



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

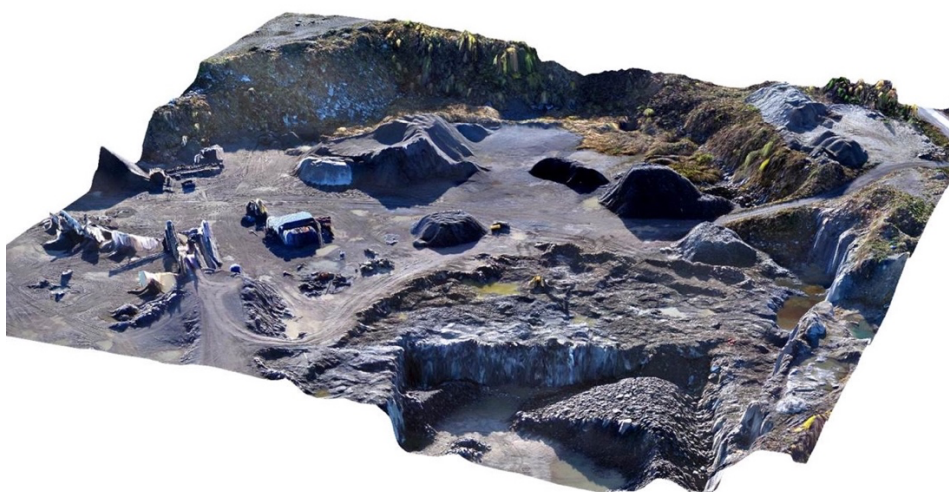
درس پناپی کامپیوتر سه بعدی

استاد درس جناب آقای دکتر جوانمردی

(تکلیف سوم)

محسن عبادپور | ۴۰۰۱۳۱۰۸۰ | m.ebadpour@aut.ac.ir

نیمسال دوم سال تحصیلی ۱۴۰۱-۱۴۰۲



فهرست پاسخ ها

- بخش اول: شبکه‌ی DISPNET ۳
- الف) آماده سازی و گرد آوری کد اولیه ۳
- ب) تهیه‌ی مجموعه داده و پردازش‌های آن ۳
- بخش اول: شبکه‌ی FLOWNET ۴
- الف) تغییر شبکه از DISPNET به FLOWNET ۴
- ب) اعمال تغییرات در شبکه ۴
- ج) آموزش انتقالی ۵
- بخش اول: آموزش و نتایج ۵
- الف) آموزش شبکه‌ها ۵
- ب) تخمین عمق نسبی و واقعی (چرا باهم متفاوت اند) ۶

بخش اول: شبکه‌ی DispNet

الف) آماده سازی و گرد آوری کد اولیه

از جایی که این سورس کد مورد نیاز برای پیاده سازی شبکه های مورد نظر بسیار سنگین و زیاد بود تصمیم گرفته شد که از سورس خود مقاله^۱ که بصورت عمومی و گیتهاب خود نویسنده موجود بود استفاده شود. دو نسخه با دو چارچوب pytorch و tensorflow موجود بود که در ابتدای کار نسخه‌ی تنسرفلو انتخاب و برای پیاده سازی clone شد اما در مسیر پیاده سازی یک مشکلی در اجرا روی سیستم های ویندوز به وجود آمد و آن عدم اجازه دسترسی به تعریف و استفاده از متغیر های سراسری در چند قسمت مختلف از کد بود. در قسمت issue های گیتهاب این مورد توسط نویسنده مورد تایید بود. به همین جهت نسخه pytorch برای ادامه مسیر جایگزین شد.

ب) تهیه‌ی مجموعه داده و پردازش‌های آن

برای تهیه مجموعه داده و برای اینکه هر سه مجموعه آزمون و آموزش و اعتبارسنجی دارای نمونه هایی باشند بایستی بیش از یک توالی از مجموعه داده KITTI انتخاب و مورد پردازش قرار میگرفت. برای تهیه‌ی مجموعه داده فایل های مورد نظر بصورت مستقیم از آقای آیار دریافت و مورد استفاده قرار گرفت. این دو توالی مربوط به روز 2011/09/28 با شماره های 1 و 2 می باشد.

برای ورودی دادن این مجموعه به سورس کد در فایل test_scenes نام های مورد نظر تنظیم شده و سپس با فایل Prepare_train_data.py اقدام به تولید فایل های نهایی کردم و آنان را در پوشه‌ی PrimaryFolder خود قرار دادم

^۱ <https://github.com/ClementPinard/SfmLearner>.

تا در زمان آموزش این فولدر را به dataloader پاس دهم. (این قسمت در لینوکس انجام شد چرا که طبق توضیحات قسمت قبل نیاز به تعریف متغیر سراسری بود)

شبکه‌ی FlowNet

بخش اول: شبکه‌ی FlowNet

الف) تغییر شبکه از DispNet به FlowNet

طبق خواسته‌ی مسئله ما بایستی شبکه را تغییر دهیم که در اینجا باید به ویژگی‌های شبکه توجه و تغییرات مورد نیاز را نسبت به آن انجام دهیم. اولین مورد ورودی شبکه است که دو تصویر مرجع و هدف رنگی می باشد لذا ورودی شبکه ۶ کانال تصویر(هر کدام RGB) می باشد. مورد بعدی استخراج ویژگی و افزایش کانال های ویژگی می باشد که طی عملیات های CNN تبدیل به ۱۰۲۴ کانال ویژگی می شود.

برای اصلاح و تقویت ویژگی های استخراجی شده و در نتیجه بدست آوردن کانال مطلوب تر در مقیاس بزرگتر، از DeConvolution استفاده شده و به ۶۴ کانال کاهش می یابد. حال بر اساس ۶۴ کانال ویژگی وجود و شبکه‌ی CNN دیگر اقدام به تخمین بردار های جریان می کنیم که نشان دهنده‌ی حرکت ظاهری می باشد.

ب) اعمال تغییرات در شبکه

برای اعمال تغییرات در شبکه و بدست آوردن FlowNet از گیت‌هاب^۲ استفاده شده است. از جایی که خروجی دو شبکه باهم متفاوت بود و ما بایستی آنان را یکی در نظر بگیریم تا مقایسه پذیر باشد، از چهار خروجی متناظر اول استفاده کردیم.

^۲ <https://github.com/ClementPinard/FlowNetPytorch>

و در آموزش و ایجاد مدل نیز کدهای مربوط وزن‌دهی اولیه و لود کردن وزن نیز تغییر یافت.

ج) آموزش انتقالی

در قسمتی از سوال خواسته شده است که از وزن‌های آموزش داده شده و pre-trained استفاده شود؛ برای این منظور از وزن‌های موجود در گوگل درایو^۳ استفاده شده است. برای استفاده از این وزن‌ها آن را دانلود و در پوشه pre_trained پروژه قرار دادم و نامش را در argument معادل pretrained_disp قرار دادم. اگر این مقدار باقی بماند از وزن‌های قبلی استفاده خواهد کرد. اگر برابر با None تعیین شود از وزن‌های اولیه تصادفی برای آموزش استفاده خواهد کرد.

آموزش و نتایج

بخش اول: آموزش و نتایج

الف) آموزش شبکه‌ها

طبق توضیحات داده شده در قسمت‌های قبل سه شبکه‌ی مورد نظر را آموزش دادیم که مشتمل بر شبکه‌ی DispNet و شبکه‌ی FlowNet که یکبار با وزن‌های تصادفی و یکبار با وزن‌های pre-trained می‌باشد.

*	Avg abs_diff	abs_rel	sq_rel	a1	a2	a3	ATE	RTE
DispNet	6.151	0.317	3.316	0.457	0.757	0.881	0.038	0.012
FlowNet(pre-trained)	18.529	1.040	21.268	0	0	0	0.050	0.01
FlowNet	34.932	3.102	184.761	0.015	0.040	0.073	0.085	0

^۳ https://drive.google.com/drive/folders/1dTpSyc7rIYYG19p1uiDfilcsmSPNy-_3

در جدول فوق، ATE نشان‌دهنده میانگین تفاوت در translation بین دو موقعیت است، در حالی که RE میزان تفاوت زاویه‌ای در rotation را اندازه‌گیری می‌کند. همچنین در زیر بترتیب نمونه ای از depth و disparity آورده شده و برای نمونه یک گیف از نیز از عمق مربوط به یکی از خروجی ها ایجاد و ضمیمه شده است؛ همانطور که مشاهده می شود خروجی چندان مطلوب نیست که دلیل آن را میتوان آموزش با تعداد تکرار کم دانست که دلیل آن منابع محدود آموزشی بوده است؛ در مقاله تعداد تکرار ۳۰۰۰ بوده اما در اجرا های بنده برابر با ۱۰۰ تعیین شده است.



ب) تخمین عمق نسبی و واقعی (چرا باهم متفاوت اند)

طبق توضیحات داده شده در خصوص چارچوب معرفی شده برای تخمین عمق، خروجی چارچوب عمق نسبی بوده و مبتنی بر شار نوری و تغییرات روشنایی و intensity تصاویر می باشد چرا که دو فریم ورودی مد نظر دو فریم متوالی و غیرهمزمان از صحنه بوده و نمیتوان آن را به عنوان تصاویر چپ/راست از یک صحنه در زمان یکسان در نظر گرفت و عمق واقعی را تخمین نمود لذا عمقی که حاصل از دو فریم متوالی بدست می آید نسبی می باشد.

سورس کد اجرا شده در گوگل درایو به آدرس زیر بوده و البته خروجی های فایل ژوپیتر حاصل از run آن ضمیمه شده است؛ سورس train شبکه‌ی Net Disp را آموزش داده، سورس new_train_test شبکه FlowNet با وزن انتقالی را آموزش داده و سورس new_train_test_no_w نیز شبکه FlowNet با وزن تصادفی اولیه را آموزش داده است.

<https://drive.google.com/drive/folders/1qP6xL5RbIsEyLpTTGZzJhk8bc5LFkewF?usp=sharing>