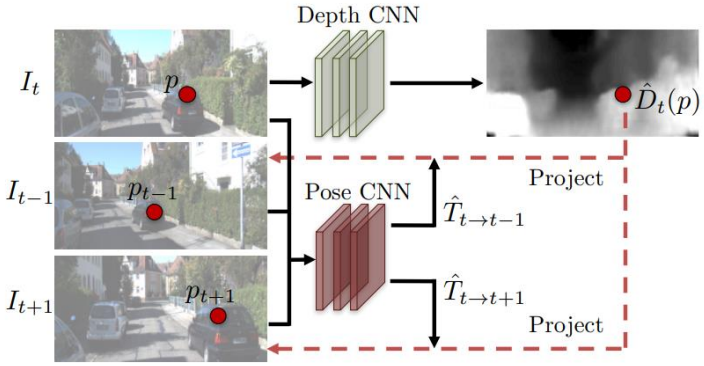
پروژه سوم: بازسازی ساختار از حرکت مبتنی بر یادگیری عمیق



شکل ۱-چارچوب تخمین عمق خودنظارتی از ویدیو

روش‌های کلاسیک بازسازی ساختار از حرکت[۱] علارغم عملکرد مناسب و سریع در اکثر موقعیت‌ها، وابستگی بسیار زیادی به یافتن تطبیق‌ها بین دو تصویر متوالی دارند. این وابستگی در مواردی که تغییرات شدید بین تصویرها وجود دارد یا نواحی که بافت یکنواخت دارند، این روش‌ها را با کاهش عملکرد شدید مواجه می‌کند. در سال‌های اخیر مسئله مشابه بازسازی ساختار از حرکت، در حوزه یادگیری عمیق تحت عنوان تخمین عمق تک دوربینی خودنظارتی[۲] مورد توجه محققین بوده‌است. در این پروژه یکی از اولین کارهای مبتنی بر یادگیری عمیق برای تخمین عمق را بررسی می‌کنیم[۳].

فرض کنید مطابق شکل ۱، سه فریم متوالی از یک ویدیو با نمادهای I_{t-1}، I_t و I_{t+1} داریم. فریم میانی را فریم هدف و دو فریم دیگر را فریم‌های مرجع (s) می‌نامیم. فرض کنید یک شبکه کانولوشنی با ساختار دلخواه (Depth CNN)، به ازای هر پیکسل از تصویر ورودی، عمق را تخمین می‌زند($\widehat{D}_t$). همچنین فرض کنید یک شبکه کانولوشنی با ساختار دلخواه (Pose CNN) تبدیل بین موقعیت دوربین در تصویر هدف و هر یک از دو تصویر مرجع را تخمین می‌زند($\widehat{T}_{t\to t\pm1}$). در صورتی که هر دو شبکه به صورت ایده‌آل عمل کنند. مختصات همگن[۴] یک نقطه در تصویر هدف $p_t = [u \quad v \quad 1]^T$ با رابطه زیر به مختصات همگن نرمال نشده همان نقطه در تصویر مرجع p_s نگاشت می‌شود:

---

[۱] Structure from motion

[۲] Self-Supervised Monocular Depth Estimation

[۳] Zhou, Tinghui, et al. "Unsupervised learning of depth and ego-motion from video." Proceedings of the IEEE conference on computer vision and pattern recognition. ۲۰۱۷.

[۴] homogeneous

$$p_s \sim K\hat{T}_{t\to s}\hat{D}_t(p_t)K^{-1}p_t$$

در این صورت اگر به ازای هر نقطه p_t از تصویر هدف، مقدار RGB متناظر با آن را با توجه به رابطه بالا از مختصات p_s تصویر مرجع جاگذاری کنیم، تصویر دوربین مرجع از نمای دوربین هدف $\hat{I}_s$ را خواهیم داشت[1]. از آنجایی که این $\hat{I}_s$ و I_t تصویرهایی از یک موقعیت هستند، در شرایطی که شبکه‌ها به صورت ایده‌آل عمل کنند باید $\mathrm{I}_t = \hat{I}_s$. بنابراین با شروع از وزن‌های اولیه برای هر دو شبکه، می‌توان از تابع هزینه زیر برای آموزش توامان هر دو شبکه استفاده کرد.

$$\mathcal{L}_{vs} = \sum_{s\epsilon\{t-1,t+1\}} \sum_{p\epsilon H\times W} \left|I_t(p) - \hat{I}_s(p)\right|$$

علاوه بر این، فرض هموار بودن عمق که در درس به وسیله مدل‌های گرافیکی احتمالاتی در خروجی اعمال می‌شد، در این شبکه با استفاده از یک تابع هزینه اضافه اعمال می‌شود.

$$\mathcal{L}_{smooth} = |\partial_x d_t^*|e^{-|\partial_x I_t|} + \left|\partial_y d_t^*\right|e^{-|\partial_y I_t|}$$

که در رابطه بالا $d_t^* = d_t/\bar{d_t}$ عمق نرمال شده با میانگین عمق $\bar{d_t}$ و $\partial_x I_t$ گرادیان تصویر در راستای x است[2].

یک فرض مهم در این چهارچوب این است که اشیا موجود در صحنه ساکن هستند. همچنین فرض شده که بین دو تصویر مرجع و هدف هیچگونه انسدادی وجود ندارد. این دو فرض در ویدیوهایی که از محیط‌های واقعی گرفته شده‌اند تقریبا هیچگاه برقرار نیست. برای این‌کار یک شبکه سوم برای پیش‌بینی پیکسل‌های که این دو فرض در آن‌ها نقض نمی‌شود، به چارچوب فوق اضافه می‌شود و تابع هزینه $\mathcal{L}_{vs}$ به صورت زیر تغییر می‌کند.

$$\mathcal{L}_{vs} = \sum_{s\epsilon\{t-1,t+1\}} \sum_{p\epsilon H\times W} \hat{E}_s(p)\left|I_t(p) - \hat{I}_s(p)\right|$$

که $\hat{E}_s(p)$ خروجی شبکه پیش‌بینی ماسک[3] ذکر شده‌است. از آنجایی که برای $\hat{E}_s(p)$ هیچگونه نظارت مستقیمی وجود ندارد، $\hat{E}_s(p)$ برای کمینه کردن تابع هزینه، به سمت صفر شدن کامل متمایل می‌شود. به همین منظور از یک تابع هزینه دیگر به صورت زیر استفاده می‌شود.

$$\mathcal{L}_{reg}\left(\hat{E}_s\right) = CrossEntropy(\hat{E}_s(p), 1\epsilon\mathbb{R}^{H\times W})$$

این تابع هزینه، $\hat{E}_s$ را تا حد ممکن غیر صفر نگه می‌دارد. تابع هزینه نهایی به صورت زیر تعریف‌می‌شود.

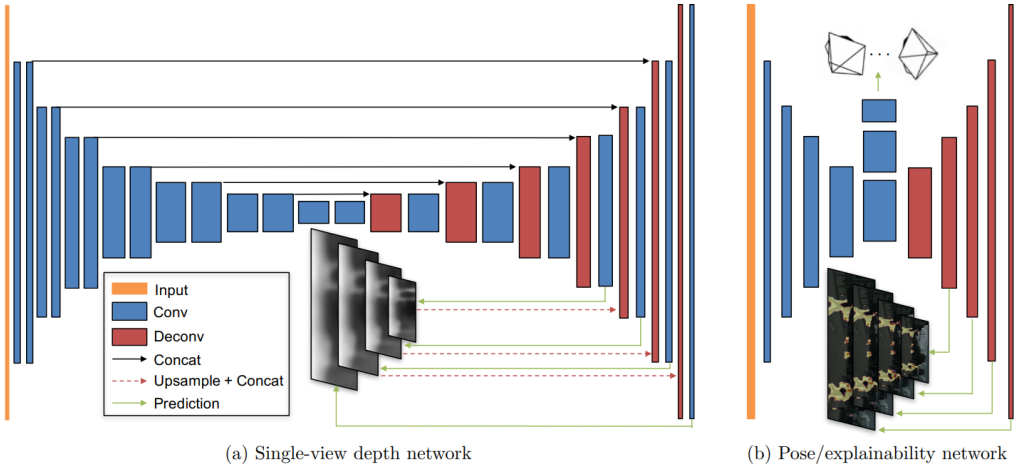
$$\mathcal{L}_{\text{final}} = \sum_l \mathcal{L}_{vs}^l + \lambda_s\mathcal{L}_{smooth}^l + \lambda_e\sum_s \mathcal{L}_{reg}\left(\hat{E}_s^l\right)$$

---

[1] در حقیقت دوربین مرجع و هدف یک دوربین هستند که با فاصله زمانی متفاوت و از موقعیت‌های متفاوتی از یک صحنه عکس برداری کرده‌اند.

[2] به زبان ساده این تابع هزینه الزام می‌کند که لبه‌های تصویر عمق و لبه‌های تصویر اصلی، یکسان باشند.

[3] این شبکه در مقاله اصلی explainability network نامیده شده‌است.



(a) Single-view depth network (b) Pose/explainability network

شکل ۲-ساختار کلی شبکه

این تابع هزینه مطابق شکل ۲، روی مقیاس‌های مختلف خروجی محاسبه می‌شود. برای اطلاعات بیشتر در مورد جزئیات پیاده سازی به مقاله اصلی مراجعه کنید.

الف) ساختار بالا را با تغییرات زیر بر روی بخشی از دیتاست kitti آموزش داده و بر روی حداقل ۱۰۰ فریم دیگر به صورت بصری ارزیابی کنید (ارزیابی با معیارهای کمی نمره امتیازی دارد).

- تنها از یک فریم به عنوان فریم مرجع برای محاسبه تابع هزینه و آموزش شبکه استفاده کنید. (s∈{t-1})
- از آنجایی که تخمین شار نوری[۱] و تخمین عمق شباهت زیادی دارند، می‌خواهیم از انتقال دانش[۲] بین این دو مسئله استفاده کنیم. به همین منظور باید فرض می‌کنیم ورودی شبکه تخمین که در بالا شرح داده شد، نه یک تصویر بلکه دو تصویر متوالی باشند. همچنین شبکه را دقیقا مشابه FlowNetS و با وزن‌های آموزش دیده آن در نظر بگیرید. با این تفاوت که خروجی شبکه به جای شار نوری، عمق باشد.
- از آنجایی که با تغییرات فوق، هم ورودی شبکه اصلی و هم شبکه pose/ explainability network دو تصویر متوالی است، جهت سادگی بیشتر، بخش کدگذار هر دو شبکه را مشترک درنظر بگیرید.

توجه کنید که با این تغییرات در فاز ارزیابی شبکه نیز، نیاز به دو فریم برای تخمین عمق داریم.

[۱] Optical flow

[۲] Transfer learning

دانشکده مهندسی کامپیوتر

پروژه درس بینایی سه بعدی کامپیوتر

ب) خروجی شبکه‌های تخمین عمق با چارچوبی که در این تمرین شرح داده شد، عمق نسبی است. چرا بدون هیچ اطلاعات اضافه‌ای نمی‌توان عمق را به مقیاس واقعی تخمین زد؟

ج-امتیازی) با توجه به ارتفاع دوربین از سطح زمین در دیتاست kitti، عمق نسبی را با ضریب مناسب برای هر فریم، به مقیاس واقعی تبدیل کنید.