



به نام خدا

یادگیری ماشین

گزارش پروژه پایانی

Hough Forest for Object detection

استاد: دکتر ابراهیم پور

دانشجو: ایمان محمدی

دانشگاه تهران

دانشکده فنی مهندسی

دانشکده علوم مهندسی

گروه الگوریتم ها و محاسبات

زمستان 1391

## چکیده

در این گزارش روشی برای تشخیص اشیا در عکس نشان داده شده. این روش از جنگل تصادفی<sup>۱</sup> استفاده می کند؛ و به این صورت عمل می کند که بخش های مختلف عکس<sup>۲</sup> به محل ها گوناگون در عکس رای می دهند. هر رای در واقع احتمال وجود مرکز شیء مورد نظر در آن محل خاص است. و این ماتریس حاصل از رای های گوناگون را عکس هاف<sup>۳</sup> می نامیم. و آن مکان از عکس هاف که بیشترین میزان رای را جمع آوری کرده بعنوان مرکز شیء تشخیص داده می شود.

## مقدمه

در ابتدا جنگل های تصادفی و جنگل هاف<sup>۴</sup> را معرفی می کنیم، و سپس نحوه ساخت جنگل هاف را نشان می دهیم و در آخر نحوه تشخیص شیء در عکس با استفاده از جنگل هاف توضیح داده شده است.

## جنگل های تصادفی

جنگل های تصادفی [1, 2] برای بسیاری از مسائل طبقه بندی استفاده می شود. یک جنگل تصادفی شامل مجموعه ای از درخت های تصمیم گیری دودویی<sup>۵</sup> می باشد [19]. در مرحله آموزش<sup>۶</sup> یک تابع دودویی<sup>۷</sup> به هر گره ی بدون برگ از درخت نسبت داده می شود، که این تابع قابل اعمال روی هر یک از نمونه داده های مسئله می باشد. با توجه به نتیجه اعمال تابع دودویی روی یک نمونه داده مسئله، تصمیم گرفته می شود که داده به یکی از دو فرزند آن گره ی غیر برگ فرستاده شود. با این روش هر داده از ریشه درخت شروع کرده و به همین ترتیب در درخت پایین می رود تا به یکی از گره های برگ درخت می رسد.

جنگل های تصادفی بصورت نظارت شده<sup>۸</sup> آموزش می بینند. آموزش به این صورت است که در هنگام ساخت درختان اطلاعاتی در هر گره برگ ذخیره می شود، و این اطلاعات از داده هایی که هنگام آموزش به

---

<sup>1</sup> Random forest

<sup>2</sup> Patch appearance

<sup>3</sup> Hough Image

<sup>4</sup> Hough forest

<sup>5</sup> Binary Decision trees

<sup>6</sup> Training

<sup>7</sup> Binary Test

<sup>8</sup> Supervised

آن برگ رسیده اند جمع آوری می شود. مانند نسبت تعداد داده های هر کلاس که به آن برگ رسیده اند و غیره.

در زمان اجرا نمونه تست<sup>1</sup> از تمام درختان جنگل به سمت برگها عبور داده می شود، و خروجی در واقع میانگینی از اطلاعات برگهایی است که آن داده تست در درختان مختلف به آن ها رسیده است.

در [1, 2] نشان داده شده است که استفاده از چندین درخت که بصورت تصادفی آموزش دیده اند، خصوصیت جامعیت<sup>2</sup> و ثبات<sup>3</sup> بیشتری نسبت به یک درخت تصمیم گیری قطعی<sup>4</sup>، دارند. آموزش تصادفی به به این علت نامگذاری شده که اولاً، برای ساخت هر درخت، زیر مجموعه ای تصادفی از داده های آموزشی<sup>5</sup> استفاده شده است. ثانیاً، زیرمجموعه ای تصادفی از کل توابع ممکن دودویی برای ساخت هر گره ی غیره برگ در نظر گرفته شده است. که از بین این مجموعه بهترین آنها برای آن گره غیره برگ انتخاب می شود. معیار بهترین بودن در اینجا یعنی تابع دودویی انتخاب شده بتواند به بهترین شکل داده های رسیده به آن گره غیره برگ را بین دو فرزندش تقسیم کند، و با توجه به مسئله های خاص تقسیم بندی خوب می تواند تعریف خود را داشته باشد.

جنگل تصادفی که برای ساخت عکس هاف و تشخیص اشیا در عکس استفاده می شود را، جنگل هاف می نامیم.

## ساخت جنگل هاف

در جنگل هاف هر درخت  $T$  بر مبنای مجموعه ای از پچ ها<sup>6</sup>  $\{P_i = (I_i, C_i, d_i)\}$  ساخته می شود. که  $I_i$  تصویر پچ می باشد،  $C_i$  برچسب آن پچ است و  $d_i$  آفست پچ را مشخص می کند. هریک از این مولفه ها را به تفسیر در ادامه توضیح می دهیم.

پچ های آموزشی<sup>7</sup> در واقعه از مجموعه ای از عکس های آموزشی<sup>1</sup> بدست می آیند. بعضی از این عکس های آموزشی از زمینه<sup>2</sup> جمع آوری شده اند، که در این صورت پچ متناظر آنها دارای مقدار 0 در مولفه

---

<sup>1</sup> Test sample

<sup>2</sup> generalization

<sup>3</sup> Stability

<sup>4</sup> Single deterministic decision tree

<sup>5</sup> Training data

<sup>6</sup> Patches

<sup>7</sup> Training Patches

برچسب خود می باشد ( $C_i = 0$ ). و بعضی دیگر از داخل مرز محدود<sup>۳</sup> اشیا جمع آوری شده اند، که در این صورت پچ متناظر آنها دارای مقدار 1 در مولفه برچسب خود می باشد ( $C_i = 1$ ). هر پچ شامل یک آفست (برداری با دو مولفه) از مرکز پچ تا مرکز شی می باشد. البته این آفست برای پچ های با  $C_i = 0$  تعریف نشده می باشد. برای هر گره برگ در درخت، اطلاعاتی که ذخیره می شوند دو دسته هستند، نسبت پچ های رسیده به آن برگ با برچسب 0 یا 1، این نسبت را در برگ ذخیره می کنیم و با  $C_L$  نشان می دهیم، که عددی بین 0 و 1 است. اگر  $C_L = 0$  باشد یعنی تنها پچ های مربوط به زمینه ی تصویر به آن برگ رسیده اند. دسته دیگر از اطلاعاتی که در برگ ذخیره می شود، آفست پچ ها می باشد. تمام این آفست ها در یک مجموعه بنام  $D_L$  در برگ ذخیره می شوند. مسلماً این آفست ها تنها از پچ های با برچسب 1 جمع آوری شده اند. در زمان اجرا از این اطلاعات برای رای دادن به نقاط مختلف تصویر بعنوان کاندیدای مرکز شی بودن، استفاده می شود.

در هنگام یادگیری به هر گره ی غیره برگ یک تابع دودویی نسبت داده می شود. این تابع به هر تصویر پچ  $I_i$ ، اعمال می شود، در هر دو زمان آموزش و تست، پچ شامل یک تصویر با سایز ثابت  $18 \times 18$  پیکسل می باشد. و این تصویر با اعمال فیلتر های مختلف به چند تصویر تبدیل شده و در پچ ذخیره شده است. اگر تصویر از  $C$  کانال مختلف فیلتر شده باشد، مجموعه  $C$  تصویر برای پچ  $i$  ام را بصورت  $I_i = \{I_i^1, I_i^2, \dots, I_i^C\}$  نشان داده می شود. که هر  $I_i^a$  یک تصویر  $18 \times 18$  می باشد. تابع دودویی روی تصویر پچ  $t(I) \rightarrow \{0, 1\}$  می تواند به صورت های مختلف تعریف شود. در اینجا این تابع به این صورت تعریف شده است. پارامترهای تابع  $a \in \{1, 2, \dots, C\}$  که نمایانگر کانال مورد استفاده در تابع،  $(p, q)$  و  $(r, s)$  که موقعیت مکانی دو نقطه در تصویر پچ و  $h$  مقداری به عنوان آستانه، در نظر گرفته می شوند. تابع دودویی با پارامترهایش بصورت زیر تعریف می شود.

$$t_{a,p,q,r,s,h}(I) = \begin{cases} 0, & \text{if } I^a(p, q) < I^a(r, s) + h \\ 1, & \text{otherwise} \end{cases}$$

این تابع به سادگی مقدار دو پیکسل را در یک کانال مشترک به همراه یک مقدار آستانه مقایسه می کند.

<sup>1</sup> Training Images

<sup>2</sup> Background

<sup>3</sup> Bounding Box

## ساخت درخت

هر درخت با استفاده از یک متد بازگشتی ساخته می شود. ساخت درخت با ریشه شروع می شود. در طول ساخت درخت به هر گره مجموعه ای از پیچ های آموزشی می رسد. اگر عمق گره به حد مشخصی برسد (در اینجا عمق درخت 16 در نظر گرفته شده)، یا تعداد پیچ های رسیده به گره از یک حدی کمتر باشد (در اینجا کمترین تعداد پیچ ها 30 در نظر گرفته شده)، گره ی جاری برگ در نظر گرفته می شود، و اطلاعاتی که برای رای دادن به عکس هاف مورد نیاز است (CL, DL) با توجه به پیچ هایی که به این گره رسیده است، در این گره جمع آوری شده و ذخیره می شود. در غیره این صورت گره بعنوان غیره برگ در نظر گرفته شده، و تابع دودویی مناسب از بین استخر بزرگی از توابع دودویی، برای این گره انتخاب می شود. و مجموعه پیچ هایی که به این گره رسیده براساس این تابع دودویی بین دو فرزند گره تقسیم میشوند. در اینجا استخر توابع دودویی شامل 1000 تابع دودویی می باشد که بصورت تصادفی ایجاد شده اند.

نکته اساسی انتخاب مناسب تابع دودویی از استخر می باشد. برای این منظور باید ارزیابی مناسبی از تقسیم بندی پیچ ها توسط تابع دودویی داشته باشیم. این ارزیابی می تواند براساس دو معیار باشد. یکی از این دو معیار اینست که برچسب پیچ های تقسیم شده خالص تر شده باشند. و معیار دیگر آنست که آفست پیچ ها تقسیم شده خالص تر شده باشند. برای ارزیابی از لحاظ خالص تر شدن برچسب از معیار زیر استفاده می کنیم.

$$U_1(A) = |A| \cdot \text{Entropy}(\{C_i\})$$

که در آن A مجموعه پیچ هایی میباشد که باتوجه به تابع دودویی برای یکی از فرزندان چپ یا راست در نظر گرفته شده اند. و  $\{C_i\}$  مجموعه ای از 0ها و 1ها میباشد و هر عضو آن برچسب یکی از اعضای A می باشد. و میانگین این 0ها و 1ها را با  $c \in [0,1]$  نشان می دهیم. و Entropy بصورت زیر بدست می آید.

$$\text{Entropy}(\{C_i\}) = -c \cdot \log c - (1-c) \cdot \log(1-c)$$

معیار بعدی که خالص بودن آفست ها را ارزیابی می کند بصورت زیر محاسبه می شود.

$$U_2(A) = \sum_{i:c_i=1} (d_i - d_A)^2$$

$d_A$  میانگین مجموعه آفست های پیچ های عضو A می باشد. و  $d_i$  آفست هر پیچ عضو A است. مسلماً در این معیار پیچ هایی که برچسب 0 دارند، در نظر گرفته نمی شوند.

حال با استفاده از این دو معیار برای انتخاب مناسب تابع دودویی برای گره غیره برگ به اینصورت عمل می کنیم. ابتدا با انتخاب تصادفی پارامترهای تابع دودویی  $(a,p,q,r,s,h)$  ، استخری از توابع دودویی ایجاد می کنیم. سپس بصورت تصادفی تصمیم گرفته می شود که آیا تابع دودویی قرار است براساس خالص تر کردن پیچ ها از نظر برجسب انتخاب شود و یا خالص تر کردن از نظر آفست قرار است انتخاب شود. این تصمیم گیری با احتمال  $\frac{1}{2}$  تصمیم میگیرد، اما اگر تعداد پیچ های با برجسب 0 کم باشد مثلاً 5٪ کل پیچ های آن گره ، آنگاه فقط از معیار خالص تر کردن آفست ها استفاده می کنیم. در آخر تابع دودویی را انتخاب می کنیم که مقدار معیار ارزیابی در هر دو مجموعه پچی که این تابع تقسیم کرده است کمینه شود.

$$\text{Min}_k \{ U_*(p_i | t^k(I_i)=0) + U_*(p_i | t^k(I_i)=1) \}$$

که در آن \* باتوجه به تصمیم گیری تصادفی، 1 یا 2 می باشد. و  $t^k$  یکی از توابع دودویی از بین k تابع دودویی استخر است.  $\text{Min}_k$  مقدار کمینه از بین k تابع دودویی استخر را پیدا می کند.

با استفاده از این معیار های ارزیابی که در گره های مختلف درخت بصورت تصادفی استفاده شده اند. تضمین می شود که پیچ ها هنگام عبور از ریشه بسمت برگ ها رفته رفته از لحاظ برجسب و آفست خالص تر می شوند.

## منابع

1. Amit, Yali, and Donald Geman. "Shape quantization and recognition with randomized trees." *Neural computation* 9.7 (1997): 1545-1588.
2. Breiman, Leo. "Random forests." *Machine learning* 45.1 (2001): 5-32.
3. Gall, Juergen, and Victor Lempitsky. "Class-specific hough forests for object detection." *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009.
4. Gall, Juergen, et al. "Hough forests for object detection, tracking, and action recognition." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33.11 (2011): 2188-2202.