند. این عملیات‌ها به طور دقیق طراحی شده‌اند تا الگوها و ساختارهای موجود در داده‌ها را در بر بگیرند. مدل با نگاه کردن به مجموعه داده بزرگی از تصاویر نمونه، یاد می‌گیرد که چگونه این تبدیلات را انجام دهد.

نکته‌ی جالب درباره مدل‌های بر پایه جریان این است که آنها نه تنها یاد می‌گیرند چگونه داده‌های جدید تولید کنند، بلکه چگونگی بازگشت و تبدیل داده‌های پیچیده به داده‌های ساده را نیز یاد می‌گیرند. این بدان معنی است که می‌توانید با یک تصویر پیچیده شروع کنید و با "برگشت" عملیات‌ها به تصویر ساده اصلی بازگردید.

ایده نهفته شده در مدل‌های بر پایه جریان این است که با یادگیری چگونگی تبدیل و معکوس‌سازی

داده، می‌توانند به طور موثر توزیع تشکیل‌دهنده‌ی^۱ داده را درک کرده و نمونه‌های جدیدی تولید کنند که شبیه به داده اصلی باشند. آنها با موفقیت در وظایفی مانند تولید تصویر مورد استفاده قرار گرفته‌اند، جایی که می‌توانند تصاویر واقع‌گرایانه‌ای را از نویز ساده ایجاد کنند یا یاد بگیرند چگونه یک تصویر را به تصویر دیگر تبدیل کنند.

در مدل‌های بر پایه جریان، عملیات‌های ریاضی استفاده شده برای تبدیل داده‌ها معمولاً^۲ بر اساس توابع برگشت‌پذیر است. این توابع اطمینان حاصل می‌کنند که تبدیل قابل برگشت باشد و به مدل امکان بازگشت از داده‌های پیچیده به داده‌های ساده را می‌دهد.

یک نوع رایج از توابع برگشت‌پذیر استفاده شده در مدل‌های بر پایه جریان، لایه‌ی جفتی خطی است. در یک لایه‌ی جفتی طی، داده ورودی به دو قسمت تقسیم می‌شود: قسمت "ثابت" و قسمت "قابل یادگیری". قسمت ثابت بدون تغییر باقی می‌ماند، در حالی که به قسمت قابل یادگیری یک تابع تبدیل اعمال می‌شود.

این تبدیل معمولاً یک تابع خطی^۳ است که شامل دو مرحله است: مقیاس‌بندی^۴ و جابجایی^۵. قسمت قابل یادگیری داده ورودی ابتدا توسط یک ضریب مقیاس قابل یادگیری مقیاس‌بندی شده و سپس توسط یک پارامتر جابجایی قابل یادگیری جابجا می‌شود. با تنظیم این پارامترهای مقیاس و جابجایی، مدل می‌تواند یاد بگیرد که داده ورودی را به روش‌های مختلف تبدیل کند.

یک عمل دیگر که به طور معمول در مدل‌های بر پایه جریان استفاده می‌شود، تبدیل عنصر به عنصر^۶ است، به طوری که هر عنصر داده ورودی به طور جداگانه تبدیل می‌شود. این می‌تواند شامل استفاده از توابع غیرخطی مانند تابع سیگموئید^۷ یا تابع نمایی^۸ برای هر عنصر باشد، یا استفاده از تبدیلات پیچیده‌تر مانند تابع چندجمله‌ای^۹.

جنبه کلیدی این عملیات ریاضی این است که طراحی شده‌اند تا برگشت‌پذیر باشند، به این معنی که تبدیل قابل برگشت است. این برای مدل‌های بر پایه جریان بسیار حیاتی است، زیرا به آنها امکان می‌دهد با اعمال عملیات معکوس بر داده ورودی تصادفی، نمونه‌های جدیدی تولید کنند. با ترکیب چندین لایه از این تبدیلات برگشت‌پذیر، مدل‌های بر پایه جریان می‌توانند نگاشت‌های پیچیده‌ای بین داده‌های ساده و پیچیده یاد بگیرند. هر لایه بیشترین پیچیدگی را اضافه می‌کند و وابستگی‌های بالاتر را در داده‌ها کشف می‌کند.

¹Underlying distribution

²Affine

³Scaling

⁴Shifting

⁵Elementwise transformation

⁶Sigmoid function

⁷Exponential functions

⁸Polynomial functions

مدل‌های مولد مبتنی بر جریان دارای ویژگی‌های شاخص زیر می‌باشند [۱] :

- قابل رדיابی بودن مقدار دقیق لگاریتم احتمال^۱: لگاریتم احتمال اندازه‌گیری می‌کند که چه مقدار یک مدل مولد توزیع تشکیل‌دهنده‌ی داده‌ها را به خوبی پوشش می‌هد. در مدل‌های مولد مبتنی بر جریان [۱۰] [۱۱]، محاسبه دقیق لگاریتم احتمال قابل انجام است. این بدان معناست که می‌توانیم به طور مستقیم احتمال وقوع داده‌های مشاهده شده را ارزیابی کنیم که برای وظایفی مانند برآورد چگالی^۲ و آموزش مبتنی بر احتمال^۳ مفید است. قابلیت محاسبه دقیق لگاریتم احتمال امکان مدل‌سازی دقیق‌تر و ارزیابی بهتر نمونه‌های تولیدی را فراهم می‌کند.
- قابل رדיابی بودن مقدار دقیق متغیرهای پنهان^۴: در مدل‌های مولد مبتنی بر جریان [۱۰] [۱۱]، متغیرهای نهان نماینده بازنمایی‌های میانی^۵ یا عوامل مخفی^۶ هستند که ساختار زیرین و بنیادین داده را در بر می‌گیرند. قابلیت انجام استنتاج دقیق بر روی این متغیرهای نهان بسیار مطلوب است. این به معنای این است که با داشتن یک نقطه داده مشاهده شده، می‌توانیم به طور تحلیلی مقادیر نهان مربوطه را محاسبه کنیم بدون نیاز به تقریب یا تکنیک‌های نمونه‌برداری. استنتاج دقیق متغیرهای نهان امکان مدل‌سازی با کنترل و تفسیرپذیری بیشتر را فراهم می‌آورد. همچنین دارای قابلیت بازسازی بهینه‌تر داده‌ها می‌باشد.
- قابلیت موازی‌سازی هم برای مرحله‌ی آموزش^۷ هم برای مرحله‌ی تولید^۸: مدل‌های تولیدی مبتنی بر جریان [۱۰] [۱۱] از مزیت همزمانی در مراحل آموزش و تولید برخوردار هستند. در مرحله آموزش، می‌توان مدل را بر روی مجموعه دادگان بزرگ با پردازش همزمان چندین نقطه داده^۹، آموزش داد که باعث افزایش قابل توجه سرعت فرآیند آموزش می‌شود. این همزمانی به این دلیل است که تبدیلات در مدل‌های مولد مبتنی بر جریان [۱۰] [۱۱] می‌توانند به صورت مستقل بر روی نقاط داده مختلف اعمال شوند. به طور مشابه، در مرحله تولید نمونه‌های جدید نیز می‌توان به صورت همزمان عمل کرد و این امکان را فراهم کرد که نمونه‌های چندگانه به صورت کارآمد و سریع تولید شوند.

دو مشکل حل نشده‌ای که در حوزه‌ی یادگیری ماشین وجود دارند عبارت‌اند از [۱] :

¹log-likelihood

²Density estimation

³Likelihood-based training

⁴Latent variable

⁵Intermediate representations

⁶Hidden factors

⁷Training stage

⁸Synthesis stage

⁹Datapoint

- بهینه بودن داده^۱: توانایی یادگیری از داده‌های کم مانند انسان‌ها.
- تعمیم پذیری^۲: مقاوم بودن نسبت به تغییرات در آن وظیفه‌ی محول شده و یا زمینه یا محتوا.

برای حل این دو مشکل دو راهکار پیشنهاد شده‌است:

- یادگرفتن مدل‌های واقعی بر مبنای جهان. بدین صورت عامل هوش مصنوعی قبل از تعامل در دنیای واقعی در آن آموزش داده شده‌است.
- یادگیری ویژگی‌های با معنای داده‌های ورودی بدون نیاز به نظارت انسان و یا برچسب‌گذاری. این ویژگی‌ها از مجموعه دادگان بزرگ بدون برچسب آموخته می‌شوند و مختص یک کار خاص نمی‌باشند در نتیجه برای وظایف پایانی^۳ مقاوم‌تر و بهینه‌تر می‌باشند.

حال اگر بخواهیم تا تقسیم‌بندی دقیق‌تر و درست‌تری از مدل‌های مولد داشته باشیم تا بتوانیم این مدل‌ها را با یکدیگر درست‌تر مقایسه کنیم این تقسیم‌بندی را باید بدین صورت انجام دهیم [۱].

- مبتنی بر شبکه‌های مولد تقابلی^۴

- مبتنی بر لگاریتم احتمال^۵

□ مدل‌های اتورگرسیو [۱۲] [۱۳] [۱۴]^۶: از مزایای آن‌ها می‌توان گفت ساده هستند اما برای سنتز یا ترکیب در موازی‌سازی محدود می‌باشند زیرا طول محاسبات سنتز مناسب است با ابعاد داده که در نتیجه در این امر در عکس‌ها و ویدئوهای با حجم بالا مشکل‌ساز می‌شود.

□ خودکدگذارهای تغییراتی^۷ [۱۵] [۱۶]: قابل موازی سازی بودن هم برای آموزش داده شدن هم برای تولید کردن اما بهینه کردن آن دشوار می‌باشد. این مدل یک کران پایین برای لگاریتم احتمال داده بهینه می‌کند. یعنی یک بازنمایی فشرده‌شده از داده‌های ورودی یاد می‌گیرند که احتمال تولید کردن نقاط مشابه را بیشینه می‌کند. در نتیجه با بهینه کردن این کران پایین این مدل‌ها سعی می‌کنند تا به صورت کارآمدی توزیع داده‌ها را پوشش دهند تا بتوانند به تولید مناسب داده‌های جدید بپردازنند.

□ مدل‌های مولد مبتنی بر جریان [۱] [۱۰]^۸

¹Data efficiency

²Generalization

³Downstream

⁴GAN based

⁵likelihood based

⁶Autoregressive

⁷VAE: Variational Auto-Encoders

⁸Flow based generative

حال با توجه به این دسته‌بندی می‌توانیم به نقاط قوت مدل‌های مبتنی بر جریان نسبت به سایر مدل‌ها اشاره کنیم [۱]:

- استنتاج دقیق متغیر پنهان و ارزیابی لگاریتم احتمال: در خودکدگذارهای تعمیراتی [۱۵] [۱۶] تنها می‌توان مقدار متغیر پنهان را تخمین زد و دقیق نیست. در گن [۶] اصلاح کدگذاری نداریم تا متغیر پنهان را استنتاج کنیم. اما در مدل‌های مولد مبتنی بر جریان [۱۰] [۱۱] این کار به صورت دقیق قابل انجام است که همین امر باعث می‌شود تا بتوان برای لگاریتم احتمال داده‌ها بهینه‌سازی دقیق انجام داد.
- استنتاج و سنتر بهینه: مدل‌های اتورگرسیو [۱۲] [۱۳] [۱۴] نیز بازگشت‌پذیر هستند اما به سختی می‌توان آن‌ها را موازی‌سازی نمود و در سخت‌افزار قابل موازی‌سازی بهینه نیستند. اما مدل‌های مولد مبتنی بر جریان [۱۰] [۱۱] برای موازی‌سازی بسیار بهینه می‌باشند.
- استفاده از فضای پنهان^۱ برای وظایف پایانی: در لایه‌های پنهان مدل‌های اتورگرسیو [۱۲] [۱۳] [۱۴] توزیع‌های حاشیه‌ای نامشخصی^۲ وجود دارند که عمل دستکاری کردن^۳ داده را دشوار می‌کند. در گن [۶] داده‌ها در یک فضای پنهان مدل نمی‌شوند زیرا کدگذاری وجود ندارد اما در خودکدگذارهای تعمیراتی [۱۵] [۱۶] و مدل‌های مولد مبتنی بر جریان [۱۰] [۱۱] بدین صورت نمی‌باشد و آن‌ها قابلیت درون‌یابی^۴ کردن میان نقاط داده و قابلیت تغییر با معنای داده را دارند.
- پتانسیل بالا جهت صرفه‌جویی در حافظه: محاسبه گرادیان در شبکه‌های عصبی برگشت‌پذیر نیاز به یک مقدار ثابت از حافظه دارد و با عمق خطی نمی‌باشد.

۲-۲-۳ معماری مدل گلو

این مدل از یک جریان بر پایه‌ی دو جریانی که در دو مقاله‌ی NICE [۱۰] و RealNVP [۱۱] پیشنهاد شده استفاده می‌کند. این معماری از سری‌هایی از جریان‌ها در یک معماری چند مقیاسی^۵ تشکیل شده‌است که به صورت شکل زیر می‌باشد:

هر مرحله از جریان از ۳ عنصر لایه‌ی اکتنرم^۶ و سپس بعد از آن یک کانولوشن معکوس‌پذیر^۷ و سپس یک لایه‌ی جفتی خطی^۸ تشکیل شده‌است که در ادامه به توضیح هر یک می‌پردازیم. همچنین

¹Latent space

²Unknown marginal distributions

³Manipulate

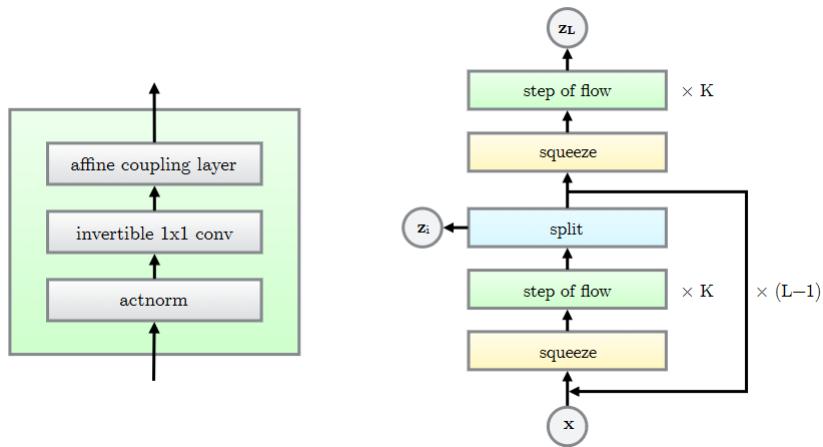
⁴interpolate

⁵Multi-scale architecture

⁶Actnorm

⁷invertible 1*1 convolution

⁸Affine coupling layer



شکل ۱-۳ سمت چپ: نمونه‌ای از یک جریان در مدل گلو، سمت راست: معماری چند مقیاسی که در آن عمق جریان مقدار K و تعداد سطح‌های آن L می‌باشد [۱]

این مدل چند مقیاسی می‌باشد بدلیل محدودیت فضا.

۱-۲-۲-۳ لایه‌ی اکتنرم

قبل از توضیح این عنصر نیاز است تا یادآوری‌ای داشته باشیم بر مفهوم نرمال‌سازی بج بج^۱. نرمال‌سازی بج بج وظیفه‌ی نورمال کردن خروجی تابع فعالیت در لایه‌های شبکه‌ی عصبی را دارد که فرمول آن در زیر آمده است. در این فرمول x ورودی تابع فعالیت و x' خروجی نرمال‌سازی شده و μ میانگین یک بج و σ^2 واریانس آن بج و γ ضریب مقیاس و β ضریب جابجایی و در نهایت ϵ یک مقدار ثابت برای پایداری محاسبات می‌باشند.

$$x' = \gamma \left(\frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \right) + \beta \quad (1-3)$$

این عملیات اطمینان حاصل می‌کند که وزن‌های نورون‌های شبکه عصبی مقدار زیاد و یا کم به صورت غیر بالанс نمی‌گیرند تا شبکه را بهم بربند. همچنین این عملیات می‌تواند سرعت آموزش مدل را نیز بالا ببرد زیرا تابع زیان را متوازن^۲ و هموار^۳ می‌کند در نتیجه پیمایش تابع و رسیدن به مینیمم آن آسان‌تر می‌شود. از تاثیرات مثبت دیگر این عملیات، کم کردن اهمیت وزن‌های اولیه‌ی شبکه است. حال این عملیات در هر بج^۴ یا دوره انجام می‌شود که مقدار آن را هنگام ایجاد مدل تعیین می‌کنیم.

¹Batch normalization

²Symmetric

³Loss smoothing

⁴batch

اما مشکلی که نرمال‌سازی بچ‌بچ دارد این است که واریانسی که اضافه می‌شود به نویز فعالیت‌ها^۱ به صورت معکوس تناسب دارد با سایز مینی‌بچ^۲ ها در هر واحد پردازشی^۳. به عنوان مثال برای عکس‌های با حجم بالا سایز مینی‌بچ^۱ قرار داده می‌شود برای هر واحد پردازشی به دلیل محدودیت‌های حافظه‌ای.

حال در این مقاله [۱] یک روش دیگر شبیه نرمال‌سازی بچ‌بچ پیشنهاد می‌شود و آن استفاده از یک لایه‌ی لایه‌ی اکتنرم به معنای activation normalization یا نرمال‌سازی فعالیت می‌باشد. لایه‌ی اکتنرم یک نوع لایه‌ی نرمال‌سازی است که به عنوان مرحله‌ای از پیش‌پردازش داده در کانولوشن معکوس‌پذیر^۴ عمل می‌کند.

نقش لایه‌ی اکتنرم در نرمال‌سازی فعالیت‌های داده برای داشتن میانگین صفر و واریانس یک در هر کanal است. این نرمال‌سازی بر روی هر ورودی کانولوشن معکوس‌پذیر^۱ قبل از انجام عمل کانولوشن اعمال می‌شود. با نرمال‌سازی فعالیت‌ها، لایه‌ی اکتنرم کمک به پایدارسازی فرآیند آموزش می‌کند و یادگیری بیشتری را ممکن می‌سازد.

لایه‌ی اکتنرم برای رفع چالش‌های مربوط به مقداردهی اولیه و بهینه‌سازی که در مدل‌های مولد عمیق معمولاً با آن روبرو می‌شویم، طراحی شده است. این لایه اطمینان حاصل می‌کند که فعالیت‌ها به اطراف صفر متتمرکز شده و دارای گستره‌ی معقولی باشند که جلوی انهدام زودهنگام مدل را می‌گیرد. نرمال‌سازی همچنین به کاهش مشکلات گرادیان ناپدید شونده یا بزرگ شونده^۴^۵ کمک می‌کند و پایداری کلی فرآیند یادگیری را بهبود می‌بخشد.

علاوه بر این، لایه‌ی اکتنرم به عنوان یک شکل از نظم‌بخشی عمل می‌کند و با کاهش تکرار^۶ و همبستگی بین ویژگی‌ها، تنوع^۷ را ترویج می‌دهد و مدل را تشویق می‌کند تا بازنمایی‌های متنوع‌تری از داده را یاد بگیرد.

۲-۲-۲-۳ کانولوشن معکوس‌پذیر^۱

این کانولوشن به نوع خاصی از عملیات کانولوشنی اشاره دارد که بر روی داده اعمال می‌شود. در این نوع کانولوشن، از یک کرنل با سایز 1×1 استفاده می‌شود و همچنین اطمینان حاصل می‌شود که کانولوشن به طور دقیق قابل برگشت است.

هدف از این کانولوشن تبدیل داده ورودی با حفظ محتوای اطلاعاتی آن است. بر خلاف کانولوشن‌های سنتی که به طور معمول شامل کرنل‌های بزرگتر و فیلترهای قابل یادگیری هستند، کانولوشن 1×1 بر روی

¹activations

²mini batch

³processing unit

⁴Vanishing gradient

⁵Exploding gradient

⁶Redundancy

⁷Diversity

هر کanal به صورت جداگانه اعمال شده و تبدیلات خطی روی کanal‌های ورودی انجام می‌دهد. خاصیت قابل برگشت بودن بسیار مهم است زیرا این امکان را به مدل می‌دهد تا هم عملیات تبدیل را به صورت جلورونده^۱ (کانولوشن) اعمال کند و هم آن را برعکس کند (تبدیل معکوس) بدون از دست دادن اطلاعاتی. این خاصیت به مدل امکان می‌دهد تا تبدیلات قابل معنا و قابل برگشتی بین فضای ورودی و فضای خروجی یاد بگیرد.

با استفاده از کانولوشن‌های قابل برگشت ^۲ نویسنده‌گان این مقاله قصد آن را دارند تا یادگیری تبدیلات پیچیده و بیان کننده^۳ را تسهیل کنند در حالی که اطمینان حاصل می‌کنند که مدل قابل برگشت است و می‌تواند ورودی اصلی را بازسازی کند. این امر امکان آموزش و استنباط کارآمد در مدل‌های مولد را فراهم می‌کند، همچنین قابلیت انجام وظایفی مانند تولید تصویر، حذف نویز یا بازسازی تصویر را فراهم می‌سازد.

۳-۲-۲-۳ لایه‌ی جفتی خطی

لایه‌ی جفتی خطی^۴ به یک نوع خاص از لایه‌های استفاده شده در مدل مولد مبتنی بر جریان گفته می‌شود. این لایه‌ها نقش حیاتی در تحول داده و قابلیت تولید نمونه‌های متنوع و با کیفیت بالا توسط مدل دارند. این لایه به گونه‌ای طراحی شده است تا ورودی را به دو بخش تقسیم کند: بخش ثابت و بخش متغیر. بخش ثابت در حین اعمال تغییرات^۵ بدون تغییر باقی می‌ماند، در حالی که بخش متغیر به شکلی انعطاف‌پذیر بر اساس بخش ثابت تغییر می‌کند. در نتیجه استفاده از این لایه‌ها این امکان را می‌دهد تا توزیع‌های پیچیده داده را مدل کرده و در عین حال قابلیت تسهیل و معکوس‌پذیری را حفظ کند.

اعمال توابع تغییرات بر بخش متغیر در این لایه به طور معمول از طریق یک تبدیل خطی^۶ انجام می‌شود که شامل یک عامل مقیاس^۷ و یک عامل جابجایی^۸ است. این عوامل پارامترهای یادگیری شده هستند که در طول فرآیند آموزش به طور تطبیقی تنظیم می‌شوند تا عملکرد مدل بهینه شود. عامل مقیاس، کنترل میزان کشیدگی یا کوچک‌شدن بخش متغیر را بر عهده دارد.

مهم‌ترین نکته آن است که این لایه‌ها به گونه‌ای طراحی شده اند که قابلیت معکوس‌پذیری دارند، به این معنی که تبدیلی که به بخش متغیر اعمال می‌شود، قابل بازسازی است تا ورودی اصلی را بازسازی کند. این خاصیت برای مدل‌های مولد بسیار حیاتی است، زیرا امکان نمونه‌برداری و استنتاج به صورت کارآمد را فراهم می‌کند. با استفاده از این لایه‌ها در مدل مولد مبتنی بر جریان یک تبدیل غنی و قابل

¹Forward

²Expressive

³Affine Coupling Layer

⁴Transformations

⁵affine transformation

⁶Scale factor

⁷Shift factor

تنظیم از داده ورودی یاد گرفته می‌شود و این امکان را به مدل می‌دهد تا با تغییر بخش‌های متغیر و همزمان حفظ بخش‌های ثابت، وابستگی‌های پیچیده را درک کند و نمونه‌های متنوعی تولید کند.

۳-۳ مدل سایکل گن [۲]

۱-۳-۳ مقدمه‌ی ترجمه‌ی تصویر به تصویر

در مسائل ترجمه‌ی تصویر به تصویر^۱ که از جمله مسائل دسته‌ی بینایی ماشین و گرافیک می‌باشد، هدف یادگرفتن یک نگاشت^۲ از یک تصویر به تصویری دیگر با کمک داده‌های موجود می‌باشد. این کار با توجه به داده‌هایی که در اختیار داریم می‌تواند به دو صورت جفتی^۳ یا غیرجفتی^۴ انجام پذیرد.

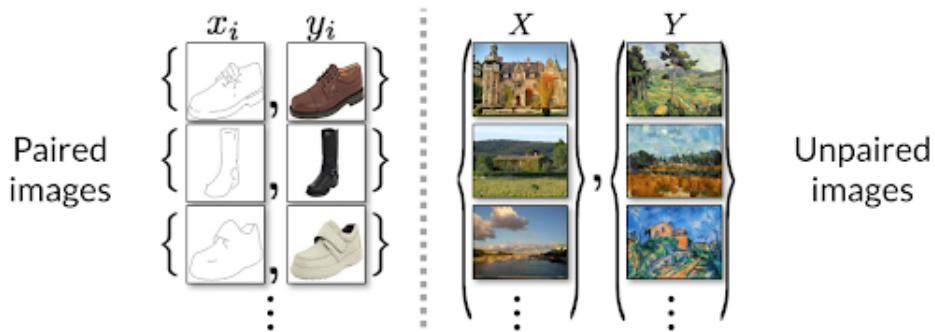
مدل سایکل گن یک مدل غیرجفتی می‌باشد بدین معنا که یک نگاشت بین دو دسته از عکس‌ها می‌باشد که در آن محتوای غالب در هر گروه از عکس‌ها و همچنین اختلافات میان این گروه‌ها پیدا شده و مورد استفاده قرار می‌گیرند. حالت جفتی برای زمانی مناسب می‌باشد که ما داده‌هایی جفت‌جفت داشته باشیم اما در حالت غیرجفتی دیگر مسئله به این سرراستی نمی‌باشد و مشکل می‌باشد زیرا دیگر دیتا‌یی وجود ندارد که به صورت دقیق جفت عکسی که ما داریم باشد و در اینجا یک سری گروه از عکس‌ها داریم که نیاز است به صورت دقیق تر مورد بررسی قرار گیرند و ساختار آن‌ها توسط مدل یادگرفته شود. در واقع در سایکل گن نظارت ما در سطح گروه‌ها می‌باشد نه در سطح جفت عکس‌های مجموعه دادگان که همین مورد مزیت این مدل محسوب می‌شود. هر گروه دارای استایل خاص و محتوا می‌باشد و هنگام جفت کردن تصاویر، محتوای یک تصویر با استایل تصویر دیگر ترکیب می‌شود. در این کار، اشتراکات و تفاوت‌هایی از منظر استایل و سبک وجود دارد که هدف انتقال دادن بخش یکتای هر گروه که همان سبک و استایل آن است به محتوای غالب در گروه دیگر یا عکسی از گروه دیگر که همان محتوا و ساختار اصلی آن است می‌باشد. به عنوان مثال یک دسته از نقاشی‌های ون‌گوگ داریم و یک عکس از یک ساختمان. حال می‌خواهیم که عکس آن ساختمان به سبک نقاشی ون‌گوگ باشد. در اینجا، سبک و استایل از دسته‌ی نقاشی‌های ون‌گوگ به محتوای اصلی عکس که منظره‌ای از یک ساختمان می‌باشد انتقال داده می‌شود و خروجی نهایی نقاشی‌ای به سبک ون‌گوگ از آن ساختمان می‌شود.

¹Image to image translation

²Mapping

³Paired image to image translation

⁴Unpaired image to image translation



شکل ۲-۳ نمونه‌ای از ترجمه‌ی تصویر به تصویر جفت و غیر جفت [۲]

۲-۳-۳ مقدمه‌ی مدل سایکل گن

این مدل یک مدل غیرجفتی که در بخش قبل توضیح داده شد می‌باشد و از دو مدل مولد که با یکدیگر ترکیب شده‌اند تشکیل شده‌است. طرز کار این مدل بدین صورت می‌باشد که تصویر از یک گروه با سبک گروه دیگر ترکیب می‌شود. سپس برای بالا بردن کیفیت تصاویر ایجاد شده، تصویر خروجی با سبک تصویر اصلی‌اش ترکیب می‌شود. خروجی نهایی باید دقیقاً تصویر اولیه باشد که در این حالت می‌توان از یکتابع زیان پیکسل به پیکسل استفاده کرد. این عامل که یکی از عناصر اصلی این مدل می‌باشد سازگاری چرخه^۱ نام دارد. حال برای ساخت این چرخه باید از دو مدل شبکه‌ی مولد تقابلی استفاده کرد. یکی که تصویر اولیه را با سبک و استایل گروه دوم ترکیب می‌کند. و یکی دیگر که خروجی این ترکیب را به سبک اولیه بازمی‌گرداند. درنتیجه هرچه مقدار این تابع زیان کمتر باشد (در واقع هرچه سازگاری چرخه بیشتر باشد) بدین معنا است که دو مدل مولد ما توانایی تولید خروجی‌های واقعی‌تر و طبیعی‌تر را دارند. همچنین این عامل باعث می‌شود تا مدل بتواند صرفاً سبک و استایل را تغییر دهد و به محتوای اصلی تصویر دست نزد.

۳-۳-۳ ساختار کلی مدل سایکل گن

این مدل همانطور به پیش‌تر اشاره شد دارای دو شبکه‌ی مولد تقابلی می‌باشد که در نتیجه دارای چهار عنصر اصلی می‌باشد که دوتای آن مولد و دوتای آن تمایزگر می‌باشند. یکی از این شبکه‌های مولد تقابلی مسئول انتقال تصویر از گروه ۱ به ۲ و دیگری مسئول انتقال تصویر از گروه ۲ به ۱ می‌باشد. از آنجایی که تمایزگر این دو مدل PathGAN می‌باشد، خروجی آن به صورت یک ماتریس دسته‌بندی^۲ می‌باشد.

¹Cycle consistency

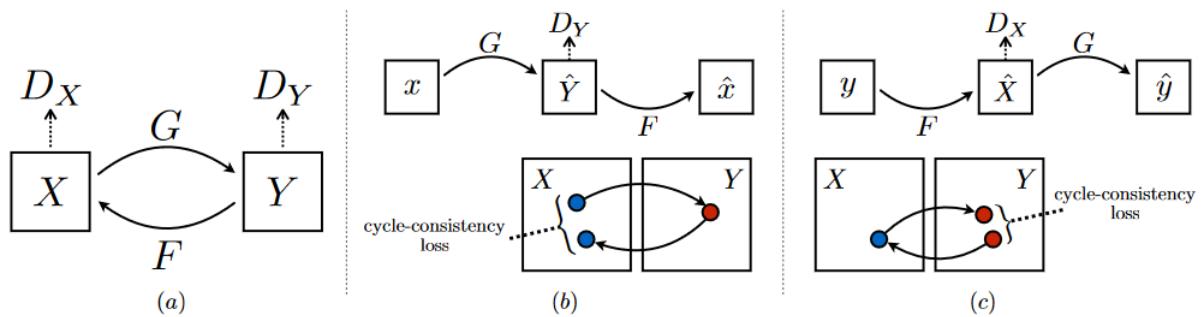
²Classification matrix

۱-۳-۳-۳ سازگاری چرخه

برای بالا بردن سازگاری چرخه در مدل، نیاز به دو تابع زیان داریم. هر مدل مولد باید یک تابع زیان برای بالا بردن سازگاری چرخه داشته باشد. در نهایت مجموع مقادیر این دو زیان را به عنوان زیان سازگاری چرخه‌ی نهایی در نظر گرفته می‌شود.

حال این مقدار نهایی زیان سازگاری چرخه^۱ با زیان اصلی مدل^۲ (که در زیر بخش بعدی به توضیح آن می‌پردازیم) جمع شده و مقدار نهایی را برای بهینه کردن در نظر می‌گیریم. این نکته نیز باید اشاره شود که یک ضریب λ برای زیان سازگاری چرخه‌ی نهایی نیز در نظر می‌گیریم. پس تا به این جای کار برای زیان مدل داریم:

$$OverallLoss = AdversarialLoss + \lambda * CycleConsistencyLoss \quad (2-3)$$



شکل ۳-۳ شمایی از نحوه کارکرد سازگاری چرخه [۲]

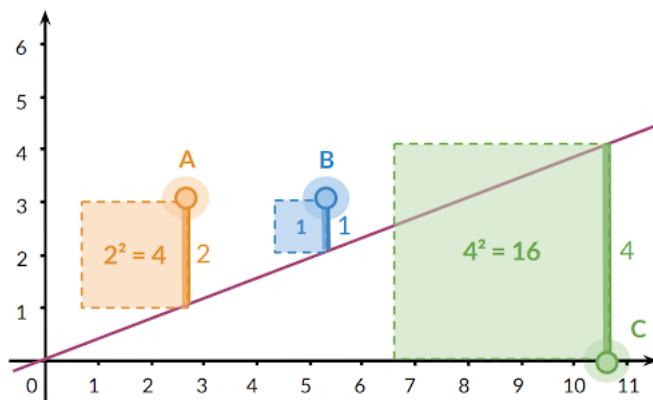
۲-۳-۳-۳ زیان حداقل مربعات

زیان حداقل مربعات^۳، زیان اصلی‌ای است که در مدل سایکل‌گن مورد استفاده قرار می‌گیرد. این زیان به طور خلاصه روشی می‌باشد که در آن خطی پیدا می‌کنیم که مجموع مربع‌های فواصل نقاط تا آن خط کمترین مقدار باشد. به عنوان مثال در شکل زیر می‌توان دید که خطی از هر نقطه تا آن خط کشیده شده. سپس مربعی که ضلع آن فاصله‌ی آن نقطه تا خط است رسم شده. حال این روش سعی دارد تا مجموع مساحت‌های این مربع‌ها را با جابه‌جا کردن خط به کمترین مقدار خود برساند. این مقدار همان هزینه یا زیان حداقل مربعات نام دارد.

¹Cycle Consistency Loss

²Adversarial loss

³Least squares error



شکل ۴-۳ هزینه‌ی حداقل مربعات محاسبه شده برای سه نقطه‌ی A، B و C

حال در تمايزگر تصویر واقعی دارای بروجسب ۱ و غیرواقعی دارای بروجسب ${}^{\circ}$ می‌باشد. در نتیجه مقدار حداقل مربعات برای آن به صورت زیر می‌شود که در آن مربع فواصل تصاویر ورودی با بروجسب ۱ که واقعی هستند تا خط نماینده‌ی بروجسب ۱ و تصاویر غیرواقعی که دارای بروجسب ${}^{\circ}$ می‌باشند تا خطی که نماینده‌ی بروجسب ${}^{\circ}$ می‌باشد با یکدیگر جمع شده. فرمول زیان حداقل مربعات تشخیص‌دهنده در ادامه آمده است که در آن $D(x)$ نشانگر مقدار پیش‌بینی شده‌ی دسته‌بندی کننده‌ی تمايزگر از تصویر واقعی x می‌باشد که با بروجسب ۱ مقایسه می‌شود و $D(x)$ نیز مقدار پیش‌بینی شده‌ی دسته‌بندی کننده‌ی تمايزگر از تصویر غیرواقعی می‌باشد که با بروجسب ${}^{\circ}$ مقایسه می‌شود.

$$E_x[(D(x) - 1)^2] + E_z[(D(z))^2] \quad (3-3)$$

حال مقدار این زیان برای مولد بدین صورت می‌باشد که سعی دارد خروجی‌های غیرواقعی یعنی $G(z)$ را تا حد ممکن واقعی نشان دهد پس در زیان آن باید مربع فاصله‌ی تصاویر غیرواقعی تا ۱ را یعنی $(D(G(z)) - 1)^2$ که خط نماینده‌ی تصاویر واقعی می‌باشد را کمینه کنیم. فرمول آن به صورت زیر می‌باشد:

$$E_z[(D(G(z)) - 1)^2] \quad (4-3)$$

مزیتی که این تابع زیان دارد این است که احتمال رخداد مشکل ناپدید شدن گرادیان^۱ را کاهش می‌دهد. زیرا این تابع تنها هنگامی مسطح می‌شود که نقطه دقیقاً بر روی خط (${}^{\circ}$ یا ۱) می‌باشد که بدین

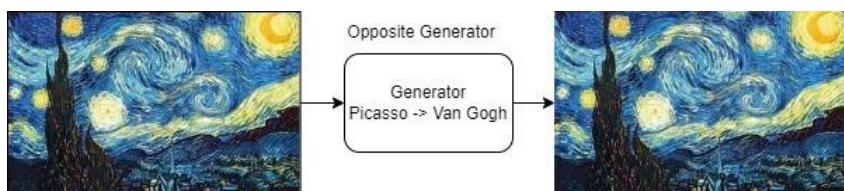
^۱Vanishing gradient problem

معناست که مدل پیش‌بینی کاملاً دقیق انجام داده است. در نتیجه‌ی آن مدل ما دارای پایداری بهتری در فرآیند آموزش دیدن می‌باشد.

۳-۳-۳-۳ زیان هویت

این زیان یعنی زیان هویت^۱ در مقاله پیشنهاد داده شده است و عضو ثابت این مدل نمی‌باشد اما می‌تواند در نگهداری رنگ‌های تصاویر^۲ بسیار مفید واقع شود و نگاشت انجام شده توسط مدل را معنادارتر کند. وظیفه‌ی این تابع زیان این می‌باشد که یک تصویر را گرفته و با کمک مدل مولدی که در جهت عکس آن تصویر عمل می‌کند، تصویر را به خودش نگاشت کند. خروجی این عملیات باید دقیقاً تصویر اولیه باشد زیرا مدل مولدی که از آن استفاده شده برای ترکیب کردن سبک و استایل تصاویر در گروه دیگر آموزش داده شده است. پس اگر تصویر از گروهی که به آن متعلق نیست به این مولد داده شود نباید تغییری در سبک و استایل آن بدهد.

به عنوان مثال فرض کنید دو گروه از نقاشی‌های ون‌گوگ و گروه دوم نقاشی‌های پیکاسو می‌باشد. حال ما یکی از نقاشی‌های ون‌گوگ را به مدل مولدی می‌دهیم که آموزش دیده است تا به نقاشی‌های پیکاسو سبک نقاشی‌های ون‌گوگ را بدهد یعنی مدل بر عکس گروه اول. حال خروجی باید دقیقاً همان نقاشی ون‌گوگ باشد و سبک آن هیچ تاثیری از مدل مولد نگرفته باشد. شما آن به صورت زیر می‌باشد:



شکل ۳-۵ تاثیر نگرفتن طرح و استایل تصویر ورودی از گروه اول با اعمال شبکه‌ی مولد تقابلی گروه مخالف

در این تابع زیان نیز می‌توان از به صورت پیکسلی فاصله‌ی دو عکس را سنجید و در صورتی که فاصله‌ی دو عکس صفر باشد به معنای کاملاً مشابه بودن دو تصویر می‌باشد که این حالت مطلوب مدل است. این مقدار زیان که دوباره برای هر دو مدل مولد می‌باشد (هر دو جهت) نیز با مقادیر زیان قبلی (با یک ضرب) جمع می‌شود تا زیان نهایی مدل بدست بیاید. بدون استفاده از این زیان، احتمال وقوع تحریف رنگ^۳ وجود دارد. اما دلیلی که باعث شده‌است تا این زیان به صورت ثابت و همیشگی در مدل قرار نگیرد این است که در برخی مواقع این زیان کمک چندانی به بهبود مدل نمی‌کند و در نتیجه یک

¹Identity loss

²Color preservation

³Color distortion

بار پردازشی اضافی به مدل تحمیل می‌شود.

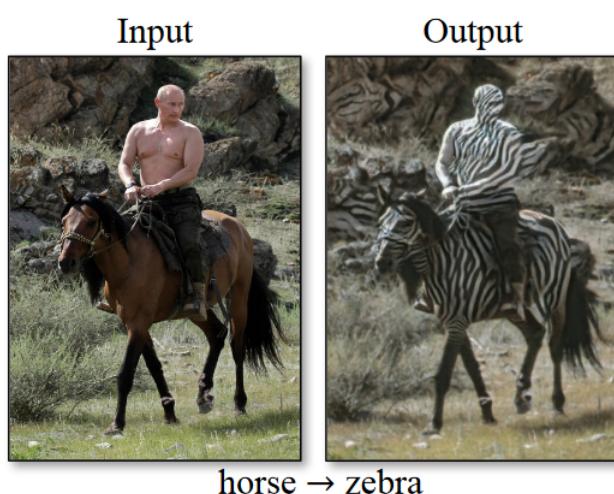
۴-۳-۳ معايب سايكل گن

به طور کلی اين مدل در كارهایی که نیاز است تا علاوه بر سبک تصاویر، تغییرات بیشتری در ترجمه‌ی تصویر انجام شود دچار مشکل می‌شود. به عنوان مثال همانطور که در مقاله‌ی اصلی آورده شده است در تبدیل کردن تصویر سگ به گربه و برعکس اصلاً به خوبی عمل نکرده است زیرا نیاز به تغییراتی فراتر از صرفا سبک تصویر است. در شکل زیر این تغییر ناموفق را می‌بینید:



شكل ۶-۳ عملکرد بد مدل سایکل گن بر روی تصویر آزمون با تغییرات ساختاری زیاد [۲]

حالت دیگری که در آن این مدل نمی‌تواند به خوبی عمل کند موقعی است که شبیه تصویری که به مدل داده می‌شود در مجموعه دادگان وجود نداشته باشد. به عنوان مثال در برنامه‌ی تبدیل اسب و گورخر به یکدیگر، چون در تصاویر موجود در دادگان بر هیچ اسب یا گورخری یک انسان سوار نبوده در نتیجه اگر به مدل تصویر یک اسب که یک سوار بر روی خود دارد را بدهیم خروجی نامناسبی می‌هد. در شکل زیر مشخص است:

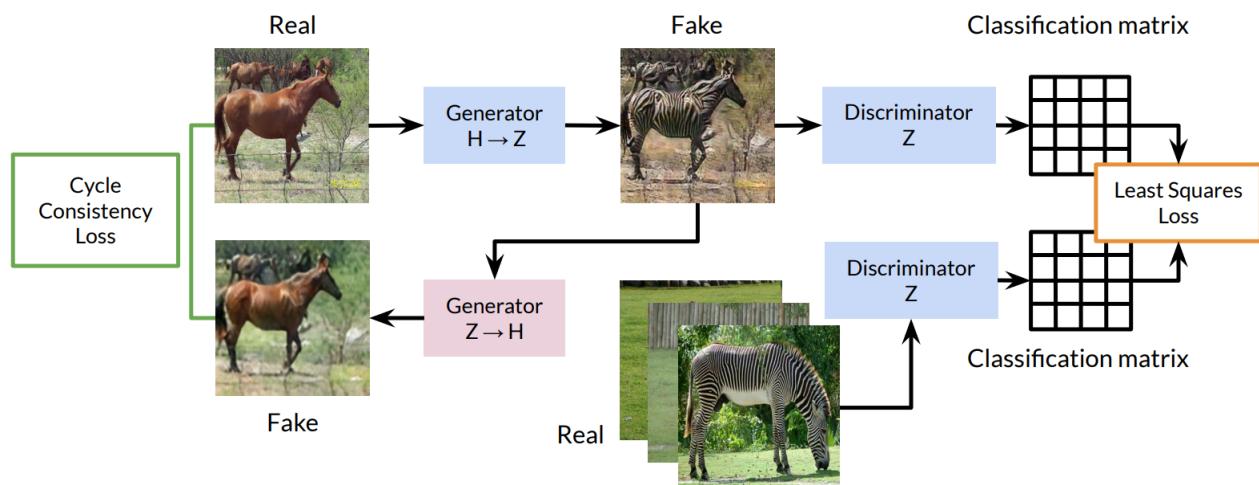


شكل ۷-۳ عملکرد بد مدل سایکل گن بر روی تصویر آزمون ناشناخته [۲]

همانطور که مشخص است در این تصویر مردی که سوار بر اسب می‌باشد نیز دارای رده‌های بدن گورخر شده‌است که مشخصاً این خروجی مطلوب ما نیست.

۵-۳-۳ خلاصه و جمع‌بندی

رونده‌کلی آموزش دادن این مدل بدین صورت می‌باشد که در یک جهت، تصویر از گروه اول توسط مولد با سبک و استایل تصاویر گروه دوم ترکیب می‌شود. سپس خروجی به تشخیص‌دهنده گروه دوم داده می‌شود تا تشخیص بدهد که چه مقدار تصویر ایجاد شده واقعی بنظر می‌رسد که برای این مرحله از تابع زیان حداقل مربعات استفاده می‌شود. حال خروجی‌ای که توسط مولد اول ایجاد شده به مولد دوم (که این مولد وظیفه‌ی ترکیب کردن تصاویر را با سبک تصاویر گروه اول دارد) داده می‌شود و سپس خروجی‌ای که بدست می‌آید با کمک زیان سازگاری چرخه، شباهتش با تصویر اولیه بررسی می‌شود. همچنین تمام این مراحل به صورت وارونه نیز انجام می‌شود تا مدل نهایی توانایی انتقال سبک از هر دو گروه به گروه دیگر را به خوبی داشته باشد. در شما زیر نمونه‌ای از این روند آموزش را برای فقط یک جهت می‌بینید:



شکل ۸-۳ ساختار کلی مدل سایکل‌گن [۲]

همچنین در صورت نیاز در هر دو جهت از زیان هویت نیز استفاده می‌شود تا تابع زیان نهایی از مزایای زیان هویت نیز برخوردار شود و خروجی‌هایش معنادارتر باشند و رنگ تصاویر بهم نریزد. به صورت خلاصه، در تابع زیان مولد این مدل از زیان‌های زیر استفاده می‌شود.

$$\begin{aligned}
 Loss_{Generator} = & LeastSquaresLoss_{M^1 M^2} + LeastSquaresLoss_{M^2 M^1} + \\
 & CycleConsistencyLoss_{M^1 M^2 M^1} + CycleConsistencyLoss_{M^2 M^1 M^2} + \\
 & IdentityLoss_{M^1 M^2} + IdentityLoss_{M^2 M^1}
 \end{aligned} \tag{۵-۳}$$

اما برای تمایزگر دیگر بدین پیچیدگی نمی‌باشد و تابع زیان آن تنها از دو زیان حداقل مربعات ایجاد شده است.

۴-۳ خلاصه

در این فصل به بررسی عمیق‌تر دو مدل انتخاب شده یعنی مدل گلو و سایکل‌گن پرداختیم. دیدیم که مدل گلو یک مدل مولد مبتنی بر جریان است و مزایای مدل‌های مبتنی بر جریان را نسبت به سایر مدل‌های مولد دیدیم.

به طور خلاصه مطرح شد که مدل گلو بر پایه‌ی دو مدل NICE [۱۰] و RealNVP [۱۱] می‌باشد و با اضافه نمودن یک بخش به جریان این مدل توانسته است تا تبدیل‌های^۱ های بهتری را ایجاد کند که در نتیجه می‌تواند یک نگاشت بسیار دقیق و بدون از دست داده‌ی ورودی به فضای پنهان میانی^۲ داشته باشد.

سپس به بررسی مدل سایکل‌گن پرداختیم و دیدیم که این مدل از دو شبکه‌ی مولد تقابلی^۳ [۶] ایجاد شده‌است که یکی از این شبکه‌ها وظیفه‌ی ایجاد تغییر در تصویر ورودی با استفاده از طرح و استایل تصاویر گروه دوم و شبکه‌ی دیگر نیز وظیفه‌ی معکوس دارد. همچنین به توضیح مقادیر مختلف زیان برای توابع زیان^۴ این مدل پرداختیم که این زیان‌ها اطمینان حاصل می‌کنند که در کار ما یعنی پیر کردن چهره، چهره‌ی پیر شده‌ی خروجی چهره‌ی همان شخصی که در تصویر ورودی است باشد و هویت فرد دچار تغییر نشود.

¹transformation

²Latent space

³Generative Adversarial Network

⁴Loss functions

فصل چهارم

ارزیابی و چالش‌ها

۱-۴ مقدمه

در فصل‌های گذشته دیدیم که چه روش‌هایی را می‌توانستیم بکار بگیریم برای انجام دادن عمل پیر کردن چهره و چرا دو مدل گلو^[۱] و سایکل گن^[۲] را انتخاب کردیم. سپس به توضیح این دو مدل پرداختیم. حال در این فصل قصد داریم تا خروجی‌های این دو مدل را با کمک یک مدل سوم یعنی یک مدل تخمین سن با یکدیگر مقایسه کنیم و ببینیم که کدام مدل توانسته است تا عمل پیر کردن را بهتر انجام دهد.

۲-۴ تخمین سن

۱-۲-۴ مقدمه

تخمین سن می‌تواند به عنوان یک مسئله‌ی دسته‌بندی^[۳] تشخیص بازه‌ی سنی، و یا یک مسئله‌ی رگرسیونی^[۴] درنظر گرفته شود که ما در این پژوهه به جهت رسیدن به مقادیر دقیق‌تر از روش رگرسیونی استفاده کردیم تا به سن دقیق بررسیم و کار مقایسه ساده‌تر شود. از نکات مهم دیگر برای عمل تخمین سن نرمال‌سازی کردن^[۵] تصاویر برای این کار می‌باشد که شامل تراز کردن چهره^[۶] و حذف کردن بخش‌های دیگر تصویر با کوچکتر کردن تصویر می‌باشد که البته در مسئله‌ی ما این بحث مشکلی ندارد چون تصویر به صورت خوبی تراز می‌شود.

۲-۲-۴ ساختار کلی مدل تخمین سن [۳] [۴]

روندهای این مدل‌ها به صورت زیر می‌باشد که ابتدا تصویر چهره‌ی ورودی تراز می‌شود تا مدل بتواند بهتر و تنها بر روی چهره مرکز کند تا اجزای دیگری که در تصویر قابل مشاهده است. سپس یک سری ویژگی‌ها از تصویر استخراج می‌شود و در نهایت این ویژگی‌ها به یک دسته‌بندی کننده و یا یک رگرسور داده می‌شود تا خروجی مدنظر را بدهد.

همانند کارهای دیگر، برای انجام دادن این کار نیز می‌توانیم از مدل‌های مختلفی برای بخش استخراج ویژگی‌های صورت بهره بگیریم. می‌توان گفت که این کار به صورت کلی به سه دسته می‌تواند تقسیم شود. ویژگی‌های مبتنی بر هندسه^[۷] که شامل ثبت و تحلیل نقاط برجسته صورت مانند موقعیت چشم‌ها، بینی و دهان، برای استخراج اطلاعات مرتبط با تخمین سن استفاده می‌شوند. ویژگی‌های مبتنی بر ظاهر

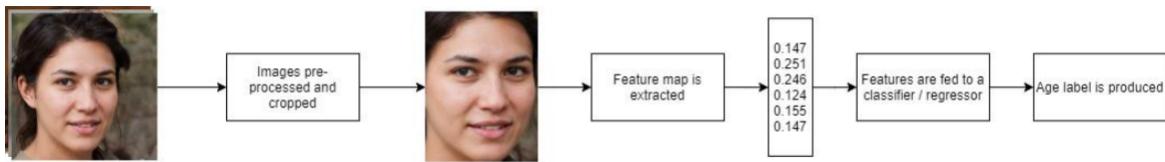
^[۱]Classification

^[۲]Regression

^[۳]Normalization

^[۴]Face Alignment

^[۵]Geometric-based features



شکل ۱-۴ شماتیکی آموزش یک مدل تخمین سن [۳][۴]

^۱ که بر روی ویژگی‌های بصری چهره، مانند بافت، چروک‌ها و رنگ پوست، تمرکز دارند و با استفاده از آنها سن تخمین زده می‌شود. در نهایت ویژگی‌های مبتنی بر یادگیری عمیق^۲ که از قدرت شبکه‌های عصبی عمیق استفاده می‌کنند تا به صورت خودکار ویژگی‌های تمایزدهنده را از تصاویر چهره استخراج کنند و یاد بگیرند.

هر روش استخراج ویژگی نقاط قوت و ضعف خاص خود را دارد. ویژگی‌های مبتنی بر هندسه عموماً در مقابل تغییرات نورپردازی و حالت چهره^۳ خوب عمل می‌کنند، اما ممکن است در تشخیص عناصر جزئی تر و دقیق‌تر پیری چهره دچار مشکل شوند. ویژگی‌های مبتنی بر ظاهر اطلاعات جزئی تری را ارائه می‌دهند، اما ممکن است در برابر تغییرات شرایط نوری و کیفیت تصویر حساس باشند. در نهایت ویژگی‌های مبتنی بر یادگیری عمیق نتایج قابل قبولی در درک الگوهای پیچیده داشته‌اند، اما ممکن است به مقدار زیادی داده آموزش برچسب‌گذاری شده و منابع محاسباتی گسترده نیاز داشته باشند.

۳-۲-۴ چالش‌های عمل تخمین سن

البته این نکته نیز قابل ذکر است که این مسئله دارای پیچیدگی‌هایی می‌باشد از جمله این که تشخیص سن و فرآیند پیر شدن برای هر فرد با توجه به برخی عوامل درونی مانند سبک زندگی، نژاد، جنسیت، وضعیت سلامت و برخی عوامل خارجی مانند میزان در معرض آفتاب بودن، زخم بر روی صورت، آرایش و موارد دیگر متفاوت می‌باشد.

۴-۲-۴ پایگاهداده‌های بکار گرفته شده برای عمل تخمین سن

تهیه مجموعه دادگان آموزشی مناسب برای عمل تخمین سن، مرحله‌ای بسیار حیاتی برای ساخت این مدل‌ها است. با این حال، بدست آوردن یک مجموعه داده کامل به عنوان مجموعه آموزش، به دلیل عدم توازن و عدم توزیع یکنواخت نمونه‌ها در بیشتر موارد چالش‌برانگیز است. زیرا مجموعه دادگان برای این منظور باید به صورتی باشد که از چهره‌های اشخاص با نژادهای مختلف، جنسیت‌های متفاوت، رنگ‌های

¹Appearance-based features

²Deep learning-based features

³Pose

پوست متفاوت و در بازه‌ی سنی زیاد را شامل شود. در غیر این صورت خواهیم دید که مدلی که با استفاده از این مجموعه دادگان آموزش می‌بینند دارای دقت مناسبی نمی‌باشد و در حالات زیادی دارای بایاس خواهد بود که مطلوب ما نیست. در این بخش به معرفی برخی از مجموعه دادگانی که برای این هدف بیشتر مورد استفاده قرار گرفته‌اند با توضیح مختصر بعضی از آن‌ها می‌پردازیم. [۳] [۴] :

۱-۴-۲-۴ مجموعه دادگان IMDB-WIKI [۵]

تاکنون، این مجموعه داده بزرگ‌ترین مجموعه داده‌ی در دسترس برای عموم است که ۵۲۳۰۰۰ تا نمونه برچسبدار از ۲۰۲۸۴ فرد با رده سنی بین ۱ تا ۹۰ سال را شامل می‌شود. این مجموعه داده شامل ترکیب ۴۶۰۷۲۳ و ۶۲۳۲۸ نمونه از شرایط بدون شرط و محدودیت هنرمندان از IMDB و ویکی‌پدیا می‌باشد. بیشتر نمونه‌ها مربوط به افرادی با سن بین ۲۰ تا ۵۰ سال هستند و تعداد کمتری تصویر افراد زیر ۲۰ سال در این مجموعه داده وجود دارد. این مجموعه داده برای استفاده در تحقیقات علمی آنلاین در دسترس می‌باشد.

۲-۴-۲-۴ مجموعه دادگان CACD

این مجموعه داده در ابتدا برای وظایف تشخیص چهره معرفی شد، اما سپس برای آموزش مدل‌های تخمین سن استفاده شد. این مجموعه داده شامل ۱۶۳۰۰۰ تصویر چهره از ۲۰۰۰ هنرمند با سن بین ۱۶ تا ۶۲ سال است. نمونه‌های این مجموعه داده به صورت همزمان در شرایط کنترل شده و دون شرط و محدودیت گرفته شده‌اند. در این مجموعه داده تقسیم‌بندی واضحی از توزیع نمونه‌ها بین گروه‌های سنی و جنسیت‌ها وجود ندارد.

۳-۴-۲-۴ مجموعه دادگان WebFace

این مجموعه داده شامل ۴۹۴۰۰۰ تصویر چهره از ۱۰۰۰۰ فرد در شرایط دون شرط و محدودیت است. این مجموعه داده سینین بین ۱ تا ۸۰ سال را پوشش می‌دهد و نتیجه جمع‌آوری تصاویر از گوگل^۱ و فلیکر^۲ است.

۴-۴-۲-۴ مجموعه دادگان MORPH

این مجموعه داده تا به حال پراستفاده‌ترین مجموعه داده برای ساخت و آموزش مدل‌های تخمین سن بوده است. این مجموعه داده شامل ۵۵۰۰۰ تصویر گرفته شده در شرایط کنترل شده است. نمونه‌های این

¹Google

²Flicker

مجموعه داده نماینده ۱۳۰۰۰ فرد با سن بین ۱۶ تا ۷۷ سال هستند. تصاویر این مجموعه داده در دو آلبوم MORPH-II و MORPH توزیع شده‌اند.

۵-۴-۲-۴ مجموعه دادگان UTKFace

این مجموعه داده شامل بیش از ۲۰۰۰۰ تصویر از افراد با سنین بین ۰ تا ۱۱۶ سال است. این تصاویر در شرایط دون شرط و محدودیت با روشنایی‌ها، پوشش‌ها ووضوح‌های مختلف گرفته شده‌اند. این تصاویر با استفاده از عناصری مانند سن، جنسیت، نژاد و زمانی^۱ مشخص شده‌اند.

۶-۴-۲-۴ مجموعه دادگان FGNET

این مجموعه داده به طور معمول برای ساخت مدل‌های تخمین سن استفاده می‌شود. این مجموعه داده شامل ۱۰۰۲ تصویر رنگی و خاکستری از ۸۲ فرد با سنین بین ۰ تا ۶۹ سال است. تمام تصاویر در شرایط کنترل شده گرفته شده‌اند.

۵-۲-۴ مدل استفاده شده برای عمل تخمین سن

برای این بخش از مدل DEX^۲ [۱۸] [۵] بهره برده شده است. روش پیشنهادی یک روش یادگیری عمیق برای تخمین سن از یک تصویر چهره بدون استفاده از نشانگرهای چهره^۳ است. این روش بر پایه‌ی یک فرآیند مقاوم^۴ تطابق چهره، معماری عمیق VGG-16 و یک دسته‌بندی کننده و پس از آن یک فرمولاسیون مقدار مورد^۵ انتظار مسئله تخمین سن ساخته شده است.

ابتدا، تصویر چهره با استفاده از یک روش مقاوم تطابق چهره به گونه‌ای تنظیم و بریده می‌شود که چشمان و دهان در تمام تصاویر در موقعیتی ثابت قرار داشته باشند. سپس، تصویر تنظیم و تراز شده به شبکه از پیش آموزش دیده شده VGG-16 داده می‌شود تا ویژگی‌ها استخراج شوند. خروجی لایه‌ی آخر VGG-16 که یک لایه‌ی کاملاً متصل^۶ است به عنوان ورودی به یک دسته‌بندی کننده‌ی سافت‌مکس^۷ داده می‌شود که توزیع سن بر روی ۱۰۰ کلاس را پیش‌بینی می‌کند.

در نهایت، یک مرحله‌ی بهبود مقدار مورد انتظار^۸ اعمال می‌شود تا دقت را بهبود ببخشد. این مرحله

¹Timestamp

²Deep EXpectation

³Facial landmarks

⁴Robust

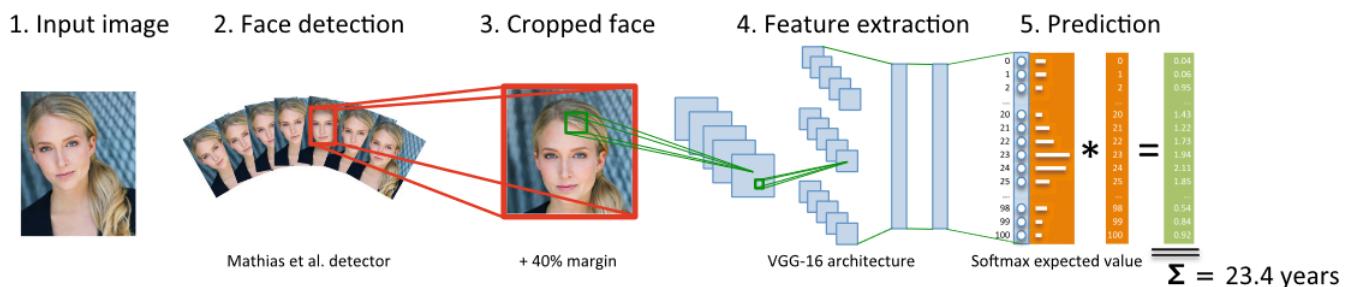
⁵Expected value formulation

⁶Fully connected

⁷Softmax

⁸Expected value refinement

شامل محاسبه مقدار انتظار توزیع سن پیش‌بینی شده و استفاده از آن به عنوان سن تخمینی نهایی است. برای محاسبه مقدار انتظار هر کلاس سن پیش‌بینی شده با احتمال مربوطه‌ی آن کلاس در توزیع پیش‌بینی شده ضرب می‌شود. جمع این حاصل‌ضرب‌ها مقدار انتظار توزیع سن پیش‌بینی شده‌ی نهایی را به ما می‌دهد. سپس این مقدار انتظار به عنوان سن تخمینی نهایی استفاده می‌شود. نویسنده‌گان نشان می‌دهند که این مرحله بهبود قابل توجهی در دقت نسبت به استفاده فقط از حالت یا میانگین توزیع پیش‌بینی شده دارد.



شکل ۲-۴ روند کلی مدل DEX [۵]

علاوه بر موارد گفته شده برای مدل تخمین سن، این مقاله مجموعه دادگان IMDB-WIKI را که در بخش قبل به توضیح آن پرداختیم ایجاد و معرفی کرده است.

۳-۴ چالش‌های ارزیابی کمی مدل‌های پیرکننده‌ی چهره

در این بخش به بررسی برخی راهکارهایی که می‌توان در پیش گرفت برای ارزیابی مدل‌هایی که عمل پیرکردن چهره را انجام می‌دهند می‌پردازیم و بیان می‌کنیم که پیش‌نیاز هر کدام از این راهکارها چیست.

۱-۳-۴ استفاده از مجموعه دادگان دارای برچسب سن

اولین راهی که به ذهن می‌رسد برای ارزیابی مدل‌های پیرکننده‌ی چهره، استفاده از یک مجموعه دادگانی است که دارای سن افراد می‌باشد. همانطور که در بخش قبل توضیح داده شد به چنین مجموعه دادگانی دسترسی داریم. اما برای ارزیابی درست این پروژه، نیاز به مجموعه دادگانی است که در آن تصویر چهره‌ی افراد در سنین مختلف موجود باشد. زیرا اگر بدین صورت نباشد دیگر حالت درستی برای مدل‌های خود نداریم تا خروجی‌هایشان را با آن مقایسه کنیم. به عنوان مثال فرض کنید که چهره‌ای از فردی ۳۰ ساله داریم. حال با کمک مدل‌هایمان چهره‌ی فرد را پیر می‌کنیم و به عنوان مثال مدل تخمین سن تخمین می‌زند که سن فرد به ۵۰ سال رسیده است اما چون چهره‌ی واقعی آن شخص را در ۵۰ سالگی اش نداریم نمی‌توانیم خروجی مدل‌مان را مورد مقایسه قرار دهیم و نمی‌توانیم متوجه شویم که چه مقدار چهره‌ی خروجی به چهره‌ی ۵۰ ساله‌ی واقعی آن شخص شبیه است.

اما در حال حاضر مجموعه دادگانی موجود هستند که چهره‌ی افراد را در سنین مختلف داشته باشند. به عنوان مثال مجموعه دادگان چهره‌ی یک شخص را هم در سن ۳۰ سالگی و هم در سن ۴۰ سالگی دارد. حال مشکل دیگری که برای این سری از مجموعه دادگان پیش می‌آید این است که مدل‌های ما چون از لحاظ مشخص کردن سن خروجی شرطی و قابل کنترل نیستند نمی‌توانیم چهره‌ی شخصی که در ورودی تصویرش را می‌گیریم به سن مشخص و یا دسته‌ی سنی مشخصی ببریم. به عنوان مثال نمی‌توانیم مشخص کنیم که چهره‌ی شخص به مقدار ۱۰ سال پیر شود یا ۱۵ سال یا به یک رده‌ی سنی مشخص به عنوان مثال بازه‌ی ۴۰ تا ۴۵ سال تبدیل شود. این محدودیت باعث می‌شود تا از مجموعه دادگانی که چهره‌ی اشخاص در سنین مختلف را دارند نتوانیم استفاده کنیم. زیرا در این مجموعه دادگان به عنوان مثال چهره‌ی شختی در ۳۰ و ۵۰ سالگی را داریم. حال با استفاده از مدل‌هایمان چهره‌ی شخص را پیر کرده‌ایم و مدل تخمین سن عدد ۶۰ را برای چهره‌ی خروجی تشخیص داده‌است. در این حالت دیگر برچسب سن مجموعه دادگان به کار نمی‌آید زیرا ما سن تخمینی ۶۰ سال را برای چهره‌ی ورودی داریم و نمی‌توانیم آن را با برچسب ۵۰ سالگی مجموعه دادگان مقایسه کنیم. ممکن است مدل ما سن ۶۰ سال فرد را به خوبی ترسیم کرده باشد اما چون برچسب ما ۵۰ سالگی فرد را نشان می‌دهد آن وقت دقت مدل ما پایین محاسبه می‌شود که درست و دقیق نمی‌باشد و نمی‌تواند معیار قرار گیرد. بدین دلایل برای رفع این مشکل دو روش را می‌توان پیش گرفت:

- استفاده از مجموعه دادگانی که سن افراد را در بازه‌ی گسترده‌ای داشته باشد تا تخمینی که از خروجی مدل ما بدست می‌آید بتواند با یک حالت درست چهره در آن سن مقابسه شود که متسافنه چنین مجموعه دادگان کامل و پیچیده‌ای موجود نیست.
- استفاده از مدل‌های مولدی که توانایی تولید شرطی تصاویر در سنین مختلف را داشته باشند که درست کردن چنین مدلی خود در گام اول نیاز به مجموعه دادگانی دارد که چنین برچسب‌هایی را داشته باشد که دوباره به مورد یک برمی‌گردیم.

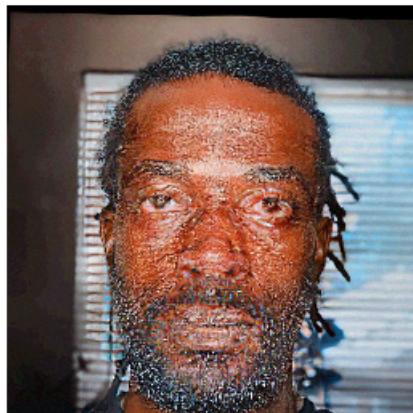
۲-۳-۴ استفاده از یک نگاشت از مقدار آلفا بر سنین و حالات مختلف

روش دیگری که می‌توان پیش گرفت این است که مجموعه‌ی ^۳تایی هایی تشکیل دهیم از تصویر ورودی، تصویر پیر شده با سن تخمین زده شده و تصویر واقعی شخص در سن پیری. سپس با امتحان کردن مقادیر مختلف آلفا در مدل گلو ببینیم که کدام مقدار آلفا به چه مقداری چهره را پیر می‌کند. در واقع یک نگاشتی پیدا کنیم از مقدار آلفا و میزان سن پیر شدن خروجی. اما این روش دارای مشکلات زیر می‌باشد:

- این روش نیاز به تست کردن مقادیر زیاد آلفا بر روی تعداد بسیار زیادی از تصاویر دارد تا بتواند یک نگاشت بدست بیاورد و از آنجایی که هر بار اجرای مدل گلو حدود ۱ دقیقه به طول می‌انجامد واضح است که این روش عملی نمی‌باشد و نیاز به زمان بسیار زیادی دارد.

- حتی اگر بتوان این کار را انجام داد از آنجایی که مدل گلو مدل دارای بسیار زیادی در تنوع نژادها می‌باشد که دلیل آن نیز آموزش دیدن بر روی مجموعه دادگان محدودی است نمی‌توان به نگاشت مناسبی رسید زیرا این مدل نمی‌تواند تمام چهره‌ها را با یک نسبت مشخص پیر کند و حتی برخی از چهره‌ها را به کلی بهم می‌ریزد مانند تصویر زیر:

Glow Output (alpha: 1.5)



estimated age Glow model: 46

difference: 12 years

شکل ۳-۴ تصویر خروجی بهم ریخته مدل گلو از چهره‌ی یک فرد سیاهپوست

- این روش هیچ راه چاره‌ای برای مدل سایکل‌گن ندارد و حتی اگر به نگاشتی بررسیم نمی‌توانیم آن را با مدل سایکل‌گن مقایسه کنیم.

۳-۳-۴ استفاده از مدل‌های تخمین سن مختلف

روش دیگری که با آن می‌توان تا حدودی خروجی مدل‌های مولدمان را مقایسه کرد این است که به جای استفاده از یک مدل تخمین سن از چند مدل تخمین سن استفاده کنیم و سپس خروجی‌های این مدل‌ها را با یکدیگر مقایسه کنیم. حال اگر خروجی این مدل‌ها یعنی تخمین سن‌هایشان نزدیک به یکدیگر بودند آنگاه می‌توانیم ادعا کنیم که مدل‌های مولدمان خروجی‌های با کیفیت مناسبی را داده‌اند که در نتیجه‌ی آن چندین مدل تخمین سن خروجی‌های نزدیکی داشته‌اند. اما مشکلاتی که این روش دارد در ادامه ذکر شده‌است:

- نزدیک بودن تخمین‌های مدل‌های تخمین سن هیچ تضمینی از درست بودن همه‌ی آن‌ها ندارد.
- این که تخمین که چه باشند تا بتوانیم ادعا کنیم که نزدیک به هم هستند خود بسیار جای بحث دارد.

- این روش نیاز به پیاده‌سازی مدل‌های دیگر تخمین سن دارد که پیاده‌سازی‌های مدل‌های بهتر تخمین سن به غیر از مدلی که در این پروژه استفاده شده است موجود نبود.

۴-۳-۴ استفاده از یک مدل تشخیص واقعی بودن تصویر

می‌توان خروجی‌های مدل‌های مولد را به یک مدل تشخیص واقعی بودن یا نبودن تصویر داد و سپس با توجه به مقدار احتمال واقعی بودن یا نبودن تصویر بر کیفیت تصاویر خروجی پیر شده‌ی مدل‌های مولد قضاوت کرد که مشکل این راه این که متناسبه هنوز مدل پخته‌ی مناسبی برای این کار موجود نیست.

۴-۴ ارزیابی کیفی مدل‌های پیر کننده‌ی چهره

حال که مشکلات ارزیابی کمی برای عمل پیر کردن چهره را دیدیم می‌خواهیم تا در این بخش ارزیابی کیفی‌ای بر خروجی مدل‌های این پروژه داشته باشیم. ابتدا خروجی مدل گلو بر با مقادیر آلفای بین ۱.۲ تا ۲ تا بر روی چهره‌ی یک شخص با رنگ پوست سفید امتحان می‌کنیم. می‌توانیم مشاهده کنیم که مدل گلو عملکرد قابل قبولی داشته است و توانسته است تا به صورت بسیار ظرفی و جزئی چین و چروک‌هایی بر روی چهره دور لب‌ها، زیرچشم‌ها و روی پیشانی بیاندازد و هرچه مقدار آلفا افزایش یافته است مقدار این چین و چروک‌ها نیز زیاد شده است. همچنین تغییر دیگری که به وضوح مشخص است ایجاد ریش و سبیل برای شخص است.

مقدار آلفا بهتر است که بین ۱ و ۲ داده شود تا تصاویر خروجی دارای کیفیت مناسبی باشند و واقع‌گرایی خوبی داشته باشند. اما می‌توان مقادیر بیشتر نیز به آن داد اما در آن صورت دیگر تصویر کاملاً بهم می‌ریزد و دیگر خروجی قابل قبولی تولید نمی‌شود مانند تصویر ۴-۵ که مقدار ۵ برای آلفا در نظر گرفته شده است و تصویر خروجی دیگر بی معنا شده است.

اما مدل سایکل‌گن عملکرد بسیار متفاوتی نسبت به مدل گلو دارد. در تصویر ۶-۴ خروجی مدل سایکل‌گن را در کنار خروجی مدل گلو با آلفا ۷.۱ روی عکسی که پیش‌تر دیدیم ملاحظه می‌کنید. همانطور که در تصویر مشخص می‌باشد مدل سایکل‌گن تغییرات بسیار کمتری را بر روی چهره اعمال کرده است و بیشتر توانسته است تا با تغییر رنگ برخی از نقاط صورت مانند گونه‌ها و زیرچشم‌ها عمل پیر کردن چهره را انجام دهد. همانطور که در فصل ۳ در بخش مدل سایکل‌گن توضیح دادیم یکی از نقاط ضعف این مدل در وظایفی است که باید علاوه بر تغییر رنگ تصویر عمل تغییر دادن طرح و استایل تصویر را با ایجاد تغییرات در ساختار تصویر اعمال کند که در اینجا نیز می‌توان چنین موردی را مشاهده کرد زیرا ایجاد چین و چروک بر روی صورت به عنوان مثال دور دهان و یا روی پیشانی نیاز به ایجاد تغییرات عمیق‌تری در تصویر دارد که مدل سایکل‌گن توانایی آن را ندارد. لازم به ذکر است که اگرچه تغییراتی که مدل سایکل‌گن در چهره ایجاد کرده است شاید در این چهره خیلی واضح نباشد اما مدل



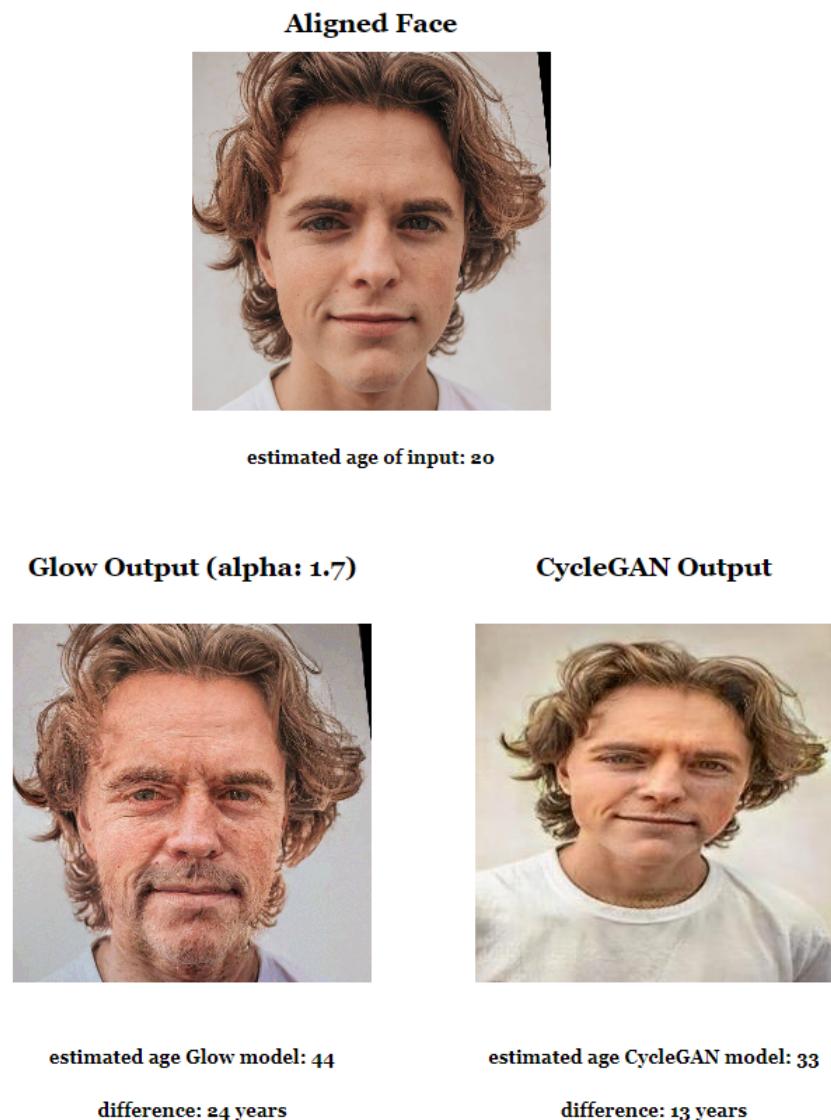
شکل ۴-۴ خروجی مدل گلو با آلفاهای ۱.۲ (بالا سمت چپ) تا ۵ (پایین سمت راست)

Glow Output (alpha: 5)



شکل ۵-۴ خروجی مدل گلو با آلفای ۵

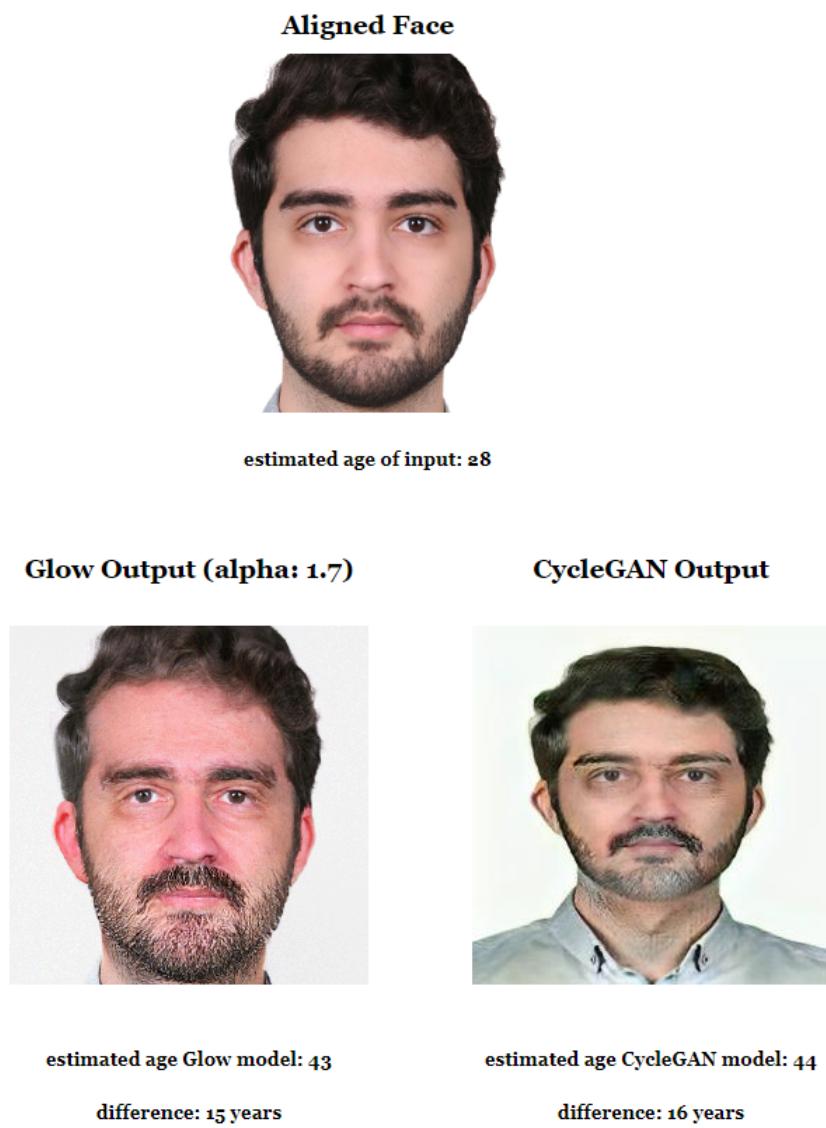
تخمین سن تشخیص داده است که چهره‌ی فرد ۱۳ سال پیرتر شده است. این مقدار برای مدل گلو ۲۴ سال است و می‌بینیم که گلو توانسته است تا تغییرات بسیار بیشتری را بر روی چهره ایجاد کند. حال این مقایسه را بر روی یک چهره‌ی دیگر انجام می‌دهیم.



شكل ۴-۶ مقایسه‌ی خروجی مدل سایکل‌گن با مدل گلو با آلفا ۱.۷

در تصویر ۷-۴ مشاهده می‌کنیم که سن تخمینی هر دو مدل حدوداً یکسان است و هر دو مدل توانسته‌اند تا عمل پیر کردن را به خوبی انجام دهند. مدل گلو با ایجاد چین و چروک دور چشم‌ها و سفید کردن رنگ ریش به طرز بسیار ظرفی و همچنین به مقدار کمی سفید کردن موهای شقیقه عمل پیر کردن را انجام داده است. از آن طرف می‌بینیم که در این تصویر مدل سایکل‌گن توانسته است تا با ایجاد تغییرات رنگی بر روی چهره عمل پیر کردن تا بهتر از حالتی که چهره‌ی شخص ریش نداشت انجام دهد. این مدل با سفید کردن رنگ بخش‌هایی از ریش و شقیقه‌ها و کمی تغییر دادن رنگ دور چشم‌ها توانسته است تا عمل پیر کردن را انجام دهد.

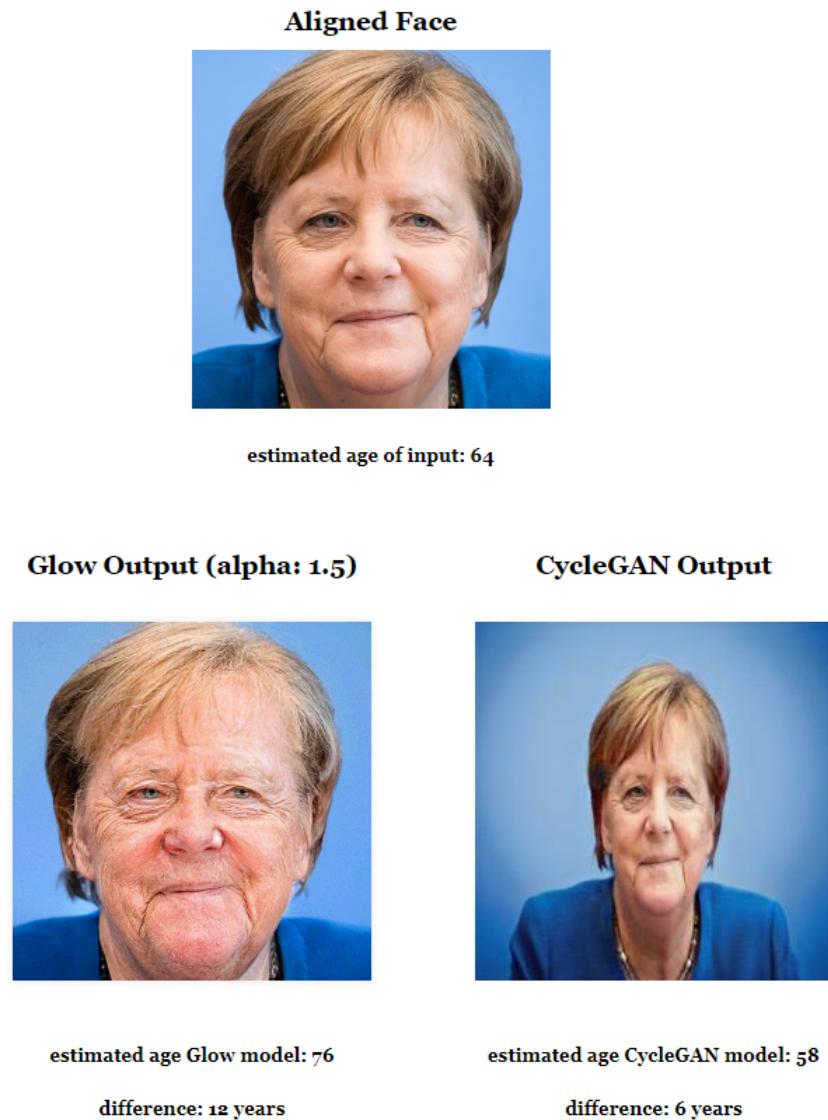
به طور کلی در این دو مقایسه دیدیم که عملکرد هر دو مدل قابل قبول بود. اما حالا می‌خواهیم چندین حالت را بررسی کنیم که در آن‌ها ضعف‌های اساسی این مدل‌ها آشکار می‌شود.



شکل ۷-۴ مقایسه‌ی دوم خروجی مدل سایکل‌گن با مدل گلو با آلفا ۱.۷

مورد اول مربوط به حالتی است که در آن سن شخص بالا (بالای ۶۰) می‌باشد و همچنین جنسیت شخص مونث می‌باشد پس دیگر نمی‌توان به راحتی با گداشتن ریش و یا سفید کردن رنگ ریش عمل پیر کردن را انجام داد. در تصویر ۷-۴ می‌بینیم که مدل گلو توانسته با ایجاد چروک‌های بسیار ظریف و حساب شده در اطراف چشم‌ها عمل پیر کردن را انجام دهد. اما مدل سایکل‌گن عملکرد بسیار بدی را از خود نشان داده است و سن تخمینی را نه تنها زیاد نکرده بلکه به مقدار ۶ سال کم نیز کرده است.

در شکل ۹-۴ چهره‌ی زن جوانی را در ورودی می‌بینیم که با مقدار آلفای بالایی سعی در پیر کردن شده است. نکته‌ای که در اینجا مشهود است این است که مدل گلو با سعی کرده است تا مانند تصاویر

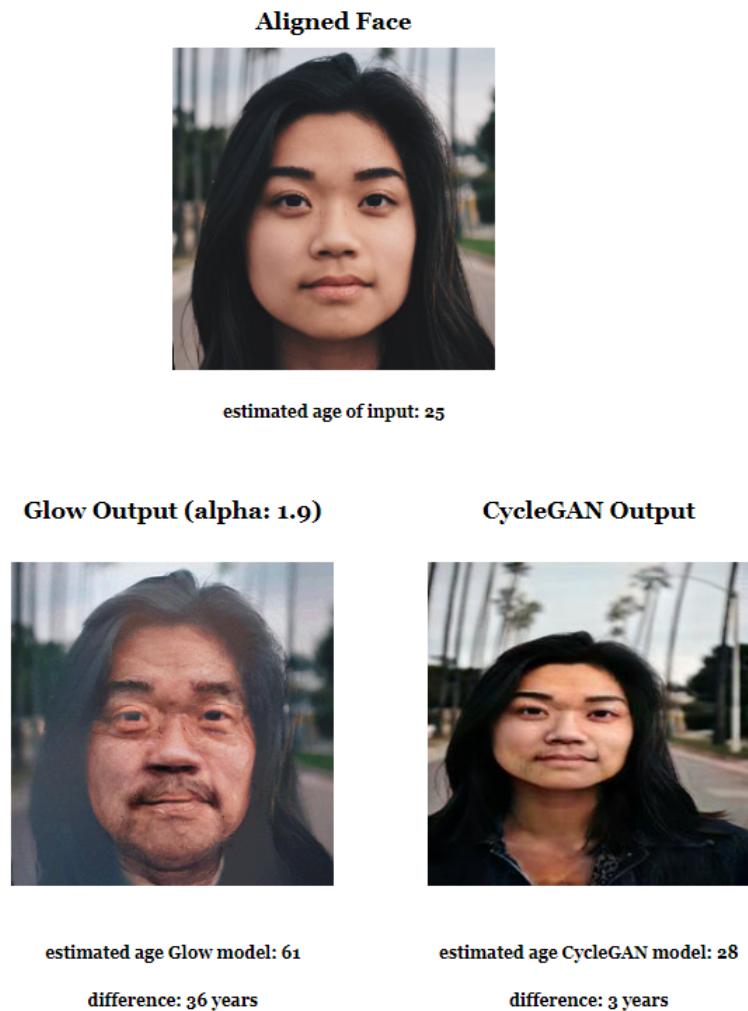


شکل ۸-۴ مقایسه سوم خروجی مدل سایکل گن با مدل گلو با آلفا ۱.۵

چهره‌ی مردها دوباره با قرار دادن ریش و سبیل به پیر کردن چهره بپردازد که در این مثال مطلوب نیست. مدل سایکل گن نیز عملکرد ضعیفی از خود نشان داده و عملاً نتوانسته است تا نشانه‌های پیری در چهره ایجاد کند.

حال در تصویر ۸-۴ چهره‌ی یک شخص با رنگ پوست بسیار روشن و دارای یکسری کک بر روی گونه‌ی آن را داریم. می‌بینیم که هر دو مدل اگرچه توانسته‌اند سن تخمی را افزایش دهند اما عملکرد خوبی نداشته‌اند زیرا کک‌های روی صورت این مدل‌ها را منحرف کرده‌اند و مدل گلو نتوانسته است تا مانند حالات قبل چین و چروک‌های مناسب روی چهره بیاندازد و چهره‌های خروجی هر دو مدل از حالت طبیعی چهره‌ی فرد خارج شده‌اند.

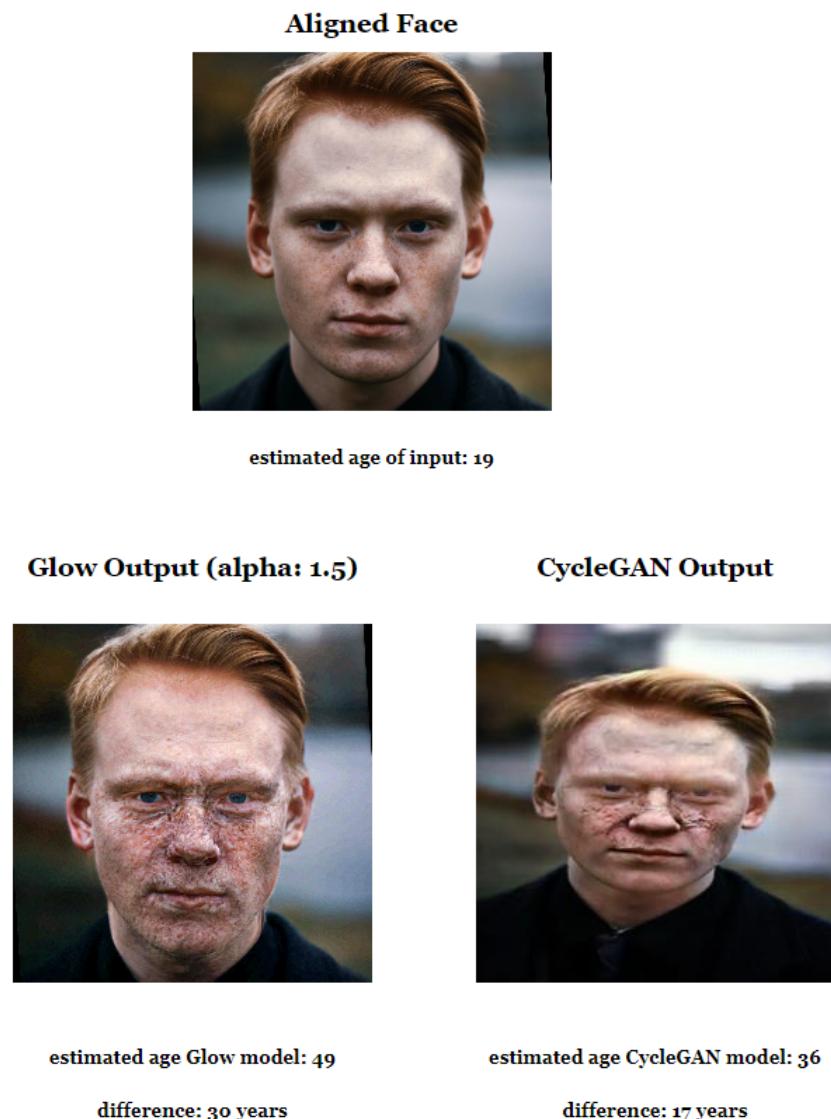
حالت بعدی‌ای که با آن مواجه هستیم زمانی است که تصویر ورودی دارای کیفیت مناسبی نمی‌باشد.



شکل ۹-۴ مقایسه‌ی چهارم خروجی مدل سایکل‌گن با مدل گلو با آلفا ۱.۹

همچنین این مشکل برای بسیاری از مجموعه دادگانی که پیش‌تر معرفی کردیم وجود دارد. در این حالت می‌بینیم که مدل سایکل‌گن تقریباً تغییری بر روی چهره نداده است و تخمین سن خروجی آن با تصویر ورودی تفاوتی ندارد. مدل گلو نیز عملکرد حدوداً نامناسبی از خود نشان داده و با اینکه مقدار ۱.۵ برای آلفا مقدار بالایی نیست که بخواهد چهره را بهم ببریزد اما می‌بینیم که چهره‌ی خروجی این مدل دارای چین و چروک‌های بسیار زیاد مصنوعی می‌باشد.

درنهایت چهره‌ی یک شخص که رنگ پوست تیره‌ای دارد را بررسی می‌کنیم. در تصویر ۱۲-۴ مشاهده می‌کنیم که با اینکه تصویر ورودی دارای کیفیت مناسبی می‌باشد با مقدار آلفای ۱.۵ چهره‌ی خروجی مدل گلو کاملاً بهم ریخته است و چین و چروک‌های روی صورت و تغییرهای صورت بیشتر به حالت نویز می‌باشد تا علائم پیری. همچنین مدل سایکل‌گن نتوانسته است تا تغییرات زیادی را ایجاد کند و سن تخمینی خروجی آن با سن تخمینی ورودی تفاوت چندانی ندارد. دلیل این واقعه را می‌توان به وجود بایاس‌های بسیار زیادی در مجموعه دادگانی که از چهره‌های انسان‌ها برای این کار موجود

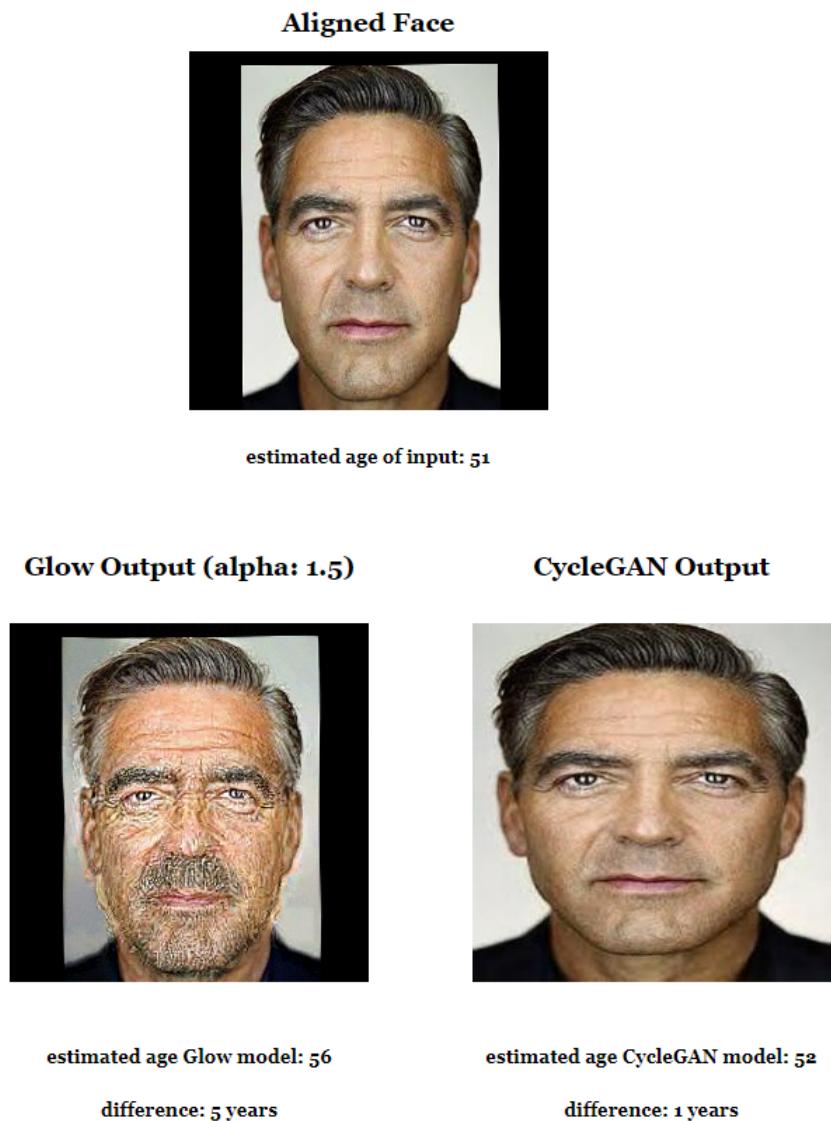


شكل ۱۰-۴ مقایسه پنجم خروجی مدل سایکل‌گن با مدل گلو با آلفا ۱.۵

است دانست. زیرا در این مجموعه دادگان نسبت تصاویر چهره‌های افراد با رنگ پوست تیره کمتر است از افراد با رنگ پوست روشن.

۱-۴-۴ پیشنهادها و کارهای آتی

همانطور که در ارزیابی کیفی مدل‌ها در بخش قبل دیدیم، این مدل‌ها قادر نبودند تا در برخی از حالات به صورت مناسبی عمل پیر کردن چهره را انجام دهند. به صورت خلاصه این موارد شامل بود از حالاتی که شخص دارای رنگ پوست تیره‌ای باشد، زمانی که شخص دارای یکسری کک بر روی چهره‌اش باشد و همچنین در برخی حالات مربوط به زنان به عنوان مثال هنگامی که مقدار آلفا را زیاد قرار می‌دادیم

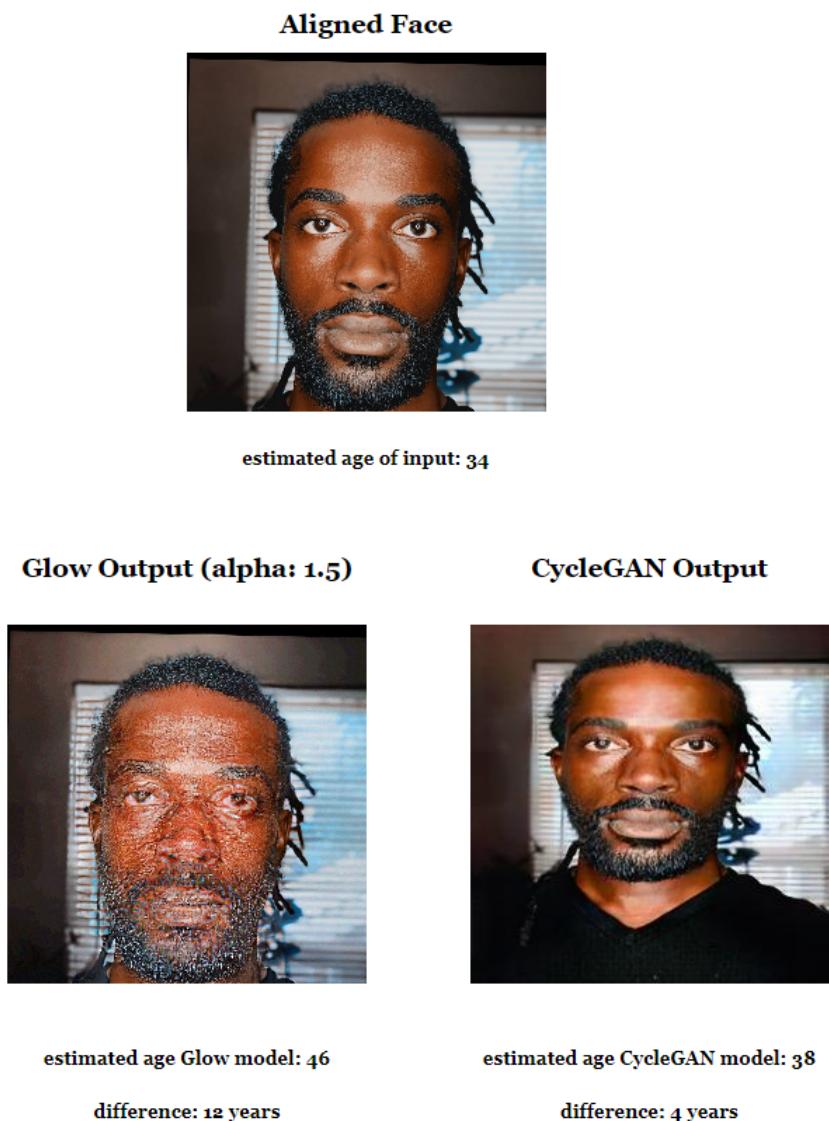


شکل ۱۱-۴ مقایسهٔ ششم خروجی مدل سایکل‌گن با مدل گلو با آلفا ۱.۵

مدل گلو سعی می‌کرد تا با ایجاد ریش و سبیل عمل پیر کردن را انجام دهد.

راهکارهایی که می‌توان در پیش گرفت برای حل این مشکلات از جمله زیر است:

- ایجاد مجموعه دادگانی که در آن‌ها بایاس کمتری وجود داشته باشد تا در نتیجه مدل‌هایی که با این مجموعه دادگان آموزش می‌بینند مدل‌های جامع‌تری باشند و بایاس کمتری داشته باشند.
- یک مدل تشخیص جنسیت قبل از این مدل‌ها قرار گیرد و سپس با توجه به جنسیت فرد عمل پیر کردن انجام شود. بدین صورت می‌توان از مشکل ریش گذاشتن برای خانم‌ها در مدل گلو جلوگیری کرد و همچنین می‌توان از مدل‌های بهینه‌تری برای هر جنسیت استفاده کرد.
- آموزش دادن بسیار بیشتر مدل سایکل‌گن بر روی داده‌های آموزشی با بایاس کمتر و کیفیت بالاتر.



شکل ۱۲-۴ مقایسه‌ی هفتم خروجی مدل سایکل‌گن با مدل گلو با آلفا ۱.۵

چون مدل سایکل‌گن تنها با تغییر رنگ و طرح تصویر سعی به ایجاد نشانه‌های پیری در چهره می‌کند کیفیت تصاویری که روی آن‌ها آموزش می‌بیند اهمیت ویژه‌ای پیدا می‌کند.

۵-۴ خلاصه

در این فصل ابتدا به بررسی یک مدل جهت تخمین سن چهره‌ها [۵] پرداختیم و همچنین چالش‌های عمل تخمین سن را ذکر کردیم. ساختار این مدل را مورد تحلیل قرار دادیم و از آن برای تحلیل کیفی مدل‌های مولد بکار گرفته شده در پروژه بهره بردیم. همچنین به چالش‌ها و مشکلات تحلیل کمی این کار یعنی عمل پیر کردن چهره پرداختیم. علاوه بر این‌ها به برخی از مجموعه دادگانی که برای مدل‌های

مخالف تخمین سن استفاده شده بودند اشاره کردیم.

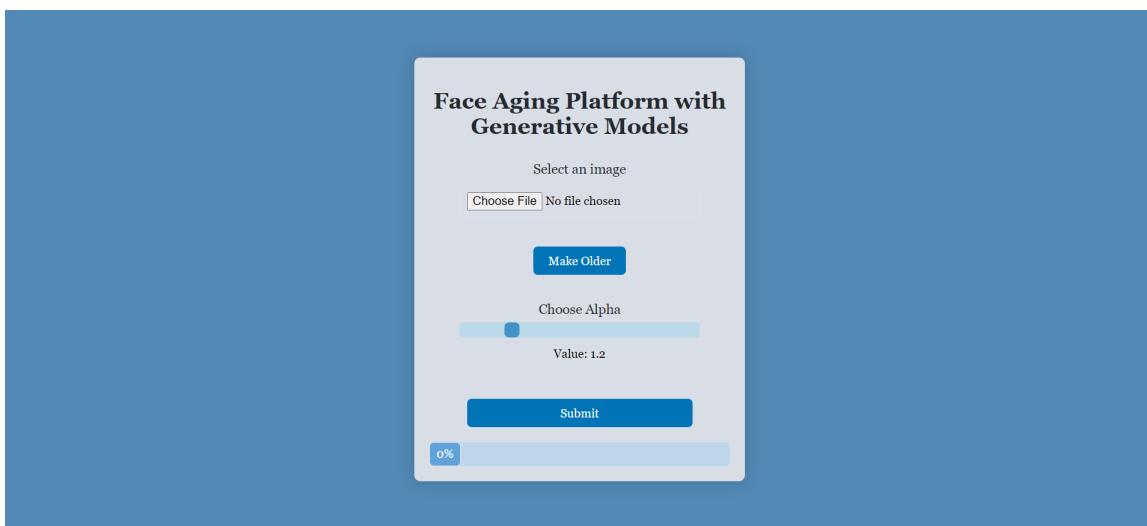
فصل پنجم

پلتفرم ایجاد شده با چارچوب جنگو بر پایه‌ی معماری مايكروسرويس

۱-۵ مقدمه

در این بخش به توضیح و تشریح سامانه‌ای که برای کار کردن ساده‌تر با این برنامه و مدل‌های ذکر شده ایجاد شده است پرداخته می‌شود و ابزارها و مفاهیمی که برای درست کردن این سامانه استفاده شد ذکر می‌شوند. در ابتدا به توضیح ساختار کلی سامانه پرداخته می‌شود و سپس در ادامه به توضیح دقیق‌تر سرور اصلی برنامه که با چارچوب جنگو می‌باشد می‌پردازیم و توضیح می‌دهیم که بخش‌های مختلف آن چه کاری را انجام می‌دهند و چگونه باعث افزایش سرعت توسعه و کم شدن کد می‌شوند. سپس به سراغ تشریح ساختار مایکروسرویسی سامانه می‌رویم و توضیح داده می‌شود که چرا چنین ساختاری برای این برنامه نیاز بود و چه مشکلاتی را حل می‌کرد. در نهایت نیز به توضیح سرورهایی که با چارچوب فلسفک، که چارچوبی تقریباً همانند جنگو اما سبک‌تر می‌باشد، برای این برنامه توسعه داده شد می‌رویم.

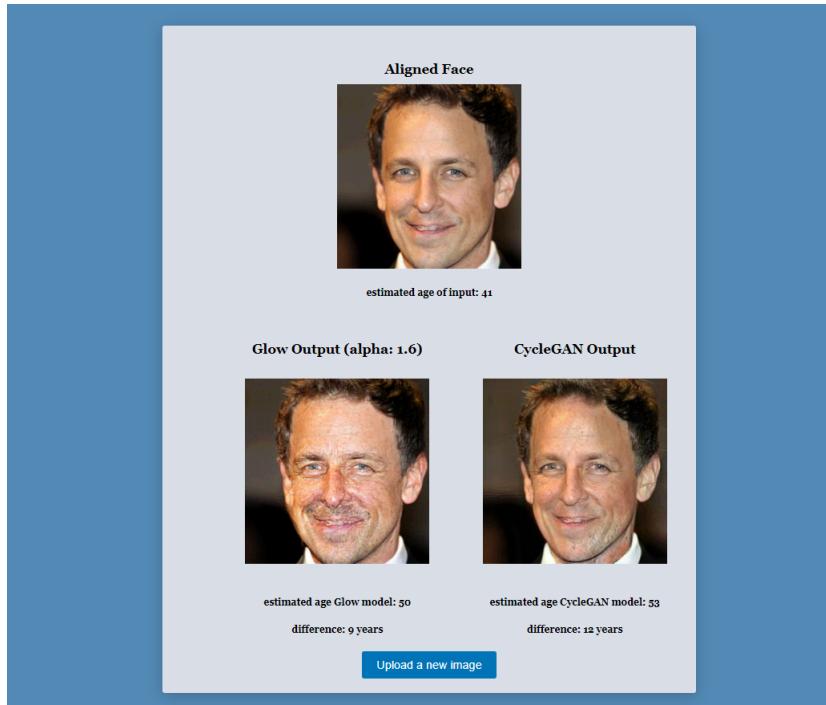
در اینجا تصویری از صفحه‌ی اول سامانه که در آن عکس مورد نظر انتخاب می‌شود و سپس مقدار متغیر آلفا مشخص می‌شود را می‌بینیم. همچنین در این صفحه می‌توان انتخاب کرد که مدل‌های استفاده شده به پیر کردن چهره و یا جوان کردن چهره بپردازند.



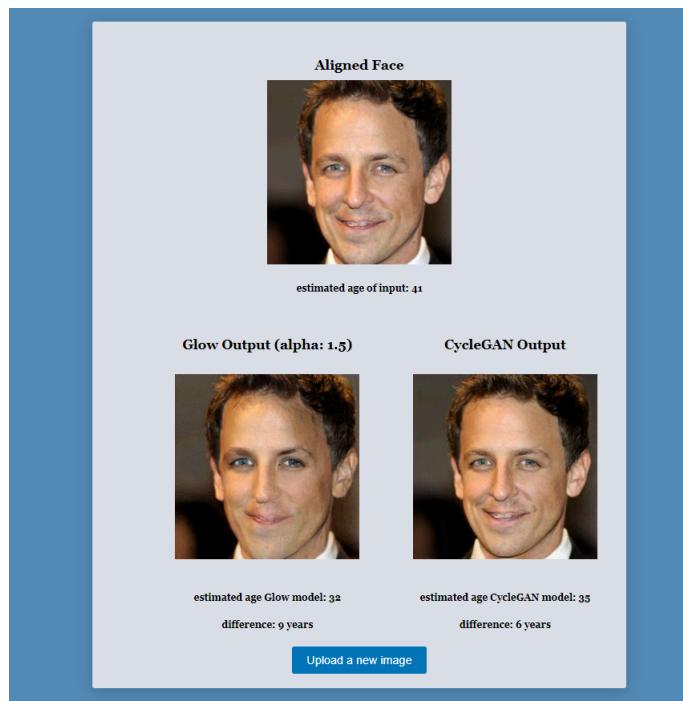
شکل ۱-۵ صفحه‌ی اول سامانه

در تصویر ۲-۵ نیز تصویری از صفحه‌ی دوم سامانه می‌بینیم که عکس ورودی را تراز کرده است و در بالای صفحه قرار داده است. تصاویر خروجی مدل‌های گلو و سایکل کن نیز در زیر آمده‌اند و سن تخمینی هر کدام از چهره‌ها در زیر آن‌ها نوشته شده است.

همچنین صفحه‌ی نتایج را در حالتی که گزینه‌ی جوان شدن انتخاب شده است می‌بینید.



شکل ۲-۵ صفحه‌ی نتایج در حالت پیر شدن چهره



شکل ۳-۵ صفحه‌ی نتایج در حالت جوان شدن چهره

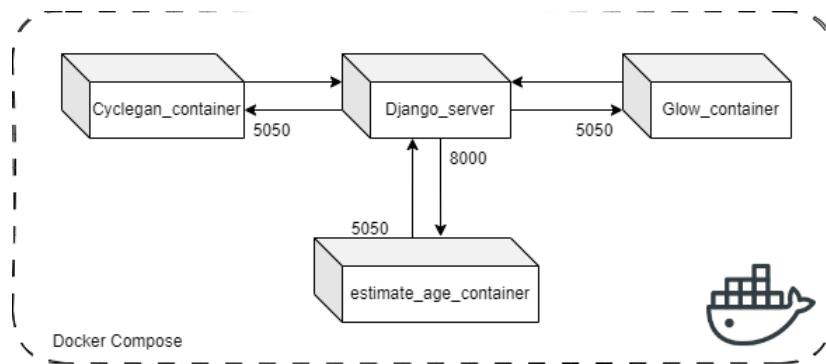
۲-۵ ساختار کلی سامانه

۱-۲-۵ پیاده‌سازی سامانه به صورت یکپارچه

در این پژوهه ما نیاز به پیاده‌سازی ۳ مدل متفاوت هوش‌منوعی داشتیم. اولین و ساده‌ترین راهی که برای ایجاد و پیاده‌سازی این سیستم به ذهن هر شخصی می‌رسد این است که یک سورور ایجاد بشود و سپس نیازمندی‌های این مدل‌های هوش‌منوعی بر روی آن سورور ایجاد و نصب شوند و سپس سامانه به خوبی شروع به کار کند. اما مسئله‌ی مهمی که در این پژوهه با آن رو برو بودیم آن بود که هر کدام از این مدل‌ها دارای ویژگی‌های متفاوت و نیازمندی‌های متفاوتی بودند. برخی از این نیازمندی‌ها نیز با یکدیگر مغایرت داشتند. به عنوان مثال یک مدل هوش‌منوعی نیاز به استفاده از کتابخانه‌ی تنسورفلو نسخه‌ی ۱.۸ داشته و یک مدل دیگر نیاز به استفاده از کتابخانه‌ی تنسورفلو نسخه‌ی ۲.۱. همچنین سنگینی و حجمی بودن نیازمندی‌ها و کتابخانه‌هایی که این مدل‌ها با آن‌ها سر و کار داشتند مانند کتابخانه‌های تنسورفلو و پایتورج و دلیب نیز از مشکلات دیگر پیاده‌سازی این سیستم بود. از مشکلات دیگری که پیاده‌سازی یکپارچه‌ی این سیستم به همراه داشت عیب‌یابی سخت آن می‌بود. زیرا تمام مدل‌های هوش‌منوعی مورد نظر در کنار یکدیگر ایجاد شده‌بودند و بروز مشکل در هر کدام باعث از کار افتادن کل برنامه می‌شد. در نتیجه ایجاد این سامانه به صورت یکپارچه کاری دشوار و غیر منطقی بود.

۲-۲-۵ پیاده‌سازی سامانه به صورت معماری مایکروسرویس

اما راهکار جایگزین روش قبل این است که هر کدام از مدل‌ها در فضایی جدا شده و منزوی قرار گیرند و به صورت سرویس‌های کوچکی در بیانند که این سرویس‌ها با یکدیگر در ارتباط هستند. در این حالت نیاز به یک سرویس اصلی مرکزی برای مدیریت سایر سرویس‌ها و ایجاد ارتباط میان آن‌ها نیز می‌باشد. از مزایای این سیستم می‌توان رفع مشکلات اشاره شده در بخش قبل اشاره کرد. به عنوان مثال از اصلی‌ترین نقاط مثبت این سیستم این است که می‌توان مشکل تضاد داشتن کتابخانه‌ها و نیازمندی‌های مدل‌ها را به سادگی رفع کرد. بدین صورت که هر سرویس دارای کتابخانه‌ها و نیازمندی‌های تنها مدل هوش‌منوعی‌ای که بر روی آن قرار دارد می‌باشد و دیگر نیازی ندارد تا با بقیه‌ی سرویس‌ها تراز باشد. این قابلیت بسیار حیاتی می‌باشد زیرا همانطور که گفته شد برنامه شامل مدل‌های سنگین و دارای مغایرت با یکدیگر می‌باشد. از مزایای دیگر این سیستم عیب‌یابی ساده‌تر آن به نسبت سیستم یکپارچه می‌باشد. در سیستم مایکروسرویس ما هر سرویس که یک مدل هوش‌منوعی بر روی آن سوار است را به صورت جداگانه و مجزا پیاده‌سازی و سپس می‌آزماییم. بدین صورت می‌توانیم مطمئن شویم که قبل از وصل کردن سرویس‌ها و مدل‌ها به یکدیگر، هر سرویس دارد به صورت بی‌نقص اجرا می‌شود و خروجی‌های مورد نظر را می‌دهد. در نتیجه در ادامه‌ی راه کار توسعه‌دهندگان برای اضافه کردن ویژگی‌ها به برنامه بسیار ساده‌تر خواند بود و در صورت بروز مشکل، عیب‌یابی و رفع آن ساده‌تر از حالتی است که سیستم به صورت یکپارچه پیاده‌سازی شده باشد. حال سامانه‌ی پیاده‌سازی شده برای این پژوهه به شکل زیر می‌باشد.



شکل ۴-۵ مایکروسرویس‌های تشکیل‌دهنده‌ی سامانه

همانطور که در تصویر نیز پیداست، سامانه‌ای که این پروژه بر روی آن سوار می‌باشد متتشکل از چهار جز اصلی می‌باشد.

- سرور اصلی جنگو که این سرور که در واقع مرکز اصلی سیستم می‌باشد، وظیفه‌ی برقراری ارتباط میان سرورهای کوچک‌تر فلسف و مدیریت آن‌ها را بر عهده دارد. همچنین ظاهر برنامه بر روی این سرور پیاده‌سازی شده است.
- یک سرور فلسف برای ایجاد مدل مولد گلو و استفاده از آن.
- یک سرور فلسف برای ایجاد مدل مولد تقابلی سایکل‌گن و استفاده از آن.
- یک سرور فلسف برای ایجاد مدل رگرسیونی تخمین سن و استفاده از آن.

۱-۲-۲-۵ مشکلات معماری مایکروسرویس

البته این نکته را نیز باید خاطر نشان کرد که این سیستم نیز بدون مشکل نمی‌باشد. از مشکلات این سیستم می‌توان به اضافه کردن پیچیدگی به برنامه اشاره کرد. زیرا در سیستم مایکروسرویس، برای ایجاد ارتباط میان سرویس‌ها باید نقاط انتهایی را طراحی و پیاده‌سازی کرد و از کار کرد آن‌ها مطمئن شد که این موضوع خود می‌توان مشکل‌ساز باشد. به طور خلاصه می‌توان گفت اگر برنامه‌ی مورد نظر برنامه‌ی سیک و ساده‌ای می‌باشد و در نیازمندی‌هایش تضاد نمی‌باشد بهتر است تا به صورت یکپارچه پیاده‌سازی شود و درگیر پیچیدگی‌های سیستم مایکروسرویس نشد. اما اگر برنامه مانند این پروژه دارای پیچیدگی باشد، آن وقت بهتر است تا سراغ معماری مایکروسرویس برویم. حال در ادامه به توضیح چارچوب جنگو که مرکز این سیستم می‌باشد می‌رویم. بعد از آن به توضیح این موضوع می‌پردازیم که هر کدام از این سرویس‌ها چگونه پیاده‌سازی و هماهنگ شده‌اند.

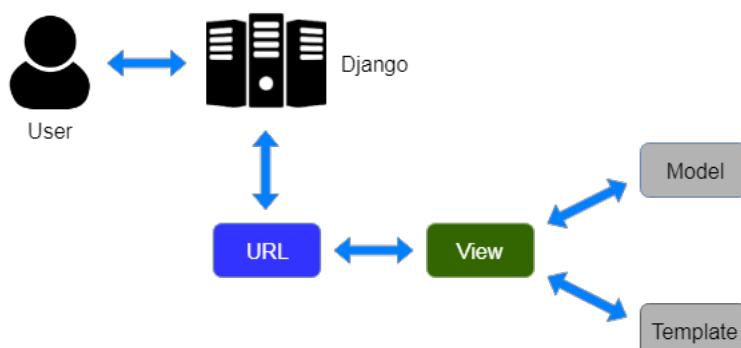
۳-۵ جنگو

۱-۳-۵ مقدمه

در این بخش به توضیح نسبتاً کاملی از چارچوب جنگو خواهیم پرداخت. ابتدا ساختار کلی برنامه‌های تحت توسعه‌ی جنگو را بررسی می‌کنیم و سپس بخش‌های مختلف آن را توضیح می‌دهیم. منبع اکثر این آموزش‌ها مستندات جنگو می‌باشد. [۱۹]

۲-۳ ساختار برنامه‌ها در جنگو

جنگو دارای معماری MTV^۱ می‌باشد که مشابه معماری پرکاربرد MVC است با این تفاوت که بخش تمپلیت^۲ به ویو^۳ و بخش ویو به کنترلر^۴ تبدیل شده است. این ساختار بدین صورت می‌باشد که بعد از اینکه کاربر یک درخواست^۵ را به سرور^۶ ارسال می‌کند، سرور با استفاده از توزیع کننده‌ی URL^۷ درخواست را به ویو مرتبط با آن ارسال می‌کند. در نهایت ویو به کمک لایه‌ی مدل^۸ اطلاعات مورد نیاز را استخراج کرده و سپس با استفاده از لایه‌ی تمپلیت آن را به کاربر نشان می‌دهد.



شکل ۵ معماری ساختار MVT

¹Model Template View

²Template

³View

⁴Controller

⁵Request

⁶Server

⁷Uniform Resource Locator

⁸Model

۳-۳-۵ ساختار فایل‌های اولیه‌ی پروژه جنگو

ساختار فایل‌های اولیه‌ی جنگو به صورت زیر می‌باشد که به اختصار هر کدام را توضیح می‌دهیم.



شکل ۵-۶ ساختار فایل‌های اولیه‌ی پروژه جنگو

- `__init__.py`: این فایل پوشه‌ای که شاملش می‌شود را به یک پکیج پایتون^۱ تبدیل می‌کند که در این صورت می‌توان به اطلاعات فایل‌هایی که درون آن پوشیده قرار دارند دسترسی داشت (از طریق ایمپورت^۲ کردن).
- `manage.py`: این فایل پل ارتباطی میان میان توسعه‌دهنده و پروژه‌ی جنگوی او می‌باشد. در واقع ابزار خط فرمانی^۳ می‌باشد که جنگو در اختیار توسعه دهنده‌گان قرار داده است.
- `settings.py`: بسیاری از تنظیمات پروژه مانند تنظیمات مربوط به احراز هویت کاربران^۴، اپ‌های نصب شده‌ی برنامه، تنظیمات مربوط به سیستم ایمیل، برخی تنظیمات مربوط به پایگاهداده و تنظیمات دیگر در این فایل قرار دارند.
- `urls.py`: در این فایل آدرس‌هایی که قابل دسترسی خواهند بود، قرار می‌گیرند.
- `wsgi.py` و `asgi.py`: این دو فایل مربوط به وب‌سرورهایی^۵ هستند که جنگو می‌تواند منطبق بر آنها کار کند.

¹Python Package

²Import

³Command Line

⁴User Authentication

⁵Web Servers

۴-۳-۵ مدل

مدل‌ها در جنگو برای ذخیره‌ی اشیاء در پایگاه داده مورد استفاده قرار می‌گیرند. مدل‌های جنگو کلاس‌های پایتون هستند که توسط جنگو یکسری ویژگی‌های اضافه به آن‌ها اضافه شده‌است. حال برای استفاده از این مدل‌ها در جنگو باید models را از django.db имپورت کرد و همچنین کلاس مدلی که می‌خواهیم پیاده‌سازی کنیم باید از models.Model ارث بری کند. سپس ویژگی‌های مربوط به آن مدل را در خطوط جداگانه می‌آوریم بدین صورت که ابتدا نام آن ویژگی سپس نوع فیلد^۱ آن ویژگی و سپس پارامترهای مربوط به آن ویژگی. همچنین جنگو علاوه بر این فیلدها یک فیلد آیدی^۲ که منحصر بفرد می‌باشد برای هر شئ در پایگاه‌داده می‌سازد. همچنین مدل‌های جنگو چون کلاس‌های پایتون هستند این قابلیت را دارند تا درون آن‌ها تابع^۳ تعریف شود. در مدل‌های جنگو می‌توان روابط^۴ بین آن‌ها را نیز ایجاد کرد.

- رابطه‌ی یک به یک^۵: این رابطه برای زمانی است که بین دو مدل رابطه‌ی یک به یک وجود دارد. به عنوان مثال اگر سیستمی داشته باشیم که در آن هر نویسنده تنها اجازه‌ی نوشتن یک کتاب را داشته باشد آنگاه رابطه‌ی بین نویسنده و کتاب یک به یک می‌باشد.
- رابطه‌ی یک به چند (با چند به یک)^۶: این رابطه برای زمانی است که بین دو مدل رابطه‌ی یک به چند (یا چند به یک) وجود داشته باشد. به عنوان مثال اگر در سیستم ما هر نویسنده اجازه‌ی نوشتن چندین کتاب را داشته باشد آنگاه رابطه‌ی بین نویسنده و کتاب یک رابطه‌ی یک به چند می‌باشد.
- رابطه‌ی چند به چند^۷: این رابطه برای زمانی است که بین دو مدل رابطه‌ی چند به چند وجود دارد. به عنوان مثال اگر سیستمی داشته باشیم که در آن نویسنده‌ها اجازه‌ی نوشتن چند کتاب را داشته باشد و همچنین چند نویسنده بتوانند باهم یک کتاب بنویسند. آنگاه رابطه‌ی بین نویسنده و کتاب یک رابطه‌ی چند به چند می‌باشد.

¹Field

²ID

³Function

⁴Relations

⁵One-to-One

⁶One-to-Many (Many-to-One)

⁷Many-to-Many

۵-۳-۵ ویو

در فایل views.py کنترل جریان درخواست‌ها اتفاق می‌افتد. در واقع برای متصل کردن منطق برنامه یا همان کدهایی که برای کنترل درخواست‌ها و همچنین دریافت و ارسال اطلاعات میان کلاینت^۱ و سور نوشته شده است به مسیرهایی که قابل دسترس هستند از URL‌ها استفاده می‌کنیم. حال اگر برنامه بزرگ باشد بهتر است که در هر اپ URL‌های مربوط به آن آورده شود و سپس در فایل urls.py اصلی، شامل^۲ شوند. بدین صورت می‌توان خوانایی کد را بالا برد و از پیچیدگی آن کم کرد. این URL‌ها توانایی پویا شدن نیز دارند بدین صورت که با URL بتوان یک متغیر از کاربر دریافت کرد. این کار با کمک تابع path صورت می‌گیرد. این متغیر (یا متغیرها) که در تابع path آورده می‌شود با کمک مبدل‌های URL به ویو صدا زده شده در این تابع پاس داده می‌شود و در آن جا عملیات‌های مورد نیاز با استفاده از آن متغیر انجام می‌شود. در مرحله‌ی آخر با کمک توابعی که در فایل views.py نوشته شده‌اند خروجی مورد نظر به کاربر نمایش داده می‌شود.

۶-۳-۵ نگاشت رابطه‌ای شیئی

^۳ ORM یا نگاشت رابطه‌ای شیئی بخشی از جنگو می‌باشد که وظیفه‌ی اتصال به پایگاهداده و دریافت و ارسال اطلاعات به آن را دارد. با نگاشت رابطه‌ای شیئی دیگر نیازی به نوشتن دستورات مختلف برای پایگاهداده‌های مختلف نداریم و فقط یکبار با نوشتن دستورات قادر خواهیم بود تا تنها با عوض کردن تنظیمات کوچکی در فایل تنظیمات جنگو، دستورات را در پایگاهداده‌های متفاوت اجرا کنیم. در واقع نگاشت رابطه‌ای شیئی نوعی واسطه میان کد نوشته شده به زبان پایتون و کوئری‌های پایگاهداده‌های متفاوت می‌باشد. نگاشت رابطه‌ای شیئی کدهای نوشته شده به زبان پایتون را به کوئری‌های پایگاهداده‌ای تبدیل می‌کند که به این کار مپ کردن^۴ گفته می‌شود.

حال برای کار با پایگاهداده نیازمند ساختماندادهایی^۵ هستیم که بتوان به راحتی داده‌ها را در خود ذخیره کرده و تغییرات مورد نیار را روی آن‌ها اعمال کنند. بدین منظور در نگاشت رابطه‌ای شیئی از ساختمانداده‌ی کوئری است^۶ از استفاده می‌کنیم که شباهت زیادی با لیست^۷ پایتون دارد. کوئری است یکسری شروط نیز در ورودی خود می‌گیرد و به عنوان خروجی دوباره یک کوئری است بر می‌گردد که این ویژگی باعث می‌شود تا بتوان از آن به صورت زنجیره‌ای^۸ استفاده کرد و مجموعه‌ای از شروط را به

¹Client

²Include

³Object-Relational Mapping

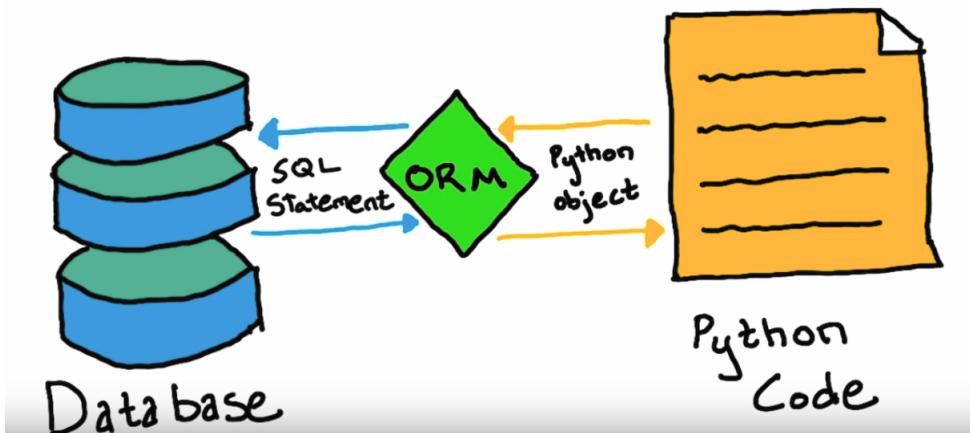
⁴Mapping

⁵Data Structures

⁶QuerySet

⁷List

⁸Chain



شکل ۷-۵ شمایی از نگاشت رابطه‌ای شیئی

روی یک کوئری سمت اعمال کرد.

۷-۳-۵ تمپلیت

تمپلیت سیستمی است جهت تولید صفحات اجتنی امال^۱ پویا. با کمک تمپلیت می‌توان داده‌های پویا را وارد صفحات اجتنی امال کرد و موتور جنگو تمپلیت برای جدا کردن طراحی بکاند^۲ و فرانتاند^۳ می‌باشد و با آن می‌توان منطق را وارد صفحات اجتنی امال کرد. از این طریق می‌توان توسعه پذیری برنامه را افزایش داد و از تکرار و افزونگی کد جلوگیری به عمل آورد. در فرآیند به وجود آوردن این صفحات پویا علاوه بر صفحات اجتنی امال به فایل‌های دیگر مانند سی‌اس‌اس^۴ و جاوااسکریپت^۵ و برخی دیگر نیز نیاز داریم (که به این فایل‌ها فایل‌های static می‌گوییم) که طریقه‌ی استفاده از این فایل‌ها بدین صورت می‌باشد که باید آدرس‌هایی که این فایل‌ها در آن‌ها قرار دارند را به لیستی با نام STATICFILES_DIRS در فایل تنظیمات جنگو اضافه کنیم. حال برای برقراری ارتباط میان ویو و صفحات اجتنی امال و دیگر فایل‌های استاتیک از تابع render استفاده می‌کنیم و مقادیری که می‌خواهیم به این فایل‌ها و صفحات پاس بدهیم به آرگومان context که یک لغتنامه می‌گیرد می‌دهیم. سپس برای استفاده از این متغیرها در این صفحات اجتنی امال می‌توان با قرار دادن آن‌ها میان بهره برد.

همچنین در تمپلیت قابلیت‌هایی وجود دارد همچون برچسب‌ها که با آن‌ها می‌توان منطق و جریان برنامه را کنترل کرد. به عنوان مثال می‌توان با برچسب { %if% } { %end% } شرط درست کرد و یا با

¹HTML

²Back-End

³Front-End

⁴ CSS

⁵ JavaScript

برچسب `{%for obj in list%}{%endfor%}` حلقه ایجاد کرد درست مانند یک زبان برنامه‌نویسی. از قابلیت‌های خوب دیگر تمپلیت ارثبری می‌باشد که به حذف کد تکراری کمک می‌کند. با این قابلیت می‌توان ساختار کلی صفحات که تکرارشونده هستند را به صورت بک قالب ساخت و در بخش‌های دیگر برنامه و تمپلیت‌های دیگر استفاده کرد. با استفاده از برچسب `extends` در اول بک کد نشان می‌دهیم که می‌خواهیم از کدام کلاس ارثبری کنیم.

۸-۳-۵ فرم^۱

فرم‌ها ابزاری هستند که توسط آن‌ها می‌توان تبادل اطلاعات از سمت کاربر به سمت سرور انجام داد. فرم‌ها بسیار پرکاربرد می‌باشند به عنوان مثال می‌توان به یک فرم ثبت‌نام در یک سرویس اشاره کرد که در کد زیر یک نمونه‌ی اولیه از آن را می‌بینید.

```
1 from django import forms
2
3 class SignUpForm(forms.Form):
4     username = forms.CharField()
5     password = forms.CharField()
6     email = forms.EmailField(required=False)
```

شکل ۸-۵ نمونه‌ای از فرم ثبت‌نام

در کد شکل ۸-۵ `required` فیلد را که برابر با ناصحیح^۲ قرار دارد نشان‌دهنده‌ی آن می‌باشد که پرکردن این فیلد (که در اینجا ایمیل می‌باشد) توسط کاربر الزامی نمی‌باشد و این متغیر در حالت عادی مقدار صحیح^۳ دارد.

بحث مهم دیگری که جنگو به خوبی به آن پرداخته است اعتبارسنجی فرم‌ها می‌باشد. در جنگو می‌توان با استفاده از تابع `is_valid()` فرم‌ها را اعتبارسنجی کرد که اگر فرم به درستی گر شده باشد خروجی این تابع صحیح می‌باشد و در غیر این صورت ناصحیح. همچنین بعد از صدا زده شدن این تابع، اطلاعات فرم در یک لغتنامه به نام `cleaned_data` ذخیره می‌شود که کلیدهای این لغتنامه فیلد‌های فرم و مقادیر این لغتنامه مقادیر فیلدها می‌باشند و می‌توان از این اطلاعات با استفاده از این لغتنامه استفاده کرد. در صورتی که فرم به درستی پر نشده باشد اطلاعات خطاهای فرم در یک لغتنامه به نام `errors` ذخیره می‌شود. همچنین می‌توان اعتبارسنجی‌های سفارشی ایجاد کرد.

¹Form

²False

³True

ویژگی دیگری که در رابطه با فرم‌ها وجود دارد مدل فرم‌ها^۱ می‌باشد. مدل فرم‌ها برای زمانی هستند که ما یک مدل در برنامه خود داریم و می‌خواهیم که از این مدل یک فرم طراحی کنیم. با استفاده از مدل فرم‌ها دیگر نیازی نیست تا به صورت دستی تمام ویژگی‌های مدل را در فرم آورد و فقط کافیست تا یک مدل فرم بسازیم و مشخص کنیم که کدام ویژگی‌ها را از آن مدل نیاز داریم. فرض کنید مدلی داریم که دارای ویژگی‌های نام‌کاربری، رمز عبور و ایمیل می‌باشد. حال با کمک یک مدل فرم می‌توانیم فرم این مدل را مانند کد شکل ۹-۵ ایجاد کنیم.

```

1  from django import forms
2  from .models import User
3
4  class SignUpForm(forms.ModelForm):
5
6      class Meta:
7          model = User
8          fields = ('username', 'password', 'email',)

```

شکل ۹-۵ نمونه‌ای از مدل فرم

۹-۳-۵ پنل ادمین^۲

اپ ادمین جنگو ابزاری قدرتمند است جهت ساخت، بروزرسانی، مشاهده و حذف داده‌ها و مدیریت سایتها توسط یک رابط کاربری ساده و کاربردی است. همچنین جنگو این قابلیت را دارد تا بتوان یک کاربر همه‌کاره^۳ ساخت که این کاربر اجازه دسترسی به تمام اطلاعات درون پنل ادمین را دارد.

پنل ادمین جنگو بسیار قابلیت شخصی‌سازی شدن دارد که در زیر به چند مورد پرکاربرد آن اشاره می‌کنیم.

- قابلیت `list_display`: با این قابلیت می‌توان مشخص کرد که کدام فیلدها از مدل‌مان نمایش داده شود.
- قابلیت `sortable_by`: با این ویژگی می‌توانید یک لیست تعریف کنید سپس صفحه‌ی مربوط به اشیاء یک مدل را براساس ویژگی‌های آن لیست به صورت صعودی یا نزولی مرتب کنید.
- قابلیت `list_filter`: با این قابلیت می‌توانید داده‌هایی از یک مدل را که دارای یک ویژگی خاص می‌باشند مشاهده کنید.

¹Model Forms

²Admin Panel

³Super user

- قابلیت `list_editable`: با افزودن هر فیلدی به این لیست می‌توان بدون باز کردن یک مدل در پنل ادمین، آن فیلد را ویرایش کرد.
- قابلیت `search_fields`: این قابلیت امکان جستجو بر اساس فیلدهایی که در لیستش قرار می‌گیرند را به ادمین می‌دهد.
- قابلیت `inlines`: این قابلیت برای نشان دادن کلید خارجی^۱ در روابط میان مدل‌ها می‌باشد مه به صورت عادی نشان داده نمی‌شود.
- قابلیت `readonly_fields`: فیلدهایی که در این لیست قرار می‌گیرند دیگر قابلیت ویرایش شدن توسط پنل ادمین را ندارند.

۱۰-۳-۵ جنگو رست^۲

رابط برنامه‌نویسی برنامه یا ای‌پی‌آی (API^۳) مجموعه‌ای از اعمال و ابزارها می‌باشد که توسعه‌دهندگان با کمک آن‌ها، بدون دانستن نحوه‌ی دقیق پیاده‌سازی یک سرویس از آن سرویس استفاده کنند. در واقع در تیم‌های برنامه‌نویسی ای‌پی‌آی برای ارتباط میان تیم فراتر از بک‌اند به این ای‌وی‌آی‌ها نیاز داریم تا توسعه‌دهندگان هر تیم بتوانند بدون آنکه از پیاده‌سازی‌های انجام شده در تیم دیگر اطلاعی داشته باشند به صورت مستقل به توسعه‌ی سمت خود بپردازنند و با سمت دیگر ارتباط داشته باشند و این ارتباط از سرعت و کیفیت توسعه‌ی برنامه نکاهد. حال یک ای‌پی‌آی پرکاربرد و معروف رست می‌باشد که به زبان جیسون (json^۴) می‌باشد.

حال برای استفاده از این ای‌پی‌آی نیاز به یکسری ابزارهای جانبی داریم زیرا در بسیاری از بخش‌ها باید اطلاعات را به فرم جیسون در بیاوریم و منتقل بکنیم. به عنوان مثال نیاز به سریالایزر^۵ داریم تا اشیاء و داده‌ها را به جیسون تبدیل کند و برعکس. همچنین به ویوهای ای‌پی‌آی نیاز داریم که توانایی برقراری ارتباط با جیسون داشته باشند و اشیاء درخواست و جواب نیز باید از جیسون پشتیبانی کنند. برای انجام این کارها از کتابخانه‌ی Django REST framework استفاده می‌کنیم.

احراز هویت کاربران از قابلیت‌هایی است که توسط ای‌پی‌آی‌های رست به خوبی و آسانی قابل مدیریت کردن می‌باشد. رست بدین منظور از توکن^۶ استفاده می‌کند بدین صورت که هر کاربر پس از وارد

¹Foreign Key

²Django Rest

³Application Programming Interface

⁴Javascript Object Notation

⁵Serializer

⁶Token

کردن اطلاعات خود (نام کاربری و رمز عبورش) و درست بودن اطلاعاتش یک توکن دریافت می‌کند که منحصر بفرد می‌باشد و پس از آن هر درخواستی که کاربر به سرور ارسال کند باید این توکن را در بخش هدر^۱ درخواست قرار دهد تا سرور هویت آن کاربر را همواره به درستی تشخیص دهد و دسترسی‌های لازم برای آن کاربر را به او بدهد. سپس برای استفاده از این قابلیت‌ها در ویو کافیست تا لیستی از این اجازه‌ها^۲ به کاربر بدهیم مانند `IsAuthenticated`.

۱۱-۳ میان‌افزار^۳

میان‌افزار یا میدلور یک لایه‌ی میانی میان کاربر و هسته‌ی مرکزی برنامه یعنی ویو می‌باشد. میان‌افزارها کلاس‌های پایتونی می‌باشند که در میان چرخه‌ی ریکوئست و ریسپانس قرار دارند. درواقع کار این میان‌افزارها انجام یک سری پردازش‌ها روی درخواست، قبل از رسیدن درخواست به ویو و یکسری پردازش‌ها روی جواب، قبل از رسیدن جواب به کاربر می‌باشد. در شکل ۱۰-۵ میان‌افزارهای پیش‌فرض جنگو را می‌بینیم که به صورت کوتاه به توضیح برخی از آن‌ها می‌پردازیم. باید توجه کرد که ترتیب قرار گرفتن این میان‌افزارها در فایل تنظیمات جنگو به همین صورت از بالا به پایین می‌باشد و در صورتی که قصد اضافه کردن میان‌افزاری را داشته باشیم باید به ترتیب قرار گرفتن آن در فایل تنظیمات توجه کرد.
حال به توضیح برخی از این میان‌افزارها می‌پردازیم

- CommonMiddleware: این میان‌افزار تعدادی کار ساده‌ی روتین انجام می‌دهد از جمله بازنویسی دامنه‌ها (اضافه کردن `www` به ابتدای آن‌ها و بکسلش به انتهای آن‌ها و بررسی معتبر بودن یا نبودن آدرس‌ها).

- SecurityMiddleware: این میان‌افزار چرخه‌ی درخواست و جواب را از جنبه‌ی امنیتی بررسی می‌کند و آن را بهبود می‌بخشد.

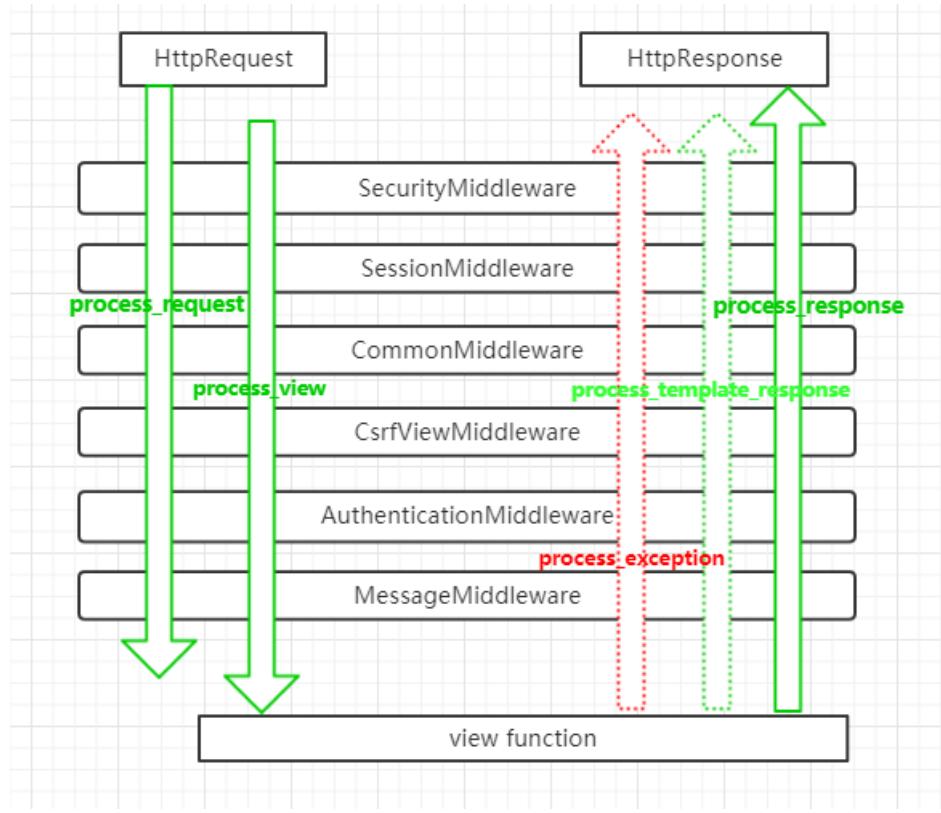
- SessionMiddleware: این میان‌افزار قابلیت استفاده از نشست^۴ را به ما می‌دهد که با آن می‌توان داده‌های شخصی‌سازی شده‌ی کاربران را نگهداری کرد که با این کار می‌توان کارهای بسیار زیاد مفیدی انجام داد از جمله احراز هویت کاربران و بسیاری کار دیگر.

¹Header

²Permissions

³Middleware

⁴session



شکل ۵-۱۰ ترتیب قرارگیری میان افزارها میان کاربر و ویو

۴-۵ تشریح سرویس‌ها

همانطور که پیش‌تر توضیح داده شد سامانه‌ی ما متشکل از چهار سرویس مجزا می‌باشد که در مرکز آن سرور جنگو قرار دارد و سه سرویس فلسفک را مدیریت می‌کند. حال در این بخش می‌خواهیم توضیح بدھیم که پیاده‌سازی این سرویس‌ها به چه صورت است و چگونه هر کدام از این سرویس‌ها جداگانه اجرا می‌شوند و سپس با یکدیگر ارتباط می‌گیرند.

۱-۴-۵ داکر

داکر نوعی پلتفرم متن باز می‌باشد که بر مبنای سیستم‌عامل لینوکس است و با آن می‌توان فرآیندهای ایجاد، پیاده‌سازی و اجرای برنامه‌ها را با کمک مفهومی به نام کانتینر^۱ انجام داد. کانتینرها محیط‌های اجرایی را جدا کرده به طوری که برنامه‌ها در محیطی ایزوله و مستقل از یکدیگر اجرا شوند مانند ماشین‌های مجازی^۲ اما با این تفاوت که کانتینرها بسیار سبک‌تر و سریع‌تر هستند و از منابع کمتری استفاده می‌کنند. با کمک کانتینرها توسعه‌دهنده بسته‌ی کاملی از برنامه‌های مورد نیاز برای اجرای

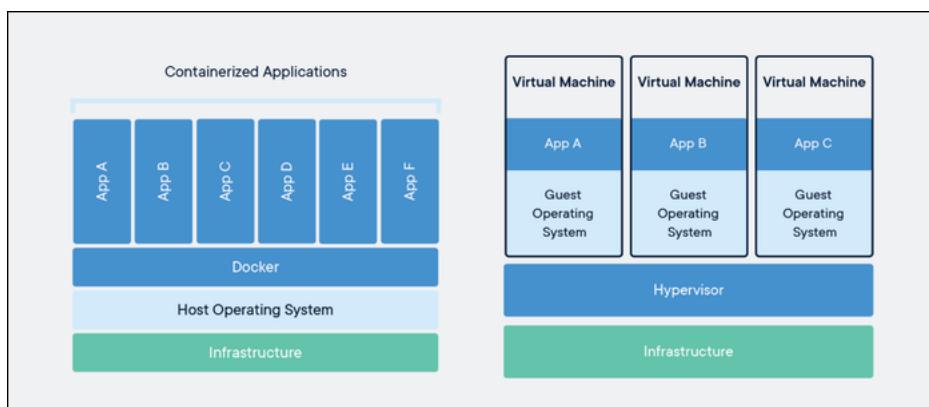
¹Container

²Virtual Machines

برنامه‌ی خود را به صورت یک تیکه در اختیار دارد و این بسته را می‌توان در هر ماشین دیگری با هر سیستم‌عاملی بدون مشکل اجرا کند.

برای استفاده از داکر نیاز است تا یک داکرفایل^۱ برای برنامه نوشته شود که در واقع همان تنظیمات برنامه می‌باشد. سپس این داکرفایل باید بیلد^۲ شود تا اینکه یک ایمیج^۳ از آن داکرفایل درست شود. حال در هر جایی که نیاز است تا این برنامه اجرا شود کافیست این ایمیج را اجرا کنیم. این ایمیج‌ها را نیز می‌توان در داکرهاب^۴ که یک مخزن برای مدیریت و به اشتراک‌گذاری کانتینرها است قرار داد و یا اینکه از آن دانلود کرد.

داکر در واقع یک لایه‌ی واسطه میان سیستم‌عامل و بسته‌ی نرم‌افزاری ایجاد می‌کند که با کمک آن نرم‌افزارها را از یکدیگر تفکیک کرده و هر نرم‌افزار را در محیطی ایزوله قرار می‌دهد. در شکل ۱۱-۵ نیز می‌توانید تفاوت ساختاری میان داکر و ماشین‌مجازی را ببینید.



شکل ۱۱-۵ تفاوت ساختاری میان داکر و ماشین‌مجازی

۲-۴-۵ پیاده‌سازی سرویس‌ها

حال با توجه به توضیحات داکر این موضوع را درمی‌یابیم که این تکنولوژی دقیقاً همان چیزی است که ما برای پیاده‌سازی سرویس‌هایمان در این پروژه نیاز داریم بدین صورت که برای هر سرویس نیاز است تا یک داکرفایل نوشته شود تا سپس کانتینر مورد نظر از آن سرویس ایجاد شده و به اجرا در بیاید. سپس این کانتینرها به کمک داکر کامپوز^۵ به یکدیگر متصل شده و تحت عنوان یک برنامه واحد اجرا می‌شوند.

¹Dockerfile

²Build

³Image

⁴Docker Hub

⁵Docker Compose

حال در ادامه به توضیح مختصری از داکرفایل‌های سرویس‌ها می‌پردازیم.

۱-۲-۴-۵ سرویس سرور اصلی

در تصویر ۱۲-۵ داکرفایل مربوط به سرویس اصلی یعنی سرور جنگو قابل مشاهده است. در این سرویس از تصویر پایتون نسخه ۱۰ به عنوان تصویر اصلی کانیتر استفاده می‌شود و درواقع تمام نیازمندی‌های بعدی بر روی این تصویر نصب و اجرا می‌شود. سپس یک پورت مشخص را برای آن باز می‌کنیم تا برنامه بتواند از طریق این پورت با سرویس‌های دیگر در ارتباط باشد. سپس محلی که برنامه در کانتینر باید قرار بگیر و اجرا شود را مشخص می‌کنیم و بعد از آن نیازمندی‌های برنامه را در کانتینر کپی کرده و نصب می‌کنیم. در انتها نیز برنامه را با دستور مورد نیاز جنگو برای اجرای سرور اجرا می‌کنیم.

```
1 # Use official Python runtime as a parent image
2 FROM python:3.10-slim-buster
3
4 ENV PYTHONUNBUFFERED=1
5 ENV PYTHONDONTWRITEBYTECODE=1
6
7 EXPOSE 8000
8
9 # Set the working directory in the container
10 WORKDIR /app
11
12 # Copy the requirements file to the container
13 COPY requirements.txt .
14
15 # Install the dependencies from the requirements file
16 RUN pip install -r requirements.txt
17
18 # Copy the rest of the application code to the container
19 COPY . .
20
21 # Start the Django server
22 CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]
23
```

شکل ۱۲-۵ داکرفایل سرویس اصلی

۲-۲-۴-۵ سرویس مدل گلو

در تصویر ۱۳-۵ داکرفایل مربوط به سرویس مدل هوش‌مصنوعی گلو قابل مشاهده است. در این سرویس یک نسخه‌ی مشخصی از تنسورفلو (نسخه ۰.۸.۱) که برنامه تنها با همین نسخه قابل اجرا است، را به عنوان تصویر پایه قرار می‌دهیم و سایر نیازمندی‌ها را بر روی آن سوار می‌کنیم. از آنجایی که این تصویر یک تصویر خام نمی‌باشد (یعنی تنها دارای کتابخانه‌ی تنسورفلو نیست) کتابخانه‌های دیگری نیز بر روی

آن نصب شده است. در نتیجه برای به مشکل نخوردن برای اجرای این سرویس هنکام نصب برخی دیگر از نیازمندی‌ها از برچسب ignore-installed استفاده شده است. در نهایت نیز محلی که برنامه در کانتینر باید قرار بگیر و اجرا شود را مشخص می‌کنیم و تمام فایل‌های برنامه را در آن کپی می‌کنیم و سپس سرویس را اجرا می‌کنیم.

```

1  FROM tensorflow/tensorflow:1.8.0-py3
2
3  RUN apt update && \
4      apt install -y locales cmake libsm6 libxext6 libxrender-dev && \
5      locale-gen en_US.UTF-8
6
7  ENV LC_ALL=en_US.UTF-8
8
9  RUN pip install pip==20.3.4
10
11 RUN pip install --ignore-installed tqdm
12 RUN pip install numpy==1.18.5
13 RUN pip install --ignore-installed imutils==0.5.3
14 RUN pip install --ignore-installed opencv-python==4.3.0.36 --no-build-isolation
15 RUN pip install --ignore-installed dlib==19.21.0
16 RUN pip install flask flask_cors
17
18 WORKDIR /app
19
20 COPY ./ ./
21
22 CMD ["python", "server.py"]

```

شکل ۱۳-۵ داکرفایل سرویس مدل گلو

۳-۲-۴-۵ سرویس مدل سایکل گن

در تصویر ۱۴-۵ داکرفایل مربوط به سرویس مدل هوش مصنوعی تقابلی سایکل گن قابل مشاهده است. در این سرویس از تصویر پایتون نسخه‌ی ۹ به عنوان تصویر اصلی کانیتر استفاده شده است. در این داکرفایل نیز همانند قبل محلی که برنامه در کانتینر باید قرار بگیر و اجرا شود را مشخص می‌کنیم و بعد از آن نیازمندی‌های برنامه را در کانتینر کپی کرده و نصب می‌کنیم. اما در این سرویس نیاز است تا چند کار اضافه‌تر نیز انجام دهیم و آن ایجاد چند مسیر در برنامه و سپس کپی کردن چند فایل در مسیرهای مورد نظر جهت اجرای صحیح برنامه می‌باشد. و در نهایت اجرای این سرویس.

```

1 # Use official Python runtime as a parent image
2 FROM python:3.9
3
4 # Set the working directory in the container
5 WORKDIR /app
6
7 # Upgrade pip
8 RUN pip install pip==23.1.2
9
10 # Copy the requirements file to the container
11 COPY requirements.txt .
12
13 # Install the dependencies from the requirements file
14 RUN pip install -r requirements.txt --no-cache-dir
15
16 # Install Flask
17 RUN pip install flask
18
19 # Copy the rest of the application code to the container
20 COPY ..
21
22 # Create following directories
23 RUN mkdir -p ./datasets/y2o/testA/
24 RUN mkdir -p ./datasets/y2o/testB/
25 RUN mkdir -p ./checkpoints/aging_cyclegan/
26
27 # Copy the pre-trained model
28 # RUN cp ./trained_model/1_wiki_base_mix/latest_net_G_A.pth ./checkpoints/aging_cyclegan/latest_net_G.pth
29
30 RUN cp ./trained_model/1_wiki_base_mix/latest_net_G_A.pth ./checkpoints/aging_cyclegan/latest_net_G_A.pth
31 RUN cp ./trained_model/1_wiki_base_mix/latest_net_G_B.pth ./checkpoints/aging_cyclegan/latest_net_G_B.pth
32
33 # Run the flask server
34 CMD ["python", "server.py"]

```

شکل ۱۴-۵ داکلر فایل سرویس مدل سایکل گن

۴-۲-۴-۵ سرویس مدل تخمین سن

در تصویر ۱۵-۵ آخرین داکرفایلی که برای پروژه استفاده شده است که مربوط به سرویس مدل هوش مصنوعی رگرسیونی تخمین سن افراد است قابل مشاهده است. این مدل نیاز به پایتون نسخه ۹.۶.۳ و کتابخانه تنسورفلو نسخه ۳.۲ برای اجرا دارد.

حال این سرویس‌ها برای اجرا شدن به صورت همزمان، متصل شدن به یکدیگر از طریق یک شبکه‌ی مشخص و انجام سایر تنظیماتشان نیاز دارند تا همانطور که اشاره شد از طریق داکر کامپوز اجرا شوند.

۵-۵ داکر کامپوز

داکر کامپوز ابزاری است که برای تعریف و مدیریت برنامه‌های دارای چندین کانتینر مانند این پروژه در داکر استفاده می‌شود. این ابزار به توسعه‌دهنده امکان می‌دهد تا مجموعه‌ای از سرویس‌ها، وابستگی‌های

```

1  # Use official Python runtime as a parent image
2  FROM python:3.6.9
3
4  # Set the working directory in the container
5  WORKDIR /app
6
7  # Upgrade pip
8  # RUN pip install pip==23.1.2
9
10 # Install tensorflow
11 RUN pip install tensorflow==2.3.0
12
13 # Copy the requirements file to the container
14 COPY requirements.txt .
15
16 # Install the dependencies from the requirements file
17 RUN pip install -r requirements.txt --no-cache-dir
18
19 # Install Flask
20 RUN pip install flask
21
22 # Copy the rest of the application code to the container
23 COPY . .
24
25 # Run the flask server
26 CMD ["python", "server.py"]

```

شکل ۵-۵ داکلر فایل سرویس مدل تخمین سن

آن‌ها، تنظیمات شبکه و سایر تنظیمات مرتبط را در یک فایل یمل واحد تعریف کند. با استفاده از داکر کامپوز شما می‌توانید به راحتی چندین کانتینر داکر را به عنوان یک برنامه یکپارچه اجرا و مدیریت کنید.

فاصله یمل که به طور معمول docker-compose.yml نامگذاری می‌شود، مشخصات مختلف اجزای برنامه شما را توصیف می‌کند، مانند سرورهای وب، پایگاهداده‌ها و سایر سرویس‌ها و تنظیمات مربوطه آن‌ها. هر سرویس تعریف شده در فایل نشان‌دهنده‌ی یک کانتینر است که به صورت مستقل قابل اجرا است، اما می‌تواند با سرویس‌های دیگر در ارتباط و تعامل باشد. داکر کامپوز فرایند تنظیم و مدیریت ساختارهای پیچیده برنامه را با ارائه یک رابط یکپارچه ساده‌تر می‌کند. با استفاده از داکر کامپوز، می‌توانید برنامه‌های چند کانتینری را با چند دستور ساده تعریف و راهاندازی کنید، که امکان نگهداری و مقیاس‌پذیری محیط برنامه‌تان را آسان‌تر می‌کند. پس از ایجاد این فایل، برنامه به سادگی با اجرای تنها دستور Docker-compose up قابل اجرا می‌باشد.

۶-۵ رابط کاربری گرافیکی

در رابط کاربری گرافیکی^۱ این سامانه از فناوری‌های پایه‌ای اچ‌تی‌ام‌ال^۲ و سی‌اس‌اس^۳ و کمی‌جاوا‌اسکریپت^۴ استفاده شده است. همانطور که در بخش چارچوب جنگو توضیح دادیم بخش ویو^۵ در جنگو محل ارتباط با صفحات مربوط به رابط کاربری گرافیکی می‌باشد. بدین صورت که درخواست کاربر به همراه اطلاعات مورد نیاز از صفحه‌ی مورد نظر گرفته می‌شود و به توابع مختلف در ویو جنگو داده می‌شود. سپس این اطلاعات پردازش می‌شود و جواب نهایی یا اطلاعات نهایی دوباره به صفحه‌ی مورد نظر داده می‌شود. این سامانه دارای دو صفحه برای رابط کاربری گرافیکی می‌باشد.

- صفحه‌ی اول: این صفحه وظیفه‌ی دریافت اطلاعات توسط کاربر و سپس انتقال دادن اطلاعات به سرور جنگو را دارد. اطلاعاتی که این صفحه از کاربر می‌گیرد شامل یک عکس که عملیات پیر کردن یا جوان کردن چهره بر روی این عکس انجام می‌شود، مشخص کردن گزینه‌ی پیر کردن یا جوان کردن، مشخص کردن مقدار آلفا (بین عدد ۱ و ۲) که این بخش برای استفاده‌ی راحت‌تر کاربر به صورت یک اسلایدر^۶ طراحی شده‌است تا کاربر با کشیدن آن بر روی صفحه مقدار مورد نظرش را انتخاب کند. و در نهایت در این صفحه یک نواری تعییه شده است که درصد پیشرفت محاسبات مدل‌ها را نشان می‌دهد و به کمک آن می‌توان فهمید که چه مقدار دیگر از کار پردازش مدل‌ها بر روی تصویر باقی مانده است.
- صفحه‌ی دوم: این صفحه وظیفه‌ی نشان دادن خروجی برنامه را به کاربر دارد بدین صورت که در بالای صفحه تصویر تراز شده‌ی چهره قرار می‌گیرد به همراه سن تخمینی به دست آمده برای چهره‌ی ورودی. در زیر آن دو تصویر که یکی برای خروجی مدل گلو و دیگری برای خروجی مدل سایکل‌گن می‌باشد قرار دارند که در زیر هر کدام سن تخمینی به دست آمده توسط مدل تخمین سن به همراه مقدار اختلافی که بین سن ورودی و خروجی توسط آن مدل می‌باشد قرار گرفته است جهت مقایسه‌ی بهتر و آسان‌تر مدل‌ها. در نهایت در پایین صفحه دکمه‌ای برای برگستن به صفحه‌ی اول و انجام دادن دوباره‌ی آزمایش تعییه شده است.

¹GUI: Graphical User Interface

²HTML

³CSS

⁴JavaScript

⁵View

⁶Slider

۷-۵ خلاصه

در این فصل به بررسی سامانه‌ای که برای این پروژه ایجاد شده بود پرداختیم. ابتدا ساختار کلی آن را مورد تحلیل قرار دادیم و دیدیم که چرا معماری یکپارچه مناسب این پروژه نبود و در نتیجه به سراغ معماری مایکروسرویس رفتیم. به طور خلاصه به دلیل وجود مغایرت میان نیازمندی‌ها و کتابخانه‌های مورد نیاز مدل‌های هوش مصنوعی بکار گرفته شده در برنامه و آزمون راحت‌تر و جداگانه‌ی هر بخش برنامه از معماری یکپارچه استفاده نکردیم.

سپس به سراغ توضیح چارچوب جنگو که در مرکز این سامانه قرار دارد و وظیفه‌ی مدیریت و ایجاد ارتباط میان سایر سرویس‌های برنامه را دارد رفتیم و با اجزای مختلف آن آشنا شدیم. همچنین بخش رابط کاربری سامانه بر روی همین سرور جنگو پیاده‌سازی شده است.

در نهایت به توضیح جداگانه‌ی سرویس‌های برنامه رفتیم و نحوه‌ی پیاده‌سازی هر کدام را دیدیم. هر سرویس برنامه مسئولیت میزبانی یکی از مدل‌های هوش مصنوعی یعنی گلو، سایکل‌گن و مدل تخمین سن بکار برده شده در پروژه را بر عهده دارد. همچنین هر سرویس با چارچوب فلساک^۱ پیاده‌سازی شده است و به طور مستقل قابل دسترسی می‌باشد که همین امر باعث پایده‌سازی آسان‌تر و عیب‌یابی راحت‌تر برنامه شده است.

¹Flask

فصل ششم

جمع‌بندی و نتیجه‌گیری

در این پژوهه ابتدا به روش‌های متفاوتی که می‌توانستیم با آن عمل پیر کردن چهره را انجام دهیم پرداختیم و دیدیم که می‌توان به صورت کلی این روش‌ها را به دو دسته‌ی روش‌های سنتی و روش‌های مبتنی بر داده تقسیم نمود. سپس به بررسی هر کدام از این گروه‌ها پرداختیم و مزایا و معایب هر کدام را مورد بحث و مقایسه قرار دادیم. همچنین در بخش روش‌های مبتنی بر داده دیدیم که به دو دسته‌ی کلی‌تر مبتنی بر یادگیری عمیق و مبتنی بر مدل‌های مولد بودند و به مزایا و معایب این روش‌ها پرداختیم. سپس به سراغ چرایی انتخاب دو مدلی که در این پژوهه استفاده شده بود رفتیم و دیدیم که چرا این دو مدل مدل‌های مناسبی برای انجام عمل پیر کردن چهره بودند.

سپس در فصل سه به توضیح دقیق‌تر معماری و ساختار هر یک از مدل‌های گلو [۱] و سایکل‌گن [۲] پرداختیم و بر مقالات آن‌ها مروری کردیم.

در فصل چهار به بررسی مدلی که برای تخمین سن [۱۸] [۵] از آن در پژوهه استفاده کرده بودیم پرداختیم و همچنین مروری کردیم بر برخی از مجموعه دادگانی که برای عمل تخمین سن در تحقیقات مختلف [۳] [۴] مورد استفاده قرار گرفته بودند. سپس چند چالش بزرگ ارزیابی مدل‌های پیر کردن چهره را نام بردیم.

پس از آن در فصل پنج به سراغ سامانه‌ی تحت وبی که برای این پژوهه توسعه داده شده بود رفتیم. ابتدا به توضیح این قضیه پرداختیم که چرا نمی‌توانستیم برای این سامانه از معماری ساده‌ی یکپارچه استفاده کنیم. سپس به توضیح معماری مایکروسرویس پرداختیم و از آن جایی که چارچوب جنگو در قلب این سیستم قرار داشت به توضیح بیشتر این چارچوب و اجزای مختلف آن پرداختیم. پس از آن هر یک از سرویس‌هایی که برای مدل‌های مختلف استفاده شده بود را توضیح دادیم.

مراجع و منابع

- [1] Kingma, Diederik P. and Dhariwal, Prafulla. Glow: Generative flow with invertible 1x1 convolutions, 2018.
- [2] Zhu, Jun-Yan, Park, Taesung, Isola, Phillip, and Efros, Alexei A. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2020.
- [3] Dahlan, Hadi A. A survey on deep learning face age estimation model: Method and ethnicity. International Journal of Advanced Computer Science and Applications, 12(11), 2021.
- [4] ELKarazle, Khaled, Raman, Valliappan, and Then, Patrick. Facial age estimation using machine learning techniques: An overview. Big Data and Cognitive Computing, 6(4):128, 2022.
- [5] Rothe, Rasmus, Timofte, Radu, and Van Gool, Luc. Deep expectation of real and apparent age from a single image without facial landmarks. International Journal of Computer Vision, 126(2-4):144–157, 2018.
- [6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza Bing Xu David Warde-Farley Sherjil Ozair Aaron Courville Yoshua Bengio. "generative adversarial networks". 2014.
- [7] Grimmer, Marcel, Ramachandra, Raghavendra, and Busch, Christoph. Deep face age progression: A survey. IEEE Access, 9:83376–83393, 2021.

- [8] Angulu, Raphael, Tapamo, Jules R, and Adewumi, Aderemi O. Age estimation via face images: a survey. *EURASIP Journal on Image and Video Processing*, 2018(1):1–35, 2018.
- [9] Agbo-Ajala, Olatunbosun, Viriri, Serestina, Oloko-Oba, Mustapha, Ekundayo, Olufisayo, and Heymann, Reolyn. Apparent age prediction from faces: A survey of modern approaches. *Frontiers in big Data*, 5, 2022.
- [10] Dinh, Laurent, Krueger, David, and Bengio, Yoshua. Nice: Non-linear independent components estimation, 2015.
- [11] Dinh, Laurent, Sohl-Dickstein, Jascha, and Bengio, Samy. Density estimation using real nvp, 2017.
- [12] Graves, Alex. Generating sequences with recurrent neural networks, 2014.
- [13] van den Oord, Aaron, Kalchbrenner, Nal, and Kavukcuoglu, Koray. Pixel recurrent neural networks, 2016.
- [14] van den Oord, Aaron, Dieleman, Sander, Zen, Heiga, Simonyan, Karen, Vinyals, Oriol, Graves, Alex, Kalchbrenner, Nal, Senior, Andrew, and Kavukcuoglu, Koray. Wavenet: A generative model for raw audio, 2016.
- [15] Kingma, Diederik P and Welling, Max. Auto-encoding variational bayes, 2022.
- [16] Kingma, Diederik P. and Welling, Max. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.
- [17] Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

-
- [18] Rothe, Rasmus, Timofte, Radu, and Van Gool, Luc. Dex: Deep expectation of apparent age from a single image. in Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops, December 2015.
 - [19] Django documentation. <https://docs.djangoproject.com/en/3.2/>.