Key.Net: Keypoint Detection by Handcrafted and Learned CNN Filters

Axel Barroso-Laguna

Edgar Riba

Daniel Ponsa

Krystian Mikolajczyk

{axel.barroso17, k.mikolajczyk}@imperial.ac.uk

{eriba, daniel}@cvc.uab.es

خلاصه:

در این مقاله روش جدید برای تشخیص keypoint معرفی شده است که روش های سنتی و شبکه عصبی کانولوشنی با معماری چند لایه اما کم عمق ترکیب می شود. فیلترهای دست ساز ساختارهای anchor را برای فیلترهای آموخته شده فراهم می کنند که ویژگی های تکرار شونده را بومی سازی ، امتیازدهی و رتبه بندی می کند. نمایش فضای مقیاس برای استخراج نقاط کلیدی در سطوح مختلف در شبکه استفاده می شود. و یک تابع هزینه را برای شناسایی ویژگی های مناسب و قوی که در طیف وسیعی از مقیاس ها وجود دارد و به حداکثر رساندن نمره تکرارپذیری طراحی می کنیم.

مدل Key.Net معرفی شده در مقاله بر روی داده هایی که به صورت مصنوعی از ImageNet ایجاد شده و در معیار HPatches ارزیابی می شود ، آموزش دیده است. نتایج نشان می دهد که رویکرد این مقاله از نظر تکرارپذیری ، عملکرد و پیچیدگی بهتر از detectors پیشرفته موجود عمل می کند.

مقدمه:

پیشرفتهای تحقیقاتی در آشکارسازها و توصیفگرهای ویژگی محلی وجود داشته منجر به پیشرفتهای چشمگیری در زمینه هایی مانند تطبیق تصویر ، تشخیص object ، ناوبری خود هدایت شده یا بازسازی سه بعدی شده است.

اگرچه جهت کلی روشهای تطبیق تصویر در حال حرکت به سمت سیستمهای مبتنی بر یادگیری و بخصوص شبکه عصبی کانوولوشنی است ، اما مزیت روشهای یادگیری نسبت به روشهای دست ساز به روشنی در تشخیص کلیدواژه اثبات نشده است.

با وجود ناکارآمدی غیر عملی روشهای اولیه به طور خاص ، شبکه های عصبی Convolutional (CNN) توانستند به طور قابل توجهی خطای تطبیق را در توصیف کنندگان محلی کاهش دهند.

این کارها باعث تلاشهای تحقیقاتی بیشتر و در نتیجه بهبود کارایی توصیف کنندگان مبتنی بر CNN می شود.

با این حال ، محبوبیت روزافزون هدست های واقعیت افزوده و همچنین برنامه های گوشی های هوشمند ، توجه بیشتری را به detectors ویژگی محلی قابل اعتماد و کارآمد جلب کرده است که که می تواند برای تخمین سطح ، بازسازی سه بعدی پراکنده ، کسب مدل سه بعدی یا ترازبندی اشیا و ... استفاده شود.

به طور سنتی ، آشکارسازهای ویژگی محلی بر اساس فیلترهای مهندسی شده به اصطلاح دست ساز ساخته می شدند.

به عنوان مثال ، رویکردهایی مانند Differenceof Gaussians، Harris-Laplace یا Hessian-Affine از ترکیب مشتقات تصویر برای محاسبه نقشه های ویژگی استفاده می کنند ، که به طرز قابل توجهی مشابه عملیات در لایه های آموزش دیده CNN است.

فقط با چند لایه ، یک شبکه می تواند رفتار detectors های سنتی را با یادگیری مقادیر مناسب در فیلترهای کانولوشن تقلید کند.

با این حال، برخلاف موفقیت در توصیفات تصویر محلی مبتنی بر CNN ،پیشرفت در ارائه شده توسط روشهای کاملاً مبتنی بر CNN که اخیراً پیشنهاد شده اند، از نظر معیارهای پذیرفته شده گسترده ای مانند تکرارپذیری محدود هستند.

یکی از دلایل دقت کم آنها هنگام تخمین پارامترهای ترکیبی مناطق مشخصه است. مقاومت به تغییرات مقیاس به ویژه مشکل ساز به نظر می رسد در حالی که پارامترهای دیگر مانند جهت گیری غالب را می توان به خوبی توسط CNN کنترل کرد

این باعث ایجاد انگیزه در معماری جدید در ایم مقاله می شود که Keynet نامیده می شود که از فیلترهای دست ساز و آموخته شده و همچنین نمایش چند مقیاس استفاده می کند.

معماری Key.Net در شکل ۱ نشان داده شده است. معماری پیشنهادی Key.Net ترکیبی از فیلترهای ساخته شده و آموخته شده برای استخراج ویژگی ها در مقیاس های مختلف است. نقشه های ویژگی نمونه برداری شده و بهم پیوسته اند فیلتر آخرین آموخته شده برای به دست آوردن نقشه پاسخ نهایی ، میزان فضای مقیاس را ترکیب می کند.

معرفی فیلترهای دست ساز ، که به عنوان لنگر نرم عمل می کنند ، امکان کاهش تعداد پارامترهای استفاده شده توسط detectors پیشرفته را فراهم می کند ، در حالی که عملکرد را از نظر تکرارپذیری حفظ می کنند.

این مدل بر روی نمایش چند مقیاس از تصاویر در اندازه کامل عمل می کند و یک نقشه پاسخ را شامل نمره نقطه کلیدی برای هر پیکسل برمی گرداند.

ورودی چند مقیاسه به شبکه اجازه می دهد تا نقاط کلیدی پایدار را ارائه دهد ، بنابراین قدرت تغییرات مقیاس را فراهم می کند.

در حالت ایده آل ، یک detectors قوی قادر است ویژگی های یکسانی را برای تصاویری که تحت تغییرات هندسی یا فوتومتریک مختلفی قرار دارند ، ارائه دهد.

تعدادی از کارهای مرتبط عملکرد هدف خود را برای پرداختن به این موضوع متمرکز کرده اند ، اگرچه این کارها یا براساس تکه های محلی یا از دست دادن رگرسیون نقشه کلی یا به اصطلاح نقشه جهانی بود

ما یک اپراتور کاملاً متفاوت ، چند مقیاس را طراحی می کنیم که نکات کلیدی را در مناطق چند مقیاس پیشنهاد می کند.

ما به طور گسترده ای از معیار HPatches که اخیراً معرفی شده برای میزان دقت استفاده میکنیم.

به طور خلاصه ، کارهای انجام شده در این تحقیق به شرح زیر است:

- یک detectors key point که ویژگی های دست ساخته و آموخته CNN را با هم ترکیب می کند.
- یک تابع هزینه و اپراتور جدید برای شناسایی و رتبه بندی نقاط کلیدی پایدار در مقیاس ها
- تشخیص ویژگی چند مقیاس با معماری کم عمق

بقیه مقاله به شرح زیر است.

ما کارهای انجام شده مربوطه را در بخش ۲ مشاهده می کنیم. بخش ۳ معماری ترکیبی پیشنهادی Key.Net فیلترهای CNN ساخته و ساخته شده و آموخته شده را ارائه می دهد و بخش ۴ تابع هزینه را معرفی می کند. جزئیات اجرا در بخش ۵ آورده شده و نتایج در بخش ۶ ارائه شده است.

2. کارهای مرتبط

تحقیقات بسیاری وجود دارد که به طور گسترده در مورد روش های تشخیص ویژگی بحث می کند. ما کارهای مرتبط را در دو دسته اصلی ارائه می دهیم: روش های سنتی و مبتنی بر یادگیری.

2.1 ردیاب های دست ساز یا ردیاب های مهندسی شده

ردیاب های ویژگی های سنتی ساختارهای هندسی را از طریق الگوریتم های مهندسی شده بومی سازی می کنند ، که اغلب از آنها به عنوان روش های دستی یاد می شود.

ردیاب های Harris و Hessian از مشتقات تصویر درجه اول و دوم برای یافتن گوشه ها یا لکه های تصاویر استفاده کردند. این آشکارسازها بیشتر برای مقابله با تحولات چند مقیاسی و تغییر شکل گسترش یافتند.

بعداً ، SURF با استفاده از تصاویر یکپارچه و تقریب ماتریس Hessian ، روند شناسایی را تسریع کرد. اگرچه آشکارسازهای گوشه قوی و کارآمد هستند ، اما روش های دیگر به دنبال ساختارهای جایگزین در تصاویر هستند. SIFT به دنبال لکه هایی در سطح چند مقیاس بود و MSER مناطق پایدار را به عنوان نقاط کلیدی تقسیم و انتخاب کرد.

2.2. detectors های یاد گرفته شده

موفقیت روشهای آموخته شده در تشخیص object و توصیف ویژگی ها ، محققین را بر آن داشت تا تکنیک های مشابه را برای ردیاب های ویژگی کشف کند. FAST یکی از اولین تلاشها برای استفاده از یادگیری ماشین برای استخراج یک آشکارساز گوشه بود.

کارهای بعدی با بهینه سازی FAST ، افزودن توصیفگر یا برآورد جهت گیری ، FAST را گسترش دادند. آخرین پیشرفت های CNN نیز در شناسایی ویژگی ها تأثیر داشته است.

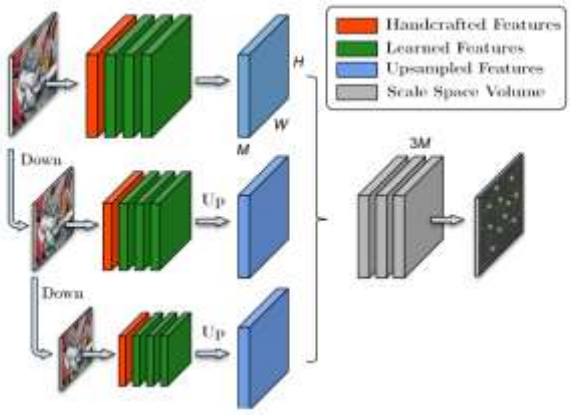
TILDE چندین مدل رگرسیون خطی چند قطعه ای را آموزش داده است تا نقاط بهره ای را که تحت تغییرات شدید مقاوم هستند ، شناسایی کند.

در تحقیقی دیگر فرمول جدیدی را برای آموزش CNN براساس محدودیت های متغیر ویژگی معرفی شد.

در تحقیقی دیگر پیش بینی کرد که چه ویژگی ها و توصیفاتی با هم مطابقت دارند.

در تحقیقی دیگر برای یادگیری پیش بینی پارامترهای ترکیبی یک ویژگی محلی ، از ضریب توصیف کننده استفاده کرد.

اخیراً ، یک شبکه را معرفی کرد تا یاد بگیرد. علاوه بر این ، CNN های دیگر نیز برای انجام وظایفی فراتر از تشخیص یا تطبیق مطالعه شد.



تصویر ۱ : معماری پیشنهادی Key.Net ترکیبی از فیلترهای ساخته شده و آموخته شده برای استخراج ویژگی ها در مقیاس های مختلف است. نقشه های ویژگی نمونه برداری و منشور می شوند. فیلتر آخرین آموخته شده برای به دست آوردن final response map، میزان فضای مقیاس را ترکیب می کند.

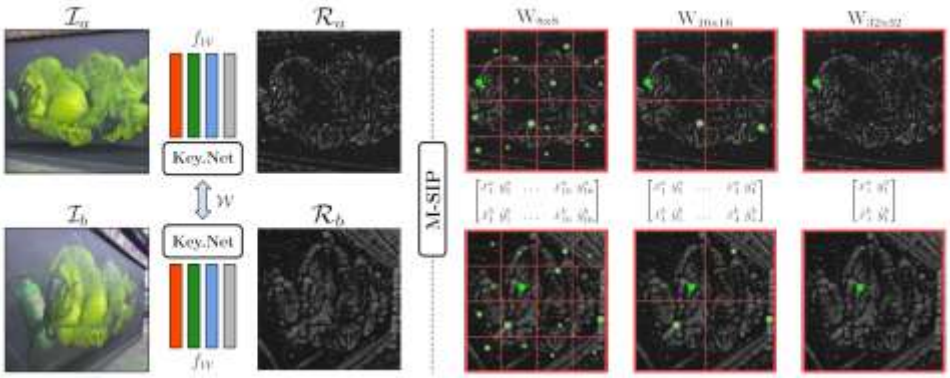
۱. معماری Key.net

معماری Key.Net ترکیبی از ایده های موفقیت آمیز از روش های دست ساز و آموخته شده یعنی استخراج ویژگی های مبتنی بر شیب ، ترکیبات آموخته شده از ویژگی های سطح پایین و نمایش هرم در مقیاس چند است.

۱،۱ فیلترهای دست ساز و آموخته شده

طراحی فیلترهای handcrafted از موفقیت ردیاب های هریس و هسیان الهام گرفته شده است که از مشتقات مرتبه اول و دوم برای محاسبه پاسخ های برجسته گوشه استفاده می کنند. مجموعه کاملی از مشتقات LocalJet نامیده می شود و آنها سیگنال موجود در محله محلی را که از گسترش تیلور شناخته می شود تقریبی می کنند:

$$I_{i_1,\ldots,i_n} = I_0 * \partial_{i_1},..,i_n g\sigma(\underset{x}{\rightarrow})$$



تصویر ۲ : روند آموزش سیامی. تصویر I_a و I_b برای تولید نقشه های پاسخ خود ، R_a و R_b از طریق Key.Net می روند. M-SIP مختصات نقطه علاقه را برای هر یک از پنجره ها در مناطق چند مقیاس پیشنهاد می کند. تابع از دست دادن نهایی به عنوان یک رگرسیون از شاخص های مختصات از I_a و حداکثر مختصات محلی از I_b محاسبه می شود. بهتر تجسم رنگ است.

جایی که g_σ نشانگر Gaussian با عرض σ است که در $\vec{x} = \vec{0}$ قرار دارد و جهت را نشان می دهد. مشتقات مرتبه بالاتر یعنی n>2 به نویز حساس هستند و به هسته های بزرگ احتیاج دارند ، بنابراین مشتقات و ترکیبات آنها را فقط تا مرتبه دوم شامل می کنیم:

- First order : از تصویر I شیب درجه یک I_x و I_y را بدست می آوریم. علاوه بر این ، ما $I_x * I_y$، I_x^2 و I_y^2 را در ماتریس لحظه دوم ردیاب هریس محاسبه می کنیم.
- Second order : از تصویر I ، مشتقات مرتبه ۲ ، I_{xx}، I_{yy} و I_{xy} نیز مانند ماتریس هسیان استفاده شده در آشکارسازهای هسیان و DoG هستند. از آنجا که آشکارساز هسی از تعیین کننده ماتریس هسی استفاده می کند ، $I_{xx} * I_{yy}$ و I^2_{xy} را اضافه می کنیم.
- Learned : یک لایه کانولوشن با فیلترهای M ، یک لایه نرمال سازی دسته ای و یک تابع فعال سازی ReLU یک بلوک یاد گرفته شده را تشکیل می دهد.

فیلترهای رمزگذاری شده سخت، تعداد پارامترهای قابل یادگیری برای آموزش معماری را کاهش می دهند و باعث بهبود ثبات و همگرایی در حین تولید مجدد می شوند.

۳,۲ هرم چند مقیاس

ما معماری خود را به گونه ای طراحی می کنیم که نسبت به مقیاس کوچک بدون نیاز به محاسبه چندین پاس به جلو محکم باشد. همانطور که در شکل ۱ نشان داده شده است ، شبکه شامل سه سطح مقیاس تصویر ورودی است که با ضریب ۱,۲ تار شده و از آن نمونه برداری می شود. تمام نقشه های ویژگی حاصل از فیلترهای ساخته شده دستی برای تغذیه پشته فیلترهای آموخته شده در هر یک از مقیاس ها بهم پیوسته اند. هر سه جریان دارای وزن هستند ، به گونه ای که یک نوع لنگر از سطوح مختلف حاصل می شود و مجموعه نامزدهای کلیدهای نهایی را تشکیل می دهد. سپس نقشه های مشخصه از تمام سطوح مقیاس نمونه برداری ، بهم پیوسته و به آخرین فیلتر کانولوشن منتقل می شوند تا نقشه پاسخ نهایی بدست آید.

۲. تابع جریمه – Loss Functions

در آموزش تحت نظارت ، تابع ضرر به حقیقت زمینی متکی است. در مورد نقاط کلیدی ، حقیقت زمین به خوبی مشخص نشده است زیرا مکانهای نقطه کلیدی مفید هستند تا زمانی که بتوانند بدون در نظر گرفتن تغییر شکل هندسی یا فوتومتریک ، آنها را به دقت تشخیص دهند. برخی از ردیاب های آموخته شده شبکه را برای شناسایی نقاط کلیدی بدون محدود کردن مکان آنها آموزش می دهند ، جایی که فقط تحول هموگرافی بین تصاویر به عنوان حقیقت زمین برای محاسبه تلفات به عنوان تابعی از تکرارپذیری کلیدهای کلیدی استفاده می شود.

سایر کارهای مشابه فواید استفاده از لنگر برای هدایت آموزش آنها را نشان می دهد. اگرچه لنگرها آموزش را پایدارتر کرده و منجر به نتایج بهتری می شوند ، اما در صورت عدم وجود لنگر در مجاورت ، شبکه از ارائه نکات کلیدی جدید جلوگیری می کند.

در مقابل ، فیلترهای ساخته شده در Key.Net با بهره گیری از روش های مبتنی بر لنگر محدودیت ضعیفی را ارائه می دهند در حالی که به آشکارساز اجازه می دهد تا کلیدهای پایدار جدیدی را ارائه دهد. در رویکرد ما ، فقط تغییر هندسی بین تصاویر برای هدایت ضرر مورد نیاز است.

۴,۱ لایه پیشنهاد شاخص

این بخش به معرفی لایه پیشنهاد (Index Proposal : IP) می پردازد که در بخش ۴,۲ به نسخه چند مقیاس آن گسترش یافته است.

استخراج مختصات برای آموزش آشکارسازهای کلیدی به طور گسترده ای مورد مطالعه قرار گرفته و پیشرفت های بزرگی را نشان داده است: مختصات استخراج شده در سطح وصله ، SuperPoint از یک softmax کانال هوشمند استفاده کرده است تا حداکثر متعلق به شبکه های ثابت x8۸ باشد و برای محاسبه حداکثر جهانی نقشه ویژگی ، از یک لایه softmax فضایی استفاده کرده و در هر نقشه ویژگی ، یک کاندید کلیدی را بدست آورده است. بر خلاف روش های قبلی ، لایه IP قادر است چندین مختصات نقطه کلید را با محوریت حداکثر محلی از یک تصویر واحد برگرداند بدون اینکه تعداد نقاط اصلی را به عمق نقشه ویژگی یا اندازه شبکه محدود کند.

به طور مشابه با تکنیک های دست ساز ، مکان های نقطه کلیدی با حداکثر محلی نقشه پاسخ فیلتر خروجی R توسط Key.Net نشان داده می شوند. اپراتور Spatial softmax یک روش موثر برای استخراج مکان حداکثر نرم در یک پنجره است. بنابراین ، برای اطمینان از اینکه لایه IP کاملاً متمایز است ، به اپراتور فضایی softmax اعتماد می کنیم تا مختصات یک نقطه کلید را در هر پنجره بدست آوریم. یک w_i پنجره را به اندازه N × N در R در نظر بگیرید ، با مقدار نمره در هر مختصات [u، v] داخل پنجره ، به طور نمایی مقیاس بندی شده و نرمال شده است:

$$m_i(u,v) = \frac{e^{w_i(u,v)}}{\Sigma_{j,k}^{N} e^{w_I}(j,k)}$$

با توجه به مقیاس نمایی ، حداکثر غلبه دارد و مکان مورد انتظار به عنوان میانگین وزنی محاسبه می شود $[\ \overline{u_i}\ ,\ \overline{v_i}]$ تقریبی حداکثر مختصات را می دهد:

$$[x_i, y_i]^T = [\ \overline{u_i}\ , \overline{v_i}]^T = \Sigma_{u,v}^{T}\,[W \odot m_i, W^T \odot m_i]^T + c_w$$

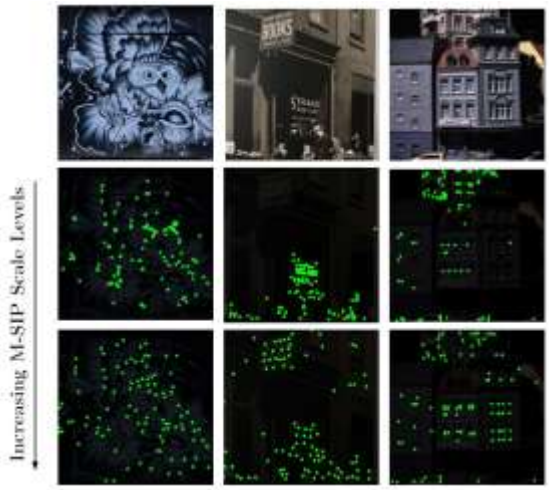
جایی که W یک هسته به اندازه N × N با مقادیر شاخص j = 1: N در امتداد ستون های آن ، محصول به صورت نقطه ای و cw مختصات گوشه بالا سمت چپ wi است. این شبیه سرکوب غیر حداکثر (NMS) است اما بر خلاف NMS ، لایه IP قابل تغییر است و یک میانگین وزنی از حداکثر جهانی پنجره است تا محل دقیق آن. بسته به پایه بیان توان در معادله ۲ ، حداکثرهای محلی چندگانه ممکن است تأثیر کم و بیش قابل توجهی در مختصات حاصل داشته باشند.

اگر ویژگی های مشابه تحت تغییرات مختلف تصویر شناسایی شوند ، یک ردیاب متغیر است. محدودیت متغیر به عنوان یک مشکل رگرسیون در فرموله شد. با توجه به تصاویر I_a و I_b ، و هموگرافی حقیقت زمین H_b، a بین آنها ، از دست دادن L بر اساس اختلاف مربع بین نقاط استخراج شده توسط لایه IP و حداکثر مختصات واقعی (NMS) در پنجره های مربوطه از I_a و I_b است:

$$\mathscr{L}_{IP}\left(I_a, I_b, H_{a,b}, N\right) = \Sigma_i \alpha_i \,||\, [\ x_i, y_i\ ]_a^T - H_{b,a}[\widehat{x_i}, \widehat{y_i}]_b^T \,||^2$$

$$\text{and } \alpha_i = R_a(x_i, y_i)_a + R_b(\widehat{x_i}, \widehat{y_i})_b$$

که در آن R_a و R_b نقشه پاسخ I_a و I_b با مختصات مربوط به هموگرافی $H_{b,\,a}$ است. ما از مختصات همگن برای سادگی عبور می کنیم. پارامتر α_i سهم هر مکان را بر اساس مقدار نمره آن کنترل می کند ، بنابراین فقط برای ویژگی های قابل توجه ضرر را محاسبه می کند. از آنجا که NMS غیرقابل تمیز است، گرادیان ها فقط در جایی که لایه IP اعمال می شود ، به صورت منفی تولید می شوند ، بنابراین ، I_a و I_b را تغییر می دهیم و برای اجرای سازگاری هر دو تلفات را ترکیب می کنیم.



تصویر ۳ : نکات کلیدی پس از افزودن پنجره های بزرگتر به اپراتور M-SIP بدست می آیند. نقاطی که پایدارتر هستند همچنان که اپراتور M-SIP اندازه پنجره خود را افزایش می دهد ، باقی می مانند. نقشه های مشخصه در ردیف میانی حاوی نقاطی در اطراف لبه ها یا مناطق غیرمتمایز است ، در حالی که ردیف پایین تشخیص هایی را نشان می دهد که در زیر تبدیلات هندسی قوی تر هستند.

۴,۲ لایه پیشنهاد مقیاس چند مقیاس

لایه IP در هر پنجره یک مکان را برمی گرداند ، بنابراین ، تعداد نقاط اصلی در هر تصویر به شدت به اندازه پنجره از پیش تعریف شده N بستگی دارد ، به ویژه ، با یک اندازه در حال افزایش ، فقط چند کلید اصلی در تصویر زنده می مانند. در، نویسندگان با جمع آوری ویژگی های تصویر نه تنها در یک پنجره فضایی بلکه در مقیاس های همسایه، عملکرد بهتر ویژگی های محلی را نشان دادند. ما پیشنهاد می کنیم از بین رفتن لایه IP را با ترکیب نمایندگی چند مقیاس از یک محله محلی گسترش دهیم. اندازه پنجره های متعدد شبکه را ترغیب می کند تا نقاط کلیدی را در طیف وسیعی از مقیاس ها پیدا کند. مزیت اضافی گنجاندن پنجره های بزرگتر این است که سایر کلیدهای داخل پنجره می توانند به عنوان لنگرگاه برای محل تخمین زده شده کلید اصلی غالب عمل کنند. ایده مشابه در، جایی که از مرزهای منطقه پایدار استفاده می شود ، موفقیت آمیز بود.

بنابراین ، ما لایه پیشنهاد مقیاس چند مقیاس (M-SIP) را پیشنهاد می دهیم. M-SIP چندین برابر نقشه پاسخ را به شبکه ها تقسیم می کند، هر کدام دارای اندازه پنجره $N_s \times N_s$ هستند و موقعیت کلید اصلی کاندیدا را برای هر پنجره محاسبه می کند ، همانطور که در شکل ۲ نشان داده شده است. سطح:

$$\mathscr{L}_{MSIP}(I_a, I_b, H_{a,b}) = \Sigma_s \lambda_s \mathcal{L}_{IP}(I_a, I_b, H_{a,b}, N_s)$$

در جایی که s شاخص سطح مقیاس با N_s به عنوان اندازه پنجره است ، LIP از دست دادن محدودیت متغیر است و λ_s پارامتر کنترل در سطح مقیاس s است که متناسب با افزایش سطح پنجره کاهش می یابد زیرا پنجره های بزرگتر منجر به ضرر بیشتر می شوند ، تا حدودی شبیه به نرمال سازی مقیاس-فضا است.

ترکیبی از مقیاس های مختلف فرایند ذاتی امتیازدهی و رتبه بندی همزمان نقاط کلیدی را در شبکه تحمیل می کند. به منظور به حداقل رساندن از دست دادن، شبکه باید بیاموزد که به ویژگی های قوی که در طیف وسیعی از مقیاس ها مسلط هستند ، امتیازات بالاتری بدهد. شکل ۳ نقشه های مختلف پاسخ را برای افزایش اندازه پنجره نشان می دهد.

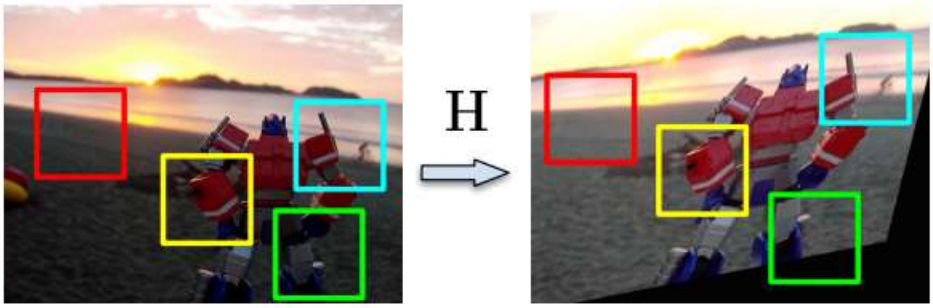
۳. تنظیمات تجربی

در این بخش ، ما جزئیات پیاده سازی ، معیارها و مجموعه داده مورد استفاده برای ارزیابی روش را ارائه می دهیم.

۳,۱ داده های آموزشی

ما یک مجموعه آموزش مصنوعی از مجموعه داده ImageNet ILSVRC 2012 ایجاد می کنیم. ما تحولات هندسی تصادفی را بر روی تصاویر اعمال می کنیم و به عنوان مجموعه آموزش خود ، جفت مناطق متناظر را استخراج می کنیم. این فرآیند در شکل ۴ نشان داده شده است. پارامترهای تحولات عبارتند از: scale [۰,۵ ، ۳,۵] ، skew [۰,۸. ، ۰,۸] و rotation [−۶۰° ، ۶۰°]. مناطق بدون بافت افتراق آور نیستند ، بنابراین ، ما با بررسی اینکه آیا پاسخ هر یک از فیلترهای دست ساز کمتر از آستانه است ، آنها را کنار می گذاریم. ما کنتراست ، روشنایی و مقدار رنگ را در فضای HSV به یکی از تصاویر تغییر می دهیم تا قدرت شبکه در برابر تغییرات روشنایی را بهبود ببخشیم. علاوه بر این ، برای هر جفت ، ماسک های باینری تولید می کنیم که نشان دهنده منطقه مشترک بین تصاویر است. از این ماسک ها در آموزش استفاده می شود تا از بازگرداندن شاخص های نقاط کلیدی که در منطقه مشترک وجود ندارد جلوگیری کند.
۱۲۰۰۰ جفت تصویر با اندازه ۱۹۲ × ۱۹۲ وجود دارد. ما از ۹۰۰۰ مورد به عنوان داده آموزشی و ۳۰۰۰ به عنوان مجموعه اعتبار سنجی استفاده می کنیم.

۳,۲ معیارهای ارزیابی



تصویر ۴ : ما تحولات هندسی و فوتومتریک تصادفی را به تصاویر اعمال می کنیم و به عنوان مجموعه آموزش ، جفت مناطق متناظر را استخراج می کنیم. منطقه قرمز با بررسی پاسخ فیلترهای دست ساز کنار گذاشته می شود.

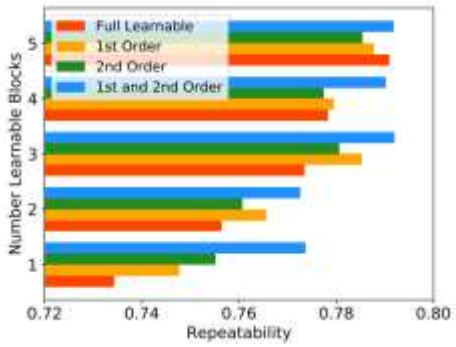
ما پروتکل ارزیابی پیشنهاد شده در کارهای پیگیری شده را بهبود می بخشیم. نمره تکرارپذیری برای یک جفت تصویر به عنوان نسبت بین تعداد کلیدهای متناظر و تعداد پایین تر کلیدهای شناسایی شده در یکی از دو تصویر محاسبه می شود. ما تعداد کلیدهای کلیدی استخراج شده را برای مقایسه بین روش ها اصلاح می کنیم و اجازه می دهیم هر کلیدواژه فقط یکبار مطابقت داشته باشد مانند [۲۵ ، ۱۴]. علاوه بر این، ما تعصب مربوط به عامل بزرگنمایی را که برای تسریع در محاسبه خطای همپوشانی بین نقاط کلیدی چند مقیاس اعمال شده است ، برطرف می کنیم. نقاط کلیدی توسط مختصات مکانی و مقیاسی که ویژگی ها در آن شناسایی شده اند ، شناسایی می شوند. برای شناسایی نقاط کلیدی متناظر ، خطای IoU ،Intersectionover-Union را بین حوزه های دو نامزد محاسبه می کنیم.
برای ارزیابی صحت مکان و مقیاس نقطه کلیدی به طور مستقل ، ما دو مجموعه آزمایش را انجام می دهیم. یکی براساس مقیاس های

شناسایی شده و دیگری فرض می کند که مقیاس ها با استفاده از پارامترهای حقیقت زمین به درستی تشخیص داده می شوند. در معیار ما ، ما از ۱۰۰۰ نقطه علاقه برتر که مربوط به منطقه مشترک بین تصاویر هستند استفاده می کنیم و زمانی که IoU کوچکتر از ۰,۴ باشد ، یک تطابق درست تلقی می شود ، یعنی همپوشانی بین مناطق مربوطه بیش از ۶۰٪ است. مقیاس ها نرمال می شوند، که اندازه بزرگتر را در یک جفت نقطه به ۳۰ پیکسل می رسانند ، و مقیاس دیگر را بر این اساس مقیاس بندی می کنند. سرکوب غیر حداکثر ۱۵ ۱۵ ۱۵ در زمان استنتاج تحت ارزیابی انجام می شود. مجموعه داده HPatches برای آزمایش استفاده می شود. HPatches شامل ۱۱۶ توالی است که به ترتیب بین دیدگاه و تبدیل روشنایی ، ۵۹ و ۵۷ توالی تقسیم می شوند. HPatches وصله های تصویری از پیش تعریف شده را برای ارزیابی توصیف کنندگان ارائه می دهد، در عوض ، ما از تصاویر کامل برای ارزیابی آشکارسازهای نقطه کلیدی استفاده می کنیم.

۳,۳ یادداشت های پیاده سازی

آموزش در یک خط لوله سیامی انجام می شود ، با دو نمونه Key.Net که دارای وزن مشترک هستند و همزمان به روز می شوند. هر لایه کانولوشن دارای ۸ فیلتر M = 8 به اندازه ۵ × ۵ ، با مقدار اولیه He و تنظیم کننده هسته L2 است. ما از دست دادن محدودیت متغیر LM-SIP را برای پنج سطح مقیاس ، با اندازه پنجره های M-SIP Ns ∋ [۸ ، ۱۶ ، ۲۴ ، ۳۲ ، ۴۰] و مدت ضرر comp λ_s [۲۵۶ ، ۶۴ ، ۱۶ ، ۴ ، محاسبه می کنیم ۱] ، که با انجام جستجوی ابر پارامتر در مجموعه اعتبار سنجی تعیین شد. اندازه پنجره های کاندیدای بزرگتر دارای خطاهای میانگین بیشتری بین نقاط مختصات هستند زیرا حداکثر فاصله متناسب با اندازه پنجره است. بنابراین ، λ_s بیشترین مقدار را برای کوچکترین پنجره دارد. ما از یک دسته دسته ۳۲ ، یک Adam Optimizer با میزان یادگیری 10^{-3} و ضریب پوسیدگی ۰,۵ بعد از ۲۰ دوره استفاده می کنیم. به طور متوسط ، معماری در ۳۰ دوره ، ۲ ساعت در دستگاهی با پردازنده i7-7700 با فرکانس ۳,۶۰ گیگاهرتز و NVIDIA GeForce GTX 1080 Ti همگراست. معیار ارزیابی ، مولد داده های مصنوعی ، شبکه Key.Net و ضرر و زیان با استفاده از TensorFlow اجرا شده و در GitHub موجود است.

M-SIP Region Sizes					
W_{8x8}	W_{16x16}	W_{24x24}	W_{32x32}	W_{40x40}	Repeatability
✓	-	-	-	-	70.5
✓	✓	-	-	-	74.6
✓	✓	✓	-	-	76.8
✓	✓	✓	✓	-	77.6
-	-	-	-	✓	65.7
-	-	-	✓	✓	71.4
-	-	✓	✓	✓	73.2
-	✓	✓	✓	✓	74.9
✓	✓	✓	✓	✓	**79.1**



تصویر ۵ : چپ: مقایسه نتایج تکرارپذیری برای چندین سطح در عملگر M-SIP. ما ترکیبات مختلفی از ضررهای زمینه را به عنوان ضرر نهایی ، از مناطق کوچکتر به بزرگتر نشان می دهیم. بهترین نتیجه هنگام استفاده از پنج اندازه پنجره از ۸ × ۸ تا ۴۰ ۴۰ ۴۰ است. درست: نتایج تکرار برای ترکیبات مختلف فیلترهای handcrafted و تعدادی لایه قابل یادگیری (هر کدام ۸ فیلتر M =). تعداد بیشتری از لایه ها منجر به نتایج بهتر می شوند. تمام نمرات تکرارپذیری براساس اعتبارسنجی مصنوعی تنظیم شده از ImageNet محاسبه می شوند.

	Num. Pyramid Levels					
	1	2	3	4	5	6
Rep.	72.5	74.6	79.1	79.4	**79.5**	78.6

(a) Number of input scale levels in Key.Net.

	Spatial Softmax Base					
	1.2	1.4	2.0	e	5.0	7.5
Rep.	77.5	78.4	77.9	**79.1**	74.6	73.2

(b) Spatial softmax base used in equation 2.

جدول ۱ : نتایج تکرار برای گزینه های مختلف طراحی در مجموعه اعتبار سنجی تنظیم شده از ImageNet.

۴. نتایج

در این بخش ، ما آزمایشات را ارائه می دهیم و نتایج را بحث می کنیم. ما در ابتدا نتایج مربوط به داده های اعتبارسنجی را برای چندین نوع از معماری پیشنهادی نشان می دهیم. در مرحله بعد ، نمرات تکرارپذیری Key.Net در مقیاس تک و چند مقیاس به همراه ردیاب های پیشرفته در HPatches ارائه می شود. علاوه بر این ، ما عملکرد تطبیق ، تعداد پارامترهای قابل یادگیری و زمان استنباط ردیاب پیشنهادی خود را ارزیابی می کنیم و با سایر تکنیک ها مقایسه می کنیم.

۴,۱ تجزیه و تحلیل اولیه

ما چندین ترکیب اصطلاحات از دست دادن ، فیلترهای مختلف دست ساز و اثرات تعداد لایه های قابل یادگیری یا سطح هرم را در معماری مطالعه می کنیم. سطح M-SIP در شکل ۵ (چپ) بررسی می شود که نشان می دهد تکرارپذیری افزایش یافته با سطوح مقیاس بیشتری در اپراتور M-SIP افزایش می یابد. علاوه بر این ، ما نشان می دهیم که چگونه افت با اندازه پنجره کوچکتر N ، تکرارپذیری را بهبود می بخشد. با این حال ، بهترین نتیجه زمانی حاصل می شود که همه سطوح با هم ترکیب شوند. ترکیب فیلترها در شکل ۵ (راست) تحلیل شده است. ما نتایج مربوط به فیلترهای درجه ۱ و ۲ و همچنین ترکیب آنها را نشان می دهیم. تعداد فیلترها در همه شبکه ها برابر است ، با این حال ، ما یا لایه اول ۱۰ فیلتر را با هسته های دست ساز مسدود می کنیم (cf بخش ۳,۱) یا بسته به نوع شبکه ما آنها را یاد می گیریم ، به عنوان مثال ، در Key کاملاً قابل یادگیری. فیلترها همانطور که بطور تصادفی مقداردهی اولیه شده و آموخته می شوند. نتایج نشان می دهد که اطلاعاتی که توسط فیلترهای دست ساز ارائه می شود ، در صورت کم بودن تعداد لایه های قابل یادگیری ، ضروری است. فیلترهای دست ساز به عنوان محدودیت های نرم عمل می کنند ، که مستقیماً مناطق بدون شیب را دور می زنند ، یعنی غیر تمیز با تکرارپذیری کم است. با این حال ، وقتی بلوک های قابل یادگیری بیشتری اضافه می کنیم ، نمرات تکرارپذیری برای شبکه های ترکیبی و کاملاً قابل یادگیری قابل مقایسه می شوند. به طور طبیعی ، فیلترهای دست ساز مبتنی بر گرادیان ساده هستند و در صورت نیاز ، معماری هایی با پیچیدگی کافی می توانند آنها را یاد بگیرند. با این حال ، استفاده از ویژگی های مهندسی شده با حفظ عملکرد به معماری کوچکتر منجر می شود ، که اغلب برای برنامه های زمان واقعی بسیار مهم است. به طور

خلاصه ، ترکیب هر دو نوع فیلتر باعث می شود تعداد لایه های قابل یادگیری به میزان قابل توجهی کاهش یابد. ما در آزمایش های بعدی از معماری Key.Net با سه بلوک قابل یادگیری استفاده می کنیم.

چندین سطح هرمی در ورودی شبکه همچنین بر عملکرد شناسایی تأثیر می گذارد همانطور که در جدول a۱ نشان داده شده است. برای یک سطح هرمی تنها ، از تصویر اصلی به عنوان ورودی استفاده می شود. افزودن سطح هرم مانند افزایش اندازه میدانهای پذیرا در معماری است. آزمایش ما نشان می دهد که استفاده از بیش از سه سطح منجر به بهبود قابل توجه نتایج نمی شود. در مجموعه اعتبار سنجی ، ما برای تکرار نمره تکرار ۷۲٫۵٪ ، یک سطح ۶٫۶٪ و پنج سطح ۷٫۰٪ افزایش می دهیم. بنابراین ، ما از سه سطح استفاده می کنیم که ضمن پایین نگه داشتن هزینه محاسبات ، به عملکرد خوبی نیز دست می یابند. Spatial Softmax Base در معادله ۲ میزان نرم بودن تخمین مختصات نقطه کلیدی را تعریف می کند. مقادیر زیاد مکان حداکثر جهانی را در داخل پنجره برمی گردانند ، در حالی که مقادیر کم حداکثر محلی را نشان می دهند. پایه در جدول b۱ متنوع است. هنگام استفاده از پایه در معادله ۲ نزدیک به مقدار e ، که با تنظیمات استفاده شده نمرات بهینه به دست می آیند.

	Viewpoint					Illumination				
	Repeatability		$\bar{\epsilon}_{IoU}$		S_{range}	Repeatability		$\bar{\epsilon}_{IoU}$		S_{range}
	SL	L	SL	L	SL	SL	L	SL	L	SL
SIFT-SI [5]	43.1	57.6	**0.18**	0.12	78.6	47.8	60.4	0.18	0.12	84.5
SURF-SI [20]	46.7	60.3	**0.18**	0.18	24.8	53.0	64.0	0.15	0.11	27.4
FAST-TI [24]	30.4	63.1	0.21	**0.10**	-	63.6	63.6	**0.09**	**0.09**	-
MSER-SI [23]	56.4	62.8	**0.12**	**0.08**	**503.7**	46.5	54.5	0.12	0.10	**524.8**
Harris-Laplace-SI [34]	45.1	62.0	0.20	0.13	**95.9**	52.7	62.0	0.17	**0.08**	**90.4**
KAZE-SI [21]	53.3	65.7	0.20	0.11	12.5	56.9	65.7	0.12	0.10	12.7
AKAZE-SI [22]	54.0	65.6	0.19	**0.10**	13.5	64.9	69.1	0.11	**0.09**	13.6
TILDE-TI [14]	31.0	65.1	0.20	0.15	-	**70.4**	70.4	0.11	0.11	-
LIFT-SI [7]	43.4	59.4	0.20	0.13	13.3	51.6	65.4	0.18	0.12	13.8
DNet-SI [9]	49.4	62.2	0.21	0.14	11.4	59.1	65.1	0.14	0.13	17.1
TCDET-SI [10]	49.6	61.6	0.23	0.16	6.7	66.9	**71.0**	0.16	0.15	11.4
SuperPoint-TI [13]	33.3	67.1	0.20	0.17	-	69.9	69.9	**0.10**	0.10	-
LF-Net-SI [11]	32.3	62.2	0.23	0.12	2.00	68.6	69.1	**0.10**	0.10	2.0
Tiny-Key.Net-SI	**57.8**	70.3	0.20	0.12	7.6	56.1	62.8	0.14	0.11	7.6
Key.Net-TI	34.2	**71.5**	0.20	0.11	-	**72.0**	**72.0**	**0.10**	0.10	-
Key.Net-SI	**60.5**	**73.2**	0.19	0.14	7.6	61.3	66.2	0.12	0.10	7.6

جدول ۲ : نتایج تکرارپذیری (٪) برای ترجمه (TI) و مقیاس (SI) آشکارسازهای ثابت در HPatches. ما همچنین خطای متوسط همپوشانی $\bar{\epsilon}_{IoU}$ و نسبت حداکثر به حداقل مقیاس استخراج شده SRange را گزارش می دهیم. در SL ، مقیاس ها و مکان ها برای محاسبه خطای همپوشانی استفاده می شود ، در عین حال ، در L ، فقط مکان ها استفاده می شوند و مقیاس ها به درستی تخمین زده می شوند. Key.Net و TinyKey.Net برای L و SL بهترین الگوریتم های دیدگاه هستند. در توالی های روشنایی ، Key.Net-TI بی تغییر ترجمه بهترین دقت را به دست می آورد. در میان آشکارسازهای ثابت SI ثابت ، TCDET بهترین در L و LF-Net در SL است.

۴,۲ تشخیص Keypoint

این بخش نتایج مربوط به پیشرفته ترین ردیاب های ویژگی محلی را به همراه روش پیشنهادی ما ارائه می دهد. جدول ۲ نمره تکرار ، میانگین خطای تقاطع- overunion unIoU و محدوده مقیاس Srange را نشان می دهد که نسبت بین مقادیر حداکثر و حداقل مقیاس نقاط بهره استخراج شده است. پسوندهای -TI و -SI ، به ترتیب به ترجمه (تشخیص فقط در یک مقیاس منفرد) و عدم تغییر مقیاس (تشخیص در مقیاس های مختلف) اشاره دارند. مکان Keypoint تنها با فرض تشخیص صحیح مقیاس تحت L ارزیابی می شود ، در حالی که مقیاس و مکان (SL) از مقیاس و مکان شناسایی شده واقعی برای محاسبه تکرارپذیری و خطای همپوشانی استفاده می کنند.

علاوه بر Key.Net ، ما Tiny-Key.Net را پیشنهاد می دهیم ، که یک معماری با اندازه کمتر با تمام فیلترهای ساخته شده دستی است اما فقط یک لایه قابل یادگیری با یک فیلتر (M = ۱) و یک ورودی تک مقیاس دارد. ایده پشت Tiny-Key.Net نشان دادن این است که با حفظ عملکرد خوب ، تا چه حد می توان از پیچیدگی کاسته شود. Key.Net و Tiny-Key.Net با ارزیابی ردیاب در چندین تصویر مقیاس بندی شده ، تا عدم تغییر مقیاس گسترش می یابند. ما همچنین نتایج را در ورودی تک مقیاس Key.Net-TI نشان می دهیم تا آن را مستقیماً با آشکارسازهای TI دیگر مانند SuperPoint یا TILDE مقایسه کنیم. ما آستانه الگوریتم ها را به گونه ای تنظیم می کنیم که حداقل ۱۰۰۰ امتیاز در هر عکس بازگرداند. همانطور که MSER مناطق بدون امتیاز و رتبه بندی را پیشنهاد می کند ، ما به طور تصادفی ۱۰۰۰ امتیاز را برای محاسبه نتایج انتخاب می کنیم. ما این آزمایش را ده بار تکرار می کنیم و میانگین نتایج را برای MSER می گیریم. Key.Net از نظر موقعیت مکانی و مقیاس بهترین نتایج را در توالی دیدگاه دارد. Tiny-Key.Net به خوبی Key.Net عمل نمی کند اما بعد از Key.Net-TI و Key.Net-SI در سه هسته قابل تکرار برتر قرار دارد.

در توالی های روشنایی ، Key.Net-TI بهترین عملکرد را در میان ردیاب های TI دارد ، که تحت تأثیر خطاهای تخمین مقیاس قرار نمی گیرد. TCDET که از نقاطی که توسط TILDE به عنوان لنگر شناسایی شده استفاده می کند ، در مقایسه با سایر ردیاب های SI دقیق ترین در برآورد مکان است. توجه داشته باشید که آشکارسازهای مبتنی بر TILDE به طور خاص برای توالی های روشنایی طراحی و آموزش داده شده اند. LF-Net با توجه به همپوشانی SL ، بهترین آشکارساز SI است و از تخمین مقیاس نادرست رنج زیادی نمی برد. با این حال ، تکرارپذیری آن بیشترین کاهش را از L به SL در بین تمام ردیاب های SI در توالی دید دارد. Key.Net-SI تغییرات مقیاس را بهتر از سایر روش ها نشان می دهد ، اما خطاهای موجود در نمونه برداری چند مقیاس بر روی آن تأثیر می گذارد وقتی که هیچ تغییری در مقیاس بین تصاویر یعنی توالی های روشنایی وجود نداشته باشد. این امر اغلب برای آشکارسازهایی با عدم تغییر بیشتر از آنچه در داده ها مورد نیاز است مشاهده شده است. ردیاب های دست ساز دارای کمترین خطای همپوشانی متوسط $\overline{\text{IoU}}$ در بین تمام ردیاب ها هستند. طیف گسترده ای از مقیاس ها S_{range} توسط MSER شناسایی می شود ، که به دلیل ماهیت تقسیم ویژگی های خود ، توانایی زیادی در استخراج ویژگی های محلی از مقیاس های مختلف دارد.

	Matching Score	
	View	Illum
MSER [23] + HardNet [38]	11.7	18.8
SIFT [5] + HardNet [38]	23.2	24.8
HarrisLaplace [34] + HardNet [38]	30.0	31.7
AKAZE [22] + HardNet [38]	36.4	41.4
TILDE [14] + HardNet [38]	32.3	39.3
LIFT [7] + HardNet [38]	30.3	32.8
DNet [9] + HardNet [38]	33.5	34.7
TCDET [10] + HardNet [38]	27.6	36.3
SuperPoint [13] + HardNet [38]	37.4	43.0
LF-Net [11] + HardNet [38]	26.9	**43.8**
LIFT [7]	21.8	26.5
SuperPoint [13]	38.0	41.5
LF-Net [11]	23.0	29.1
Tiny-Key.Net + HardNet [38]	37.9	37.3
Key.Net + HardNet [38]	**38.4**	39.7

جدول ۳ : نمره تطبیق (٪) بهترین آشکارسازها با HardNet و پیشرفته ترین ردیاب ها / توصیف کننده ها. نتایج در توالی HPatches ، هم از نظر دید و هم از نظر میزان روشنایی. معماری Key.Net بهترین نمره تطبیق را برای viewpoint کسب می کند ، در حالی که LF-Net + HardNet برای توالی های روشنایی است.

۴,۳ *تطابق* Matching

Number of Learnable Parameters				
TCDET	SuperPoint	LF-Net	Tiny-Key.Net	Key.Net
548k	940k	39k	**280**	5.9k

جدول ۴ : مقایسه تعداد پارامترهای قابل یادگیری برای معماری های پیشرفته. Tiny-Key.Net فقط یک بلوک قابل یادگیری با یک فیلتر دارد.

علاوه بر این ، برای نشان دادن اینکه ویژگی های شناسایی شده برای تطبیق مفید هستند ، جدول ۳ نمرات مطابقت را برای آشکارسازها همراه با توصیف کننده HardNet نشان می دهد. از آنجا که روش ما فقط بر روی قسمت تشخیص متمرکز است و برای مقایسه منصفانه ، از همان توصیفگر استفاده کرده و جهت گیری را برای همه روش های ارائه دهنده کنار می گذاریم. علاوه بر این ، ما در جدول LIFT، SuperPoint و LF-Net را با توصیفگرهای آنها آورده ایم ، اما تخمین جهت گیری آنها را نادیده می گیریم. SuperPoint و LF-Net دارای ۲۵۶ بعد توصیف کننده هستند ، در حالی که بعد از HardNet و 128 LIFT است. امتیاز تطبیق به عنوان نسبت بین ویژگی های مطابقت یافته و شناسایی شده محاسبه می شود (۱۰۰۰ برتر). امتیازات برتر تطبیق توسط Key.Net از نظر ، و LF-Net + HardNet در مورد روشنایی بدست می آید. ردیاب های مشخصه ای که به طور مشترک با توصیفگر بهینه سازی شده اند [۷ ، ۱۳ ، ۱۱] امتیاز مطابقت بهتری نسبت به ردیاب های یاد گرفته شده عادی در توالی های روشنایی دارند ، اما از نظر دیدگاه نه. AKAZE دست ساز نزدیک به روش های برتر آموخته شده برای هر دو توالی دید و روشنایی را انجام می دهد.

۶,۴ بهره وری

ما همچنین تعداد پارامترهای قابل یادگیری را مقایسه می کنیم ، سپس پیچیدگی پیش بینی کننده را نشان می دهد ، که منجر به افزایش خطر نصب بیش از حد و نیاز به مقدار زیادی از داده های آموزشی می شود. جدول ۴ تعداد تقریبی پارامترها را برای معماری های مختلف نشان می دهد. پارامترهای قابل یادگیری که در هنگام استنباط در قسمت ردیاب استفاده نمی شوند ، برای آشکارسازهای SuperPoint و LF-Net محاسبه نمی شوند. بیشترین پیچیدگی مربوط به SuperPoint با پارامترهای قابل یادگیری k۹۴۰ است. پارامترهای Key.Net تقریباً ۱۶۰ برابر و Tiny-Key.Net دارای ۳،۱۰۰ برابر پارامترهای کمتر از SuperPoint با تکرارپذیری بهتر برای صحنه های دیدگاه است. زمان استنباط تصویر ۶۰۰ × ۶۰۰ به ترتیب ۵,۷ میلی ثانیه (۱۷۵ FPS) و ۳۱ میلی ثانیه (۳۲,۲۵ FPS) برای Tiny-Key.Net و Key.Net است.

۵. نتیجه گیری

ما یک روش جدید برای شناسایی ویژگی های محلی معرفی کرده ایم که ترکیبی از فیلترهای CNN ساخته شده و آموخته شده است. ما یک لایه پیشنهادی شاخص چند مقیاسی را ارائه کرده ایم که نقاط کلیدی را در طیف وسیعی از مقیاس ها پیدا می کند ، با یک تابع از دست دادن که ویژگی های مقاوم و قابل تشخیص را بهینه می کند. ما چگونگی محاسبه و ترکیب افت قابل تشخیص کلیدواژه برای نمایش چند مقیاس را نشان دادیم. نتایج ارزیابی در مورد معیار بزرگ نشان می دهد که ترکیب ویژگی های دست ساز و آموخته شده و همچنین تجزیه و تحلیل چند مقیاس در مراحل مختلف شبکه ، نمرات تکرارپذیری را در مقایسه با روش های پیشرفته تشخیص کلید اصلی بهبود می بخشد.

بیشتر نشان می دهیم که افزایش بیش از حد پیچیدگی شبکه منجر به بهبود نتایج نمی شود. در مقابل ، استفاده از فیلترهای ساخته شده دستی می تواند به میزان قابل توجهی از پیچیدگی معماری منجر به ردیاب با ۲۸۰ پارامتر قابل یادگیری و استنباط ۱۷۵ فریم در ثانیه بکاهد. ردیاب های پیشنهادی هنگامی که با یک توصیفگر از نظر استفاده می شوند منجر به عملکرد مطابق پیشرفته می شوند.

منابع

[1] Karel Lenc and Andrea Vedaldi. Large scale evaluation of local image feature detectors on homography datasets. BMVC, 2018.

[2] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. CVPR, 2017.

[3] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. CVPR, 2015.

[4] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. CVPR, 2015.

[5] David G. Lowe. Distinctive image features from scaleinvariant keypoints. IJCV, 2004.

[6] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. ICCV, 2004.

[7] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. ECCV, 2016.

[8] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Toward geometric deep slam. arXiv preprint arXiv:1707.07410, 2017.

[9] Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. ECCV, 2016.

[10] Xu Zhang, Felix X. Yu, Svebor Karaman, and Shih-Fu Chang. Learning discriminative and transformation covariant local feature detectors. CVPR, 2017.

[11] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: Learning Local Features from Images. NIPS, 2018.

[12] Kwang Moo Yi, Yannick Verdie, Pascal Fua, and Vincent Lepetit. Learning to assign orientations to feature points. CVPR, 2016.

[13] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. CVPR Workshop, 2017.

[14] Yannick Verdie, Kwang Moo Yi, Pascal Fua, and Vincent Lepetit. Tilde: a temporally invariant learned detector. CVPR, 2015.

[15] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. TPAMI, 2005.

[16] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: a survey. Foundations and Trends in Computer Graphics and Vision, 2008.

[17] Chris Harris and Mike Stephens. A combined corner and edge detector. Alvey Vision Conference, 1988.

[18] Paul Beaudet. Rotationally invariant image operators. ICPR, 1978.

[19] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. IJCV, 2005.

[20] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). Computer Vision and Image Understanding, 2008.

[21] Pablo Fernandez Alcantarilla, Adrien Bartoli, and Andrew J. ´Davison. Kaze features. ECCV, 2012.

[22] Pablo Fernandez Alcantarilla, Jes ´ us Nuevo, and Adrien Bar- ´ toli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. BMVC, 2013.

[23] Jiri Matas, Chum Ondrej, Urban Martin, and Pajdla Toms. Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing, 2004.

[24] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. ECCV, 2006.

[25] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. TPAMI, 2010.

[26] Stefan Leutenegger, Chli Margarita, and Siegwart Roland. Brisk: Binary robust invariant scalable keypoints. ICCV, 2011.

[27] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. ICCV, 2011.

[28] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: unsupervised learning to rank for interest point detection. CVPR, 2017.

[29] Georgios Georgakis, Srikrishna Karanam, Ziyan Wu, Jan Ernst, and Jana Kosecka. End-to-end learning of keypoint detector and descriptor for pose invariant 3d matching. CVPR, 2018.

[30] Wilfried Hartmann, Michal Havlena, and Konrad Schindler. Predicting matchability. CVPR, 2014.

[31] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. CVPR, 2018.

[32] Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Repeatability is not enough: Learning affine regions via discriminability. ECCV, 2018.

[33] Luc Florack, Bart Ter Haar Romeny, Max Viergever, and Jan Koenderink. The gaussian scale-space paradigm and the multiscale local jet. IJCV, 2002.

[34] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. ICCV, 2001.

[35] Supasorn Suwajanakorn, Noah Snavely, Jonathan Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. NIPS, 2018.

[36] Jingming Dong and Stefano Soatto. Domain-size pooling in local descriptors: Dsp-sift. CVPR, 2017.

[37] Stepan Obdrzalek and Jiri Matas. Object recognition using local affine frames on distinguished regions. BMVC, 2002.

[38] Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. NIPS, 2017.