دانشکده مهندسی کامپیوتر

توسعه یک برنامه کاربردی جهت پاکسازی داده ها در مجموعه داده های تصویر

پایان‌نامه برای دریافت درجه کارشناسی
در رشته مهندسی کامپیوتر

امیرحسین درخشان

استاد راهنما

دکتر محمدرضا محمدی

شهریور ۱۴۰۲


بسم الله الرحمن الرحیم

تاییدیه‌ئ هیات داوران جلسه دفاع از پایان‌نامه/رساله

نام دانشکده: دانشکده مهندسی کامپیوتر

نام دانشجو: امیرحسین درخشان

عنوان پایان‌نامه یا رساله: توسعه یک برنامه کاربردی جهت پاکسازی داده ها در مجموعه داده های تصویر

تاریخ دفاع: شهریور ۱۴۰۲

رشته: مهندسی کامپیوتر

ردیف	سمت	نام و نام خانوادگی	مرتبه دانشگاهی	دانشگاه یا موسسه	امضا
۱	استاد راهنما	دکتر محمدرضا محمدی	استاد یار	علم و صنعت ایران	
۲	استاد داور	دکتر عادل ترکمان رحمانی	استاد یار	علم و صنعت ایران	

تاییدیه‌ی صحت و اصالت نتایج

بسمه‌تعالی

این‌جانب امیرحسین درخشان به شماره دانشجویی ۹۸۵۲۱۱۷۱ تایید می‌نمایم که کلیه نتایج این پایان‌نامه/رساله حاصل کار اینجانب و بدون هر گونه دخل و تصرف است و موارد نسخه‌برداری شده از آثار دیگران را با ذکر کامل مشخصات منبع ذکر کرده‌ام. در صورت اثبات خلاف مندرجات فوق، به تشخیص دانشگاه مطابق با ضوابط و مقررات حاکم (قانون حمایت از حقوق مولفان و مصنفان و قانون ترجمه و تکثیر کتب و نشریات و آثار صوتی، ضوابط و مقررات آموزشی، پژوهشی و انضباطی ...) با اینجانب رفتار خواهد شد و حق هر گونه اعتراض در خصوص احقاق حقوق مکتسب و تشخیص و تعیین تخلف و مجازات را از خویش سلب می‌نمایم. در ضمن، مسولیت هر گونه پاسخگویی به اشخاص اعم از حقیقی و حقوقی و مراجع ذی‌صلاح (اعم از اداری و قضایی) به عهده اینجانب خواهد بود و دانشگاه هیچ‌گونه مسولیتی در این خصوص نخواهد داشت.

نام و نام‌خانوادگی: امیرحسین درخشان

امضا و تاریخ:

مجوز بهره‌برداری از پایان‌نامه

بهره‌برداری از این پایان‌نامه در چهارچوب مقررات کتابخانه و با توجه به محدودیتی که توسط استاد راهنما به شرح زیر تعیین می‌شود، بلامانع است:

- ☐ بهره‌برداری از این رساله/پایان‌نامه برای همگان بلامانع است.
- ☐ بهره‌برداری از این رساله/پایان‌نامه با اخذ مجوز از استاد راهنما، بلامانع است.
- ☐ بهره‌برداری از این رساله/پایان‌نامه تا تاریخ ———————— ممنوع است.

نام استاد یا اساتید راهنما:
تاریخ:
امضا:

سپاس

قبل از هر چیز بر خود لازم میدانم که خداوند متعال را بابت نعمت های بیکران و الطاف بی منتهایش شکر کنم.

نعمتت بار خدایا ز عدد بیرون است شکر انعام تو هرگز نکند شکرگزار

سپس از باب "من لم یشکر المخلوق لم یشکر الخالق" از پدر، مادر، برادر و خواهرم که در کلیه مراحل زندگی پشتیبانم بودند ، سپاسگزارم.

همچنین از استاد بزرگوارم جناب آقای دکتر محمدرضا محمدی که وقت خود را در اختیار بنده قرار داده و با راهنمایی های ارزنده خود زمینه ساز انجام این پروژه شدند نیز تشکر میکنم.

چکیده

یکی از نکات مهم در الگوریتم های یادگیری ماشین استفاده از داده های درست و با کیفیت جهت آموزش مدل میباشد. داده های جمع آوری شده بنابر دلایل متعددی میتوانند دارای ایرادات مختلفی باشند. یکی از ایرادات رایج و تاثیر گذار، وجود داده هایی با برچسب اشتباه میباشد. در این پروژه سعی شده است راهکاری برای کشف چنین داده هایی ارائه شود. در راهکار ارائه شده ابتدا نیاز است یک شبکه عصبی siamese روی مجموعه داده ای که قرار است اشتباهات آن یافت شوند، جهت کشف میزان شباهت دو داده به یکدیگر، آموزش داده شود. سپس به کمک این شبکه شبیه ترین داده های train به داده های validation که اشتباه پیشبینی شده اند یافت شوند. در ادامه با اصلاح این داده ها با نظارت کاربر یا انجام یک مرحله فیلتر و حذف خودکار آن ها انتظار داریم کیفیت داده ها افزایش یافته باشد و به دنبال آن مدل دقیق تری روی داده آموزش داده شود. در این پروژه نتایج به دست آمده از این روش با نتایج به دست آمده از الگوریتم cleanlab مقایسه شده اند. و در نهایت راهکارهایی جهت افزایش دقت راهکار پیشنهادی نیز ذکر شده است.

کلیدواژه‌ها: پاکسازی داده ها، شبکه های عصبی مصنوعی، شبکه های siamese

فهرست مطالب

فهرست جدول‌ها

فهرست شکل‌ها

فصل ۱

مقدمه

یادگیری ماشین امروزه سهمی مهم و انکار ناشدنی در صنعت، تحقیقات و حتی زندگی بشر دارد. راهکار هایی که جهت پیشرفت این فناوری استفاده میشوند گاهی متکی بر بهبود و پیشرفت مدل های پیشنهادی و گاهی متکی بر افزایش کیفیت داده میباشند. برای مثال برخی راهکار ها مدل های جدید یا تابع خطاهای هوشمندانه تری را ارائه میدهند و برخی دیگر از راهکار ها روش هایی را برای افزایش تعداد داده های آموزشی و یا افزایش کیفیت داده و حذف داده های نامناسب از مجموعه داده معرفی میکنند. در این پروژه سعی شده است راهکاری جهت افزایش کیفیت داده ها معرفی شود.

۱-۱ توضیح مسئله

در زمان آموزش شبکه های عصبی مصنوعی وجود داده هایی با برچسب اشتباه میتواند سبب افت دقت مدل شود. بنابراین میتوان امید داشت که اگر این داده ها یافت شده و برچسب آنها اصلاح شود و یا حذف شوند، کارایی مدل افزایش یابد. به فرایند کشف و اصلاح یا حذف چنین داده هایی که تاثیر منفی در آموزش مدل دارند، پاکسازی داده ها گفته میشود.

بنابر دلایل مختلفی ممکن است تعدادی از داده ها در یک مجموعه داده، برچسبی اشتباه و یا کیفیت پایینی داشته باشند.برای مثال در شکل ۱-۱ ۵ داده از مجموعه داده MNIST [۱]نمایش داده شده است که دارای برچسب اشتباهی هستند و یا به طور کلی میتوان انتظار داشت که این داده ها باعث افت دقت مدل میشوند.

در برخی از مجموعه داده ها احتمال وجود چنین داده هایی بیشتر است. برای مثال شرکت سامانه

شکل ۱-۱: داده هایی که برچسب اشتباه دارند یا بهتر است از مجموعه داده **MNIST** حذف شوند

های هوشمند کشاورزی والی [1] مجموعه ای از تصاویر خرما که در ۴ کلاس سالم، مشکوک سالم، مشکوک خراب و خراب دسته بندی شده اند را در اختیار ما قرار داده است. برخی از تصاویر این مجموعه داده به گونه ای است که اگر یک فرد دوبار بخواهد برچسب گذاری کند ممکن است برچسب های متفاوتی روی آنها بگذارد. در شکل ۱-۲ ۵ تصویر از هر کلاس این مجموعه داده نمایش داده شده است.

۱-۲ هدف پژوهش

هدف از انجام این پروژه، ارائه روشی جهت کشف و زمینه سازی حذف یا اصلاح داده هایی که احتمال خطا در آن ها بیشتر است، میباشد. در این پروژه پس از کشف داده های خطای احتمالی، به دو صورت میتوان عمل نمود. در روش اول میتوان با استفاده از رابط کاربری طراحی شده خطاهای احتمالی را به کاربر نمایش داد تا کاربر با نظارت بر داده ها، آنها را اصلاح کند و با آموزش مجدد مدل تاثیر اصلاح داده ها بر دقت مدل را مشاهده کند. در روش دوم میتوان به صورت خودکار داده های کشف شده را حذف کرد و تاثیر آن بر دقت مدل مشاهده کرد. لازم به ذکر است که به دلیل طولانی بودن فرایند آموزش شبکه های عصبی با استفاده از CPU روش اول دچار محدودیت هایی میباشد.

[1] https://vali-system.com

شکل ۱-۲: نمونه ای از داده خرما در کلاس های مختلف، **m salem** به معنای مشکوک سالم و **m kharab** به معنای مشکوک خراب میباشد

فصل ۲

مرور کارهای پیشین

در یک نگاه ابتدایی جهت کشف ایرادات موجود در یک مجموعه داده میتوان هر داده از مجموعه داده را جدا کرده سپس مدل را با مجموعه داده جدید آموزش داد و در صورتی که که خطای مدل کاهش یافته بود، آن را به عنوان خطا در نظر گرفت.[۲] بدیهی است که این کار بسیار زمان بر و پرهزینه میباشد. بنابراین باید به دنبال راهکارهای هوشمندانه تری بود.در این بخش تعدادی از راهکارهای مورد استفاده جهت پاکسازی داده ها معرفی میشوند.

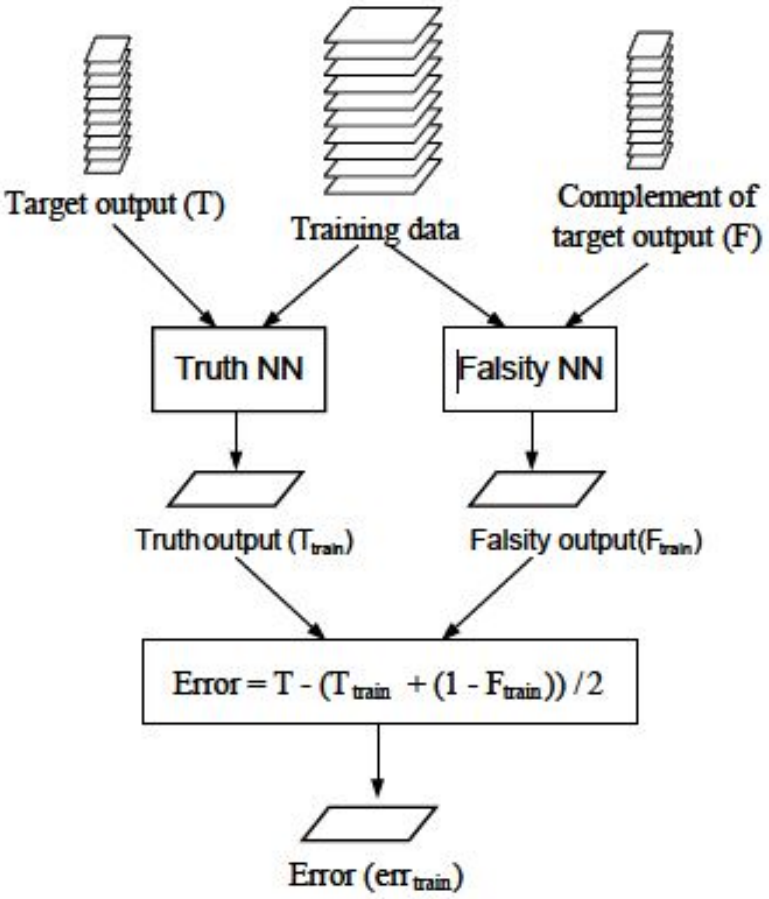
۲-۱ تحلیل دسته بندی های غلط جهت پاکسازی داده ها

در این بخش ابتدا به معرفی شبکه های عصبی مکمل (CMTNN) میپردازیم، سپس چگونگی پاکسازی داده ها با استفاده از CMTNN معرفی میشود.

۲-۱-۱ معرفی CMTNN

CMTNN روشی است که از یک زوج از شبکه های عصبی مکمل با نام های Truth Neural Network و Falsity Neural Network استفاده میکند. این روش برای هر دو حالت دسته بندی دوکلاسه و دسته بندی چند کلاسه پیاده سازی شده است.

در دسته بندی دوکلاسه، یک شبکه عصبی برای تعیین میزان درستی و یک شبکه دیگر برای تعیین میزان نادرستی به کار میرود. همانطور که در شکل ۲-۱ مشخص شده است با استفاده از تفاوت میزان درستی و نادرستی یک داده میتوان میزان خطا یا عدم اطمینان آن را به دست آورد.



شکل ۲-۱: شبکه های عصبی مکمل

[۳]

۲-۱-۲ پاکسازی داده ها با CMTNN

برای یافتن داده های خطا با استفاده از CMTNN به دو صورت میتوان عمل کرد. مراحل روش اول به شرح زیر میباشد:[۳]

۱. مکمل کردن برچسب داده های آموزشی تا به عنوان مجموعه داده آموزشی شبکه Falsity استفاده شوند

۲. آموزش شبکه های عصبی مصنوعی Truth و Falsity

۳. مقایسه پیشبینی های هر دو شبکه با مقادیر واقعی آن ها و در نظر گرفتن داده هایی که پیشبینی آن با داده واقعی مطابقت ندارد به عنوان داده خطا

۴. ایجاد مجموعه داده جدید با حذف کردن داده های تشخیص داده شده توسط هر شبکه

مراحل روش دوم نیز به شرح زیر میباشد:

۱. تکرار گام های ۱ تا ۳ روش قبل

۲. ایجاد مجموعه داده جدید با حذف داده هایی که توسط هر دو شبکه به عنوان خطا تشخیص داده شده اند

۲-۲ استفاده از cleanlab جهت پاکسازی داده ها

یکی از پروژه های موفق در زمینه پاکسازی داده ها در سال های اخیر پروژه cleanlab بوده است. این پروژه که به عنوان تز دکتری آقای Curtis Northcutt در دانشگاه MIT معرفی شده است، کارایی بسیار بالایی در یافتن و حتی اصلاح کردن برچسب های اشتباه مجموعه داده های مختلفی چون مجموعه داده های صوت، تصویر و متن را دارد.

۲-۲-۱ cleanlab چگونه داده های خطا را تشخیص میدهد؟

cleanlab با استفاده از تحلیل برچسب های یک مجموعه داده و برچسب های پیشبینی شده توسط یک مدل، پاکسازی داده ها را انجام میدهد. بنابراین برای استفاده از cleanlab نیاز به یک مدل

داریم تا احتمال تعلق هر داده به هر کلاس را پیشبینی کند. برای این کار میتوان مدل را با استفاده از اعتبارسنجی متقابل [1] آموزش داد.

الگوریتم cleanlab جهت یافتن اشتباه های احتمالی یک مجموعه داده، از دو بخش تشکیل شده است. در بخش اول حد آستانه هر کلاس یافت میشود. در بخش دوم بررسی میکنیم که اگر احتمال تعلق یک داده به یک کلاس بیشتر از حد آستانه آن کلاس باشد، داده را متعلق به آن کلاس بدانیم. در الگوریتم ۱ نحوه محاسبه حد آستانه و در الگوریتم ۲ نحوه یافتن خطاهای احتمالی ذکر شده است.

در این دو الگوریتم، منظور از m تعداد کلاس ها ، n تعداد داده ها، $\hat{p}$ یک ماتریس $m \times n$ از میزان احتمال هر داده به هر کلاس، $\tilde{y}$ برچسب های شامل خطا و y^* برچسب های بدون خطا که دسترسی به آن نداریم میباشد.[۴]

Algorithm ۱ Calculate the threshold of each class

Input $\hat{p}$

Input $\tilde{y}$

Output t

```
for j:=۱ → m do
    for i:=۱ → n do
        l=[]                                  ▷ define new empty list
        if ỹ[i] = j then
            append p̂[i][j] to l
    t[j] = average(l)
```

لازم به ذکر است که اثبات درستی این الگوریتم ها و همچنین نحوه محاسبه میزان اطمینان تعلق یک داده به کلاسی دیگر، به طور کامل در مقاله ارجاع داده شده، ذکر شده است. به علاقه مندان جهت کسب اطلاعات و آگاهی از جزییات بیشتر این الگوریتم توصیه میشود مقاله ارجاع داده شده را مطالعه کنند.

[1] cross-validation

Algorithm ۲ find possible errors

for i:=۱ $\rightarrow$ n do

cnt = ۰

for j:=۱ $\rightarrow$ m do

if $\hat{p}[i][j] \geqslant t$ then

cnt+=۱

$y^*[i] = j$

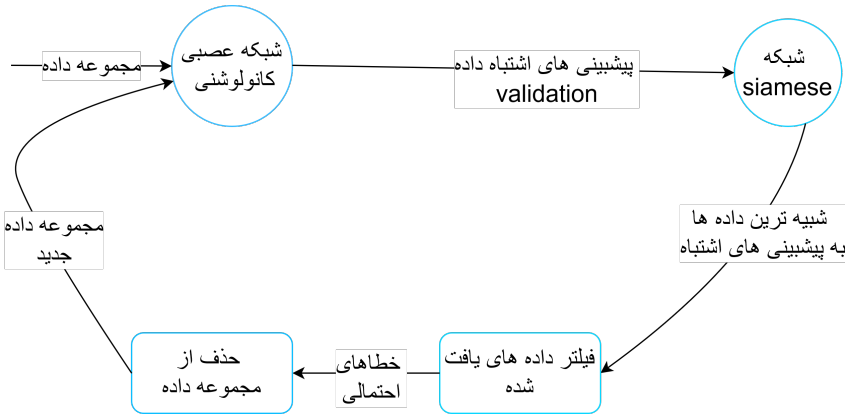
if $cnt > 1$ then

$y^*[i] = argmax\hat{p}[i]$

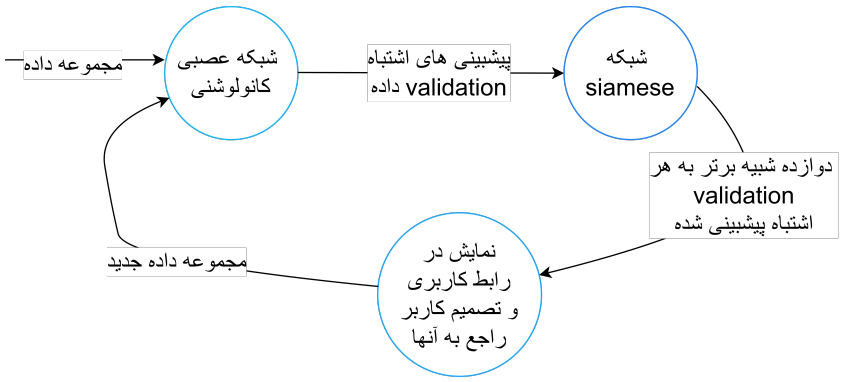
فصل ۳

روش پیشنهادی

در این بخش به معرفی راهکار انجام شده در این پروژه، جهت پاکسازی داده ها میپردازیم. در این پروژه جهت پاکسازی داده ها ابتدا یک شبکه عصبی مصنوعی را با استفاده از داده های آموزشی آموزش میدهیم. سپس داده های validation که توسط مدل اشتباه پیشبینی میشوند را می یابیم. سپس با استفاده از یک شبکه عصبی siamese شبیه ترین داده های train به این داده ها را می یابیم. در این مرحله میتوانیم این داده های شبیه را با استفاده از رابط کاربری طراحی شده به کاربر نمایش دهیم تا کاربر راجع به اصلاح آن ها تصمیم بگیرد و یا میتوانیم پس از انجام یک مرحله فیلتر و انتخاب تعدادی از داده های یافت شده آن ها را حذف کرده و با آموزش مجدد مدل تاثیر حذف داده ها بر دقت مدل را مشاهده کنیم. لازم به ذکر است که میتوان این چرخه را چند بار تکرار کرد. در شکل ۳-۱ و شکل ۳-۲ چرخه انجام این کار نمایش داده شده است.



شکل ۳-۱: چرخه حذف داده های پیشبینی شده، بدون استفاده از رابط کاربری



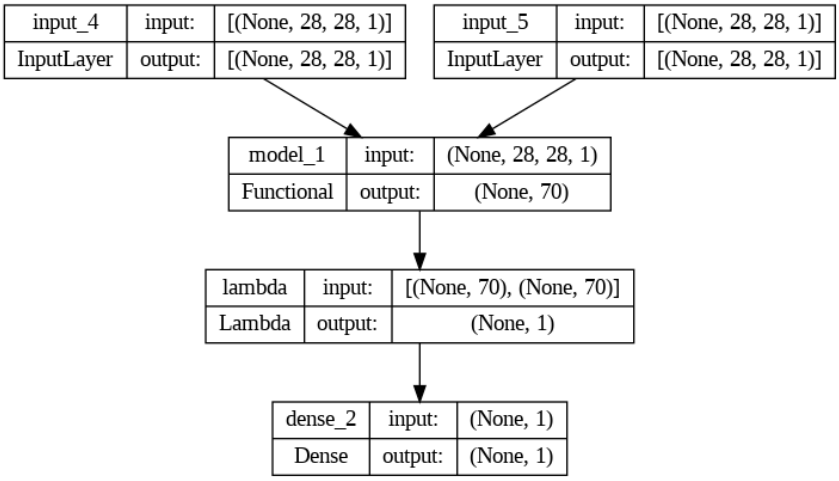
شکل ۳-۲: چرخه اصلاح داده ها به کمک رابط کاربری

۳-۱ ساخت شبکه عصبی siamese

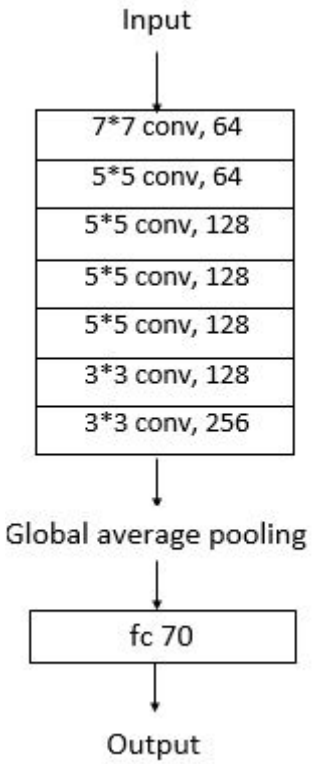
شبکه های عصبی siamese نوعی از شبکه های عصبی مصنوعی هستند که شامل دو یا چند شبکه کاملا یکسان میباشند. کاملا یکسان یعنی این چند شبکه دارای معماری و وزن های یکسانی میباشند و بروز رسانی وزن ها در هر دو شبکه نیز یکسان است. این شبکه ها با استفاده از ویژگی های استخراج شده از هر داده و مقایسه آن ها با هم، جهت کشف میزان شباهت چند داده با هم به کار میروند.[۵]

۳-۱-۱ ساختار شبکه های siamese به کار رفته

معماری کلی شبکه های siamese استفاده شده در این پروژه در شکل ۳-۳ نمایش داده شده است. همانطور که مشخص است ابتدا یک زوج که دارای برچسب ۱ به معنی مشابه یا ۰ به معنی متفاوت هستند، وارد یک شبکه عمیق میشوند تا ویژگی های آن ها استخراج شود. جهت شباهت سنجی مجموعه داده MNIST در این بخش از یک شبکه کانولوشنی ۸ لایه استفاده شده است و در نهایت با استفاده از Global Average Polling و یک لایه Dense، ۷۰ ویژگی از تصاویر استخراج شده است. معماری این شبکه در شکل ۳-۴ نمایش داده شده است. برای مجموعه داده خرما از شبکه Resnet50 [۶] جهت استخراج ۷۰ ویژگی از تصاویر استفاده شده است. سپس با استفاده از فاصله اقلیدسی ویژگی های استخراج شده از تصاویر کار آموزش مدل و شباهت سنجی تکمیل میشود. لازم به ذکر است جهت افزایش دقت مدل و بسته به دشواری مجموعه داده مورد نظر میتوان ساختار شبکه میانی و تعداد ویژگی های استخراج شده از هر تصویر را تغییر داد تا شاهد دقت مطلوب تری از شبکه siamese بود. آن چه که در این مرحله به عنوان خروجی مدل در نظر گرفته میشود میزان شباهت دو تصویر به هم میباشد.



شکل ۳-۳: معماری شبکه **siamese**



شکل ۳-۴: معماری شبکه به کار رفته جهت استخراج ویژگی های تصاویر مجموعه داده **MNIST**

۳-۱-۱-۱ نحوه ساخت زوج ها

همانطور که در قسمت قبل ذکر شد جهت آموزش شبکه siamese نیاز به تشکیل یک زوج از دو تصویر به عنوان داده شبیه یا متفاوت داریم. ایجاد زوج ها بسته به مجموعه داده، متفاوت میباشد. برای مثال در مجموعه داده خرما جهت تشکیل یک زوج شبیه میتوان از یک تصویر و چرخش ۱۸۰ درجه ای همان تصویر استفاده کرد. اما در مجموعه داده MNIST اجازه چرخش ۱۸۰ درجه ای یک تصویر جهت ساخت زوج شبیه را نداریم.

در یک نگاه ابتدایی جهت ساخت زوج های شبیه میتوان از داده هایی که در یک کلاس هستند استفاده کرد و جهت ساخت زوج های متفاوت از داده های موجود در دو کلاس متفاوت استفاده کرد. اما این روش ساده برای مجموعه داده های پیچیده کارایی چندانی ندارد. بنابراین باید به گونه ای هوشمندانه تر زوج ها را تشکیل داد. در ادامه به توضیح چگونگی ساخت زوج ها جهت آموزش شبکه siamese روی دو مجموعه داده MNIST و خرما پرداخته میشود.

۳-۱-۱-۱-۱ ساخت زوج ها در مجموعه داه تصاویر خرما

در این مجموعه داده نیمی از زوج ها را شبیه و نیم دیگر را زوج متفاوت میسازیم. جهت ساخت زوج های شبیه با توجه به نزدیک بودن کلاس ها به یکدیگر با چالش هایی مواجه هستیم بنابراین جهت ساخت زوج های شبیه نباید روی انتخاب از دو تصویر هم کلاس و جهت ساخت زوج های منفی نباید روی انتخاب از دو تصویر با کلاس های متفاوت تمرکز کرد. در این پروژه جهت ساخت زوج های شبیه در این مجموعه داده، با احتمال حدود ۱۵% تصاویری با برچسب مشابه، با احتمال حدود ۶۰% از یک تصویر و چرخش ۱۸۰ درجه ای آن، با احتمال حدود ۱۰% از یک تصویر و قرینه افقی آن، با احتمال حدود ۱۰% از یک تصویر و قرینه عمودی آن و با احتمال حدود ۵% از یک تصویر و همان تصویر به همراه مقداری نویز گاوسی استفاده شده است.

جهت ساخت زوج های متفاوت نیز با احتمال حدود ۴۵% از یک تصویر کلاس سالم و یک تصویر کلاس خراب، با احتمال حدود ۲۰% از یک تصویر کلاس مشکوک سالم و یک تصویر کلاس خراب، با احتمال حدود ۲۰% از یک تصویر کلاس سالم و یک تصویر کلاس مشکوک خراب، با احتمال حدود ۵% از یک تصویر کلاس سالم و یک تصویر کلاس مشکوک سالم، با احتمال حدود ۵% از یک تصویر کلاس مشکوک سالم و مشکوک خراب و با احتمال حدود ۵% از یک تصویر کلاس مشکوک خراب و یک تصویر کلاس خراب استفاده شده است.

۳-۱-۱-۱-۲ ساخت زوج ها در مجموعه داده MNIST

در مجموعه داده MNIST با چالش کمتری برای ساخت زوج مواجه هستیم. زیرا میتوان این ادعا را داشت که میزان شباهت عدد ۳ با عدد ۴ و ۵ تفاوتی ندارد و در هر دو میزان شباهت ۰ میباشد. جهت ساخت زوج در MNIST نیمی از زوج ها را شبیه و نیمی را متفاوت میسازیم. جهت ساخت زوج های مثبت با احتمال ۹۰% از دو تصویر هم کلاس، با احتمال ۵% از یک تصویر به همراه چرخش بین ۰ تا ۱۰ درجه ای آن به سمت چپ یا راست و با احتمال ۵% از یک تصویر به همراه همان تصویر با مقداری نویز گاوسی استفاده میکنیم. جهت ساخت زوج های منفی نیز سعی شده است از تمام حالات ممکن انتخاب دو کلاس متفاوت به تعداد یکسان داشته باشیم.

۳-۱-۲ عملکرد شبکه های siamese آموزش داده شده

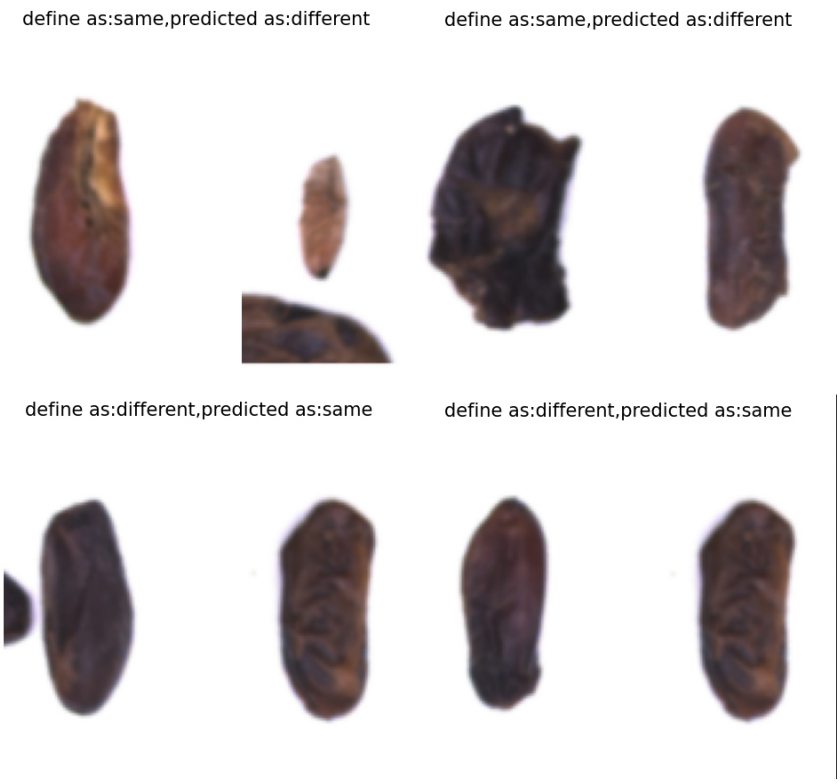
در مجموعه داده MNIST و مجموعه داده خرما به ترتیب به دقت حدود ۹۹% و ۹۰% در داده های train و validation دست یافته ایم. در شکل ۳-۵ چند نمونه از پیشبینی های اشتباه مدل آموزش داده شده روی تصاویر خرما نمایش داده شده است که نشان میدهد همچنان جهت دستیابی به دقت های بالاتر نیاز است نحوه ساخت زوج ها را بهبود بخشید.

انتظار داریم تصویر یک خرما و چرخش ۱۸۰ درجه ای آن کاملا مشابه باشند. زیرا یک خرما هستند و تنها نحوه تصویر برداری از آن ها متفاوت میباشد. در شکل ۳-۶ ۳ داده از ۱۰۰۰ داده اول که کمترین میزان شباهت به چرخش خود را دارند نمایش داده شده است. که با توجه به حاشیه های موجود در تصاویر این اتفاق طبیعی میباشد.

جهت اینکه یافتن شبیه ترین تصویر به یک تصویر دیگر زمان کمتری ببرد، یکبار ۷۰ ویژگی تصاویر استخراج و ذخیره شده اند و جهت یافتن شبیه ترین تصویر به تصویری دیگر ویژگی های از قبل ذخیره شده با هم مقایسه میشوند. در شکل ۳-۷ یک داده validation خرما و ۱۲ شبیه اول از تصاویر train به آن نمایش داده شده است.

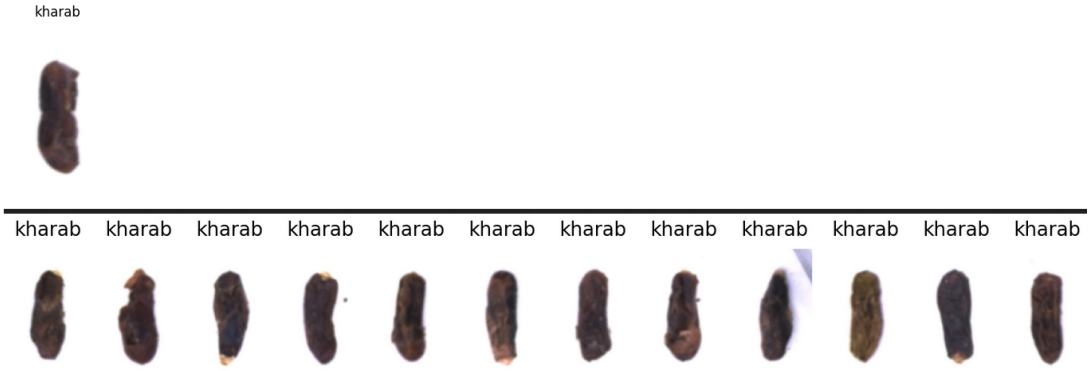
۳-۲ یافتن خطاهای احتمالی و اصلاح یا حذف آن ها

بعد از این که با استفاده از شبکه siamese، شبیه ترین داده ها به داده های validation اشتباه پیشبینی شده، یافت شدند این داده ها را میتوانیم به عنوان خطاهای احتمالی در نظر بگیریم و زمینه اصلاح یا حذف آن ها را فراهم کنیم. برای این کار میتوان از رابط کاربری طراحی شده و یا حذف خودکار



شکل ۳-۵: زوج هایی که اشتباه تشخیص داده شده اند

شکل ۳-۶: تصاویری که کمترین میزان شباهت به چرخش خود را دارند



شکل ۳-۷: ۱۲ شبیه اول تشخیص داده شده به یک داده **validation**

داده های یافت شده استفاده کرد.

۳-۲-۱ معرفی رابط کاربری

جهت اصلاح برچسب داده های کشف شده، یک رابط کاربری طراحی شده است که با نمایش یک داده validation اشتباه پیش بینی شده و ۱۲ شبیه برتر به آن، زمینه اصلاح برچسب ها را فراهم میکند. برای این کار ابتدا باید n تصویر validation که اشتباه پیشبینی شده اند را در پوشه هایی با نام های ۱ تا n ذخیره شوند. سپس در هر پوشه ۱۲ شبیه به آن ها باید با نام های ۱ تا ۱۲ ذخیره شوند و کل این مجموعه در پوشه ای دیگر قرار بگیرد. سپس با انتخاب دکمه Result path پوشه مورد نظر به برنامه معرفی میشود. جهت اصلاح برچسب ها نیاز به ذخیره اطلاعات تصویر در فرمت json میباشیم. و از این جهت استفاده از این رابط کاربری برای بعضی مجموعه داده ها دارای محدودیت هایی میباشد.

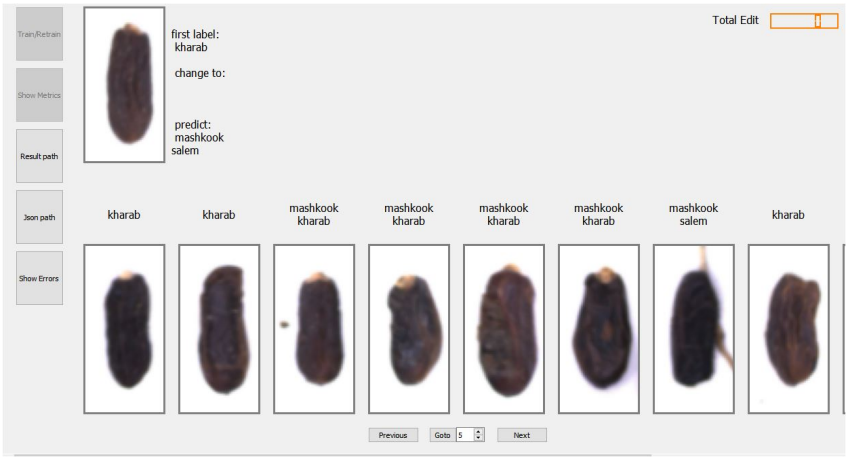
پس از انتخاب پوشه، تصویر اول به همراه شبیه های آن و برچسب های ابتدایی و پیشبینی شده به کاربر نمایش داده میشود. جهت اصلاح برچسب یک تصویر، باید روی آن کلیک راست کرده، سپس در منوی ظاهر شده برچسب جدید را انتخاب کرد. با این کار قالب سبز رنگی در حاشیه عکس اصلاح شده ظاهر میشود که نشان از موفقیت آمیز بودن تغییر برچسب میدهد. با استفاده از شمارشگر موجود در بالا سمت راست رابط کاربری، میتوان تعداد اصلاحات انجام شده را مشاهده کرد.

در رابط کاربری دو دکمه Train/Retrain و show metrics نیز به منظور آموزش دوباره مدل و مشاهده تغییرات در دقت و دیگر معیار های مدل، بعد از انجام اصلاحات قرار گرفته اند. اما از آنجا که آموزش مدل بدون استفاده از GPU زمان بر است و بیشتر سیستم های کاربران GPU ندارند، این امکان در حال حاضر غیر فعال میباشد.

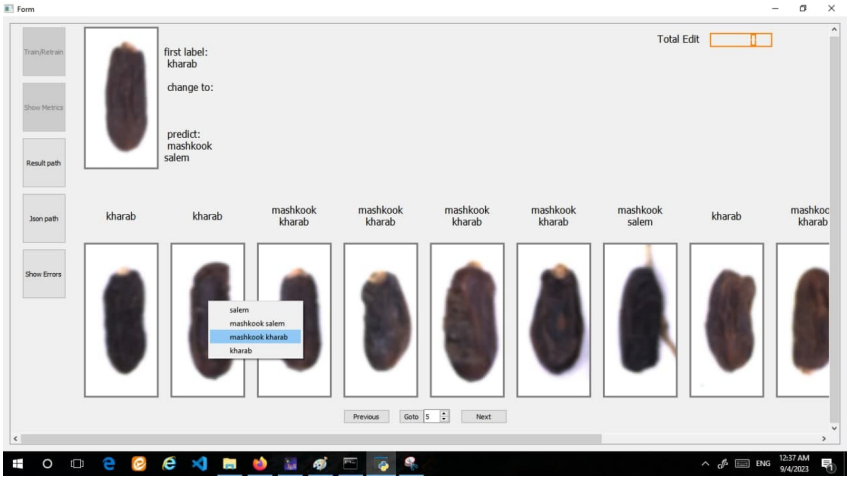
در شکل ۳-۸ و شکل ۳-۹ و شکل ۳-۱۰ محیط رابط کاربری نمایش داده شده است.

۳-۲-۲ حذف خودکار داده های تشخیص داده شده

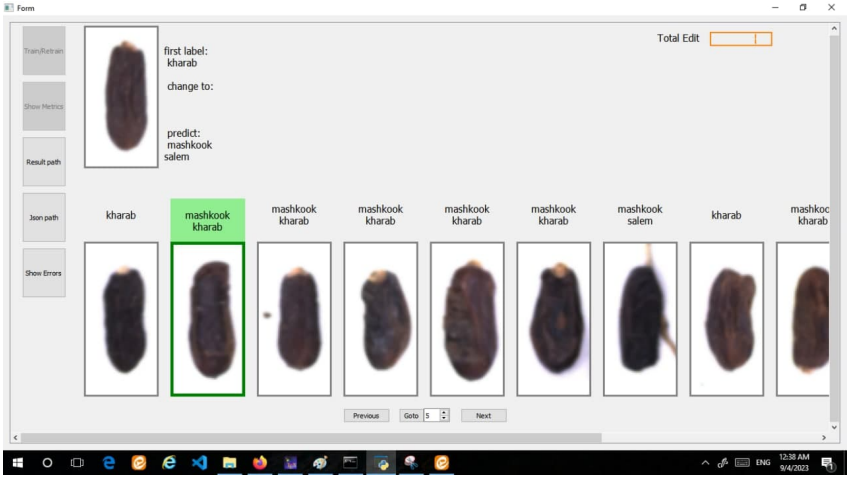
جهت تشخیص داده های خطا بدون نظارت کاربر، میتوان از الگوریتم ۳ استفاده کرد. در این الگوریتم از ۳ پارامتر تعداد شبیه های یافت شده به یک تصویر(n)، حداقل تعداد تکرار یک داده در بین شبیه های یافت شده(MINREPEAT) و حداقل درصد اطمیان تعلق یک داده به کلاسی



شکل ۳-۸: محیط رابط کاربری



شکل ۳-۹: نمایش برچسب های ممکن برای یک داده



شکل ۳-۱۰: رابط کاربری بعد از اصلاح یک داده

دیگر(MIN CONFIDENCE) استفاده میشود.

Algorithm ۳ find possible errors of dataset

Input: wrong prediction valids ▷ list of indexes of wrongly predicted validations data(w items) Input: n

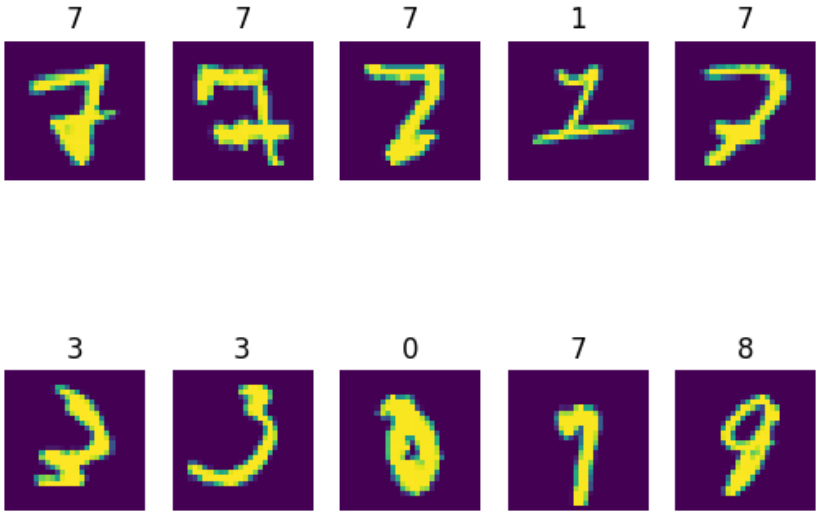
Input Min_Repeat

Input Min_Confidence

Output Possible errors ▷ list of indexes of possibel errors

```
predicted_wrongs = new empty dictionary
for i:=۱ → w do
    top_n = FindTopNSimilars(wrong prediction valids[i] ، n)
    for j:=۱ → n do
        if top_n[j] key not in dictionary then
            predicted_wrongs[top_n[j]] = [0 for classes number]
            predicted_wrongs[top_n[j]][y_valid[wrong prediction valids[i]]] += 1
        else
            predicted_wrongs[top_n[j]][y_valid[wrong prediction valids[i]]] += 1
Sort the dictionary in descending order based on sum of values
Delete the items whose value sum is less than Min_Repeat
Convert each of the values in the list of values to a percentage and consider
the highest percentage along with the class corresponding to the highest
percentage instead of the value
Delete the items whose confidence is less than Min_Confidence
for each key in dictionary if y_valid[key] not equal to highest percentage
label add key to possible errors list
```

جهت تصحیح خودکار برجسب ها میتوان حداقل درصد دیگری را در نظر گرفت و اگر با احتمال بیشتری یک داده به کلاس دیگری تعلق داشت آن داده را متعلق به آن کلاس دانست. در شکل ۳-۱۱ نمونه ای از تصاویر تشخیص داده شده توسط این الگوریتم در مجموعه داده MNIST نمایش داده شده است.



شکل ۳-۱۱: ۱۰ تصویر تشخیص داده شده به عنوان خطا در MNISTبا الگوریتم پیاده سازی شده

فصل ۴

مقایسه نتایج

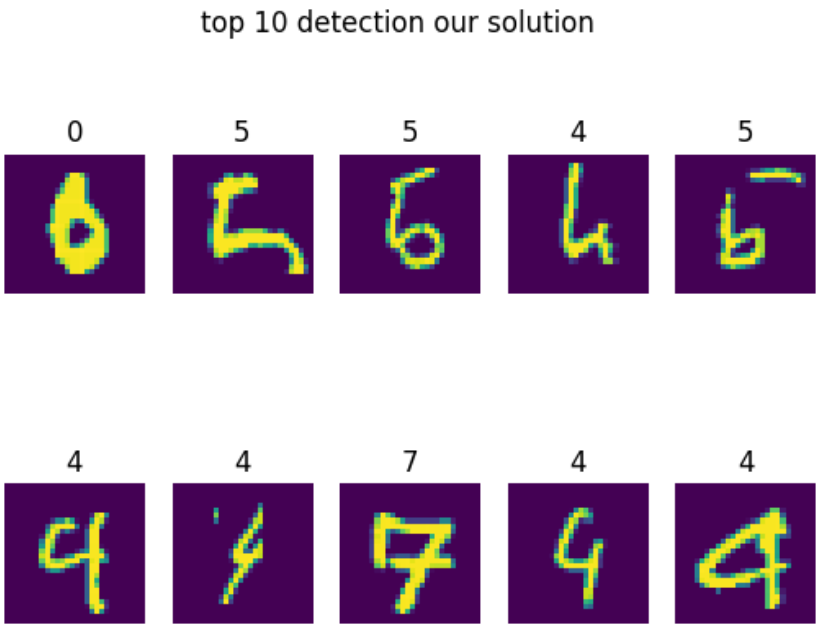
در این بخش به مقایسه نتایج به دست آمده و کیفیت پاکسازی داده ها، با روش پیاده سازی شده و الگوریتم cleanlab میپردازیم. برای این کار ابتدا یک مجموعه داده را با یک شبکه عصبی مصنوعی آموزش میدهیم. سپس کار پاکسازی داده ها را روی مجموعه داده انجام میدهیم و با حذف خطاهای احتمالی از مجموعه داده، مجدد شبکه عصبی مصنوعی را آموزش میدهیم و نتایج را با نتایج اولیه مقایسه میکنیم.

جهت بررسی عملکرد پاکسازی داده ها، ابتدا مجموعه داده MNIST را بررسی کرده ایم. در مرحله بعد روی این مجموعه داده ۷۰۰۰ و باری دیگر ۲۱۰۰۰ خطا ایجاد میکنیم و انتظار داریم با استفاده از الگوریتم پیاده سازی شده این خطاها را کشف کنیم. در مرحله آخر نیز به بررسی عملکرد پاکسازی داده ها روی مجموعه داده خرما میپردازیم.

۴-۱ پاکسازی داده ها روی مجموعه داده MNIST

در این بخش به بررسی عملکرد الگوریتم پیاده سازی شده و الگوریتم cleanlab در پاکسازی مجموعه داده MNIST میپردازیم. در جدول ۴-۱ دقت مدل قبل و بعد از انجام چند مرحله پاکسازی داده ها به همراه پارامترهای الگوریتم و همچنین دقت بعد از حذف موارد تشخیص داده شده توسط cleanlab نوشته شده است. لازم به ذکر است در الگوریتم cleanlab جهت آموزش مدل با اعتبار سنجی متقابل مقدار k-fold برابر با ۵ در نظر گرفته شده است.

در شکل ۴-۱ ۱۰ تشخیص برتر الگوریتم پیاده سازی شده به عنوان خطا و در شکل ۴-۲ ۱۰



شکل ۴-۱: ۱۰ خطای اول کشف شده با استفاده از الگوریتم پیاده سازی شده

تشخیص برتر الگوریتم cleanlab نمایش داده شده است.

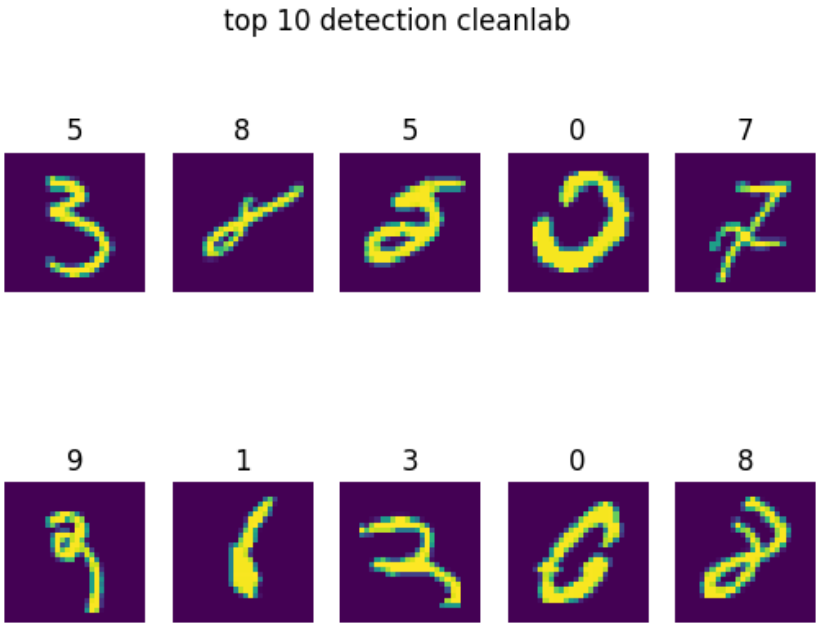
۴-۲ ایجاد ۷۰۰۰ خطا روی مجموعه داده MNISTو انجام پاکسازی داده ها

در این بخش ابتدا ۷۰۰۰ خطا روی مجموعه داده MNIST ایجاد کرده ایم. یعنی برچسب ۷۰۰۰ داده را به برچسبی غیر از برچسب اصلی اشان تغییر داده ایم. سپس با انجام فرایند باکسازی داده ها مشاهده میکنیم که چه تعداد از داده های خطا تشخیص داده شده اند.

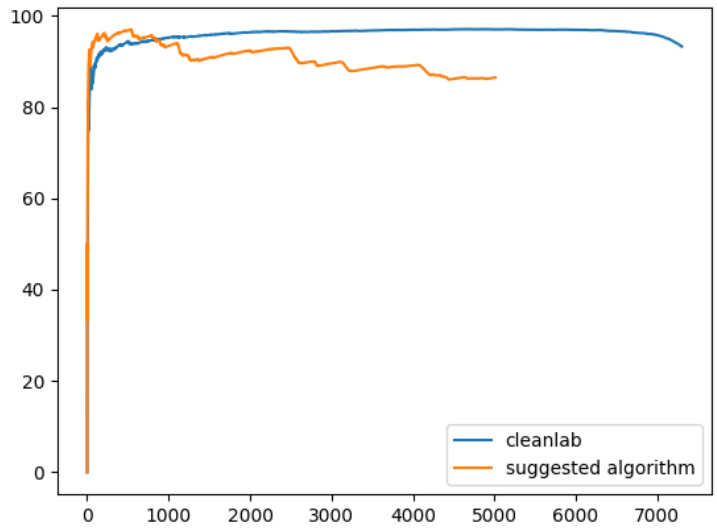
در جدول ۴-۲ تعداد تشخیص های الگوریتم پیاده سازی شده و الگوریتم cleanlab مقایسه شده اند. منظور از درصد اشتراک در n% اول یعنی چند درصد از n% داده اول یافت شده جزو ۷۰۰۰ خطای ایجاد شده هستند.

در شکل ۴-۳ نسبت تعداد تشخیص های درست به کل تشخیص ها به ازای هر تشخیص در مقیاس درصد برای هر دو الگوریتم پیاده سازی شده و cleanlab نمایش داده شده است.

همچنین در جدول ۴-۳ میزان تاثیر حذف تشخیص ها بر دقت مدل نوشته شده است.



شکل ۴-۲: ۱۰ خطای اول کشف شده با استفاده از الگوریتم **cleanlab**



شکل ۴-۳: درصد نسبت تعداد تشخیص درست به کل تشخیص ها با الگوریتم پیاده سازی شده و الگوریتم **cleanlab** در مجموعه داده **MNIST** با ۷۰۰۰ خطا

جدول ۴-۱: نتایج به دست آمده از پاکسازی داده ها روی مجموعه داده **MNIST**

	تعداد شبیه	حداقل تکرار	حداقل اطمینان	تعداد تشخیص الگوریتم	دقت تست به ازای کمترین خطای تست	دقت داده آموزشی به ازای کمترین خطای تست	بیشترین دقت داده تست	دقت داده آموزشی به ازای بیشترین دقت تست
قبل از پاکسازی داده ها	—	—	—	—	۹۹/۲۲	۹۹/۵۷	۹۹/۴۱	۱۰۰
cleanlab	—	—	—	۲۳۴	۹۹/۲۸	۹۹/۶۱	۹۹/۴۴	۱۰۰
روش پیشنهادی مرحله اول	۲۰۰۰	۴	۵۰	۶۹۴	۹۹/۴۲	۹۹/۸۵	۹۹/۴۴	۱۰۰
مرحله دوم	۲۰۰۰	۴	۵۰	۱۵۰	۹۹/۳۱	۹۹/۶۹	۹۹/۳۶	۱۰۰
مرحله سوم	۱۰۰۰	۴	۷۰	۲۱۶	۹۹/۲۵	۹۹/۷۳	۹۹/۳۹	۱۰۰

جدول ۴-۲: میزان دقت الگوریتم ها در یافتن ۷۰۰۰ خطای ایجاد شده

	تعداد تشخیص	درصد اشتراک ۱% اول	درصد اشتراک ۲۰% اول	درصد اشتراک ۵۰% اول	درصد اشتراک ۷۰% اول	درصد اشتراک کل
الگوریتم پیاده سازی شده	۵۰۱۳	۹۲/۱۶	۹۳/۴۲	۹۲/۳۰	۸۸/۶۸	۸۶/۴۷
cleanlab	۷۳۰۲	۸۶/۴۸	۹۵/۶۸	۹۶/۸۷	۹۷/۰۴	۹۳/۲۷

جدول ۴-۳: نتایج به دست آمده از پاکسازی داده ها روی مجموعه داده **MNIST** با ۷۰۰۰ خطا

	تعداد شبیه	حداقل تکرار	حداقل اطمینان	تعداد تشخیص الگوریتم	دقت تست به ازای کمترین خطای تست	دقت داده آموزشی به ازای کمترین خطای تست	بیشترین دقت داده تست	دقت داده آموزشی به ازای بیشترین دقت تست
قبل از پاکسازی داده ها					۹۸/۷۶	۸۷/۲۷	۹۹/۰۷	۸۸/۲۹
cleanlab				۷۳۰۲	۹۹/۳۹	۹۹/۶۵	۹۹/۳۹	۹۹/۶۵
روش پیشنهادی مرحله اول	۵۰۰۰	۳	۵۰	۵۰۱۳	۹۸/۹۷	۹۴/۰۳	۹۹/۲۷	۹۵/۰۰
مرحله دوم	۵۰۰۰	۳	۵۰	۱۷۳۴	۹۸/۹۹	۹۶/۷۶	۹۹/۰۸	۹۶/۶۴
مرحله سوم	۲۰۰۰	۴	۷۰	۴۶۴	۹۹/۰۱	۹۶/۹۶	۹۹/۰۹	۹۶/۸۰

جدول ۴-۴: میزان دقت الگوریتم ها در یافتن ۲۱۰۰۰ خطای ایجاد شده

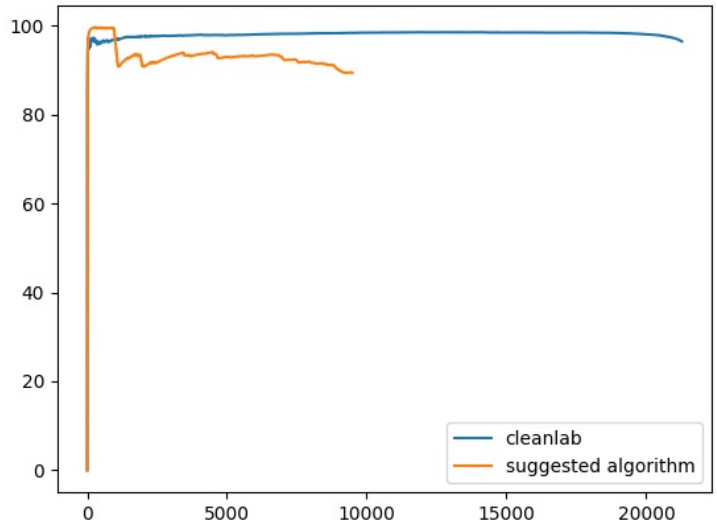
	تعداد تشخیص	درصد اشتراک ۱% اول	درصد اشتراک ۲۰% اول	درصد اشتراک ۵۰% اول	درصد اشتراک ۷۰% اول	درصد اشتراک کل
الگوریتم پیاده سازی شده	۹۴۸۶	۹۸/۹۵	۹۳/۳۱	۹۲/۷۷	۹۳/۴۸	۸۹/۴۱
الگوریتم cleanlab	۲۱۲۷۶	۹۷/۱۸	۹۷/۱۹	۹۸/۵۰	۹۸/۴۸	۹۶/۴۷

۴-۳ ایجاد ۲۱۰۰۰ خطا روی مجموعه داده MNIST و انجام پاکسازی داده ها

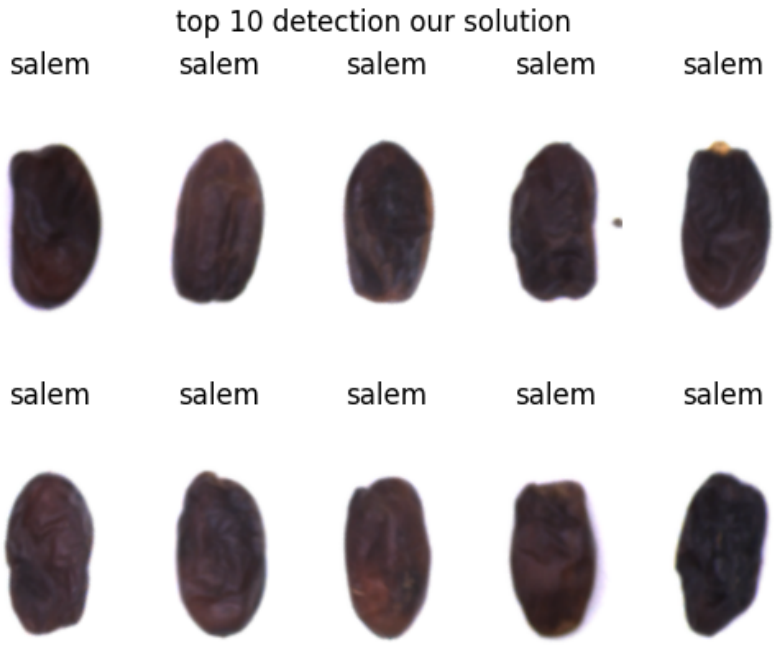
این بخش مشابه بخش قبلی میباشد تنها با این تفاوت که به جای ۷۰۰۰ خطا ۲۱۰۰۰ خطا ایجاد کرده ایم. در جدول ۴-۴ میزان موفقیت هر دو الگوریتم cleanlab و پیاده سازی شده در یافتن این خطاها نوشته شده است. همچنین در شکل ۴-۴نمودار میزان تشخیص درست بر کل تشخیص ها مشابه بخش قبل برای هر دو الگوریتم نمایش داده شده است. همچنین در جدول ۴-۵ تغییر عملکرد مدل بعد از انجام پاکسازی داده ها با هر دو الگوریتم نوشته شده است.

۴-۴ پاکسازی داده ها روی مجموعه داده تصاویر خرما

در این بخش کار پاکسازی داده ها را روی مجموعه داده خرما را انجام داده ایم که نتایج آن در جدول ۴-۶ قابل مشاهده است. البته عملکرد ضعیف cleanlab در این قسمت را میتوان به دلیل حذف بسیاری از داده های مشکوک سالم و مشکوک خراب دانست. همچنین ۱۰ تشخیص برتر هر دو الگوریتم در شکل ۴-۵ و شکل ۴-۶ نمایش داده شده است.



شکل ۴-۴: درصد نسبت تعداد تشخیص درست به کل تشخیص ها با الگوریتم پیاده سازی شده و الگوریتم **cleanlab** در مجموعه داده **MNIST** با ۲۱۰۰۰ خطا



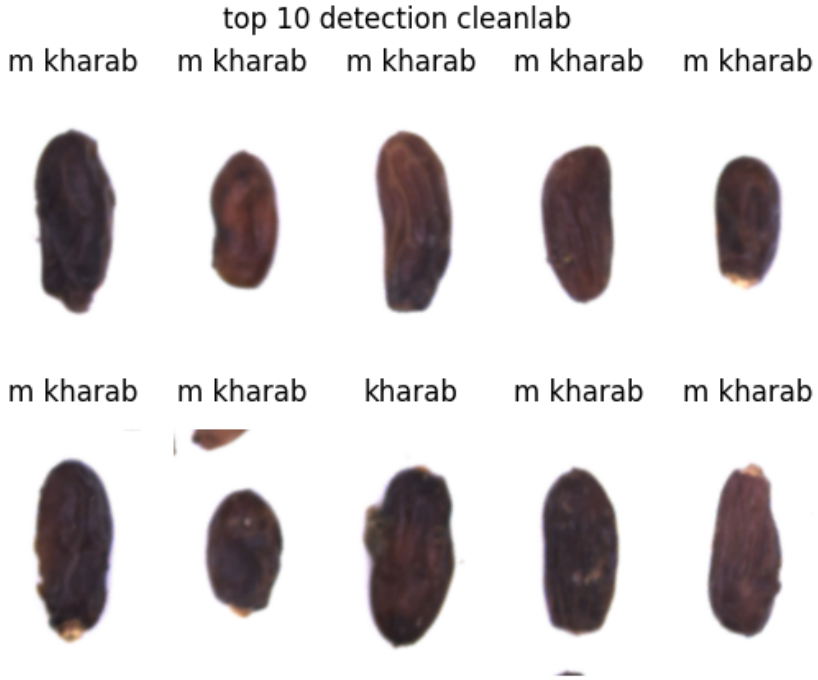
شکل ۴-۵: ۱۰ تشخیص برتر در مجموعه داده خرما با استفاده از روش پیاده سازی شده

جدول ۴-۵: نتایج به دست آمده از پاکسازی داده ها روی مجموعه داده **MNIST** با ۲۱۰۰۰ خطا

	تعداد شبیه	حداقل تکرار	حداقل اطمینان	تعداد تشخیص الگوریتم	دقت تست به ازای کمترین خطای تست	دقت داده آموزشی به ازای کمترین خطای تست	بیشترین دقت داده تست	دقت داده آموزشی به ازای بیشترین دقت تست
قبل از پاکسازی داده ها					۹۸/۴۰	۶۴/۲۳	۹۸/۶۰	۶۴/۰۱
cleanlab				۲۱۲۷۶	۹۸/۸۱	۹۹/۲۴	۹۸/۸۱	۹۹/۲۴
روش پیشنهادی مرحله اول	۵۰۰۰	۴	۵۰	۹۴۸۶	۹۸/۹۱	۷۴/۹۳	۹۸/۹۱	۷۴/۹۳
مرحله دوم	۲۰۰۰	۴	۵۵	۲۸۲۷	۹۸/۶۱	۷۶/۹۸	۹۸/۶۱	۷۶/۹۸
مرحله سوم	۴۰۰۰	۵	۶۰	۱۹۴۲	۹۸/۵۵	۸۰/۱۱	۹۸/۶۸	۷۹/۱۴

جدول ۴-۶: تاثیر پاکسازی داده های مجموعه داده خرما بر دقت مدل

	تعداد شبیه	حداقل تکرار	حداقل اطمینان	تعداد تشخیص الگوریتم	دقت تست به ازای کمترین خطای تست	دقت داده آموزشی به ازای کمترین خطای تست	میانگین دقت کلاس ها
قبل از پاکسازی داده ها					۹۱/۳۵	۹۰/۲۵	۵۵/۴۵
cleanlab				۱۶۱۶	۹۱/۳۵	۹۸/۰۸	۴۹/۴۵
روش پیشنهادی مرحله اول	۵۰۰	۴	۷۵	۳۳۵	۹۱/۳۵	۹۰/۴۳	۶۳/۱۱
مرحله دوم	۵۰۰	۵	۸۵	۱۶۴	۹۱/۸۲	۹۰/۴۰	۵۸/۱۰
مرحله سوم	۵۰۰	۷	۸۵	۸۷	۹۱/۷۱	۸۹/۵۹	۶۱/۹۵



شکل ۴-۶: ۱۰ تشخیص برتر در مجموعه داده خرما با الگوریتم **cleanlab**

فصل ۵

نتیجه‌گیری و کارهای آینده

همانطور که مشاهده شد توانایی الگوریتم پیاده سازی شده، ارتباط بسیار زیادی با کیفیت شبکه siamese دارد و در زمانی که در مجموعه داده خطای زیادی ایجاد شده است، چون عملکرد شبکه siamese نیز ضعیف میشود، توانایی یافتن خطاهای ایجاد شده نیز کم میشود. اما زمانی که خطای مجموعه داده کم است با توجه به اینکه شبکه siamese به خوبی آموزش داده میشود عملکرد روش پیاده سازی شده خوب و تا حدودی بهتر از cleanlab میباشد.

در مجموعه داده تصاویر خرما الگوریتم cleanlb هر چند باعث افزایش بسیار زیاد دقت داده آموزشی شد اما توانایی خوبی در افزایش معیار میانگین دقت کلاس ها نداشت. البته همانطور که ذکر شد این اتفاق به دلیل حذف بیشتر داده های مشکوک سالم و مشکوک خراب بود و شاید اگر همه تشخیص های cleanlab حذف نمیشدند عملکرد cleanlab بهتر از حالت فعلی میشد.

جهت پیشرفت روش پیشنهادی این پروژه راهکارهایی وجود دارد که در بخش های بعدی ذکر خواهند شد.

۵-۱ بهبود شبکه های siamese

مطمئنا یکی از راهکارها جهت افزایش دقت روش پیشنهادی در این پروژه، بهبود عملکرد شبکه های siamese میباشد. برای این کار میتوان نحوه ساخت زوج ها را بهبود بخشید. مثلا درصد چرخش، قرینه کردن و استفاده از زوج مشابه را تغییر داد.

یکی دیگر از کارهای قابل انجام جهت بهبود عملکرد شبکه siamese استفاده از تابع خطای سه گانه [1] میباشد. همچنین در مجموعه داده هایی مثل مجموعه داده تصاویر خرما، میتوان مسئله را به صورت regression در نظر گرفت و مثلا میزان شباهت یک داده سالم به مشکوک سالم را به جای ۰ مقدار ۰/۸ در نظر گرفت.

همچنین مشاهده شد که وجود خطای زیاد در مجموعه داده، باعث افت کارایی شبکه siamese میشود. بنابراین اگر راهکاری جهت آموزش خوب شبکه siamese با تعداد داده های کم اما صحیح ارائه شود میتوان انتظار داشت که عملکرد این روش بهبود یابد.

۵-۲ بهینه سازی پارامترها

همانطور که مشاهده شد راه حل پیشنهادی دارای پارامترهایی چون تعداد شبیه ها، حداقل تکرار و حداقل اطمینان میباشد. اگر این پارامتر ها بسیار سخت گیرانه انتخاب شوند تعداد داده های یافت شده بسیار کم خواهد شد و عملا حذف آن ها از مجموعه داده تاثیر چندانی بر دقت مدل نخواهد داشت. و اگر پارامتر ها اصلا سخت گیرانه نباشند تعداد داده های یافت شده بسیار زیاد خواهد شد و ممکن است تعداد زیادی از داده های درست نیز به عنوان خطا شناخته شوند. بنابراین تعیین درست پارامتر ها در مراحل مختلف الگوریتم میتواند بسیار مهم باشد. جهت پیدا کردن بهترین پارامترها میتوان از روش هایی مثل الگوریتم های ژنتیک استفاده کرد.

[1] triplet

مراجع

[1] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol.29, no.6, pp.141–142, 2012.

[2] S. Hara, A. Nitanda, and T. Maehara, "Data cleansing for models trained with sgd," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol.32, Curran Associates, Inc., 2019.

[3] P. Jeatrakul, K. Wong, and C. Fung, "Data cleaning for classification using misclassification analysis," *JACIII*, vol.14, pp.297–302, 04 2010.

[4] C. Northcutt, L. Jiang, and I. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *Journal of Artificial Intelligence Research*, vol.70, pp.1373–1411, 2021.

[5] S. Benhur, "A friendly introduction to siamese networks," https://towardsdatascience.com/a-friendly-introduction-to-siamese-networks-85ab17522942, 2020. Accessed on 2023-09-07.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.770–778, 2016.