**سیده سحر فیض آبادی**

**Deep-CAPTCHA: یک حل کننده CAPTCHA مبتنی بر یادگیری عمیق برای ارزیابی آسیب پذیری**

**چکیده:**

CAPTCHA یک آزمایش انسان محور است تا یک اپراتور انسانی را از ربات ها ، برنامه های حمله کننده یا سایر عوامل رایانه ای تشخیص دهد که سعی در تقلید از هوش انسان دارد.

در این تحقیق ، ما روشی را برای شکستن آزمونهای CAPTCHA بصری توسط یک راه حل مبتنی بر یادگیری عمیق بررسی می کنیم. هدف از این تحقیق بررسی نقاط ضعف و آسیب پذیری سیستم های مولد CAPTCHA است.

ما برای دستیابی به این هدف یک شبکه عصبی Convolutional به نام Deep CAPTCHA ایجاد می کنیم. سیستم عامل پیشنهادی قادر به بررسی CAPTCHA های عددی و عددی است. برای آموزش و توسعه یک مدل کارآمد ، ما یک مجموعه داده 500000 CAPTCHA برای آموزش مدل خود ایجاد کرده ایم.

1. مقدمه

CAPTCHA ، مخفف تست کاملاً خودکار عمومی تورینگ برای گفتن رایانه ها و انسانها جدا ، یک آزمون رایانه ای برای تمایز بین انسان و روبات است.

در نتیجه ، می توان از CAPTCHA برای جلوگیری از انواع مختلف رفتارهای امنیتی سایبری ، حملات و نفوذ به سمت ناشناس بودن سرویس های وب ، وب سایت ها ، اطلاعات ورود به سیستم یا حتی در وسایل نقلیه نیمه خودکار و سیستم های کمک راننده در صورت نیازهای واقعی انسان برای کنترل کنترل ماشین / سیستم است.

به طور خاص ، این حملات اغلب به موقعیت هایی منجر می شود که برنامه های رایانه ای انسان را جایگزین می کنند ، و این تلاش می کند تا سرویس ها را برای ارسال مقدار قابل توجهی ایمیل ناخواسته ، دسترسی به پایگاه داده ها یا تأثیر در استخرهای آنلاین یا نظرسنجی ها به طور خودکار انجام دهد.

یکی از رایج ترین اشکال حملات سایبری ، حمله DDOS است که در آن سرویس هدف با اطلاعات غیر منتظره یا برای یافتن اعتبارنامه های هدف یا فلج کردن سیستم به طور موقت ، بارگذاری می شود.

یکی از راه حل های کلاسیک و در عین حال بسیار موفق ، استفاده از سیستم CAPTCHA در تکامل سیستم های امنیت سایبری است. بنابراین ، می توان ماشین های حمله را تشخیص داد و برای جلوگیری از آسیب دیدن ، تجهیزات غیر معمول را ممنوع یا نادیده گرفت.

CAPTCHA می تواند به صورت رشته های عددی یا عددی آلفا ، مجموعه های صوتی یا تصویری باشد. شکل 1 چند نمونه از CAPTCHA های عددی آلفا و انواع مختلف آنها را نشان می دهد.



شکل 1: نمونه های مختلف CAPTCHA عددی

یکی از روشهای معمول استفاده از CAPTCHA مبتنی بر متن است. نمونه ای از این نوع سوالات را می توان در شکل 2 مشاهده کرد ، که در آن توالی کاراکترهای تصادفی عددی یا رقمی یا ترکیبی از آنها در یک تصویر پر سر و صدا تحریف و ترسیم می شوند.

تکنیک ها و جزئیات بسیاری برای افزودن نویز و اعوجاج های کارآمد به CAPTCHA وجود دارد تا پیچیده تر شوند.

به عنوان مثال و چندین روش برای افزودن انواع مختلف سر و صدا برای بهبود امنیت برنامه های CAPTCHA مانند افزودن خطوط عبور از حروف به منظور نشان دادن یک طرح ضد تقسیم بندی را توصیه می کند. اگرچه این خطوط نباید از اندازه یک حرف بیشتر باشد.

در غیر این صورت ، با استفاده از الگوریتم تشخیص خط به راحتی می توان آنها را تشخیص داد. مثال دیگر استفاده از انواع مختلف قلم ، اندازه و چرخش در سطح کاراکتر است. یکی از روشهای اخیر در این رابطه را می توان در یافت که رمزنگاری تصویری نام دارد.



شکل 2: نمونه هایی از تصویر CAPTCHA مبتنی بر متن پنج رقمی.

از طرف دیگر ، در هنگام ایجاد CAPTCHA چند نکته مهم وجود دارد که باید از آنها اجتناب کنید. به عنوان مثال ، بیش از حد ارزیابی صداهای تصادفی.

همانطور که امروزه الگوریتم های مبتنی بر دید رایانه دقیق تر و هوشمندانه تر در جلوگیری از سر و صدا در مقابل انسان است. علاوه بر این ، بهتر است از نویسه های بسیار مشابه مانند عدد '0' و حرف 'O' ، حرف 'l' و 'I' که به راحتی توسط کامپیوتر و انسان قابل تفکیک نیستند ، خودداری کنید.

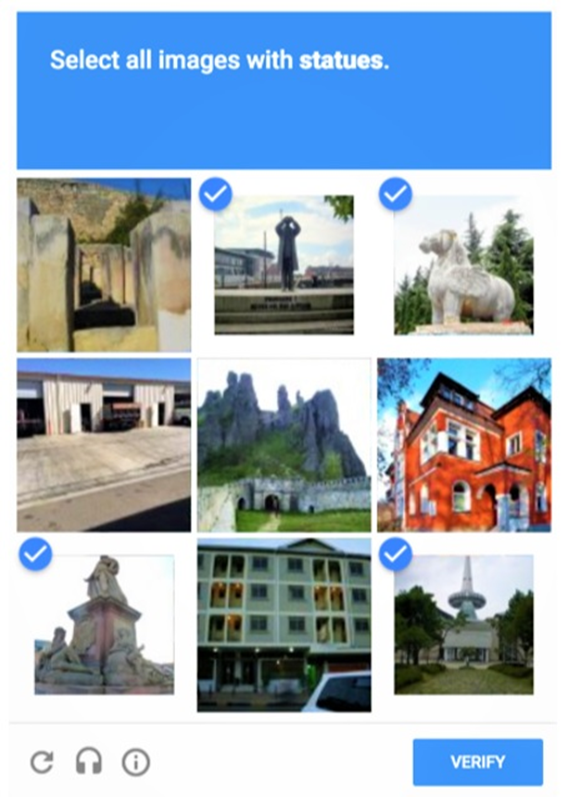
علاوه بر CAPTCHA های متنی ، اخیراً انواع دیگری از CAPTCHA نیز محبوب شده اند. یک مثال می تواند CAPTCHA های مبتنی بر تصویر باشد که شامل تصاویر نمونه ای از اشیا تصادفی مانند علائم خیابان ، وسایل نقلیه ، مجسمه ها یا مناظر است و از کاربر می خواهد یک شی خاص را در میان تصاویر داده شده شناسایی کند.

با این حال ، در این مقاله ، ما به CAPTCHA های متنی متمرکز خواهیم شد زیرا به دلیل هزینه محاسباتی پایین ، این شبکه ها در شبکه های پربازدید و متراکم و وب سایت ها بیشتر دیده می شوند.

قبل از رفتن به بخش بعدی ، ما می خواهیم یک برنامه دیگر از سیستم های CAPTCHA را ذکر کنیم که باید مورد بحث قرار گیرد ، که کاربرد آن در سیستم های OCR (Optical Character Recognition) است.

اگرچه الگوریتم های فعلی OCR بسیار قوی هستند ، اما هنوز هم در شناخت اسکریپت های مختلف دست نوشته یا متن های خراب دارای نقاط ضعفی هستند که استفاده از این الگوریتم ها را محدود می کند.

با استفاده از CAPTCHA یک پیشرفت عالی برای مقابله با چنین مشکلاتی ارائه می شود. از آنجا که محققان سعی می کنند چالش های CAPTCHA را به صورت الگوریتمی حل کنند ، این امر به بهبود الگوریتم های OCR نیز کمک می کند. علاوه بر این ، برخی از محققان دیگر ، مانند Ahn و همکاران.



شکل 3: نمونه ای از CAPTCHA های اخیراً قابل دسترسی است.

یک روش سیستماتیک برای استفاده از این روش پیشنهاد می شود. راه حل پیشنهادی reCAPTCHA نام دارد و فقط یک سیستم CAPTCHA مبتنی بر وب را ارائه می دهد که از متن درج شده برای الگوریتم های OCR خود استفاده می کند.

این سیستم از دو قسمت تشکیل شده است: اول ، مرحله آماده سازی که از دو الگوریتم OCR برای رونویسی مستقل سند استفاده می کند. سپس خروجی ها با هم مقایسه می شوند و سپس قسمت های همسان شده علامت گذاری می شوند که به درستی حل شده اند. و سرانجام ، کاربران کلمات ناسازگار را برای ایجاد یک مجموعه داده چالش CAPTCHA انتخاب می کنند.

این تحقیق سعی دارد تا مسئله شناسایی CAPTCHA را حل کند ، نقاط ضعف و آسیب پذیری رایج آن را تشخیص دهد و فناوری تولید CAPTCHA را بهبود بخشد تا اطمینان حاصل کند که از هوش روزافزون ربات ها و کلاهبرداری ها عقب نخواهد ماند.

2. کارهای مرتبط

در این بخش ، ما برخی از مهمترین و آخرین کارهای انجام شده در این زمینه را بررسی می کنیم.

Geetika Garg و Chris Pollettیک شبکه عصبی عمیق مبتنی بر پایتون برای شکستن CAPTCHA های با طول x انجام دادند. این شبکه از دو لایه Convolutional Maxpool تشکیل شده است و به دنبال آن یک لایه متراکم و یک لایه خروجی Softmax وجود دارد.

این مدل با استفاده از SGD با حرکت Nesterov آموزش داده می شود. همچنین ، آنها مدل خود را با استفاده از لایه های مراجعه به جای لایه های متراکم ساده آزمایش کرده اند. با این حال ، آنها ثابت کردند که استفاده از لایه های متراکم دقت بیشتری در این مسئله دارد.

در کار دیگری که توسط Sivakorn و همکاران انجام شده است. آنها یک سیستم مبتنی بر مرورگر وب برای حل CAPTCHA های تصویر ایجاد کرده اند. سیستم آنها با استفاده از Google Reverse Image Search (GRIS) و سایر ابزارهای منبع باز برای حاشیه نویسی تصاویر و سپس تلاش برای طبقه بندی حاشیه نویسی و پنجمین تصاویر مشابه ، منجر به 83٪ موفقیت در CAPTCHA های تصویر مشابه می شود.

استارک و همکاران همچنین از شبکه عصبی Convolutional برای غلبه بر این مشکل استفاده کرده اند. با این حال ، آنها از سه لایه Convolutional و به دنبال آن دو لایه متراکم و سپس طبقه بندی کنندگان برای حل CAPTCHA های شش رقمی استفاده کرده اند. علاوه بر این ، آنها از روشی برای کاهش اندازه مجموعه داده های آموزشی مورد نیاز استفاده کرده اند.

در تحقیقات انجام شده در مقاله های 4و9 نویسندگان اضافه کردن انواع مختلفی از سر و صدا از جمله سر و صدای خط عبور یا سر و صدا پراکنده مبتنی بر نقطه برای بهبود پیچیدگی و امنیت الگوهای CAPTCHAs.

علاوه بر این ، در مقاله های 11و12و18و31، روش های CNN نیز برای شکستن تصاویر CAPTCHA پیشنهاد شده است. در مقاله 24 برای دستیابی به نتیجه بهتر از CNN با استفاده از روش انتقال سبک استفاده کرده است. در مقاله 29 همچنین در مقایسه با ساختار DenseNet در مقاله 32به جای CNN های رایج ، از CNN با یک تغییر کوچک استفاده کرده است. همچنین در مقاله های 21و33 در مورد CAPTCHA های چینی تحقیق کرده و برای شکستن آنها از یک مدل CNN استفاده کرده اند.

از طرف دیگر ، رویکردهای دیگری نیز وجود دارد که از شبکه های عصبی کانولوشن استفاده نمی کنند ، مانند مقاله 15. آنها از روشهای کلاسیک پردازش تصویر برای حل CAPTCHA استفاده می کنند. به عنوان مثال دیگر ، در مقاله 17 از روش پنجره کشویی برای تقسیم شخصیت ها و شناسایی آنها یک به یک استفاده می کند.

یکی دیگر از تحقیقات جذاب مرتبط الگوریتم تولید خصمانه CAPTCHA است. Osadchy و همکاران در مقاله 16 برای ایجاد طبقه بندی تصویر اصلی در طبقه بندی نادرست تصویر اصلی ، یک نویز خصمانه به یک تصویر اصلی اضافه کنید ، در حالی که این تصویر همچنان برای انسانها یکسان است. در مقاله 25 همچنین از همین روش برای ایجاد تصاویر مبتنی بر متن بهبود یافته استفاده می کند.

به همین ترتیب ، از مقاله هاب 10و26 از مدل های تولیدی و شبکه های خصمانه تولیدی از دیدگاه های مختلف برای آموزش مدل های بهتر و کارآمدتر بر روی داده ها استفاده کنید.

3. روش پیشنهادی

روش های مبتنی بر یادگیری عمیق تقریباً در همه جنبه های زندگی ما از سیستم های نظارتی گرفته تا وسایل نقلیه خودمختار در مقاله 23 ، رباتیک و حتی در چالش جهانی اخیر بیماری همه گیر COVID-19 در مقاله 35 بسیار مورد استفاده قرار می گیرند.

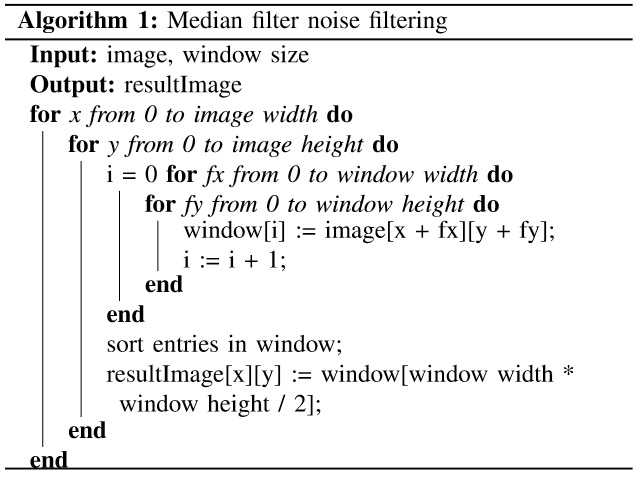
برای حل مشکل CAPTCHA ، ما یک ساختار شبکه عصبی عمیق به نام Deep-CAPTCHA با استفاده از لایه های کانولوشن سفارشی برای نیازهای خود ایجاد می کنیم. در زیر ، ما روش دقیق پردازش ، شناسایی و شکستن تصاویر CAPTCHA عددی را شرح می دهیم. این فرآیند شامل پیش پردازش داده های ورودی ، رمزگذاری خروجی و خود ساختار شبکه است.

3.1 پیش پردازش

استفاده از برخی از عملیات پیش پردازش مانند کاهش اندازه تصویر ، تبدیل فضای رنگ و کاهش نویز ، به طور کلی افزایش عملکرد شبکه بسیار زیاد است.

اندازه اصلی داده های تصویر مورد استفاده در این تحقیق 135×50 پیکسل است که بسیار گسترده است زیرا مناطق خالی زیادی در تصویر CAPTCHA و همچنین پیکسل های همسایه وابسته به کد وجود دارد. مطالعه ما نشان می دهد که با کاهش اندازه تصویر به 67 × 25 پیکسل ، تقریباً می توانیم نتایج مشابهی را بدون کاهش محسوس در عملکرد سیستم بدست آوریم. این کاهش اندازه می تواند به روند آموزش کمک کند تا سریعتر شود زیرا داده ها را کاهش می دهد بدون اینکه در آنتروپی داده ها زیاد کاهش یابد.

تبدیل فضای رنگ به فضای خاکستری روش دیگری پیش پردازش است که ما برای کاهش اندازه داده ها با حفظ همان سطح دقت تشخیص استفاده کردیم. به این ترتیب ، می توانیم مقدار داده های زائد را بیشتر کاهش دهیم و روند آموزش و پیش بینی را آسان کنیم. تبدیل از یک تصویر RGB سه کاناله به یک تصویر در مقیاس خاکستری بر نتایج تأثیر نمی گذارد ، زیرا رنگ در سیستم های CAPTCHA مبتنی بر متن بسیار مهم نیست.

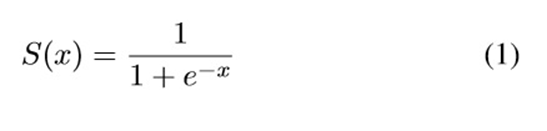


آخرین تکنیک پیش پردازش که در نظر می گیریم استفاده از الگوریتم کاهش نویز است. پس از تجزیه و تحلیل دقیق تجربی در مورد روشهای مناسب برای محاسبه ، تصمیم گرفتیم فیلتر متوسط را برای حذف نویز تصویر ورودی پیاده سازی کنیم. الگوریتم با استفاده از مقدار متوسط مقادیر پیکسل های اطراف به جای خود پیکسل ، نویز تصویر را از بین می برد. الگوریتم در الگوریتم 1 توصیف شده است که در آن ما با استفاده از یک اندازه قبلی پنجره "تصویر" نتیجه تصویر را از ورودی "تصویر" تولید می کنیم.

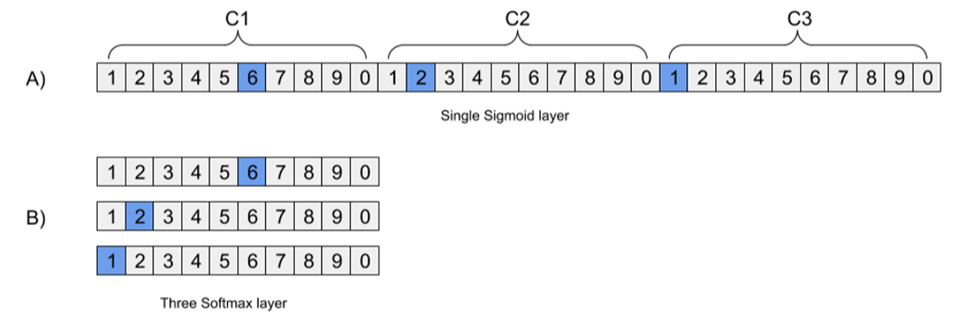
3.2 رمزگذاری

برخلاف مشکلات طبقه بندی که تعداد مشخصی کلاس در مشکلات شناسایی CAPTCHA داریم ، تعداد کلاسها به تعداد ارقام و طول کاراکتر تنظیم شده در CAPTCHA طراحی شده بستگی دارد. این بسته به تعداد طبقات شناسایی شده منجر به رشد نمایی می شود. از این رو ، برای یک مشکل CAPTCHA با پنج رقم عددی ، ما حدود 100000 ترکیب مختلف داریم. در نتیجه ، ما باید داده های خروجی را به در یک شبکه عصبی واحد رمزگذاری کنیم.

رمزگذاری اولیه ای که ما در این تحقیق استفاده کردیم استفاده از نورون های ورودی nb = D × L بود ، جایی که D طول مجموعه حروف الفبا است و L طول مجموعه کاراکترهای CAPTCHA است. این لایه از عملکرد فعال سازی سیگموئید استفاده می کند:



جایی که x مقدار ورودی است و (S (xخروجی تابع Sigmoid است. با افزایش x ، (S (x مکالمه به 1 و با کاهش آن (S (x در حال نزدیک شدن به 1 است. استفاده از عملکرد سیگموئید یک ویژگی غیرخطی را به سلولهای عصبی اضافه می کند که توانایی یادگیری و همچنین پیچیدگی این سلولهای عصبی را در برخورد با ورودی های غیر خطی بهبود می بخشد.



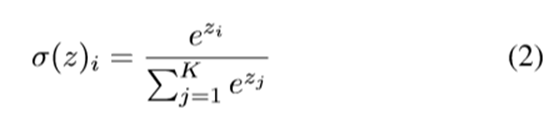
شکل 4: در این فرم ، هر دو روش برای نمایش داده های خروجی نشان داده شده است. روش A)) در این روش ، ما یک لایه Sigmoid منفرد داریم ، که سه شخصیت عددی را نشان می دهد (3 رقمی CAPTCHA ، هر کدام از آنها می تواند نشان دهنده ده وضعیت مختلف باشد - 0 تا. روش (B) ما سه لایه Softmax جداگانه داریم. در این مثال ، هر دو روش 621 را نشان می دهد.

این مجموعه از سلولهای عصبی را می توان به گونه ای تنظیم کرد که اولین مجموعه از سلولهای عصبی D نشان دهنده اولین حرف CAPTCHA باشد. مجموعه دوم سلولهای عصبی D نشان دهنده حرف دوم CAPTCHA و غیره است.

به عبارت دیگر ، با فرض D = 10 ، سلول عصبی پانزدهم می گوید که آیا حرف پنجم از شخصیت دوم با الفبای پیش بینی شده مطابقت دارد یا خیر. یک نمایش بصری در شکل 4.A دیده می شود ، جایی که این روش شامل سه رقم سریال عددی است که 621 را به عنوان خروجی نشان می دهد.

با این حال ، به نظر می رسد این روش به دلیل عدم توانایی در عادی سازی مقادیر عددی و همچنین عدم امکان استفاده از تابع Softmax به عنوان لایه خروجی شبکه عصبی مورد نظر ، شایسته نیست.

بنابراین ، ما لایه های موازی Softmax را به کار بردیم:



که در آن کلاس مربوطه است که Softmax برای آن محاسبه شده است ، zi مقدار ورودی آن کلاس است و K حداکثر تعداد کلاس است.

هر لایه Softmax به طور جداگانه نورون های D را به عنوان شکل 4.B نشان می دهد و این نورون های D نشان دهنده الفبایی است که برای ایجاد CAPTCHA استفاده می شود (به عنوان مثال 0 تا 9 یا A تا Z).

L واحد نشانگر محل رقم در الگوی CAPTCHA است (به عنوان مثال مکان های 1 تا 3). استفاده از این تکنیک به ما امکان می دهد تا هر واحد Softmax را به صورت جداگانه نسبت به سلولهای عصبی خود عادی کنیم. به عبارت دیگر ، هر واحد می تواند وزن خود را نسبت به حروف مختلف نرمال کند. از این رو در کل عملکرد بهتری دارد.

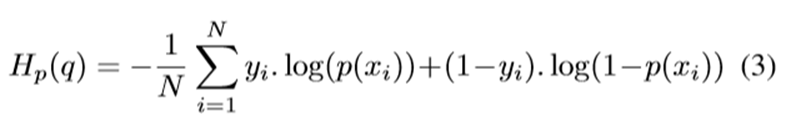
3.3 معماری شبکه

اگرچه شبکه های عصبی راجعه (RNN) می توانند یکی از گزینه های پیش بینی نویسه های CAPTCHA باشند ، در این تحقیق ما بر روی مدل های متوالی متمرکز شده ایم زیرا عملکرد آنها سریعتر از RNN است ، اما اگر مدل به خوبی طراحی شده باشد می توانیم نتایج بسیار دقیق را بدست آوریم.

ساختار شبکه پیشنهادی ما در شکل 5 نشان داده شده است. شبکه با یک لایه Convolutional با 32 نورون ورودی ، عملکرد فعال سازی ReLU و 5 × 5 هسته شروع می شود. یک لایه Max-Pooling 2 × 2 از این لایه پیروی می کند. سپس ، ما دو مجموعه از این جفتهای Convolutional-MaxPooling با همان پارامترها داریم به جز تعداد نورونها که به ترتیب روی 48 و 64 تنظیم شده اند. باید توجه داشته باشیم که تمام لایه های Convolutional پارامتر padding "یکسان" دارند.

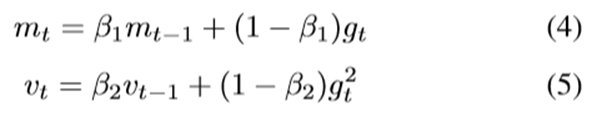
بعد از لایه های Convolutional ، 512 لایه متراکم با عملکرد فعال سازی ReLU و 30٪ نرخ ترک وجود دارد. سرانجام ، ما لایه های Softmax جداگانه L داریم ، جایی که L تعداد کاراکترهای مورد انتظار در تصویر CAPTCHA است.

تابع از دست دادن شبکه پیشنهادی آنتروپی Binarycross است زیرا ما باید این ماتریس های باینری را با هم مقایسه کنیم:

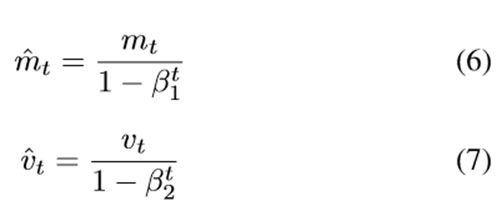


N تعداد نمونه ها و p مدل پیش بینی کننده است. xi و yi به ترتیب داده های ورودی و برچسب نمونه iام را نشان می دهند. از آنجا که برچسب می تواند صفر یا یک باشد ، بنابراین برای هر نمونه فقط یک قسمت از این معادله فعال است.

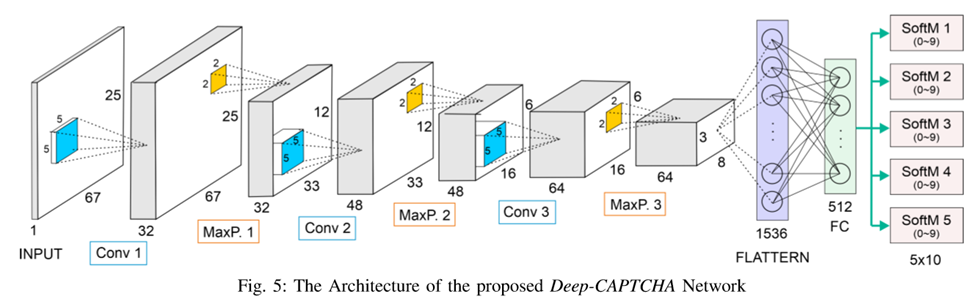
ما همچنین از بهینه ساز آدام استفاده کردیم ، که خلاصه ای از آن در معادلات 4 تا 8 توضیح داده شده است که در آن mt و vt به ترتیب نمایانگر میانگین نمایی از شیب های گذشته و شیب های مربع گذشته هستند.



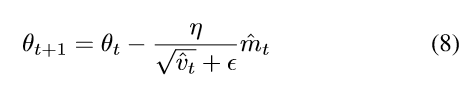
β1 و β2 ثابت های قابل اطمینان هستند. gt شیب تابع بهینه سازی و t تکرار یادگیری است. در معادلات 6 و 7 ، مقادیر لحظه ای برای m و v به شرح زیر محاسبه می شود:



سرانجام ، با استفاده از معادله 8 و با به روزرسانی θt در هر تکرار ، می توان مقدار بهینه عملکرد را بدست آورد. t mt و ˆ vt از طریق معادلات 6 و 7 و η محاسبه می شوند ، اندازه گام (همچنین به عنوان نرخ یادگیری نیز شناخته می شود) در روش ما 0.0001 تنظیم شده است.



شکل 5: معماری شبکه Deep-CAPTCHA پیشنهادی



شهود استفاده از بهینه ساز آدام توانایی آن در آموزش شبکه در یک زمان مناسب است. این را می توان به راحتی از شکل 6 a استنباط کرد که در آن بهینه ساز آدام در مقایسه با Stochastic Gradient Descent (SGD) ، اما با همگرایی بسیار سریعتر ، به همان نتایج دست می یابد.

پس از چندین آزمایش ، ما شبکه را برای 50 دوره با اندازه دسته 128 برای هر یک آموزش دادیم. همانطور که از شکل 6 a استنباط می شود ، حتی بعد از 30 دوره شبکه به سمت همگرایی قابل قبولی گرایش دارد. در نتیجه ، 50 دوره برای عملکرد ثابت شبکه کافی به نظر می رسد. علاوه بر این ، شکل 6 e نیز بر اساس معیارهای دقت اندازه گیری شده ، همین استنتاج را پیشنهاد می کند.

4. نتایج تجربی

پس از توسعه مدل توضیح داده شده در بالا ، ما شبکه را با استفاده از کتابخانه Python ImageCaptcha با استفاده از کتابخانه Python ImageCaptcha در مورد 500000 CAPTCHA تولید شده آموزش دادیم. برای برخی از CAPTCHA های عددی که به طور تصادفی ایجاد شده اند و طول های مشخص ﬁ رقمی دارند ، به شکل 7 مراجعه کنید.

برای متعادل سازی ، مجموعه داده از ده تصویر تصادفی تولید شده از هر جایگزینی برای یک متن پنج رقمی تشکیل شده است.

4.1 تجزیه و تحلیل عملکرد

ما مدل پیشنهادی را روی مجموعه دیگری از نیم میلیون تصویر CAPTCHA به عنوان مجموعه داده آزمایش خود آزمایش کردیم. همانطور که در جدول I نشان داده شده است ، شبکه به عملکرد کلی و میزان دقت 99.33٪ در مجموعه آموزش و 98.94٪ در جدول I رسیده است: متریک دقت و مقدار از دست دادن کل برای بخش آموزش و آزمون مجموعه داده.

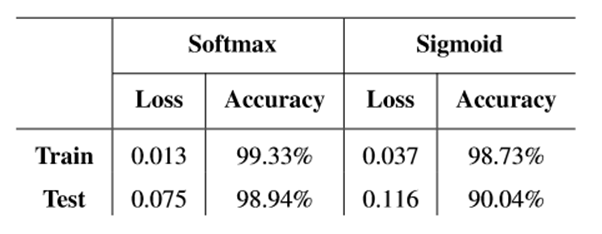
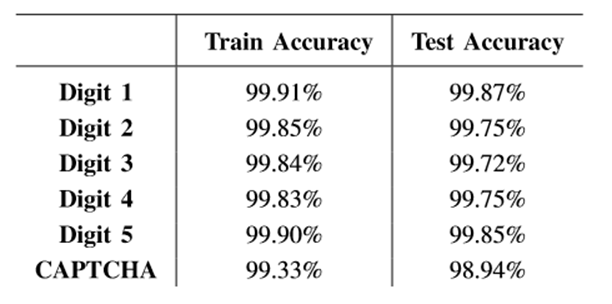


TABLE II: معیار دقت مجموعه داده برای هر رقم و CAPTCHA کامل به عنوان مجموعه ای از 5 رقم یکپارچه.

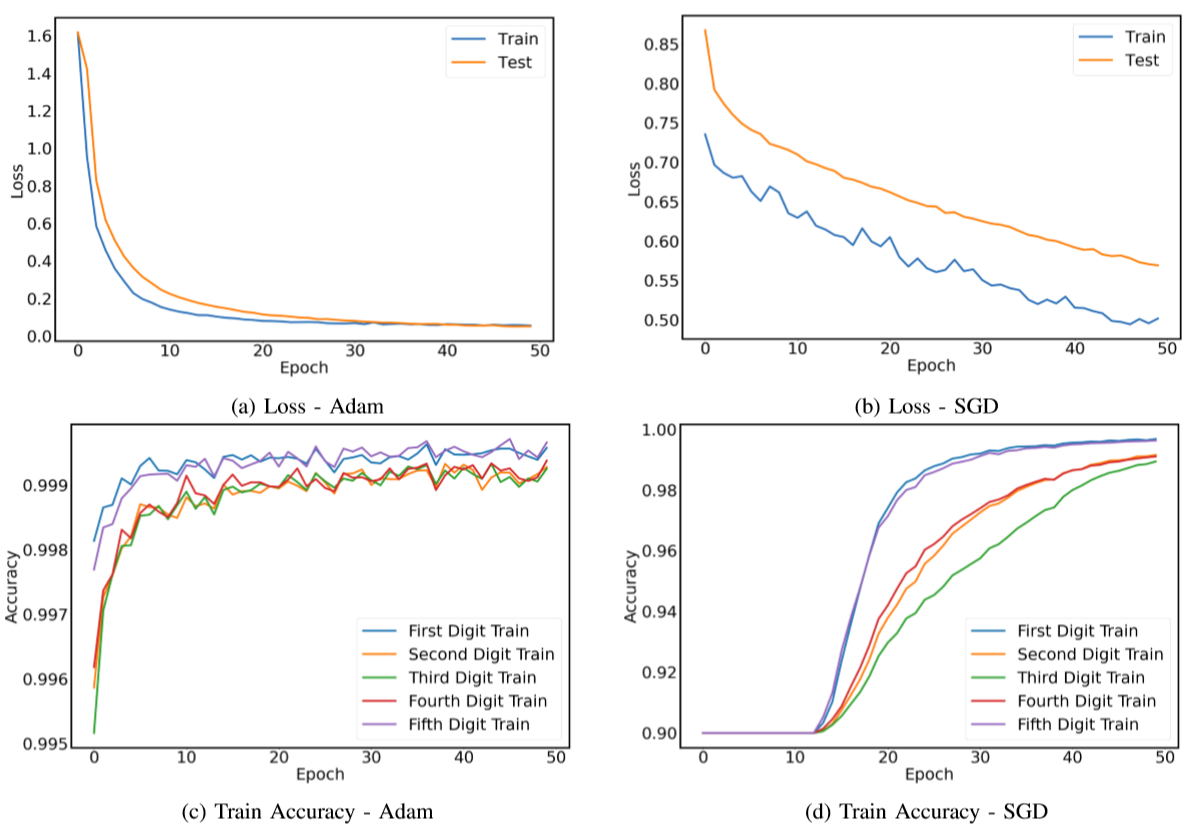


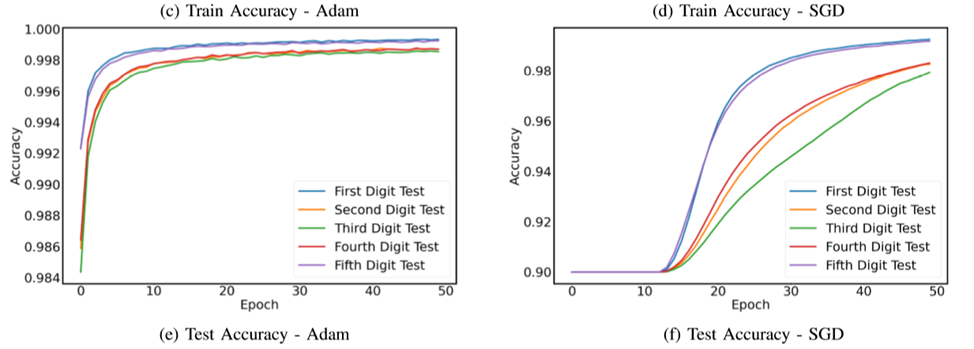
مجموعه داده آزمون. باید توجه داشته باشیم که معیارهای دقت ارائه شده بر اساس تعداد CAPTCHA های به درستی شناسایی شده به عنوان یک کل محاسبه می شوند (یعنی تشخیص صحیح همه پنج رقم منفرد در CAPTCHA داده شده). در غیر این صورت ، همانند جدول II ، دقت ارقام منفرد حتی بیشتر است.

ما همچنین یک بررسی ماتریس سردرگمی را برای تجسم بهتر نتیجه این تحقیق انجام داده ایم. شکل 8 نحوه عملکرد شبکه بر روی هر رقم را بدون در نظر گرفتن موقعیت آن رقم در رشته CAPTCHA نشان می دهد. در نتیجه ، به نظر می رسد که شبکه روی رقم ها بسیار دقیق کار می کند ، با طبقه بندی کمتر از 1٪ برای هر رقم.

4.2 ارزیابی آسیب پذیری

با تجزیه و تحلیل عملکرد شبکه و بازرسی بصری 100 نمونه طبقه بندی اشتباه ، به برخی از نتایج مهم اشاره کردیم که می تواند برای کاهش آسیب پذیری ژنراتورهای CAPTCHA در نظر گرفته شود:





شکل 6: (الف) و (ب): مقادیر از دست دادن روند آزمون و آموزش به ترتیب برای بهینه ساز آدام و SGD. (ج) و (د): معیارهای دقت شبکه برای همان بهینه سازها در مجموعه داده های آموزشی. (e) و (f): معیارهای دقت شبکه با استفاده از بهینه سازهای داده شده در مجموعه داده های آزمون.

در حالی که یک انسان متوسط ​​می تواند اکثر CAPTCHA های طبقه بندی اشتباه را حل کند ، نقاط ضعف زیر در مدل ما مشخص شد که باعث عدم موفقیت توسط حل کننده DeepCAPTCHA شد:

• در 85٪ نمونه های طبقه بندی نشده ، شدت سطح خاکستری CAPTCHA های تولید شده به طور قابل توجهی کمتر از شدت متوسط ​​صدای فلفل توزیع شده گوسی در تصویر CAPTCHA بود.

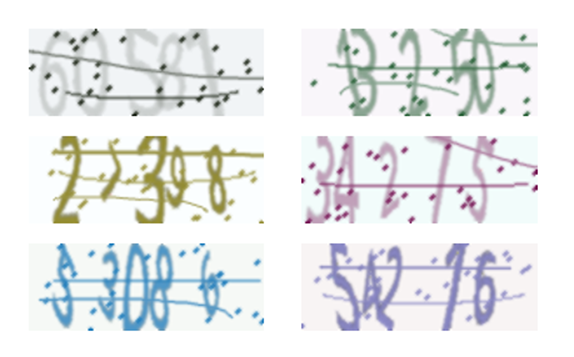
• در 54٪ موارد ، رقم 3 ، 8 یا 9 علت بوده است

طبقه بندی غلط.

• در 81.8٪ موارد ، رقم های طبقه بندی اشتباه برای 10◦ یا بیشتر چرخانده شد.

• سردرگمی بین رقم 1 و 7 نیز یکی دیگر از دلایل خرابی ها بود ، به ویژه در صورت چرخش بیش از 20 درجه خلاف جهت عقب ساعت برای رقم 7.

در نتیجه ، برای مقابله با ضعف و آسیب پذیری موجود در مولدهای CAPTCHA ، ما قاطعانه پیشنهاد می کنیم یک یا برخی از ارقام 3 ، 7 ، 8 و 9 (با چرخش خلاف جهت عقربه های ساعت) را با نرخ بسیار بالاتر تعبیه شده در CAPTCHA های تولید شده در مقایسه با رقم های دیگر. این باعث می شود CAPTCHA ها از نظر الگوریتم های خودکار مانند Deep-CAPTCHA دشوارتر باشند ، زیرا به احتمال زیاد با دیگر ارقام اشتباه گرفته می شوند ، در حالی که مغز انسان هیچ مشکلی در شناسایی آنها ندارد.



شکل 7: نمونه هایی از کتابخانه عددی Image-Captcha پایتون که برای آموزش Deep-CAPTCHA استفاده می شود.

تحقیق مشابهی برای بخش حروف الفبا از شناسایی های ناموفق توسط Deep-CAPTCHA انجام شد و اکثر موارد ناموفق به شخصیت های خیلی گرا یا کسانی که ارتباط نزدیک با شخصیت های همسایه داشتند ، مرتبط بود. به عنوان مثال ، می توان حرف "g" را در برخی از زوایا با "8" اشتباه گرفت ، یا "w" را در هنگام تماس با یک حرف قائم مانند "T" به عنوان "m" طبقه بندی اشتباه کرد. به طور کلی ، حروفی که می توانند با یک / برخی از حروف گره بزنند: w ، v ، m ، n می توانند سناریوی پیچیده ای برای DeepCAPTCHA ایجاد کنند. بنابراین ما درج بیشتر این نامه ها ، و همچنین قرار دادن این نامه ها در نزدیکی نامه های دیگران ، ممکن است باعث تقویت قدرت CAPTCHA شود.

تحقیقات ما همچنین نشان می دهد که رنگهای روشن تر (یعنی شدت مقیاس خاکستری کمتر) نویسه های عددی آلفا نیز به افزایش سطح دشواری CAPTCHA کمک می کنند.

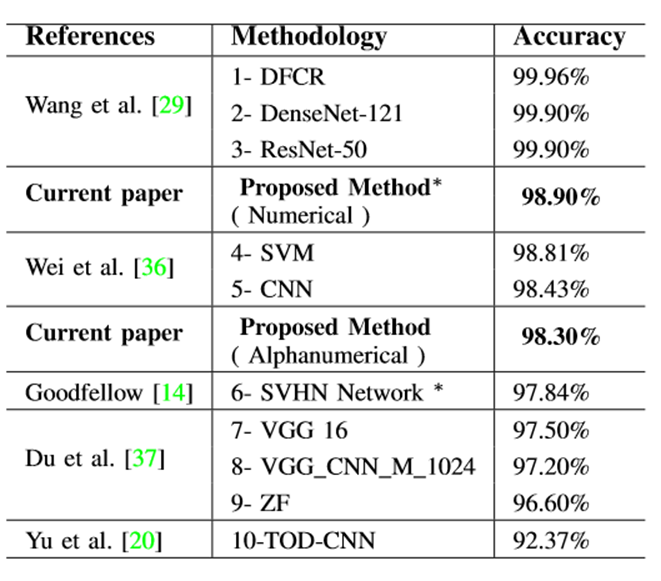
4.3 مقایسه عملکرد

در این بخش ، ما عملکرد روش پیشنهادی خود را با 10 روش پیشرفته دیگر مقایسه می کنیم. نتایج مقایسه در جدول III و به دنبال بحثهای بیشتر در مورد مشخصات هر روش نشان داده شده است.

همانطور که در بخشهای قبلی ذکر شد ، روش ما مبتنی بر شبکه عصبی Convolutional است که دارای سه جفت لایه Convolutional-MaxPool و به دنبال آن یک لایه متراکم است که به مجموعه ای از لایه های Softmax متصل است. در آخر ، این شبکه با Adam optimiser آموزش داده شده است.

در این تحقیق ما در ابتدا بهینه سازی شبکه خود را برای حل CAPTCHA عددی متمرکز کردیم. با این حال ، روشهای موجود در هر دو روش CAPTCHA عددی و عددی کار می کنند ، ما شبکه دیگری را توسعه می دهیم که قادر به حل هر دو نوع است. همچنین ، ما شبکه را روی 700000 CAPTCHA عددی آموزش دادیم. برای مقایسه بهتر و داشتن یک رویکرد سازگارتر ، ما فقط تعداد سلولهای عصبی در هر واحد Softmax را از 10 به 31 افزایش دادیم تا تمام حروف و ارقام رایج لاتین را پوشش دهیم.

جدول III: نتایج صحت روشهای مختلف شناسایی CAPTCHA. (∗ نشان دهنده نتیجه دقتی است که فقط برای CAPTCHA عددی محاسبه می شود.)

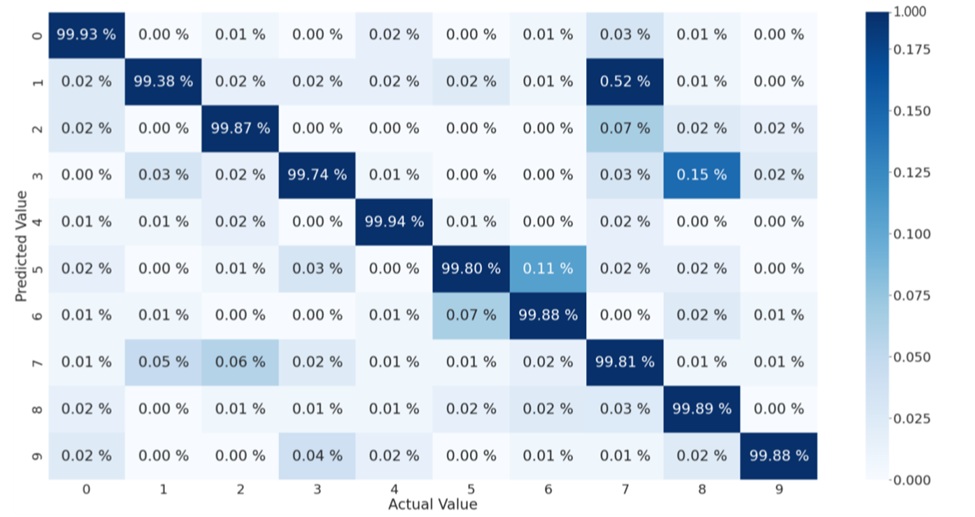


دلیل وجود 31 نورون این است که ما از همه حروف و الفبای لاتین استفاده کرده ایم به جز i ، l ، 1 ، o ، 0 به دلیل شباهت آنها به یکدیگر و دشواری های موجود برای یک انسان متوسط تا آنها را از هم جدا کند. اگرچه ما برای تولید CAPTCHA از هر دو حرف بزرگ و کوچک استفاده کرده ایم ، اما برای هر سادگی فقط یک نورون واحد تعیین می کنیم.

برای مقایسه راه حل ما ، ابتدا تحقیق انجام شده توسط وانگ و همکاران را بررسی کردیم. در مقاله 29 که شامل ارزیابی رویکردهای زیر است: DenseNet-121 و ResNet-50 که مدل تنظیم نشده شبکه های اصلی DenseNet و ResNet برای حل CAPTCHA و همچنین DFCR است که یک روش بهینه شده بر اساس شبکه DenseNet است. DFCR ادعا کرده است که دقت 99.96٪ است که بهترین معیار دقت در بین روشهای دیگر است. با این حال ، این مدل فقط در کمتر از 10000 نمونه و فقط در تصاویر چهار رقمی CAPTCHA آموزش داده شده است. اگر چه مقایسه کمی در جدول ۳ نشان‌دهنده در مقاله 29 در بالای روش پیشنهادی ما است ، اعتبار این روش نه می‌تواند بر روی مجموعه داده‌های بزرگ‌تر تایید شود ، و نه در مجموعه alphanumerical پیچیده با بیش از نیم‌میلیون نمونه ، همانطور که ما در ارزیابی‌های عملکرد خود انجام دادیم.

روش مقایسه بعدی [36] است که از روش مبتنی بر SVM و همچنین پیاده سازی شبکه VGG-16 برای حل مشکلات CAPTCHA استفاده می کند. نکته مهم این روش استفاده از پیش پردازش تصویر ، تقسیم بندی تصویر و تشخیص شخصیت به صورت یک به یک است. این روشها منجر به دقت 98.81٪ در CAPTCHA های عددی چهار رقمی شده است. این شبکه بر روی یک مجموعه داده متشکل از حدود 10،000 تصویر آموزش دیده است. به همین ترتیب ، TOD-CNN [20] علاوه بر استفاده از یک مدل CNN که بر روی یک مجموعه داده 60،000 آموزش داده شده است ، از روش تقسیم بندی برای تعیین مکان کاراکترها استفاده کرده است. این روش برای تقسیم بندی تصویر و کاراکترها از تکنیک Tensor Flow Object Detection (TOD) استفاده می کند.

گودفلو و همکاران [14] برای تشخیص دقیق تر اعداد از CNB استفاده از DistBelief استفاده کرده اند. مجموعه داده مورد استفاده در این تحقیق Street View House Numbers (SVHN) است که شامل تصاویر گرفته شده از Google Street View است.



شکل 8: ماتریس سردرگمی شبکه آموزش دیده بر روی داده های آزمون. اگرچه ارقام به طور دقیق برچسب گذاری شده اند ، نوار عمودی به صورت غیر خطی کوچک می شود تا با تجسم بهتر تفاوت ها را نشان دهد.

سرانجام ، آخرین رویکرد مورد بحث در مقاله 37 است که VGG16 ، VGG CNN M 1024 و ZF را با هم مقایسه می کند. اگرچه در مقایسه با سایر روشها دقت نسبتاً کمی دارند ، اما از روشهای R-CNN برای شناسایی هر شخصیت و تعیین موقعیت همزمان آن استفاده کرده اند.

در نتیجه ، به نظر می رسد که روشهای ما نتایج CAPTCHA عددی و عددی نسبتاً رضایت بخشی دارند. داشتن یک معماری شبکه ساده به ما امکان می دهد تا با سهولت بیشتری از این شبکه برای اهداف دیگر استفاده کنیم. علاوه بر این ، داشتن یک تکنیک تولید خودکار CAPTCHA به ما این امکان را می دهد که با حفظ تشخیص بیشتر CAPTCHA های پیچیده تر و جامع تر نسبت به پیشرفته ترین ، شبکه خود را با دقت بهتری آموزش دهیم.

5. نتیجه گیری

ما یک شبکه عصبی عمیق مبتنی بر CNN را برای تشخیص CAPTCHA مبتنی بر عدد و عددی طراحی ، تنظیم و تنظیم کردیم تا نقاط قوت و ضعف مولدهای مشترک CAPTCHA را نشان دهیم. استفاده از یک سری لایه های Softmax موازی نقش مهمی در بهبود تشخیص دارد. همانطور که در بخش 3.2 توضیح داده شده است ، ما در مقایسه با میزان دقت قبلی 94.04٪ در همان شبکه ، فقط با لایه Sigmoid به حداکثر دقت 98.94٪ رسیده ایم. و جدول I.

اگرچه الگوریتم در CAPTCHA های کاملاً تصادفی بسیار دقیق بود ، اما برخی از حالات خاص ، شکستن آنها را برای Deep-CAPTCHA بسیار چالش برانگیز می دانست. ما اعتقاد داریم که در نظر گرفتن موارد مطرح شده می تواند به ایجاد نمونه های قابل اعتماد و قوی CAPTCHA کمک کند که باعث پیچیده تر شدن و احتمال ترک خوردن توسط ربات ها یا موتورهای ترک خورنده AIbased و الگوریتم های آن می شود.

به عنوان یک مسیر بالقوه برای کارهای آینده ، ما پیشنهاد می کنیم CAPTCHA ها را با طول شخصیت متغیر حل کنید ، نه تنها به نویسه های عددی محدود می شود بلکه برای شخصیت های عددی عددی آلفا به چالش کشیده می شود که در بخش 4 بحث شده است. ما همچنین توصیه می کنیم تحقیقات بیشتری در مورد کاربرد شبکه های عصبی راجعه و همچنین روش های کلاسیک پردازش تصویر در مقاله 30 برای استخراج و شناسایی شخصیت های CAPTCHA ، به صورت جداگانه.

منابع

[1] Garg, Geetika, and Chris Pollett. ”Neural network captcha crackers.” In 2016 Future Technologies Conference (FTC), pp. 853-861. IEEE, 2016.

[2] Sivakorn, Suphannee, Iasonas Polakis, and Angelos D. Keromytis. ”I am robot:(deep) learning to break semantic image captchas.” In 2016 IEEE European Symposium on Security and Privacy (Euro S&P), pp. 388-403. IEEE, 2016.

[3] Stark, Fabian, Caner Hazrbas, Rudolph Triebel, and Daniel Cremers. ”Captcha recognition with active deep learning.” In Workshop new challenges in neural computation, vol. 2015, p. 94. Citeseer, 2015.

[4] Bostik, Ondrej, and Jan Klecka. ”Recognition of CAPTCHA characters by supervised machine learning algorithms.” IFAC-PapersOnLine 51, no. 6 (2018): 208-213.

[5] Von Ahn, Luis, Manuel Blum, Nicholas J. Hopper, and John Langford. ”CAPTCHA: Using hard AI problems for security.” In International Conference on the Theory and Applications of Cryptographic Techniques, pp. 294-311. Springer, Berlin, Heidelberg, 2003.

[6] Von Ahn, Luis, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. ”recaptcha: Human-based character recognition via web security measures.” Science 321, no. 5895 (2008): 1465-1468.

[7] Kaur, Kiranjot, and Sunny Behal. ”Designing a Secure Text-based CAPTCHA.” Procedia Comput. Sci 57 (2015): 122-125.

[8] Nazario, Jose. ”DDoS attack evolution.” Network Security 2008, no. 7 (2008): 7-10.

[9] Yousef, Mohamed, Khaled F. Hussain, and Usama S. Mohammed. ”Accurate, data-efﬁcient, unconstrained text recognition with convolutional neural networks.” arXiv preprint arXiv:1812.11894 (2018).

[10] Ye, Guixin, Zhanyong Tang, Dingyi Fang, Zhanxing Zhu, Yansong Feng, Pengfei Xu, Xiaojiang Chen, and Zheng Wang. ”Yet another text captcha solver: A generative adversarial network based approach.” In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 332-348. 2018.

[11] Karthik, CHBL-P., and Rajendran Adria Recasens. ”Breaking microsofts CAPTCHA.” Technical report (2015).

[12] Kopp, Martin, Matej Nikl, and Martin Holena. ”Breaking captchas with convolutional neural networks.” ITAT 2017 Proceedings (2017): 93-99.

[13] Rezaei, M. and Klette, R. ”Look at the Driver, Look at the Road: No Distraction! No Accident!”, CVF Computer Vision and Pattern Recognition, pp. 129–136, (2014).

[14] Goodfellow, Ian J., Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. ”Multi-digit number recognition from street view imagery using deep convolutional neural networks.” arXiv preprint arXiv:1312.6082 (2013).

[15] Wang, Ye, and Mi Lu. ”An optimized system to solve text-based CAPTCHA.” arXiv preprint arXiv:1806.07202 (2018).

[16] Osadchy, Margarita, Julio Hernandez-Castro, Stuart Gibson, Orr Dunkelman, and Daniel Prez-Cabo. ”No bot expects the DeepCAPTCHA! Introducing immutable adversarial examples, with applications to CAPTCHA generation.” IEEE Transactions on Information Forensics and Security 12, no. 11 (2017): 2640-2653.

[17] Bursztein, Elie, Jonathan Aigrain, Angelika Moscicki, and John C. Mitchell. ”The end is nigh: Generic solving of text-based captchas.” In 8th USENIX Workshop on Offensive Technologies (WOOT 14). 2014.

[18] Zhao, Nathan, Yi Liu, and Yijun Jiang. ”CAPTCHA Breaking with Deep Learning.” (2017).

[19] Chen, Jun, Xiangyang Luo, Yanqing Guo, Yi Zhang, and Daofu Gong. ”A survey on breaking technique of text-based CAPTCHA.” Security and Communication Networks 2017 (2017).

[20] Yu, Ning, and Kyle Darling. ”A Low-Cost Approach to Crack Python CAPTCHAs Using AI-Based Chosen-Plaintext Attack.” Applied Sciences 9, no. 10 (2019): 2010.

[21] Algwil, Abdalnaser, Dan Ciresan, Beibei Liu, and Jeff Yan. ”A security analysis of automated Chinese turing tests.” In Proceedings of the 32nd Annual Conference on Computer Security Applications, pp. 520-532. 2016.

[22] Sivakorn, Suphannee, Jason Polakis, and Angelos D. Keromytis. ”Im not a human: Breaking the Google reCAPTCHA.” Black Hat (2016).

[23] Rezaei, Mahdi, and Reinhard Klette, ”Simultaneous Analysis of Driver Behaviour and Road Condition for Driver Distraction Detection.”, International Journal of Image and Data Fusion, (2011).

[24] Kwon, Hyun, Hyunsoo Yoon, and Ki-Woong Park. ”CAPTCHA Image Generation Using Style Transfer Learning in Deep Neural Network.” In International Workshop on Information Security Applications, pp. 234– 246. Springer, Cham, 2019.

[25] Kwon, Hyun, Yongchul Kim, Hyunsoo Yoon, and Daeseon Choi. ”Captcha image generation systems using generative adversarial networks.” IEICE TRANSACTIONS on Information and Systems 101, no. 2 (2018): 543–546.

[26] George, Dileep, Wolfgang Lehrach, Ken Kansky, Miguel Lzaro-Gredilla, Christopher Laan, Bhaskara Marthi, Xinghua Lou et al. ”A generative vision model that trains with high data efﬁciency and breaks text-based CAPTCHAs.” Science 358, no. 6368 (2017): eaag2612.

[27] Rezaei, M. and Sarshar, M., and Sanaatiyan MM/ ”Toward next generation of driver assistance systems: A multimodal sensor-based platform.” Computer and Atomation Eengineering, (2010), pp. 62–67.

[28] Yan, Xuehu, Feng Liu, Wei Qi Yan, and Yuliang Lu. ”Applying Visual Cryptography to Enhance Text Captchas.” Mathematics 8, no. 3 (2020): 332.

[29] Wang, Jing, Jiao Hua Qin, Xu Yu Xiang, Yun Tan, and Nan Pan. ”CAPTCHA recognition based on deep convolutional neural network.” Math. Biosci. Eng. 16, no. 5 (2019): pp. 5851–5861. [30] Rezaei, Mahdi, and Reinhard Klette. ”Object Detection, Classiﬁcation, and Tracking”. Springer International Publishing, 2017.

[31] Stark, Fabian, Caner Hazrbas, Rudolph Triebel, and Daniel Cremers. ”Captcha recognition with active deep learning.” In Workshop new challenges in neural computation, vol. 2015, p. 94. Citeseer, 2015.

[32] Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. ”Densely connected convolutional networks.” In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700–4708. 2017.

[33] Jia, Yang, Wang Fan, Chen Zhao, and Jungang Han. ”An Approach for Chinese Character Captcha Recognition Using CNN.” In Journal of Physics: Conference Series, vol. 1087, no. 2, p. 022015. IOP Publishing, 2018.

[34] Banday MT, Shah NA, ”Image Flip Captcha”. ISC International Journal of Information Security (ISeCure) vol.1, no. 2, p. 105–123

[35] Rezaei M, Shahidi M, ”Zero-Shot Learning and its Applications from Autonomous Vehicles to COVID-19 Diagnosis: A Review”. In arXiv prepreint arXiv:2004.14143 (2020).

[36] Wei, Li, Xiang Li, TingRong Cao, Quan Zhang, LiangQi Zhou, and WenLi Wang. ”Research on Optimization of CAPTCHA Recognition Algorithm Based on SVM.” In Proceedings of the 2019 11th International Conference on Machine Learning and Computing, pp. 236-240. 2019.

[37] Du, Feng-Lin, Jia-Xing Li, Zhi Yang, Peng Chen, Bing Wang, and Jun Zhang. ”CAPTCHA Recognition Based on Faster R-CNN.” In International Conferenceon Intelligent Computing,pp.597-605.Springer, Cham, 2017.

[38] Python Image Captch Librady https://github.com/lepture/captcha, Last access: 15 June 2020.