به نام خدا

دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

درس یادگیری عمیق با کاربرد در بینایی ماشین و پردازش صوت

تمرین شماره ۲

نام و نام خانوادگی : مهیار ملکی

شماره دانشجویی : ۸۱۰۱۰۰۴۷۶

فروردین ماه ۱۴۰۱

فهرست

مقدمه

هدف از این تمرین آشنایی بیشتر با شبکه های عصبی کانولوشنی است. این دسته از شبکه ها در مسائل بسیاری کاربرد دارند که به دلیل برخورداری از ویژگی های اتصال محلی فیلتر ها و اشتراک گذاری پارامترها برای کار در زمینه تصاویر و ویدیو ها بسیار مناسب هستند و به نتایج بسیار خوبی دست پیدا کرده اند و پیشرفت های زیادی در این زمینه صورت گرفته است. به علاوه این شبکه ها در مسائل کار با متن و سیگنال های مختلف نیز کارامد هستند. کار با تصاویر محوریت اصلی این تمرین است.

بدین منظور در سوال اول به بررسی و پیاده سازی مدل ساده شده مقاله [۱] سال ۲۰۱۶ پرداخته و با استفاده از شبکه های کانولوشنی نقشه برجستگی تصاویر را پیش بینی می‌کنیم. برجستگی ها در واقع نواحی ای از تصویر هستند که هنگام مشاهده، بیشتر توجه یک فرد را جلب می‌کنند. در واقع نقشه برجستگی میزان اهمیت پیکسل های مختلف یک تصویر را در سیستم بینایی انسان نشان می‌دهد.

همچنین در سوال دوم با استفاده از مقاله ای [۲] دیگر در سال ۲۰۱۶ و خروجی گرفتن از لایه های مختلف شبکه کانولوشنی AlexNet به بررسی بازنمایی لایه های مختلف شبکه از تصویر ورودی می‌پردازیم. می‌دانیم که شبکه های عصبی کانولوشنی توانایی حذف ویژگی های زائد و حفظ ویژگی های مفید را دارند. در واقع در این قسمت به بررسی نحوه انجام این بازنمایی در لایه های مختلف شبکه می‌پردازیم.

توضیحات کد:

سوال اول در ۲ فایل main و model آماده شده است. در تابع dirs_loader با دادن مقادیر دلخواه ورودی به متغیر WholeDataset می‌توان تعداد ورودی ها را کاهش داده و کد را فقط بر روی قسمت کوچکی از داده ها اجرا کرد. همچنین با دادن مقدار True کل داده ها به عنوان ورودی در نظر گرفته خواهند شد.

سوال دوم در فایل های model و چهار فایل main_conv2 و main_conv5 و main_fc6 و main_fc8 آماده شده است که از هر کدام به صورت جداگانه می‌توان برای بازیابی لایه خواسته شده استفاده کرد. همچنین همانند سوال اول می‌توان تعداد ورودی ها را کاهش داد.

---

[۱] Shallow and Deep Convolutional Networks for Saliency Prediction

[۲] Inverting Visual Representations with Convolutional Networks

سوال اول :

بخش الف

در سوال اول یک شبکه کانولوشنی عمیق با استفاده از مقاله ذکر شده و شروط سوال پیاده سازی شد. ساختار لایه های شبکه در جدول ۱ قابل مشاهده است. لازم به ذکر است که طبق صورت سوال در هر لایه، تعداد کانال ها نسبت به مقاله به نصف تقلیل یافته است. همچنین در متن مقاله تعداد لایه های کانولوشن ۸ عدد ذکر شده ولی در کد مرجع مقاله از ۹ لایه کانولوشن استفاده شده است، لذا در اینجا نیز طبق مدل پیاده شده در کد مقاله از ۹ لایه کانولوشن استفاده شده است.

همچنین هایپر پارامترهای استفاده شده در شبکه نیز در جدول ۲ قابل مشاهده هستند. لازم به ذکر است که با توجه به تقلیل و تغییر شبکه و محدودیت های منابع، برخی پارامترهای شبکه اندکی با موارد ذکر شده در مقاله متفاوت است.

جدول ۱ – Net Structure

Layer	InSize	OutSize	KernelSize	Stride	Padding
Conv1	3	48	7	1	3
Norm	local_size = 5 , alpha = 0.0001 , beta = 0.75				
MaxPool1	48	48	3	2	0
Conv2	48	128	5	1	2
MaxPool2	128	128	3	2	0
Conv3	128	256	3	1	1
Conv4	256	256	5	1	2
Conv5	256	256	5	1	2
Conv6	256	128	7	1	3
Conv7	128	64	11	1	5
Conv8	64	16	11	1	5
Conv9	16	1	13	1	6
Deconv	1	1	8	4	2

وزن دهی اولیه: مقاله از وزن های آماده شبکه VGG استفاده می‌کند اما در اینجا برای سادگی کار، وزن دهی اولیه صورت نپذیرفت لذا مشاهده شد که شبکه به خوبی آموزش نمی‌بیند. بنابراین با روش He و توزیع نرمال چناجه در جدول ۲ قابل مشاهده است، وزن دهی اولیه صورت گرفت.

نرمال سازی: طبق مقاله نرمال سازی باید به گونه ای باشد که مقادیر بین [-1,1] قرار بگیرند. بدین منظور پس از استفاده از ماژول ToTensor[۱] کتابخانه پایتورچ از میانگین و واریانس ۰.۵ استفاده شد.

جدول ۲ - Net Hyper Parameters

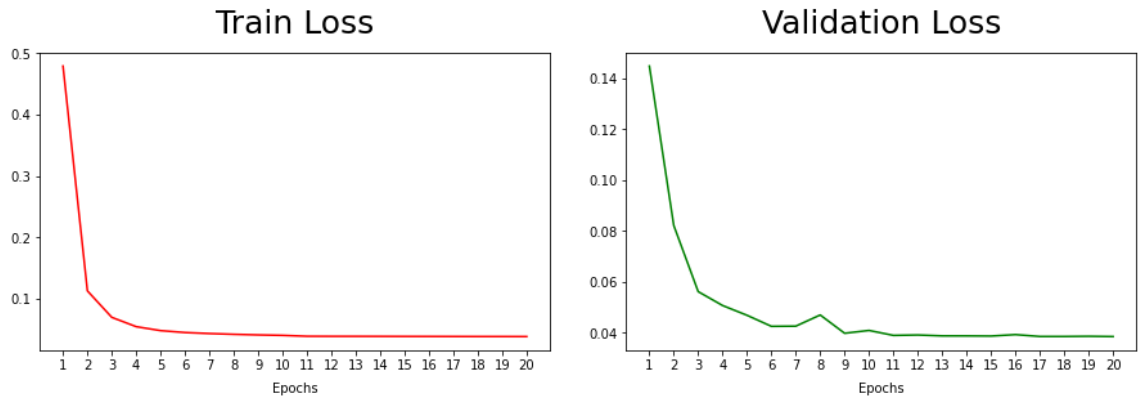
Network Used Hyper Parameters	
• Batch Size	2
• Epochs	20 (refrence: 24000)
• Input Image Size	240 , 320
• Output Image Size	236 , 316
• Learning Rate	0.0001 (refrence: 1.3e-7)
• Weight Decay	0.0001
• Momentum	0.9
• Optimizer	SGD
• Loss Function	MSE
• Activation Functions	ReLU
• Scheluler	LR decreases by factor of 0.1 every 10 epochs (refrence: 0.5 every 100 epoch)
• Conv layers weight initialization	He (kaiming)
• DeConv layer weight initialization	Normal (mean=0, std=0.00001)
• Minimum Train Loss reached	0.0385
• Minimum Test Loss reached	0.0384

[۱] این ماژول مقادیر ماتریس تصویر را به بازه ۰ تا ۱ می‌برد

بخش ب

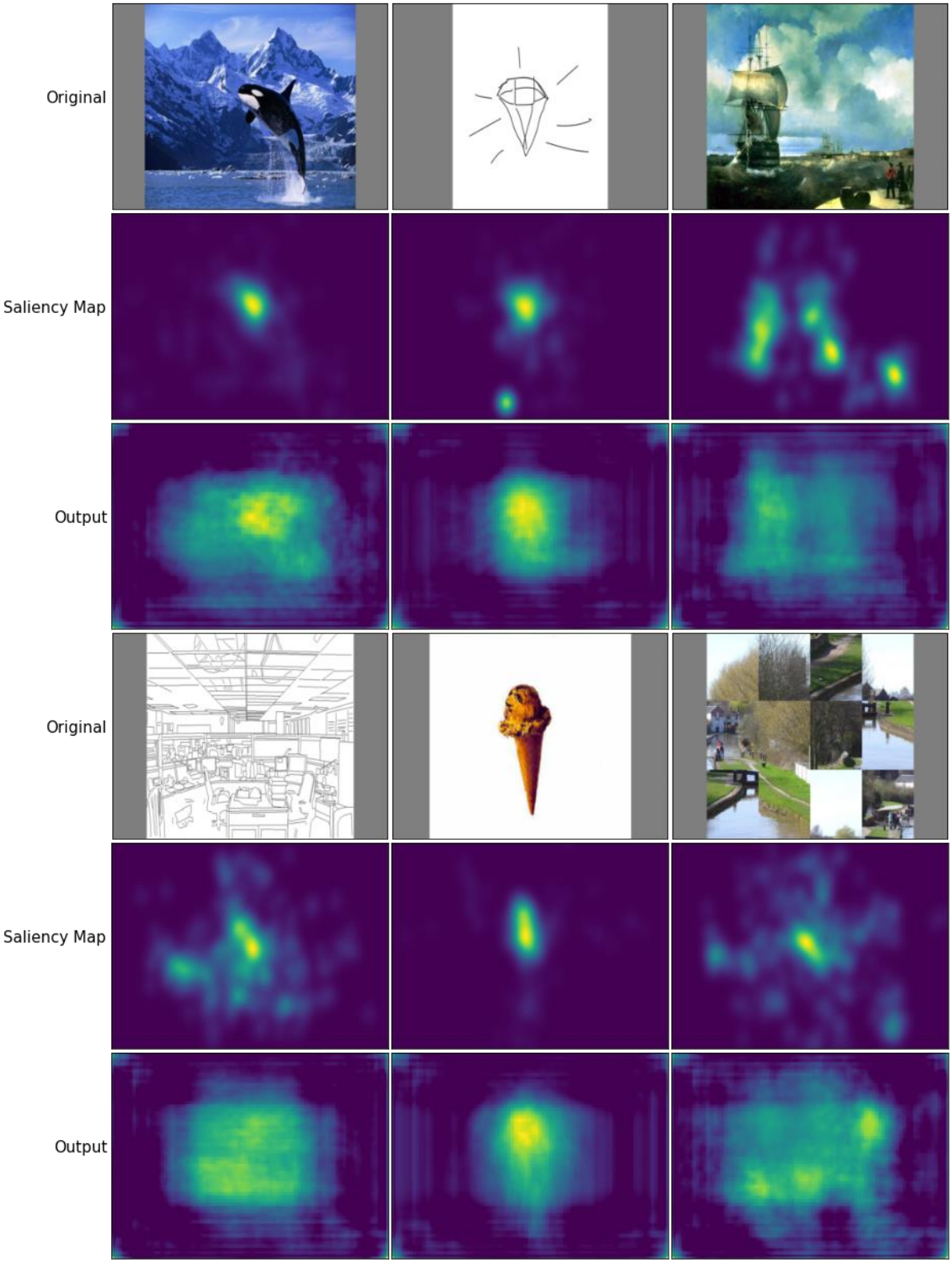
با توجه به شکل ۱ مشاهده می‌شود که بعد از ۱۰ الی ۱۵ ایپاک مقدار خطا هم برای داده های آموزش و هم ارزیابی به کمتر از ۰.۰۴ رسیده و دیگر بهبودی در ان دیده نمی‌شود. همچنین با توجه به جدول ۲ در ایپاک دهم سرعت آموزش یک دهم مقدار خود می‌شود، ولی همچنان تغییر قابل توجهی در خطای حاصل از شبکه رخ نمی‌دهد.

چنان چه در بخش ج خواهیم دید، نقشه برجستگی حاصل از شبکه حتی با وجود خطای بسیار کم ۰.۰۳ نیز اختلاف زیادی با خروجی مدنظر نظر دارد. این امر بدلیل کوچکتر کردن شبکه و تغییر هایپر پارامترهای آن با توجه محدودیت منابعی که داریم رخ داده است. لذا این نتایج قابل قبول است.



شکل ۱ - Train & Val Loss

بخش ج



شکل ۲- Saliency & Predictions

بخش ه

روش و نحوه سنجش و ارزیابی نقشه برجستگی پیش بینی شده توسط شبکه یکی از مواردی است که مورد توجه محققان قرار گرفته، لذا روش های مختلفی برای این امر پیشنهاد شده است.

یکی از روش های ارزیابی برجستگی سطح زیر نمودار ROC است که آن را با نام AUC می‌شناسیم. در این روش نقشه برجستگی پیشبینی شده را مانند یک طبقه بند نقاط تثبیت شده (Fixation Points) ارزیابی می‌کنیم. این ایده بدین شکل است که هدف پیشبینی را به صورت پیکسل هایی ثابت روی تصویر در نظر می‌گیریم و یک نقشه برجستگی را به عنوان طبقه بندی که مشخص می‌کند نقاط فیکس شده را در بر گرفته یا خیر در نظر می‌گیریم. حال یک طبقه بند باینری داریم که می‌توان نمودار ROC آن را بر اساس مقادیر false positive و true positive رسم کرده و با حد آستانه های مختلف نقشه برجستگی پیشبینی شده را ارزیابی کرد.

یکی از این روش های ارزیابی توسط نمودار ROC با نام Judd AUC در مقاله[1] سال ۲۰۱۲ معرفی شده است. در این روش نرخ TP برابر با نسبت پیکسل های ثابت درست تشخیص داده شده به کل پیکسل های تثبیت شده در نظر گرفته شده است (تشخیص های درست برجستگی هایی هستند که مقدار آنها بیشتر از حد آستانه است) همچنین نرخ FP را برابر با نسبت پیکسل هایی اشتباها به عنوان برجستگی تشخیص داده شده به کل پیکسل ها در نظر می‌گیرند.

(کد این بخش پیاده سازی نشده است)

[1] A benchmark of computational models of saliency to predict human fixations

سوال دوم :

بخش الف

در سوال دوم با خروجی گرفتن از لایه های مختلف شبکه AlexNet به بررسی بازنمایی لایه های مختلف آن از تصویر ورودی می‌پردازیم. لایه های خواسته شده در صورت سوال برای بازنمایی عبارتند از لایه های دوم و پنجم که کانولوشن هستند و لایه های ششم و هشتم که از نوع تماما متصل می‌باشند. به منظور بازنمایی خروجی هر یک از این لایه ها باید از تعدادی لایه های کانولوشن و تماما متصل دیگر نیز استفاده کرد. ساختار هر کدام از آنها در جداول ۳ و ۴ قابل مشاهده است. همچنین پارامترهای مورد استفاده برای آموزش شبکه نیز به شرح جدول ۵ می‌باشد.

جدول ۳- ساختار بازنمایی لایه CONV2 و CONV5

Layer	Input	InSize	OutSize	Kernel	Stride	Padding
Conv1	AlexNet-CONV2 AlexNet-CONV5	192	192	3	1	1
Conv2	Conv1	192	192	3	1	1
Conv3	Conv2	192	192	3	1	1
Upconv1	Conv3	192	192	5	2	Input=1 Output=2
Upconv2	Upconv1	192	128	5	2	Input=1 Output=2
Upconv3	Upconv2	128	64	5	2	Input=1 Output=2
Upconv4	Upconv3	64	3	5	2	Input=1 Output=2
Upconv5	Upconv4	32	3	5	2	Input=1 Output=2
UpSample	Upconv5	-	-	-	-	-

جدول 4- ساختار بازنمایی لایه FC6 و FC8

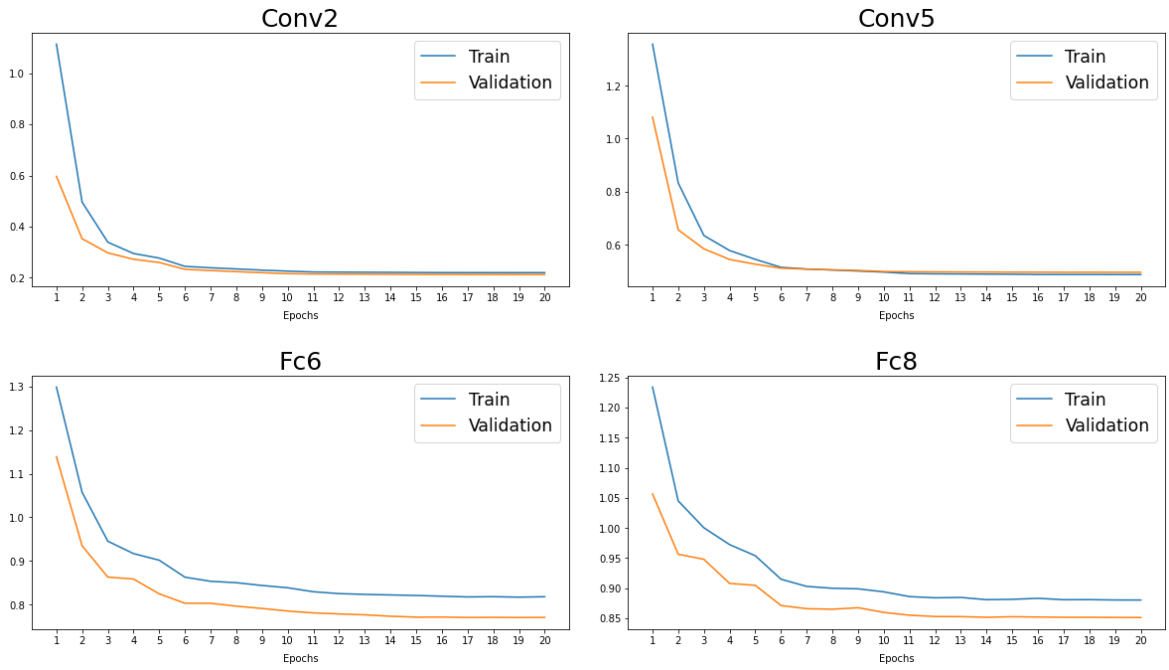
Layer	Input	InSize	OutSize	Kernel	Stride	Padding
Fc1	AlexNet-FC6	4096	4096	-	-	-
	AlexNet-FC8	1000				-
Fc2	Fc1	4096	4096	-	-	-
Fc3	Fc2	4096	4096	-	-	-
reshape	Fc3	4096	256	-	-	-
Upconv1	reashape	256	256	5	2	Input=1 Output=2
Upconv2	Upconv1	256	128	5	2	Input=1 Output=2
Upconv3	Upconv2	128	64	5	2	Input=1 Output=2
Upconv4	Upconv3	64	32	5	2	Input=1 Output=2
Upconv5	Upconv4	32	3	5	2	Input=1 Output=2
UpSample	Upconv5	-	-	-	-	-

جدول ٥- Net Hyper Parameters

Network Used Hyper Parameters	
• Batch Size	64
• Epochs	20
• Input Image Size	227*227
• Output Image Size	Conv2 : 208*208
	Conv5 : 192*192
	Fc6 : 128*128
	Fc8 : 128*128
• Normalization	mean = [0.485, 0.456, 0.406] std = [0.229, 0.224, 0.225]
• Learning Rate	0.001
• Weight Decay	0.0001
• Optimizer	ADAM ($\beta 1$ = 0.9, $\beta 2$ = 0.999)
• Loss Function	MSE
• Activation Functions	Leaky ReLU
• Scheluler	LR decreases by 0.1 every 5 epochs
• Minimum Train Loss reached	Conv2 : 0.203
	Conv5 : 0.490
	Fc6 : 0.818
	Fc8 : 0.880
• Minimum Test Loss reached	Conv2 : 0.201
	Conv5 : 0.515
	Fc6 : 0.771
	Fc8 : 0.851

بخش ب

با توجه به نمودارهای شکل ۳ مشاهده می‌شود که پس از ۲۰ ایپاک در بازنمایی لایه دوم به خطای نزدیک به ۰.۲ و در لایه پنجم به خطای نزدیک به ۰.۶ و در لایه ششم و هشتم به خطاهای نزدیک به ۰.۸ رسیده‌ایم. به عبارت دیگر در بازنمایی لایه تمام متصل هشتم به خطای هشت برابری نسبت به بازنمایی لایه کانولوشن دوم رسیده ایم. بنابراین می‌توان نتیجه گرفت که با عمیق تر شدن شبکه خطا افزایش می‌یابد یعنی با افزایش عمق، خروجی هر لایه تفاوت بیشتری با تصویر ورودی پیدا کرده و بازنمایی تصویر ورودی سخت تر می‌شود.



شکل ۳- Loss Plots

بخش ج

شکل ٤- تصاویر بازیابی شده

بخش ه

چنانچه در شکل ۴ قابل مشاهده است:

- **لایه دوم (کانولوشن)** : تصویر کمی تار شده است ولی تفکیک رنگی تصویر به خوبی صورت گرفته است، سوژه و خطوط تصویر قابل تشخیص اند به طوری که با دقت خوبی می‌توان کلاس تصویر را حدس زد
- **لایه پنجم (کانولوشن)** : تصویر تارتر شده است اما همچنان رنگ ها قابل تشخیص هستند همچنین در برخی موارد سوژه و خطوط تصویر نیز قابل رویت بوده و موضوع را می‌توان حدس زد
- **لایه ششم (تماما متصل)** : تصویر خیلی تارتر شده و تغییر زیادی در تصویر بازنمایی شده نسبت به لایه پنجم رخ داده است ولی همچنان محدوده رنگی تصویر اصلی حفظ شده است، مکان سوژه در برخی از تصاویر قابل تشخیص است ولی موضوع سوژه و تصویر قابل حدس نیست
- **لایه هشتم (تماما متصل)** : در لایه آخر تصویر به تارترین حالت خود می‌رسد ولی همچنان طیف رنگی تصویر اصلی حفظ شده و تعداد بسیار کمی از موارد نیز موقعیت مکانی سوژه قابل تشخیص است

نتیجه : بازنمایی تصویر از لایه های کانولوشنی نسبت به بازنمایی از لایه های عمیق تر به مراتب شباهت بیشتری به تصویر اصلی دارد. اگر چه در تمام لایه ها رنگ های تصویر و موقعیت تقریبی سوژه تا حدودی حفظ شده است. تصاویر بازنمایی شده از لایه های تماما متصل همچنان شباهت هایی به تصویر اصلی دارند اما تار شده اند. این بدین معنی است که که ویژگی های خروجی از لایه های عمیق تر و تماما متصل، بر خلاف انتظار نسبت به رنگ و موقعیت مکانی سوژه تغییرپذیری زیادی دارند.