به نام خدا

تشخیص لبه‌های معنایی اجسام با استفاده از مدل‌های عمیق

گزارش روند پیشرفت

۲۴ فروردین ۱۳۹۴

عرفان نوری

تعریف مسئله و کاربردهای تشخیص لبه

مسئله‌ی تشخیص لبه‌ها در تصاویر، یکی از مهم‌ترین مسائل پایه‌ای بینایی ماشین می‌باشد و کاربردهای متعددی نیز دارد. هدف تشخیص لبه‌ها و محیط اشیای مستقل موجود در تصویر است، به همین دلیل به آن مسئله‌ی تشخیص لبه‌های معنایی[۱] گفته می‌شود. روش‌های ابتدایی مانند Canny [1] بر روی ویژگی‌های محلی[۲] تاکید داشتند و با استفاده از ویژگی‌های محلی به تشخیص لبه‌بودن پیکسل‌ها می‌پرداختند. بعد از این خط فکری در مورد مسئله، روش‌های متعددی برای دخالت دادن ویژگی‌های سطح بالاتر برای تشخیص بهتر لبه‌بودن پیکسل‌ها مطرح شد. یکی از مهم‌ترین این روش‌ها، [2] می‌باشد که از همبستگی آماری سطح بالای میان پیکسل‌های متعلق به یک جسم، برای رسیدن به یک نقشه‌ی لبه‌ی معنایی مناسب برای تصویر استفاده می‌کند. روش‌های مبتنی بر شبکه‌های عمیق با ترکیب ویژگی‌های سطح بالا و سطح پایین، به موفقیت‌هایی در زمینه‌ی تشخیص لبه‌های معنایی دست‌یافته‌اند، هر چند هنوز این روش‌ها به مانند مسائل دیگر، از روش‌های غیرمبتنی بر شبکه‌های عمیق، پیشی نگرفته‌اند [3] [4].

مسئله‌ی تشخیص لبه را می‌توان به صورت مسئله‌ی کلاسه‌بندی پیکسلی نیز فرمول‌بندی کرد. در این صورت مسائلی همچون تقسیم‌بندی[۳] نیز در این رده قرار می‌گیرند. در مسائل تقسیم‌بندی، روش‌های مبتنی بر شبکه‌های عمیق، به نتایج بهتری نسبت به روش‌های غیرمبتنی بر شبکه‌های عمیق دست یافته‌اند [5]. می‌توان از ایده‌ها و طرح‌های شبکه‌های عمیق این روش‌ها برای یافتن طرحی مناسب برای مسئله‌ی تشخیص لبه‌های معنایی استفاده کرد.

از خروجی الگوریتم تشخیص لبه می‌توان در مسائلی همچون تشخیص اشیا [6] و تقسیم‌بندی تصویر [7] استفاده کرد.

در زیر خلاصه‌ای از عملکرد روش‌های مختلف آمده است:

دسته‌بندی روش	نام روش	ODS	OIS	AP
	انسان	**0.80**	**0.80**	-
روش‌های غیرمبتنی بر شبکه‌های عمیق	Canny [1]	0.60	0.63	0.58
	gPb-owt-ucm [7]	0.73	0.76	0.73
	Sketch Tokens [8]	0.73	0.75	0.78
	PMI [2]	0.74	0.77	0.78
	SE [9]	0.75	0.77	**0.80**
	MCG [10]	0.75	**0.78**	0.76
روش‌های مبتنی بر شبکه‌های کامل برای تشخیص لبه (راه‌حل دوم)	SCG [11]	0.74	0.76	0.77
	DeepNet [4]	0.74	0.76	0.76
	DeepEdge [3]	0.75	0.77	**0.81**
روش‌های مبتنی بر مدل‌های از پیش آموزش دیده شده (راه‌حل اول)	CSCNN [12]	**0.76**	**0.78**	**0.80**

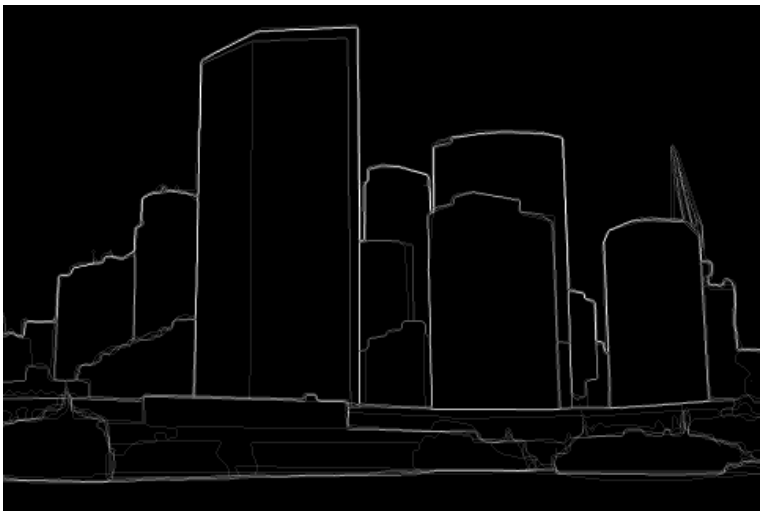
۱ Semantic Contour Detection

۲ Local Features

۳ Segmentation

مجموعه داده

برای این مسئله از مجموعه داده‌ی BSDS500 [7] استفاده خواهد شد. این مجموعه داده دارای ۲۰۰ تصویر آموزش، ۱۰۰ تصویر اعتبارسنجی و ۲۰۰ تصویر آزمون می‌باشد. هر یک از این تصاویر از سه کانال رنگی تشکیل شده‌اند و اندازه‌ی آنها ۳۲۱×۴۸۱ پیکسل می‌باشد. برای همه‌ی این تصاویر تعدادی تصویر لبه توسط متخصصان تهیه شده است. تعداد این تصاویر لبه برای بیشتر تصاویر ۵ عدد می‌باشد، هر چند این مقدار بین ۴ و ۸ متغییر است. به دلیل تفاوت میان تصاویر لبه‌ی به‌دست‌آمده برای تصاویر و همچنین تعدد آنها، از تصویر لبه‌ی میانگین به عنوان تصویر لبه‌ی اصلی هر تصویر استفاده می‌شود و مراحل یادگیری، اعتبارسنجی و آزمون خروجی نیز با استفاده از تصویر لبه‌ی میانگین[۴] انجام می‌گیرد. دیر زیر نمونه‌ای از یک تصویر و همچنین تصویر لبه‌ی میانگین نمایش داده شده است.



معیارهای تشخیص کیفیت

مسئله‌ی تشخیص لبه‌ها را می‌توان به صورت یک مسئله‌ی کلاسه‌بندی فرمول‌بندی کرد که هدف در آن تشخیص پیکسل‌های لبه از پیکسل‌های غیرلبه می‌باشد. بنابراین می‌توان چارچوب معیار دقت-فراخوانی[۵] را با استفاده از لبه‌های نشان‌گذاری شده

---

[۴] mean Contour Map

[۵] Precision-Recall

توسط انسان از مجموعه داده‌های BSDS500 به عنوان حقیقت مبنا[۶] استفاده کرد[13] . استفاده از این بخصوص زمانی اهمیت خود را نشان می‌دهد که کاربردهای تشخیص لبه‌ها را در مسائلی که از لبه‌های تصویر استفاده می‌کنند، مانند تشخیص عمق از دو تصویر stereo یا تشخیص اشیا، در نظر بگیریم. برای محاسبه‌ی این معیار باید مشخص کنیم که چه زمانی لبه‌های تشخیص داده شده صحیح هستند و چه زمانی خطا در تشخیص رخ داده است. هر نقطه بر روی منحنی دقت-فراخوانی با در نظر گرفتن خروجی تشخیص‌دهنده در یک آستانه‌ی مشخص است. تصویر خروجی باید آستانه‌گذاری[۷] شود تا یک تصویر دوگانی[۸] بوجود بیایید. زیرا تصاویر نشان‌گذاری شده توسط انسان نیز تصاویر دوگانی هستند. ابتدا محاسبه‌ی دقت و فراخوانی یک تصویر خروجی الگوریتم در مقابل یک تصویر نشان‌گذاری شده توسط انسان را بررسی می‌کنیم. یک راه‌حل ابتدایی در نظر گرفتن همه‌ی پیکسل‌هایی که در هر دو تصویر به عنوان لبه تشخیص داده شده‌اند و رد کردن بقیه پیکسل‌ها می‌باشد. مشکل این راه‌حل این است که الگوریتم‌هایی که لبه‌های مناسبی تولید می‌کنند ولی مکان آنها دقیق نیست، امتیاز کمی از این معیار کسب می‌کنند. پس لازم است به این نکته توجه شود که در انتخاب معیار مناسب به مسئله‌ی دقیق نبودن مکان لبه‌ها در تصویر خروجی الگوریتم توجه شود، زیرا حتی تصاویر نشان‌گذاری شده توسط انسان نیز این مشکل را نسبت به یکدیگر دارند. به عنوان راه‌حل بهتری برای محاسبه‌ی مقدار معیار، نقشه‌ی لبه‌ی خروجی الگوریتم را با همه‌ی نقشه‌های لبه‌ی نشان‌گذاری شده توسط انسان مقایسه می‌کنیم. اگر پیکسلی که به عنوان لبه تشخیص داده شده است در هیچ یک از نقشه‌های لبه‌ی نشان‌گذاری شده به عنوان لبه نباشد، در این صورت تشخیص آن پیکسل به عنوان لبه نادرست است و آن پیکسل به عنوان یک یقین کاذب[۹] شناخته می‌شود. تعداد اشتراکات در مورد تشخیص لبه‌بودن یک پیکسل نیز در مقایسه با همه‌ی تصاویر نشان‌گذاری شده شمارش می‌شود و میانگین گرفته می‌شود. بنابراین برای اینکه مقدار فراخوانی برای یک الگوریتم ۱ باشد باید اطلاعات همه‌ی تصاویر نشان‌گذاری شده توسط انسان را تولید کند. پس بنابراین اگر خروجی یک الگوریتم $P_b(x,y)$ باشد، یک منحنی دقت-فراخوانی از آن محاسبه می‌کنیم. هر نقطه بر روی این منحنی به طور مستقل با آستانه‌گذاری تصویر خروجی و در نتیجه تبدیل آن به یک تصویر دوگانی محاسبه شده و به دست می‌آید. منحنی دقت-فراخوانی اطلاعات بسیاری را در مورد الگوریتم نمایش می‌دهد. اگر لازم باشد فقط یک مقدار به عنوان معیار گزارش شود، می‌توان مقدار بیشینه‌ی F-measure را محاسبه کرد و آن را گزارش کرد.

راه‌حل‌های ممکن

استفاده از مدل‌های از پیش آموزش دیده شده

در این راه‌حل از مدل‌های از پیش آموزش دیده شده برای عمل کلاسه‌بندی که بر روی مجموعه داده‌های ImageNet [14] انجام گرفته شده است استفاده می‌شود. به این صورت که از مدل‌های آموزش دیده شده برای استخراج ویژگی‌های مناسب بهره برده می‌شود. این مدل‌های عمیق در طول مرحله‌ی آموزش خود بر روی تعداد فراوان تصاویر از مجموعه داده‌های ImageNet به سلسله مراتبی از ویژگی‌ها رسیده‌اند که برای همه‌ی تصاویر ویژگی‌های مناسبی می‌تواند باشد. تنها مشکل در این مورد این موضوع است که این شبکه تبدیل ویژگی‌ها را بر روی کل تصویر انجام می‌دهند، ولی برای کاربرد در این مسئله، لازم است که

---

[۶] Ground truth
[۷] Thresholding
[۸] binary
[۹] False Positive

ترجمه‌های زیر برای این اصطلاحات پیشنهاد می‌شوند:
یقین کاذب: False Positive شک کاذب: False Negative شک راستین: True Negative یقین راستین: True Positive

بتوان برای هر پیکسل از تصویر ورودی، یک بردار ویژگی استخراج کرد. بنابراین لازم است که تغییری در ساختار مدل عمیق مورد استفاده ایجاد کرد و همچنین آن را برای مسئله‌ی مورد نظر بهینه‌تر کرد.

طراحی شبکه کامل برای تشخیص لبه

در این راه‌حل، یک شبکه‌ی عمیق از ابتدا برای مسئله‌ی مورد انتظار طراحی می‌شود و سپس بر روی مجموعه داده‌های موجود آموزش داده می‌شود. برای مسئله‌ی تشخیص لبه‌ها می‌توان یک شبکه‌ی عمیق را برای تبدیل یک تصویر به نقشه‌ی لبه‌ها یا تبدیل ناحیه‌ای از تصویر به پیش‌بینی احتمال لبه‌بودن یک پیکسل طراحی کرد و آموزش داد.

استفاده از شبکه‌ی شرطی خصمانه مولد[۱۰]

در مدل‌های مولد، هدف دستیابی به مدلی است که بتواند توزیع داده‌های ورودی را به دست آورد و نمونه‌هایی از این توزیع تولید نماید. یکی از روش‌های حل این مسئله و دست‌یابی به چنین مدلی، استفاده از روشی مشابه minimax در حل بازی‌ها می‌باشد. به این صورت که یک مدل مولد و یک مدل تفکیک‌کننده[۱۱] در مقابل یک‌دیگر قرار می‌گیرند. هدف مدل مولد اطلاع از توزیع داده‌های ورودی و همچنین تولید داده‌هایی از این توزیع می‌باشد. هدف مدل تفکیک‌کننده نیز تشخیص این می‌باشد که یک داده متعلق به مجموعه داده‌های ورودی می‌باشد یا توسط مدل مولد تولید شده است. اگر هر دو این مدل‌ها را با شبکه‌های عمیق پیاده‌سازی نماییم، می‌توان کل مجموعه‌ی مدل‌ها را با استفاده از backtracking آموزش داد. در نهایت مدل مولد باید بتواند نمونه‌هایی از توزیع واقعی داده‌های ورودی توزیع نماید به طوری که مدل تفکیک‌کننده فقط کاملاً مطابق با شانس بتواند تشخیص دهد که نمونه‌ی داده شده از توزیع مدل مولد است یا یک داده‌ی واقعی است (با احتمال $\frac{1}{2}$)[15] . اگر توزیع داده‌های ورودی را بر روی متغیرهای تصادفی دیگری شرطی نماییم، به مدل شبکه‌ی شرطی خصمانه‌ی مولد خواهیم رسید[16] .

مشکلات پیش رو

مشکل اصلی در رابطه با راه‌حل اول، یافتن مدل‌های از پیش آموزش دیده شده‌ی مناسب است. مدل‌ها باید به گونه‌ای باشند که بتوان ساختار آنها را به گونه‌ای تغییر داد که بردار ویژگی برای هر پیکسل تولید کنند. مدلی مانند AlexNet برای این عمل مناسب می‌باشد، هر چند خطای آن بر روی مجموعه داده‌های ImageNet در حدود ۲۰٪ درصد می‌باشد و مدل‌های دیگری با دقت بیشتری وجود دارند.

مشکل اصلی در رابطه با راه‌حل دوم، طراحی یک شبکه‌ی مناسب برای این مسئله می‌باشد. با وجود اینکه شبکه‌های عمیق نیاز به مهندسی ویژگی‌ها را رفع کرده‌اند، ولی در مقابل باید در طراحی شبکه‌ی مناسب، نکات مهمی را در نظر گرفت. مشکل دیگر کمبود حجم داده‌های آموزش است. تعداد کل تصاویر در مجموعه داده‌های BSDS برای مرحله‌ی آموزش ۲۰۰ تصویر می‌باشد که برای یک شبکه با چند صد میلیون پارامتر بسیار ناچیز است. بنابراین نیاز است در مورد روش‌های ساخت داده‌های مصنوعی و همچنین افزایش داده[۱۲] بررسی‌هایی را انجام داد.

---

۱۰ Conditional Generative Adversarial Nets

۱۱ Discriminative

۱۲ Data Augmentation

مشکل اصلی در رابطه با راه‌حل سوم، بررسی امکان‌پذیری استفاده از ایده‌ی مدل‌های مولد شرطی برای مسئله‌ی تشخیص لبه‌های معنایی می‌باشد. آیا می‌توان با شرطی کردن توزیع لبه‌های یک تصویر بر روی خود تصویر، به نقشه‌ی لبه‌های مناسبی برای تصویر رسید؟

مشکل کلی دیگر که برای همه‌ی سه راه‌حل مشترک است، مسئله‌ی نیاز به توان پردازشی موازی بالا (ترجیحاً با استفاده از GPU) می‌باشد. زیرا توان محاسباتی مورد نیاز برای مراحل آموزشی مدل‌های عمیق بسیار بالا است.

برنامه‌ی پیش رو

- اولویت‌بندی سه راه‌حل مطرح‌شده

فهرست

[1] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* pp. 679-698, 1986.

[2] P. Isola, D. Zoran, D. Krishnan and E. H. Adelson, "Crisp boundary detection using pointwise mutual information," in *Computer Vision--ECCV 2014*, 2014.

[3] G. Bertasius, J. Shi and L. Torresani, "DeepEdge: A Multi-Scale Bifurcated Deep Network for Top-Down Contour Detection," *arXiv preprint arXiv:1412.1123,* 2014.

[4] J. J. Kivinen, C. K. Williams and N. Heess, "Visual boundary prediction: A deep neural prediction network and quality dissection," in *AISTATS*, 2014.

[5] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," *arXiv preprint arXiv:1411.4038,* 2014.

[6] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Computer Vision--ECCV 2014*, Springer, 2014, pp. 391-405.

[7] P. Arbelaez, M. Maire, C. Fowlkes and J. Malik, "Contour Detection and Hierarchical Image Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 33, pp. 898-916, 2011.

[8] J. J. Lim, C. L. Zitnick and P. Dollár, "Sketch tokens: A learned mid-level representation for contour and object detection," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013.

[9] P. Dollár and C. L. Zitnick, "Fast Edge Detection using Structured Forests," in *PAMI*, 2015.

[10] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques and J. Malik, "Multiscale combinatorial grouping," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2014.

[11] R. Xiaofeng and L. Bo, "Discriminatively trained sparse code gradients for contour detection," in *Advances in neural information processing systems*, 2012.

[12] J.-J. Hwang and T.-L. Liu, "Contour Detection Using Cost-Sensitive Convolutional Neural Networks," *arXiv preprint arXiv:1412.6857,* 2014.

[13] D. R. Martin, C. C. Fowlkes and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 26, pp. 530-549, 2004.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *CoRR,* vol. abs/1409.0575, 2014.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014.

[16] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784,* 2014.