**Arbitrary Shape Scene Text Detection with Adaptive Text Region Representation**

روش پیشنهادی در این مقاله دارای دو مرحله هست که این دو مرحله شامل:

1. پیشنهادات متنی (Text Proposal)
2. بهبود پیشنهادات (proposal refinement)

در مرحله ی اول از Text\_RPN استفاده می‌شود که پیشنهاد‌های متن از تصویر ورودی را تولید می‌کند. ضمنا نقشه های ویژگی تصویر ورودی در اینجا بدست می آید که در مراحل بعدی از آن استفاده می شود. در مرحله‌‌ی دوم، پیشنهادات متنی از طریق یک شبکه تأیید و تصحیح می‌شود. در این مرحله، طبقه بندی متن / غیر متن ، رگرسیون کادر محدوده و منطقه‌ی تطبیق پذیر با متن که حاصل یک شبکه ی RNN است؛ گنجانده شده است.. سرانجام ، خروجی این روش، تشخیص مکان متن در تصویر ورودی است که با چند ضلعی هایی در تصویر مشخص شده اند.

ایده ای که در این مقاله مطرح می شود این است که برای بازنمایی مکان متن در تصویر، منطقی است که از چند ضلعی هایی استفاده شود که نقاط آن متغیر و متناسب با شکل و محدوده ی متن باشد. یک کار ساده این است که از نقاط گوشه ای موجود بر روی مرز متن برای نمایش مکان متن استفاده کنیم. اما مشکلی که وجود دارد این است که یادگیری و پیدا کردن این نقاط مشکل است. برای حل این مشکل، مسئله را این گونه در نظر می گیرند که تقریبا مرز بالا و پایین یک متن با یکدیگر تقارن دارند و کافی است ما زوج نقاطی از مرز بالا و پایین متن را بدست آوریم تا بتوان مکان متن را با یک چند ضلعی نشان داد. اکنون به شرح کامل دو مرحله روش پیشنهادی این مقاله می پردازیم.

**پیشنهادات متنی (Text Proposal)**

همانطور که قبلا هم اشاره کردیم، در این مرحله برای بدست آوردن پیشنهادات مکان متن از شبکه ی Text\_RPN استفاده می شود که مشابه ی شبکه RPN در Faster R\_CNN است با این تفاوت که شبکه ی اصلی آن و سایز anchor ها متفاوت است. شبکه ی اصلی Text\_RPN شبکه ی SE\_VGG16 است که در واقع همان VGG16 است که بلاک های squeeze-and- excitation به آن اضافه شده است. علت استفاده از بلوک های SE این است که بلوک های SE با در نظر گرفتن وابستگی بین کانال ها و مدل سازی آن، ویژگی های کانال را مجدداً ارزیابی می کنند ، که به طور قابل توجهی باعث بهبود عملکرد می شود.

**بهبود پیشنهادات (proposal refinement)**

در این مرحله، از پیشنهاداتی که در مرحله ی قبل از تصویر ورودی تولید شده اند، به عنوان ورودی به یک شبکه ی تصحیح داده می شود. این شبکه دارای سه شاخه است. شاخه اول طبقه بندی می کند که این کاندید متن هست یا نه. شاخه دوم محدوده ی متن را خروجی می دهد و در نهایت شاخه ی سوم چند ضلعی که مکان متن را بازنمایی می کند را خروجی می دهد. دو شاخه ی اول همانند متد های قبلی موجود در این زمینه هستند. ورودی شاخه ی سوم ویژگی های هستند که بعد از اعمال ROI pooling بر روی نقشه ی ویژگی های شبکه ی SE\_VGG16 بدست می آید. خروجی این شاخه هم نقاط مرزی هستند که مکان متن را نشان می دهند. چون تعداد این نقاط برای هر متن متفاوت است، منطقی است که از یک شبکه ی RNN استفاده شود که در اینجا از شبکه ی LSTM استفاده می شود که یک شبکه ی معروف از RNN هاست. همان طور که قبلا توضیح دادیم، برای راحتی، نقاط را به صورت زوج هایی که بر روی مرز بالا و پایین متن قرار دارند در نظر می گیریم که به صورت (xi, yi, xi+1, yi+1) در نظر می گیریم. ورودی همه ی گام های زمانی در LSTM، ویژگی های بدست آمده بعد از اعمال ROI Pooling هستند و خروجی هر یک از این گام های زمانی یک زوج نقطه از نقاط مرزی هستند. چون تعداد نقاط مرزی برای هر متن متفاوت است از یک برچسب stop استفاده می کنیم تا زمان توقف را به شبکه اعلام کنیم. بدلیل اینکه پیش بینی برچسب stopیک مسئله ی طبقه بندی هست و پیش بینی مختصات نقاط مرزی یک مسئله ی رگرسیون است؛ پس بهتر است که هر گام زمانی در شبکه ی LSTM دو شاخه خروجی بدین منظور داشته باشد. در نهایت همانند سایر روش های تشخیص اشیا، در این روش هم لازم است که از روش NMS استفاده کنیم. که در اینجا از NMS معمولی نمی توانیم استفاده کنیم و باید از Polygon NMS استفاده کنیم که براساس مساحت چند ضلعی نشان دهنده ی مکان متن محاسبه می شود.

**Training objective**

گفتیم که شبکه یText\_RPN مشابه ی شبکه RPN در faster R\_CNN است و خطای آموزش آن هم مشابه همان محاسبه می شود. اما باید برای شبکه ی تصحیح پیشنهادی خطای آموزش را محاسبه کنیم که برابر است با مجموع خطای طبقه بندی متن/ غیر متن، خطای رگرسیون کادر محدوده، خطای رگرسیون نقاط مرزی و خطای طبقه بندی برچسب توقف/ ادامه. بنابراین می توان نوشت:

که در عبارت بالا ، و پارامتری هایی هستند برای ایجاد توازن بین ترم های ای خطا که در اینجا همگی برابر 1 قرار گرفته اند.

در ترم مربوط به خطای طبقه بندی متن / غیر متن، پارامتر نشان دهنده ی برچسب کلاس است که اگر کلاس متن باشد مقدار آن یک و اگر پس زمینه باشد مقدار آن صفر می شود. پارامتر هم نشان دهنده ی احتمال کلاس متن و پس زمینه، پس از اعمال تابع softmax است. بنابراین مقدار عبارت برابر است با یعنی منفی لگاریتم خطای کلاس درست است.

در ترم مربوط به خطای رگرسیون کادر محدوده پارامتر نشان دهنده ی کادر محدوده ی درست است و نشان دهنده ی مختصات کادر محدوده ی پیش بینی شده است.

ادامه شو باید بنویسیم

**Efficient and Accurate Arbitrary-Shaped Text Detection with Pixel Aggregation Network**

**د**ر این مقاله یک شبکه به نام PAN ارائه شده است که می تواند متن ها با شکل های دلخواه را در تصویر تشخیص دهد. این شبکه شامل دو مرحله می شود:

1. پیش بینی مکان متن ، kernel و بردارهای شباهت توسط یک شبکه segmentation.
2. بازسازی نمونه های کامل متن از kernel های پیش بینی شده.

در این مقاله بیان می کند که برای اینکه بهره وری را بالا ببریم؛ لازم است که زمان این دو مرحله را کاهش دهیم. بدین منظور یک شبکه ی سبک به عنوان شبکه اصلی و segmentation انتخاب می کنند. در اینجا از شبکه یResNet18 استفاده می کنند. اما استفاده از شبکه ی سبک دو ایراد دارد. اول اینکه در استخراج ویژگی ضعیف عمل می کند و دوم اینکه ویژگی های بدست آمده دارای receptive field کمی است و قابلیت بازنمایی ضعیفی دارد. برای حل این مشکل پس از شبکه ی اصلی، دو ماژول قرار داده اند:

1. Feature Pyramid Enhancement Module(FPEM)
2. Feature Fusion Module(FFM)

FPEM یک ماژول U شکل است که با استفاده از کانولوشن های جداکننده ساخته شده است. بنابراین FPEM با ترکیب ویژگی های سطح پایین و بالا می تواند ویژگی های مقیاس های مختلف را بهتر کند. علاوه بر این چون FPEM به صورت آبشاری است می تواند سبک بودن شبکه اصلی را جبران کند. پس از FPEM شبکه ی FFM قرار می گیرد که ویژگی های ایجاد شده توسط FPEM را با یکدیگر ادغام می کند. در آخر، برای بازسازی نمونه های کامل متن به طور دقیق ، از روش Pixel Aggregation (PA) استفاده می شود که می تواند پیکسل های متن را برای تصحیح kernel از طریق بردارهای شباهت پیش بینی شده راهنمایی کند.

به صورت خلاصه ،contribution های این مقاله سه تا می باشد.

1. استفاده FPEM و FFM که هر دو باعث می شوند تا ویژگی های بهتری بدست آورند.
2. استفاده از pixel aggregation .
3. روش پیشنهادی این مقاله توانسته است عملکرد بهتری نسبت به روش های قبلی بر روی دو دیتاست \* و \* شود که شامل تصاویری با متن های خمیده هستند . ضمناً inference time این روش نسبت به روش های قبلی تغییر نکرده است و روش پیشنهاد شده در این مقاله مکان متن ها را در تصویر به صورت real time انجام می دهد.

**Feature Pyramid Enhancement Module(FPEM)**

FPEM یک ماژول U شکل است که شامل دو مرحله است:

1. Up-scale Enhancement
2. Down-scale Enhancement

در بخش اول بهبود به صورت تکراری بر روی ویژگی های ورودی با گام های 32، 16، 8 و 4 اعمال می شود و خروجی این بخش که هرمی از ویژگی ها است را به عنوان ورودی به مرحله ی بعدی می دهد و این بار بهبود را برعکس بخش قبل از گام 4 تا 32 انجام می دهد. به منظور ساخت قمست اتصال در ماژول FPEM از کانولوشن های جداکننده به جای کانولوشن های معمولی استفاده شده است که می تواند با استفاده از محاسبات کمی receptive field را افزایش دهد (به علت وجود depth-wise کانولوشن های 3\*3) و همچنین می تواند شبکه را عمیق کند(به علت وجود کانولوشن های 1\*1).

اگر بخواهیم ماژول FPEM را با ماژول FPN مقایسه کنیم؛ می توانیم یک شباهت و دو تفاوت را برای این دو ماژول نام ببریم. شباهت این دو در این است که هر دوی آن ها می توانند با ترکیب ویژگی های سطح پایین و سطح بالا، ویژگی های بهتری را بدست آورند. اما تفاوت آن ها در این است که FPEM یک ماژول آبشاری است که با افزایش تعداد این ماژول ها در آبشار ()می تواند ویژگی ها را در مقیاس های مختلف به صورت مناسب تری ترکیب کند و همچنین receptive field را افزایش دهد. تفاوت دوم در این است که از نظر هزینه محاسباتی ماژول FPEM ارزان تر است و علت آن هم استفاده از کانولوشن های جدا کننده است.

**Feature Fusion Module(FFM)**

این ماژول قرا است خروجی مرحله ی قبل که تا هرم ویژگی است یعنی که عمق های مختلفی نیز دارند را با یکدیگر ترکیب کند. ساده ترین روشی که برای ترکیب این هرم ویژگی ها به ذهن می رسد این است که همه ی ویژگی ها را upsample کنیم و سپس آن ها را با هم concatenate کنیم. اما این روش یک ایراد دارد و ایراد آن هم این است که تعداد کانال های خروجی زیاد می شود. راه حل این مشکل این است که ویژگی هایی که از نظر مقیاس با یکدیگر متناظر هستند به صورت دو به دو بهم اضافه کنیم و سپس عمیات upsampling و concatenation را انجام دهیم. در این صورت تعداد کانال ها از 4\*128\* به 4\*128 تا کاهش می یابد.

**Pixel Aggregation**

**همانطور** که می دانید معمولا برای یک متن در تصویر چندین کاندید نزدیک بهم بدست می آید که ما باید از این کاندید ها یکی را انتخاب کنیم. که در اینجا برای این کار از روش pixel aggregation استفاده می کنیم. که ایده ی این روش استفاده از kernel است که به خوبی نشان دهنده ی یک نمونه خاص از متن در تصویر است. اما یک kernel یک کاندید کامل از متن نیست. بنابراین لازم است که کاندید کامل متن را از روی kernel بازسازی کنیم و برای این کار باید پیکسل های موجود در کاندید های متن را با kernel ادغام کنیم. در این ایده از روش clustering الهام گرفته شده است که هر نمونه از متن یک cluster است و kernel آن نمونه از متن، مرکز این cluster می باشد. پیکسل های کاندید های متن هم نمونه هایی هستند که باید cluster شوند. بنابراین فاصله ی پیکسل های متناظر با یک kernel بایدکوچک باشد. در زمان آموزش، خطای تجمیع به صورت زیر تعریف می شود:

که N تعداد نمونه های متن است و برابر است با iامین نمونه از متن. هم نشان دهنده ی فاصله ی پیکسل p تا کرنل که مربوط به نمونه متن است. یک عدد ثابت است که به صورت تجربی در اینجا برابر 0.5 قرار داده شده است. بردار مشابهت برای پیکسل p و بردار مشابهت برای کرنل است که به صورت زیر محاسبه می شود.:

علاوه بر خطای تجمیع لازم است که مراکز cluster ها به متمایز باشند به همین دلیل خطای دیگری به نام خطای تمایز نیز تعریف می کنیم که به صورت زیر بیان می شود:

سعی می کند که فاصله ی مراکز cluster ها کمتر از نباشد که در اینجا به صورت تجربی مقدار آن را برای همه ی آزمایش ها برابر 3 قرار داده شده است.

در مرحله آزمایش ، از بردارشباهت پیش بینی شده برای هدایت پیکسل ها در کاندید متن به کرنل مربوطه استفاده می کنیم. بنابراین به صورت دقیق مراحل post processing به شرح زیر است:

* 1. یافتن مؤلفه های متصل در kernel’s segmentation result که هر کدام از این مولفه های متصل یک kernel جداگانه محسوب می شوند.
  2. برای هر کرنل ، پیکسل همسایه p در کاندید متن را ادغام می کند در حالی که فاصله اقلیدسی از بردارهای شباهت آنها کمتر از d است.
  3. تکرار مرحله 2تا زمانی که هیچ پیکسل همسایه واجد شرایط نباشد.

**Loss function**

**Character Region Awareness for Text Detection**

روشی که در این مقاله پیشنهاد شده است به این صورت است که مکان هر کاراکتر را به صورت جداگانه تشخیص دهد و سپس با ترکیب کاراکتر های تشخیص داده شده، مکان متن را مشخص کنند. اسم این روش را CRAFT نامیدند. خوبی این روش این است که می تواند همه اشکال متن، متن های منحی شکل و بد شکل را تشخیص دهد.

این روش از یک شبکه کانوولوشنی استفاده می کند که دو امتیاز را خروجی می دهد. یکی امتیاز ناحیه هر کاراکتر می باشد(region score) و دیگری امتیاز نزدیکی کاراکترها(affinity score). امتیاز ناحیه ای هر کاراکتر، مکان هر کاراکتر را مشخص می کند و امتیاز نزدیکی دو کاراکتر برای گروه بندی هر کاراکتر در یک نمونه یکتا استفاده می شود. برای جبران مشکل مشخص نبودن مکان کاراکترها در دیتاست اصلی، این مقاله یک یادگیری weakly-supervised را برای تخمین مکان صحیح کاراکترها با استفاده از مکان واقعی کلمات مشخص شده در دیتاست پیشنهاد می دهد.

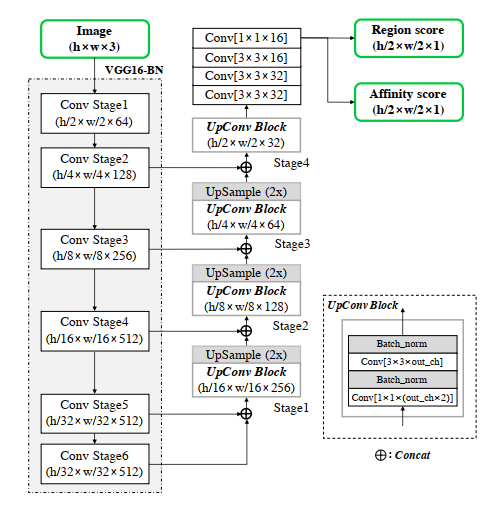
**Methodology**

همانطور که گفته شد، در این مقاله برای تشخیص مکان کاراکتر ها از یادگیری weakly-supervised استفاده می شود که در ادامه به توضیح ساختار آن می پردازیم.

**Architecture**

شبکه پایه این روش، یک شبکه کاملا کانوولوشنی بر اساس VGG-16 می باشد. این شبکه دارای skip connection در قسمت decoding می باشد که شبیه شبکه U-Net، ویژگی های سطح پایین را نیز در نظر می گیرد. خروجی این شبکه دارای دو کانال است که شامل دو امتیاز، region score و affinity score می باشد.

ساختار این شبکه را می توان در زیر مشاهده کرد:



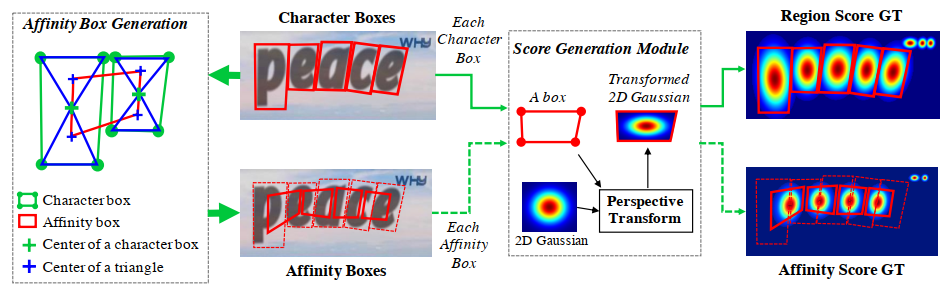
**Training**

یادگیری این روش دارای دو بخش زیر است :

1. Ground Truth Label Generation

در این بخش به ازای هر عکس، باتوجه به region score و affinity score، برای bounding box هر کاراکتر یک برچسب تولید می شود. region score احتمال مرکز کاراکتر بودن هر پیکسل را نشان می دهد و affinity score برای هر پیکسل، احتمال اینکه مرکز بین دو کاراکتر مجاور باشد را بازنمایی می نماید.

مراحل تولید برچسب را در شکل زیر می توان دید. احتمال ها در این شکل به صورت heatmap گوسی نمایش داده شده اند.



محاسبه تابع توزیع گوسی به ازای هر پیکسل در bounding box، بسیار زمانبر است.

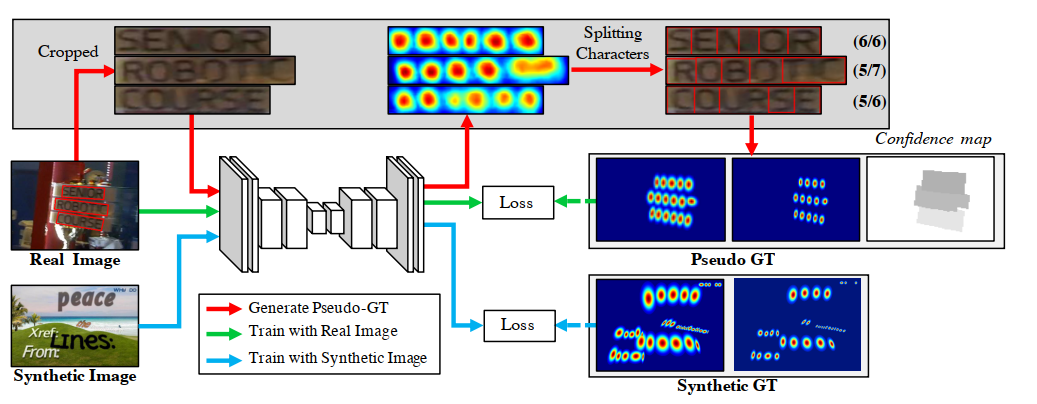
برای تخمین زدن و تولید ground truth برای هر دو امتیاز region score و affinity score، از سه مرحله استفاده می شود :

1. آماده سازی یک نقشه گوسی دو بعدی
2. محاسبه یک تبدیل perspective بین نقشه گوسی ناحیه ای و هر box کاراکتر
3. نگاشت هر توزیع گوسی به box هر ناحیه

...

1. Weakly-Supervised Learning

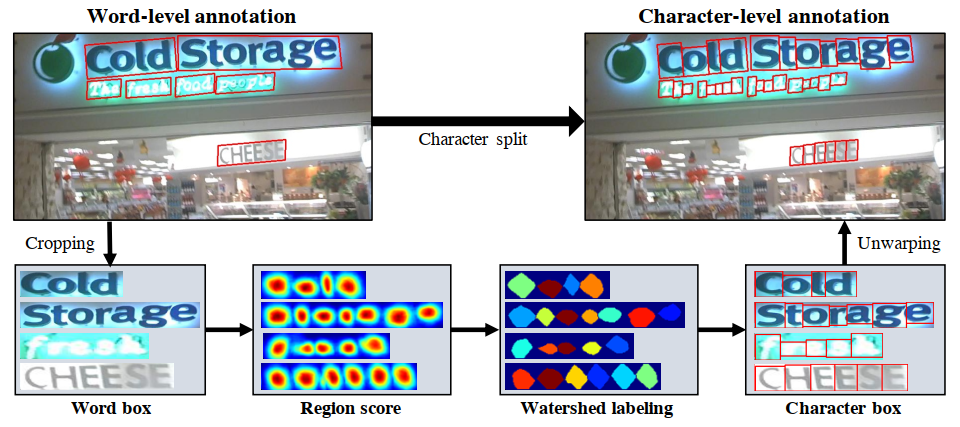
در این بخش، از تصاویر واقعی که در آن ها فقط box متن ها مشخص شده است، استفاده کرده و box کاراکتر های آن متن را به دست می آورد. با این یادگیری، با استفاده ازregion score که برای هر بخش از تصویر بریده شده متن به دست می آید، می توان bounding box را برای هر کاراکتر در تصاویر واقعی به عنوان برچسب تولید کرد. مراحل این بخش در تصویر زیر نشان داده شده است:



روند جداسازی کاراکترها دارای 4 مرحله است:

1. بخش های دارای متن از تصویر ورودی بریده می شود.
2. مدل برای پیش بینی region score آموزش می بیند.
3. الگوریتم watershid برای جداسازی ناحیه های کاراکتر استفاده می شود.
4. مختصات box کاراکترها با استفاده از تبدیل معکوس مرحله جداسازی، بر روی تصویر اصلی مشخص می شود.

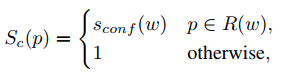
این مراحل را در تصویر زیر می توان مشاهده کرد :



به ازای هر نمونه w از دیتاست آموزشی، R(w) و l(w) به ترتیب نشان دهنده ناحیه bounding box هر کلمه در آن و طول آن کلمه می باشد. در طول روند جداسازی کاراکترها، می توان bounding box و طول متناظر هر کاراکتر (lc (w)) را به دست آورد. سپس Sconf(w) را که confidence score برای هر نمونه است به شکل زیر به دست آورد :



سپس به ازای هر پیکسل هر تصویر ورودی confidence map را که Sc معرفی می کنیم، به شکل زیر محاسبه می نماییم:



که در آن p هر پیکسل از ناحیه R(w) می باشد.

و در نهایت Loss به شکل زیر تعریف می شود :



که در آن Sr\*(p) و Sa\*(p) به ترتیب نشان دهنده psudo-ground truth دو امتیاز region score و affinity score می باشد و Sr(p) و Sa(p) به ترتیب نشان دهنده region score و affinity score پیش بینی شده هستند.

حال که شبکه آموزش داده شد نیاز است که bounding box مربوط به هر کلمه مشخص گردد. بنابراین نیاز است مراحل زیر انجام شود :

1. یک نگاشت باینری به نام M تعریف می شود که اگر Tr مرز region باشد و Ta مرز affinity، در صورتی 1 می شود که یا در غیر این صورت صفر است.
2. برچسب گذاری قسمت به هم متصل انجام می گیرد.(CCL)
3. به دست آوردن quidbox به وسیله چرخش مستطیلی با کم ترین اندازه ای که محصور به ناحیه بهم متصل باشد برای هر برچسب

یک خوبی روش CRAFT این است که هیچ post-processing ای مثل NMS ندارد. همچنین در این روش یک چند ضلعی محصور برای هر متن موجود در تصویر می توان تولید کرد و فقط به وسیله مستطیل نیست.