



**Shiraz University of Technology
Department of Computer Engineering
And Information Technology**

M.Sc. Thesis

**Optimizing Convolutional Neural Networks
for Stereo Matching by using Multi-Objective
Evolutionary Algorithms**

**BY
Ali Zoljodi**

**Supervisor
Reza Akbari Ph.D.**

**Advisor
Reza Javidan Ph.D.**

Dec-2019

IN THE NAME OF GOD
Optimizing Convolutional Neural Networks for Stereo Matching by using Multi-Objective Evolutionary Algorithms

BY

Ali Zoljodi

THESIS

SUBMITTED TO THE
DEPARTMENT OF COMPUTER ENGINEERING
AND INFORMATION TECHNOLOGY

SHIRAZ UNIVERSITY OF TECHNOLOGY
IN PARTIAL
FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE (M.Sc.)
in
COMPUTER ENGINEERING

Thesis committee members:

Akbari Reza R., Ph.D., Assistant Professor Of Computer Engineering Department
(supervisor).....

Javidan Reza R., Ph.D., Associate Professor Of Computer Engineering
Department (Advisor).....

Shamsi Nejad Babaki Pirooz P., Ph.D., Assistant Professor Of Computer
Engineering Department (Referee).....

December 2019

Abstract

Computer vision is one of the most applied sciences related to the AI. One of the computer vision applications is extracting three-dimensional scenes from stereo pictures by using convolutional deep neural networks. The problem of these networks especially in this application is their cost of processing which decreases by losing the accuracy rate in the network. The methods for optimizing these networks are generally manual which need much knowledge and effect by human mistakes. In this research, we try to utilize automatic methods like evolutionary algorithms to decrease the network's cost of processing besides preserving the current accuracy percent. To achieve this goal we optimized a piece of one of the most famous stereo matching networks by using a simulated annealing algorithm. The output of this method is saving the accuracy rate besides reducing processing costs about 22 percent.

Keywords:

Deep Neural Networks, Deep Learning, Stere Matching, NAS, AutoML



دانشگاه صنعتی شیراز

دانشکده مهندسی کامپیوتر و فناوری اطلاعات گروه مهندسی کامپیوتر

پایان نامه کارشناسی ارشد

در رشته مهندسی کامپیوتر گرایش نرم افزار

بهینه سازی شبکه های عصبی کانوولوشن در کاربرد تطبیق استریو با استفاده از
الگوریتم های تکاملی چندهدفه

نگارش:

علی ذوالجودی

استاد راهنما:

جناب آقای دکتر رضا اکبری

استاد مشاور:

جناب آقای دکتر رضا جاویدان

دی ماه ۹۸

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

بسمه تعالی

بهینه‌سازی شبکه‌های عصبی کانوولوشن در کاربرد تطبیق استریو با استفاده از
الگوریتم‌های تکاملی چندهدفه

پایان‌نامه ارائه‌شده به عنوان بخشی از فعالیت‌های تحصیلی

نگارش:

علی ذوالجودی

برای اخذ درجه کارشناسی ارشد

گروه مهندسی کامپیوتر دانشکده مهندسی کامپیوتر و فناوری اطلاعات
دانشگاه صنعتی شیراز

ارزیابی پایان‌نامه توسط هیات داوران با درجه: عالی

دکتر رضا اکبری استادیار در رشته مهندسی کامپیوتر (استاد راهنما)

دکتر رضا جاویدان دانشیار در رشته مهندسی کامپیوتر (استاد مشاور)

دکتر پیروز شمسی‌نژاد بابکی استادیار در رشته مهندسی کامپیوتر (داور)

مدیر امور آموزشی و تحصیلات تکمیلی دانشگاه:

حق چاپ محفوظ و مخصوص به دانشگاه صنعتی شیراز است.

دی ماه ۹۸

تعهد نامه اصالت اثر

اینجانب علی ذوالجودی تأیید می نماید پایان نامه/رساله با عنوان:
بهینه سازی شبکه های عصبی کانولوشن در کاربرد تطبیق استریو با استفاده از الگوریتم های

تکاملی چندهدفه

حاصل پژوهش اینجانب بوده و در صورت استفاده از انتشارات دیگران، نشانی دقیق آن مطابق مقررات ارجاع علمی نوشته شده است. ضمناً این اثر به صورت کلی یا جزئی قبلاً برای دریافت هیچ مدرک تحصیلی ارایه نشده است و بدون کسب مجوز از دانشگاه صنعتی شیراز و استاد(ان) راهنما در مجامع علمی و رسانه ها شامل همایش ها و مجلات داخلی و خارجی بصورت مکتوب یا غیرمکتوب منتشر نخواهد شد. مقالات مستخرج از این پایان نامه/رساله منحصرأً با درج اسامی استاد(ان) راهنما و وابستگی سازمانی "دانشگاه صنعتی شیراز" یا "Shiraz University of Technology" (به همراه نام دانشکده/گروه علمی) منتشر خواهد شد و از درج سایر اسامی و وابستگی های سازمانی در مقالات بدون تأیید استاد(ان) راهنما خودداری خواهد شد.

نام و نام خانوادگی و امضاء دانشجو:

حفظ حقوق پایان نامه/رساله

نتایج تحقیقات مندرج در این پایان نامه/رساله و دستاوردهای مادی و معنوی آن (شامل مقاله، فرمول، نرم افزار، سخت افزار و مواردی که قابلیت ثبت اختراع دارد) متعلق به دانشگاه صنعتی شیراز است و بدون کسب اجازه کتبی از دانشگاه صنعتی شیراز امکان فروش و ادعای مالکیت مادی یا معنوی بر آن یا ثبت اختراع از آن وجود ندارد. همچنین کلیه حقوق مربوط به چاپ، تکثیر، نسخه برداری، ترجمه، اقتباس و نظائر آن در محیط های مختلف اعم از الکترونیکی، مجازی یا فیزیکی برای دانشگاه صنعتی شیراز محفوظ است. هر گونه تضییع حقوق نامبرده موجب برخورد قانونی مطابق مقررات آموزشی، پژوهشی و انضباطی با اینجانب خواهد شد.

نام و نام خانوادگی و امضاء دانشجو:

نقل مطالب با ذکر مأخذ با رعایت مقررات ارجاع علمی بلامانع است.

مجوز بهره‌برداری از پایان‌نامه

کلیه حقوق مادی و معنوی مترتب بر نتایج پایان‌نامه متعلق به دانشگاه و انتشار نتایج نیز تابع مقرارت دانشگاهی است و با موافقت استاد راهنما به شرح زیر، بلامانع است:

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

نام استاد یا اساتید راهنما:

تاریخ:

امضا:

تقدیم به خانواده‌ی عزیزم که در تمام مراحل زندگی‌ام یاور و حامی من بوده‌اند.

تشکر و قدردانی:

اکنون که این رساله به پایان رسیده است بر خود فرض می‌دانم که از استاد ارجمندم جناب آقای دکتر اکبری و همچنین از دوست خوبم جناب آقای مهندس محمد لونی و استاد ارجمندشان جناب آقای دکتر مسعود دانش‌طلب که در تمامی مراحل تحقیق و تدوین این رساله، حامی فکری و معنوی اینجانب بوده‌اند، تشکر و قدردانی کنم.

چکیده

بهینه‌سازی شبکه‌های عصبی کانولوشن در کاربرد تطبیق استریو با استفاده از الگوریتم‌های تکاملی چندهدفه

نگارش:

علی ذوالجودی

بینایی ماشین یکی از شاخه‌های پرکاربرد در علوم مرتبط با هوش مصنوعی است. یکی از کاربردهای بینایی ماشین در سه‌بعدی سازی فضا با استفاده از دو و یا چند عکس ثبت شده از محیط و در زاویه‌های مختلف می باشد که با استفاده از شبکه‌های عصبی عمیق کانولوشن صورت می پذیرد. مشکل این شبکه‌ها در این کاربرد، توان مصرفی بالای آنهاست که کاهش آن عموماً با کاهش درستی خروجی این شبکه‌ها همراه است. روشی که عموماً برای بهینه‌سازی این شبکه‌ها استفاده می شود طراحی دستی می باشد که نیازمند دانش و تخصص بسیار زیاد و البته در معرض خطای انسانی است. در این تحقیق سعی شده است تا از روش‌های خودکار ساخت شبکه‌های عصبی نظیر الگوریتم‌های تکاملی چندهدفه بجای طراحی انسانی استفاده شود. برای این منظور در این تحقیق، قسمتی از یک شبکه‌ی عصبی پرکاربرد و شناخته شده در تطبیق استریو، با الگوریتم سردشدن شبیه‌سازی شده چندهدفه، بهبود داده شده است. خروجی این روش مدل بهینه ای از الگوریتم اولیه است که در کنار حفظ نرخ درستی، نرخ مصرف را تا ۲۲ درصد کاهش داده است.

واژه‌های کلیدی: تطبیق استریو، یادگیری عمیق، یادگیری ماشین خودکار، الگوریتم

شبیه‌سازی سردشدن

فهرست مطالب

۱	۱. فصل اول: مقدمه
۲	۱-۱- مقدمه.....
۳	۲-۱- اهداف تحقیق و اهمیت آن.....
۴	۳-۱- بیان کلیات مساله.....
۴	۴-۱- بخش‌های مختلف پایان‌نامه.....
۵	۲. فصل دوم: اصول و مبانی نظری
۶	۱-۲- مقدمه.....
۶	۲-۲- مفاهیم پایه.....
۷	۱-۲-۲- هندسه‌ی اپیپولار.....
۸	۲-۲-۲- یکسو سازی.....
۹	۳-۲-۲- جابجایی مکان پیکسل‌های متناظر.....
۱۰	۴-۲-۲- روابط برای مختصات افقی جفت تصاویر استریو.....
۱۰	۵-۲-۲- تعاریف موجود در مفهوم استریو.....
۱۱	۳-۲- روش‌های تطبیق استریو.....
۱۲	۱-۳-۲- بی‌شباهتی پیکسلی.....
۱۲	۲-۳-۲- روش‌های اندازه‌گیری معمول و تجمع هزینه.....
۱۳	۴-۲- روش‌های محاسبه‌ی عمق.....
۱۴	۱-۴-۲- روش‌های فعال.....
۱۴	۲-۴-۲- روش‌های غیرفعال.....
۱۷	۵-۲- مفاهیم پایه‌ای شبکه‌های عصبی و یادگیری عمیق.....
۱۷	۱-۵-۲- ساختار شبکه‌های عصبی کانولوشن.....
۲۱	۲-۵-۲- مدل‌های اولیه شبکه‌های عصبی کانولوشن.....
۲۹	۶-۲- جمع‌بندی و خلاصه.....
۳۱	۳. فصل سوم: مروری بر تحقیقات انجام‌شده
۳۱	۱-۳- مقدمه.....
۳۱	۲-۳- روش‌های بدون یادگیری عمیق.....

۳۲	۳-۳- یادگیری با ناظر.....
۳۴	۴-۳- یادگیری بدون ناظر.....
۳۵	۵-۳- جستجوی معماری شبکه‌های عصبی.....
۳۶	۳-۵-۱- فضای جستجو.....
۳۸	۳-۵-۲- استراتژی جستجو.....
۴۰	۳-۵-۳- استراتژی تخمین کارایی.....
۴۱	۳-۶- جمع‌بندی و مقایسه.....

۴. فصل چهارم: روش پیشنهادی برای حل مساله

۴۲	
۴۳	۴-۱- مقدمه.....
۴۳	۴-۲- بیان مساله.....
۴۳	۴-۳- علت استفاده از این روش.....
۴۴	۴-۴- شرح روش پیشنهادی.....
۴۴	۴-۴-۱- مدل پایه.....
۴۷	۴-۴-۲- الگوریتم سردشدن شبیه‌سازی شده.....
۴۸	۴-۴-۳- فضای جستجو.....
۵۰	۴-۵- پیاده‌سازی الگوریتم سردشدن شبیه‌سازی شده.....
۵۰	۴-۵-۱- تابع انتقال.....
۵۱	۴-۵-۲- تابع انرژی.....
۵۲	۴-۶- اجراهای الگوریتم سردشدن شبیه‌سازی.....
۵۳	۴-۶-۱- اجرای اول.....
۵۵	۴-۶-۲- اجرای دوم.....
۵۸	۴-۶-۳- اجرای سوم.....

۵. فصل پنجم: جمع‌بندی و پیشنهادها

۶۳	
۶۴	۵-۱- خلاصه.....
۶۴	۵-۲- جمع‌بندی.....
۶۴	۵-۳- پیشنهاد.....

مراجع

۶۶

پیوست‌ها

۷۳

فهرست شکل‌ها

- شکل ۱-۲: محل قرارگیری دوربین‌ها نسبت به نقطه‌ی مورد نظر از منظره..... ۷
- شکل ۲-۲: خط اپیپولار مشخص شده برای پیکسل x_0 در تصویر دیگر..... ۷
- شکل ۳-۲: صفحه‌ی اپیپولار..... ۸
- شکل ۴-۲: مدل هندسی یکسو سازی..... ۹
- شکل ۵-۲: هندسه‌ی سیستم استریوی مبنا..... ۱۱
- شکل ۶-۲: شمایی از روش سایه..... ۱۵
- شکل ۷-۲: عملیات لایه‌ی کانولوشن..... ۱۹
- شکل ۸-۲: عملیات کاهش دهنده‌ی بعد..... ۲۰
- شکل ۹-۲: ترکیب لایه‌ی کانولوشن و کاهش دهنده بعد..... ۲۰
- شکل ۱۰-۲: مدل S-cell در NeoCognitron..... ۲۲
- شکل ۱۱-۲: اتصال لایه‌ها در Neocognitron پایه..... ۲۳
- شکل ۱۲-۲: ساختار شبکه عصبی ارائه شده توسط LeCun برای تشخیص دست خط..... ۲۵
- شکل ۱۳-۲: شبکه عصبی کانولوشن LeNet-5..... ۲۷
- شکل ۱۴-۲: ساختار LeNet-7..... ۲۹
- شکل ۱-۳: نمایی از روابط میان سه بخش NAS..... ۳۶
- شکل ۲-۳: نمایی از فضای جستجوی زنجیره‌ای..... ۳۷
- شکل ۳-۳: فضای حالت پیچیده‌ی چند شاخه‌ای..... ۳۸
- شکل ۱-۴: معماری مدل Content-CNN..... ۴۵
- شکل ۲-۴: مدل مفهومی شبکه..... ۴۶
- شکل ۳-۴: تغییرات درستی در اجرای اول..... ۵۳
- شکل ۴-۴: تغییرات تعداد عملیات ممیز شناور در اجرای اول..... ۵۴
- شکل ۵-۴: تغییرات انرژی در اجرای اول..... ۵۴
- شکل ۶-۴: نمودار تغییرات درستی در اجرای دوم..... ۵۶
- شکل ۷-۴: نمودار تغییرات تعداد عملیات ممیز شناور در اجرای دوم..... ۵۶
- شکل ۸-۴: نمودار تغییرات انرژی در اجرای دوم..... ۵۷
- شکل ۹-۴: نمودار تغییرات در اجرای سوم..... ۵۹

- شکل ۴-۱۰ : نمودار تغییرات تعداد عملیات ممیز شناور در اجرای سوم.....۵۹
- شکل ۴-۱۱ : نمودار تغییرات انرژی در اجرای سوم.....۶۰
- شکل ۴-۱۲ : نمایی از بهترین مدل به دست آمده توسط الگوریتم سرد شدن.....۶۱
- شکل ۴-۱۳ : نمودار کاهش میزان خطا بر حسب تکرار یادگیری.....۶۱

فهرست جدول‌ها

جدول ۱-۲: ارتباط داخلی مربوط به لایه‌های کاهش بعد دوم و کانولوشن سوم.....	۲۸
جدول ۱-۳: تفکیک مقالات تطبیق استریو بر اساس روش.....	۳۵
جدول ۲-۳: تفکیک مقالات NAS بر اساس روش جستجو.....	۳۹
جدول ۳-۳: خلاصه‌ی روش‌های تخمین کارایی در NAS.....	۴۰
جدول ۱-۴: کاراکترهای موجود در رشته‌های تولیدشده توسط الگوریتم سردشدن شبیه‌سازی شده و معادل آن‌ها در شبکه‌های عصبی عمیق.....	۴۸
جدول ۲-۴: نتایج اجرای اول الگوریتم سردشدن شبیه‌سازی شده.....	۵۳
جدول ۳-۴: تغییرات بهترین مدل در اجرای دوم.....	۵۶
جدول ۴-۴: تغییرات بهترین مدل در اجرای سوم.....	۵۸
جدول ۵-۴: مقایسه‌ی نتایج مدل بهینه‌شده‌ی به دست آمده و مدل پایه‌ی Content-CNN.....	۶۲

فهرست کلمات اختصاری

FLOPS	Floating Point Operation per Second
ACC	Accuracy
CV	Computer Vision
DNN	Deep Neural Network
DL	Deep Learning
SM	Simulated Annealing
ML	Machine Learning
TF	TensorFlow
TB	TensorBoard
GPU	Graphical Processing Unit
FPGA	Field-Programmable Gate Array
CPU	Central Processing Unit
RAM	Random-Access Memory

فصل اول: مقدمه

۱-۱-مقدمه

در سال‌های اخیر فعالیت‌های گسترده‌ای در زمینه بهینه‌سازی شبکه‌های عصبی کانولوشن، در جهت افزایش درجه‌ی درستی^۱، کاهش زمان یادگیری^۲ و متوسط زمان پاسخ‌گویی^۳ این شبکه‌ها شده است. یکی از کاربردهای بروز این مدل از شبکه‌های عصبی در زمینه‌ی تطبیق استریو^۴ در تصاویر ثبت شده از یک موقعیت، برای تشخیص و تخمین عمق اجزا موقعیت و بازسازی فضای سه‌بعدی آن موقعیت هست. اما با توجه به رشد فناوری و پدیدار شدن تلفن‌های هوشمند^۵ و بردهای دیجیتال برنامه پذیر نظیر FPGA و کاربردهای گسترده‌ی آن‌ها، نیاز به قابلیت پیاده‌سازی و به‌کارگیری این شکل از شبکه‌ها در پلتفرم^۶‌های مذکور احساس شد. اما مشکل اساسی بر سر به‌کارگیری شبکه‌های کانولوشن بر روی گوشی‌های موبایل و بردهای مذکور، توان مصرفی بسیار بالای آن‌ها هست. به شکلی که اجرای یک شبکه‌ی ساده بر روی گوشی موبایل، می‌تواند در عرض چند دقیقه تمام توان مصرفی باتری آن را بکار بگیرد.

راه‌کار اصلی جهت بهینه‌سازی این کار کاهش توان مصرفی این شبکه‌های عصبی عمیق در کنار حفظ درستی آن‌ها هست که این مهم از طریق تغییر در طراحی و لایه‌بندی این شبکه‌ها حاصل می‌شود. از آنجا که برای یک شبکه‌ی عصبی عمیق بکار گرفته‌شده در یک کاربرد تعداد بی‌نهایتی طراحی منحصر به فرد وجود دارد که هر یک نتایج متفاوتی برای فاکتورهای درستی و توان مصرفی دارند، در نتیجه این مساله از نوع NP-Complete است که به صورت کلی حل آن توسط انسان و یا روش‌های جستجوی معمولی شدنی نیست. بنابراین یکی از راه‌کارهای ارائه‌شده برای این مسائل به‌کارگیری الگوریتم‌های تکاملی چندهدفه^۷، هست.

الگوریتم‌های تکاملی مبتنی بر ژنتیک^۸، علی‌رغم اینکه کارایی بسیار بالایی دارند و احتمال رسیدن به جواب در آن‌ها بسیار بالاست، اما در عمل بسیار کند می‌باشند و برای رسیدن به جواب

¹ Accuracy rate

² training time

³ Average response time

⁴ Stereo matching

⁵ Smart phone

⁶ Platform

⁷ Multi-Objective Evolutionary Algorithms

⁸ Genetic

ملزم به ایجاد افراد^۱ زیادی در نسل خود هستند. به همین دلیل برای مسائلی مانند تطبیق استریو که به صورت کلی برای سنجش هر مدل نیازمند زمان بسیار زیادی می‌باشند، زیاد کاربردی نخواهند بود. از طرفی الگوریتم‌های مبتنی بر تپه نوردی^۲ نیز علی‌رغم سرعت بالاتر با مشکل گیر افتادگی در نقاط بهینه‌ی محلی روبرو هستند. در نتیجه به نظر می‌آید یکی از روش‌هایی که می‌تواند برای این نوع مسائل مفید باشد روش سرد شدن شبیه‌سازی^۳ است که اگرچه مبتنی بر تپه نوردی بوده ولی تا حدودی مشکل گیر افتادگی در بهینه‌ی محلی نیز در آن حل شده است. در این تحقیق از الگوریتم سرد شدن شبیه‌سازی شده چندهدفه برای بهینه‌سازی شبکه‌های عصبی عمیق کانولوشن در کاربرد تطبیق استریو استفاده شده است.

۱-۲- اهداف تحقیق و اهمیت آن

با توجه به گسترش روزافزون بردهای اینترنت اشیا^۴ و گوشی‌های موبایل هوشمند^۵ و از طرفی ناپایداری اتصالات شبکه‌ای و همچنین سرعت پایین و هزینه‌ی انتقال اطلاعات میان آن‌ها و ابزار مذکور، نیاز به توسعه‌ی کاربردهایی نظیر سه‌بعدی سازی مناظر به وسیله‌ی تصاویر استریو بروی این ابزار بسیار محسوس هست. که البته این مهم با چالش‌هایی نظیر توان مصرفی بالای این کاربردها روبروست. از طرفی کاهش توان مصرفی به گونه‌ای که در آن کیفیت کار شبکه مطرح نباشد می‌تواند باعث کاهش چشم‌گیری در درستی و دقت آن شبکه، باشد.

از این رو نیاز است که این مساله نه به دست انسان بلکه با به‌کارگیری روش‌های ایجاد شبکه‌های عصبی عمیق به صورت خودکار و به وسیله‌ی جستجوهای هدفمند و تکاملی حل شود. هدف اصلی این تحقیق بهینه‌سازی شبکه‌های عصبی مورد کاربرد در زمینه‌ی تطبیق استریو به وسیله روش‌های جستجوی چندهدفه^۱ است. در این تحقیق دو فاکتور میزان درستی و بهینگی مصرف اهداف در نظر گرفته‌شده برای بهبود حداکثری خواهند بود.

¹ Individuals

² Hill Climbing

³ Simulated Annealing

⁴ Internet of Things (IoT)

⁵ Smart Phones

۱-۳- بیان کلیات مساله

یکی از مسائلی که امروزه در توسعه‌ی شبکه‌های عصبی عمیق علی‌الخصوص برای کاربردهای با محاسبات سنگین نظیر تطبیق استریو وجود دارد، پیدا کردن معماری شبکه‌ای است که در کنار دقت پردازشی بالا، از نظر زمان و هزینه‌ی پردازش هم بهینه باشد. روش‌های معمول برای طراحی چنین شبکه‌هایی عموماً دستی است که ملزم به تخصص بسیار زیاد در توسعه این شبکه‌ها هست و همچنین در معرض خطاهای انسانی نیز قرار دارد علاوه بر آن، از آن جا که طراحی‌های دستی اساساً چهارچوب محور بوده و در آن‌ها ابتکار عمل تا حدودی پایین است، فرصت پیدا کردن مدل‌های ابتکاری در آن‌ها پایین است. از این رو، جهت کاهش خطاها و ساده‌سازی عملیات طراحی یک شبکه‌ی عصبی، از روش‌های خودکار مبتنی بر الگوریتم‌های تکاملی استفاده می‌شود که در کنار سرعت و اتکای بالا، توانایی جستجو بر پایه‌ی چندهدف بهینه‌سازی را نیز دارا می‌باشند. اما مشکل عمده‌ای که در شبکه‌های نظیر کاربرد تطبیق استریو وجود دارد هزینه بر بودن سنجش برازندگی این شبکه‌هاست که باعث می‌شود مدل‌های خیلی زیادی را نتوان تولید نمود. این مشکل، کار را برای استفاده از الگوریتم‌های چندهدفه مبتنی بر ژنتیک سخت می‌کند. از این رو، یکی از راه‌حل‌های چنین مسائلی استفاده از روش‌های پیمایشی نظیر تپه نوردی و سرد شدن شبیه‌سازی شده است. در این تحقیق برای طراحی شبکه‌ی عصبی عمیق از الگوریتم سرد شدن شبیه‌سازی شده‌ی چندهدفه استفاده شده است.

۱-۴- بخش‌های مختلف پایان‌نامه

در بخش دوم مفاهیم مرتبط با پایان‌نامه بررسی می‌شوند. بخش سوم مروری بر تحقیقات صورت گرفته بررسی می‌شود و در فصل چهارم روش ارائه شده و نتایج آن تحلیل می‌شود و در فصل پنجم به خلاصه‌ی کارهای صورت گرفته و جمع‌بندی و پیشنهادات برای تحقیقات آینده اختصاص داده خواهد شد.

فصل دوم: اصول و مبانی نظری

۲-۱-مقدمه

انطباق استریو فرایندی است که با هدف استخراج منظره‌های سه‌بعدی از دو و یا چند تصویر دوبعدی ثبت شده از منظره‌ی مورد نظر به کار می‌رود. برای اینکه بتوان از یک تصویر واقعیت سه‌بعدی آن را استخراج کنیم، نیاز داریم که عمق هر شی در آن تصویر، یعنی فاصله‌ی آن شی با لنز دوربین، را تخمین بزنیم و سپس با استفاده از فرمولی که در ادامه به آن اشاره می‌کنیم عمق هر شی را تشخیص دهیم. در این فصل با مفاهیم، لغات کلیدی و اساس کار در فرایند انطباق استریو، روش‌های تشخیص نقاط مشابه در تصاویر ثبت شده از یک منظره و تکنیک‌های بهینه‌سازی آن، آشنا می‌شویم.

۲-۲-مفاهیم پایه

روش محاسبه‌ی عمق اشیا در منظره به این صورت است که نقاط مشابه را در تصاویر ثبت شده پیدا می‌کنند و با استفاده از فرمول زیر عمق هر شی را بر اساس اختلاف فاصله‌ی نقاط مشابه در تصاویر از یک نقطه‌ی صفر ثابت، محاسبه می‌کنیم [۱].

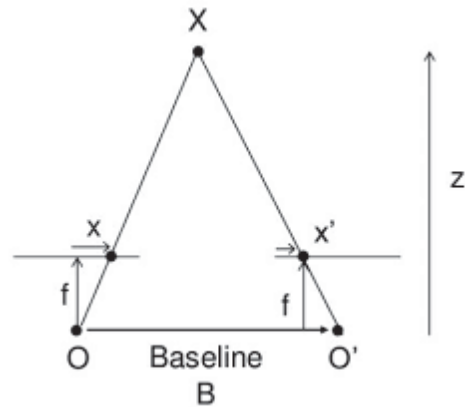
$$d = f \frac{B}{z} \quad (۱-۲)$$

که در آن d ، f ، B و z به ترتیب بیانگر اختلاف نقاط^۱، فاصله‌ی کانونی عدسی^۲ که برای تمام دوربین‌ها باید یکسان باشد، طول خط واصل دو عدسی^۳ و عمق شی مورد نظر می‌باشند.

^۱ Disparity

^۲ Focal Length

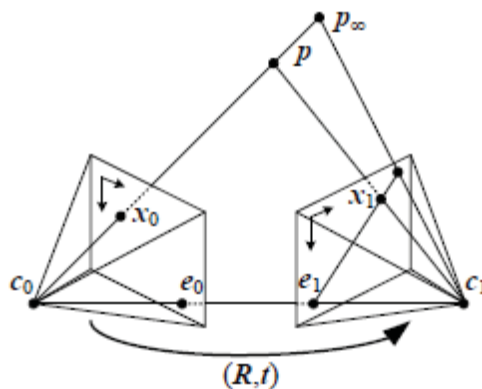
^۳ Baseline



شکل ۱-۲: محل قرارگیری دوربین‌ها نسبت به نقطه‌ی مورد نظر از منظره [۲]

۱-۲-۲- هندسه‌ی اپیپولار^۱

از آنجا که تعداد نقاط در دو تصویر بسیار زیاد است نیاز است که تعداد نقاط کاندید برای مشابهت را کاهش داد. در تصویر زیر دیده می‌شود که به ازای پیکسل x_0 در تصویر چپ یک خط به نام خط اپیپولار^۲ وجود دارد که از مرکز دوربین c_0 به بی‌نهایت رفته و از پیکسل مورد نظر عبور می‌کند. [۳]



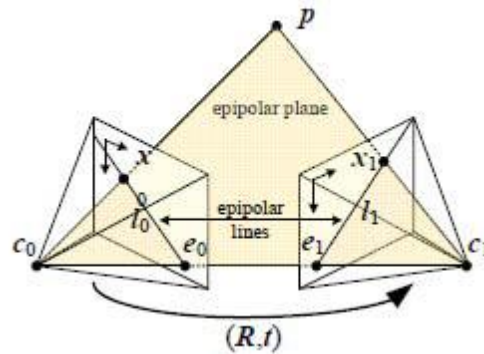
شکل ۲-۲ خط اپیپولار مشخص شده برای پیکسل x_0 در تصویر دیگر [۳]

حال اگر بازتاب خط اپیپولار تصویر راست را بروی تصویر چپ و بازتاب خط اپیپولار تصویر چپ را بروی تصویر راست به دست آوریم از تلاقی آن‌ها و بسط آن تا بی‌نهایت، یک صفحه به نام صفحه‌ی

¹ Epipolar Geometry

² Epipolar Line

اپیپولار^۱ به دست می‌آید که می‌توان گفت تمام نقاط کاندید برای مشابهت با نقطه‌ی مورد نظر در تصویر دیگر در این صفحه قرار دارند.



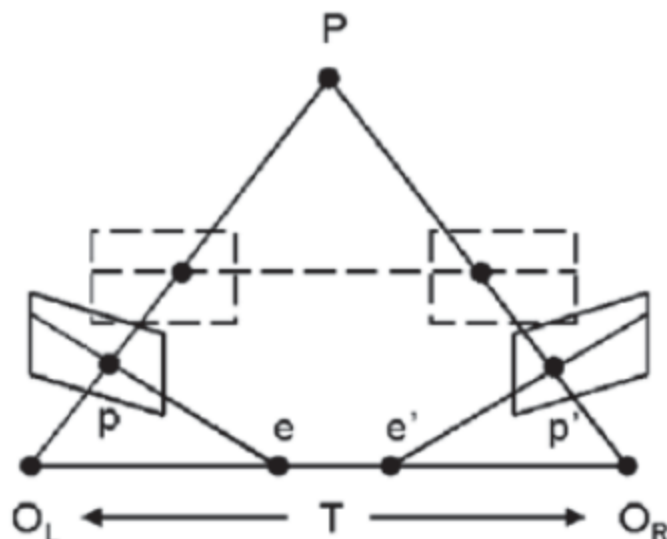
شکل ۲-۳ صفحه‌ی اپیپولار [۳]

۲-۲-۲- یکسو سازی^۲

برای کاهش پیچیدگی تطبیق پیکسل‌ها، لازم است که یکسو سازی تصاویر اعمال شود. هدف از یکسو سازی تصاویر این است که خطوط اپیپولار دو تصویر، به صورت افقی هم خط شوند. این عمل می‌تواند با انتقال خطوطی که دچار چرخش شده‌اند، جابه‌جایی و مایل کردن تصاویر دوربینیها انجام شود. در انتقالها از اطلاعات درونی دوربین و اطلاعات موقعیت دوربینیها و جهت دوربینیها استفاده می‌شود. شکل (۲-۴) این فرآیند را نمایش می‌دهد. نقطه‌ی سه بعدی P روی صفحات تصاویر دوربینیهای چپ و راست روی نقاط p و p' نگاشته می‌شود. خطوط اپیپولار از نقاط p و p' در طول صفحات تصویر تا خط پایه که بین نقاط کانونی دوربینیها یعنی O_L و O_R قرار دارد، کشیده می‌شود. این خطوط، خط پایه را در نقاط e و e' قطع می‌کنند. دو نقطه‌ی منطبق p و p' همواره به ترتیب جایی روی خطوط اپیپولار ep و $e'p'$ قرار دارند. این ساختار هندسی اساس یکسو سازی تصاویر را تشکیل می‌دهد. [۴]

¹ Epipolar Plane

² Rectification



شکل ۴-۲: مدل هندسی یکسو سازی [۴]

پس از اینکه یکسو سازی تصویر انجام شد، خطوط اپیپلار دو نقطه‌ای تصویر شده به صورت موازی و افقی در طول صفحه‌ی جدید تصاویر قرار می‌گیرند. مسئله‌ی تطبیق استریو به این ترتیب به یک جستجوی یک بعدی در طول خطوط افقی به جای جستجوی دو بعدی تبدیل می‌شود.

۳-۲-۲- جابجایی مکان پیکسل‌های متناظر

در تطابق استریو، هدف یافتن پیکسل‌های منطبق در تصویر ورودی داده شده هست و نتایج یافتن پیکسل‌های منطبق در یک نگاشت جابه‌جایی^۱ ذخیره می‌شود. واژه جابجایی می‌تواند به عنوان اختلاف افقی بین دو پیکسل تعریف شود و نگاشت جابجایی، یک مقدار مشخص برای این اختلاف افقی پیکسل‌ها برای هر پیکسل تعریف می‌کند.

¹ Disparity Map

۲-۲-۴- روابط برای مختصات افقی جفت تصاویر استریو

با استفاده از نگاشت جابجایی، مختصات پیکسلی اشیاء در تصاویر استریو می‌توانند به صورت زیر به هم مرتبط شوند [۵، ۶].

$$x_{RIGHT} = x_{LEFT} - d_{LEFT}(x_{LEFT}, y) \quad (2-2)$$

$$x_{LEFT} = x_{RIGHT} - d_{RIGHT}(x_{RIGHT}, y) \quad (3-2)$$

$$x_{SEARCH} = x_{REF} - Sd_{REF}(x_{REF}, y); S = \pm 1 \quad (4-2)$$

$$y = y_{LEFT} = y_{RIGHT} \quad (5-2)$$

در روابط بالا، d_{RIGHT} و d_{LEFT} نگاشت‌های جابجایی برای دیدهای چپ و راست را نشان می‌دهند و x برابر است با موقعیت عرضی پیکسل. که فرمول کلی آن برابر است با فرمول (۲-۴).

۲-۲-۵- تعاریف موجود در مفهوم استریو

در این بخش، برخی از اصطلاحاتی که در مطالب قبلی و یا در آینده مورد استفاده قرار گرفتند و توصیف کاملی از آن‌ها ارائه نشده است را به تفصیل شرح می‌دهیم:

- فاصله‌ی کانونی: در یک سیستم تصویربرداری مبتنی بر لنز و CCD^1 ، مجموعه لنز که متشکل از چندین عدسی محدب و مقعر است با یک عدسی محدب به فاصله‌ی کانونی f ، مدل می‌شود.
- صفحه‌ی تصویر^۲: محل تشکیل تصویر که آن را صفحه‌ی تصویر می‌گویند، صفحه‌ای عمود بر محور عدسی و به فاصله‌ی f در پشت آن هست.
- مرکز تجسم: معمولاً محور نوری عدسی را محور مختصات Z در نظر می‌گیریم و محل

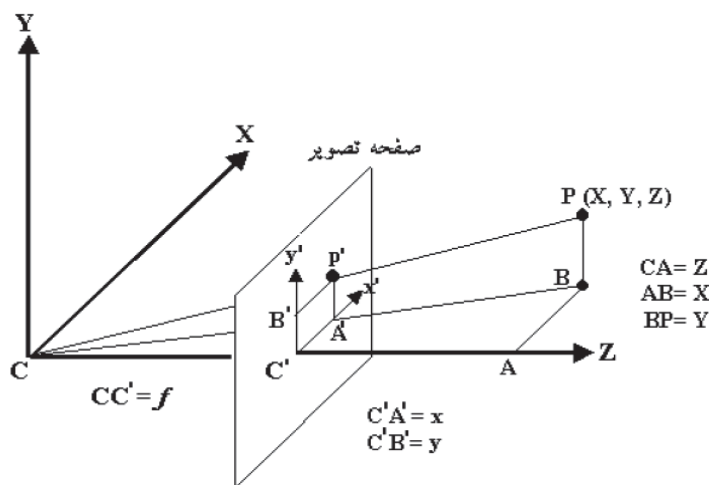
¹ Charged Couple Device

² Image Plane

تقاطع محور Z و عدسی را مرکز مختصات سه بعدی یعنی C در نظر می‌گیرند. C را مرکز تجسم نیز می‌گویند. برای سادگی در اکثر مراجع، صفحه تصویر در جلوی مرکز تجسم و به فاصله از آن در نظر گرفته می‌شود.

روابط پرسپکتیو: اگر P نقطه‌ای در فضای سه بعدی باشد، P' یعنی تصویر P در صفحه تصویر، از برخورد CP با صفحه تصویر به دست می‌آید. اگر (X, Y, Z) مختصات P در فضای سه بعدی و (x, y) مختصات P' در صفحه تصویر باشد، با توجه به شکل زیر می‌توان رابطه‌ی زیر را نوشت.

$$y = f \frac{Y}{Z}, x = f \frac{X}{Z} \quad (۶-۲)$$



شکل ۲-۵: هندسه‌ی سیستم استریوی مبنا [۴]

۲-۳- روش‌های تطبیق استریو

این قسمت برخی مفاهیمی که در روش‌های استریوی امروزی رایج هستند، را معرفی می‌کند. هدف این قسمت ایجاد مقدمه‌ای در این زمینه است، پیش از آنکه نگاه دقیق‌تر به برخی روش‌های تطبیق استریوی امروزی داشته باشیم.

۲-۳-۱- بی‌شابهتی پیکسلی^۱

به منظور یافتن پیکسل‌های متناظر در یک تصویر مرجع و تصویر جستجو، مشخص است که به یک اندازه‌گیری شباهت بین پیکسل‌ها نیاز است. اگرچه رایج تر این است که از واژه اندازه‌گیری بی‌شابهتی یا هزینه‌ی تطبیق استفاده شود که با کاهش شباهت بین دو پیکسل در حال مقایسه، افزایش می‌یابد. یک روش رایج برای نمایش هزینه‌ی تطبیق از طریق تابع $C(x,y,d)$ با مختصات تصویر مرجع و اختلاف پیکسلی هست. فضای اختلاف پیکسلی توسط مختصات پیکسل‌های تصویر در دست و محدوده‌ی جستجوی اختلاف ایجاد می‌شود. به طور معمول، یک محدوده‌ی جستجوی اختلاف باید به طور دستی مشخص شود و به ویژگی‌های دو تصویر ورودی بستگی دارد.

۲-۳-۲- روش‌های اندازه‌گیری معمول و تجمع هزینه

در طی سال‌ها، تعداد زیادی روش برای اندازه‌گیری جابجایی ارائه شده است. روش‌های مرسوم که برای مقایسه دو پیکسل متناظر استفاده می‌شوند عبارت‌اند از: اختلاف مطلق شدت روشنایی^۲ (AD) مربع اختلاف شدت روشنایی^۳ (SD) و اختلاف مطلق گرادیان^۴ ($GRADE$). با گسترش مقایسه به نواحی مربعی پنجره مانند که مرکز آن‌ها بر روی پیکسل‌های مرجع و جستجو قرار دارد، این اندازه‌گیری‌ها به جمع اختلاف مطلق شدت روشنایی (SAD)، جمع مربع اختلاف روشنایی (SSD) و جمع اختلاف مطلق گرادیان ($SGRAD$) تبدیل شده است. هزینه‌های جمع شده روی نواحی پنجره‌ای به تجمع هزینه معروف است.

مهم‌ترین مزیت تجمع هزینه کاهش حساسیت به نویز نسبت به مقایسه پیکسلی است. اگرچه، روش‌های مبتنی بر ناحیه، نتایج با جزئیات کمتری تولید می‌کنند. بنابراین، یک تناسب بین حساسیت به نویز و از دست رفتن جزئیات هنگام انتخاب سائز پنجره باید اعمال شود. توابع هزینه برای روش‌های اندازه‌گیری بی‌شابهتی بحث شده در بالا می‌توانند به صورت زیر باشند:

¹ Pixel Dissimilarity² Absolute Intensity Difference³ Squared Intensity Difference⁴ Absolute Gradient Difference

$$C_{AD}(x, y, d) = |I_L(x, y) - I_R(x - d, y)| \quad (۷-۲)$$

$$C_{SD}(x, y, d) = |I_L(x, y) - I_R(x - d, y)|^2 \quad (۸-۲)$$

$$C_{GRAD}(x, y, d) = |\nabla_x I_L(x, y) - \nabla_x I_R(x - d, y)| + |\nabla_y I_L(x, y) - \nabla_y I_R(x - d, y)| \quad (۹-۲)$$

$$C_{SAD}(x, y, d) = \sum_{(u,v) \in W(x,y)} |I_L(u, v) - I_R(u - d, v)| \quad (۱۰-۲)$$

$$C_{SAD}(x, y, d) = \sum_{(u,v) \in W(x,y)} |I_L(u, v) - I_R(u - d, v)|^2 \quad (۱۱-۲)$$

$$C_{GRAD}(x, y, d) = \sum_{(u,v) \in W(x,y)} |\nabla_x I_L(u, v) - \nabla_x I_R(u - d, v)| + \sum_{(u,v) \in W(x,y)} |\nabla_y I_L(u, v) - \nabla_y I_R(u - d, v)| \quad (۱۲-۲)$$

در اینجا I_L و I_R توابع شدت رو شنایی تصاویر چپ و راست هستند و C برابر است با پارامتر اختلاف. $W_{(x,y)}$ پنجره ایست که موقعیت (x, y) را احاطه می‌کند و ∇_x و ∇_y عملگرهای گرادیان هستند.

۲-۴- روش‌های محاسبه‌ی عمق

در حالت کلی دو دسته‌بندی اصلی به منظور تخمین عمق وجود دارد: دسته اول روش‌های فعال که شامل روش‌های مبتنی بر انواع ارسال پرتوهای انرژی به سمت شی و اندازه‌گیری میزان دریافت آن است و دسته دوم روش‌های غیرفعال که مبتنی بر تصویر هستند.

۲-۴-۱ - روش‌های فعال^۱

روش‌های فعال در حالت کلی مبتنی بر اطلاعات مربوط به نورپردازی صحنه می‌باشند و علاوه بر دوربین، ابزار کمکی دیگری نیز در محاسبه فاصله جسم مورد نظر، نقش ایفا می‌کنند. از انواع این روش‌ها می‌توان به ارسال پالس‌های مربعی اشعه لیزر اشاره کرد که در آن نشانه‌های منعکس شده از اشیا به سمت دوربین شامل تمام اطلاعات مورد نیاز برای ساخت نقشه عمق می‌باشند.

۲-۴-۲ - روش‌های غیرفعال^۲

از این دسته از روش‌ها، روش‌های مبتنی بر تصویر هستند و شامل دو روش کلی استفاده از دو دوربین و یک دوربین می‌باشند. در این دسته بعضی از متدها فقط از یک تصویر استفاده می‌کنند و در بعضی به بیشتر از یک تصویر نیاز دارند که برای بدست آوردن آن‌ها باید از چند دوربین و یا یک دوربین که پارامترهای آن و یا موقعیت آن تغییر پیدا می‌کند استفاده کرد. در این روش‌ها بدون هیچ دانش قبلی از صحنه‌ی مورد بررسی، تخمین عمق با کمک فقط یک تصویر از صحنه امکان‌پذیر نیست. بنابراین در این دسته از اطلاعاتی از قبیل گرادیان بافت، سطح سایه و غیره باید استفاده کرد و به یک سری پیش‌فرض‌های اکتشافی نیاز داریم که باعث شده نتوانیم عمق واقعی را تخمین بزنیم. در بررسی روش‌های غیرفعال می‌توان این‌گونه در نظر گرفت که این روش‌ها مبتنی بر اطلاعات متن و موقعیت اشیا و صحنه و همچنین اطلاعات سه بعدی اشیا می‌باشند. این روش‌ها خود بر اساس چگونگی عملکردشان به زیرگروه‌هایی تقسیم می‌شوند که در زیر در مورد آن‌ها بیشتر صحبت می‌شود.

■ روش مبتنی بر مدل سه بعدی وسایه

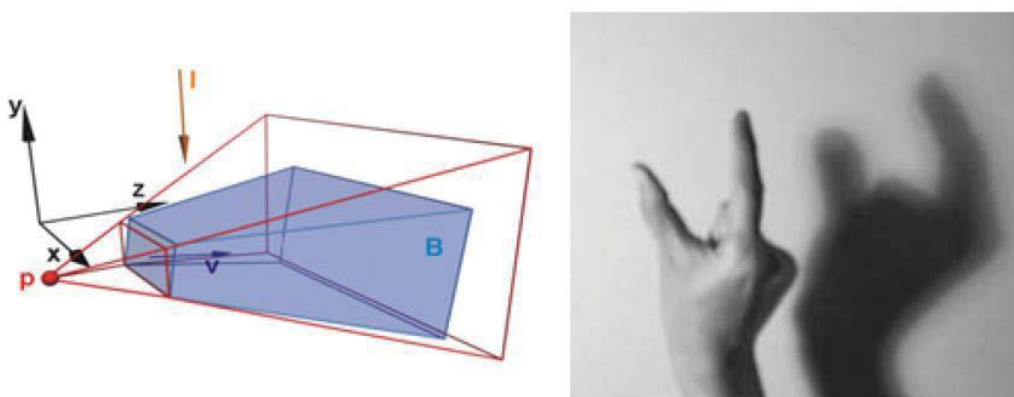
روش یک دوربین خود شامل دو نوع استاتیک و دینامیک است. در نوع دینامیک یک دوربین، با در نظر گرفتن اطلاعات مدل سه بعدی اشیا و هم‌چنین اطلاعات مربوط به موقعیت دوربین و صحنه، عمق آن بررسی می‌شود. در این روش، برای محاسبه فاصله یا عمق از یک دنباله تصاویر برداشته شده با یک دوربین استفاده می‌شود که بر مبنای مدل هندسی جسم و موقعیت دوربین ارائه شده است. در

¹ Active

² Passive

این روش‌ها اساس اندازه‌گیری بر میزان تغییر اندازه تصویر شی مورد نظر بر اثر حرکت دوربین به اندازه معلوم هست.

در برخی روش‌ها از میزان تغییرات سطح خاکستری تصویر و نور موجود در صحنه به منظور تخمین عمق استفاده می‌شود که از انواع آن می‌توان به روش استفاده از سایه‌های موجود در صحنه اشاره کرد (شکل زیر). در روش استفاده از سایه‌های موجود، با داشتن تصویری از صحنه و یک منبع نوری شناخته شده و همچنین مدل بازتابندگی می‌توان از تغییرات سایه گذاری برای ارزیابی شکل در صحنه استفاده کرد [۷، ۸].



شکل ۲-۶: شمایی از روش سایه [۷]

□ روش مبتنی بر زوم^۱

در روش استاتیک تک دوربین دو روش عمده عمق با استفاده از زوم و عمق با استفاده از روش بدون نیاز زوم وجود دارد. در روش‌های مرسوم و قراردادی عمق با استفاده از زوم یا بدون زوم، تصاویر متعدد از یک نقطه دید با فاصله کانونی‌های مختلف توسط دوربین گرفته می‌شود. در این روش‌ها دوربین ثابت در نظر گرفته شده و پارامترهای داخلی دوربین نیز محاسبه می‌شوند. در روش مبتنی بر زوم از تصاویر گرفته شده از شی در فواصل مختلف استفاده می‌شود یا با تنظیم کردن پارامترهای دوربین از جمله فاصله لنز دوربین از پرده تصویر V ، فاصله کانونی F و میزان باز و

¹ Focus

بسته بودن روزنه دوربین، سطح زوم‌های متفاوتی را می‌توانیم به دست آوریم و تصاویر متعددی را از صحنه با تغییر پارامترهای دوربین به دست آوریم. اساسی‌ترین ضعف این روش مدت زمان لازم برای به دست آوردن تصاویری است که برای محاسبه عمق لازم است. در عمل در حدود ده و یا بیشتر تصویر بدین منظور لازم است [۹].

□ روش بدون نیاز به زوم^۱

در روش بدون نیاز به زوم از تصاویر گرفته شده با فاصله کانونی متفاوت استفاده می‌شود [۱۰]. در حقیقت مهم‌ترین مسئله در روش مبتنی بر زوم و بدون نیاز به زوم، به دست آوردن مقیاسی برای اندازه‌گیری میزان تاری در مشاهدات مختلف هست. روش بدون نیاز به زوم در حالت کلی به دو دسته فعال و غیرفعال تقسیم می‌شود.

حالت فعال روش بدون نیاز به زوم اولین بار توسط Girod و Scherrock در سال ۸۹ مطرح شد. در این روش پرتو مشخصی از نور به صحنه تابانده می‌شود. تصویر توسط یک دوربین مجهز به اسپلیتر پرتو نور ثبت شده و با توجه به اینکه پرتو نور تابیده مشخص است، میزان تاری ناشی از عمق شیء قرار گرفته در بدون نیاز به زوم، سنجیده می‌شود. این روش با اینکه نتایج مناسبی دارد ولی فقط در محیط‌های کنترل شده مناسب است و در بیشتر موارد کاردی نمی‌توان استفاده کرد.

حالت غیرفعال بدون نیاز به زوم توسط Pentland در سال ۸۷ بیان شد [۱۱]. در این روش از حداقل دو تصویر با فاصله کانونی متفاوت استفاده می‌شود. در حقیقت مشاهدات بی‌شماری از تصویر گرفته شده با پارامترهای متفاوت دوربین وجود دارد. بر اساس تفاوت میزان تاری بین دو مشاهده با فاصله کانونی متفاوت و خارج از فاصله زوم دوربین، عمق شیء در صحنه تخمین زده می‌شود. در این روش تصویر کاملاً زوم شده و لازم است که یک تصویر واضح از صحنه داشته باشیم.

روشی دیگر به منظور به دست آوردن عمق اشیای متحرک در صحنه با استفاده از میزان تاری تصاویر ناشی از غیر زوم دوربین و ویژگی‌های آن مطرح شده است. برخلاف دیگر روش‌های رایج به منظور تخمین عمق اشیاء در صحنه که مبتنی بر ویژگی‌های سه بعدی اشیاء و صحنه می‌باشند، در اینجا تنها پارامترهای ثابت دوربین در نظر گرفته می‌شوند. بدین ترتیب تغییرات فاصله شیء از دوربین نسبت مستقیم با میزان تاری شیء در تصویر دارد. در حقیقت میزان تاری شدن با یک نسبت مستقیم در عمق مقابل دوربین انتشار می‌یابد که با تغییرات سطح خاکستری در اطراف لبه‌های اشیاء قابل مشاهده هست. با توجه به پارامترهای ثابت دوربین، تابع انتشار تاری در صحنه تخمین زده می‌شود.

¹ Defocus

شود. سپس با توجه به تغییرات ایجاد شده در اثر تاری که باعث آغشتگی در لبه‌های اشیا می‌شود، مقادیر به دست آمده با تابع انتشار سنجیده می‌شود و مقدار عمق اولیه تخمین زده می‌شود. در این روش برخلاف دیگر روش‌های مشابه، به تصویر شی در حالت زوم نیاز نیست و تنها با یک تصویر می‌توان عمق یکنواخت نسبی را برای شی تخمین زد [۱۲].

□ روش استریو^۱

روش دو دوربین یا دو چشمی که از آن به عنوان روش استریو نیز نام می‌برند، به معنی استخراج و تخمین عمق با استفاده از میزان عدم تطابق بین دو تصویر از یک صحنه است و از دو تصویر گرفته شده با استفاده از دو دوربین در مکان‌های متفاوت استفاده می‌کند. این روش در واقع مشابه سیستم بینایی انسان عمل می‌کند. ایده امکان استخراج اطلاعات عمق با استفاده از دو دوربین اولین بار توسط Charles Wheatstone در سال ۱۸۳۸ مطرح گردید. وی بیان کرد به دلیل قرار گرفتن چشم‌ها در موقعیت‌های افقی متفاوت، تصاویر به دست آمده از هر چشم با دیگری متفاوت است و اشیایی که در فاصله‌های متفاوتی از چشم قرار دارند در دو تصویر در موقعیت‌های مکانی افقی متفاوتی قرار گرفته که با در نظر گرفتن این امر امکان اندازه‌گیری فاصله شی تا دوربین فراهم می‌شود.

در استریو بر پایه دید، دو تصویر از یک صحنه از دو نمای مختلف گرفته می‌شود. یک اختلاف نسبی مشخصه‌ها بین موقعیت‌های اشیا در تصاویر وجود دارد. این اختلاف به عمق صحنه مربوط می‌شود. از این رو اشیا نزدیک تر به دوربین اختلاف بیشتر و اشیا دورتر از دوربین اختلاف کمتری دارند. بر اساس این اصل می‌توان عمق را در صحنه به دست آورد. این بر اساس اصل دو بعدی است که انسان‌ها اکثراً عمق را مشاهده می‌کنند.

۲-۵- مفاهیم پایه‌ای شبکه‌های عصبی و یادگیری عمیق

۲-۵-۱- ساختار شبکه‌های عصبی کانولوشن

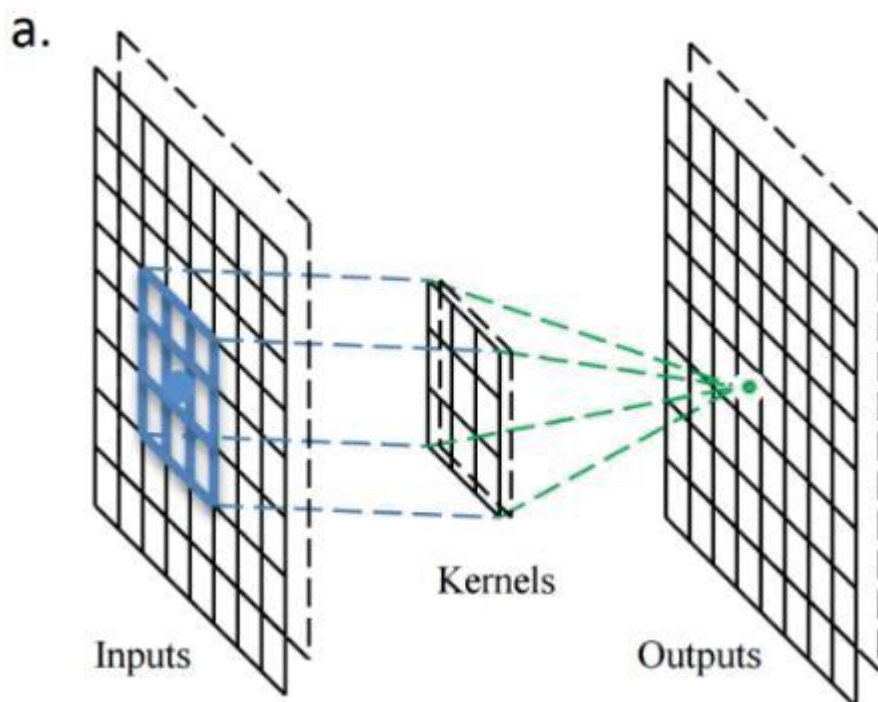
شبکه‌های عصبی کانولوشن از یک لایه ورودی، یک لایه خروجی و چندین لایه مخفی

¹ Stereo

تشکیل شده‌اند. لایه‌های مخفی شامل لایه‌های کانولوشن و لایه‌های کاهش‌دهنده بعد است که معمولاً بصورت یک‌درمیان حاضر می‌شوند. لایه‌های کانولوشن وظیفه استخراج ویژگی و لایه‌های کاهش‌دهنده بعد وظیفه کاهش وضوح ورودی‌ها را بر عهده دارند. هر لایه کانولوشن دارای یک فیلتر کانولوشن و هر لایه کاهش‌دهنده بعد دارای یک فاکتور کاهش بعد است.

ناحیه دریافت محلی، هر سلول در لایه است که ورودی را از مجموعه‌ای از سلول‌ها در یک هم‌سایگی کوچک در لایه قبل دریافت می‌کند. با ناحیه‌های دریافت محلی نرون‌ها می‌توانند ویژگی‌های بصری اصلی مانند لبه‌ها و گوشه‌ها را شناسایی کنند. CNN یک خصوصیت کلیدی تصویر را تأیید می‌کند و آن این است که پیکسل‌های نزدیک به هم وابستگی بیشتری نسبت به پیکسل‌های دور از هم دارند. این خصوصیت از طریق استخراج ویژگی‌هایی که فقط به محل کوچکی از تصویر وابسته‌اند، برآورده می‌شود. اطلاعات این ویژگی‌ها در لایه‌های بالاتر ادغام می‌شود و ویژگی‌های پیچیده‌تر استخراج می‌شود نهایتاً اطلاعات کل تصویر استخراج می‌شود.

در یک لایه کانولوشن، کل فضای داده ورودی توسط کرنل $u * v$ پیمایش شده و هر فضای ورودی که بر اساس اندازه کرنل، $u * v$ در نظر گرفته می‌شود و ناحیه دریافت محلی نام دارد در کرنل مربوطه ضرب شده و با جمع المان‌های ماتریس نتیجه و یک بایاس آموزش پذیر و تابع فعال‌سازی (که به طور معمول تانژانت هایپربولیک استفاده می‌شود) روی مقدار حاصل خروجی این لایه را خواهیم داشت. سپس کرنل یکی به را ست و یکی به پایین شیف‌ت داده می‌شود و عملیات تکرار می‌شود. در لایه‌های کانولوشن تعداد k کرنل $u * v$ داریم که به ازای هر کرنل یک تصویر ویژگی خواهیم داشت. عملیات کانولوشن را در شکل زیر مشاهده کنید:

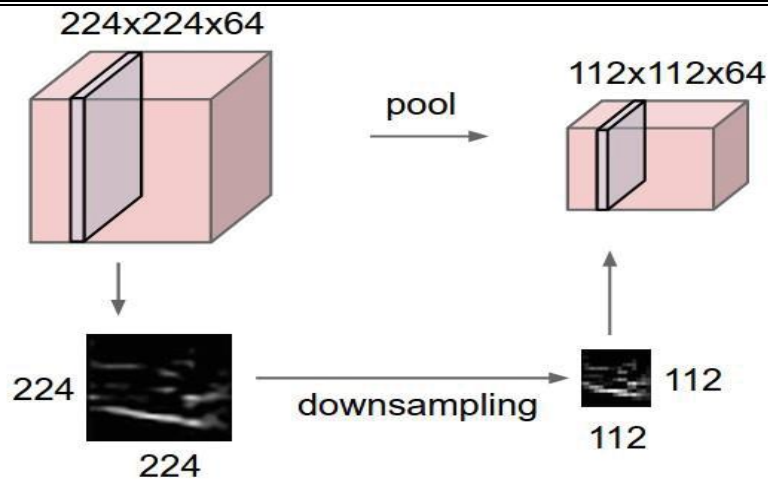


شکل ۲-۷: عملیات لایه‌ی کانولوشن [۱۳]

تصاویر ویژگی در لایه‌های کانولوشن مطابق رابطه‌ی زیر حاصل می‌شود [۱۳]:

$$C_{k,l}^i = g(I_{k,l}^i \otimes W_{k,l} + B_{k,l}) \quad (۱۳-۱)$$

در یک لایه کاهش‌دهنده بعد فاکتور کاهش بعد $n * m$ در نظر گرفته‌شده و ناحیه دریافت ورودی به ماتریس‌های $n * m$ مجاور تقسیم‌شده و با جمع المان‌های ماتریس و اعمال یک وزن و یک بایاس آموزش‌پذیر و تابع تانژانت هایپربولیک روی ماتریس مربوطه، خروجی این لایه را خواهیم داشت، به این معنا که هر $n * m$ المان مجاور که بصورت یک ماتریس $n * m$ هستند به یک المان در لایه بعد تبدیل شده و بعد خروجی بر فاکتور مربوطه تقسیم می‌شود. لایه کاهش‌دهنده بعد با کاهش دقت، مقاومت در برابر انتقال، شیفت و خرابی را افزایش می‌دهد. این عملیات را در شکل زیر مشاهده می‌کنید:

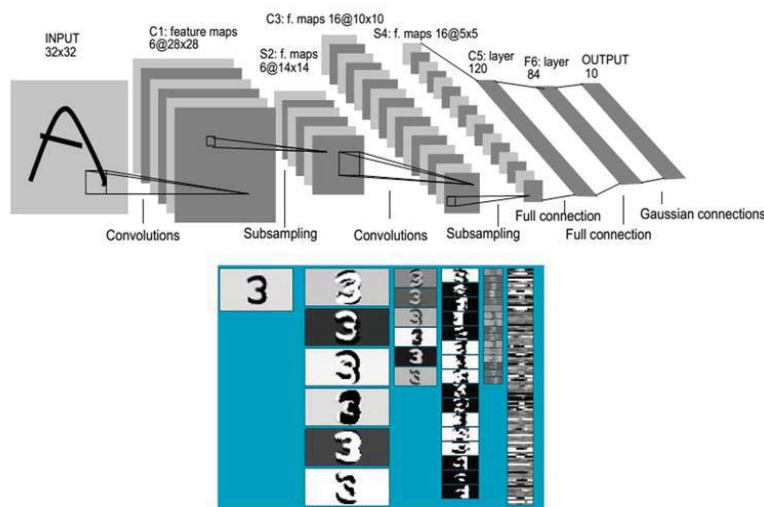


شکل ۲-۸: عملیات کاهش دهنده بعد [۱۳]

تصاویر ویژگی در لایه‌های کاهش بعد نیز مطابق رابطه‌ی زیر می‌باشند [۱۳]:

$$S_{k,l}^i = g(I \downarrow_{k,l}^i W_{k,l} + Eb_{k,l}) \quad (۱۴-۲)$$

g برابر تابع فعال‌سازی، B و b برابر بایاس‌ها، W و w برابر وزن‌ها، $I_{k,l}^i$ برابر i -امین ورودی، $I \downarrow_{k,l}^i$ کاهش بعد یافته i -امین ورودی تصویر ویژگی k از لایه I ، \otimes نمایش دهنده عملیات کانولوشن دو بعدی و E ماتریسی است که تمام عناصر آن یک در نظر گرفته شده است. توجه شود که حروف بزرگ نشان‌دهنده ماتریس‌ها و حروف کوچک نشان‌دهنده یک عدد می‌باشند. ترکیبی از این دو لایه را در شکل زیر مشاهده می‌کنید:



شکل ۲-۹: ترکیب لایه‌ی کانولوشن و کاهش دهنده بعد [۱۳]

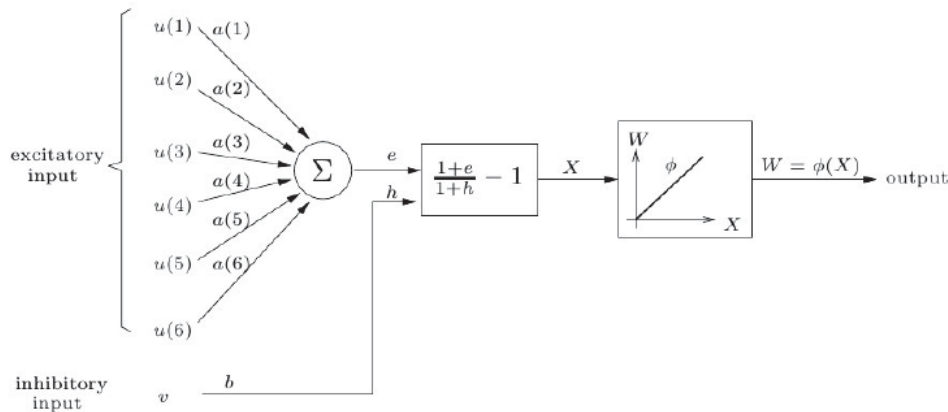
اشتراک وزن در این شبکه‌ها به این معناست که نرون‌های موجود در تصاویر ویژگی یکسان ورودی از محل‌های مختلف تصویر ورودی دریافت می‌کنند به صورتی که کل فضای ورودی را می‌پوشانند، درحالی‌که وزن‌های یکسانی را در نظر می‌گیرند. هر چند تصاویر ویژگی مختلف در یک لایه وزن‌های متفاوت دارند. اشتراک وزن دو مزیت دارد: اول این که به دلیل این که هر نرون فقط با مجموعه محدودی از اطلاعات کار می‌کند مشکل افزایش بعد نخواهیم داشت و دوم این که هر نرون به عنوان یک استخراج گر ویژگی عمل می‌کند که ویژگی‌های بصری اصلی مشابهی را از نقاط مختلف تصویر ورودی استخراج می‌کند.

۲-۵-۲- مدل‌های اولیه شبکه‌های عصبی کانولوشن

■ مدل Neocognitron

فوکوشیما [۱۴] در مطالعاتی که بر روی تشخیص دست خط انجام داد مدلی چند لایه از شبکه عصبی با نام NeoCognitron را ارائه داد. در NeoCognitron از ناحیه دریافت استفاده می‌شد که در آن هر نرون تنها به یک زیر ناحیه مطابق با تعداد مشخصی از نرون‌های مجاور در لایه قبل متصل بود. در مدل NeoCognitron با الگوبرداری از اکتشافی که از سیستم بینایی یک گربه به دست آمد، ایده وجود اتصال‌های محلی شکل گرفت. در حقیقت در این مدل دلیل استفاده از ویژگی اتصال‌های محلی، که قبلاً در شبکه‌های عصبی مورد استفاده قرار گرفته بود، تشخیص ویژگی‌های قابل‌رؤیت پایه‌ای در تصاویر، مانند لبه‌ها، نقاط پایانی و یا گوشه‌ها بیان می‌شود. از آنجایی که این ویژگی‌ها در قسمت‌های گوناگونی از ناحیه ورودی وجود دارند، لذا وزن‌ها نیز باید برای تمامی این قسمت‌ها از ناحیه ورودی یکسان باشند که این قانون اشتراک وزن نام دارد.

در مدل NeoCognitron نرون‌ها به ناحیه دریافت از مرحله قبل متصل هستند. خروجی نرون‌هایی که وزن‌های یکسانی دارند تحت عنوان S-cell نام‌گذاری می‌شوند که همگی یک S-plane را تشکیل می‌دهند. شکل زیر مدل S-cell را نشان می‌دهد. مجموعه‌ای از این S-plane ها یک S-layer را تشکیل می‌دهند. هر S-plane به دنبال خود یک C-plane دارد که وظیفه کاهش دقت S-plane مربوطه را بر اساس یک فاکتور ثابت بر عهده دارد. به عنوان مثال اگر این فاکتور دو باشد C-plane کار کاهش بعد یا محو کردن را انجام می‌دهد. این کار باعث می‌شود شبکه عصبی حساسیت کمتری به خرابی‌ها و جابجایی‌های کوچک در الگوی ورودی داشته باشد.



شکل ۲-۱۰: مدل S-cell در NeoCognitron [۱۴]

نوع سلول‌ها در لایه S از نوع بازدارنده موازی ۱ است. در شکل (۱۰-۱) ورودی‌هایی که از ناحیه دریافت می‌آیند، ورودی‌های تحریک‌کننده نام دارند. یک ورودی بازدارنده نیز که تأثیری عکس در فعال شدن نرون دارد به سلول وارد می‌شود. در صورتی که ما از e بزرگ‌تر باشد خروجی نرون صفر است که این قانون توسط تابعی اعمال می‌شود. مقدار ورودی تحریک‌کننده بصورت رابطه (۱۵-۱) محاسبه می‌شود:

$$e = \sum_{i=1}^N a(i)u(i) \quad (۱۵-۲)$$

که $a(i)$ وزن‌هایی آموزش پذیرند، $u(i)$ نشان‌دهنده ورودی‌های لایه قبل و N نیز تعداد وزن‌ها را نشان می‌دهد. ورودی بازدارنده نیز بصورت رابطه (۱۶-۱) محاسبه می‌شود:

$$h = bv \quad (۱۶-۲)$$

که b وزنی آموزش پذیر است و برای هر صفحه از لایه متفاوت است. ورودی v در NeoCognitron از ناحیه دریافت می‌آید و به صورت میانگین مربعات ریشه وزن دار محاسبه می‌شود که بیانگر نوعی نرمال سازی است. در نهایت فرمول فعال سازی یک S-cell بصورت رابطه (۱۷-۲) است:

$$u_s(i) = \phi \left(\frac{1+e}{1+h} - 1 \right) \quad (۱۷-۲)$$

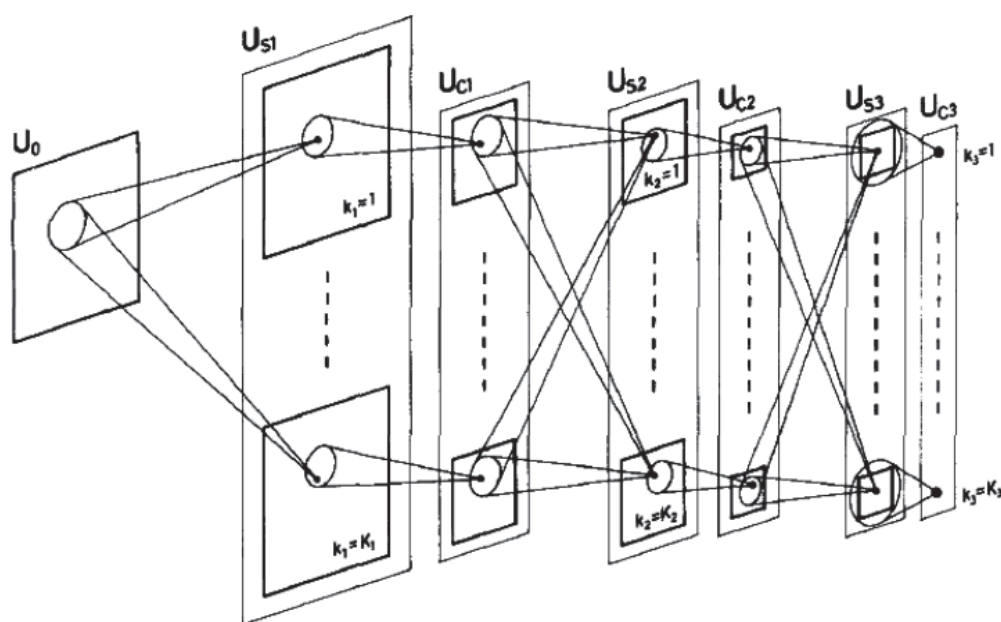
که در آن تابع فعال‌سازی ϕ به صورت زیر تعریف می‌شود و با توجه به ورودی‌اش که در رابطه (۲-۱۷) مطرح شد باعث می‌شود که اگر ورودی بازدارنده از ورودی تحریک‌کننده بیشتر باشد خروجی

صفر شود:

$$\phi(x) = \begin{cases} x, & x \geq 0 \\ 0 & \text{else} \end{cases}$$

فوکوشیما نشان داد که خروجی S-layer ها تخمینی از کانولوشن نرمال شده، بردار وزن و بردار ورودی است. چندین S-plane ها که هر کدام مجموعه متفاوتی از وزن‌ها را دارند، می‌توانند ویژگی‌های متفاوتی را از محلی یکسان استخراج کنند. با استفاده از S-layer ها و C-layer ها در جاهای مختلف و ترکیب نگاشت‌های خروجی آن‌ها می‌توان ماشینی پیچیده برای استخراج ویژگی‌ها ساخت. ارتباط لایه C با لایه k توسط یکسری سیناپس با وزن‌هایی ثابت و غیرقابل تغییر است. تعداد صفحات C با تعداد صفحات و در لایه قبل آن برابر است و این صفحات از دو لایه بصورت نظیر به نظیر به یکدیگر متصل هستند.

ساختار NeoCognitron پایه در شکل (۲-۱۱) نمایش داده شده است. این ساختار دارای یک لایه ورودی و سه لایه تناوبی S-layer و C-layer با نام‌های $U_{S1}, U_{C2}, U_{S1}, U_{C3}, \dots, U_{C3}$ است، که U_{C3} به عنوان لایه خروجی، شامل نرون‌هایی است که تنها بیانگر نتیجه عملکرد شبکه عصبی می‌باشد. توجه کنید که ورودی‌های بازدارنده ۷ در شکل (۲-۱۱) نشان داده نشده‌اند.



شکل ۲-۱۱: اتصال لایه‌ها در NeoCognitron پایه [۱۴]

در مدل NeoCognitron آموزش می‌تواند هم به صورت بدون مربی و هم با مربی صورت گیرد.

اولین پیاده‌سازی از این مدل بصورت بدون مربی و بر پایه‌ی الگوریتم خود سازمانده^۱ و یک قانون یادگیری تقویتی، آموزش داده شده بود. بعدها نشان داده شد که قدرت اصلی مدل NeoCognitron به وسیله آموزش با مربی و بهتر از الگوریتم SOA قابل دستیابی است. آموزش باید بصورت لایه به لایه صورت پذیرد. پارامترهای آموزش پذیر وزن‌های a و b هستند و وزن‌های C -plane ها نیز در طول مرحله آموزش تغییری پیدا نکرده و از ابتدا ثابت هستند.

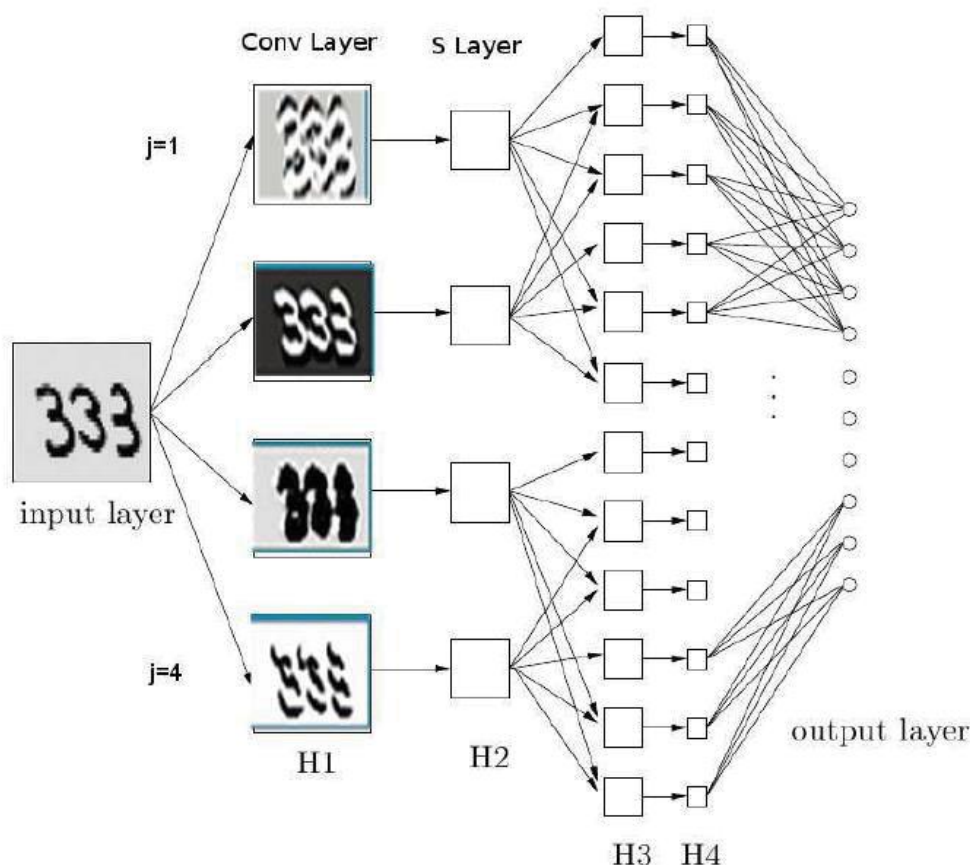
□ مدل LeCun

LeCun [۱۵] برای اولین بار مدلی از شبکه‌های عصبی کانولوشن را پیشنهاد کرد که قابلیت آموزش از طریق پس انتشار را داشت و آن را بر روی مسئله تشخیص ارقام دست‌نویس بکار برد. استفاده گسترده از روش پس انتشار خطا یک انشعاب مهم در بحث شبکه‌های عصبی کانولوشن ایجاد کرد. مدل LeCun و مدل NeoCognitron از نظرهای گوناگون به هم شباهت دارند. مدل LeCun دنباله‌ای از لایه‌های کانولوشن و کاهش‌دهنده بعد است که متناسب با S -layer ها و C -layer ها هستند و شامل تصاویر ویژگی می‌شوند که متناسب با S -plane ها و C -plane های مدل NeoCognitron است. با توجه به این که مدل یک نرون، پرسپترون پایه همراه با یک تابع فعال ساز sigmoid است، هیچ ورودی بازدارنده و در نتیجه هیچ V -plane را شامل نمی‌شود. این موضوع باعث ساده‌تر شدن کلی مدل و ساختار آن می‌شود. مزیت دیگر این مدل این است که دیگر نیاز به آموزش لایه به لایه نیست، بلکه تمام نرون‌ها را می‌توان بصورت مرحله به مرحله و توسط پس انتشار که خطای کلی تابع را کمینه می‌کند، آموزش داد. نتیجه این قابلیت این است که ویژگی‌هایی که باید استخراج شوند، مانند گوشه‌ها و ... بصورت خودکار و طی فرایند آموزش ماسک‌های کانولوشن پیدا شده و به وسیله وزن‌های تصاویر ویژگی مربوطه شکل می‌گیرند. ساختاری از شبکه عصبی کانولوشن که توسط LeCun ارائه شد بعدها LeNet-1 نام گرفته شد. LeNet-4 نسخه‌ای گسترش‌یافته از مدل LeNet-1 است و شامل تصاویر ویژگی و لایه‌ای مخفی علاوه بر مدل قبلی است. با توجه به این که مدل LeNet-4 مدلی کارا است، انواع دیگر ترکیبات آن از جمله Boosted LeNet-4، LeNet-4 با K نزدیک‌ترین همسایه و ترکیب آن با آموزش محلی [۱۶] نیز بررسی شده است. از توسعه‌های دیگر این شبکه‌ها می‌توان LeNet-5 و LeNet-7 را نام برد که در ادامه به بررسی این مدل‌های پایه می‌پردازیم.

¹ Self Organization Algorithm(SOA)

LeNet-1

شکل (۱-۱۱) ساختار از شبکه عصبی کانولوشن که توسط LeCun ارائه شد و بعدها LeNet-1 نام گرفته شد را نشان می‌دهد. لایه ورودی اندازه 28828 را داراست و تصویری خاکستری از رقمی را که باید تشخیص دهد، دریافت می‌کند. شدت نور پیکسل‌ها بین ۱- و ۱+ نرمال می‌شوند. لایه مخفی اول شامل 4 تصویر ویژگی می‌شود که هر کدام دارای 25 وزن هستند، دارای یک کرنل 575 و یک بایاس نیز هست. به علت تأثیر حاشیه‌ها در عملیات کانولوشن، تصویر ویژگی نتیجه کوچک‌تر و برابر 24824 است.



شکل ۲-۱۲: ساختار شبکه عصبی ارائه‌شده توسط LeCun برای تشخیص دست خط [۱۵]

لایه کاهش‌دهنده بعد دارای اندازه 12×12 است. لایه‌های $3H$ و $4H$ هر یک به ترتیب شامل ۱۲ تصویر کانولوشن و کاهش‌دهنده با ابعاد 878 و 464 می‌باشند. تابعی که در این لایه‌ها اعمال می‌شود. دقیقاً مانند همان تابعی است که در لایه‌های $1H$ و $2H$ اجرا می‌شود، با این تفاوت که تصویر ویژگی لایه $3H$ با کانولوشن 363 به جای 565 تولید می‌شود. در نهایت، لایه خروجی شامل مجموعه‌ای از ۱۰ نرون است که بصورت کامل به تصاویر کاهش‌دهنده قبلی از لایه $4H$ متصل شده‌اند و بیانگر ۱۰ رقمی هستند که باید تشخیص داده شوند. برای نرون برنده مقدار پاسخ ۱+ و برای سایر

نرونها مقدار پاسخ را برابر ۱- در نظر می‌گیریم. در مجموع شبکه دارای 4635 عنصر و 98442 اتصال است که به علت وجود خاصیت وزن‌های مشترک تنها 2578 پارامتر مستقل برای آموزش وجود دارد.

LeNet-4

در مورد LeNet-4 که نسخه‌ای گسترش‌یافته از مدل LeNet-1 است و شامل تصاویر ویژگی و لایه‌ای مخفی علاوه بر مدل قبلی است، نمونه‌ای [۱۶] توسعه داده شد که نیاز به 260000 جمع و ضرب داشت و دارای 17000 پارامتر آزاد بود. این نمونه خطای ۱/۱ درصد را برای آزمایش با پایگاه داده MNIST نتیجه داد.

نمونه‌ای دیگر [۱۶] با نام Boosted LeNet-4 نیز بررسی شده است. این مدل برگرفته از روشی با نام تقویت شونده است که قبلاً توسط شاپایر^۱ ارائه شد و توسط دراکر^۲ و دیگران توسعه یافت. ایده اصلی بدین صورت بود که اگر مجموعه و یا اجتماعی از ماشین‌های یادگیر داشته باشیم، به صورتی که آن‌ها به کمک هم مسائلی را حل کنند نتایج با دقت بهتری به دست می‌آید. در نمونه‌ای که در این آزمایش توسط لئون بوتو^۳ و دیگران بررسی شده سه ماشین وجود دارد. اولین آن‌ها LeNet-4 است و ۱٪ خطا دارد. خروجی این ماشین به ماشین دوم داده می‌شود و خروجی این دو ماشین هر دو به ماشین سوم که آن هم LeNet-4 است وارد می‌شود. در این نمونه هم خطایی که برای پایگاه داده MNIST ارائه شد، ۰/۷ درصد است.

با توجه به این که مدل LeNet-4 مدلی کاراست، انواع دیگر ترکیبات آن نیز بررسی شده است. نمونه بعدی LeNet-4 با نزدیک‌ترین هم سایه است. در این مقاله شبکه LeNet-4 چنان تنظیم شد که در خروجی آن 50 مقدار حاضر شود و در نتیجه یک بردار ویژگی مناسب برای جستجو در فضای اقلیدسی به دست آید. در نهایت مقدار خطای ۱/۱ درصد برای این روش بیان شد. نمونه بعدی از LeNet-4 نیز ترکیب آن با آموزش محلی است. با توجه به مواردی که در آن‌ها داده‌های آموزشی موجود در فضای ورودی کمیاب هستند، آموزش محلی به دنبال ایجاد نوعی مطابقت در

¹ schapire

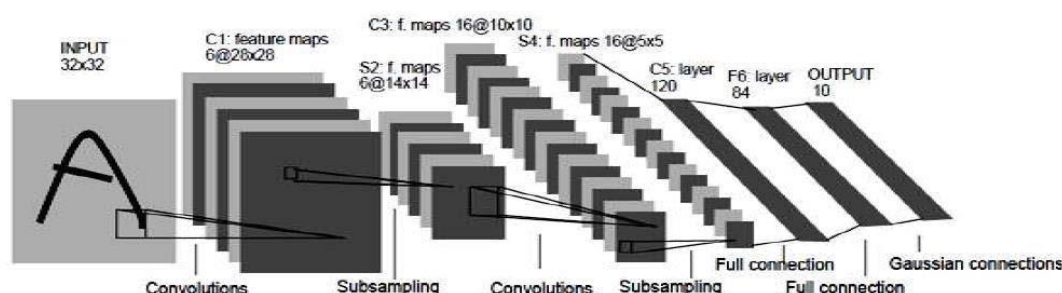
² Drucker

³ Leon Bottou

سیستم آموزش پذیر است تا مشکلاتی نظیر بیش آموزش رخ ندهد. در این روش از روشی برای وزن دهی به نمونه‌های آموزشی استفاده می‌شود. آموزش محلی تنها از K تا از نزدیک‌ترین الگوها به الگوی آزمایشی استفاده می‌کند. آن‌ها با بیان این که شبکه عصبی کانولوشن در برخورد با الگوهای نادر ضعیف عمل می‌کند، این مشکل را ناشی از ظرفیت شبکه تفسیر کردند. آن‌ها حدس زدند که مدل ظرفیت شبکه، زمانی که الگوهای کمیابی در فضای ورودی است بالا و زمانی که الگوهای بسیار زیادی در فضای ورودی است کم است. برای کاستن از این مشکل آن‌ها تصمیم به آموزش یک دسته‌بندی کننده خطی بر روی بردار ویژگی تولید شده توسط لایه ماقبل آخر از LeNet-4 گرفتند. در نتیجه با آزمایش آن مدل بر روی MNIST خطای $1/1$ درصد حاصل شد.

LeNet-5

در سال ۱۹۹۸، LeCun روش‌های مختلف بر پایه گرادیان را روی حروف دست‌نویس مورد بررسی قرار داده است و شبکه عصبی کانولوشن LeNet-5 که در شکل (۱۲-۱) نمایش داده شده است را ارائه داد و نشان داد که شبکه‌های عصبی کانولوشن کارایی بالاتری نسبت به بقیه روش‌های بر پایه گرادیان دارند. شبکه LeNet-5 از ۸ لایه که شامل یک لایه ورودی، سه لایه کانولوشن، دو لایه کاهش بعد، یک لایه کاملاً متصل و یک لایه خروجی از نوع RBF است تشکیل شده است. سایز لایه ورودی است. اولین لایه کانولوشن ۶ تصویر ویژگی با وضوح ۲۸۷۲۸ دارد. دومین لایه، یا اولین لایه کاهش بعد شامل ۶ تصویر ویژگی با سایز ۱۴۶۱۴ است. سومین لایه یک لایه کانولوشن دیگر است و تصویر ویژگی با سایز ۱۰۷۱۰ دارد. چهارمین لایه، یک لایه کاهش بعد است که شامل ۱۶ تصویر ویژگی، هر کدام با سایز 5×5 و پنجمین لایه، یک لایه کانولوشن با ۱۲۰ تصویر ویژگی که کاملاً متصل با لایه قبلی است، هست.



شکل ۲-۱۳: شبکه عصبی کانولوشن LeNet-5 [۱۵]

اتصالات بین تصاویر ویژگی لایه‌های کاهش بعد دوم و کانولوشن سوم در شکل (۱۳-۱) نمایش داده شده است، بدین صورت که نتیجه ترکیب نتایج حاصل از کانولوشن تصاویر ویژگی به چه صورت

بوده و در چه تصاویر ویژگی از لایه بعدی قرار خواهند گرفت. علاوه بر این، لایه‌ها بصورت کامل به هر یک از تصاویر کاهش داده‌شده خود متصل هستند.

جدول ۲-۱: ارتباط داخلی مربوط به لایه‌های کاهش بعد دوم و کانولوشن سوم [۱۵]

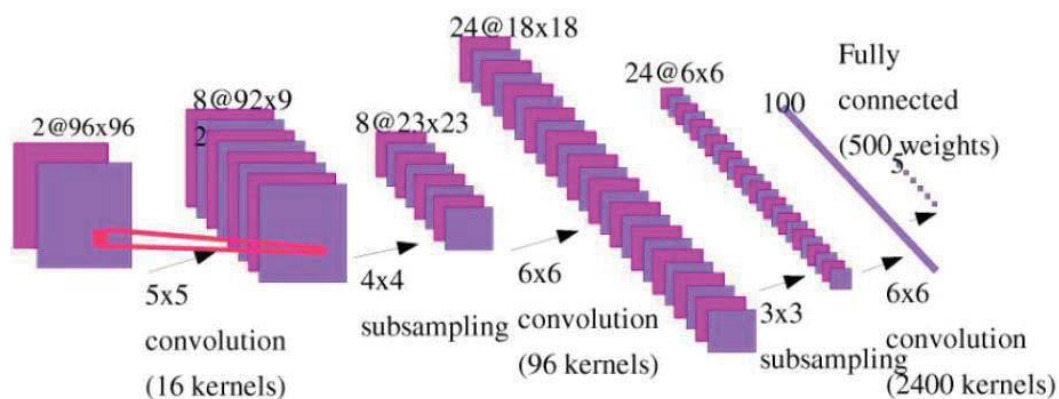
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

دو دلیل برای عدم اتصال کامل تصاویر از لایه‌های کاهش‌دهنده به کانولوشن ذکر شده است، اول این که این کار باعث کاهش تعداد اتصالات به اندازه قابل قبول شده است و دلیل دوم که مهم تر است این که تقارن در شبکه رخ ندهد، زیرا که تصاویر مختلف ویژگی برای استخراج ویژگی‌های متفاوت بوده، لذا باید ورودی‌های مختلفی نیز داشته باشد.

لایه F6 یک لایه مخفی همانند مدل های معمول شبکه عصبی است که به طور کامل به لایه 5C متصل شده و از 84 واحد تشکیل شده است و پس از محاسبه جمع وزن دار ورودی‌ها، آن‌ها را از یک تابع فعال ساز که تابع تانژانت هایپربولیک است می‌گذرانند. نهایتاً، آخرین لایه شامل 10 نرون توابع بر پایه شعاعی است [۱۵]. این شبکه نیز روی پایگاه داده MNIST با خطایی حدود 95 درصد شناسایی را انجام می‌دهد. طبق مقایسه‌ای که انجام شده روش Boosted LeCun-4 دارای بالاترین دقت شناسایی است اما به دلیل محاسبات بسیار زیاد روش Boosted LeNet-4 روش LeNet-5 به عنوان بهترین شبکه در میان این شبکه‌ها در نظر گرفته شده است.

LeNet-7

ساختار این شبکه در شکل (۱-۱۴) نمایش داده شده است. این شبکه 90857 پارامتر آموزش پذیر و 466 میلیون اتصال دارد. این شبکه روی پایگاه داده NORB تست شده و خطایی برابر 0/6 درصد حاصل شده است [۱۵].



شکل ۲-۱۴: ساختار LeNet-7 [۱۵]

۲-۶- جمع‌بندی و خلاصه

در این بخش سعی شد به مفاهیم و ادبیات مربوط به تطبیق استریو چه در روش‌های سنتی و چه روش‌های مدرن پرداخته شود. همچنین سعی شد تا حدودی به شرح شبکه‌های عصبی علی‌الخصوص شبکه‌های عصبی کانولوشن و ویژگی‌ها و نمونه‌های مختلف آن پرداخته شود.

فصل سوم: مروری بر تحقیقات انجام شده

۳-۱- مقدمه

چالش‌های مهم در مبحث تطبیق استریو در سال‌های اخیر عبارت‌اند از نرخ درستی^۱، متوسط زمان پاسخ‌گویی^۲، بهینگی در مصرف انرژی و سخت‌افزار به دلیل استفاده از این فرایند در سیستم‌های نهفته و بلادرنگ^۳ و گوشی‌های تلفن [۱۷].

اما یکی از اساسی‌ترین و مهم‌ترین چالش‌های این فرایند، پیدا کردن مجموعه داده مناسب و منطبق با نیاز جهت انجام فرایند یادگیری در شبکه‌های عصبی به‌کاررفته در این زمینه هست. البته در سال‌های اخیر مجموعه داده‌های مناسبی ایجاد شده است نظیر مجموعه داده‌های MiddleBury[18] که توسط کالج میدلبری ارائه شده است و یا KITTI[19] که در دو ورژن^۴ ۲۰۱۲ (سیاه‌وسفید) و ۲۰۱۵ (رنگی) تهیه شده‌اند، اما دو مشکل اساسی وجود دارد اول اینکه این مجموعه داده‌ها بسیار حجیم بوده و مستلزم به‌کارگیری زمان زیاد و سخت‌افزارهای مناسب می‌باشند و دیگر اینکه این مجموعه داده‌های عام^۵ هستند و برای مصارف خاص منظوره مناسب نمی‌باشند [17]. از این رو تحقیقات انجام‌گرفته در سال‌های اخیر به سه دسته‌ی کلی روش‌های بدون یادگیری، یادگیری با ناظر^۶ و یادگیری بدون ناظر^۷ تقسیم می‌شوند.

۳-۲- روش‌های بدون یادگیری عمیق

در روشی که در سال ۱۳۹۲ توسط دانشجویان دانشگاه خواجه نصیر ارائه شده است [۲۰]، برای تشخیص اشیاء تصاویر فرض می‌شود که پیکسل‌های مشابه و همسایه نمایانگر شیء خاصی هستند و هزینه‌ی تطبیق بر اساس تعداد پیکسل‌های تطبیق داده شده، محاسبه می‌شود. در این تحقیق برای بهبود نقشه اختلاف اولیه از روش بهبود ناحیه‌ی تقاطعی استفاده شده و در انتها با اعمال فیلتر میانه دو بعدی مقادیر جابجایی نابجا حذف می‌شوند.

¹ Accuracy Rate

² Response Time

³ Embedded Systems

⁴ Version

⁵ General

⁶ Supervisor Learning

⁷ Unsupervisor Learning

در روشی دیگر [۲۱] با این نگاه که همبستگی پیکسل‌های تصاویر در دو بعد است، از اتوماتای سلولی دوبعدی برای بهبود دادن نقشه‌ی اختلاف استفاده شده است. در این تحقیق از معیارهای پیکسل‌های غیر مسدود^۱، همه‌ی پیکسل‌ها^۲ و پیکسل‌هایی که نزدیک به ناپیوستگی اختلاف مکان^۳ هستند، به عنوان معیار سنجش کیفیت مدل استفاده می‌شود. در تحقیق ارائه شده در دومین کنفرانس بازشناسی الگو و تحلیل تصویر [۲۲] از مشخصه‌های گوشه‌ها به منظور شناسایی اشیاء و تطبیق استریو استفاده شده است. همچنین در این روش برای محاسبه شباهت از روش محاسبه‌ی خطای مجموع قدر مطلق تفاضل‌ها استفاده شده است. دانشجویان دانشگاه صنعتی شاهرود در سال ۱۳۹۲ [۲۳] برای تشخیص اشیاء و پیکسل‌های متناظر در آن‌ها از مؤلفه‌های رنگ و بافت اشیاء استفاده کرده‌اند.

در روشی ابتکاری و بر اساس الگوی بینایی انسان که توسط جناب آقای فرامرزی نجفی [۲۴] ارائه شده است، تطبیق استریو در دو مرحله صورت می‌پذیرد. در مرحله‌ی اول، ابتدا تخمینی از عمق اشیاء با در نظر گرفتن تاری در تصاویر صورت می‌پذیرد و یک تصویر زمخت از نقشه اختلاف ایجاد می‌شود. سپس با استفاده از الگوریتم‌های متداول تطبیق استریو این تصویر بهبود داده می‌شود. مزیت این روش، کاهش هزینه‌ی تطبیق با استفاده از روش‌های کم هزینه نظیر تشخیص عمق با در نظر گرفتن تاری اشیاء هست. در روش ارائه‌شده در دومین کنفرانس ملی مهندسی برق [۲۵]، از ترکیب روش مبتنی بر ویژگی و مدل مبتنی بر سطح استفاده شده است. در این روش، از معیار شباهت مجموع فاصله همینگ برای افزایش دقت استفاده شده است. همچنین برای افزایش سرعت از روش‌های مبتنی بر برنامه‌نویسی پویا استفاده شده است.

۳-۳- یادگیری با ناظر

یک مدل معماری شبکه به نام Siamese Net [26] در سال ۲۰۱۵ توسط آقای لی کان^۴ معرفی شده است که برای پوشش هدف میزان درستی ابتدا در دو شبکه‌ی کانولوشن جداگانه اما با وزن‌های

¹ nonocc

² all

³ disc

⁴ Yann Lecunn

مشترک^۱، ویژگی‌های مورد نیاز تصاویر را استخراج می‌کند و سپس در یک لایه‌ی اتصال^۲ آن دو را به هم متصل می‌کند و با به‌کارگیری چندین لایه اتصال کامل^۳ نقاط مشابه را استخراج می‌کند. اگرچه این مدل از نظر درستی دقت بسیار بالای ۹۷ درصد را برای مجموعه داده‌های KITTI داشته است اما به دلیل استفاده از لایه‌های متعدد کانولوشن در زیر شبکه‌ی اتصال خود، بار پردازشی بالایی را تحمیل می‌کرده است که موجب کندی و توان مصرفی بالا در آن می‌شده است. برای کاهش زمان پاسخگویی، محققین دانشگاه تورنتو در سال ۲۰۱۵ بجای استفاده از لایه‌های کانولوشن پر هزینه در زیر شبکه‌ی اتصال خود، از یک لایه‌ی ضرب داخلی ساده استفاده کردند [۲۷]، در این مدل اگرچه در بهترین حالت میزان درستی ۴ الی ۵ درصد کاهش داشت اما تأثیر بسیار زیادی در زمان پردازش و توان مصرفی شبکه داشت. محققین دانشگاه نیویورک در سال ۲۰۱۶ [۲۸] جهت افزایش سرعت و درستی روش بکار گرفته شده در روش ارایه شده توسط لی‌کان، این مدل را به شکلی تغییر دادند که بجای انجام مراحل استخراج ویژگی‌ها و ضرب داخلی بروی تمام تصویر، این روند بروی برید کوچک تر از تصویر انجام گیرند. از مزیت‌های این روش می‌شود به یکسان‌سازی در تصاویر ورودی اشاره کرد.

در تحقیقاتی دیگر [۲۸-۳۰] برای افزایش سرعت مدل Siamese جای استفاده از روش‌های تطبیق سنتی از روش جدید تری استفاده کردند که در آن تطبیق به صورت زیر پیکسلی^۴ استفاده می‌شود که البته این روش نیازمند به استفاده از کارت‌های گرافیک^۵های قوی برای رسیدن به سرعت موثر خود می‌باشند. برای بررسی تطبیق از روش‌های موثر دیگری نیز استفاده شده است، به طور مثال در سال ۲۰۱۷ در روشی برای این منظور از اطلاعات محیطی و استخراج اطلاعات معنایی استفاده شده است [۳۱].

در کنار معماری Siamese، مدل‌های دیگری نظیر مدل هرمی^۶ [۳۲] معرفی شده‌اند. مدل هرمی روشی است که به صورت آبشاری ابتدا ویژگی‌های تصویر را در سطح پیکسل استخراج کرده و به سطح بعد که در گستره‌ی بزرگ‌تری است تحویل می‌دهد و این سلسله‌مراتب تا سطح استخراج ویژگی‌های منطقه‌ای^۷ پیش می‌رود. این روش به منظور افزایش نرخ درستی در مرحله‌ی محاسبه‌ی هزینه‌ی

¹ Weight Sharing

² Concatenate

³ Dense Layer

⁴ Subpixel

⁵ GPU

⁶ Pyramid Architecture

⁷ Region

تطبیق^۱ مورد استفاده قرار می‌گیرد.

از مسائلی که می‌تواند در سرعت رسیدن به نقشه‌ی اختلاف موثر واقع شود، انتخاب نقشه‌ی تصادفی اولیه مناسب جهت انجام فرایند یادگیری تکرار شونده^۲ هست. از این رو در برخی تحقیقات از فیلد تصادفی شرطی^۳ برای این منظور استفاده کرده‌اند [۳۳، ۳۴] و در سال ۲۰۱۸، در روشی از ترکیب مفهوم دید زمخت به مناسب^۴ و روش‌های جستجوی تصادفی موثر برای انتخاب تصویر اولیه استفاده شده است [۳۵].

در سال‌های اخیر تحقیقاتی در زمینه‌ی تشخیص عمق با بکارگیری تک تصویرها^۵ صورت گرفته است به طوری که یک تصویر دریافت می‌شود و تصویر زاویه‌ی دیگر با استفاده از شبکه‌های عصبی کانولوشن تولید می‌شود [۳۶]. یکی از اساسی‌ترین مشکل‌ها در تطبیق استریو، وجود تصویرهای خراب و دارای پارازیت هست. برای رفع این مشکل از دو زیر شبکه‌ی کانولوشنی استفاده می‌شود، یکی برای ایجاد نقشه‌ی اختلاف برای اشیا بزرگ و دیگری برای بهینه کردن آن با استفاده از مقایسه با تصاویر چرخیده شده [۳۷].

به طور مثال برای محاسبه‌ی هزینه‌ی تطبیق از تکنیک تخمین شباهت و برای جمع هزینه‌ها از تکنیک SPS استفاده می‌شود [۳۸]. به طور کلی اکثریت تحقیقات از جمله موارد بالا بروی مراحل یک و دو هست. اما سال ۲۰۱۸ مدلی با سه زیر شبکه جهت پوشش هر ۴ مرحله ارائه شد [۳۹].

۳-۴- یادگیری بدون ناظر

با توجه به نبود داده‌های یادگیری مناسب و یا گران قیمت بودن آن‌ها برخی تحقیقات در زمینه‌ی بکارگیری تکنیک‌های یادگیری بدون ناظر صورت گرفته است. در تحقیقی که در سال ۲۰۱۷ و تحت عنوان یادگیری بدون ناظر برای تطبیق استریو^۶ [۴۰] صورت گرفته است، برای پیاده‌سازی روش بدون ناظر بجای استفاده از ناظر انسانی، از دو تصویر چپ و راست به عنوان ناظر خود استفاده کرده است. به این شکل که ابتدا یک تصویر رندوم^۷ به عنوان نقشه‌ی اختلاف ایجاد شده و در تکرارهای

¹ Matching Cost

² Iterative

³ Conditional Random Field(CRF)

⁴ Coarse_to_fine

⁵ Monocular Stereo Vision

⁶ Unsupervised learning for Stereo Matching

⁷ Random

متفاوت و با توجه به شباهت رنگ‌ها و ویژگی‌های اشیاء دو تصویر چپ و راست تصویر ایجاد شده را اصلاح می‌کند.

روش یادگیری بدون ناظر همراه با خطای انسداد آگاه^۱ [۴۱]، روشی مبتنی بر مدل محاسبه‌ی خطای خود هست که خطای انسداد آگاه نام دارد که در آن دو از روش محاسبه‌ی خطای صافی^۲ و محاسبه‌ی خطای سازگاری تصاویر چپ و راست^۳ جهت کنترل درستی و یکپارچگی تصویر نقشه‌ی اختلاف استفاده می‌شود. در تحقیقی دیگر مدل DispNet[42] را به شکلی تغییر داده است که به صورت بدون ناظر فعالیت کند این امکان با وزن دهی به خطاهای جمع شده از هر پیش‌بینی میسر شده است [۴۳].

جدول ۳-۱: تفکیک مقالات تطبیق استریو بر اساس روش

روش استریو	منابع
روش‌های بدون یادگیری عمیق	[32-27]
یادگیری با ناظر	[45-32]
یادگیری بدون ناظر	[49-46]

۳-۵- جستجوی معماری شبکه‌های عصبی^۴

جستجوی معماری شبکه‌های عصبی یک فرایند در خودکارسازی یادگیری ماشین است که زیرمجموعه علم یادگیری اتوماتیک^۵ هست. در حال حاضر NAS، در زمینه‌های مختلفی نظیر طبقه‌بندی تصاویر^۶ [۴۴، ۴۵]، تشخیص اشیاء^۷ [۴۶] و یا بخش‌بندی معنایی^۸ [۴۷] تقریباً جایگزین طراحی دستی شده است. NAS را می‌توان زیرشاخه‌ی AutoML دانست که دارای اشتراکاتی با

¹ Unsupervised Stereo Matching with Occlusion-Aware Loss

² Smoothness

³ Left-Right Consistency

⁴ Neural Architecture Search (NAS)

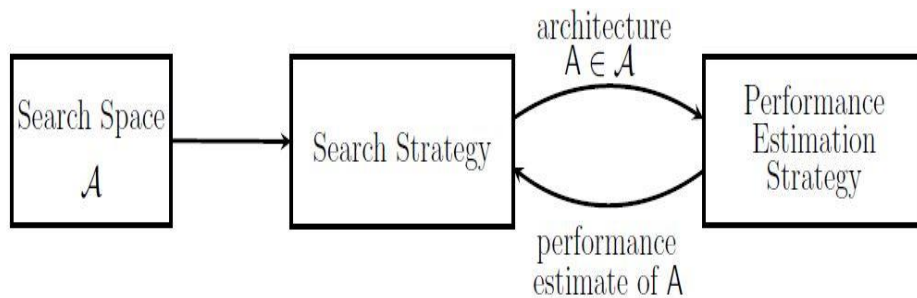
⁵ AutoML

⁶ Image Classification

⁷ Object Detection

⁸ Semantic Segmentation

بهینه‌سازی فراپارمترها^۴ [۴۸] و فرایادگیری^۲ [۴۹] هست. به طور کلی روش‌های NAS در ۳ دسته تقسیم می‌شوند: فضای جستجو^۳، استراتژی جستجو^۴ و استراتژی تخمین کارایی^۵.



شکل ۳-۱: نمایی از روابط میان سه بخش NAS

۳-۵-۱- فضای جستجو

فضای جستجو به تعداد معماری‌های مختلفی که یک الگوریتم می‌تواند ایجاد کند اطلاق می‌شود. شماری از تحقیقات مربوط به NAS در زمینه‌ی کاهش این فضا در راستای بهینگی جستجوها و کاهش زمان متوسط رسیدن به معماری مطلوب هست.

یکی از فضاهاى جستجوی ساده و معمول، مدل شبکه‌های عصبی زنجیره‌ای^۶ هست که در آن شبکه زنجیره‌ای از n لایه است که لایه l_i ورودی خود را از لایه l_{i-1} گرفته و خروجی خود را به لایه l_{i+1} به عنوان ورودی آن لایه می‌دهد. در اینجا، فضای حالت با مشخصه‌های ماکزیمم لایه، انواع عملیاتی که در هر لایه اجرا می‌شوند نظیر کاهندگی^۷، کانولوشن و یا عملیات پیشرفته‌تر نظیر پیچیدگی قابل تفکیک عمق^۸ [۵۰] و کانولوشن گسسته شده^۹ [۵۱] و در انتها فراپارمترهای مربوط به

^۱ Hyperparameter Optimization

^۲ Meta-Learning

^۳ Search Strategy

^۴ Search Strategy

^۵ Performance Estimation Strategy

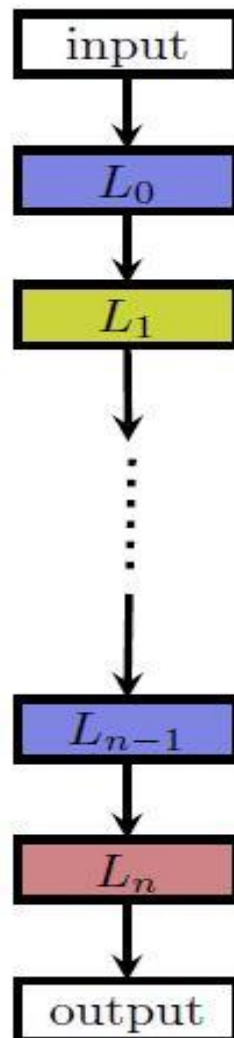
^۶ Chain-Structured Neural Networks

^۷ pooling

^۸ Depthwise Seperable Convolution

^۹ Dilated Convolution

هر لایه نظیر تعداد فیلترها، سائز کرنل، گام‌های^۱ مربوط به لایه‌ی کانولوشن [۵۲-۵۴] و یا تعداد واحدهای لایه‌های تمام اتصال^۳ [۵۵]، مشخص می‌شود.



شکل ۳-۲: نمایی از فضای جستجوی زنجیره‌ای

اما در کارهای اخیر صورت گرفته در NAS، یک المان جدید اضافه شده است به نام اتصال پرش^۴ که این امکان را می‌دهد تا شبکه‌های پیچیده‌تری با چند انشعاب بسازیم [44, 45, 54, 56-58]. در این مدل ورودی لایه‌ی i را می‌توان بصورت $g_i(L_{i-1}^{out}, \dots, L_0^{out})$ که ترکیبی از خروجی چند لایه‌ی قبلی است، نمایش داد. استفاده از این مدل آزادی بیشتری در ساخت مدل‌های مختلف فراهم می‌کند. مدل

¹ Strides

² Units

³ Fully-Connected Layers

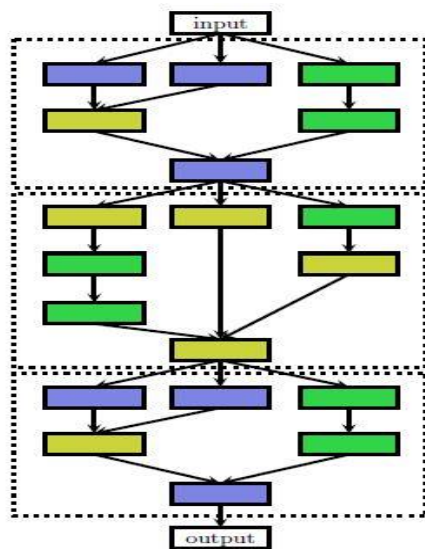
⁴ Skip Connection

های مختلف این فضای جستجو عبارت‌اند از مدل زنجیره‌ای (این مدل زنجیره‌ای با مدل زنجیره‌ای بالا کمی تفاوت دارد اما به صورت کلی آن‌ها را می‌توان از یک جنس قلمداد کرد) که در آن L_{i-1}^{out} برابر است با $g_i(L_{i-1}^{out}, \dots, L_0^{out})$ ، شبکه‌های رسوبی^۱ [۵۹] که در آن خروجی‌های لایه‌های قبلی با یکدیگر جمع می‌شوند:

$$g_i(L_{i-1}^{out}, \dots, L_0^{out}) = L_{i-1}^{out} + L_j^{out}, j < i - 1$$

و شبکه‌های متراکم^۲ [۶۰] که در آن خروجی لایه‌های پیشین به یکدیگر الحاق^۳ می‌شوند:

$$g_i(L_{i-1}^{out}, \dots, L_0^{out}) = \text{concat}(L_{i-1}^{out}, \dots, L_0^{out})$$



شکل ۳-۳: فضای حالت پیچیده‌ی چند شاخه‌ای

۳-۵-۲- استراتژی جستجو

روش‌های متنوعی برای جستجو در معماری‌های شبکه‌های عصبی وجود دارد نظیر جستجوی

¹ Residual Networks

² DenseNets

³ Concatenate

تصادفی^۱، بهینه‌سازی بیزی^۲، روش‌های تکاملی، یادگیری تقویتی^۳ و روش‌های مبتنی بر گرادیان^۴. از روش‌های تکاملی در دهه‌های گذشته نیز برای بهینه‌سازی شبکه‌های عصبی استفاده می‌شد [۶۱-۶۵]. روش بهینه‌سازی بیزی از سال ۲۰۱۳ موفقیت‌های بسیاری در NAS داشت نظیر بهینه‌ترین مدل برای بینایی [۶۶]، بهینه‌ترین مدل برای Cifar-10 بدون تقویت داده‌ها^۵ [۶۷] و اولین پیروزی مدل خودکار در چالش دیتاست در مقابل متخصصین [۵۵]. در روش یادگیری تقویتی نیز NAS توانست برای چالش‌های Cifar-10 و Penn Treebank مدلی با ضریب درستی قابل رقابت عرضه کند [۶۸] که این خود باعث جلب شدن همه‌ی نگاه‌ها به NAS و توسعه‌ی آن شد.

یکی دیگر از استراتژی‌های جستجو، neuro-evolutionary است که در آن از الگوریتم‌های تکاملی در جهت بهینه‌سازی شبکه‌های عصبی استفاده می‌شود. اولین استفاده از این الگوریتم‌ها برای توسعه‌ی شبکه‌های عصبی بر می‌گردد به سه دهه‌ی پیش [۶۹] که در آن از الگوریتم ژنتیک برای بهبود معماری و از روش انتشار به عقب^۶ برای بهینه‌سازی وزن‌ها استفاده شد. پس از آن تحقیق‌های زیادی برای استفاده از الگوریتم ژنتیک هم برای توسعه‌ی معماری و هم برای تنظیم وزن‌ها، انجام گرفت [۶۱، ۶۲، ۶۴] اما به دلیل اینکه یک شبکه عموماً میلیون‌ها وزن دارد، روش بهینه‌سازی وزن کاهش شیب تصادفی^۷ تقریباً در تمام موارد الگوریتم ژنتیک را شکست داده است. به همین دلیل بیشتر تحقیقات اخیر صورت گرفته در neuro-evolutionary از همان روش کاهش شیب تصادفی برای بهینه‌سازی وزن‌ها در کنار ایجاد مدل با الگوریتم ژنتیک استفاده کرده‌اند [۴۵، ۵۳، ۵۸، ۷۰-۷۳]

جدول ۳-۲: تفکیک مقالات NAS بر اساس روش جستجو

روش جستجو	منابع
روش‌های تکاملی	[79-75, 71-67, 64, 59, 51]
روش‌های بهینه‌سازی بیزی	[74-72, 61]

¹ Random Search² Bayesian Optimization³ Reinforcement Learning (LR)⁴ Gradient-Based Methods⁵ Data Augmentation⁶ BackPropagation⁷ Stochastic Gradient Descent

۳-۵-۳- استراتژی تخمین کارایی

منظور از استراتژی تخمین کارایی انتخاب روشی است که بسنجد که آیا مدل ایجاد شده می‌تواند برای داده‌های مشاهده نشده بهترین کارایی را داشته باشد و یا خیر. ساده‌ترین روش برای این کار انجام پروسه یادگیری برای آن مدل و آزمون مدل یادگیری شده است درحالی‌که این روش بسیار پرهزینه بوده و برای اجرای آن به هزاران روز-کارت گرافیک^۱، نیازمند هست [۴۴, ۴۵, ۶۸, ۷۰]. به همین دلیل روش‌های متنوعی برای کاهش این هزینه مورد بررسی قرار گرفته‌اند که در جدول زیر خلاصه‌ای از آن‌ها مشاهده می‌شود.

جدول ۳-۳: خلاصه‌ی روش‌های تخمین کارایی در NAS

منبع	نحوه اجرای روش	روش افزایش سرعت
[۴۴, ۴۵, ۷۴-۷۷]	کاهش مراحل فرایند یادگیری، کاهش حجم داده-ها، کوچک کردن مدل‌ها	تخمین کاهش وفاداری ^۲
[۶۷, ۷۸-۸۰]	در صورتی که پس از چند مرحله یادگیری بتوان منحنی یادگیری را به دست آورد و برای مراحل بعدی یادگیری نتیجه حدس زده شود، تعداد مراحل یادگیری کم‌تر می‌شود.	برون‌یابی منحنی یادگیری ^۳
[۵۴, ۵۷, ۵۸, ۷۰, ۸۱]	بجای اینکه مدلی از پایه یادگیری شود، سعی می‌شود از یادگیری‌های قبلی برای آن استفاده شود.	ارث‌بری وزن
[۸۲-۸۷]	در این روش فقط مدل‌هایی که بار اول است که بکار می‌روند یادگیری می‌شوند و باقی از وزن‌های آن‌ها استفاده می‌کنند.	یادگیری تک‌شات/ اشتراک وزن

¹ GPU-days a metric to measure consuming resources in learning process

² Lower Fidelity Estimate

³ Learning Curve Extrapolation

۳-۶- جمع‌بندی و مقایسه

به صورت کلی تحقیقاتی که در این سال‌ها در مبحث تطبیق استریو صورت گرفته شامل موارد

زیر هست:

- افزایش کارایی الگوریتم‌ها چه از نظر مدیریت بهینه‌ی منابع در سیستم‌های موبایل و کامپیوترهای رو میزی و یا حتی سرورهای پردازش ابری
- افزایش سرعت پردازش برای سیستم‌های بلادرنگ
- کاهش نیاز به داده‌های یادگیری از طریق بکارگیری الگوریتم‌های مبتنی بر آموزش بدون ناظر

- امکان استفاده از تک تصویرها جهت پردازش‌های تصاویر ۳ بعدی

- افزایش دقت در بکارگیری این الگوریتم‌ها

همه‌ی مواردی که ذکر شد مورد بررسی واقع شده و بهبودهای چشمگیری در هر یک از آن‌ها صورت گرفته است به گونه‌ای که به صورت کلی نرخ خطا در این الگوریتم‌ها به حدود ۰,۷ درصد و زمان‌های پاسخ‌گویی به زمانی کمتر از ۵۰ صدم ثانیه رسیده است.

فصل چهارم: روش پیشنهادی برای حل مساله

۴-۱- مقدمه

با توجه به گسترش روزافزون بهره‌گیری از تلفن‌های هوشمند^۱ و اینترنت اشیاء^۲، نیاز به پردازش‌های دقیق اما کم‌هزینه بیشتر احساس می‌شود. شبکه‌های عصبی عمیق به دلیل حجم بالای پردازش‌های ماتریسی یک از پرمصرف‌ترین روندهای محاسباتی هستند که رفته‌رفته نیاز به استفاده از آن‌ها در سیستم‌های ذکر شده، بیشتر احساس می‌شود. از این رو، امروزه تحقیقات گسترده‌ای در زمینه‌ی کاهش حجم پردازش این شبکه‌های در کنار حفظ دقیق محاسباتی آن‌ها صورت می‌پذیرد.

۴-۲- بیان مساله

یکی از راه‌کارهای طراحی شبکه‌های عصبی عمیق کم‌مصرف و دقیق، استفاده از الگوریتم‌های بهینه‌سازی هست. هدف از این تحقیق، بهینه‌سازی یک شبکه‌ی عصبی عمیق در مصرف تطبیق استریو با استفاده از الگوریتم سرد شدن شبیه‌سازی شده هست. از آنجا که نرخ درستی و بهینگی مصرف دو هدف جداگانه می‌باشند، این فرایند بهینه‌سازی به صورت چندهدفه^۳ طراحی خواهد شد.

۴-۳- علت استفاده از این روش

از آن جا که فرایند یادگیری در مدل‌های تطبیق استریو به طور کلی عملیاتی پرهزینه است. در نتیجه، نیاز است که برای بهینه‌سازی آن از روش‌هایی استفاده شود که نه تنها به مدل جهانی مناسب در فضای جستجو برسند بلکه این نتیجه در آن‌ها از کوتاه‌ترین مسیر حاصل شود. در نتیجه به صورت کلی استفاده از روش‌های مبتنی بر الگوریتم ژنتیک نظیر الگوریتم رقابت موازی میان امپراتوری‌ها^۴ [۸۸] مقرون به صرفه نخواهد بود. اما در کنار الگوریتم‌های مبتنی بر ژنتیک روش‌های کم‌هزینه‌تری

¹ Smart Phone

² Internet of Things (IoT)

³ Multi-Objective

⁴ Parallel Imperialists Competition Algorithm (PICA)

نظیر بهینه‌سازی بیزی^۱ [۸۹] و سردشدن کوانتوم^۲ [۹۰] نیز مورد بررسی قرار گرفتند که به دلایل عدم تطبیق مناسب با چندهدفی بودن و پیچیدگی محاسباتی مورد استفاده قرار نگرفتند.

یکی دیگر از روش‌های بهینه‌سازی تکاملی روش تپه نوردی است که اگرچه مناسب‌تر از روش‌های بالا است، اما دارای مشکل گیر افتادگی در نقاط بهینه‌ی محلی هست. برای حل این مشکلات در این مساله از الگوریتم سردشدن شبیه‌سازی شده استفاده شده است که در کنار سرعت و سادگی توانایی چندهدفه شدن را دارد و همچنین احتمال گیر افتادن در یک نقطه‌ی بهینه‌ی محلی در آن بسیار پایین‌تر از روش تپه نوردی است.

۴-۴- شرح روش پیشنهادی

همان‌طور که گفته شد، این روش از ترکیب یک مدل پایه و الگوریتم سردشدن شبیه‌سازی شده به وجود آمده است. در این بخش به معرفی مدل پایه، فضای جستجو و توابع انتقال و محاسبه‌ی انرژی در الگوریتم سردشدن شبیه‌سازی شده خواهیم پرداخت.

۴-۴-۱- مدل پایه

در این تحقیق به دلیل تنوع در روش‌های انطباق استریو و نبود روش مرجع، نیاز است که از یک مدل مناسب به عنوان چارچوب مرجع استفاده شود و از روش سردشدن شبیه‌سازی شده در جهت تغییر قسمتی از آن مدل مرجع استفاده شده و در انتها نتایج به دست آمده با نتایج مدل ذکر شده مقایسه شود تا درستی و کارایی روش استفاده شده مورد تصدیق قرار گیرد.

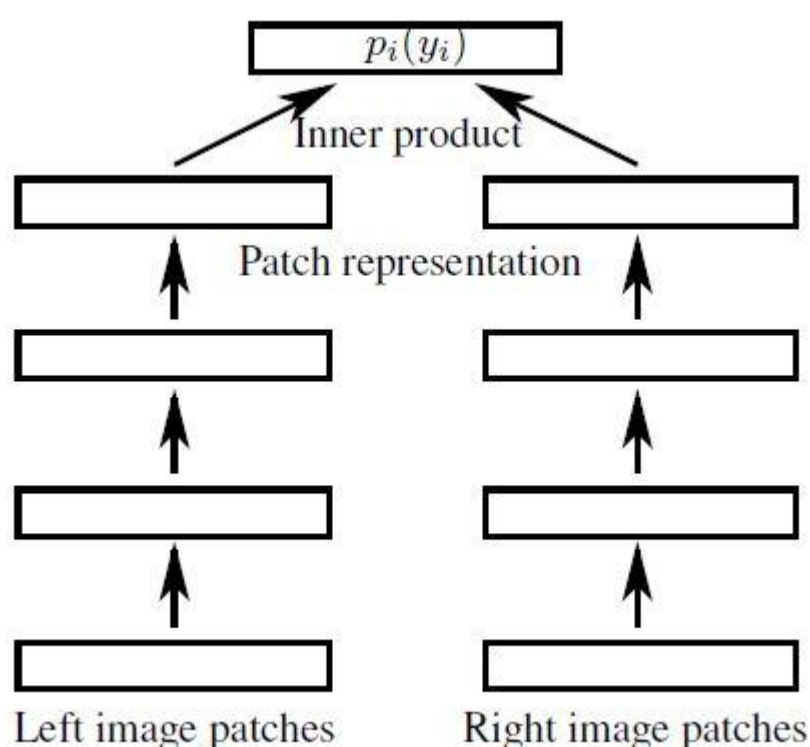
¹ Bayesian Optimization

² Quantum Annealing

برای این منظور در این تحقیق از مدل Content-CNN [۲۷] استفاده شده است. این مدل به دو دلیل زیر مورد استفاده قرار گرفته شده است:

- ارائه‌ی نتایج مورد قبول (۹۰ درصد درستی)
- انعطاف مناسب (توانایی بالا در ایجاد تغییر در قسمت اعظم شبکه بدون نیاز به تغییر ثابت‌های مدل)

این مدل که به صورت مختصر در بخش مروری بر تحقیقات معرفی شده است، از جمله مدل‌هایی است که از معماری Siamese استفاده می‌کند.

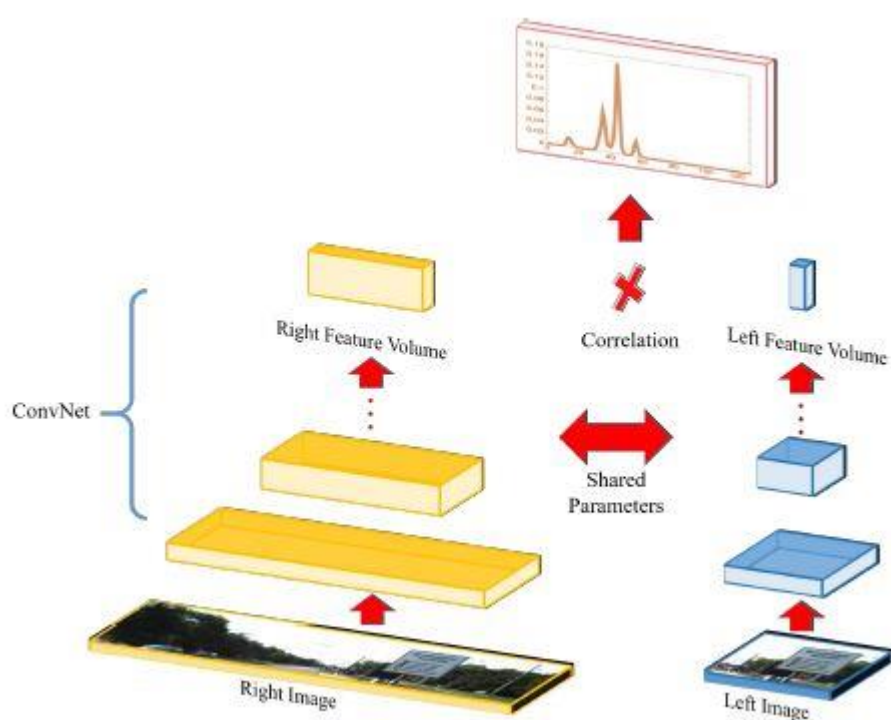


شکل ۴-۱: معماری مدل Content-CNN [27]

همان‌گونه که در شکل (۴-۱) مشاهده می‌شود، در این روش از دو زیر شبکه استفاده می‌شود که وظیفه‌ی پردازش و استخراج ویژگی‌های تصاویر چپ و راست را دارند و در نهایت خروجی این دو زیر شبکه در لایه‌ی آخر باهم ضرب داخلی^۱ شده و نقشه‌ی اختلاف که خروجی نهایی است را می‌دهند.

¹ Inner Product

در این مدل بجای انجام پردازش بر روی تصویر کامل از پاره تصویر^۱ استفاده شده است به این شکل که زیر شبکه‌ی سمت چپ پاره تصویری مربعی شکل می‌پذیرد اما زیر شبکه‌ی راست پاره‌ی تصویری با ارتفاع برابر با ارتفاع سمت چپ و عرض برابر با مجموع عرض سمت چپ و بازه‌ی اختلاف^۲ می‌پذیرد. مطابق شکل (۲-۴)



شکل ۲-۴ : مدل مفهومی شبکه [۲۷]

در این روش مدل‌های ایجاد شده توسط سرد شدن شبیه‌سازی شده در زیر شبکه‌های مربوط به تصاویر چپ و راست قرار می‌گیرند و از آن جا که هدف اصول معماری Siamese بر پایه‌ی اشتراک‌گذاری وزن^۳ها در میان دو زیر شبکه‌ی چپ و راست است، در نتیجه لازم است که دو زیر شبکه کاملاً مشابه یکدیگر باشند.

¹ Patch

² Disparity Range

³ Weight-Sharing

۴-۴-۲- الگوریتم سردشدن شبیه‌سازی شده

در این الگوریتم، حلقه‌ی خارجی فرایند درجه‌ی حرارت را کاهش می‌دهد و حلقه‌ی داخلی روند رسیدن به تعادل در هر درجه حرارت را بیان می‌کند. یعنی تعداد تعویض‌های همسایگی را که باید در یک درجه‌ی حرارت انجام شود، مشخص می‌کند.

۱. جواب اولیه i را از مجموعه جواب‌های امکان‌پذیر انتخاب کنید. $i \in S$

۲. درجه‌ی حرارت اولیه T_0 را انتخاب کنید. $T_0 \geq 0$

۳. فرایند کاهش درجه‌ی حرارت را انتخاب کنید.

۴. تابع مربوط به تعداد تعویض‌ها در هر درجه را انتخاب کنید.

۵. شمارنده مربوط به تغییر درجه حرارت را مساوی ۰ قرار دهید. $t = 0$

۶. حلقه زیر را تکرار کنید. (فرآیند انجماد)

a. شمارنده مربوط به تعداد تعویض‌ها، در هر درجه حرارت را مساوی صفر قرار

دهید. $n = 0$

b. حلقه زیر را تکرار کنید. فرآیند برقراری تعادل (در هر درجه حرارت).

c. جواب j را در همسایگی جواب i ایجاد کنید.

d. مقدار $d_f = f(j) - f(i)$ را محاسبه کنید.

e. اگر $d_f < 0$ ، جواب را بپذیرید. $i = j$

f. در غیر این صورت، عدد تصادفی بین صفر و یک انتخاب کنید. $x \in u(0,1)$

g. اگر $x < e^{\left(\frac{-d_f}{T}\right)}$ باشد، آنگاه $i = j$

h. $n = n + 1$

۷. تابع برقراری رابطه $n = N(t)$ حلقه را تکرار کنید.

۸. $t = t + 1$

۹. درجه حرارت جدید را محاسبه نمایید. $T = T(t)$

۱۰. تا برقراری شرط توقف، حلقه را تکرار کنید.

کیفیت جواب‌های الگوریتم تبرید شبیه‌سازی شده نسبت به پارامترهای کنترلی حساسیت بالایی

دارد و تعیین پارامترهایی که جواب‌های خوب را ایجاد نماید، مشکل است.

۴-۴-۳- فضای جستجو

در این روش فضای جستجو بصورت رشته‌ای^۱ از لایه‌های مختلف است که طول این رشته قابل تغییر بوده و از رشته‌ای به طول ۱۰ کاراکتر شروع می‌شود. در جدول (۴-۳) هر کاراکتر و لایه‌ی معادل آن معرفی شده است.

جدول ۴-۱: کاراکترهای موجود در رشته‌های تولید شده توسط الگوریتم سرد شدن شبیه‌سازی شده و معادل آن‌ها در شبکه‌های عصبی عمیق

کاراکتر	لایه معادل
['None',0,'None',0]	Nothing
['Conv2d',32,'Same',3]	Conv2d(filters=32,padding=Same,kernel=[3× 3])
['Conv2d',64,'Same',3]	Conv2d(filters=64,padding=Same,kernel=[3× 3])
['Conv2d',32,'Same',5]	Conv2d(filters=32,padding=Same,kernel=[5× 5])
['Conv2d',64,'Same',5]	Conv2d(filters=64,padding=Same,kernel=[5× 5])
['Conv2d',32,'Same',7]	Conv2d(filters=32,padding=Same,kernel=[7× 7])
['Conv2d',64,'Same',7]	Conv2d(filters=64,padding=Same,kernel=[7× 7])
['Conv2d',32,'Same',9]	Conv2d(filters=32,padding=Same,kernel=[9× 9])
['Conv2d',64,'Same',9]	Conv2d(filters=64,padding=Same,kernel=[9× 9])
['Conv2d',32,'Same',11]	Conv2d(filters=32,padding=Same,kernel=[11× 11])
['Conv2d',64,'Same',11]	Conv2d(filters=64,padding=Same,kernel=[11× 11])
['Conv2d',32,'Valid',3]	Conv2d(filters=32,padding='Valid',kernel=[3× 3])
['Conv2d',64,'Valid',3]	Conv2d(filters=64,padding='Valid',kernel=[3× 3])
['Conv2d',32,'Valid',5]	Conv2d(filters=32,padding='Valid',kernel=[5× 5])
['Conv2d',64,'Valid',5]	Conv2d(filters=64,padding='Valid',kernel=[5× 5])
['Conv2d',32,'Valid',7]	Conv2d(filters=32,padding='Valid',kernel=[7× 7])
['Conv2d',64,'Valid',7]	Conv2d(filters=64,padding='Valid',kernel=[7× 7])
['Conv2d',32,'Valid',9]	Conv2d(filters=32,padding='Valid',kernel=[9× 9])
['Conv2d',64,'Valid',9]	Conv2d(filters=64,padding='Valid',kernel=[9× 9])
['Conv2d',32,'Valid',11]	Conv2d(filters=32,padding='Valid',kernel=[11× 11])
['Conv2d',64,'Valid',11]	Conv2d(filters=64,padding='Valid',kernel=[11× 11])
['Batch',0,'None',0]	Batch_Normalization()

¹ String

همان‌گونه که در جدول (۳-۴) مشاهده می‌شود در مجموع ۲۲ کاراکتر متفاوت وجود دارد که البته اکثر آن‌ها مربوط به یک لایه‌ی کانولوشن ۲ بعدی (Conv2d) هست و تفاوت آن‌ها در تعداد فیلترهای یک لایه‌ی کانولوشن (Filters) که برابر است با تعداد تصاویر خروجی یک لایه کانولوشن، اندازه‌ی کرنل‌های هر فیلتر (Kernel) و در نهایت نحوه‌ی حاشیه گذاری تصاویر قبل از ضرب‌های کانولوشنی (Padding) که مشخص می‌کند تصاویر خروجی در اندازه‌ی مشابه با تصاویر ورودی باشند و یا طبق فرمول زیر از اندازه‌ی آن‌ها کاسته شود.

$$size(l_{n+1}) = size(l_n) - (size(kernel) - 1) \quad (۱-۴)$$

در کنار لایه‌های کانولوشن دوبعدی، لایه‌ی نرمال سازی دسته‌ای^۱ نیز وجود دارد که وظیفه‌ی نرمال سازی خروجی لایه‌ها را برای جلوگیری از بیش برآزش^۲ شدن نتایج را دارد. البته لازم به ذکر است که در این مدل از مساله^۳ محدودیت‌هایی در انتخاب لایه‌ها وجود دارد به این صورت که به طور کلی نمی‌شود از لایه‌های کاهش دهنده^۴ استفاده کرد زیرا نیاز است که همواره سایز تصاویر مقادیر فرد باشد. از طرف دیگر، در قالب مدل پایه نیاز است که در زیر شبکه‌های مورد بررسی اندازه‌ی تصاویر از (37×37) (اندازه‌ی پاره تصویر) به (1×1) کاهش پیدا کند. برای این منظور محدودیتی برای تعداد و نوع لایه‌های کانولوشن دوبعدی و با حاشیه گذاری معتبر^۵ وجود دارد که از فرمول زیر پیروی می‌کند.

$$Sum_{kernels \text{ in valid layers}} - Num_{valid \text{ layers}} = 36 \quad (۲-۴)$$

همان‌گونه که در فرمول مشاهده می‌شود نیاز است که حاصل تفریق مجموع کرنل‌های لایه‌های معتبر از تعداد لایه‌های معتبر همواره برابر با ۳۶ شود.

^۱ Batch_Normalization

^۲ OverFit

^۳ منظور مساله تطبیق استریو است

^۴ Pooling

^۵ Valid Padding

۴-۵- پیاده‌سازی الگوریتم سردشدن شبیه‌سازی شده

در این تحقیق برای پیاده‌سازی الگوریتم سردشدن شبیه‌سازی شده از کتابخانه‌ی آماده‌ی این الگوریتم در زبان برنامه‌نویسی پایتون^۱ به نام Simanneal^۱ استفاده شده است. برای پیاده‌سازی این فرایند به وسیله کتابخانه‌ی مذکور نیاز است که دو تابع انتقال^۲ که مسئولیت تغییر در ساختار رشته را دارد و انرژی^۳ که وظیفه‌ی میزان کاهش انرژی بر حسب انتقال صورت گرفته را دارد، نوشته شوند.

۴-۵-۱- تابع انتقال

همان‌گونه که ذکر شد، رشته‌های لایه‌ای تولیدشده ملزم به رعایت قوانینی هست. برای پیروی از قوانین نیاز است که اضافه کردن لایه‌های با حاشیه‌ی مشابه^۴ و لایه‌های با حاشیه‌ی معتبر به صورت جداگانه و در دو فضای مجزا تغییر یابند. از طرفی، در روش سردشدن شبیه‌سازی شده در هر تکرار^۵ فقط یک تغییر باید اتفاق بیفتد. در نتیجه، برای تغییر در رشته مورد نظر از توابع احتمال استفاده می‌شود. نهایتاً، الگویی که برای تغییر در ساختار رشته مورد نظر به دست آمد به صورت زیر است:

- رشته‌ی اولیه، به گونه‌ای انتخاب می‌شود که در آن فقط یک لایه با حاشیه‌ی معتبر و با بزرگ‌ترین کرنل ممکن (یعنی کرنل ۳۷ موجود باشد) و باقی لایه‌ها به صورت تصادفی^۶ مقداردهی می‌شوند.
- با احتمال ۲۰ درصد یکی از لایه‌های با حاشیه‌ی معتبر انتخاب شده و اگر سائز کرنل آن بزرگ‌تر از ۳ باشد. به دو لایه با کرنل‌های تصادفی ولی با شرط فرمول زیر تقسیم می‌شود.

^۱ Python

^۲ Move

^۳ Energy

^۴ Same Padding

^۵ Iteration

^۶ Random

$$kernel_{new\ layer\ 1} + kernel_{new\ layer\ 2} = kernel_{old\ layer} + 1 \quad (۳-۴)$$

- با احتمال ۵۰ درصد یکی از لایه‌هایی که جزو مجموعه‌ی لایه‌های با کرنل معتبر نیستند با لایه‌ای دیگر از خارج از آن مجموعه جایگزین می‌شود.
 - با احتمال ۳۰ درصد دو کاراکتر از رشته بصورت تصادفی انتخاب شده و جای آن‌ها باهم عوض می‌شود.
- لازم به ذکر است که الگوریتم به گونه‌ای طراحی شده است که تمامی فرایندهای تصادفی فقط و فقط در این تابع اتفاق بیفتند و رشته‌ی حاصل شده در دفعات مکرر زیر شبکه‌ی یکسانی تولید خواهد کرد.

۴-۵-۲- تابع انرژی

تابع انرژی کتابخانه‌ی آماده‌ی Simanneal برابر است با تابع برازندگی در ادبیات کلی الگوریتم‌های بهینه‌سازی، اما با این تفاوت که این مقدار که حتماً عددی است باید به صورت نزولی باشد و به سمت ۰ حرکت کند. در نتیجه نیاز است که مقادیری مانند درستی که عموماً به صورت صعودی هستند، ابتدا معکوس شده و سپس در تابع انرژی اعمال شوند.

تابع انرژی این فرایند همان گونه که گفته شد به صورت چندهدفه بوده و از دو فاکتور مرتبط با میزان درستی و حجم شبکه تشکیل شده است. در ادامه به معرفی هر یک از این فاکتورها و نحوه‌ی ادغام آن‌ها صحبت خواهد شد. فاکتور اول در این مساله که درستی است با فرمول زیر محاسبه می‌شود:

$$accuracy = (num_{correct\ pixels} \div num_{all\ pixels}) \times 100 \quad (۴-۴)$$

که در آن $num_{correct\ pixels}$ برابر است با تعداد پیکسل‌هایی که مقدار بین ۰ تا ۲۵۵ آن‌ها درست پیش‌بینی شده است و $num_{all\ pixels}$ برابر است با کل پیکسل‌های تصویر خروجی. فاکتور دوم، عملیات ممیز شناور به صورت نمایانگر این هستند که پردازه‌ی مورد نظر چقدر از توان مصرفی سیستم را درگیر می‌کند پس یکی از استانداردهای رایج برای محاسبه توان مصرفی یک سیستم،

تعداد عملیات ممیز شناور بر ثانیه است که به راحتی و با دقت بالا قابل محاسبه است. عموماً اندازه این مقدار بین چند میلیون تا چند میلیارد است.

برای محاسبه‌ی مقدار تابع انرژی، لازم است که ابتدا مقادیر درستی و تعداد عملیات ممیز شناور نرمال شوند. خوشبختانه مقادیر درستی به خودی خود نرمال و بین ۰ و ۱ می‌باشند اما برای متغیرهای شناور کار کمی پیچیده است. برای نرمال سازی این مقادیر از فرمول زیر استفاده شده است:

$$normalized_{FLOPs\ numbers} = \frac{max_{FLOPs\ numbers} - current_{FLOPs\ Numbers}}{max_{FLOPs\ numbers}} \quad (5-4)$$

که در آن $max_{FLOPs\ numbers}$ برابر است با بیش‌ترین مقدار ممکن برای تعداد عملیات ممیز شناور که در این مدل حدود ۲۰ پیش‌بینی شده است و $current_{FLOPs\ Numbers}$ برابر است با تعداد متغیرهای به دست آمده از مدل ساخته‌شده فعلی است. در نهایت فرمول محاسبه‌ی مقدار انرژی بصورت زیر است:

$$e = \alpha \times \left(\frac{1}{accuracy} \right) + (\alpha - 1) \times normalized_{FLOPs\ numbers} ; \quad (6-4)$$

$$0 \leq \alpha \leq 1$$

که در آن e برابر است با مقدار انرژی و α یک ثابت بوده که میزان اثرگذاری هر فاکتور را در مقدار نهایی تعیین می‌کند علاوه بر آن همان‌گونه که مشاهده می‌شود مقدار درستی به دلیل ماهیت رو به پایین تابع انرژی قبل از اثر دادن ابتدا معکوس شده است. در انتها باید به این مساله اشاره شود که زمان یادگیری کامل^۱ هر مدل در این نوع مساله بسیار طولانی است. بنابراین مدل‌ها به در زمان جستجو به صورت ناقص و با ۵۰۰۰ تکرار یادگیری^۲، آموخته‌شده‌اند، به همین دلیل درصد درستی در نمودارها تا حدی پایین است.

۶-۴-۶- اجرای الگوریتم سردشدن شبیه‌سازی

الگوریتم سرد شدن شبیه سازی شده در مجموعه سه مرتبه و با سه تنظیم متفاوت برای تابع

¹ Full Training

² epochs

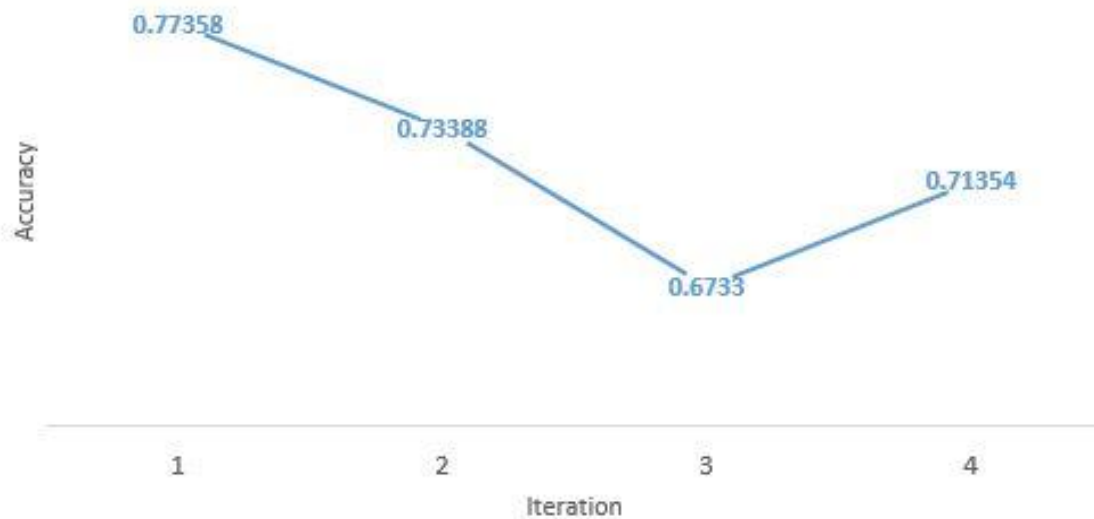
درستی انجام شده است.

۴-۶-۱- اجرای اول

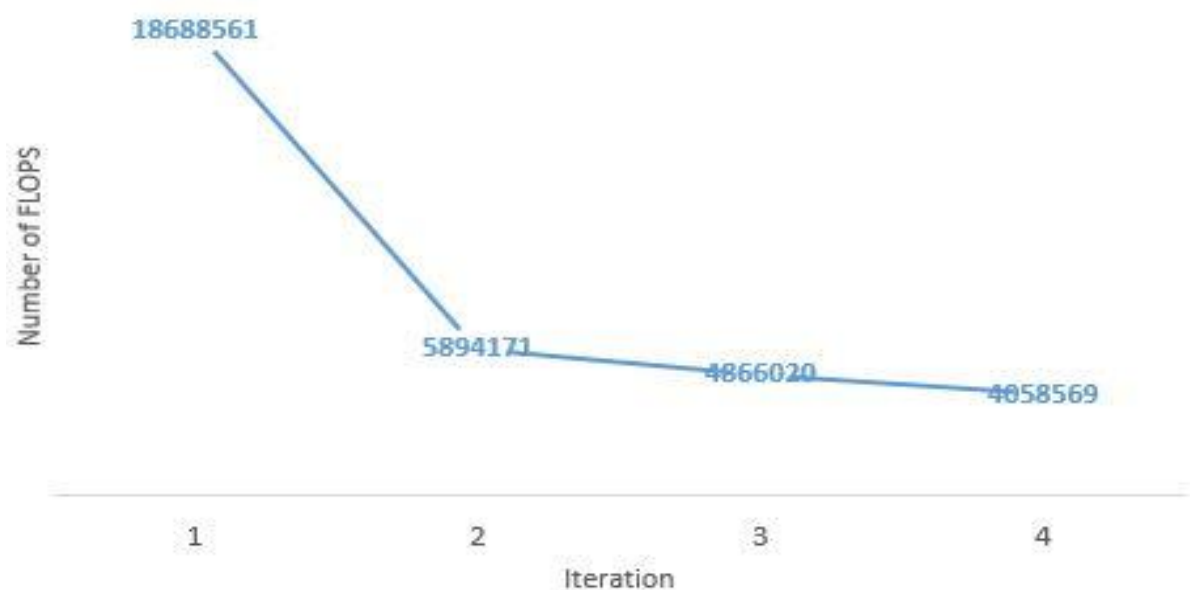
در اجرای اول α برابر با ۰/۵ در نظر گرفته شده است یعنی اهمیت دو فاکتور درستی و تعداد ممیزهای شناور شبکه به یک اندازه در نظر گرفته شده است. در این اجرا و پس از جستجو در ۱۷۸ مدل مختلف مجموعاً ۴ مرتبه بهترین مدل ارائه شده تغییر کرد مختصراً تغییرات به صورت زیر است.

جدول ۴-۲: نتایج اجرای اول الگوریتم سردشدن شبیه‌سازی شده

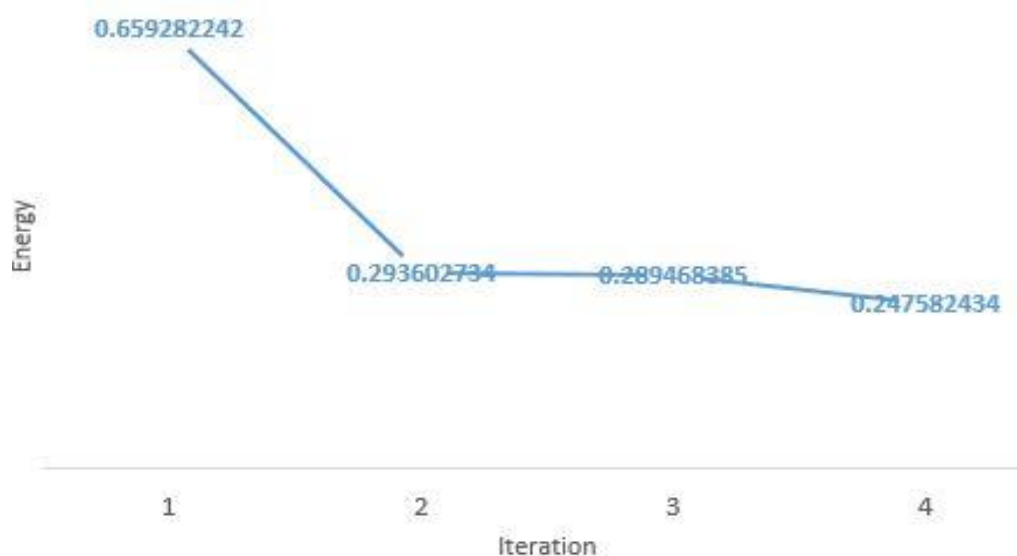
Number	Number of FLOPS	Accuracy	Energy
۱	18688561	0/77358	0/659282
۲	5894171	0/73388	0/289468
۳	4866020	0/6733	0/289468
۴	4058569	0/71354	0/247582



شکل ۴-۳: تغییرات درستی در اجرای اول



شکل ۴-۴: تغییرات تعداد عملیات ممیز شناور در اجرای اول



شکل ۴-۵: تغییرات انرژی در اجرای اول

تحلیل نمودارهای اجرای اول:

همان‌گونه که در شکل (۴-۳) مشاهده می‌شود مقدار درستی ابتدا به صورت کاهشی است و سپس بعد از تکرار سوم ناگهان شروع به افزایش می‌کند. برای توجیه این روند، نیاز است که نگاهی به شکل‌های (۴-۴) و (۴-۵) انداخت. از آنجا که تابع برازندگی به صورت مرکب بوده و در این مرحله از اجرا ضریب اثرگذاری دو مؤلفه‌ی درستی و تعداد عملیات ممیز شناور بر ثانیه یکسان بوده است. در نتیجه در ابتدای اجرا، کاهش چشمگیر تعداد عملیات ممیز شناور بر ثانیه، باعث شده است تا کاهش

نرخ درستی به چشم نیاید و در نتیجه جهت پیشروی به سمتی باشد که با کاهش هر دو مؤلفه همراه است. اما از تکرار سوم، با کاهش شیب نزولی تعداد عملیات ممیز شناور بر ثانیه، تأثیر کاهش نرخ درستی بیش از پیش به چشم آمده است و همین باعث شده که جهت حرکت به سمتی برگردد که با ثابت نگه‌داشتن نرخ تعداد عملیات ممیز شناور بر ثانیه و افزایش نرخ درستی همراه است. از آنجا که هدف اصلی تحقیق کاهش حداکثری تعداد عملیات ممیز شناور بر ثانیه و ثابت نگه‌داشتن نرخ درستی است، در نتیجه به نظر می‌رسد که این فرایند مسیر بهینه‌ی خود را طی نکرده است و خیلی زودتر از انتظار کاهش تعداد عملیات‌های ممیز شناور را متوقف کرده است و هنوز این امید وجود دارد که بتوان با تغییر نسبت اثرگذاری دو مؤلفه تغییر جهت حرکت کاوش را به سمت افزایش درستی کمی به تأخیر بیندازیم.

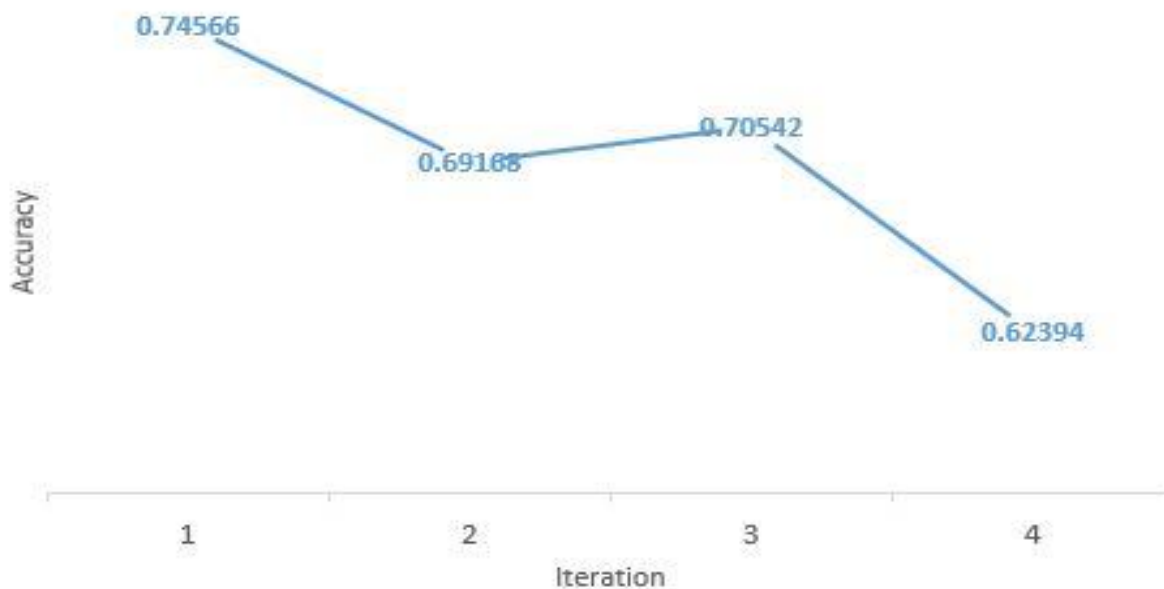
علاوه بر آن، شیب و سرعت همگرایی نمودار شکل (۴-۵) این احتمال را می‌دهد که کاوش در یک نقطه‌ی بهینه‌ی محلی قرار گرفته است که الگوریتم سردشدن شبیه‌سازی شده توان گریز از آن را نداشته است. در نتیجه امکان پیدا کردن نقطه‌ی پاسخ بهینه‌تر بسیار زیاد است.

۴-۶-۲- اجرای دوم

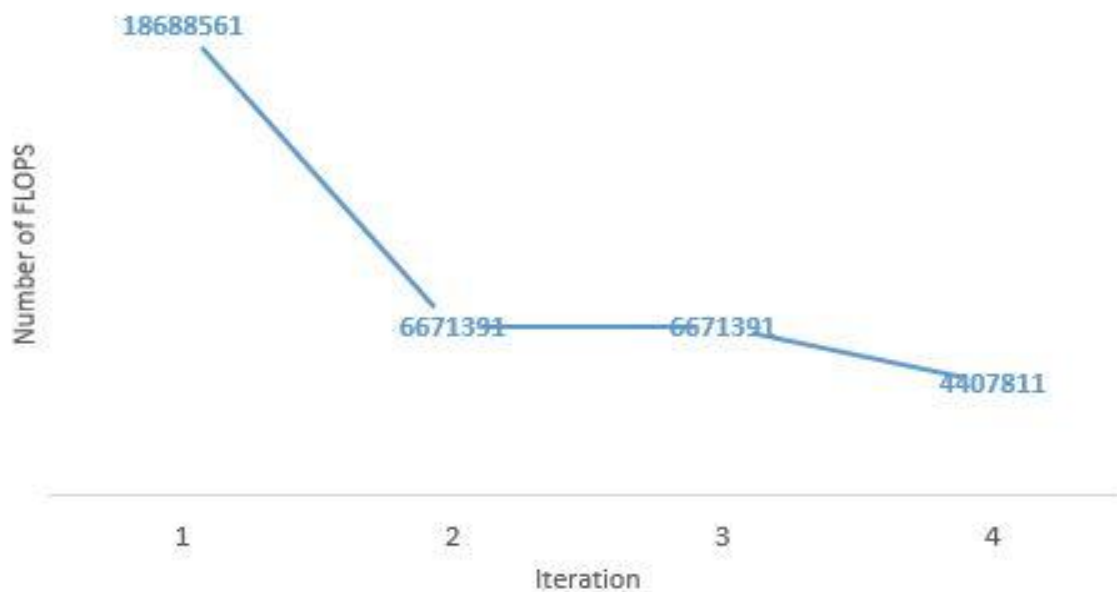
در این مرحله مقدار ثابت α از 0.5 به 0.44 کاهش پیدا کرد تا تأثیر تعداد عملیات ممیز شناور در تابع انرژی کمی بیشتر از میزان درستی شود. نتایج اجرای دوم و پس از جستجو در ۳۰۲ مدل ایجادشده و پس از ۴ بار بهبود به شکل زیر در آمد.

جدول ۴-۳: تغییرات بهترین مدل در اجرای دوم

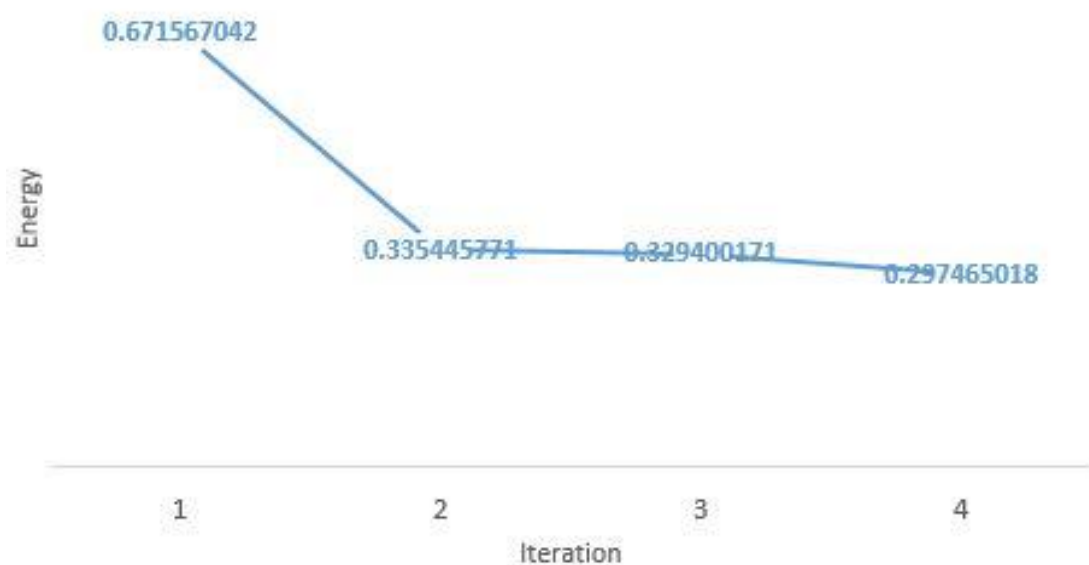
Number	Accuracy	Number of FLOPs	Energy
0	0/74566	1868561	0/671567
1	0/69168	6671391	0/335446
2	0/70542	6671391	0/3294
3	0/62394	4407811	0/297465



شکل ۴-۶: نمودار تغییرات درستی در اجرای دوم



شکل ۴-۷: نمودار تغییرات تعداد عملیات ممیز شناور در اجرای دوم



شکل ۴-۸: نمودار تغییرات انرژی در اجرای دوم

تحلیل نمودارهای اجرای دوم:

در اجرای دوم، همان‌گونه که در نمودار شکل (۴-۶) مشاهده می‌شود نرخ درستی ابتدا کاهش داشته است و سپس افزایش اما در انتها باز هم کاهش داشته و به کمینه‌ترین مقدار خود رسیده است. برای توجیه آن می‌شود به تأثیر کاهش مقدار α اشاره کرد که باعث شده است تا شیب نمودار (۴-۷) که تقریباً مشابه با اجرای اول است بیشتر غالب شده و کاهش مقدار درستی از اهمیت کمتری در مقابل آن برخوردار باشد. به صورت کلی و در زمانی که نیاز باشد مقدار درستی به حداکثر خود برسد باید مقدار α را به حالت اولیه برگرداند. اما در این مساله، از آن جایی که یادگیری هر مدل به صورت ناقص است، مقدار درستی حتی می‌تواند کمتر از کمینه‌ی به دست آمده و در حدود ۵۰ درصد باشد. اما کاهش بیشتر مقدار α می‌تواند موجب شود که مقدار درستی عموماً به چشم نیاید و جهت‌گیری کاوش فقط به سمت بهبود هرچه بیشتر تعداد عملیات ممیز شناور بر ثانیه شود. علاوه بر این در این مساله مقداردهی استاندارد محاسبه شده و در حدود ۵۰ درصد است. در نتیجه باید از سازوکاری استفاده شود که جهت پیشروی را به سمتی ببرد که این مقدار ثابت بماند.

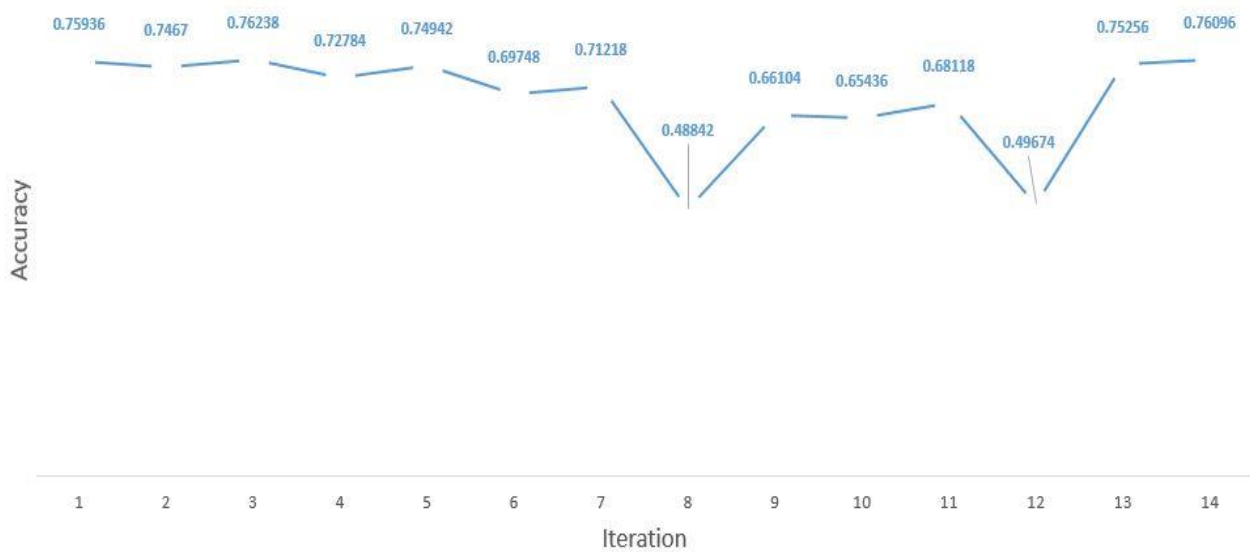
۴-۶-۳- اجرای سوم

در این مرحله برای برقراری تعادل بهتر میان دو متغیر و تا حدودی اعمال اهمیت بیشتر به تعداد عملیات ممیز شناور، از تابع جریمه^۱ در تابع انرژی استفاده شد. به این شکل که در صورتی که میزان درستی از ۵۰ درصد بیشتر شد، مقدار انرژی با ۰/۱ جمع شود. نتایج به دست آمده در ۳۰۴ مدل جستجو شده و پس از ۱۶ بار تغییر بهترین به صورت زیر است.

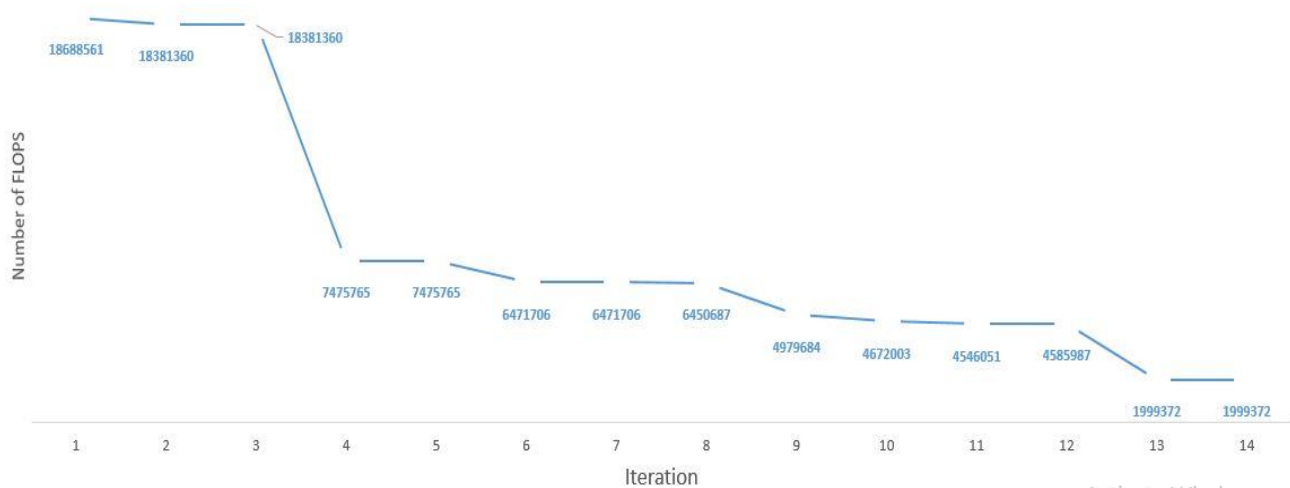
جدول ۴-۴: تغییرات بهترین مدل در اجرای سوم

Number	Accuracy	Number of FLOPs	Energy
0	0/75936	۱۸۶۸۸۵۶۱	0/765539
1	0/7467	18381360	0/76191
2	0/76238	18381360	0/755011
3	0/72784	7475765	0/443624
4	0/74942	7475765	0/434128
5	0/69748	6471706	0/426914
6	0/71218	6471706	0/420446
7	0/48842	6450687	0/418271
8	0/66104	4979684	0/398267
9	0/65436	4672003	0/391992
10	0/68118	4546051	0/376419
11	0/49674	4585987	0/358769
12	0/75256	1999372	0/26748
13	0/76096	1999372	0/265052
14	0/6073	1564118	0/254651

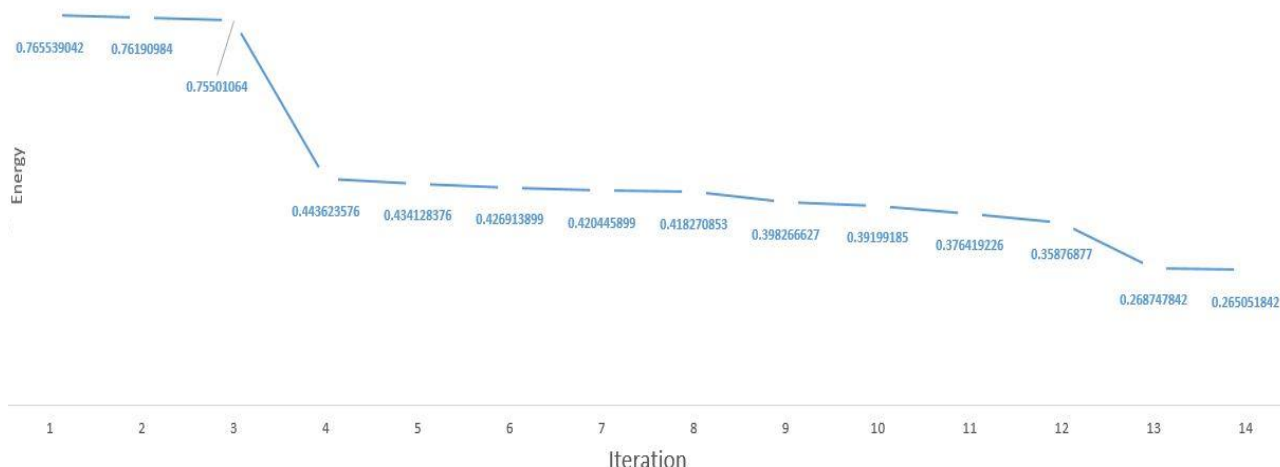
¹ Penalty Function



شکل ۴-۹: نمودار تغییرات در اجرای سوم



شکل ۴-۱۰: نمودار تغییرات تعداد عملیات ممیز شناور در اجرای سوم



شکل ۴-۱۱: نمودار تغییرات انرژی در اجرای سوم

تحلیل نمودارهای اجرای سوم:

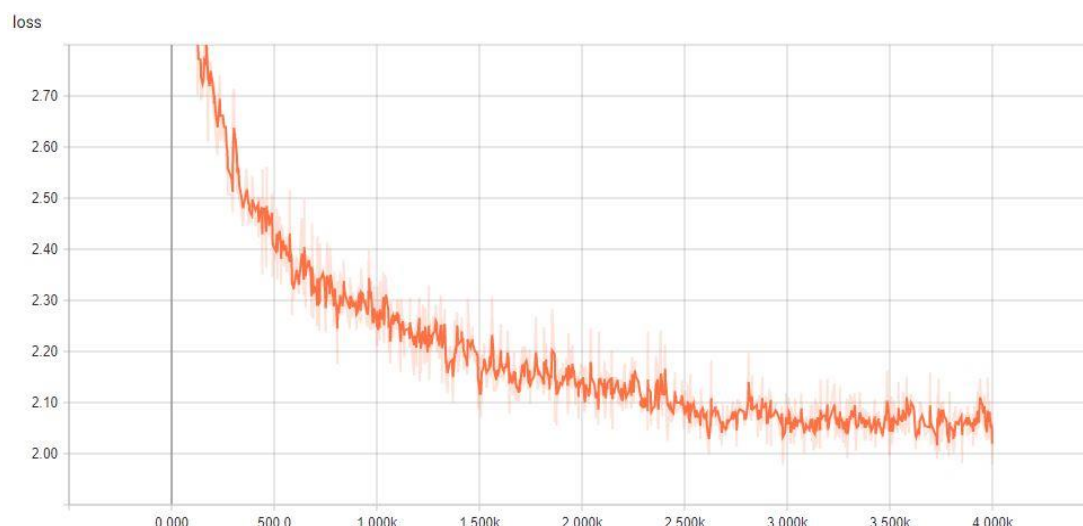
در این اجرا، همان‌گونه که انتظار می‌رفت مقدار درستی (مطابق با نمودار شکل ۴-۹) در یک محدوده‌ی ثابت فقط نوسان می‌کند که این باعث می‌شود مقدار درستی همانی شود که انتظار می‌رود و در کنار آن این فرصت برای تعداد عملیات ممیز شناور بر ثانیه، به وجود می‌آید که تا آخرین حد ممکن کاهش پیدا کنند. البته لازم به ذکر است که این اجرا هم لزوماً به بهینه‌ترین نقطه‌ی خود نرسید یعنی هنوز این امکان وجود دارد تا با تغییر و بهینه‌سازی تابع برازندگی نتیجه را بهبود داد. اما به دلیل محدودیت منابع و از طرفی ارضای نیازمندی مساله باعث شد تا عملیات جستجو در همین مرحله متوقف شود و نقطه‌ای که بهترین پوشش را برای نیازمندی مساله داشت از میان جواب‌های ارائه‌شده انتخاب شود و به صورت کامل یادگیری شود.

نقطه‌ی انتخاب شده با پیکربندی لایه‌ای شکل ۴-۱۲) می‌باشد که نتایج یادگیری کامل آن در شکل ۴-۱۳) و جدول ۴-۵) قابل مشاهده است.

Conv2D(64, Same, [5*5])
Conv2D(64, Valid, [3*3])
Conv2D(64, Same, [5*5])
Conv2D(32, Valid, [3*3])
Conv2D(32, Valid, [3*3])
Conv2D(32, Valid, [5*5])
Conv2D(64, Same, [5*5])
BatchNormalization
Conv2D(32, Valid, [5*5])
Conv2D(32, Same, [5*5])
Conv2D(32, Valid, [7*7])
BatchNormalization
BatchNormalization
Conv2D(32, Valid, [9*9])
Conv2D(32, Valid, [7*7])
Conv2D(32, Valid, [3*3])

شکل ۴-۱۲: نمایی از بهترین مدل به دست آمده توسط الگوریتم سرشدن

پس از اجرای این مدل به صورت یادگیری کامل، مدل درستی ای در برابر با ۰/۸۹۳۴۵ را ارائه نمود که تقریباً برابر با درستی مدل پایه بود. شکل (۴-۱۳) نمودار کاهش مقدار خطای مدل به دست آمده را بر حسب ادوار^۱ یادگیری نشان می‌دهد. همان‌گونه که مشاهده می‌شود، شیب نمودار ابتدا بسیار زیاد است و با گذشت دوره‌ها و تقریباً از دوره‌ی ۳ هزارم به سمت انتها، شیب تقریباً صفر شده است که این نمایانگر پایان ظرفیت ارتقا درستی در مدل است و می‌توان انتظار داشت که درصد درستی در همان حدود ۹۰ درصد باقی خواهد ماند و افزایش تعداد ادوار یادگیری کمک چندانی به افزایش درصد درستی نخواهد کرد.



شکل ۴-۱۳: نمودار کاهش میزان خطا بر حسب تکرار یادگیری

¹ Epochs

جدول ۴-۵: مقایسه‌ی نتایج مدل بهینه‌شده‌ی به دست آمده و مدل پایه‌ی Content-CNN

	Accuracy	Number of FLOPS
مدل بهینه‌شده	۸۹/۳٪	۱/۵۶M
مدل Content-CNN	۹۰٪	۲M
درصد تفاوت	۰/۰۰۷٪	۲۲٪

همان‌گونه که در جدول (۴-۵) مشاهده می‌شود بهینه‌ترین مدل بدست آمده توسط الگوریتم سرد شدن شبیه‌سازی شده در مقایسه با مدل پایه که معیار بهینگی کار است، توانست با کاهش کمتر از یک درصد از نرخ درستی، بیش از ۲۰ درصد از بار پردازشی شبکه را کاهش دهد.

فصل پنجم: جمع‌بندی و پیشنهادها

۵-۱- خلاصه

به طور خلاصه هدف از این تحقیق بررسی عملکرد و توانایی الگوریتم سردشدن شبیه‌سازی شده در جستجو و معرفی مدلی بهینه از دو جنبه‌ی درصد درستی و حجم پردازش‌ها در کاربرد سیستم‌های تعبیه شده و تلفن‌های هوشمند بود. خروجی این تحقیق مدلی در چارچوب شبکه‌ی Siamese بود که علاوه بر حفظ میزان درستی تقریبی ۹۰ درصد مدل پایه، حجم پردازش‌های شبکه را ۲۵ درصد کاهش داد.

۵-۲- جمع‌بندی

امروزه یکی از نیازهای صنعت رایانه، کاهش میزان پردازش‌های یک مورد نیاز در محاسبات است. از آنجایی که حجم بسیار زیادی از پردازش‌های کنونی در سیستم‌های هوش مصنوعی و علی‌الخصوص شبکه‌های عصبی است، این فرایندها مورد توجه بسیار زیادی قرار گرفته‌اند. یکی از دلایل تحقیقات عمده در کاهش هزینه‌ها، گسترش بازار تلفن‌های هوشمند و ابزار مبتنی بر اینترنت اشیا که عمدتاً دارای منابع انرژی و پردازشی محدود هستند، است.

یکی از روش‌های پرکاربرد در طراحی شبکه‌های عصبی بهینه و دقیق، استفاده از یادگیری ماشین خودکار^۱ است که در آن‌ها هدف، بهره‌گیری از روش‌های بهینه‌سازی مبتنی بر تکامل نظیر الگوریتم ژنتیک و سردشدن شبیه‌سازی شده در جهت تولید و طراحی مدل‌های مناسب برای کاربردهای مختلف است که امروزه به صورت مجزا تبدیل به یک شاخه‌ی علم شده است متخصصان خود را دارد. از کاربردهای این روش، بهینه‌سازی روش‌های تطبیق استریو بوده که به خودی خود حجم پردازش‌های سنگینی دارند.

۵-۳- پیشنهاد

در این تحقیق، هدف طراحی شبکه‌ی عصبی‌ای بود که علاوه بر پردازش صحیح، توان مصرفی پایینی نیز داشته باشد. در این راه به دلیل محدودیت‌های منابع پردازشی سعی شد که فضای جستجو

¹ AutoML

حتی‌الامکان کوچک باشد و علاوه بر آن از روشی برای جستجو استفاده شد که برای رسیدن به جواب نهایی، مدل کمتری تولید کند. به علاقه‌مندان به تحقیق در این زمینه توصیه می‌شود که با در نظر گرفتن موارد زیر فضای جستجو را تا حد امکان گسترش دهند.

۱. استفاده از لایه‌های متنوع‌تر

۲. اضافه کردن فرامپارامترها^۱ نظیر روش‌های بهینه‌سازی شبکه، نرخ یادگیری^۲، توابع

فعال‌سازی^۳ و توابع خطا^۴

۳. گسترش فضای قرارگیری لایه‌ها از یک بعد به چندبعدی

۴. استفاده از بهینه‌سازی‌های چند مرحله‌ای

¹ Hyper Parameters

² Learning Rate

³ Activation Functions

⁴ Loss Functions

مراجع

مراجع

- [1] D. Scharstein, and Richard Szeliski. "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms." *International journal of computer vision* 47.1-3 (2002): 7-42.
- [2] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, pp. 7-42, 2002.
- [3] O. Faugeras and Q.-T. Luong, "The geometry of multiple images. With contributions from Theo Papadopoulos," ed: MIT Press, Cambridge, MA, 2001.
- [4] S. Tobisch and T. Ziegler, "Catalytic Oligomerization of Ethylene to Higher Linear α -Olefins Promoted by the Cationic Group 4 [(η 5-Cp-(CMe₂-bridge)-Ph) MII (ethylene) 2]+(M= Ti, Zr, Hf) Active Catalysts: A Density Functional Investigation of the Influence of the Metal on the Catalytic Activity and Selectivity," *Journal of the American Chemical Society*, vol. 126, pp. 9059-9071, 2004.
- [5] D. A. Forsyth and J. Ponce, *Computer vision: a modern approach*: Prentice Hall Professional Technical Reference, 2002.
- [6] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*: Cambridge university press, 2003.
- [7] J. Segen and S. Kumar, "Shadow gestures: 3D hand pose estimation using a single camera," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, 1999, pp. 479-485.
- [8] M. Wimmer, D. Scherzer, and W. Purgathofer, "Light space perspective shadow maps," *Rendering Techniques*, vol. 2004, p. 15th, 2004.
- [9] V. P. Namboodiri and S. Chaudhuri, "On defocus, diffusion and depth estimation," *Pattern Recognition Letters*, vol. 28, pp. 311-319, 2007.
- [10] S. Bae and F. Durand, "Defocus magnification," in *Computer Graphics Forum*, 2007, pp. 571-579.
- [11] A. P. Pentland, "A new sense for depth of field," *IEEE transactions on pattern analysis and machine intelligence*, pp. 523-531, 1987.
- [12] T. Rajabzadeh, A. Vahedian, and H. Pourreza, "Static Object Depth Estimation Using Defocus Blur Levels Features," in *2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, 2010, pp. 1-4.
- [13] P. Földiák, "Learning invariance from transformation sequences," *Neural Computation*, vol. 3, pp. 194-200, 1991.
- [14] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural networks*, vol. 1, pp. 119-130, 1988.
- [15] Y. LeCun, Fu Jie Huang, and Leon Bottou. "Learning methods for generic object recognition with invariance to pose and lighting." *CVPR* (2). 2004.
- [16] L. Bottou, and Vladimir Vapnik. "Local learning algorithms." *Neural computation* 4.6 (1992): 888-900.

- [17] R. A. Hamzah, and Haidi Ibrahim. "Literature survey on stereo vision disparity map algorithms." *Journal of Sensors*, 2016,
- [18] D. Scharstein and R. Szeliski, "Middlebury stereo vision page," ed, 2002.
- [19] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "The KITTI vision benchmark suite," *URL <http://www.cvlibs.net/datasets/kitti>*, 2015.
- [20] "استخراج عمق در تصاویر دو چشمی با استفاده از روش تطبیق تقاطعی," وزارت علوم، تحقیقات و فناوری - دانشگاه صنعتی خواجه نصیرالدین طوسی - دانشکده برق، ۱۳۹۲.
- [21] ر. پیمان، ر. وحید، ط. زینب، "بهینه سازی نگاشت اختلاف مکان در تطبیق استریو با استفاده از اتوماتای سلولی دوبعدی،" ارائه شده در دومین کنفرانس بین المللی بازشناسی الگو و تحلیل تصویر، ۱۳۹۳.
- [22] س. محمدرضا غریب و خ. حسین، "بهبود تطابق استریو در سطح ویژگی با استفاده از گوشه یابی،" ارائه شده در دومین کنفرانس بین المللی بازشناسی الگو و تحلیل تصویر، ۱۳۹۳.
- [23] "تطابق استریو با استفاده از ویژگی های رنگ و بافت،" وزارت علوم، تحقیقات و فناوری - دانشگاه صنعتی شاهرود - دانشکده برق و کامپیوتر، ۱۳۹۲.
- [24] "تشخیص عمق تصویر با استفاده از عمق یابی از تار و آنالیز استریو،" ۱۳۸۱.
- [25] ت. سکینه شیرازی، م. پیمان، ع. محسن، "ارائه روشی نوین برای تطبیق استریو به کمک مجموع فاصله همینگ،" ارائه شده در دومین کنفرانس ملی مهندسی برق، ۱۳۸۸.
- [26] J. Zbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1592-1599.
- [27] W. Luo, A. G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5695-5703.
- [28] J. Zbontar and Y. LeCun, "Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches," *Journal of Machine Learning Research*, vol. 17, p. 2, 2016.
- [29] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4353-4361.
- [30] S. Khamis, S. Fanello, C. Rhemann, A. Kowdle, J. Valentin, and S. Izadi, "Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 573-590.
- [31] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, et al., "End-to-end learning of geometry and context for deep stereo regression," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 66-75.

- [32] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5410-5418.
- [33] P. Knobelreiter, C. Reinbacher, A. Shekhovtsov, and T. Pock, "End-to-end training of hybrid CNN-CRF models for stereo," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2339-2348.
- [34] R. Slossberg, A. Wetzler, and R. Kimmel, "Deep stereo matching with dense CRF priors," *arXiv preprint arXiv:1612.01725*, 2016.
- [35] Y. Li, Y. Hu, R. Song, P. Rao, and Y. Wang, "Coarse-to-fine PatchMatch for dense correspondence," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, pp. 2233-2245, 2017.
- [36] Y. Luo, J. Ren, M. Lin, J. Pang, W. Sun, H. Li, *et al.*, "Single view stereo matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 155-163.
- [37] J. Pang, W. Sun, J. S. Ren, C. Yang, and Q. Yan, "Cascade residual learning: A two-stage convolutional neural network for stereo matching," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 887-895.
- [38] L. Chen, L. Fan, J. Chen, D. Cao, and F. Wang, "A full density stereo matching system based on the combination of CNNs and slanted-planes," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.
- [39] Z. Liang, Y. Feng, Y. Guo, H. Liu, W. Chen, L. Qiao, *et al.*, "Learning for disparity estimation through feature constancy," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2811-2820.
- [40] C. Zhou, H. Zhang, X. Shen, and J. Jia, "Unsupervised learning of stereo matching," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1567-1575.
- [41] N. Luo, C. Yang, W. Sun, and B. Song, "Unsupervised stereo matching with occlusion-aware loss," in *Pacific Rim International Conference on Artificial Intelligence*, 2018, pp. 758-746.
- [42] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, *et al.*, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4040-4048.
- [43] A. Tonioni, M. Poggi, S. Mattoccia, and L. Di Stefano, "Unsupervised adaptation for deep stereo," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1605-1613.
- [44] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697-8710.
- [45] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proceedings of the aaai conference on artificial intelligence*, 2019, pp. 4780-4789.

-
- [46] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.
 - [47] L.-C. Chen, M. Collins, Y. Zhu, G. Papandreou, B. Zoph, F. Schroff, *et al.*, "Searching for efficient multi-scale architectures for dense image prediction," in *Advances in Neural Information Processing Systems*, 2018, pp. 8699-8710.
 - [48] M. Feurer and F. Hutter, "Hyperparameter optimization," in *Automated Machine Learning*, ed: Springer, 2019, pp. 3-33.
 - [49] J. Vanschoren, "Meta-learning: A survey," *arXiv preprint arXiv:1810.03548*, 2018.
 - [50] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251-1258.
 - [51] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.
 - [52] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning," *arXiv preprint arXiv:1611.02167*, 2016.
 - [53] M. Suganuma, S. Shirakawa, and T. Nagao, "A genetic programming approach to designing convolutional neural network architectures," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2017, pp. 497-504.
 - [54] H. Cai, T. Chen, W. Zhang, Y. Yu, and J. Wang, "Efficient architecture search by network transformation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
 - [55] H. Mendoza, A. Klein, M. Feurer, J. T. Springenberg, and F. Hutter, "Towards automatically-tuned neural networks," in *Workshop on Automatic Machine Learning*, 2016, pp. 58-65.
 - [56] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "SMASH: one-shot model architecture search through hypernetworks," *arXiv preprint arXiv:1708.05344*, 2017.
 - [57] T. Elsken, J.-H. Metzen, and F. Hutter, "Simple and efficient architecture search for convolutional neural networks," *arXiv preprint arXiv:1711.04528*, 2017.
 - [58] T. Elsken, J. H. Metzen, and F. Hutter, "Efficient multi-objective neural architecture search via lamarckian evolution," *arXiv preprint arXiv:1804.09081*, 2018.
 - [59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.
 - [60] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700-4708.
 - [61] P. J. Angeline, G. M. Saunders, and J. B. Pollack, "An evolutionary algorithm that constructs recurrent neural networks," *IEEE transactions on Neural Networks*, vol. 5, pp. 54-65, 1994.

- [62] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary computation*, vol. 10, pp. 99-127, 2002.
- [63] D. Floreano, P. Dürri, and C. Mattiussi, "Neuroevolution: from architectures to learning," *Evolutionary intelligence*, vol. 1, pp. 47-62, 2008.
- [64] K. O. Stanley, D. B. D'Ambrosio, and J. Gauci, "A hypercube-based encoding for evolving large-scale neural networks," *Artificial life*, vol. 15(2), pp. 185-212, 2009.
- [65] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *International Conference on Machine Learning*, 2015, pp. 2342-2350.
- [66] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," 2013.
- [67] T. Domhan, J. T. Springenberg, and F. Hutter, "Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
- [68] I. Bello, B. Zoph, V. Vasudevan, and Q. V. Le, "Neural optimizer search with reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017, pp. 459-468.
- [69] G. F. Miller, P. M. Todd, and S. U. Hegde, "Designing Neural Networks using Genetic Algorithms," in *ICGA*, 1989, pp. 379-384.
- [70] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, J. Tan, *et al.*, "Large-scale evolution of image classifiers," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017, pp. 2902-2911.
- [71] H. Liu, K. Simonyan, O. Vinyals, C. Fernando, and K. Kavukcuoglu, "Hierarchical representations for efficient architecture search," *arXiv preprint arXiv:1711.00436*, 2017.
- [72] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, *et al.*, "Evolving deep neural networks," in *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, ed: Elsevier, 2019, pp. 293-312.
- [73] L. Xie and A. Yuille, "Genetic cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1379-1388.
- [74] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *arXiv preprint arXiv:1603.06560*, 2016.
- [75] A. Zela, A. Klein, S. Falkner, and F. Hutter, "Towards automated deep learning: Efficient joint neural architecture and hyperparameter search," *arXiv preprint arXiv:1807.06906*, 2018.
- [76] S. Falkner, A. Klein, and F. Hutter, "BOHB: Robust and efficient hyperparameter optimization at scale," *arXiv preprint arXiv:1807.01774*, 2018.
- [77] F. Runge, D. Stoll, S. Falkner, and F. Hutter, "Learning to Design RNA," *arXiv preprint arXiv:1812.11951*, 2018.

-
- [78] K. Swersky, J. Snoek, and R. P. Adams, "Freeze-thaw Bayesian optimization," *arXiv preprint arXiv:1406.3896*, 2014.
 - [79] A. Klein, S. Falkner, J. T. Springenberg, and F. Hutter, "Learning curve prediction with Bayesian neural networks," 2016.
 - [80] B. Baker, O. Gupta, R. Raskar, and N. Naik, "Accelerating neural architecture search using performance prediction," *arXiv preprint arXiv:1705.10823*, 2017.
 - [81] H. Cai, J. Yang, W. Zhang, S. Han, and Y. Yu, "Path-level network transformation for efficient architecture search," *arXiv preprint arXiv:1806.02639*, 2018.
 - [82] S. Saxena and J. Verbeek, "Convolutional neural fabrics," in *Advances in Neural Information Processing Systems*, 2016, pp. 4053-4061.
 - [83] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, "Efficient neural architecture search via parameter sharing," *arXiv preprint arXiv:1802.03268*, 2018.
 - [84] G. Bender, "Understanding and simplifying one-shot architecture search," 2018.
 - [85] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *arXiv preprint arXiv:1806.09055*, 2018.
 - [86] H. Cai, L. Zhu, and S. Han, "Proxylessnas: Direct neural architecture search on target task and hardware," *arXiv preprint arXiv:1812.00332*, 2018.
 - [87] S. Xie, H. Zheng, C. Liu, and L. Lin, "SNAS: stochastic neural architecture search," *arXiv preprint arXiv:1812.09926*, 2018.
 - [88] A. Majd, S. Lotfi, G. Sahebi, M. Daneshtalab, and J. Plosila, "PICA: multi-population implementation of parallel imperialist competitive algorithms," in *2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, 2016, pp. 248-255.
 - [89] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, "BOA: The Bayesian optimization algorithm ", in *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1*, 1999, pp. 525-532.
 - [90] A. Das and B. K. Chakrabarti, *Quantum annealing and related optimization methods* vol. 679: Springer Science & Business Media, 2005.

پیوست‌ها

پیوست الف

