دانشگاه شهید بهشتی

دانشکده مهندسی و علوم کامپیوتر

یادگیری الگوی دستنویس ارقام
با کمک بازوهای رباتیکی

پروژه کارشناسی

دانشجو:

آریا پرویزی

استاد راهنما:

آقای دکتر آرمین سلیمی بدر

تابستان ۱۴۰۱

چکیده

در این پروژه، هدف ترسیم خطوط و اشکال توسط یک ربات است بدون اینکه به ربات نحوه‌ی دقیق رسم شکل را بدهیم. این پروژه می‌تواند بسیار حائز اهمیت باشد و کاربردهایی در صنعت برشکاری، جوشکاری، نقشه‌کشی و طراحی صنعتی داشته باشد. برای سادگی ما ابتدا رسم ارقام ۰ تا ۹ در محیط شبیه‌سازی به کمک یک بازوی رباتیکی را در نظر گرفتیم. منظور از ترسیم، کشیدن خطوط به صورت پیوسته است که با کار چاپگرها تفاوت دارد. در این پایان‌نامه برای پیدا کردن الگوی ترسیم از الگوریتم‌های تکاملی استفاده شد و نتایج در محیط شبیه‌سازی ویباتز توسط بازوی رباتیک irb4600 ترسیم گردید. روش‌های دیگر نیز در این پایان‌نامه مطالعه شد و مزایا و معایب هر کدام مورد بررسی قرار گرفت. در کارهای انجام شده قبلی از روش‌های متعددی استفاده است. ابتدایی‌ترین روش احتمالا چاپگرها هستند که سطر به سطر رسم می‌کنند. مزیت الگوریتم ما در این است که به ما اجازه می‌دهد بدون آموزش قبلی و نیاز به داده‌های حجیم، هر شکل دلخواهی را توسط ربات رسم کند. این اولین قدم در راستای طراحی ربات‌های طراح خواهد بود.

واژگان کلیدی: ربات طراح با قلم، بازوهای رباتیکی، الگوی شکل‌ها، ترسیم شکل‌های ساده، الگوریتم‌های تکاملی

سپاسگزاری

در این قسمت به دلیل علاقه‌ی وافری که به موضوعات رباتیک، هوش مصنوعی، ریاضیات، آمار، فیزیک و ... داشتم، برای من بهترین و جذاب‌ترین شاخه از رشته‌های مهندسی، رباتیک است. از این رو از صبوری و راهنمایی‌های ارزشمند استاد راهنمایم آقای دکتر سلیمی سپاسگزاری و قدردانی می‌نمایم. و همچنین از حمایت‌ها و پیشتیبانی پدر و مادرم تشکر می‌کنم.

آریا پرویزی
تابستان ۱۴۰۱

فهرست مطالب

فصل اول: کلیات ۱

۱-۱ مقدمه ۲

۱-۲ بیان مسئله ۲

۱-۲-۱ پیشینه تحقیق ۳

۱-۳ کلیات روش پیشنهادی ۴

۱-۳-۱ تولیدکننده الگوی رقم ۴

۱-۳-۲ تشخیصگر الگوی رقم ۵

۱-۳-۳ رسم و پیمایش الگوی تولید شده ۵

۱-۳-۴ ارزیابی ۶

۱-۳-۵ ملاحظات دنیای واقعی ۶

۱-۴ ساختار پروژه ۷

فصل دوم: مفاهیم پایه و کارهای مرتبط ۸

۲-۱ مقدمه ۹

۲-۲ مفاهیم پایه ۹

۲-۲-۱ بررسی نرم‌افزارهای استفاده شده ۹

شبیه‌ساز ویباتز ۹

زبان برنامه نویسی پایتون ۹

۲-۲-۲ بررسی بازوهای رباتیکی ۱۰

۲-۲-۳ توصیف محیط ربات در شبیه‌ساز ۱۱

۲-۳ تحلیل کارهای مرتبط و معماری پروژه ۱۲

۲-۳-۱ معماری اول ۱۲

۲-۳-۲ معماری دوم ۱۳

۲-۳-۳ بخش ورودی ۱۳

۲-۳-۴ بخش تولید الگو ۱۳

۲-۳-۵ بخش ترسیم الگو ۱۵

۲-۳-۶ بخش ارزیابی خروجی ۱۶

فصل سوم: روش پیشنهادی و نتیجه‌گیری ۱۸

۳-۱ نسخه اولیه (پروتوتایپ) ۱۹

۳-۱-۱ ورودی ۱۹
۳-۱-۲ تشخیص عکس ورودی ۱۹
۳-۱-۳ تولید الگو به روش الگوریتم‌های صریحانه و ابتکاری ۲۰
۳-۱-۴ اشکالات الگوریتم پارامتری‌سازی زمانی ۲۲
۳-۱-۵ جمعبندی الگوریتم پارامتری‌سازی زمانی ۲۴
۳ – ۱- ۶ ترسیم الگوی تولید شده ۲۵
۳ – ۱ -۷ تبدیلات هندسی سیگنال ورودی ۲۵
۳-۱-۸ نتایج حاصل از پروتوتایپ ۲۵
۳-۲ پیاده‌سازی روش اصلی: الگوریتم ژنتیک ۲۶
۳-۳ نتایج ۲۸
۳-۴ جمعبندی و پیشنهادها ۳۰
منابع ۳۱
پیوست ۳۲
الف) فایل‌های حاوی متن پروژه ۳۲

فهرست شکل‌ها

شکل ۱-۱- مراحل طراحی یک چهره برای یک نقاش. ۳
شکل ۲-۱– مختصات ربات در محیط شبیه‌ساز، به همراه نمایش دستگاه مختصات ربات و دنیای شبیه‌سازی ۱۱
شکل ۲-۲– ربات irb4600 از شرکت abb ۱۲
شکل ۲-۳ – معماری مدل مارکوف ۱۲
شکل ۲-۴ – معماری تجزیه شده ۱۳
شکل ۲-۵ – سمت چپ الگوریتم رسم به نزدیکترین نقطه و سمت راست خروجی. ۱۵
شکل ۲-۶ – اجزای بخش‌های تولید الگو و ترسیم الگو ۱۵
شکل ۲-۷ – معماری پیشنهادی سوم ۱۶
شکل ۳-۱ – اشکال دستی ارقام ۱۹
شکل ۳-۲ – نمودار نتایج شبکه عصبی، دقت و تابع ضرر بر حسب هر ایپاک. ۲۰
شکل ۳-۳ – آخرین نقاط انتهایی پیدا شده در هر رقم ۲۲
شکل ۳-۴ – نتایج حاصل از dilation برای ارقام صفر تا ۹ ..... ۲۲
شکل ۳-۵ – تصاویر نازک و کلفت شده‌ی ارقام. ۲۳
شکل ۳-۶ - سه نوع فیلتر. ۲۳
شکل ۳-۷ – نتایج حاصل کم کردن رزولوشن از ۳۰۰*۳۰۰ به ۱۰*۱۰ ..... ۲۴
شکل ۳-۸ – مشکل در رقم ۶ ..... ۲۴
شکل ۳-۹ – نمودار بلوکی تبدیلات مختصات از شکل ورودی تا خروجی روی کاغذ ۲۵
شکل ۳-۱۰ – نتایج حاصل کشیدن ارقام پروتوتایپ روی کاغذ توسط ربات ۲۵
شکل ۳-۱۱ – نتایج حاصل کشیدن ارقام پروتوتایپ روی کاغذ بعد از اصلاح فواصل نقاط متوالی ۲۶
شکل ۳-۱۲ – توضیح روش عملکرد کروموزوم ۲۸
شکل ۳-۱۳ – شکل‌های تخمینی حاصل از الگوریتم ژنتیک ۲۸
شکل ۳-۱۴ – نمودارهای میانگین امتیاز و بهترین امتیاز در هر نسل ۲۹
شکل ۳-۱۵ – میانگین امتیاز در ده بار اجرای الگوریتم ژنتیک. ۲۹
شکل ۳-۱۶ – انحراف معیار برای ده بار اجرا بر حسب تابعی از نسل‌ها ۳۰
شکل ۳-۱۷ – نتیجه نهایی ارقام ترسیم شده توسط ربات در محیط شبیه‌سازی ۳۰

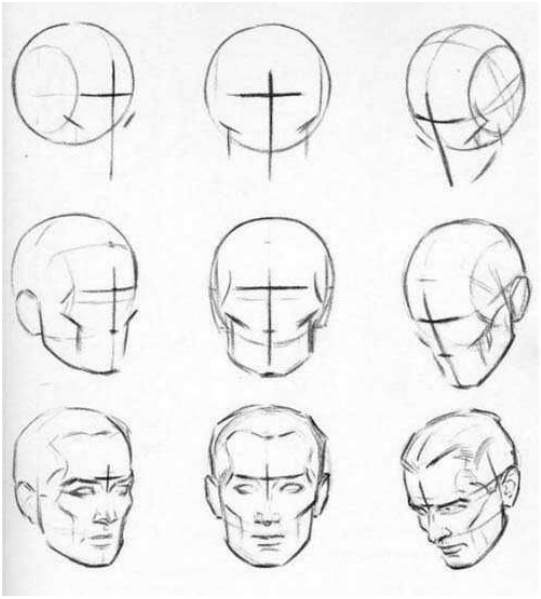
فصل اول: کلیات

۱-۱ مقدمه

امروزه رباتیک کاربردهای فراوانی در جامعه و صنعت پیدا کرده است. در صنعت بازوهای رباتیکی برای بالا بردن سرعت و دقت و اتوماسیون کارهای صنعتی بسیار مناسب و پرکاربرد هستند. از طرف دیگر وجه تمایز مهم انسان با حیوانات دیگر در داشتن بازو و دست توانا برای انجام کارهای ظریف و پیچیده است. انسان با دست‌های توانای خود ابزار را بکار گرفته و اختراعات فراوانی تولید کرده است که در سرنوشت بشر و پیشرفت آن بسیار تاثیرگذار بوده است. در این بین نوشتن خط و حروف و همچنین هنر نقاشی و طراحی در انتقال احساسات و اطلاعات و افزایش توانایی ارتباطات بین افراد و حتی نسل‌های مختلف نقش بسیار مهمی داشته است. طراحی و خطاطی تلفیقی از مهارت دست و ذهن خلاق انسان است. در راستای اختراع ربات‌هایی که هر چه بیشتر شبیه انسان باشند، و شبیه انسان رفتار کنند، بسیار جالب خواهد بود که تمام کارهایی که انسان می‌تواند انجام دهد را آن‌ها نیز بتوانند انجام دهند. در این راستا برای برداشتن قدمی در جهت نزدیک کردن رفتار ربات‌ها به انسان‌ها، می‌خواهیم با تمرکز بر مهارت کشیدن و ترسیم اشکال، این توانایی را بر روی ربات‌ها پیاده‌سازی کنیم. در این پروژه، هدف ترسیم خطوط و شکل‌ها توسط یک ربات بدون اینکه به ربات نحوه‌ی دقیق رسم شکل را بدهیم. این پروژه می‌تواند بسیار حائز اهمیت باشد و کاربردهایی در صنعت برشکاری، جوشکاری، نقشه‌کشی و طراحی صنعتی داشته باشد. همانطور که از عنوان پروژه نیز مشخص است، ما تمرکز را بر رسم ارقام ۰ تا ۹ انگلیسی گذاشتیم تا از پیچیدگی‌های فراوان احتمالی بکاهیم. به دلیل در دسترس نبودن بازوی رباتیکی و هزینه‌های فراوان، تصمیم گرفتیم که این کار را در محیط شبیه‌سازی انجام دهیم. منظور از ترسیم، کشیدن خطوط به صورت پیوسته است که متفاوت با کار چاپگر می‌باشد. در این پایان‌نامه برای پیدا کردن الگوی ترسیم از الگوریتم‌های تکاملی استفاده شد و نتایج در محیط شبیه‌سازی ویباتز توسط بازوی رباتیک irb4600 ترسیم گردید. روش‌های دیگر نیز در این پایان‌نامه مطالعه شد و مزایا و معایب هر کدام مورد بررسی قرار گرفت. در کارهای انجام شده قبلی از روش‌های متعددی استفاده است. ابتدایی‌ترین روش احتمالا چاپگرها هستند که سطر به سطر رسم می‌کنند. مزیت الگوریتم ما در این است که به ما اجازه می‌دهد بدون آموزش قبلی و نیاز به داده‌های حجیم، هر شکل دلخواهی را توسط ربات رسم کند. این اولین قدم در راستای طراحی ربات‌های طراح خواهد بود.

۱-۲ بیان مسئله

چیزی که در این پروژه هدف اصلی و بسیار مهم است، تولید یک الگوی مستقل از تصویر ورودی با روش‌های ممکن که ربات بتواند از آن برای رسم استفاده کند. بدون این الگو و صرفا استفاده ربات از تصویر ورودی در فضای حافظه‌ی خود، می‌تواند بدون نوآوری و تنها با پیمایش سطر به سطر این عکس، همانند هزاران چاپگر موجود در بازار، به چاپ عکس بر روی کاغذ بپردازد. که البته چاپگرها نیز مسئله‌ها و چالش‌های بسیاری دارند اما برای این پروژه چیزی که حائز اهمیت است، داشتن رویکردی مشابه به مغز انسان است که می‌تواند ارقام و اشکال و حتی تصاویر بسیار پیچیده تر را بدون داشتن تصویر کامل و مستقیم در ذهن با استفاده تشابه‌سازی به اشکال ساده‌تر و ساختن یک الگو برای کشیدن، رسم و طراحی کند. بنابراین چاپ سطر به سطر و سیاه کردن کاغذ مطابق عکس ورودی، مورد نظر ما نیست و یافتن مدل و الگوی کشیدن وابسته به ماهیت تصویر ورودی، بسیار اهمیت دارد.



شکل ۰-۱- مراحل طراحی یک چهره برای یک نقاش. برگرفته از مرجع[۱]

هدف پیاده‌سازی مدلی هوشمند است که با توجه به یک عکس، بتواند خطوط آن را با شکلهایی با مفهوم یکسان استخراج و بازتولید کند. همچنین قوانینی برای یافتن مسیر همیلتونی اشکال یا به عبارت دیگر ترتیب و مدلی برای نحوه کشیدن بتواند پیدا کند. در این کاربرد بخصوص و برای ساده‌سازی اولیه بر روی ارقام انگلیسی تمرکز کرده و بعد قابل بسط بودن مدل به اشکال عمومی، حروف فارسی، انگلیسی، چینی و حتی دیگر اشکال پیچیده را بررسی خواهیم کرد. در قسمت معماری پروژه به بررسی چند روش ممکن برای رسیدن به این اهداف می‌پردازیم.

۱-۲-۱ پیشینه تحقیق

در مرجع [۲]، با استفاده از تکنولوژی انتقال سبک با شبکه‌های عصبی[۱]، ویژگی‌های محتوایی و سبکی تصویر را استخراج کرده و با بازترکیب آن‌ها، اثری جدید را به وجود می‌آورند. همچنین ادعا کرده‌اند با استفاده از اضافه کردن لامسه توانسته‌اند حس ترسیم کردن انسان را شبیه‌سازی کنند.

مرجع [۳]، با بکارگیری بازخورد میزان حس لامسه از حسگر بار-سلولی[۲] توانسته است بر فشار مداد که بر کاغذ وارد می‌شود، کنترل انجام دهد و با تغییر فشار مداد سایه‌های متفاوتی از خط را در طول ترسیم خود به وجود آورد.

[۱] neural style transfer technology

[۲] load-cell sensor

یکی از ربات‌های جالب که می‌تواند نقاشی را به سبک‌های متنوعی بکشد، ربات e-David است [۴]. این ربات از دو روش برای الگوی ترسیم استفاده می‌کند. در روش اول مسیر خطوط از قبل تعیین شده، به آن داده می‌شود. در روش دوم الگوی ترسیم به صورت پویا از روش پیچشی انتگرال خطی[۳] در گرادیان تصویر[۵] بدست می‌آید. این الگو شامل خطوطی است که توصیف‌گر اصلی تصویر است.

همچنین با الهام از ربات e-David رباتی طراحی شده است که می‌تواند یک درخت یا یک گیاه را ترسیم کند. اما لزوما به این معنی نیست که مطابق یک عکس دلخواه داده شده ترسیم را انجام دهند. این ربات از الگوریتم فراکتالی برای تولید شاخه‌های درخت استفاده می‌کند [۶].

آدامیک[۴] و همکاران [۷] با استفاده از روش بردارسازی تصویر[۵] و بکارگیری الگوریتم ژنتیک مقادیر حقیقی[۶] [۸] تصویر داده شده را به صورت پاره‌خط‌های مجزایی با تعداد مشخص تجزیه کرده و این پاره‌خط‌ها را با ربات ترسیم می‌نماید به نحوی که شکل حاصل، به تصویر اصلی بسیار شبیه خواهد بود. تاکید این مقاله بر روش‌های سریع کشیدن نقاشی به کمک مداد برای تصاویر واقعگرایانه[۷] است. این مقاله مشابه‌ترین کار انجام شده به کار فعلی از نظر استفاده از الگوریتم‌های تکاملی و استفاده از رویکرد خلاقیت هنری انسان است.

۱-۳ کلیات روش پیشنهادی

۱-۳-۱ تولیدکننده الگوی رقم

چیزی که در این پروژه هدف اصلی و بسیار مهم است، تولید یک الگو به هر روش ممکن برای دادن به ربات برای رسم است. این قید اصلی مسئله است که تصویر ورودی به صورت مستقیم به ربات داده نشود، چرا که وجود آن در فضای حافظه‌ی ربات این امکان را برای ربات به وجود می‌آورد تا بدون نوآوری و صرفا پیمایش این عکس به نوعی مانند یک چاپگر، آن را بر روی کاغذ چاپ کند. که البته چاپگرها نیز مسئله‌ها و چالش‌های بسیاری دارند اما برای این پروژه چیزی که حائز اهمیت است، داشتن رویکردی مشابه به مغز انسان است که می‌تواند ارقام و اشکال و حتی تصاویر بسیار پیچیده تر را بدون داشتن تصویر کامل و مستقیم در ذهن با استفاده تشابه‌سازی به اشکال ساده‌تر و ساختن یک الگو برای کشیدن، رسم و طراحی کند. حال برای ساده‌سازی در این پروژه قصد داریم تا ارقام را با این رویکرد بر روی کاغذ با استفاده از بازو ترسیم کنیم.

---

[۳] line integral convolution (LIC)

[۴] Adamik

[۵] image vectorization

[۶] Real-valued genetic algorithms (RGA)

[۷] photorealistic

برای دستیابی به الگوی ترسیم اشکال، در این پروژه از چند رویکرد می‌توان استفاده کرد.

۱. در رویکرد اول میتوان به بررسی کلی الگوی اشکال ورودی پرداخت به نحوی که هر شکل دلخواهی را بتوان بدون قید، ترسیم کرد. برای مثال حرف جدید چینی را که قبلا مشاهده نکرده است، نیز بتواند ترسیم کند. برای این کار دو راه حل وجود دارد. ۱. در ابتدا شکل را به اجزای ساده‌تری که قبلا ترسیم آن‌ها را آموخته باشد یا مشاهده کرده باشد، تجزیه کند. ۲. یافتن ترتیب نقاط با دانستن اینکه شکل نهایی چگونه شود بدون آنکه مسیر ترسیم را از قبل بداند یا راهنمایی بگیرد.

۲. رویکرد دوم برای کاربردهای خوشنویسی و حروف چینی است که در آن ابتدا حرف یا نماد داده شده تشخیص داده می‌شود، و به یک عادت ترسیم از قبل آموخته شده، مراجعه می‌کند و آن را می‌کشد. در این رویکرد مسائل مهم شامل دقت، یادگیری عادت ترسیم، و ادراک و فهم نماد ورودی مطلوب کاربر است.

ما در این پروژه رویکرد اول و راه حل دوم مدنظرمان است اما برای سادگی، ابتدا از رویکرد دوم استفاده میکنیم. ماهیت ارقام را با یک تشخیصگر الگوی ارقام با مدل‌های شبکه‌های عصبی عمیق تشخیص می‌دهیم. و برای ترسیم آن، از یک الگوی از پیش تعیین شده که در حافظه ربات ذخیره شده است، استفاده می‌کنیم. در فاز دوم پروژه، این فرض برداشته شده و سعی در تولید الگو بدون نیاز به داشتن ماهیت آن رقم کرده‌ایم. بنابراین شکل حاصل از تولید الگو در فاز اول، برای هر شکل دسته‌بندی شده، یکتا است و در مراحل بعدی پویا و غیر یکتا خواهد بود. در این خصوص در پیاده‌سازی مفصلا بحث خواهد شد.

۱-۳-۲ تشخیصگر الگوی رقم

همانطور که در قسمت قبل نیز صحبت شد، برای تولید الگو، ابتدا به یک تشخیصگر رقم نیاز است. این تشخیصگر می‌تواند یک شبکه عصبی از پیش آموزش داده شده باشد. در ادامه در بخش پیاده‌سازی به توصیف شبکه عصبی تشخیص ارقام خواهیم پرداخت.

۱-۳-۳ رسم و پیمایش الگوی تولید شده

چنانچه بخواهیم مدلسازی و یادگیری عادت الگو را مستقیم از پیکسل‌های تصویر ورودی روی مفاصل ربات انجام دهیم، به علت اینکه درجات آزادی ربات دارای یک فضای پیوسته بزرگ است، نیاز به یک سخت‌افزار قوی و زمان طولانی برای آموزش شبکه عصبی خواهیم داشت. در اصطلاح به این مدل آموزش، یادگیری پایانه به پایانه[۸] می‌گویند. برای رفع این مشکل، قسمت‌هایی از آموزش پارامترها به صورت مستقیم در لایه‌های پنهان کم کرده و به صورت جداگانه در بخش‌های دیگر با روش‌های موجود حل می‌کنیم.

[۸] end to end learning

بنابراین در این مسئله نیز آموزش یک شبکه عصبی بزرگ برای پیدا کردن یک مدل یا یک سیستم، تا سیگنال دو بعدی ورودی که همان تصویر است را به بردار فضای سینماتیک معکوس یا همان بردار سرعت زاویه‌ای ربات در مفاصل نگاشت دهد [۹] و [۱۰]، بسیار سنگین و پیچیده و طولانی خواهد بود، چرا که برای مثال، در این روش هوش مصنوعی باید بتواند نقاطی را در نظر بگیرد که مفهوم شکل کلی عکس را داشته باشند، سپس مختصات این نقاط را به صورت تابعی از زمان بدست آورد، و در نهایت با سینماتیک معکوس سرعت‌های زاویه‌ای معکوس مناسب با زمانبندی مناسب برای تولید خروجی درست را پیدا کند. و باید همه‌ی این مراحل را در یک شبکه و یک جا یاد بگیرد. بنابراین در بخش معماری به تجزیه مراحل پروژه پرداخته و آن‌ها را شرح می‌دهیم.

حال باید مکانیزمی وجود داشته باشد که بازو بتواند نقاط الگوی تولید شده را دریافت کند و با یک زمانبندی مشخص آن‌ها را به ترتیب ترسیم کند. در این حین، برای رسیدن سر بازوی ربات به هر نقطه، میزان چرخیدن هر مفصل در هر لحظه باید مشخص گردد. همچنین سرعت رسیدن سر بازو به هر نقطه نیز مهم است و در ادامه نحوه‌های مختلفی که در این قسمت می‌توان بکار گرفت و مزایا و معایب هر کدام بررسی می‌گردد.

۱-۳-۴ ارزیابی

در این قسمت لازم است تا عملکرد ربات ارزیابی گردد. آیا راهی وجود دارد که بتوان مشخص کرد ربات درست کار کرده است؟ چند درصد از اوقات نتایج راضی کننده بوده است؟ تعریف رضایت‌مندی از نتایج چیست؟ تا چه میزان این نتایج قابل تکیه و مشابه هستند؟ به دنبال این پرسش‌ها، باید پاسخی ارائه شود و هر مفهوم دارای ابهام، به صورت دقیق تعریف شود.

۱-۳-۵ ملاحظات دنیای واقعی

۱-۳-۵-۱ عکس ورودی

در دنیای واقعی تصویر ورودی بر روی یک قاب در مکان ثابت باید قرار گیرد، محل این قاب را به طور مناسب در مقابل ربات، به طوری که دوربین ربات بر صفحه‌ی قاب عمود باشد انتخاب می‌کنیم. برای گذاشتن تصویر بر روی قاب، دو رویکرد ایستا و پویا می‌توان داشت. در رویکرد ایستا تصویر را قبل از اجرای شبیه‌سازی از مکان محلی ذخیره شده در کامپیوتر به منظور آزمون، بارگزاری کرده و بر روی بافت قاب می‌گذاریم. این امکان فقط به دلیل وجود چنین ویژگی بر روی شبیه‌ساز امکان پذیر است. در رویکرد پویا، این قابلیت در شبیه‌ساز وجود دارد تا قلمی برای کاربر تهیه شود و در حین اجرای شبیه‌سازی رقم دلخواه را برای روی قاب بکشد. این کار نیازمند مکانیزمی است تا دوربین در زمان خواسته‌ی کاربر بر روی کاغذ متمرکز شده و زمانی که کار کاربر به اتمام رسید در کنار قاب دکمه‌ای برای اطلاع اتمام کار از سمت کاربر قرار داشته باشد و تمرکز دوربین پس از فشرده شدن دکمه به ربات بازو منعطف شده و ربات تلاش خود را در تقلید رسم شکل کشیده شده توسط کاربر آغاز کند.

با توجه به اینکه در محیط شبیه‌ساز نیازی به دوربین نداریم این مسئله را فعلا کنار میگذاریم و به مسئله اصلی میپردازیم.

۱-۳-۵-۲ حسگر دوربین

به منظور دریافت و پردازش تصویر کشیده شده بر روی قاب، ربات بازو نیازمند یک حسگر دوربین است. به صورت پیش فرض (سوال) بر روی ربات abb چنین حسگری موجود نیست. بنابراین در پروژه باید یک دوربین به ربات اضافه شده و باید طوری بر روی ربات قرار گیرد که در شروع شبیه‌سازی نسبت به تصویر ورودی بر روی قاب عمود باشد.

۱-۴ ساختار پروژه

در فصل دوم به معماری نرم‌افزار و االگوریتم‌های مختلفی که در هر بخش می‌توان انجام داد و به علاوه‌ی کارهایی که قبلا در این زمینه انجام شده است، می‌پردازیم. و در فصل سوم پیاده‌سازی‌های انجام شده این پروژه را همراه با نتایج و ارزیابی ذکر می‌کنیم.

فصل دوم: مفاهیم پایه و کارهای مرتبط

۲-۱ مقدمه

در این فصل به معماری نرم‌افزار و االگوریتم‌های مختلفی که در هر بخش می‌توان انجام داد و به علاوه‌ی کارهایی که قبلا در این زمینه انجام شده است، می‌پردازیم.

۲-۲ مفاهیم پایه

۲-۲-۱ بررسی نرم‌افرازهای استفاده شده

شبیه‌ساز ویباتز

این شبیه‌ساز متن باز و رایگان، به هدف شبیه‌سازی ربات‌ها در قالب سه بعدی برای پروژه‌های صنعتی، آموزشی و تحقیقاتی در ابتدا توسط EPFL در سال ۱۹۹۶ و سپس توسط شرکت Cyberbotics در سال ۱۹۹۸ توسعه یافت [۱۱]. در این شبیه‌ساز، انباشتی از مدل‌های قابل تغییر ربات‌ها، حسگرها، عملگرها و اشیا تعبیه شده است. حتی قابلیت این وجود دارد تا مدل سه بعدی ربات را مستقیم از طریق اتوکد از پایه بسازیم و در محیط شبیه‌ساز بگذاریم. در تولید این مدل‌ها، می‌توان تمام خصوصیات فیزیکی و گرافیکی را به صورت بسیار جزئی ذکر کرد.

این شبیه‌ساز برای محاسبات فیزیکی خود از کتابخانه‌ی ODE یا Open Dynamics Engine استفاده می‌کند. این کتابخانه به صورت دقیق محاسباتی مانند برخورد اجسام، نیرو، اینرسی، اصطکاک و سرعت اجسام را انجام می‌دهد.

در شبیه‌سازی ویباتز گره و یا موجودیت قلم[۹] وجود دارد. با استفاده از آن می‌توان دقیقا منحنی ارقام و دقت مدل الگوی آن‌ها و نمایش خروجی و نتایج را به راحتی آزمود.

بنابراین با بررسی شبیه‌سازهای گوناگون تصمیم بر این مبنا شد که از شبیه‌ساز ویباتز استفاده شود.

زبان برنامه نویسی پایتون

زبان بسیار ساده‌ای است که بسیاری از کتابخانه‌های لازم را مانند opencv، matplotlib، tensorflow و ... را دارد همچنین این زبان عملیات‌های مناسبی برای بردارها به صورت بهینه فراهم کرده است.

[۹] pen node entity

۲-۲-۲ بررسی بازوهای رباتیکی

لیستی از ربات‌های کاندید بررسی شدند و قابلیت‌ها و سنسورهای آن‌ها مستند شدند که به ترتیب اولویت با توجه به امکانات به صورت زیر است.

ربات نائو:

- دارای یک شتاب‌سنج
- دارای دو دوربین در بالا و پایین ربات
- دارای دو حسگر فاصله‌سنج فراصوت سونار در چپ و راست ربات
- دارای یک جی پی اس
- دارای یک جیروسکوپ
- یک IMU
- ۴۰ عدد انکودر دارد
- ۴۰ عدد موتور چرخشی
- ۶ عدد حسگر لمسی بامپر

ربات داروین op:

دارای یک دوربین

ربات تیاگو:

امکان نصب دوربین وجود دارد

ربات pr2:

دارای تعداد بسیار زیادی دوربین

ربات abb:

بدون دوربین

دارای نمونه اجرایی به همراه قلم

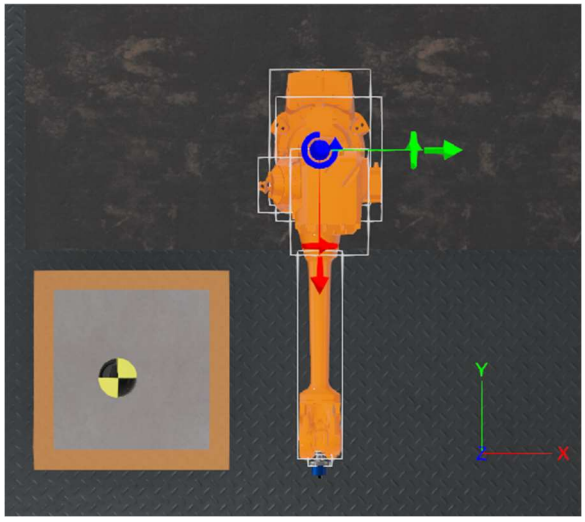
۳-۲-۲ توصیف محیط ربات در شبیه‌ساز

حال با انتخاب ربات abb به عنوان ربات بازویی که الگوی ارقام را بتواند روی کاغذ بکشد، نگاهی به هدف و شرایط مسئله می‌اندازیم.

در این پروژه همانطور که در عنوان نیز مطرح شده است یک بازوی رباتیکی abb به مختصات (۰, ۴,۸۵-) در وضعیت آماده است و یک کاغذ مقابل آن قرار دارد که مختصات آن (۱,۱-, ۵,۸-) است و فاصله نسبی آن نسبت به دستگاه مختصات ربات (۱,۱-, ۰,۹۵-) است. توجه کنید که سر ربات نسبت به مختصات دنیا ۹۰- درجه انحراف دارد. بنابراین بردار فاصله‌ی مرکز کاغذ تا مرکز ربات برای اینکه به دستگاه ربات نگاشت شود، باید مختصات آن دوران کند. برای نگاشت دورانی رابطه زیر به کار گرفته می‌شود.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

بنابراین مختصات نقطه‌ی مرکز کاغذ نسبت به دستگاه ربات (۱,۱, ۰,۹۵-) خواهد بود. که در شکل زیر نیز به صورت شهودی نیز قابل تایید است.



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos(-\frac{\pi}{2}) & \sin(-\frac{\pi}{2}) \\ -\sin(-\frac{\pi}{2}) & \cos(-\frac{\pi}{2}) \end{pmatrix} \begin{pmatrix} -0.95 \\ -1.1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} -0.95 \\ -1.1 \end{pmatrix}$$

$$= \begin{pmatrix} 1.1 \\ -0.95 \end{pmatrix}$$

شکل ۰۱- مختصات ربات در محیط شبیه‌ساز، به همراه نمایش دستگاه مختصات ربات و دنیای شبیه‌سازی

سپس با گرفتن تصویر ورودی از کاربر، عامل به تشخیص آن پرداخته و برای آن الگویی برای ترسیم تولید می‌کند. پس از تولید الگوی ترسیم به صورت یک ماتریس یا عکس، سعی می‌شود از روشی استفاده شود تا الگو را به یک دنباله‌ای از مختصات‌های ترتیب‌دار برای پیمایش و رسم شکل استفاده شود. در ابتدا فرض در این مسئله این است که بازو امکان برداشتن قلم از روی کاغذ را ندارد. در نهایت این قید را رها خواهیم کرد. اطلاعات مرتبط با ربات بازو در مرجع [۱۲] و [۱۳] آمده است.

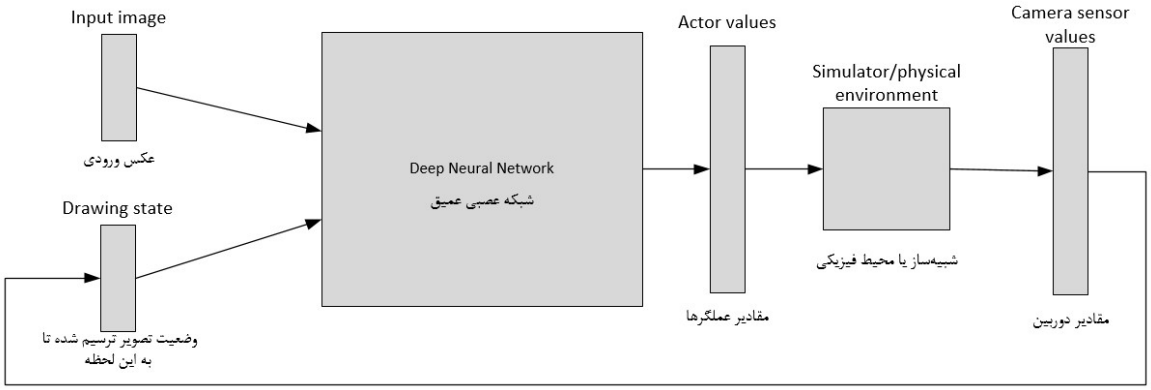
شکل ۰-۲– ربات irb4600 از شرکت abb برگرفته از مرجع [۱۲].

۲-۳ تحلیل کارهای مرتبط و معماری پروژه

این پروژه یک سیستم بزرگ نرم‌افزاری و شبیه‌سازی پیچیده است و ارتباط اجزای مستقل آن باید به درستی مشخص شود. در این بخش به تعریف هر یک از اجزای مستقل و بلوک‌های مورد نیاز برای درست عمل کردن کل جریان سیستم می‌پردازیم. معماری پیشنهادی در شکل زیر آورده شده است.

۲-۳-۱ معماری اول

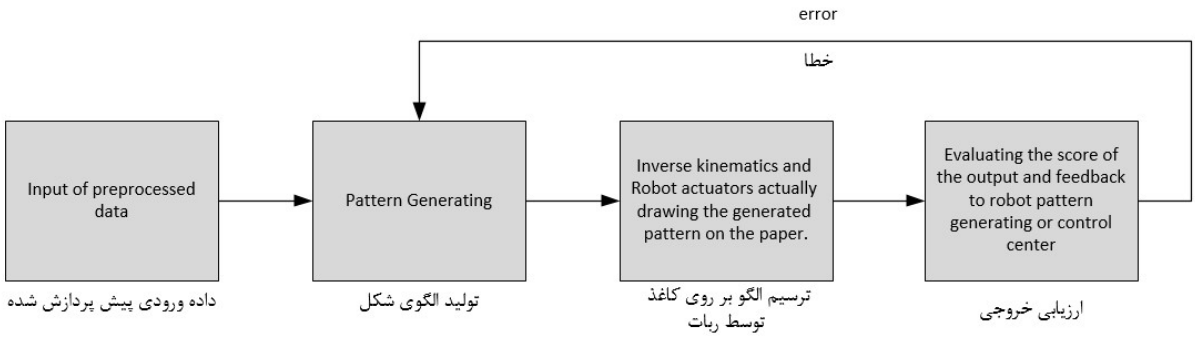
در این معماری به مسئله به دید یک فرآیند مارکوفی نگاه شده و برای مسئله یک راه حل یادگیری تقویتی عمیق پیشنهاد شده است. این معماری دارای این عیب است که رویکردی مانند یادگیری پایانه به پایانه دارد و به سیستم پردازشی قوی جهت یادگیری نیازمند است و برای تسریع یادگیری فضای پیچیده جستجو باید یک کارت گرافیکی یا زمان طولانی برای یادگیری در اختیار داشت که با امکانات حاضر قابل دستیابی نیست.



شکل ۰-۳ – معماری مدل مارکوف

۲-۳-۲ معماری دوم

برای کاهش پیچیدگی، سیستم به اجزای کاملا مستقل و بسیار ساده‌تری تقسیم شد. هر کدام از اجزا می‌تواند بدون توجه به پیاده‌سازی اجزای دیگر، پیاده‌سازی‌های مختلف خود را داشته باشد.



شکل ۰-۴ – معماری تجزیه شده

در ادامه به بررسی هر ماژول یا جزء می‌پردازیم.

۲-۳-۳ بخش ورودی

این بخش به روش‌های زیر قابل پیاده سازی است:

۱. ورودی به صورت دادن یک داده عدد صحیح بین ۰ تا ۹
۲. ورودی به صورت دادن یک تصویر گرفته شده از حسگر دوربین
   a. عکس بر روی قاب به صورت پویا در حین شبیه‌سازی
   b. عکس بر روی قاب به صورت ایستا قبل از شروع شبیه‌سازی
۳. ورودی به صورت دادن یک تصویر گرفته شده از فایل سیستم داخل ربات

در مورد دوم، مسائلی پیش می‌آید که باید در نظر گرفته شوند. دوربین می‌تواند کامل ثابت باشد و در نتیجه تصویر ورودی باید در جای ثابتی قرار گیرد. یا دوربین می‌تواند متحرک باشد، در این صورت باید یک کنترل جدایی برای دوربین تنظیم گردد. در این حالت اولا، دوربین باید محل تصویر ورودی را مکانیابی کند، و ثانیا اگر ربات تک دوربینه باشد باید محل کاغذ روبروی خود را نیز مکانیابی کند. اگر ربات دوربین داشته باشد، یک دوربین برای تصویر ورودی و یک دوربین برای کنترل تصویر خروجی استفاده می‌شود.

۲-۳-۴ بخش تولید الگو

مفهوم تولید الگو، به معنای ایجاد یک سیگنال $I(x(t), y(t))$ است که در واقع دنباله‌ای از نقاطی است که ترسیم یک شکل را نمایش می‌دهد. این سیگنال را می‌توان به چند صورت تولید کرد:

۱. تشخیص تصویر و پیدا کردن دسته‌ی آن از طریق خوشه‌بندی یک سری داده و سپس پیدا کردن شکل نماینده‌ی خوشه و پیدا کردن دنباله‌ی نقاط آن تصویر نماینده.

۲. تشخیص و دسته بندی تصویر ورودی و سپس استفاده از یک جدول جستجو که از پیش تصویر یا سیگنال $I(x(t), y(t))$ ساخته یا ذخیره کرده است.

۳. درخواست ورودی به صورت نماد یا مفهوم یا متنی از شکل درخواستی باشد و سیستم تصویری را به صورت خیالی تولید کند و دنباله آن را بیابد.

۴. می‌توان تصویر را به صورت خام به الگوریتم بدون هیچ تغییر یا تولید عکس جدیدی تحویل داد تا دنباله‌ی مدنظر را از همان عکس خام ورودی پیدا کند.

همچنین در موارد ۲ و ۳ می‌توان به جای تولید یک تصویر که ماتریس نشان دهنده الگوی درخواستی است، مستقیما سیگنال‌های $I(x(t), y(t))$ و $\vec{\phi}(t)$ را بدست آورد. که در حالت آخر کمی مشابه معماری اول و یادگیری پایانه به پایانه است.

و اما الگوریتم می‌توانند موارد بالا را به یک دنباله بر حسب زمان تبدیل کنند، عبارتند از:

۱. الگوریتم صریحانه پارامتری سازی زمانی؛ که به صورت مفصل در نسخه اولیه توضیح داده شده است.
۲. استفاده از الگوریتم بهینه‌سازی تکاملی؛ در این روش به دنبال کمینه کردن میزان خطای یک دنباله و ترتیب زمانی پیکسل‌های حدس زده شده هستیم [۱۴].

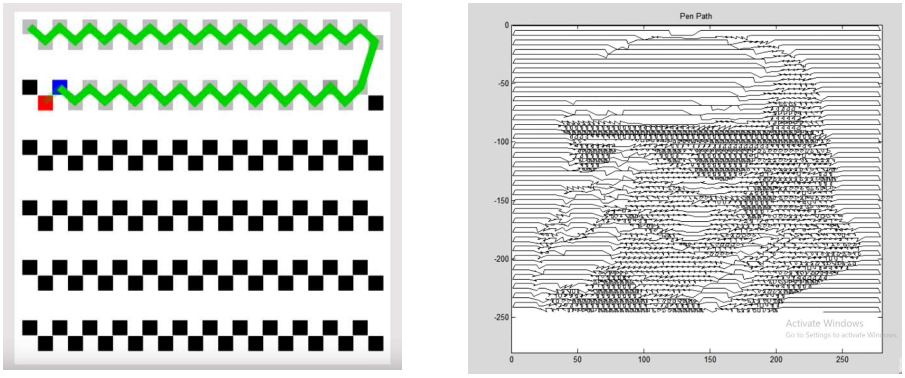
۳. استفاده از شبکه‌های عصبی بازگشتی[۱۰] و گذاشتن یک ماژول حافظه بلند کوتاه مدت[۱۱]؛ با این روش می‌توان با آموزش داده‌های ترتیبی به شبکه، به یک الگو با یک ترتیب زمانی نسبت به مفهوم تصویر ورودی، دست یافت [۱۷-۱۵].

۴. استفاده از یادگیری تقویتی عمیق؛ می‌توان با آزمون و خطا و آموزش مستقیم بازوی ربات به یادگیری چگونگی تولید خروجی و پیدا کردن بهترین اعمال بر حسب وضعیت حاضر دنیای شبیه‌سازی یا واقعی پی برد. یادگیری تقویتی یکی از محبوب‌ترین روش‌ها برای یک فضای با توصیف مارکوفی است.

[۱۰] Recurrent Neural Networks

[۱۱] Long Short-term Memory (LSTM)

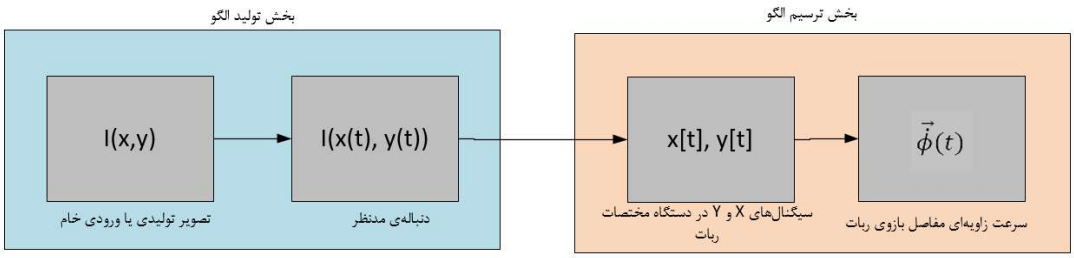
۵. روش‌های افراد دیگر، به این صورت بوده است که به روش فلوید [۱۸] تصویر ورودی را نیم‌تن[۱۲] می‌کنند و سپس با استفاده از پیدا کردن و وصل کردن نزدیکترین همسایه از نظر فاصله اقلیدسی، که در واقع همان الگوریتم DFS است، کل تصویر را ترسیم می‌کنند. متاسفانه همانطور که در مقدمه نیز گفته شد، این روش بسیار مانند یک پرینتر عمل می‌کند و خطوط اصلی شکل را مانند چهره یکجا رسم نمی‌کند (شکل ۶).



شکل ۰-۵ – سمت چپ الگوریتم رسم به نزدیکترین نقطه و سمت راست خروجی. مراجع [۱۹] و [۲۰].

۵-۳-۲ بخش ترسیم الگو

در این بخش الگوی تولید شده از بخش قبلی را به ربات ورودی داده و بعد از یکسری تغییر مختصات، ربات آن را به همراه یک سری تاخیر و انجام سینماتیک معکوس رسم می‌کند.



شکل ۰-۶ – اجزای بخش‌های تولید الگو و ترسیم الگو

[۱۲] halftoning

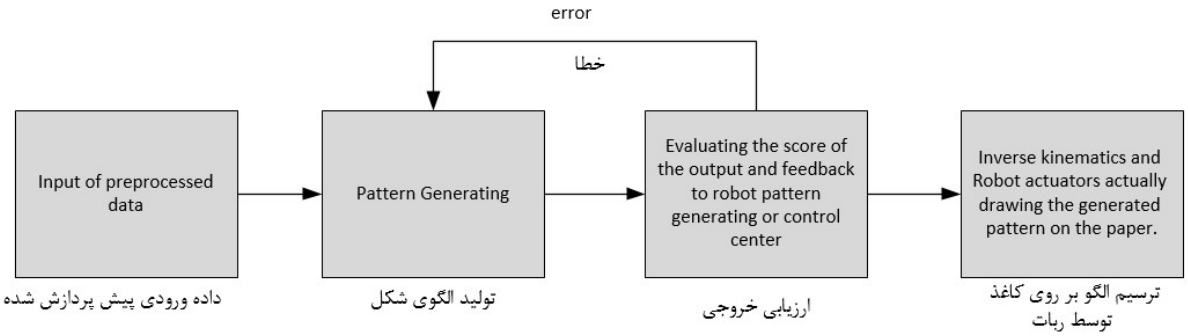
۲-۳-۶ بخش ارزیابی خروجی

این بخش نیز مانند دیگر ماژول‌ها دارای چندین روش برای پیاده‌سازی است.

۱. ابتدا ربات بر روی کاغذ الگو را رسم کرده و با گذاشتن حسگر بینایی بر روی کاغذی که شکل روی آن ترسیم شده و محاسبه خطا، ارزیابی لازم انجام می‌شود.

۲. محاسبه و تخمین خروجی به صورت غیرفیزیکی با فرض اینکه در ترسیم بازو و سینماتیک معکوس تقریبا خطایی موجود نباشد. این کار از طریق رسم خطوط بین پیکسل‌هایی که باید به ترتیب رسم شوند صورت می‌گیرد.
به عبارت دیگر رسم خطوط بر اساس این است که معادله خط بدست آمده بین دو پیکسل، از هر پیکسلی که عبور کند، آن پیکسل را سیاه کنیم. همچنین برای رسم خط از روش‌های آماده‌ی bresenham, DDA, mid point line drawing نیز استفاده کرد.

در بین این دو روش، روش دوم سرعت بیشتری دارد و می‌تواند مستقل از بخش‌های جلوتر معماری مانند ترسیم الگو تولید شده انجام شود. بنابراین نتیجه ارزیابی شده فقط از بخش تولید الگو خواهد بود که در نتیجه سرعت بیشتری نیز در آموزش تولید الگو خواهیم داشت. بنابراین معماری از شکل ۵ به شکل زیر تغییر می‌کند.



شکل ۰-۷ – معماری پیشنهادی سوم

برای محاسبه خطا روش‌های زیر را بکار برده‌ایم:

۱. محاسبه واریانس مقادیر شدت نور پیکسل‌های متناظر دو تصویر مطلوب و واقع. یا همان میانگین مربعات خطا[۱۳].

۲. میزان تعلق فازی به دسته مطلوب را به عنوان امتیاز و معکوس آن را (1-σ) به عنوان خطا. یعنی حسگر دوربین تصویر خروجی را دریافت کرده و با استفاده از همان تشخیص دهنده‌ی دسته‌ی تصویر برای تشخیص خروجی استفاده کرده و نتیجه خروجی را از ورودی کم می‌کنیم. (لایه‌ی آخر شبکه عصبی که از تابع سیگموید استفاده می‌کند و به صورت 1-hot encoding است).

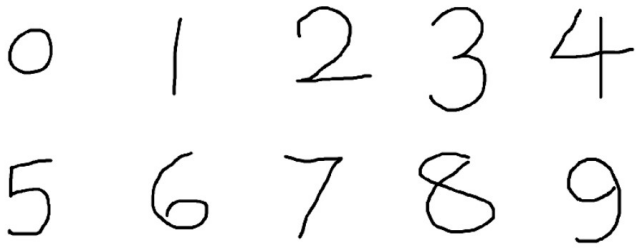
[۱۳] Mean Squared Error (MSE)

فصل سوم: روش پیشنهادی و نتیجه‌گیری

۳-۱ نسخه اولیه (پروتوتایپ)

۳-۱-۱ ورودی

در راستای پیاده‌سازی نسخه اولیه سعی شد تا در ابتدا الگوهای ارقام در جایی ذخیره شده و به صورت دستی به وجود آیند. اشکال دستی که ربات باید آن‌ها را عینا در کاغذ بکشد در شکل زیر آورده شده‌اند. این کار نیز باعث می‌شود تا نتایج سینماتیک معکوس و بخش ترسیم الگو زودتر در پروژه بررسی و ارزیابی شوند.



شکل ۳-۱ – اشکال دستی ارقام

همانطور که در شکل نیز مشهود است، رقم ۴ وابسته به دست‌خط، گاهی اوقات غیر متصل نوشته می‌شود و برای ترسیم نیاز به برداشته شدن قلم از کاغذ دارد. بنابراین تلاش در این است که نوشتن رقم ۴ در دو مرحله انجام شود یا از الگوریتمی استفاده شود که بدون برداشتن قلم با چشم‌پوشی از خطا، رقم را بنویسد.

۳-۱-۲ تشخیص عکس ورودی

در بخش معماری اگر به دوربین احتیاج پیدا شد، تشخیص تصویر ورودی را از مدل شبکه عصبی با مشخصات زیر، انجام خواهیم داد.

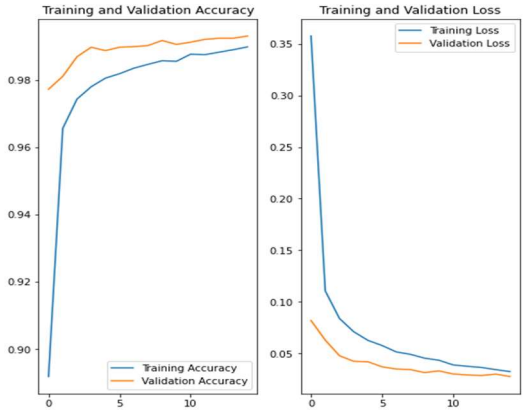
جدول شماره ۱ - مشخصات شبکه عصبی برای تشخیص تصویر ورودی

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d_2 (MaxPooling 2D)	(None, 13, 13, 32)	0
conv2d_4 (Conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_3 (MaxPooling 2D)	(None, 5, 5, 64)	0
flatten_1 (Flatten)	(None, 1600)	0
dropout (Dropout)	(None, 1600)	0
dense_5 (Dense)	(None, 10)	16010

Total params: 34,826
Trainable params: 34,826
Non-trainable params: 0

این مدل روی ۶۰۰۰۰ تصویر ۲۸*۲۸ سیاه و سفید ارقام، آموزش داده شد. این دیتاست، دیتاست معروف MNIST است [۲۱]. تعداد داده‌های تست ۱۰۰۰۰ عکس ۲۸*۲۸ سیاه و سفید است. تعداد داده‌های ارزیابی برای ارزیابی متقاطع[۱۴] ۰,۱ تعداد داده‌های آموزش است.

برای یادگیری از ۱۵ ایپاک و ۱۲۸ بچ استفاده شد. تابع ضرر[۱۵] تابع کراس آنتروپی دسته‌بندی، بهینه‌ساز آن آدام، و تابع ارزیابی آن دقت[۱۶] است.



شکل ۰-۲ – نمودار نتایج شبکه عصبی، دقت و تابع ضرر بر حسب هر ایپاک. این نمودار حاصل از اجرای کد استفاده شده از مرجع [۲۲] است.

در نهایت دقتش روی داده‌های آموزش، ۹۸,۹۹ درصد، روی داده‌های ارزیابی ۹۹,۳۲ درصد، روی داده‌های تست ۹۹,۱۱ درصد دقت داشته است و به ترتیب تابع ضرر برای هر کدام، ۰,۰۳۲، ۰,۰۲۸ و ۰,۰۲۴ است.

۳-۱-۳ تولید الگو به روش الگوریتم‌های صریحانه و ابتکاری

در حین پیاده‌سازی نسخه اولیه یا همان پروتوتایپ به این نیز فکر شد که آیا امکان پیاده‌سازی الگوریتمی صریحانه[۱۷] یا ابتکاری[۱۸] برای پیدا کردن ترتیب درست زمانی ترسیم پیکسل‌های روی یک عکس وجود دارد یا خیر. نام این الگوریتم صریحانه را پارامترسازی زمانی می‌نامیم. به این دلیل که برای نوشتن ارقام علاوه بر دانستن مکان پیکسل‌های سیاه که بر اساس رابطه ضمنی بین مولفه‌های

[۱۴] cross-validation

[۱۵] loss function

[۱۶] accuracy

[۱۷] explicit programming

[۱۸] heuristic

x و y هستند، رابطه پارامتری آن‌ها براساس زمان نیز لازم است تا پیمایش درستی صورت گیرد و مسیر به درستی و به مرور زمان طی شود. مثلا دایره یا رقم صفر اگر فقط بر اساس ترتیب بزرگ به کوچک y ترسیم شود، با فرض اینکه برداشتن قلم از روی کاغذ مجاز باشد، آنگاه برای x از دو مسیر باید حرکت کند که مانند دستگاه چاپگر سطر به سطر خواهد بود. همچنین اگر برداشتن قلم از روی کاغذ مجاز نباشد، حرکتی زیگزاگی با xهای مختلف خواهیم داشت. اما چیزی که معمولا در نوشتن و ترسیم کردن اتفاق می‌افتد رسم بر اساس یک مسیر پیوسته و بدون برداشتن قلم است که در این مورد خاص بدون دانستن رابطه پارامتری امکان‌پذیر نیست. برای همین نام آن را پارامتری‌سازی زمانی می‌نامیم.

الگوریتم پارامتری‌سازی زمانی با این دورنگاه در نظر گرفته شد که شاید در آینده نیازی به یادگیری سیگنال‌های x و y بر حسب زمان برای تولید الگوی رقم نباشد. چرا که با این ساده‌سازی در واقع کاهش فضای جستجو و پارامترهای یادگیری را خواهم داشت و سرعت و دقت در خروجی بسیار افزایش خواهد یافت. اینگونه صرفا خروجی تولید الگوی یادگیری شده $I(x,y)$ را به همین الگوریتم صریحانه داده شده و در انتها برای حرکت سر بازو به سینماتیک معکوس سینگال‌ها را به عنوان ورودی می‌دهیم.

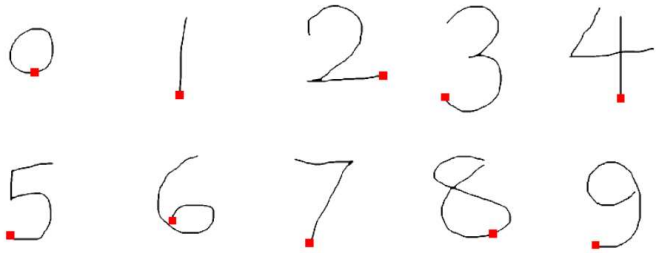
برتری الگوریتم صریحانه پارامتری‌سازی زمانی، نسبت به الگوریتم یادگیری ماشین معادل آن، این است که پیچیدگی و ابعاد جستجو را کاهش می‌دهد. و علاوه بر این نیازی به داده‌های حجیم برای آموزش نخواهد داشت. عیب این الگوریتم این است که پیاده‌سازی آن و حالت‌های استثنای آن برای اشکال مختلف به صورت کلی زیاد است و معمولا جامعیت ندارند. همچنین توصیف صریحانه انجام یک کار پیچیده، سخت‌تر است و زمان بیشتری برای پیاده‌سازی صرف می‌کند. همچنین اگر قرار باشد که از جدول جستجو یا همان Lookup-table (LUT) استفاده نشود، این الگوریتم کند خواهد بود.

این ماژول، ساده‌سازی شده‌ی ماژول پیچیده‌ی تولید الگو است. حال می‌توان عکس‌های ورودی خام را نیز به این الگوریتم داد و لزوما نیازی به عکس‌های تولید شده از الگوریتم‌های تولید الگو یا جداول LUT نیست. برای تشخیص صحت و آزمون درستی آن می‌توان به صورت مستقیم از عکس‌های ورودی و جامع استفاده کرد. به عبارت دیگر تابع $I(x, y)$ که الگوها تولید کرده‌اند یا عکس خام ورودی به ما داده است را باید به صورت $f(t) = I(x(t), y(t))$، پارامتری از زمان درآورد که در آن متغیر t یک متغیر گسسته و نمونه برداری شده از متغیر پیوسته زمان است.

در واقع عملکرد اصلی این الگوریتم استفاده از الگوریتم جستجوی اول سطحی[۱۹] است. که این الگوریتم با استفاده از یک صف همسایه‌های خود را که سیاه هستند بررسی کرده و آن‌ها را ذخیره می‌کند و سپس این کار را تکرار می‌کند تا همگی پیکسل‌های سیاه را ملاقات کند.

[۱۹] Breadth first search (BFS)

یکی از چالش‌های اصلی الگوریتم صریحانه در این است نقطه شروع رسم را در شکل پیدا کنیم. نقطه شروع در شکل همیشه از یکی از نقاط انتهایی شروع می‌شود. نمونه ای از نقاط انتهایی یافت شده در هر رقم را در شکل زیر مشاهده می‌کنید. باید توجه داشت که در حلقه‌ها نقاط انتهایی یکتا بدست نمی‌آیند.



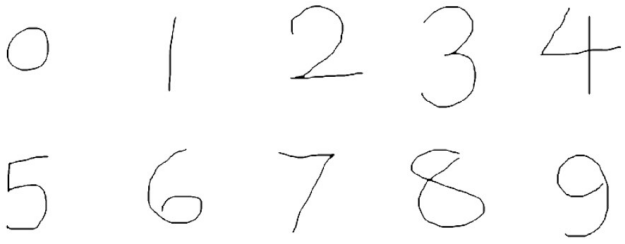
شکل ۰-۳ – آخرین نقاط انتهایی پیدا شده در هر رقم

داشتن نقاط انتهایی به ربات کمک میکند که بدون آنکه نیاز باشد از الگوریتم های یادگیری ماشین و بینایی ماشین استفاده کنیم به سرعت این نقاط را پیدا کرده و از آن مختصات‌ها استفاده کنیم. از همین انتهایی پیدا شده، به عنوان نقطه ابتدایی برای الگوریتم استفاده میکنیم.

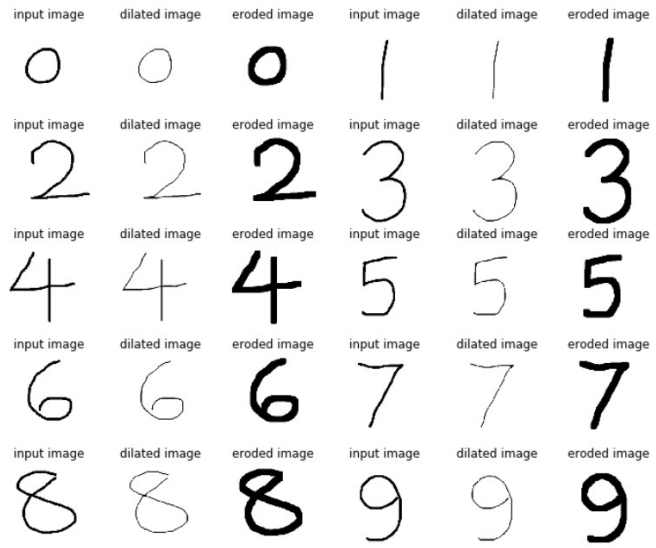
۳-۱-۴ اشکالات الگوریتم پارامتری‌سازی زمانی

مشکل اول، در این بود که در دقت مقیاس پیکسل خطوط در ارقام دارای ضخامت قابل توجهی هستند، و ترسیم به صورت پیکسل به پیکسل، مسیر بسیار زیگزاگی و دارای نویزی را برای ربات به وجود می‌آورد. در تلاش برای حل این مشکل، راه‌حل‌های زیر بررسی شد.

با استفاده از روش morphological dilation در کتابخانه openCV سعی شد تا حد امکان نازکترین مسیر از ارقام بدست آید.[۲۳]

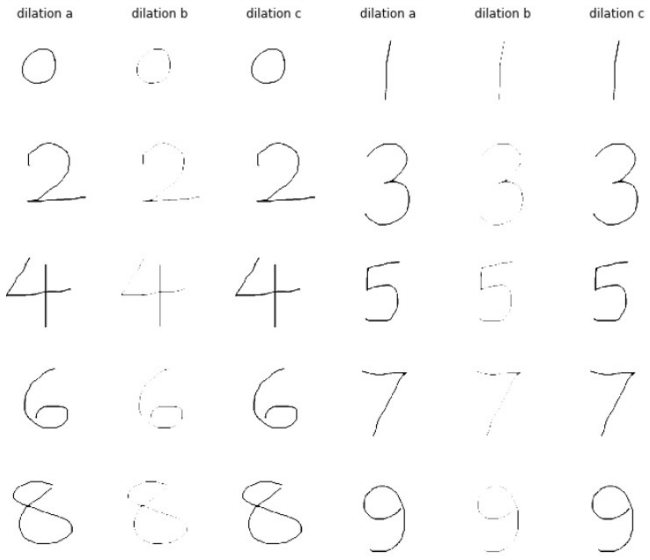


شکل ۰-۴ – نتایج حاصل از dilation برای ارقام صفر تا ۹.



شکل ۰-۵ – ستون سمت چپ تصویر اصلی ارقام زوج و ستون‌های دوم و سوم به ترتیب نازک شده و کلفت شده از ارقام است. به همین ترتیب برای ارقام فرد.

سایز کرنل برای dilation با آزمون و خطا و با توجه به نتایج شکل زیربه صورت مناسب انتخاب شد.

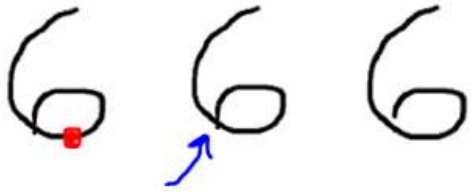


شکل ۰-۶ - همانطور که در شکل نیز مشهود است. سه دسته نتیجه از فیلترهای a، b و c حاصل شده است. فیلتر a با سایز ۵*۵ انتخاب شده است. فیلتر b با اندازه‌ی ۷*۷ انتخاب شد است. در نهایت، فیلتر c با اندازه‌ی ۳*۳ انتخاب شده است اما با این تفاوت نسبت به دو فیلتر قبل که دو بار بر روی تصویر اعمال گردیده است. همانطور که انتظار می‌رفت دو لایه از اعمال فیلتر c نتیجه‌ای بسیار مشابه با اعمال یک لایه از فیلتر a داشته است. همچنین در اعمال فیلتر b، نتایج بیش از حد کم‌رنگ و پیکسل‌ها از هم گسیخته شده‌اند. با این اوصاف فیلتر با اندازه‌ی ۵*۵ را انتخاب کردیم.

با این که در راستای نازک کردن خطوط اشکال تلاش شد، اما همچنان ضخامت شکل‌های بیشتر از یک پیکسل بود. حال برای نازک کردن روش دیگری انتخاب شد. که آن کم کردن رزولوشن تصویر بود. با این حال برخی از ارقام در برخی گوشه‌ها دارای ضخامت بیش از یک پیکسل هستند. نتیجه در شکل زیر قابل مشاهده است.

شکل ۰-۷ – نتایج حاصل کم کردن رزولوشن از ۳۰۰*۳۰۰ به ۱۰*۱۰.

مشکل دیگری که در این الگوریتم وجود دارد، این است که برای پیدا کردن نقاط انتهایی، کاری که این الگوریتم انجام می‌دهید این است که اگر تمامی نقاط همسایه یک پیکسل سیاه، سفیده باشند یا پیکسل‌های سیاهی باشند که قبلا توسط الگوریتم دیده شده‌اند، این نقطه سیاه را نقطه انتهایی در نظر می‌گیرد. و مثلا برای رقم ۶ اگر در پایین خود یک حلقه بسته میداشت، آنگاه نقطه انتهایی در پایین رقم ۶ اتفاق می‌افتاد (مشابه رقم ۸ در شکل ۱۱). که این نقطه برای شروع رسم رقم ۶ محل نامناسبی است. برای فهم بهتر شکل زیر را ملاحظه کنید.



شکل ۰-۸ – رقم ۶ سمت راست، حلقه باز دارد چنانچه مشابه رقم ۶ وسط حلقه آن را ببندیم، نقطه انتهایی در پایین ظاهر می‌شود.

بنابراین به نظر می‌رسد، برای تمام ارقام و اشکال نمی‌توان روش جامعی از طریق این الگوریتم ارائه داد.

مشکل سوم این الگوریتم، در ارقامی است که مشکل تقاطع دارند، مانند رقم ۸، ۴، ۶، ۹ و ... که هنگام رسیدن الگوریتم به تقاطع در حین بررسی مسیر و همسایگی، الگوریتم همه را ذخیره کرده و در آینده به ترتیب همه‌ی انشعاب‌ها را همزمان ذخیره و پیمایش می‌کند که برای ترسیم ربات نامناسب است.

۳-۱-۵ جمعبندی الگوریتم پارامتری‌سازی زمانی

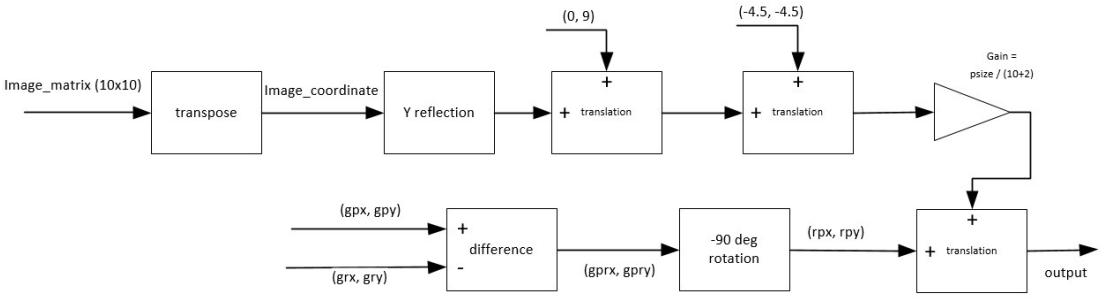
همه این مشکلات که به طور کامل شرح داده شد، نشان می‌دهد که این الگوریتم دارای استثناهای زیادی است و قوانین دستی برای تولید مسیری که بتواند تمامی شکل‌های مختلف را دربر بگیرد، اگر ناممکن نباشد، بسیار بسیار مشکل است. بنابراین از روش‌های صریحانه به سراغ روش‌های غیرصریحانه و یادگیری ماشین رفتیم که در ادامه به آن‌ها می‌پردازیم.

۳ – ۱- ۶ ترسیم الگوی تولید شده

ما در پروتوتایپ همانطور که در بخش قبل توضیح دادیم، الگوریتم صریحانه به نتایج رضایت‌بخشی نرسید. در نتیجه ما الگوی ترسیم را به صورت دستی تعیین کردیم. برای این منظور در تصویر با رزولوشن پایین (۱۰*۱۰) دنباله‌ای از مکان پیکسل‌های سیاه به عنوان مسیر ترسیم معرفی کردیم. سپس این دنباله‌ی نقاط به ربات داده شد تا با تبدیلات مختصات مناسب، به مختصات کاغذ تبدیل شده و رسم شود. جزئیات این تبدیلات مختصات در بخش توضیح داده می‌شود.

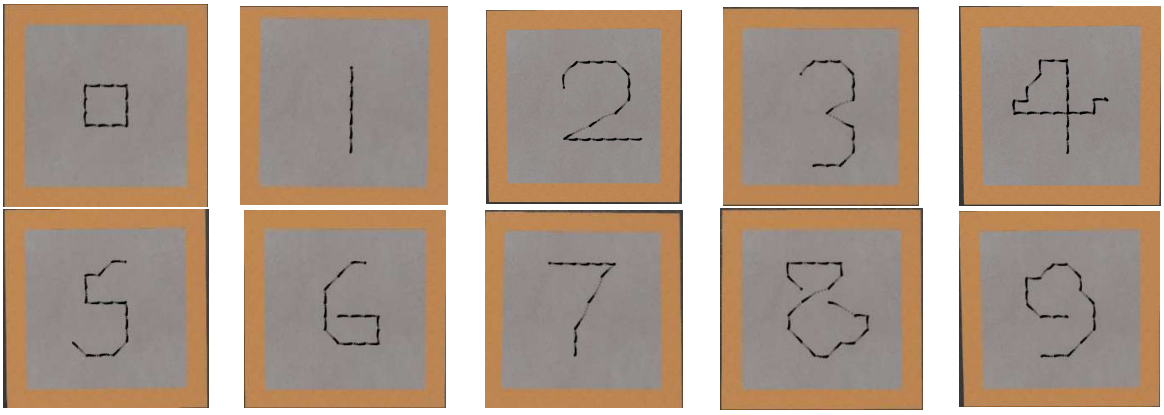
۳ – ۱ -۷ تبدیلات هندسی سیگنال ورودی

برای اینکه هر نقطه را در دستگاه ربات بتوانیم رسم کنیم، نیاز به تبدیلات مختصاتی و سیگنالی داریم تا الگوی مسیر معرفی شده، بتواند روی کاغذ قرار گیرد.



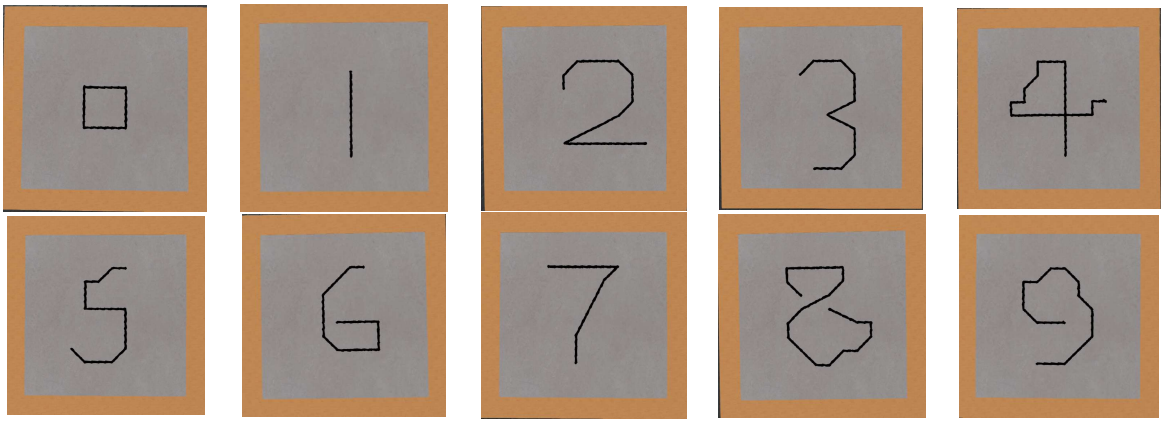
شکل ۰-۹ – نمودار بلوکی تبدیلات مختصات از شکل ورودی تا خروجی روی کاغذ

۳-۱-۸ نتایج حاصل از پروتوتایپ



شکل ۰-۱۰ – نتایج حاصل کشیدن ارقام پروتوتایپ روی کاغذ توسط ربات

همانطور که در شکل دیده می‌شود به علت تغییر سرعت ربات در برخی از خطوط که دو نقطه‌ی متوالی از یکدیگر دور هستند، خط ترسیم شده به صورت غیر یکنواخت دیده می‌شود (مانند شکل رقم ۷ و ۸ در شکل بالا). نکته دیگر این است که برای کنترل حرکت بازوی ربات، در نزدیکی نقطه مقصد سرعت ربات کمتر می‌شود و در نتیجه خط ترسیم شده پررنگ‌تر می‌شود. برای رفع هر این مشکلات، بین هر دو نقطه متوالی از مسیر به نسبت فاصله‌ی آن‌ها تعدادی نقطه میانی اضافه می‌کنیم به نحوی که فاصله هر دو نقطه در مسیر ثابت بوده و حداکثر ۳ سانتی متر باشد. با این اصلاح نتایج زیر بدست آمد.



شکل ۰-۱۱ – نتایج حاصل کشیدن ارقام پروتوتایپ روی کاغذ بعد از اصلاح فواصل نقاط متوالی

۳-۲ پیاده‌سازی روش اصلی: الگوریتم ژنتیک

برای تشخیص الگوی ترسیم، در این پروژه از الگوریتم ژنتیک استفاده کردیم. از مزایای الگوریتم ژنتیک نسبت به دیگر الگوریتم‌های بررسی شده تا به اینجا، عدم نیاز به داشتن داده‌های آموزش است. جمع‌آوری داده یکی از اصلی‌ترین چالش‌های الگوریتم‌های رایج یادگیری ماشین است. در مقایسه با یادگیری تقویتی که آن نیز نیازی به جمع‌آوری داده‌های آموزشی برچسب زده شده ندارد، این برتری را دارد که نیاز به آموزش سرعت زاویه‌ای مفاصل ربات برای کشیدن شکل نیست. بدین ترتیب از حالت یادگیری پایانه به پایانه در آمده است و در یک محیط بسیار ساده‌تر زمان بسیار کمتری صرف آموزش داده‌ها می‌شود. در این الگوریتم از معماری دوم استفاده می‌شود که می‌تواند کاملا از بخش‌های دیگر مستقل کار کند. این الگوریتم برای کشیدن همه‌ی انواع اشکال با کمی چشم‌پوشی از خطا، می‌تواند کار کند. علاوه بر این‌ها برتری نسبت به روش صریحانه این است که قانون‌گذاری فقط از طریق امتیازدهی در مقایسه با تصویر اولیه صورت می‌گیرد.

جدول ۲ – مشخصات الگوریتم ژنتیک استفاده شده برای ترسیم الگو

نوع نمایش ژنتیک[۲۰]	جایگشتی[۲۱]
نوع بازترکیبی[۲۲]	بازترکیبی یال‌ها[۲۳]
احتمال بازترکیب	٪۱۰۰
نوع جهش	تعویض[۲۴]
احتمال جهش	٪۸۰
انتخاب والدین	٪۱۰ بهترین از جمعیت
انتخاب جفت والدین	تصادفی بدون تکرار والدها
انتخاب بازماندگان	حذف بدترین‌ها
اندازه جمعیت	۵۰۰
تعداد فرزندان از هر جفت	۲
انتخاب نسل نخستین	تصادفی
شرط پایان	تعداد نسل‌ها بزرگتر از ۵۰۰۰ یا بهترین جواب خطایش صفر باشد.

برای پیاده‌سازی الگوریتم ژنتیک، کتابخانه‌ها و برخی روش‌ها را بررسی کردیم. اغلب کتابخانه‌ها معطوف به نمایش باینری بودند، همچنین طبق نظرات افراد باتجربه در مرجع [۲۴] پیاده‌سازی از پایه بهتر، راحتتر و مناسبتر است. بنابراین ما این الگوریتم با ساختن

[۲۰] representation

[۲۱] permutation

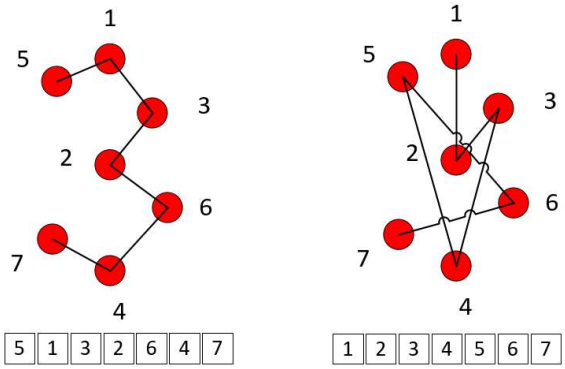
[۲۲] recombination

[۲۳] edge recombination

[۲۴] swap

یک کتابخانه انعطاف‌پذیر، پیاده‌سازی کردیم. تمام اجزایی که پیاده‌سازی شد، به ترتیب با جزئیات کامل تست شدند و نتایج آن در گیتهاب قرار دارد (پیوست الف).

در قدم اول مشابه آنچه در بخش پروتوتایپ توضیح داده شد، تصاویر ورودی را پیش پردازش کردیم و تصاویری با رزولوشن پایین (۱۰*۱۰) استخراج کردیم. سپس مختصات ماتریسی پیکسل‌های سیاه تصاویر را در یک لیست ذخیره کردیم، سپس اندیس‌های این لیست را به عنوان کروموزوم‌های جمعیت در نظر گرفتیم. این به این تعبیر است که نمایش الگوریتم ژنتیک ما به صورت جایگشتی است از این اندیس‌ها که اگر محل هر اندیس تغییر پیدا کند، ترتیب ترسیم شکل نیز عوض می‌شود (شکل زیر).

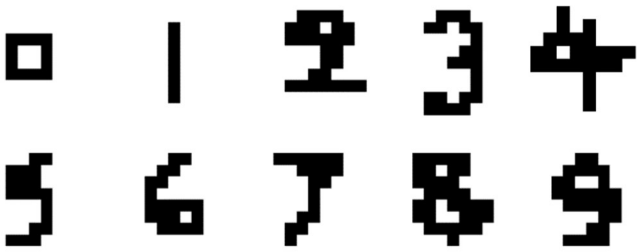


شکل ۰-۱۲ – شکل سمت راست با کروموزوم ۱۲۳۴۵۶۷ بدست آمده است که بعد از تعداد مناسبی تعویض می‌تواند به کروموزوم ۵۱۳۲۶۴۷ تبدیل شده و شکل سمت چپ را تولید کند که ترتیب درست ترسیم رقم ۳ است.

برای ارزیابی و سنجش میزان خوب بودن یک کروموزوم از همان روش دوم بخش ۲-۳-۶ استفاده شد و در واقع تصویر تخمینی را از مسیری که یک کروموزوم ارائه کرده است، با تصویر اصلی ورودی در همان رزولوشن پایین مقایسه میکنیم و میانگین مربعات خطا را محاسبه کرده و معکوس آن را به عنوان امتیاز کروموزوم در نظر می‌گیریم. در حالتی که خطا برابر صفر بدست آید مقدار آن را به ۰٫۵ تغییر می‌دهیم تا از معکوس کردن صفر جلوگیری کنیم. با این حساب برای مورد ما امتیازات کروموزوم‌ها اعدادی بین ۱ تا ۲۰۰ خواهد بود. توجه کنید که تفاوت امتیاز ۲۰۰، ۱۰۰، ۵۰ و ۲۵ به ترتیب تنها در یک پیکسل خطا است. رزولوشن پایین دقت تصویر در رزولوشن بالا به شدت کاهش می‌دهد اما فضای جستجو را برای الگوریتم ژنتیک به شدت کاهش می‌دهد.

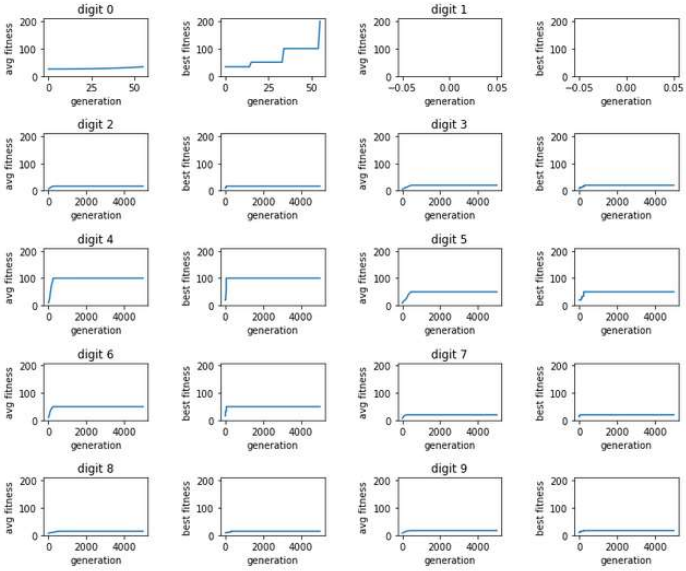
۳-۳ نتایج

با یک بار اجرای الگوریتم ژنتیک برای ارقام ۰ تا ۹، نتایج زیر بدست آمد. در شکل زیر بهترین جواب نسل آخر را نمایش داده‌ایم.



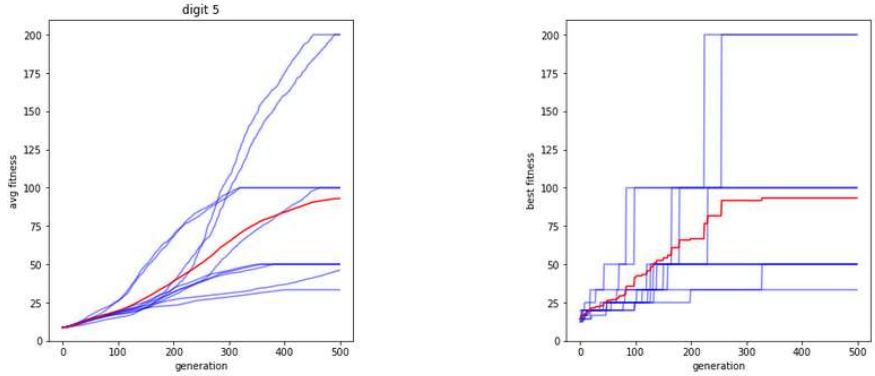
شکل ۰-۱۳ – شکل‌های تخمینی حاصل از الگوریتم ژنتیک

برای یکبار اجرای الگوریتم، داده‌های امتیاز متوسط و بالاترین امتیاز هر نسل مطابق نمودارهای زیر بدست آمد.

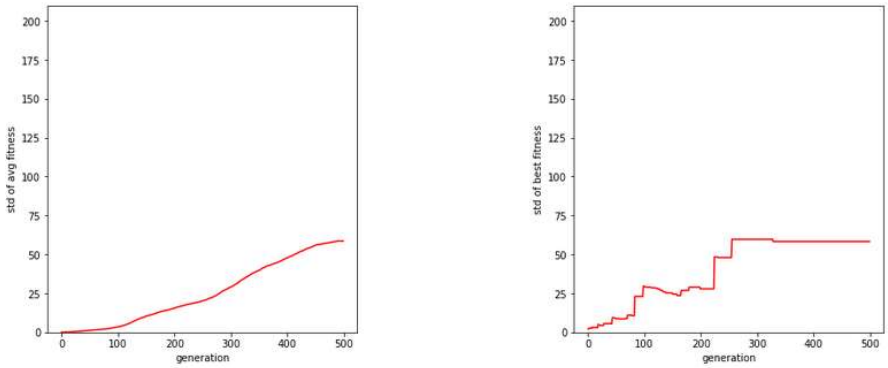


شکل ۰-۱۴ – نمودارهای میانگین امتیاز و بهترین امتیاز در هر نسل

برای رقم ۵ الگوریتم را با تعداد نسل ۵۰۰، ده بار تکرار کردیم. و میانگین و انحراف معیار بهترین امتیازات و میانگین امتیازات در هر نسل به صورت زیر بدست آمد.

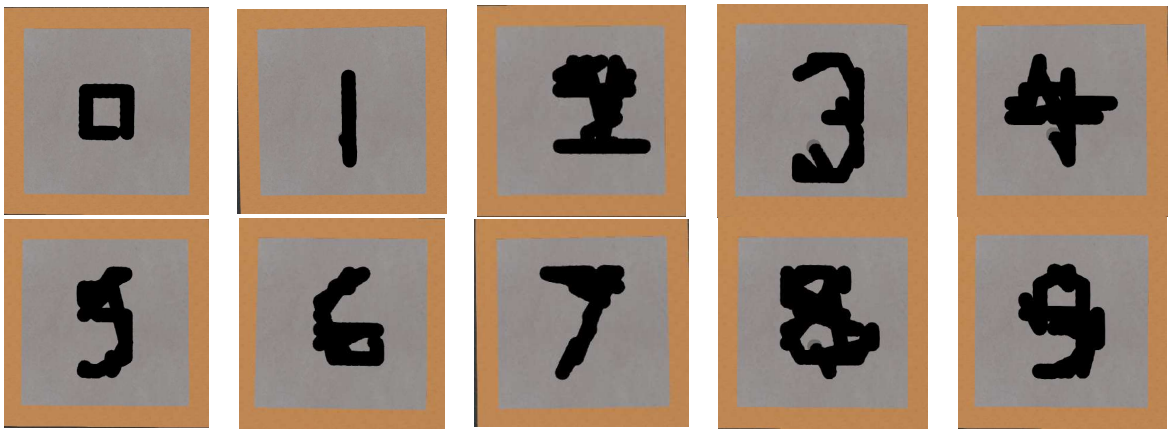


شکل ۰-۱۵ – نمودارهای آبی رنگ در پنل سمت چپ، ده منحنی میانگین امتیازها در هر نسل را نشان می‌دهد. منحنی قرمز رنگ میانگین منحنی‌های آبی است. در پنل سمت راست، نمودارهای آبی رنگ، ده منحنی بهترین امتیازها در هر نسل رسم شده است. منحنی قرمز رنگ میانگین منحنی‌های آبی است.



شکل ۰-۱۶ – انحراف معیار برای ده بار اجرا بر حسب تابعی از نسل‌ها

سپس بهترین نتایج الگوریتم ژنتیک به ربات داده شد و ارقام زیر ترسیم گردید.



شکل ۰-۱۷ – نتیجه نهایی ارقام ترسیم شده توسط ربات در محیط شبیه‌سازی

۴-۳ جمعبندی و پیشنهادها

در این پروژه راهکارهای ترسیم ارقام به نحوهای مختلف بررسی شد و با ارزیابی مقایسه هر کدام به این نتیجه رسیدیم که با شرایط و امکانات موجود، بهترین روش استفاده از الگوریتم‌های تکاملی است. در این الگوریتم ژنتیک با نمایش جایگشتی ترتیب رسم نقاط برای یک شکل بدست آمد. سپس نتایج آن را به بازو در محیط شبیه‌سازی دادیم تا ربات شکل مورد نظر را رسم کند. برای بهبود خطوط رسم شده، خطوط بلند به فواصل کوتاه‌تر تقسیم شد و نتیجه یکنواخت‌تری بدست آمد. در کل ارقام ترسیم شده بسادگی قابل تشخیص بودند و در راستای هدف پروژه نتایج قابل قبول و به نظر رضایت‌بخش بودند.

در آینده، بهینه‌سازی‌هایی در حرکت ربات می‌تواند انجام شود به نحوی که قلم ربات از روی یک خط چند بار عبور نکند. همچنین این پروژه در راستای رسم حروف، کلمات و عبارات و حتی اشکال دلخواه پیچیده می‌تواند ادامه یابد. برای حروف و اشکال پیچیده نیاز به رزولوشن‌های بالاتر داریم که به طور خودکار نیز باعث افزایش هرچه بیشتر دقت می‌شود.

منابع

[1] B. Hillburn, “Facial Anatomy and Construction,” in *Nattosoup Art and Process* (Mar 02, 2012). [Online] Available: https://nattosoup.blogspot.com/2012/03/facial-anatomy-and-construction.html

[2] P.L. Wu, Y.C. Hung, J.S. Shaw. “Artistic robotic pencil sketching using closed-loop force control.” in *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science.* 2022;236(17):9753-9762. https://doi:10.1177/09544062221096946

[3] P. O’Dowd. “A Robot That Draws and Shades with Tactile Force Feedback Sensed Through a Pencil.” (2019). in *Proceedings of EVA London 2019 (EVA 2019).* https://DOI.org/10.14236/ewic/EVA2019.19

[4] O. Deussen, T. Lindemeier, S. Pirk, M. Tautzenberger. “Feedback-guided Stroke Placement for a Painting Machine,” in *Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging.* (2012), P. 25-33.

[5] Jacobs, David. "Image Gradients". (2005). [Online] Available: http://www.cs.umd.edu/~djacobs/CMSC426/ImageGradients.pdf

[6] 7Bot, 7Bot Desktop Robot Arm – painting.(Aug 31, 2015). [Online] Available: https://www.youtube.com/watch?v=OQ3e3yCyyd4

[7] M. Adamik, J. Goga, J. Pavlovicova, A. Babinec, I. Sekaj, “Fast robotic pencil drawing based on image evolution by means of genetic algorithm,” in *Robotics and Autonomous Systems.* Volume 148, (2022), 103912, https://doi.org/10.1016/j.robot.2021.103912

[8] Y.-P. Huang, C.-H. Huang, “Real-valued genetic algorithms for fuzzy grey prediction system,” in *Fuzzy Sets and Systems*. 87 (3) (1997) 265–276.

[9] ikpy 3.3.3 (May 15, 2022). [Online] Available: https://pypi.org/project/ikpy/

[10] Phylliade/ikpy (May 15, 2022). [Online] Available: https://github.com/Phylliade/ikpy

[11] Wikipedia-Webots (December 21, 2021). [Online] Available: https://en.wikipedia.org/wiki/Webots

[12] ABB, irb4600 datasheet, “Highly productive general purpose robot.” (May 06, 2022). [Online] Available: https://search.abb.com/library/Download.aspx?DocumentID=ROB0109EN_G&LanguageCode=en&DocumentPartId=&Action=Launch

[13] ABB, Industrial robots, IRB 4600. Last updated (2022). [Online] Available: https://new.abb.com/products/robotics/industrial-robots/irb-4600

[14] A.E. Eiben_ J.E. Smith. *Introduction to Evolutionary Computing*. Springer (2015).

[15] Illustrated Guide to Recurrent Neural Networks: Understanding the Intuition (Aug 26, 2018). [Online] Available: https://www.youtube.com/watch?v=LHXXI4-IEns

[16] Illustrated Guide to LSTM's and GRU's: A step by step explanation (Sep 20, 2018). [Online] Available: https://www.youtube.com/watch?v=8HyCNIVRbSU

[17] Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) (Jun 28, 2017). [Online] Available: https://www.youtube.com/watch?v=WCUNPb-5EYI

[18] R.W. Floyd, L. Steinberg, *An adaptive algorithm for spatial grey scale*. Proceedings of the Society of Information Display **17**, 75–77 (1976).

[19] Robotic Drawing Machine https://github.com/SuperMakeSomething/robotic-drawing-machine

[20] *Super Make Something - Robotic Drawing Machine (Arduino, 3D Printing, CAD, PCB Design, Programming, Stepper Motors)*. (Aug 14, 2017). [Online Video]. Available: https://www.youtube.com/watch?v=OuCiHp43q20&t=620s

[21] Deng, L., 2012. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6), pp.141–142.

[22] *TensorFlow – cross-validation*. Last updated (Aug 12, 2022). [Online] Available: https://www.tensorflow.org/tutorials/images/classification#visualize_training_results

[23] *OpenCV 4.6.0-dev, open source computer vision - Morphological Transformations.(2022).[Online] Available: https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html*

[24] *ResearchGate question, Akhilnandh Ramesh.* (asked Jul 31, 2021). [Online] Available: https://www.researchgate.net/post/Genetic_algorithm_from_scratch_vs_library_Which_is_one_is_better

پیوست

الف) فایل‌های حاوی متن پروژه

صفحه گیتهاب پروژه حاوی فایل‌های پیاده‌سازی

https://github.com/ph504/ABB_handwriter

Abstract

In this project, the goal is to draw lines and shapes by a robot without telling the robot explicitly how to draw the shape. This project can be very important and has applications in the industry of cutting, welding, drawing and industrial design. For simplicity, we first considered drawing the English numerical digits 0 to 9 in the simulation environment by a robotic arm. We define drawing as to make continuous lines and trajectories using a pen, which is different from the work of an arbitrary printer. For this purpose, evolutionary algorithms were used to find the drawing pattern, and the results were drawn in the Webots simulation environment by the irb4600 robotic arm. We studied our suggested methods in this project, and the pros and cons of each method have been examined. There are several methods used in previous works; The most basic method is probably printers which draw line by line. There are also robots which can draw a tree or a plant, But cannot necessarily draw a given reference photo. The advantage of our algorithm is that it allows us to draw any desired shape by the robot without prior training and the need for large data i.e. the trajectory with respect to time (video data). This generalized work will also be a step towards the direction of designing designer robots.

Shahid Beheshti University

Faculty of Computer Science and Engineering

Learning hand-written digit patterns using robotic arms

By:

Arya Parvizi

A THESIS SUBMITTED

FOR THE DEGREE OF

BACHELOR OF SCIENCE

Supervisor

Dr. Armin Salimi Badr

August 2022