

پایان نامه دوره کارشناسی مهندسی کامپیوتر – نرم افزار

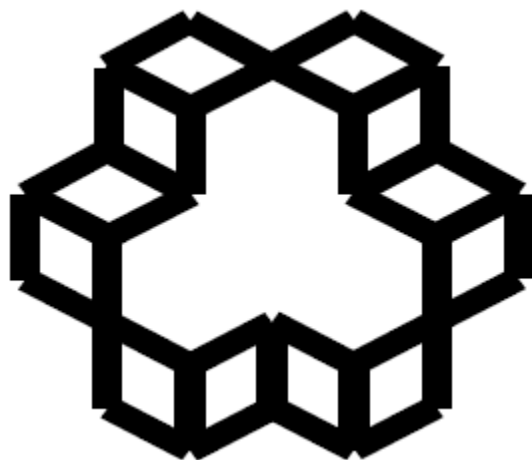
پیاده سازی الگوریتم میدان متوسط برای استنتاج در
میدان های تصادفی کاملاً متصل برای قطعه بندی
معنایی تصاویر

نگارش: حامد خشه چی

استاد راهنما: دکتر بهروز نصیحت کن

تابستان ۱۳۹۷

سُبْحَانَكَ اللَّهُمَّ رَبِّ السَّمَاوَاتِ
الْأَعْلَى رَبِّ الْعَرْشِ الْمَجِيدِ
الْكَرِيمِ إِلَهَ الْإِسْلَامِ
مَلِكِ الْيَوْمِ وَالْآخِرِ
يَا حَيُّ يَا قَيُّوْمُ
يَا ذَا الْجَلَالِ وَالْإِكْرَامِ



دانشگاه صنعتی خواجه نصیرالدین طوسی

دانشکده مهندسی کامپیوتر

تأییدیه هیات داوران

هیات داوران پس از مطالعه پایان نامه و شرکت در جلسه دفاع از پایان نامه تهیه شده تحت عنوان:

پیاده‌سازی الگوریتم میدان متوسط برای استنتاج در میدان‌های تصادفی کاملاً

متصل برای قطعه‌بندی معنایی تصاویر

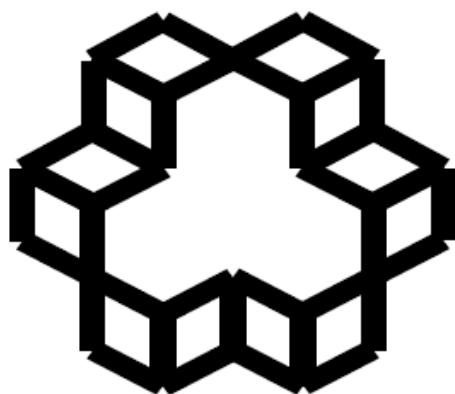
توسط آقای حامد خشه‌چی صحت و کفایت تحقیق انجام شده را برای اخذ درجه کارشناسی
در رشته مهندسی کامپیوتر گرایش نرم افزار با رتبه مورد تأیید قرار می‌دهند.

امضا

۱. استاد راهنما آقای دکتر بهروز نصیحت کن

امضا

۲. استاد ارزیاب آقای دکتر بابک ناصر شریف



دانشگاه صنعتی خواجه نصیرالدین طوسی

دانشکده مهندسی کامپیوتر

اظهار نامه دانشجو

اینجانب حامد خشه‌چی دانشجوی مقطع کارشناسی رشته مهندسی کامپیوتر گرایش نرم‌افزار گواهی مینمایم که تحقیقات ارائه شده در پروژه با عنوان:

پیاده‌سازی الگوریتم میدان متوسط برای استنتاج در میدان‌های تصادفی کاملاً متصل برای قطعه‌بندی معنایی تصاویر

با راهنمایی استاد محترم جناب آقای دکتر بهروز نصیحت‌کن توسط شخص اینجانب انجام شده است. صحت و اصالت مطالب نگارش شده در این پروژه مورد تأیید می‌باشد و در تدوین متن پروژه چارچوب (فرمت) مصوب دانشگاه را به طور کامل رعایت کرده‌ام.

امضا دانشجو:

تاریخ:

حق طبع، نشر و مالکیت نتایج

۱. حق چاپ و تکثیر این پروژه متعلق به نویسنده و استاد راهنمای آن میباشد. هرگونه تصویربرداری از کل یا بخشی از پروژه تنها با موافقت نویسنده یا استاد راهنما یا کتابخانه دانشکده مهندسی برق و کامپیوتر دانشگاه صنعتی خواجه نصیرالدین طوسی مجاز میباشد.
۲. کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی خواجه نصیرالدین طوسی می باشد و بدون اجازه کتبی دانشگاه به شخص ثالث قابل واگذاری نیست.
۳. استفاده از اطلاعات و نتایج موجود پروژه بدون ذکر مرجع مجاز نمی باشد.

خدای را بسی شاکرم که از روی کرم، پدر و مادری فداکار نصیبم ساخته تا در سایه
درخت پربار وجودشان بیاسایم و از ریشه آنها شاخ و برگ گیرم و از سایه وجودشان در
راه کسب علم و دانش تلاش نمایم. والدینی که بودنشان تاج افتخاری است بر سرم و
نامشان دلیلی است بر بودنم، چرا که این دو وجود، پس از پروردگار، مایه هستی‌ام بوده
اند دستم را گرفتند و مرا راه رفتن در این وادی زندگی پر فراز و نشیب آموختند.
آموزگارانی که برایم زندگی، بودن و انسان بودن را معنا کردند.
تقدیم به وجود با ارزشتان...

با تقدیر و تشکر شایسته از استاد فرهیخته و فرزانه جناب آقای دکتر بهروز نصیحت
کن که با نکته های دلاویز و گفته های بلند ، صحیفه های سخن را علم پرور نمود و
همواره راهنما و راه گشای من در اتمام و اکمال پایان نامه بوده است.

چکیده

با گسترش روز افزون استفاده از تصاویر نیاز به قطعه بندی و مشخص کردن قطعه ها دیده می شود. یکی از روش های مدل کردن استفاده از میدان های شرطی متصل به یکدیگر می باشد. هر چه تعداد اتصالات بیشتر باشد دقت ما بالاتر رفته اما سرعت ما پایین تر می آید. اما اگر ما از میدان های شرطی کاملاً متصل به یکدیگر استفاده کنیم می توانیم از تقریبی به نام میدان متوسط استفاده کنیم. با استفاده از این الگوریتم دقت ما تقریباً کم نشده ولی سرعت ما چندین برابر می شود.

واژه های کلیدی: قطعه بندی، میدان های شرطی، میدان های شرطی کاملاً به هم متصل، میدان متوسط.

فهرست مطالب

فصل اول..... ۱

مقدمه..... ۱

فصل دوم..... ۵

قطعه بندی معنایی تصاویر..... ۵

۱-۲-روش حد آستانه..... ۶

۲-۲-روش های مبتنی بر خوشه بندی..... ۶

۳-۲-قطعه بندی حرکتی و تعاملی..... ۶

۴-۲-قطعه بندی مبتنی بر فشرده سازی..... ۷

۵-۲-قطعه بندی مبتنی بر هیستوگرام..... ۷

۶-۲-تشخیص لبه..... ۷

۷-۲-خوشه بندی دوگانه..... ۸

۸-۲-ناحیه ی در حال رشد..... ۸

۹-۲-قطعه بندی مبتنی بر معادله دیفرانسیل مشتقات جزئی..... ۹

۱۰-۲-قطعه بندی توسط واریانس..... ۹

۱۱-۲-مدل گراف های احتمالاتی..... ۹

۱۲-۲-قطعه بندی توسط آب پخشانی..... ۹

۱۳-۲-باقی روش ها..... ۱۰

فصل سوم..... ۱۱

مدل های گرافی احتمالاتی و میدان متوسط..... ۱۱

۱-۳-مدل های گرافی احتمالاتی..... ۱۲

۱-۱-۳-شبکه های بیزین..... ۱۲

۲-۱-۳-شبکه های مارکوف..... ۱۳

۲-۳-متغیر های تصادفی شرطی..... ۱۴

۳-۳-الگوریتم میدان متوسط..... ۱۴

فصل چهارم..... ۱۷

ابزار مورد استفاده و پیاده سازی..... ۱۷

۱-۴-ابزار مورد استفاده..... ۱۸

۲-۴-پیاده سازی..... ۱۸

۳-۴-کد پیاده سازی شده..... ۱۹

فصل پنجم.....	۲۴
نتایج و کار های آتی.....	۲۴
۱-۵-نتایج.....	۲۵
۲-۵-کار های آتی.....	۲۹
مراجع.....	۳۱

فهرست شکل

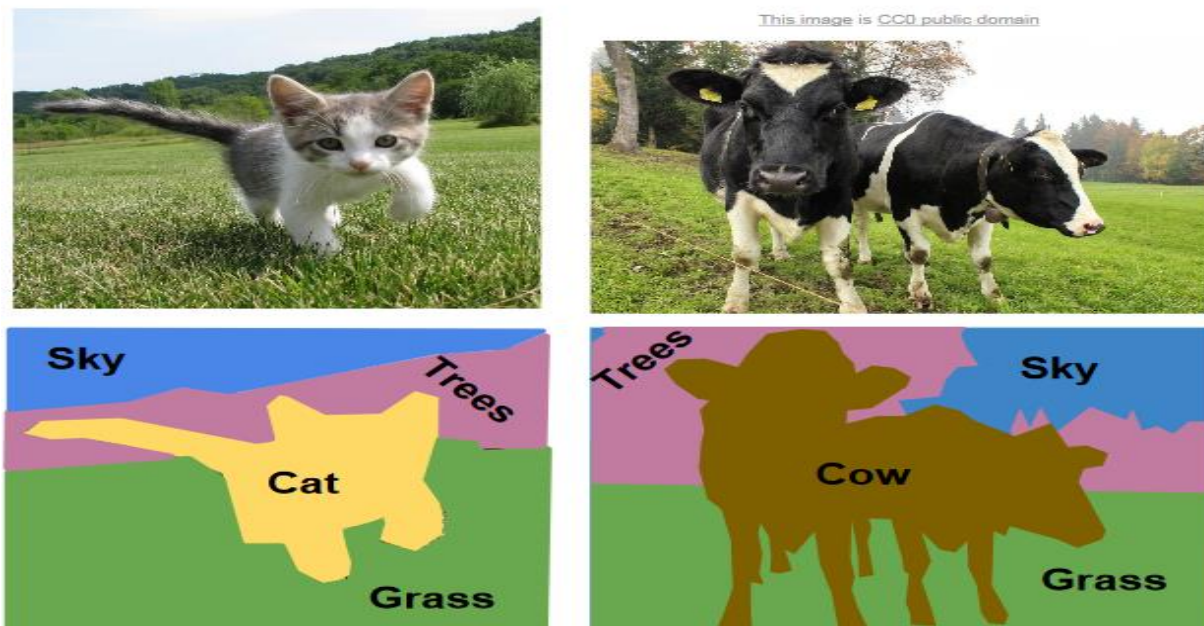
- شکل ۱-۱- گاو و گربه ی بر روی چمن قطعه بندی شده ۲
- شکل ۱-۲- خانه ی قطعه بندی شده ۳
- شکل ۱-۳- دو گاو در کنار هم قطعه بندی شده ۴
- شکل ۱-۳- مدل بیزین ۱۲
- شکل ۲-۳- مدل بیزین همراه با احتمالات ۱۳
- شکل ۱-۵- قطعه بندی کودک در ۰,۱۶۳۳ ثانیه ۲۵
- شکل ۲-۵- قطعه بندی غذاها در ۰,۱۷۳۴ ثانیه ۲۵
- شکل ۳-۵- قطعه بندی تلویزیون در ۰,۱۵۵۲ ثانیه ۲۶
- شکل ۴-۵- قطعه بندی صندلی در ۰,۱۳۵۸ ثانیه ۲۶
- شکل ۵-۵- قطعه بندی ماشین پشت درخت در ۰,۱۹۷۵ ثانیه ۲۷
- شکل ۶-۵- قطعه بندی ماشین در ۰,۱۶۹۴ ثانیه ۲۷
- شکل ۷-۵- قطعه بندی کودک بر روی صندلی در ۰,۴۱۷۵ ثانیه ۲۸
- شکل ۸-۵- قطعه بندی سوار و سوارکار در ۰,۳۷۳۳ ثانیه ۲۸
- شکل ۹-۵- قطعه بندی دو صندلی در کنار هم در ۰,۲۷۹۳ ثانیه ۲۹

فصل اول

مقدمه

امروزه با استفاده روز افزون از کامپیوتر ها در همه جنبه‌های زندگی ما انسان ها برای راحتی کارهای قابل انجام توسط ما، و با پیشرفت روز افزون علم یادگیری ماشین^۱ ما توانسته ایم در علم بینایی ماشین^۲، تحولی شگرف انجام دهیم و با دادن عکس به کامپیوتر آن عکس را به تکه‌های مشخصی تقسیم کنیم.

با استفاده از روش فوق الذکر می توان عکس های پزشکی مانند ECG، عکس های ماهواره‌ای، عکس هایی که از خیابان گرفته می شود، تشخیص صورت و اثر انگشت را به قطعه‌های با معنا تقسیم کرد و از آن ها استفاده نمود. این قطعه های با معنا که شامل مجموعه ای از پیکسل ها هستند، به کلاس های مشخص تقسیم می شوند به عنوان مثال عکس های زیر نمونه ای از تقسیم بندی معنایی می باشد [۱] [۲]:



شکل ۱-۱- گاو و گربه ی بر روی چمن قطعه بندی شده

^۱ Machine learning

^۲ Computer Vision



(a) Example Image



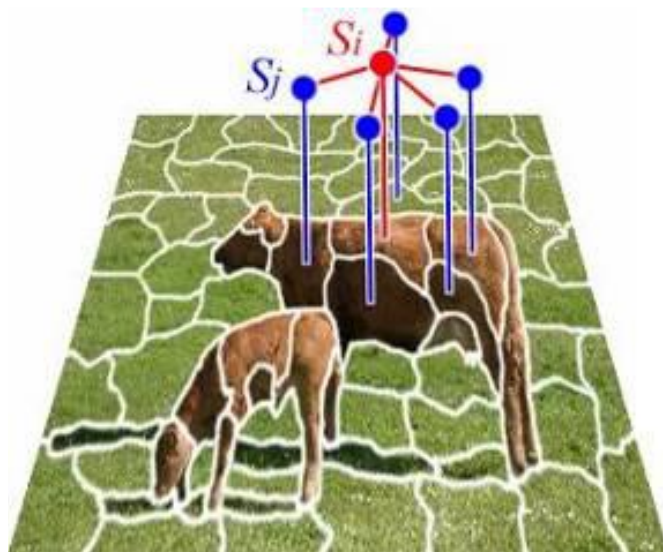
(b) Universal classification ($\arg \max_i \tilde{D}(i|f, \alpha)$)

شکل ۱-۲- خانه ی قطعه بندی شده

پس به بیان دیگر ما بر آنیم که پیکسل هایی را بیابیم که ویژگی های یکسانی را با یکدیگر به اشتراک می گذارند. اما همان طور که می دانیم این عملیات منابع و زمان زیادی را مصرف می کند. به همین علت محققان بر آنند که روشهای خودکار موجود را بهبود ببخشند.

نتایج قطعه بندی معنایی^۳ معمولاً به صورت قطعه هایی است که تمامی عکس را پوشش داده، یا مجموعه ای از خطوط تراز و جدا کننده که بخش های مختلف را از یکدیگر جدا کرده، و بخش های یکسان را با یک مشخصه مشخص می کند. البته باید توجه داشت که امکان این وجود دارد که دو بخش یکسان در نزدیکی هم قرار داشته باشند. به عنوان مثال می توان دو گاو را در کنار همدیگر داشت. جدا کردن این بخش ها یکی از چالش هایی بوده که همواره برای ما مطرح بوده است.

^۳ Image Segmentation



شکل ۱-۳- دو گاو در کنار هم قطعه بندی شده

فصل دوم

قطعه بندی معنایی تصاویر

امروزه چندین روش و الگوریتم و تکنیک برای قطعه بندی معنایی تصاویر شکل گرفته شده است. از این روش ها به اضافه ی اطلاعات مربوط به محیط مسئله مورد نظر استفاده می شود تا به بهترین و سریع ترین حالت ممکن جوابی قابل قبول به دست آورد. در این جا چند روش رایج را بررسی می کنیم.

۲-۱- روش حد آستانه^۴

در این روش که بر اساس انتخاب یک حد آستانه است. یک عکس خاکستری را به یک عکس باینری تبدیل می کند. همچنین یک نوع آستانه گذاری بر اساس هیستوگرام متوازن وجود دارد. بایستی توجه داشت که می توان مقادیر آستانه انتخاب نمود و لزوما تصاویر باینری نخواهند بود. همچنین انتخاب حدود آستانه در این روش بسیار مهم می باشد. به طور معمول از واریانس، آنتروپی و یا به صورت تجربی این اعداد انتخاب می شوند [۳]. البته روش های جدیدی بر اساس سیستم های فازی چند لایه و منطق فازی و ژنتیک نیز به روی کار آمده اند.

۲-۲- روش های مبتنی بر خوشه بندی^۵

یکی از الگوریتم های رایج برای قطعه بندی تصاویر، k-mean که الگوریتمی تکرار شونده است، می باشد. این الگوریتم به این صورت می باشد که k مرکز برای خوشه ها به صورتی تصادفی و یا اکتشافی انتخاب می کنیم. سپس به هر پیکسل یک خوشه نسبت می دهیم به قسمی که فاصله مرکز خوشه و پیکسل از هم کمینه شود. سپس مرکز خوشه ها را به میانگین از همه ی اعضای خوشه انتخاب می کنیم. این عملیات را آنقدر تکرار می کنیم تا همگرایی حاصل گردد. در مسئله ی قطعه بندی معنایی تصاویر این فاصله به طور عام شامل فاصله ی فضایی و فاصله ی شدت روشنایی و فاصله ی شدت رنگ و ترکیبی وزن دار از این عامل هاست. کیفیت این راه حل به شرایط اولیه ی خوشه ها و انتخاب مقدار k بسیار وابسته است.

۲-۳- قطعه بندی حرکتی و تعاملی^۶

قطعه بندی مبتنی بر حرکت تکنیکی مبتنی بر پایه حرکت شی در تصویر است. ایده ی اصلی بسیار ساده می باشد. بدین صورت که به تفاوت های یک جفت از تصاویر نگاه می کنیم. و تفاوت بین دو عکس ناحیه ی اشغال شده توسط شی است. این روش توسط kenney برای قطعه بندی تعاملی بهبود یافت [۴].

^۴ Thresholding

^۵ Clustering Method

^۶ Motion and Interactive Segmentation

۲-۴-قطعه بندی مبتنی بر فشرده سازی^۷

قطعه بندی مبتنی بر فشرده سازی بر این اساس استوار است که بهینه ترین قطعه بندی ممکن آن است که طول داده ذخیره سازی را بر روی تمامی داده ها کمینه کند. این روش بر این ارتباط متکی است که ما در قطعه بندی دنبال آن هستیم الگویی را بیابیم که از آن برای کم کردن حجم داده ها می توان استفاده کرد. در این روش هر بخش را با بافت و شکل مرز آن می توان توصیف کرد و هر یک از این اجزا را می توان با یک تابع توزیع احتمال مدل کرد [۵].

۲-۵-قطعه بندی مبتنی بر هیستوگرام

روش های مبتنی بر هیستوگرام در بین روش های کلاسیک نسبتاً از بقیه ی روش ها سر آمد تر هستند. در این روش ها به طور معمول از قله ها و دره های هیستوگرام برای پیدا کردن محل کلاس ها استفاده می شود. در این تکنیک شدت روشنایی و رنگ معیار سنجش ما می باشد. بر روی این روش نیز اصلاحاتی در جهت بهبود اعمال شده است. به عنوان مثال خوشه بندی عکس بر اساس جست و جو در هیستوگرام تا آنجا که دیگر نتوان خوشه ای یافت. این خوشه بندی که تصویر را برای یافتن کلاس های کوچکتر بهبود می بخشد، نقطه ضعفی دارد که دشوار بودن پیدا کردن محل دقیق قله ها و فرود هاست [۶]. این نوع بخش بندی در هنگام تعدد فریم ها کاربرد بالایی دارد. این روش برای یافتن اجسام متحرک در ویدیوها بسیار رایج می باشد.

۲-۶-تشخیص لبه^۸

از آنجایی که لبه ها با قطعه ها بسیار رابطه ی نزدیکی دارند و از آنجایی که یافتن و تشخیص لبه یک عملیات بسیار کارشده در پردازش تصویر است می توان لبه ها را برای قطعه بندی استفاده کرد. برای یافتن لبه ها معمولاً از این موضوع استفاده می شود که در لبه ها تغییرات شدت رنگ زیاد است. پس با کرنل های مختلف می توان آنها را با دقت خوبی یافت. اگرچه یافتن لبه ها آسان است اما در یافتن لبه ها احتمال آن وجود دارد که تغییرات رنگ نا پیوسته باشد و در لبه ها ناپیوستگی دیده شود. در حالیکه ما برای قطعه بندی تصاویر به محیط هایی بسته نیازمندیم. دو شخصیت به نام های Tony Lindberg و Meng-Xiang Li از لبه ها برای ساخت روشی مجتمع برای پیدا کردن قطعه های با معنا استفاده کرده اند [۷].

^۷ Compression-based methods

^۸ Edge detection

۷-۲- خوشه بندی دوگانه^۹

این روش براساس سه ویژگی تصاویر بنا شده است. قسمت کردن تصاویر بر اساس آنالیز هیستوگرام که توسط فشردگی بالای خوشه ها و گرادیان مرز های کلاس ها انجام می گردد. برای این منظور بایستی دو فضا معرفی کنیم: فضای یک بعدی هیستوگرام روشنایی ($H=H(B)$)، فضای سه بعدی تصویر اصلی ($B=B(x, y)$). فضای اول به ما اجازه می دهد که مقدار پراکندگی روشنایی یک تصویر را توسط روش های خوشه بندی محاسبه کنیم. یک حد آستانه توسط این روش خوشه بندی انتخاب می شود که با آن یک نظیر رنگی که از حدمان بیشتر یک و از حدمان کمتر صفر باشد انتخاب می کنیم. این نظیرمان میزان فشردگی و مقدار توزیع روشنایی را برای اشیا و پس زمینه مشخص میکند. با استفاده از همه ی حدود به دست آمده می توان قطعه بندی معنایی را انجام داد.

۸-۲- ناحیه ی در حال رشد^{۱۰}

این روش که روشی بسیار نوپز پذیر است از این اصل موضوع استفاده می کند که پیکسل های مجاوری که معیار های شباهت را می گذرانند به یک کلاس تعلق دارند. به همین دلیل انتخاب معیار های شباهت در این روش بسیار مهم و تاثیر گذار می باشد.

روش های ادغام ناحیه های آماری با ساخت یک گرافی از پیکسل ها با چهار ارتباط با یال های وزن دار شده توسط قدرمطلق تفاوت شدت رنگ و روشنایی شروع می کند [۸]. سپس این روش ها هر پیکسل را یک قطعه تصور می کند سپس یال ها را بر اساس صفوف اولویت مرتب نموده و با توجه به پیش بینی های آماری و احتمالاتی تصمیم می گیرد که هر ارتباط را به آن پیکسل برای آن قطعه اضافه بکند یا خیر.

یک از روش ها که روش بذر دار نام دارد. این کار را می توان با چندین پیکسل به عنوان ورودی انجام می دهد. آن پیکسل ها با مجاورت خود بر اساس شدت روشنایی مقایسه می شوند. و در صورت شباهت دو ناحیه مجاور به هم الحاق می شوند. بایستی توجه داشت که از میانگین شدت روشنایی و رنگ هر منطقه به عنوان کلید کاندید آن منطقه برای مقایسه با مجاورت خود انتخاب می شود.

روش دیگر آن که روش بی بذر نام دارد. نیازمند ورودی اولیه به الگوریتم نمی باشد. در این روش که شباهت بسیاری به روش بالا دارد. به این صورت می باشد که ابتدا یک پیکسل را در نظر گرفته و به آن ناحیه را انتساب

^۹ Dual clustering Method

^{۱۰} Region-growing Methods

می کنیم سپس از آن پیکسل طبق روش بالا عمل کرده و در صورت عدم شباهت یک ناحیه ی جدید به آن پیکسلی که شباهت نداشت انتساب کرده و برای آن پیکسل هم همین عملیات را انجام می دهیم.

۲-۹- قطعه بندی مبتنی بر معادله دیفرانسیل مشتقات جزئی

با استفاده از روش های مبتنی بر معادلات مشتقات جزئی می توان تصویر را با حل عددی بر روی آنها قطعه بندی نمود. منحنی انتشار یک روش محبوب در این زمینه با برخورداری از استفاده های زیاد در استخراج اشیا، ردیابی اشیا، بازسازی استریو می باشد. ایده اصلی حول این محور می چرخد که منحنی ابتدایی را به کمترین مقدار تابع هزینه اش سوق دهیم [۹]. همانند اکثر حل مسائل معکوس کمینه کردن مقدار تابع هزینه نیازمند ارضا کردن محدودیت های مشخصی از جمله محدودیت های هندسی است. سه روش پارامتری، مجموعی سطح، راهپیمایی سریع از روش هایی هستند که در قطعه بندی معنایی استفاده می شوند.

۲-۱۰- قطعه بندی توسط واریانس

هدف غایی روش واریانسی پیدا کردن قطعه بندی است که بهینه ترین قطعه بندی ممکن با توجه به یا تابع انرژی مشخص است [۱۰]. این تابع همواره شامل یک بخش برای فیت کردن داده ها و یک بخش رگولارایز کردن است. یک روش کلاسیک استفاده از مدل پاتز می باشد.

۲-۱۱- مدل گراف های احتمالاتی^{۱۱}

مدل های گرافی احتمالاتی تاثیر خود را در قطعه بندی معنایی تصاویر نشان داده اند [۱۱]. در این مدل ها بخش را به بخش های دیگر ارتباط می دهند. هر بخش یک گره و هر ارتباط یک یال محسوب می شود. یال ها در این گراف ها معمولاً وزن دار می باشند. در فصل بعدی اطلاعاتی کامل تر راجع به این مدل ها به دست می آوریم. روش مورد استفاده ی ما که در این مجموعه کاربرد دارد.

۲-۱۲- قطعه بندی توسط آب پخشانی^{۱۲}

تبدیل آب پخشانی، دامنه گرادیان یک تصویر را به عنوان سطح توپوگرافی در نظر می گیرد. پیکسل هایی که بیشترین مقدار دامنه گرادیان را دارند، به خطوط آب پخشانی نظیر می شوند که این خطوط نمایانگر مرز نواحی خواهند بود. آب در هر ناحیه، که توسط خطوط آب پخشانی محصور شده است، شروع به بالا آمدن می کند تا به

^{۱۱} Probabilistic Graph Model

^{۱۲} Watershed transformation

سطح مینیمم محلی برسد. پیکسل‌هایی که آب در آن‌ها به سطح مینیمم محلی می‌رسد، یک قطعه ی با معنا را مشخص می‌کنند.

۲-۱۳- باقی‌روش‌ها

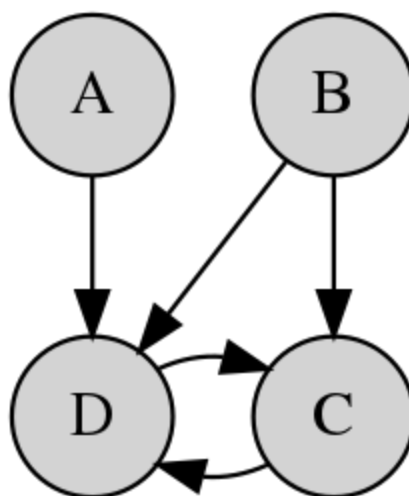
علاوه بر روش‌های ذکر شده، روش‌هایی بر اساس یادگیری ماشین و یادگیری عمیق امروزه رواج پیدا کرده است. به طور معمول از روش‌های بالا برای یافتن ویژگی‌های مناسب نیز برای آموزش دادن ماشین استفاده می‌شود.

فصل سوم

مدل های گرافی احتمالاتی و میدان متوسط

۳-۱-۱- مدل های گرافی احتمالاتی

مدل گراف های احتمالاتی یک مدل احتمالاتی است که بیانگر ساختار وابستگی بین متغیر های تصادفی توسط یک گراف است. مدل های گرافی معمولاً در آمار و احتمالات به طور دقیق تر در آمار بیزی و یادگیری ماشین استفاده می شود.

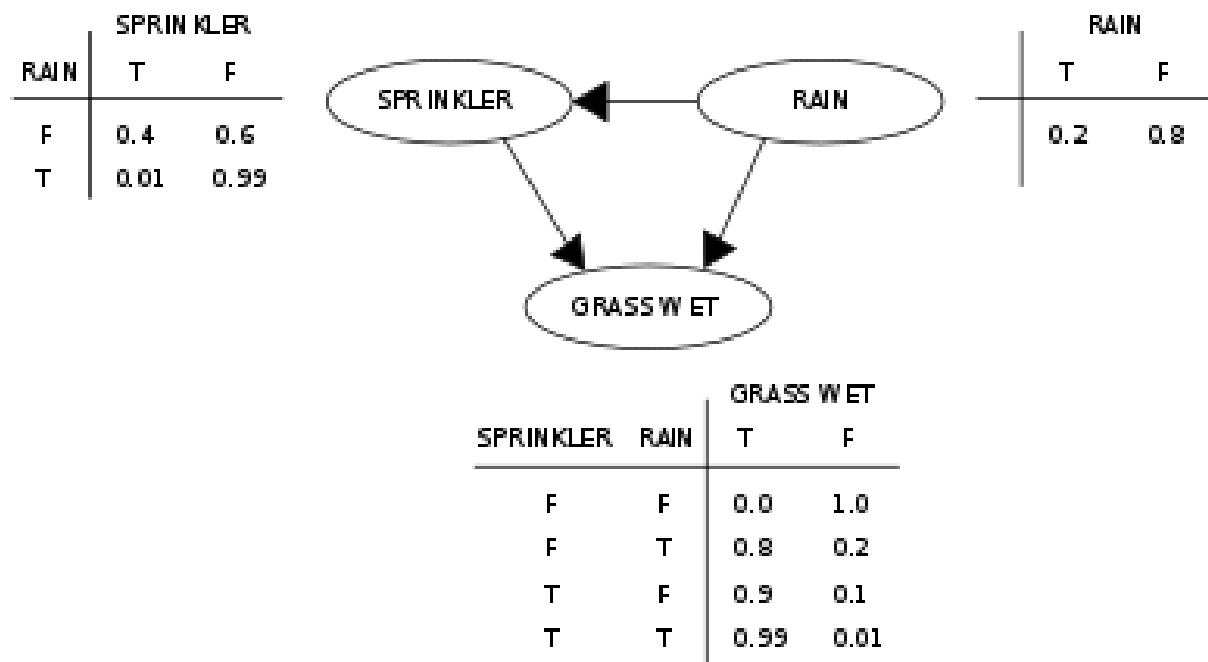


شکل ۳-۱-۱- مدل بیزین

به طور معمول این مدل ها توسط یک گراف نمایش داده می شود. این مدل ها یک توزیع آماری از یک فضای پیچیده و چند بعدی را مدل می کنند. ما با استفاده از فاکتورسازی و ادغام و جمع می توانیم وابستگی های مختلف را مدل کرده و همچنین با تعداد کمی متغیر مدل های بسیار پیچیده را پیاده سازی کنیم.

۳-۱-۱- شبکه های بیزین

این نوع شبکه ها که اساس بسیاری از الگوریتم های یادگیری ماشین می باشند، مدلی است که اگر بتوانیم مدل را فاکتوری از روابط وابستگی بین متغیر های تصادفی بنویسیم می گویند. بایستی توجه داشت که گراف نمایانگر این نوع شبکه ها گرافی جهت دار می باشد. به عنوان مثال رابطه ی بین علائم یک بیماری و آن بیماری از این نوع رابطه است.



شکل ۳-۲-مدل بی‌زین همراه با احتمالات

۳-۱-۲-شبکه‌های مارکوف

یک گراف بی جهت از متغیرهای تصادفی است که خاصیت مارکوف را دارا هستند. خاصیت مارکوف خاصیتی است که به معنی آن است که حالاتی که در آینده پیش می آیند فقط به حالاتی که در حال حاضر سیستم داراست ارتباط دارند. از این شبکه در بینایی ماشین بسیار استفاده شده است. یکی از خصوصیات این نوع شبکه ها فاکتوریزه شدن بر روی کلیک ها می باشد. اگر تابع توزیع را بتوان بر روی کلیک‌ها^{۱۳} روی گراف G فاکتورایز کرد:

$$P(X = x) = \prod_{C \in cl(G)} \phi_C(x_C) \quad (۱-۳)$$

در اینجا $cl(G)$ به معنی مجموعه‌ی کلیک‌های تشکیل دهنده‌ی گراف است. و تابع ϕ_C معنی تابع پتانسیل یا انرژی آن کلیک می باشد [۱۲].

^{۱۳} cliques

۳-۲- متغیرهای تصادفی شرطی

یک روش مدل کردن آماری می‌باشد که در شناسایی الگو و یادگیری ماشین کاربرد دارد. الگوریتم‌های کلاس‌بندی کردن موجود بدون در نظر گرفتن اتفاقات همسایه و متن یک نشان را به یک داده نسبت می‌دهند. اما با این نوع متغیرها ما می‌توانیم با استفاده از پارامتر کم، متن را لحاظ کرده و نتیجه‌گیری کنیم البته بایستی توجه داشت که این نتیجه‌گیری با وجود تعداد پارامتر و متغیر کم بسیار کاری سخت و حجیم می‌باشد. این نوع گراف‌ها گراف‌هایی بی‌جهت و متمایز کننده هستند. این نوع گراف‌ها از آنجایی که رابطه‌ی بین همسایه‌ها را لحاظ می‌کند بسیار کاربرد زیادی در قطعه‌بندی معنایی تصاویر دارند. در این نوع مدل ما دو نوع متغیر داریم: مشاهده شده و خروجی، که ما بر اساس متغیرهای مشاهده شده تصمیم‌گیری انجام می‌دهیم که خروجی مان چه باشد. تصمیم‌گیری‌ها بر اساس الگوریتم‌های پاس دادن پیغام انجام می‌شود. که همان طور که ذکر شد نتیجه‌گیری دقیق تقریباً کاری غیر ممکن می‌باشد پس از الگوریتم‌های تقریبی برای نتیجه‌گیری استفاده می‌شوند. روش mean field نیز در این زمره قرار می‌گیرد. در این روش از این ایده ساده استفاده می‌شود که بایستی تابع توزیعی نزدیک به تابع توزیع اصلی تقریب بزنیم. این تابع توزیع بایستی از خانواده‌ی قابل محاسبه باشد و این تقریب بایستی تا آنجا که می‌تواند دقت داشته باشد پس نیازمند آن است تا اختلاف دو تابع را بهینه کند.

۳-۳- الگوریتم میدان متوسط

برای انجام دادن الگوریتم میدان متوسط قدم‌های زیر را دنبال می‌کنیم:

۱- در ابتدا یک ارزیاب فاصله برای شباهت بین تابع توزیعی که قرار است حدس بزنیم و تابع توزیع اصلی برقرار می‌کنیم.

۲- یک کلاس احتمالاتی که علاقه داریم مدلمان شبیه آن باشد انتخاب می‌کنیم.

۳- یک Q که فاصله اش از P کمینه شود انتخاب می‌کنیم.

۴- Q را پیدا می‌کنیم.

پس از مشخص شدن الگوریتم که طبق بالا بود یک معیار نزدیکی که KL divergence است انتخاب می‌کنیم. این معیار به صورت زیر تعریف می‌شود:

$$D_{KL}(Q||P) = \sum_{x \in L^n} Q(x) \log \left(\frac{Q(x)}{P(x)} \right) \quad (۲-۳)$$

$$= - \sum_{x \in L^n} Q(x) \log(P(x)) + \sum_{x \in L^n} Q(x) \log(Q(x)) \quad (۳-۳)$$

با فرض آنکه P از نوع تابع احتمالاتی گیبز است، و از آنجایی که بیانگر تابع احتمالاتی گیبز به صورت فرمول

(۴-۳) مدل می شود فرمول به صورت فرمول (۵-۳) تغییر میکند:

$$P(x) = \frac{1}{Z} e^{-E(x)} \quad (۴-۳)$$

$$D_{KL}(Q||P) = - \sum_{x \in L^n} Q(x) \log \left(\frac{1}{Z} e^{-E(x)} \right) + \sum_{x \in L^n} Q(x) \log(Q(x)) \quad (۵-۳)$$

با توجه به این که جمع روی Q که یک توزیع احتمالاتی است یک می شود فرمول به صورت زیر میشود:

$$= \sum Q(x)E(x) + \log(Z) + \sum Q(x)\log(Q(x)) \quad (۶-۳)$$

پس در نهایت تابعی که باید مینیمم شود:

$$F(P, Q) = \sum Q(x)E(x) + \sum Q(x)\log(Q(x)) \quad (۷-۳)$$

که به معنی امید ریاضی و آنتروپی است.

با باز کردن بخش اول به :

$$\sum_{x \in L^n} Q(x)E(x) = \sum_{x \in L^n} Q(x) \sum_{c \in C} E_c(x_c) = \sum_{c \in C} \sum_{x \in L^n} Q(x)E_c(x_c) \quad (۸-۳)$$

می رسیم که با پخش کردن بر روی کلیک ها به این صورت می توان معنا کرد که :

$$\sum_{x \in L^n} Q(x)E_c(x_c) = \sum_{x_c} \sum_{x_{c'}} Q(x)E_c(x_c) = \sum_{x_c} E_c(x_c) \sum_{x_{c'}} Q(x) \quad (۹-۳)$$

این عمل احتمالاتی شبیه مارجینال است:

$$\sum_{x \in L^n} Q(x) E(x) = \sum_{c \in C} \sum_{x_c} E_c(x_c) Q(X_c = x_c) \quad (10-3)$$

و این عمل شبیه اعمال یک فیلتر می باشد که در نهایت فرمول به این صورت می شود:

$$Q_i(x_i = l) = \frac{1}{Z_i} e^{-\Psi_u(x_i) - \sum_{l' \in L} \mu(l, l') \sum_{m=1}^K w_m \sum_{j \neq i} k_m(f_i, f_j) Q_j(l')} \quad (11-3)$$

که به صورت زیر فرموله می شود [۱۳]:

- › Initialize $Q_j(l') = \frac{1}{Z_i} e^{-\varphi(x_i)_u}$
 - While not converged
 - Message passing: $Q_i(l) = \sum_{j \neq i} k_m(f_i, f_j) Q_j(l)$
 - Compatibility transform: $\widehat{Q_i(x_i)} = \sum_{l \in L} \mu(x_i, l) \sum_m w_m Q_i(l)$
 - Local update: $Q_i(x_i) = e^{-\Psi(x_i)_u - \widehat{Q_i(x_i)}}$
 - Normalize: $Q_i(x_i)$

فصل چهارم

ابزار مورد استفاده و پیاده سازی

۴-۱- ابزار مورد استفاده

زبان برنامه نویسی استفاده شده در این تحقیق python می باشد. این زبان که زبانی رایج در علوم داده ای است، دارای کتابخانه هایی غنی برای راحتی استفاده می باشد.

کتابخانه های استفاده شده در این تحقیق Numpy و Matplotlib و Scipy می باشد.

کتابخانه ی Numpy کتابخانه ای است که برای استفاده و کار با ماتریس ها ساخته شده است. این کتابخانه محاسبات را با استفاده از الگوریتم های بهینه و با استفاده از خاصیت های GPU و CPU بهبود می بخشد.

کتابخانه ی Matplotlib برای به تصویر کشیدن اجسام بسیار مناسب است. این کتابخانه قابلیت های زیادی از جمله نشان دادن چندین تصویر در یک فرم را داراست.

کتابخانه ی Scipy که کتابخانه ای بسیار رایج برای کار با داده ها می باشد قابلیت خواندن و نوشتن فایل های PPM را دارد که برای ما مورد استفاده بود.

۴-۲- پیاده سازی

با توجه به این موضوع که برای پیاده سازی این الگوریتم ابتدا نیازمند یک میدان شرطی کاملاً بهم متصل هستیم در ابتدا کلاس آن را ایجاد می کنیم. در این کلاس ما تعداد متغیر ها و پارامتر های موجود و تعداد کلاس های موجود را گرفته و مشخص می کنیم که دو نوع تابع داریم.

دو نوع تابع ما تابع دوگانی^{۱۴} و یک گانی^{۱۵} می باشند. تابع دوگانی نمایانگر رابطه ی بین گره ها با همدیگر می باشد. رابطه ی تک گانی بیانگر احتمال آنکه یک گره چه چیزی است می باشد.

ما دو نوع تابع دو گانی خواهیم داشت. یک نوع تابع دوگانی بدون توجه به رنگ و فاصله که در اصل همان فیلتر گاوسی ساده است. نوع دوم تابع دوگانی با توجه به فاصله و اختلاف رنگ دو گره از گراف می باشد.

برای استنتاج ابتدا طبق فرمول که اثبات شده مقدار منفی و نرمالایز شده ی مقادیر را به Q نسبت می دهیم. سپس به تعدادباری که منجر به همگرا شدن تابع شود باید الگوریتم میدان متوسط را اجرا کنیم اما مشکل اساسی

^{۱۴} Pairwise potential function

^{۱۵} Unary potential function

آن است که نمی توان همگرایی تابع را مشخص کرد. اما طبق تجربه می دانیم که در بیشترین مقدار پس از ۱۰ بار اجرای الگوریتم ما به همگرایی می رسیم. و برای بخش محاسبات کافیست که هر فیلتر را اعمال کنیم و با یکدیگر جمع کنیم طبق فرمولمان که پس از پایان استنتاج اگر نمایش دهیم نتایج بخش نتایج به دست می آید.

۴-۳-کد پیاده سازی شده

```
1. from scipy.misc
2. import imread

3. import matplotlib.pyplot as plt

4. import numpy as np
   from numbers
5. import Number

   class DenseCRF:
   def __init__(self, nvar, nlabels):
       self.nvar = nvar
       self.nlabels = nlabels
       self.pair_wise_function = None
       self.unary_function = None

       def setUnaryEnergy(self, unary):
           print("unary setted")
           self.unary_function = unary

       def addPairwiseEnergy(self, pair, compat):
           print("added")
           if self.pair_wise_function is None:
               self.pair_wise_function = pair

6. else :
       self.pair_wise_function = np.concatenate((self.pair_wise_function, pair), 0)

       def exp_normalizer(self, inp):
           print("initializing")
           temp = np.subtract(inp, np.max(inp, 0))
           print(temp)
           print(temp.shape)
           tempo = np.exp(temp)
           normalized = np.divide(tempo, np.sum(tempo))
           return normalized

       def inference(self, n_step):
           print("start inferring:")
           Q = self.exp_normalizer(-self.unary_function)
           print(Q.shape)
           print(Q)
           print(self.pair_wise_function.shape)
           for i in range(n_step):
               temp1 = -self.unary_function

7. for k in range(self.pair_wise_function.shape[0]):
           temp2 = self.apply(Q, self.pair_wise_function[k])
           temp1 = np.subtract(temp1, temp2)
```

```

    Q = self.exp_normalizer(temp1)
    return Q

    def apply(self, Q, pair_row):
    print("apply")
    print(Q.shape)
    print(pair_row.shape)
    Q_bar = np.multiply(Q, pair_row)
    print(Q_bar)
    print(Q_bar.shape)
    print("end of applying")
    return Q_bar

    root = 'C:/Users/Hamed Khashehchi/Downloads/Compressed/show me bobs/densecrf/densecrf/example
s/'
8. drat = 'C:/Users/Hamed Khashehchi/Desktop/kha/'
9. root = drat
    number_classes = 21
    ground_truth_probability = 0.5

    def get_color(pixel): return pixel[0] + 256 * pixel[1] + 256 * 256 * pixel[2]

    def zero_list_maker(n):
    list_of_zeros = [0] * n

10. return list_of_zeros

    def unary_from_labels(labels, n_labels, gt_prob, zero_unsure = True):
    assert 0 < gt_prob < 1, "`gt_prob must be in (0,1).`"

11. labels = labels.flatten()

    n_energy = -np.log((1.0 - gt_prob) / (n_labels - 1))
    p_energy = -np.log(gt_prob)

    unary_function = np.full((n_labels, len(labels)), n_energy, dtype = 'float32')
    unary_function[labels - 1]
12.     if zero_unsure
13.     else labels, np.arange(unary_function.shape[1]) = p_energy

14. if zero_unsure:
    unary_function[:, labels == 0] = -np.log(1.0 / n_labels)

    return unary_function

    def create_pairwise_gaussian(sdims, shape):
    #create mesh
    print("gaussian")
    print(shape)
    hcoord_range = [range(s) for s in shape]

```

```

    print(hcord_range)
    mesh = np.array(np.meshgrid( * hcord_range, indexing = 'ij'), dtype = np.float32)

    # scale mesh accordingly

15. for i, s in enumerate(sdims):
    print(i)
    print(s)
    mesh[i] /= s

16. return mesh.reshape([len(sdims), -1])

def create_pairwise_bilateral(sdims, schan, img, chdim = -1):
    #Put channel dim in right position
    print("bilateral")
    if chdim == -1:
    #We don 't have a channel, add a new axis
        im_feat = img[np.newaxis].astype(np.float32)
    else:
        # Put the channel dim as axis 0, all others stay relatively the same
        im_feat = np.rollaxis(img, chdim).astype(np.float32)

    # scale image features per channel
    # Allow for a single number in `schan` to broadcast across all channels:
    if isinstance(schan, Number):
        im_feat /= schan
    else:
        for i, s in enumerate(schan):
            im_feat[i] /= s

    # create a mesh
    cord_range = [range(s) for s in im_feat.shape[:4]]
    mesh = np.array(np.meshgrid(*cord_range, indexing='
17. ij '), dtype=np.float32)

    # scale mesh accordingly
    for i, s in enumerate(sdims):
        mesh[i] /= s

    feats = np.concatenate([mesh, im_feat])
    return feats.reshape([feats.shape[0], -1])

image = imread(root+' / im6.ppm ')
print(image.shape)
height_image, width_image, channel_image = image.shape
annotation = imread(root+' / anno6.ppm ')
print(annotation.shape)
print(annotation.dtype)
height_annotation, width_annotation, channel_annotation = annotation.shape
annotation_label = annotation[:, :, 0] + (256*annotation[:, :, 1]) + (256*256*annotation[:, :,
1])
print(annotation_label)
colors, labels = np.unique(annotation_label, return_inverse=True)
is_there_anything_unknown = 0 in colors
print(colors)
print(labels)

```



```

print(is_there_anything_unknown)
if is_there_anything_unknown:
    colors = colors[:i]
color_table = np.empty((len(colors), 3))
color_table[:, 0] = (colors & 0x0000FF)
color_table[:, 1] = (colors & 0x00FF00) / 256
color_table[:, 2] = (colors & 0xFF0000) / (256*256)
n_labels = len(set(labels.flat)) - int(is_there_anything_unknown)
print(n_labels, " labels", (" plus \"unknown\" 0: " if is_there_anything_unknown else ""), set
(labels.flat))

fig, axs = plt.subplots(1, 2)
axs[0].set_title('
18. Image: ')
axs[1].set_title('
19. Annotation: ')
axs[0].imshow(image)
axs[1].imshow(annotation)
plt.show()

print("done showing off")

d = DenseCRF(width_image*height_image, n_labels)

unary_functions = unary_from_labels(labels, n_labels, gt_prob=0.666, zero_unsure=is_there_anyt
hing_unknown)
d.setUnaryEnergy(unary=unary_functions)

print(unary_functions)
print(unary_functions.shape)
feats = create_pairwise_gaussian(sdims=(3, 3), shape=image.shape[:2])
print(feats)
print(feats.shape)
d.addPairwiseEnergy(pair=feats, compat=3)

print("popopopop")
print(d.pair_wise_function.shape)

feats = create_pairwise_bilateral(sdims=(80, 80), schan=(13, 13, 13), img=image, chdim=2)
print(feats)
print(feats.shape)
d.addPairwiseEnergy(pair=feats, compat=10)

print("popopopop")
print(d.pair_wise_function.shape)

Q = d.inference(n_step=10)
print("end of inference")
print(Q)
print(Q.shape)

MAP = np.argmax(Q, axis=0)
MAP = color_table[MAP, :]
MAP = MAP.reshape((height_image, width_image, channel_image))
print(MAP)
print(MAP.shape)

```

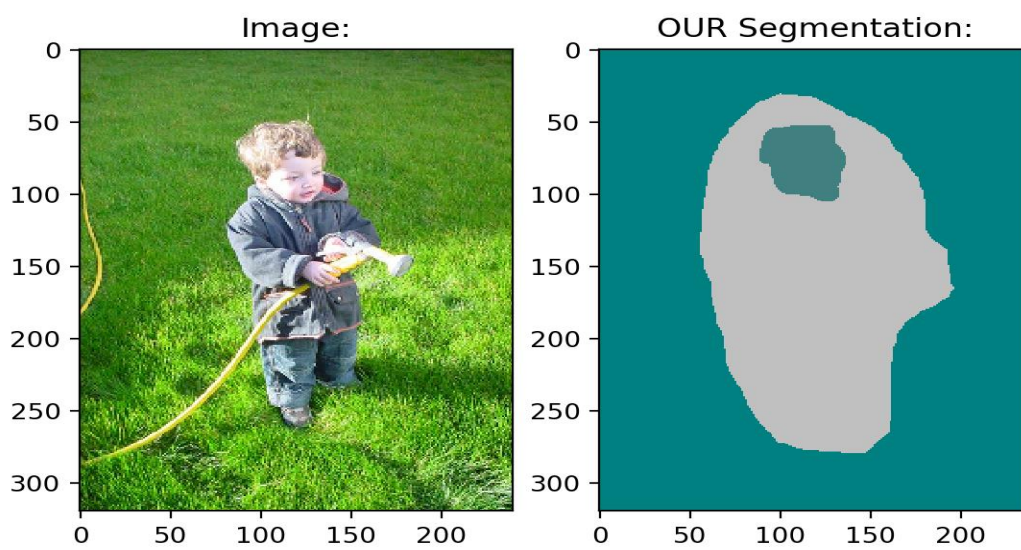
```
fig, axs = plt.subplots(1, 2)
axs[0].set_title('
20. Image: ')
axs[1].set_title('
21. OUR Segmentation: ')
axs[0].imshow(image)
axs[1].imshow(MAP, '
22. Blues_r ')
plt.show()
```

فصل پنجم

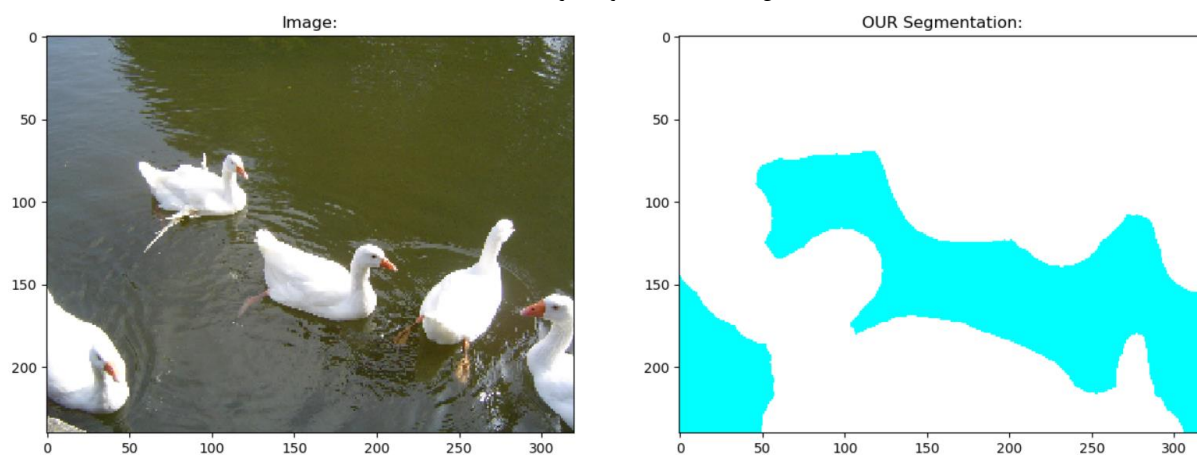
نتایج و کارهای آتی

۵-۱-نتایج

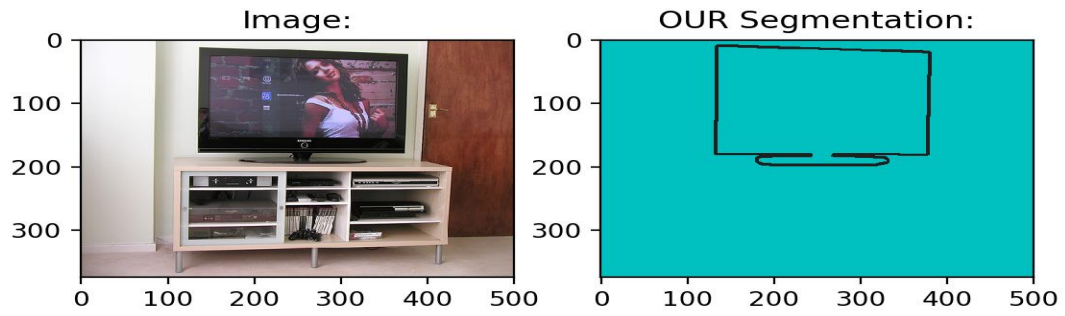
پس از اجرای برنامه و گرفتن خروجی جواب عکس های مختلف به صورت زیر می باشد، به بیان دیگر ما عکس را به شکل روبرویش قطعه بندی می کنیم:



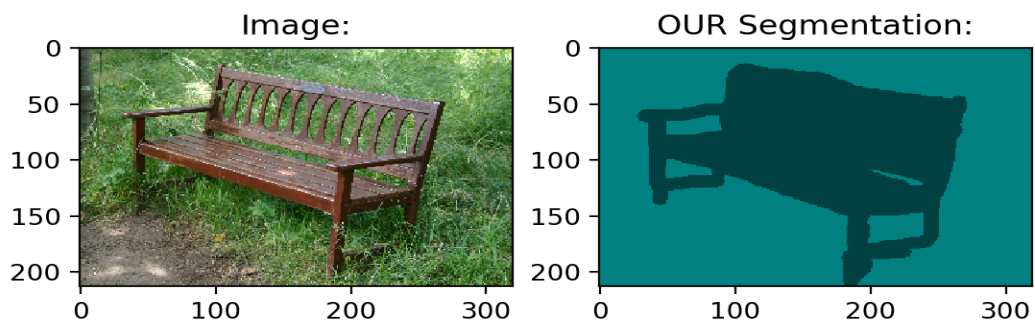
شکل ۵-۱- قطعه بندی کودک در ۰,۱۶۳۳ ثانیه



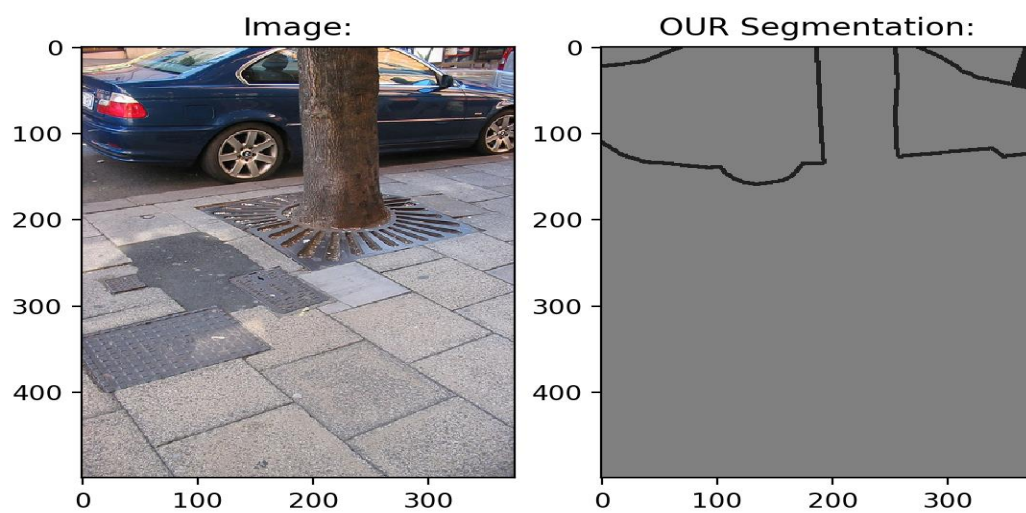
شکل ۵-۲- قطعه بندی غازها در ۰,۱۷۳۴ ثانیه



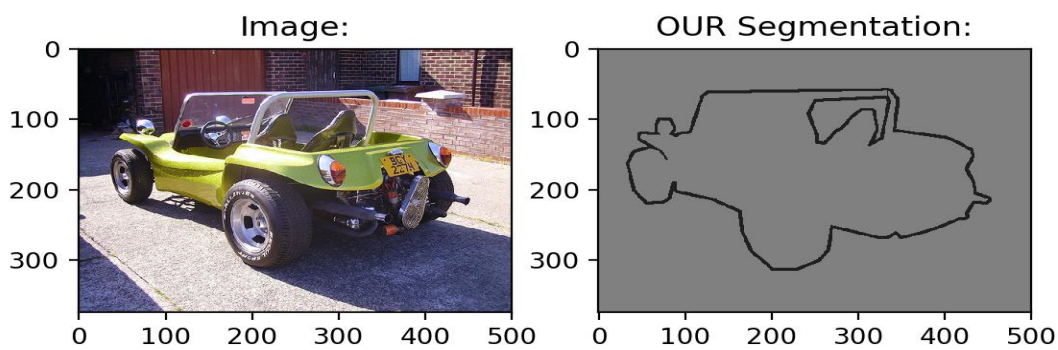
شکل ۵-۳- قطعه بندی تلویزیون در ۰,۱۵۵۲ ثانیه



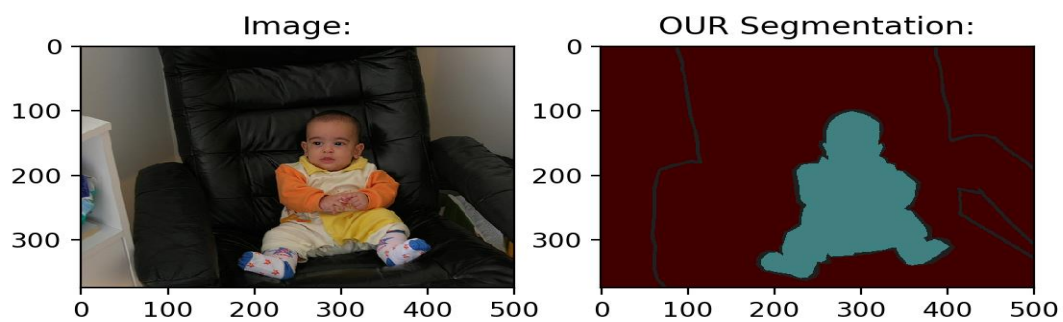
شکل ۵-۴- قطعه بندی صندلی در ۰,۱۳۵۸ ثانیه



شکل ۵-۵- قطعه بندی ماشین پشت درخت در ۰,۱۹۷۵ ثانیه



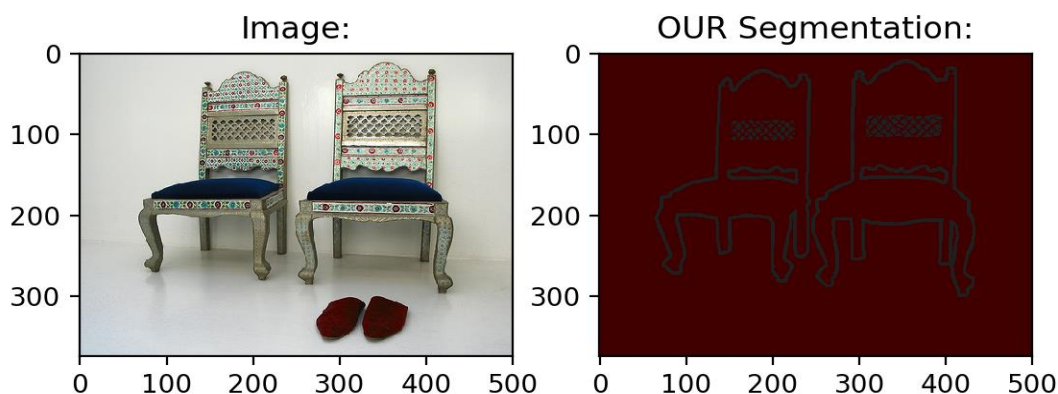
شکل ۵-۶- قطعه بندی ماشین در ۰,۱۶۹۴ ثانیه



شکل ۵-۷- قطعه بندی کودک بر روی صندلی در ۰,۴۱۷۵ ثانیه



شکل ۵-۸- قطعه بندی سوار و سوارکار در ۰,۳۷۳۳ ثانیه



شکل ۵-۹- قطعه بندی دو صندلی در کنار هم در ۰.۲۷۹۳ ثانیه

همان طور که مشاهده می کنیم عکس ها با سرعت بسیار زیاد، و با دقت به نسبت خوبی قطعه بندی می شوند که بیانگر برتری این روش نسبت به روش های دیگر است. پس با استفاده از استنتاج تقریبی در میدان های شرطی با دقت و سرعت بسیار بالایی می توان قطعه بندی معنایی را انجام داد.

۵-۲- کار های آتی

برای ادامه ی این تحقیق می توان چندین مسئله را ادامه داد:

اولین بهبودی که می توان به این روش داد، بهبود سرعتی می باشد. اگر ما به جای استفاده از کانولوشن از پرمیوتوهیدرال [۱۴] استفاده کنیم، می توانیم با یک انجام نظیر به ابر صفحه های صحیح^{۱۶} سرعت این عملیات را چندین برابر بیشتر کنیم.

بهبود بهتری که به ذهن می رسد استفاده از داده ها برای آموزش یک تابع یکتا برای تک گانی است. و با استفاده از آن داده ها هم چنین می توان تابع های دوگانی بهتری نیز تعبیه نمود.

^{۱۶} Integer hyper planes

یک بهبود دیگر استفاده از مدل های وزن دار و پارامتری برای محاسبات است که غیر خطی بودن مدل را بهتر شبیه سازی می کند.

مراجع

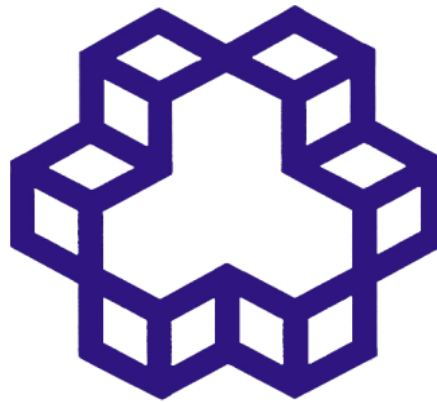
- [١] F. SHAIKH, 30 November 2017. [Online]. This site is Available at this URL: <https://www.analyticsvidhya.com/blog/2017/11/heart-sound-segmentation-deep-learning/>. [Accessed 20 9 2018].
- [٢] C. Hazırbaş, Feature Selection and Learning, 2014.
- [٣] K. J. Batenburg and J. Sijbers, "Optimal Threshold Selection for Tomogram Segmentation by Projection Distance Minimization," IEEE Transactions on Medical Imaging, vol. 28, pp. 676-686, 2009.
- [٤] J. Kenney, T. Buckley and O. Brock, "Interactive Segmentation for Manipulation in Unstructured Environments," in International Conference on Robotics and Automation, Kobe, Japan, 2009.
- [٥] H. Mobahi, S. R. Rao, A. Y. Yang, S. S. Sastry and Y. Ma, "Segmentation of Natural Images by Texture and Boundary Compression," Int J Comput Vis, vol. 95, p. 86–98, 2011.
- [٦] RonOhlander, KeithPrice and D. RajReddy, "Picture segmentation using a recursive region splitting method," Computer Graphics and Image Processing, vol. 8, no. 3, pp. 313-333, 1978.
- [٧] TonyLindeberg and Meng-XiangLi, "Segmentation and Classification of Edges Using Minimum Description Length Approximation and Complementary Junction Cues," Computer Vision and Image Understanding, vol. 67, no. 1, pp. 88-98, 1997.
- [٨] R. Nock and F. Nielsen, "tatistical Region Merging," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 11, pp. 1452-1458, 2004.
- [٩] V. Caselles, R. Kimmel and G. Sapiro, " Geodesic active contours," International Journal of Computer Vision, vol. 2, pp. 61-79, 1997.
- [١٠] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," Communications on Pure and Applied Mathematics, vol. 42, no. 5, pp. 577-685, 1989.
- [١١] S. Geman and D. Geman, "Stochastic relaxation, Gibbs Distributions and Bayesian Restoration of Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 6, no. 6, pp. 721-741, 1984.

- [۱۲] D. Koller and N. Friedman, Probabilistic Graphical Models(Principles and Techniques), MIT Press, 2009.
- [۱۳] P. Krähenbühl and V. Koltun, "Efficient Inference in Fully Connected CRFs with," 2016.
- [۱۴] J. B. M. A. D. Andrew Adams, "Fast High-Dimensional Filtering Using the," The Eurographics Association and Blackwell Publishing Ltd, 2009.

Abstract

Nowadays, pictures are getting used more and more. They are part of our daily lives. Segmenting images is one of the oldest computer vision problem. In this thesis we tackled this problem using Conditional Random Field (CRF). One of the approaches toward modeling the pictures is using fully connected CRF. Having more connections means more accuracy and more time consumption. However, when we are using fully connected CRF we can use an approximation approach for inferring which is called mean field. In this approach our accuracy won't suffer that much; however, time and resources consumed for inferring will decrease a lot.

Keywords: Image segmentation, CRF, fully connected CRF, mean field.



1928

K. N. Toosi University of Technology
Faculty of Computer Engineering

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of Bachelor of Science (B.Sc)
in Computer Engineering.

Implementing Mean field Algorithm for inference in fully connected
CRF in Image Segmentation

By:

Hamed Khashehchi

Supervisor:

Dr. Behrooz Nasihatkon