

## Context-Aware Probabilities (CAP) for Semantic Image Segmentation

Majid Nasiri<sup>1</sup>, M.Sc., Hamidreza Rashidy Kanan<sup>2\*</sup>, Assistant Professor, Seyed Hamid Amiri<sup>3</sup>, Assistant Professor

1- Department of Computer Engineering, Shahid Rajaei Teacher Training University, Tehran, Iran.

2\*- Department of Computer Engineering, Shahid Rajaei Teacher Training University, Tehran, Iran.

3- Department of Computer Engineering, Shahid Rajaei Teacher Training University, Tehran, Iran.

<sup>1</sup>majid.nasiri@sru.ac.ir, <sup>2</sup>h.rashidykanan@sru.ac.ir, <sup>3</sup>s.hamidamiri@sru.ac.ir

\*Corresponding author address: Hamidreza Rashidy Kanan, Department of Computer Engineering, Shahid Rajaei Teacher Training University, Tehran, Iran.

**Abstract-** Semantic image segmentation based on Convolutional Neural Networks (CNNs) is one of the main approaches in computer vision area. The methods based on deep convolutional neural networks, typically use a pre-trained CNN trained on the large image classification datasets as a backend to extract features (image descriptors) from the images. Whereas, the special size of output features from CNN backends are smaller than the input images, by stacking multiple deconvolutional layers on the last layer of backend network, the dimension of the output will be the same as the input image. Segmentation using local image descriptors without involving relationships between these local descriptors yield weak and uneven segmentation results. Inspired by these observations, in this research we propose Context-Aware Probabilities (CAP) unit. CAP unit generates probabilities for classes using local-image descriptors. This unit can be used in any semantic image segmentation architectures. We used CAP unit in Fully Convolutional Network (FCN) and DeepLab-v3-plus architectures and propose the new FCN-CAP and DeepLab-v3-plus-CAP architectures. Training the proposed architectures on PASCAL VOC2012 dataset shows 1.9% and 0.4% accuracy improvement compared to the corresponding basic architectures, respectively.

**Keywords-** Semantic Image Segmentation, Deep Neural Network (DNN), Convolutional Neural Network (CNN), Context-Aware Probabilities (CAP) Unit.

## احتمالات آگاه به محتوا برای قطعه بندی معنایی تصویر

مجید نصیری<sup>۱</sup>، دانش‌آموخته کارشناسی ارشد؛ حمیدرضا رشیدی کنعان<sup>۲\*</sup>، استادیار؛ سید حمید امیری<sup>۳</sup>، استادیار

۱- دانشکده مهندسی کامپیوتر - دانشگاه تربیت دبیر شهید رجایی - تهران - ایران

۲\* - دانشکده مهندسی کامپیوتر - دانشگاه تربیت دبیر شهید رجایی - تهران - ایران

۳- دانشکده مهندسی کامپیوتر - دانشگاه تربیت دبیر شهید رجایی - تهران - ایران

<sup>۱</sup>majid.nasiri@sru.ac.ir, <sup>۲</sup>h.rashidyanan@sru.ac.ir, <sup>۳</sup>s.hamidamiri@sru.ac.ir

\* نشانی نویسنده مسئول: حمیدرضا رشیدی کنعان، تهران، لویزان، دانشگاه تربیت دبیر شهید رجایی، دانشکده مهندسی کامپیوتر

چکیده- قطعه‌بندی معنایی تصویر مبتنی بر شبکه‌های عصبی عمیق، از رویکردهای مهم محققان بینایی ماشین می‌باشد. در روش‌های مبتنی بر شبکه‌های عصبی عمیق، بطور کلی از یک شبکه-پایه که برای کاربرد شناسایی تصویر، آموزش دیده است، بمنظور استخراج ویژگی از تصویر استفاده می‌شود. از آنجا که ابعاد ویژگی‌های خروجی از این شبکه‌های-پایه، کوچکتر از تصویر ورودی می‌باشد، لذا با اضافه کردن چندین لایه پیچشی به انتهای این شبکه‌های-پایه، ابعاد ویژگی‌های خروجی از این شبکه‌ها را به اندازه ابعاد تصویر ورودی می‌رسانند. استفاده از ویژگی‌های محلی خروجی از شبکه‌های-پایه، بدون در نظر گرفتن ارتباط کلی بین این ویژگی‌های محلی، منجر به قطعه‌بندی ضعیف و ناهموار می‌شود. بر این اساس، در این تحقیق واحدی با نام "واحد احتمالات آگاه به محتوا" پیشنهاد می‌شود. این واحد با کمک ویژگی‌های محلی خروجی از شبکه‌های-پایه، بردار احتمال حضور کلاس‌های مختلف را در تصویر تولید می‌کند. واحد پیشنهادی را می‌توان در معماری‌های مختلف قطعه‌بندی معنایی تصویر قرار داد. در این تحقیق، با اضافه کردن واحد پیشنهادی CAP به معماری‌های پایه FCN<sup>2</sup> و DeepLab-v3-plus، به ترتیب معماری‌های FCN-CAP و DeepLab-v3-plus-CAP پیشنهاد شده است. به‌منظور آموزش معماری‌های پیشنهادی از دادگان PASCAL VOC2012 استفاده شده است. نتایج آزمایش‌ها نشان می‌دهد که معماری‌های پیشنهادی نسبت به معماری‌های پایه مربوطه، به ترتیب ۱۹/۹ درصد و ۴/۰ درصد بهبود دقت (mIoU) دارد.

واژه‌های کلیدی: قطعه‌بندی معنایی تصویر، شبکه‌های عصبی عمیق، شبکه‌های عصبی پیچشی، واحد احتمالات آگاه به محتوا.

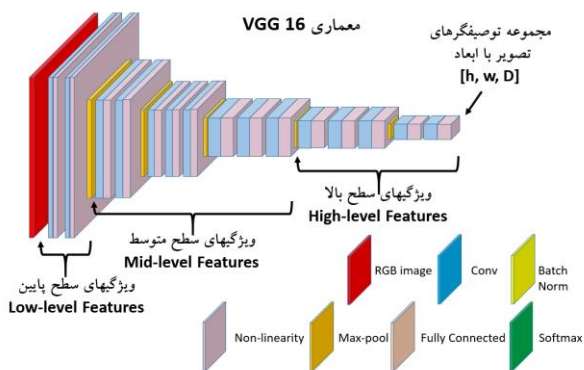
### ۱- مقدمه

در سالهای اخیر روش‌های مبتنی بر شبکه‌های عصبی عمیق، پیشرفت بسیاری نسبت به روش‌های پیشین که بر پایه ویژگی‌های مهندسی است، داشته‌اند. این روش‌ها در کاربردهای متفاوتی مانند طبقه‌بندی تصویر [۱]، قطعه‌بندی تصویر [۲]، بازنشاسی صوت [۳]، بازنشاسی رفتار [۴] و غیره پیشرفت‌های زیادی نسبت به روش‌های پیشین داشته‌اند. به فرآیندی که پیکسل‌های مشابه را در تصویر در یک دسته بندی قرار می‌دهد، قطعه‌بندی تصویر می‌گویند [۵]، در حالی که در قطعه‌بندی معنایی تصویر، بخش‌هایی از تصویر که مربوط به

با پیشرفت‌های انجام شده در زمینه تصویر برداری و تولید تصاویر دیجیتال با تفکیک‌پذیری بالا، نیاز به قطعه‌بندی دقیق تصویر بیش از پیش احساس می‌گردد، به نحوی که یکی از اساسی‌ترین مراحل پردازش تصویر، قطعه‌بندی تصویر می‌باشد. از طرف دیگر، از آنجا که قطعه‌بندی تصویر از مراحل اولیه الگوریتم‌های بینایی ماشین می‌باشد، لذا انجام دقیق قطعه‌بندی، تاثیر زیادی در نتایج مراحل بعدی خواهد داشت.

با توجه به مقاله [۱۴]، ویژگی‌های استخراج شده در لایه اول شامل خطوط، نقاط و خم‌های ساده است. به این ویژگی‌های استخراج شده در لایه اول، ویژگی‌های سطح-پایین<sup>۸</sup> می‌گویند. این ویژگی‌های سطح-پایین هر چند در کاربرد شناسایی تصویر (کاربردی که برای آن طراحی شده‌اند) بسیار مفید است، ولی برای کاربرد قطعه‌بندی معنایی تصویر مشکل‌ساز هستند. مشکل ساز بودن لایه میانگین تجمع بدین علت است که، این لایه، ویژگی‌های سطح-پایین را حذف می‌کند، در حالی که این ویژگی‌ها برای بازسازی حاشیه کلاس‌های موجود در تصویر قطعه‌بندی شده خروجی، ضروری است.

در شبکه‌های عصبی پیچشی، با قرار دادن لایه‌های مختلف پیچشی و میانگین تجمع به صورت پشت سر هم، رفته رفته ابعاد مکانی ویژگی‌های استخراج شده کوچکتر، عمق ویژگی‌های استخراج شده بیشتر می‌شود. بسته به میزان پیچیدگی ویژگی‌های استخراج شده، می‌توان این ویژگی‌ها را به سه دسته ویژگی‌های سطح-پایین، ویژگی‌های سطح-متوسط<sup>۹</sup> و ویژگی‌های سطح-بالا<sup>۱۰</sup> طبقه‌بندی کرد (شکل (۲)).



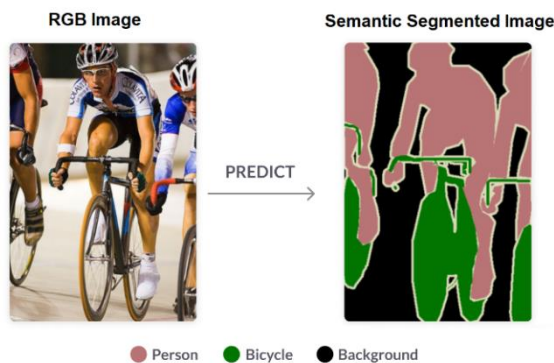
شکل ۲: معماری شبکه‌های عصبی پیچشی عمیق VGG16.

در رویکردهای مبتنی بر معماری FCN، از آنجا که ابعاد مکانی ویژگی‌های خروجی از شبکه‌های پایه کوچکتر از اندازه تصویر ورودی می‌باشد، با اضافه کردن چندین لایه پیچشی به انتهای این شبکه‌های پایه، اندازه ویژگی‌های خروجی از این شبکه‌ها را با استفاده از لایه‌های deconvolutional به اندازه تصویر ورودی می‌رسانند.

مجموعه ویژگی‌های خروجی از یک شبکه-پایه، تصویر ورودی را توصیف می‌کنند. بعبارت دیگر، خروجی این شبکه‌های پایه، مجموعه‌ای از بردارهای توصیفگر محلی است، که کل این

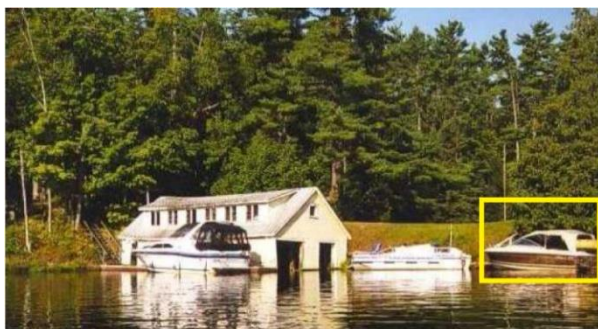
یک شیء از یک کلاس خاص می‌باشد، در یک دسته‌بندی قرار می‌گیرد. همانطور که در شکل (۱) مشاهده می‌شود، تصویر سمت چپ ورودی مدل و تصویر سمت راست خروجی قطعه‌بندی شده معنایی مدل می‌باشد. به عبارت دیگر، قطعه‌بندی اجزای تصویر به کلاس‌های از پیش تعیین شده بطوریکه هر کلاس معنای یک بخش را مشخص کند، قطعه‌بندی معنایی تصویر نامیده می‌شود. قطعه‌بندی معنایی تصویر از عملیات پایه-ای در بسیاری از کاربردهای بینایی ماشین مانند، خودروهای بدون سرنشین<sup>۳</sup> [۶]، تجزیه اجزای بدن<sup>۴</sup> [۷] و درک صحنه<sup>۵</sup> [۸] است.

بیشتر رویکردهای مرزهای دانش، مبتنی بر رویکرد معماری FCN [۲] می‌باشد. این معماری از معماری‌های مبتنی بر شبکه‌های عصبی عمیق است. در روش‌های مبتنی بر شبکه‌های عصبی عمیق، بطور کلی از یک شبکه-پایه<sup>۶</sup>، که برای کاربرد شناسایی تصویر با دادگان بسیار بزرگ مانند ImageNet [۹] آموزش دیده است، برای استخراج ویژگی از تصویر استفاده می‌شود. از جمله شبکه‌های پایه استفاده شده در این مقالات، شبکه‌های AlexNet [۱]، VGG [۱۰]، ResNet [۱۱]، GoogleNet [۱۲] و Xception [۱۳] هستند. تمامی شبکه‌های پایه مذکور، از نوع شبکه‌های عصبی پیچشی هستند که معمولاً از پشت سرهم قرار دادن لایه‌های پیچشی و میانگین تجمع<sup>۷</sup> بوجود آمده‌اند. با استفاده از لایه‌های پیچشی، ویژگی‌های تصویر استخراج می‌شوند و با استفاده از لایه‌های میانگین تجمع، مقدار بیشینه این ویژگی‌ها در پنجره لایه میانگین تجمع انتخاب شده و به لایه بعدی انتقال داده می‌شوند. در لایه میانگین تجمع، با انتخاب مقدار بیشینه و حذف بقیه مقادیر، اندازه ابعاد مکانی تصویر و روی (طول و عرض تصویر) کاهش می‌یابد. لازم به ذکر است که با حذف مقادیر زیادی از ویژگی‌ها در لایه میانگین تجمع، اطلاعات زیادی از بین می‌رود.



شکل ۱: قطعه‌بندی معنایی تصویر، تصویر سمت چپ ورودی مدل و تصویر سمت راست خروجی قطعه‌بندی شده معنایی مدل است [۳۳].

PASCAL VOC2012 [۳۳] مشاهده شد که این معاری، دقت (mIoU) بهتری نسبت به معماری پایه FCN دارد. بطور خلاصه نوآوری ما در این تحقیق، طراحی یک واحد شبکه عصبی برای استخراج بردار احتمال کلاس ها، به نام واحد "احتمالات آگاه به محتوا" (شکل (۴)) می باشد. این واحد را می توان در معماری های دیگر قطعه بندی معنایی تصویر نیز بکار برد. در ادامه مقاله، در بخش ۲، مروری بر کارهای گذشته بر پایه رویکردهای مبتنی بر یادگیری عمیق را خواهیم داشت. سپس در بخش ۳، الگوریتم های مرتبط با روش پیشنهادی را بطور مختصر شرح خواهیم داد. در بخش ۴، روش پیشنهادی را تشریح و فرمول بندی خواهیم کرد. در نهایت، به ارائه نتایج و نتیجه گیری خواهیم پرداخت.



شکل ۳: قطعه بندی اشتباه بدون در نظر گرفتن ویژگی های سطح- تصویر [۱۸].

## ۲- مروری بر کارهای پیشین

همانطور که در مقالات [۲، ۱۵، ۱۶] نشان داده شده است، استفاده از ویژگی های سطح-بالا برای دسته بندی دقیق پیکسل های تصویر، بسیار مهم است. همچنین استفاده از ویژگی های سطح-تصویر برای دسته بندی دقیق و یکنواختی قطعه بندی بسیار مفید است [۱۷، ۱۸]. روش های زیادی برای بهبود ویژگی های سطح-تصویر ارائه شده است که در ادامه آنها را مرور می کنیم. استفاده از ویژگی های سطح-تصویر، برای اولین بار در مقاله [۳۱] ارائه شد. فرض می کنیم مجموعه بردارهای توصیفگر محلی خروجی از شبکه-پایه، یک ماتریس  $[h, w, D]$  است، که در این ماتریس  $h$  ارتفاع،  $w$  عرض و  $D$  عمق ماتریس خروجی است (شکل (۲)). در مقاله [۳۱]، ابتدا از این ماتریس ویژگی ها در راستای  $h, w$  میانگین گرفته و به بردار میانگین با ابعاد  $[1, D]$  می رسد. سپس این بردار را به اندازه قبلی خود تغییر اندازه داده و به ماتریسی به ابعاد  $[h, w, D]$  می رسد. از آنجا که این ماتریس بدست آمده، میانگین بردارهای توصیفگر محلی تصویر در هر بعد (در راستای عمق) است، لذا حاوی اطلاعاتی است که

مجموعه، تصویر ورودی را توصیف می کند. در قطعه بندی معنایی تصویر، استفاده از بردارهای توصیفگر محلی خروجی از شبکه-های-پایه، بدون در نظر گرفتن ارتباط کلی بین این ویژگی ها، منجر به قطعه بندی ضعیف و ناهموار می شود. لذا، در قطعه بندی معنایی تصویر، برای بدست آوردن تصویر قطعه بندی شده با دقت بالا و یکنواختی بیشتر، بایستی ارتباط بین ویژگی های سطح-بالا را در نظر گرفت. عبارت دیگر، در قطعه بندی معنایی تصویر، به ویژگی های سطح-تصویر<sup>۱۱</sup>، یعنی ویژگی هایی که کل تصویر را توصیف می کنند، نیازمند هستیم.

با توجه به ضعف های دیده شده در قطعه بندی معنایی تصویر در قسمت ویژگی های سطح-تصویر، تحقیقات انجام شده در این مقاله در جهت بهبود این ضعف متمرکز شده است. برای واضح شدن اهمیت ویژگی های سطح-تصویر، این موضوع را با یک تصویر تشریح می کنیم. شکل (۳) را در نظر بگیرید، با استفاده از معماری FCN [۲] در این تصویر قایق روی آب به اشتباه به عنوان خودرو، قطعه بندی شده است. این اشتباه به این علت رخ می دهد که بردارهای توصیفگر محلی استخراج شده از شبکه های عمیق (شبکه-پایه) برای دو شیء قایق و خودرو بسیار به هم شبیه هستند. برای جلوگیری از این اشتباه، کفایت ویژگی های سطح-تصویر بطور موثری در قطعه بندی پیکسل ها دخالت داده شوند، عبارت دیگر، در صورتی که بدانیم در تصویر، مجموعه کلاس های درخت، سبزه، برکه آب و کلبه جنگلی وجود دارد، می توان از قطعه بندی اشتباه قایق بعنوان خودرو جلوگیری کرد.

بعد از مشاهده اشتباهات قطعه بندی بوجود آمده ناشی از کمبود اطلاعات مرتبط با محتویات کلی در قطعه بندی معنایی تصویر، در این مقاله، واحد "احتمالات آگاه به محتوا" برای تقویت ویژگی های سطح-تصویر پیشنهاد می شود. این واحد بردارهای توصیفگر محلی خروجی از شبکه های-پایه را در ورودی دریافت کرده و در خروجی بردار احتمالات آگاه به محتوا را تصویر را تولید می کند. بردار احتمالات خروجی از واحد احتمالات آگاه به محتوا، احتمال حضور هر یک از کلاس ها را در تصویر مشخص می کند. وجود این احتمالات مانند این است که بدانیم محتویات درون تصویر ورودی، چه فضایی را نمایش می دهند (بطور مثال فضای جنگلی، فضای داخل خانه، فضای بازار و ...)، که با داشتن این دانش در مورد محتویات درون تصویر، دقت تشخیص اشیاء بالاتر خواهد رفت. در این تحقیق، با اضافه کردن واحد احتمالات آگاه به محتوا به معماری پایه FCN، معماری FCN-CAP پیشنهاد می گردد. با آموزش این معماری با استفاده از داده گان

تصویر است. و در نهایت با ضرب این بردار، احتمال بدست آمده در لایه نهایی شبکه به دسته‌بندی دقیق‌تر شبکه کمک می‌کند.

### ۳- مروری بر الگوریتم‌های استخراج ویژگی‌های سطح-بالا

استفاده از ویژگی‌های سطح-تصویر که نماینده کل تصویر باشد در کاربردهای مختلف بینایی ماشین دیده شده است [۱۹،۲۰]. از جمله الگوریتم‌هایی که تصویر را با ویژگی‌های سطح بالا ارائه می‌دهند، می‌توان BOW [۲۱]، HOG [۲۲]، SIFT [۲۳]، Fisher Vector [۲۴] و VLAD [۲۵] را نام برد. این ویژگی‌های سطح-بالا دارای حجم کمتر و اطلاعات سطح بالاتری می‌باشند. در ادامه این الگوریتم‌ها را بطور مختصر شرح خواهیم داد.

#### ۳-۱- الگوریتم BOW

یکی از قدیمی‌ترین الگوریتم‌های استخراج کننده ویژگی‌های سطح-تصویر، الگوریتم BOW [۲۱] می‌باشد. این الگوریتم بردارهای توصیف‌گر محلی را در گروه‌هایی قرار می‌دهد طوری که بایستی یک codebook با تعداد  $k$  مرکز تعریف کرد (عبارات بصری<sup>۱۴</sup>)، که معمولاً این مراکز به کمک الگوریتم  $k$ -means [۲۶] بدست می‌آید. سپس، هر یک از بردارهای توصیف‌گر محلی به یکی از نزدیکترین مراکز، اختصاص داده می‌شوند. در نهایت، خروجی الگوریتم، هیستوگرام تعداد بردارهای محلی اختصاص داده شده به هر عبارت بصری (مرکز خوشه) می‌باشد.

#### ۳-۲- الگوریتم HOG

الگوریتم HOG [۲۲] بطور محلی هیستوگرام گرادیان جهت دار را در محل مورد نظر در تصویر استخراج کرده و از این اطلاعات بردارهایی را استخراج می‌کند که این بردارها توصیف‌گر تصویر هستند.

#### ۳-۳- الگوریتم SIFT

الگوریتم SIFT [۲۳] که از الگوریتم‌های بسیار قوی در این حوزه می‌باشد، برای استخراج ویژگی‌هایی با حجم کم، برای ارائه تصاویر در دادگان با حجم (تعداد) بالا ارائه شده است. این الگوریتم برای یک تصویر، بردارهای توصیف‌گری را ارائه می‌دهد که نسبت به انتقال، تغییر اندازه و چرخش اشیاء در تصویر مقاوم هستند. بردارهای ویژگی استخراج شده در این الگوریتم، محلی و به شکل ظاهری اشیاء وابسته هستند. این الگوریتم در کاربردهای شناسایی اشیاء [۲۷]، شناسایی چهره [۲۸]، شناسایی رفتار [۲۹] و دیگر موارد مورد استفاده قرار گرفته است.

مربوط به کل تصویر است. مقاله [۴۴]، با استفاده از هسته‌های مشبک پیچشی و همچنین استفاده از پس-پردازش CRF، توانسته ویژگی‌های عمومی‌تری را جهت دسته‌بندی بکار گیرد. در مقاله [۴۵]، برای بهبود دسته‌بندی با استفاده از ماژول GCN و بصورت سلسله مراتبی، ویژگی‌های سطح-پایین، سطح-متوسط و سطح-تصویر را ترکیب کرده است. استفاده از لایه‌های max-pooling و پیچشی بصورت زنجیره‌ای برای ترکیب ویژگی‌های استخراج شده از تصویر ورودی با ابعاد متفاوت، در مقاله [۴۶] منجر به بهبود تشخیص کلاس‌های تصویر شده است. مقاله [۴۷] با استفاده از معماری DenseNet [۴۸] در بخش رمزکننده و الگوبرداری از این معماری در بخش رمزگشا و همچنین استفاده از ارتباطات میانبر، توانسته اطلاعات تصویر را به لایه دسته‌بندی انتقال دهد تا بتواند دقت بالاتری در قطعه‌بندی بدست آورد. مقاله [۴۹] با اضافه کردن یک تابع هزینه بیشتر بر روی ماژول رمزکننده محتوای ارائه شده، توانسته ارتباط ویژگی‌های سطح-تصویر را بهتر استخراج کند. مقاله [۱۸] برای افزایش میزان ویژگی‌های سطح-تصویر، بر روی ویژگی‌های خروجی از شبکه-پایه هسته‌های<sup>۱۲</sup> پیچشی با ابعاد مکانی متفاوتی را اعمال کرده است. اعمال هسته‌های پیچشی با ابعاد مختلف باعث جمع کردن اطلاعات در ابعاد متفاوت می‌شود. این هسته‌ها در بزرگترین اندازه خود تقریباً تمامی ویژگی‌های محلی را در بر می‌گیرند. این بدان معنیست که با این ایده می‌توان ویژگی‌های سطح-ناحیه<sup>۱۳</sup>، و هم ویژگی‌های سطح-تصویر را استخراج کرد. با توجه به افزایش چشمگیر میزان پارامترها با استفاده از این روش، مقاله [۳۲] هسته‌های پیچشی را با هسته‌های مشبک پیچشی جایگزین کرده و میزان پارامترهای کمتری را، برای استخراج ویژگی‌های سطح-تصویر بکار گرفته است. در حقیقت مقاله [۳۲] با استفاده از اعمال هسته‌های مشبک پیچشی با اندازه‌های متفاوت، ویژگی‌های سطح-ناحیه، و هم ویژگی‌های سطح-تصویر را با تعداد پارامترهای کمتری استخراج کرده است.

متفاوت با تمامی روش‌های ارائه شده در رویکردهای مبتنی بر یادگیری عمیق، واحد پیشنهادی CAP مجموعه بردارهای توصیف‌گر محلی خروجی از شبکه-پایه را گرفته و در خروجی بردار احتمال کلاس‌ها را تولید می‌کند. واحد پیشنهادی CAP، در مرحله اول مجموع فاصله‌ی تک تک بردارهای توصیف‌گر محلی را، با تک تک مراکز خوشه‌های خود حساب کرده و یک ماتریس تولید می‌کند. سپس مقادیر این ماتریس را با استفاده از یک شبکه عصبی چند لایه به یک بردار تبدیل می‌کند. بردار بدست آمده در حقیقت یک بردار احتمال وجود کلاس‌ها در سطح



### ۳-۴- الگوریتم FV

خروجی حاصله از این الگوریتم، یک ماتریس  $K \times D$  بعدی بصورت  $V = [v_1, v_2, \dots, v_n]$  می باشد که از الحاق  $K$  بردار  $D$  بعدی بدست می آید، که  $K$  تعداد خوشه ها و  $D$  ابعاد بردار ویژگی ورودی می باشد.

الگوریتم VLAD مانند الگوریتم FV است، با این تفاوت که در VLAD از الگوریتم k-means استفاده شده است ولی در FV از تابع GMM استفاده شده است. خروجی حاصل از الگوریتم VLAD، همانند الگوریتم FV نرمال سازی می شود. نرمال سازی ماتریس خروجی VLAD در دو مرحله انجام می شود. در مرحله اول، بردارهای  $v_i$  با استفاده از روش نرمال سازی L2، نرمال می شوند (رابطه (۲)).

$$v_i = \frac{v_i}{\|v_i\|_2} \quad (2)$$

و در مرحله دوم، کل ماتریس  $V$  خروجی از الگوریتم با استفاده از روش نرمال سازی L2، نرمال می شوند (رابطه (۳)).

$$V = \frac{V}{\|V\|_2} \quad (3)$$

### ۴- روش قطعه بندی معنایی تصویر پیشنهادی

همانطور که قبلاً گفته شد، الگوریتم VLAD [۲۵] یک روش برای جمع آوری اطلاعات سطح-تصویر از بردارهای توصیفگر محلی می باشد. با اعمال این الگوریتم بر روی بردارهای ویژگی خروجی از شبکه های عصبی عمیق (شبکه-پایه)، می توان این بردارها را بگونه ای دیگر ارائه کرد. عبارتی دیگر می توان گفت، این الگوریتم بردارهای توصیفگر تصویر را خوشه بندی کرده و در خروجی، مجموع فاصله این بردارها از مراکز خوشه ها را ارائه می کند.

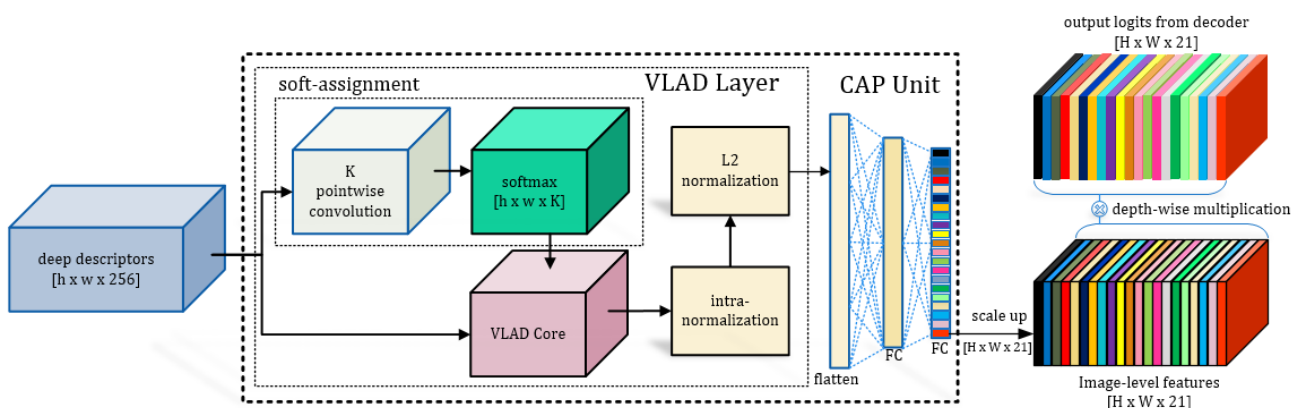
الگوریتم Fisher Vector (FV) [۳۰] یک الگوریتم برای ارائه تصویر در فضایی دیگر است. بردارهای توصیفگر حاصل از FV با جمع آوری بردارهای توصیفگر محلی از تصویر بدست می آیند. این الگوریتم در کاربردهایی که بر روی تصویر اعمال می شود، ویژگی های سطح-تصویر را استخراج می کند. این الگوریتم با استفاده از برازش مدل مخلوط گوسی (GMM) توزیع بردارهای توصیفگر تصویر را بدست می آورد. برخلاف الگوریتم BOW که اطلاعات آماری مرتبه-صفر را استخراج می کند، این الگوریتم اطلاعات آماری مرتبه-یک (میانگین) و مرتبه-دو (پراکندگی) را نیز استخراج می کند.

### ۵-۳ الگوریتم VLAD

الگوریتم VLAD [۲۵] از دیگر الگوریتم های استخراج بردارهای ویژگی سطح بالا از تصویر می باشد. در ادامه، این الگوریتم که بخشی از واحد پیشنهادی CAP را تشکیل می دهد، با دقت بیشتری تشریح خواهد شد.

در این روش، ابتدا با استفاده از الگوریتم k-means [۲۶] مجموعه مراکز خوشه ها،  $C = \{c_k, k = 1 \dots K\}$  از مجموعه بردارهای توصیفگر ورودی  $X = \{x_i, i = 1 \dots n\}$  یاد گرفته می شود. این مراکز خوشه ها در واقع همان عبارات بصری می باشند (همانند الگوریتم BOW). هر بردار ویژگی ورودی، بصورت  $x_i = \{x_i(j), j = 1 \dots D\}$  تعریف می شود که  $D$  بعد آن می باشد. برای هر بردار ویژگی ورودی  $x_i$ ، فاصله این بردار تا تمامی مراکز خوشه ها با هم جمع می شوند (رابطه (۱)).

$$v_i = \sum_{k=1}^K X - \mu_k \quad (1)$$



شکل ۴: نمای کلی واحد احتمالات آگاه به محتوا (CAP).

وزن اختصاص داده شده را با نسبت اندازه فاصله بردار ویژگی تا مرکز خوشه، کنترل می‌کند. یعنی هر چه میزان  $\alpha$  بزرگتر باشد، میزان عضویت اختصاص داده شده به بردارهایی که فاصله بیشتری تا مرکز خوشه دارند، کمتر می‌شود. در صورتی که مقدار  $\alpha$  به سمت مثبت بینهایت میل کند، عملکرد الگوریتم به سمت انتساب-سخت میل خواهد کرد. با بسط دادن رابطه و حذف ترم  $e^{-\alpha\|x_i\|^2}$  از صورت و مخرج، رابطه (۶) حاصل می‌شود [۳۴].

$$\bar{a}_k(x_i) = \frac{e^{W_k^T x_i + b_k}}{\sum_{k'} e^{W_{k'}^T x_i + b_{k'}}} \quad (۶)$$

که  $W_k = 2\alpha c_k$  و  $b_k = -\alpha\|c_k\|^2$  می‌باشد. در نهایت، رابطه (۴) به صورت رابطه (۷) خواهد شد.

$$V(j, k) = \sum_{i=1}^n \frac{e^{W_k^T x_i + b_k}}{\sum_{k'} e^{W_{k'}^T x_i + b_{k'}}} (x_i(j) - c_k(j)) \quad (۷)$$

در این رابطه، مجموعه‌های  $\{W_k\}$ ،  $\{b_k\}$  و  $\{c_k\}$  پارامترهای هر خوشه  $k$  هستند که در حین فرآیند آموزش، یاد گرفته می‌شوند. این پارامترها همگی به صورت سرتاسر در شبکه یاد گرفته می‌شوند. همچنین می‌توان در یک مرحله مراکز خوشه‌ها را با استفاده از الگوریتم  $k$ -means بدست آورده و در لایه VLAD قرار داده و حین آموزش سرتاسری شبکه، دوباره مقادیر این مراکز را دقیق‌تر کرد.

در ادامه فرآیند، خروجی  $V$  از الگوریتم VLAD که یک ماتریس  $K \times D$  است را به صورت یک بردار  $\vartheta = [1, N]$  در می‌آوریم ( $N = K \times D$ ). این بردار را به ورودی یک شبکه عصبی متشکل از دو لایه تماماً-متصل می‌دهیم، که لایه‌های آن بترتیب دارای ۱۰۲۴ و ۲۱ نورون می‌باشند. روابط این دو لایه در معادلات (۸) و (۹) آورده شده است.

$$V_{fc1} = H_1(W_1 \cdot \vartheta + b_1) \quad (۸)$$

$$V_{fc2} = H_2(W_2 \cdot V_{fc1} + b_2) \quad (۹)$$

$$V_p = \text{softmax}(V_{fc2}) \quad (۱۰)$$

در روابط (۸) و (۹)،  $W_1$  یک ماتریس  $[1024, N]$ ،  $W_2$  یک ماتریس  $[1024, 21]$ ،  $b_1$  یک بردار ۱۰۲۴ عنصری و  $b_2$  یک بردار ۲۱ عنصری می‌باشند (۲۱ تعداد کلاس‌های موجود در دادگان می‌باشد). همچنین توابع  $H_i$  توابع فعال‌ساز<sup>۱۵</sup> می‌باشند.  $V_{fc1}$  خروجی اولین لایه تماماً متصل و  $V_{fc2}$  خروجی دومین لایه می‌باشد که این خروجی‌ها به ترتیب بردارهای ۱۰۲۴ و ۲۱ عنصری هستند. با اعمال تابع بیشینه هموار<sup>۱۶</sup> بر روی بردار  $V_{fc2}$ ، این بردار به یک بردار ( $V_p$ ) احتمالاتی تبدیل می‌شود. بردار  $V_p$  بردار احتمالات سطح-تصویر می‌باشد، که بایستی بطور موثری در معماری بکار

خروجی شبکه‌های عصبی عمیق، مجموعه‌ای از بردارهای توصیفگر تصویر هستند که به عنوان ورودی به الگوریتم VLAD داده می‌شود. فرض کنیم بردارهای توصیفگر خروجی از شبکه-پایه برای تصویر  $Z$ ، مجموعه  $X = \{x_{ij}, i = 1 \dots n\}$  باشد، که برای سادگی، این مجموعه را برای یک تصویر به صورت  $X = \{x_i, i = 1 \dots n\}$  نمایش می‌دهیم. این مجموعه شامل  $n = h \times w$  بردار توصیفگر محلی با عمق  $D$  می‌باشد که مقدار پارامتر  $D$  برای معماری FCN برابر با ۵۱۲ است (شکل (۲)). الگوریتم VLAD دارای  $K$  مرکز خوشه است که برای این الگوریتم، ابتدا با استفاده از الگوریتم  $k$ -means [۲۶] مجموعه مراکز خوشه‌ها  $C = \{c_k, k = 1 \dots K\}$  را بدست می‌آوریم. برای بدست آوردن مقدار بهینه  $K$ ، به ازای مقادیر مختلف ۱۶، ۲۴، ۳۲، ۴۸، ۶۴ و ۱۲۸ دقت متفاوتی برای شبکه خواهیم داشت که این نتایج در جدول (۱) آورده شده است. نتایج این جدول نمایانگر این است که تعداد ۳۲ مرکز، دقت بالاتری را ارائه می‌دهد.

سپس برای هر بردار توصیفگر محلی  $x_i$  همانند [۲۵]، فاصله این بردار تا تمامی مراکز خوشه‌ها در ضریب  $a_k(x_i)$  ضرب شده و سپس با هم جمع می‌شوند (رابطه (۴))

$$V(j, k) = \sum_{i=1}^n a_k(x_i)(x_i(j) - c_k(j)) \quad (۴)$$

در این رابطه،  $a_k$  اندیس عمق بردار توصیفگر و  $k$  اندیس خوشه می‌باشد. بعبارت ساده‌تر،  $x_i(j)$  مقدار بردار توصیفگر  $i$ ام در بعد  $j$ ام و  $c_k(j)$  مقدار مرکز خوشه  $k$ ام در بعد  $j$ ام است. مقدار  $a_k(x_i)$  نمایانگر میزان عضویت بردار توصیفگر  $x_i$  در خوشه  $k$ ام می‌باشد. در صورت استفاده از الگوریتم پایه VLAD [۲۵]، مقدار عضویت صفر یا یک می‌باشد که در این صورت می‌گوییم الگوریتم به صورت انتساب-سخت است. ولی با روش ارائه شده در [۳۴]، مقدار عضویت می‌تواند عددی بین صفر و یک باشد. که در این صورت می‌گوییم الگوریتم به صورت انتساب-نرم است. در نهایت ماتریس خروجی  $V(j, k)$  را بصورت ستونی نرمال‌سازی کرده تا مقادیر در بعدهای مختلف جداگانه نرمال شوند (همانند [۳۴])، که این عمل همان نرمال‌سازی برون-خوشه‌ای است. بعد از نرمال‌سازی برون-خوشه‌ای، کل آرایه‌های ماتریس را دوباره نرمال‌سازی می‌کنیم. همانند مقاله [۳۴] برای انتساب-نرم، از رابطه (۵) برای بدست آوردن مقادیر  $a_k(x_i)$  استفاده شده است.

$$\bar{a}_k(x_i) = \frac{e^{-\alpha\|x_i - c_k\|^2}}{\sum_{k'} e^{-\alpha\|x_i - c_{k'}\|^2}} \quad (۵)$$

رابطه (۵) با توجه به نزدیکی بردار توصیفگر  $x_i$  به خوشه  $c_k$ ، وزنی را به آن اختصاص می‌دهد.  $\alpha$  یک مقدار ثابت مثبت است که میزان

### ۵-۱- معیارهای ارزیابی

روش‌های مختلفی برای ارزیابی خروجی الگوریتم‌های قطعه‌بندی تصویر وجود دارد، که دو نمونه از آنها را، که در تمامی مقالات مورد استفاده قرار می‌گیرد، در روابط (۱۱) و (۱۲) توضیح خواهیم داد. در این روابط  $k \in \mathbb{N}$  تعداد کلاس‌ها را مشخص می‌کند. همچنین در این روابط  $C_{ij} \in \mathbb{N}_0$  با  $i, j \in 1, \dots, k$  که  $C_{ij}$  تعداد پیکسل‌های متعلق به کلاس  $i$  هستند که به کلاس  $j$  نسبت داده شده‌اند. به  $C_{ij}$  ماتریس پراکندگی می‌گویند و  $G_i$  با رابطه  $G_i = \sum_{j=1}^k C_{ij}$  نشان دهنده تمامی پیکسل‌های کلاس  $i$  می‌باشد و  $P_j = \sum_i C_{ij}$  تمامی پیکسل‌هایی است که به کلاس  $j$  نسبت داده شده‌اند. دقت پیکسلی ( $PA^{17}$ ): این معیار نسبت تمامی پیکسل‌های درست دسته دسته‌بندی شده به کل پیکسل‌ها می‌باشد. دقت در این روش از رابطه‌ی (۱۱) محاسبه می‌شود.

$$PA = \frac{\sum_{i=1}^k C_{ij}}{\sum_{i=1}^k G_i} \quad (11)$$

دقت میانگین اشتراک به اجتماع پیکسل‌ها ( $mIoU^{18}$ ): در این معیار ابتدا نسبت اشتراک پیکسل‌های درست دسته‌بندی شده، به اجتماع کل پیکسل‌های کلاس مربوطه و پیکسل‌هایی که به آن کلاس نسبت داده شده‌اند محاسبه شده و سپس میانگین این مقادیر برای همه کلاس‌ها محاسبه می‌شود. دقت در این روش از رابطه‌ی (۱۲) محاسبه می‌شود.

$$mIoU = \frac{1}{k} \sum_{i=1}^k \frac{C_{ij}}{G_i + P_i - C_{ii}} \quad (12)$$

### ۵-۲- دادگان مورد استفاده

در آزمایش‌های انجام شده برای آموزش شبکه‌های FCN-CAP و DeepLab-v3-plus-CAP، از مجموعه دادگان PASCAL VOC2012 [۳۳] استفاده شده است. این مجموعه دادگان دارای ۱۴۶۴ تصویر استاندارد و ۱۰۵۸۲ تصویر از مجموعه تصاویر تهیه شده توسط مقاله [۳۵] برای بخش آموزش و تعداد ۱۴۴۹ تصویر برای ارزیابی است. همچنین ابعاد تصاویر متفاوت می‌باشد. بسیاری از مقالات برای ارزیابی عملکرد الگوریتم ارائه شده خود و مقایسه آن با دیگر الگوریتم‌ها، از این مجموعه دادگان استفاده می‌کنند. لذا در این تحقیق نیز این مجموعه دادگان انتخاب شده است.

### ۵-۳- شرایط پیاده سازی

برای آموزش معماری‌های پیشنهادی FCN-CAP و DeepLab-v3-plus-CAP، از واحد پردازنده گرافیکی GTX1070 TI با حافظه ۸ گیگابایت استفاده شده است. همچنین، تمامی کدهای مربوط به پیاده‌سازی این معماری‌ها در چهارچوب تنسورفلو [۳۶] و به زبان

گرفته شود. بردار  $V_p$  حاوی احتمال وجود هر یک از ۲۱ کلاس در تصویر است، که هر یک از درایه‌های آن مربوط به یک کلاس جداگانه است. بردار خروجی  $V_p$  را بصورت یک بردار  $[1, 1, 21]$  تغییر شکل داده و سپس آن را تا ابعاد  $F=[h, w, 21]$  بزرگ می‌کنیم. که  $h$  و  $w$  ارتفاع و عرض مجموعه بردارهای توصیفگر خروجی از شبکه-پایه برای تصویر ورودی به ابعاد  $[H, W, 3]$  است. از طرفی دیگر در لایه خروجی معماری FCN [۲]، یک ماتریس احتمالات کلاس‌ها که هم بعد با ماتریس  $F$  است، را داریم. در مرحله نهایی به منظور وزن‌دهی ماتریس خروجی معماری FCN و تقویت یا تضعیف کردن مقادیر هر کلاس، مقادیر ماتریس  $F$  را در این ماتریس ضرب می‌کنیم (شکل (۵)). این عمل با عملیات ضرب نقطه‌ای بین این دو ماتریس انجام می‌شود.

از آنجایی که بردار خروجی نهایی  $V_p$  حاوی احتمالات محتوایی تصویر می‌باشد، ما مجموعه لایه VLAD [۲۵] به همراه لایه‌های اضافه شده را به عنوان واحد احتمالات آگاه به محتوا معرفی می‌کنیم. این واحد در ورودی خود ویژگی‌های سطح بالای تصویر را دریافت کرده و با استفاده از روش خوشه‌بندی، این ویژگی‌های سطح بالا را خوشه‌بندی می‌کند. در این واحد، اطلاعات ویژگی‌های خوشه‌بندی بندی شده با استفاده از دو لایه شبکه عصبی تماماً متصل به ویژگی‌های سطح-تصویر و سپس احتمالات کلاس‌ها تبدیل می‌شود. آزمایش‌های انجام شده نشان می‌دهد که این واحد قدرت زیادی در استخراج احتمالات سطح-تصویر را دارد. معماری واحد پیشنهاد شده در شکل (۴) آورده شده است.

جدول ۱: نتایج بدست آمده معماری FCN-CAP با مقادیر متفاوتی از مراکز خوشه‌ها

معماری	K	Pixel Accuracy (PA)	mIoU
FCN-CAP-8s	۱۶	۹۰/۱	۶۲/۰
FCN-CAP-8s	۲۴	۹۱/۶	۶۳/۷
FCN-CAP-8s	۳۲	۹۱/۸	۶۴/۱
FCN-CAP-8s	۴۸	۹۱/۱	۶۳/۲
FCN-CAP-8s	۶۴	۹۰/۶	۶۲/۷
FCN-CAP-8s	۱۲۸	۹۰/۶	۶۲/۵

### ۵-۵- نتایج آزمایش‌ها و شبیه‌سازیها

در این بخش ابتدا به بیان معیارهای استفاده شده برای ارزیابی قطعه‌بندی معنایی تصویر می‌پردازیم، سپس شرایط پیاده سازی را تشریح خواهیم کرد. در نهایت، نتایج حاصل از قطعه‌بندی معنایی تصویر با معماری‌های پیشنهادی FCN-CAP و DeepLab-v3-plus-CAP که به ترتیب با اضافه شدن واحد CAP به معماری‌های FCN و DeepLab-v3-plus [۴۵] بدست آمده‌اند را ارائه می‌کنیم.



نظر استفاده شده است. عبارت دیگر، برای دسته‌بندی هر پیکسل هم از خود پیکسل و هم از ویژگی‌های استخراج شده در سطح بالا، سطح متوسط و سطح پایین (شکل (۲)) استفاده شده است. روش استفاده شده در [۴۲] شباهت زیادی به روش [۴۰] دارد، با این تفاوت که، ابتدا ویژگی‌های تصویر با استفاده از شبکه عصبی پیچشی استخراج شده، سپس برای دسته‌بندی پنجره‌های کاندید شده از ویژگی‌هایی که با پنجره کاندید ماسک می‌شوند، استفاده شده است. این تغییر باعث افزایش سرعت مدل تا ۵۰ برابر شده است. لازم به ذکر است ماژول پیشنهادی CAP تنها دارای ۱۶ هزار پارامتر است که در مقابل ۱۳۴ میلیون پارامتر معماری پایه FCN، درصد بسیار ناچیزی می‌باشد.

در گام نهایی تست‌ها، برای اثبات توانمندی واحد پیشنهادی CAP، این واحد را به معماری Deeplab-v3-plus که از قدرتمندترین معماری‌های اخیر قطعه بندی معنایی تصویر می باشد، اضافه کردیم، از آنجا که این معماری در بخش کدکننده، از شبکه-پایه ۱۰۱ لایه resnet استفاده کرده است، آموزش این شبکه به واحد پردازش گرافیکی با حجم حافظه بیشتری نیاز دارد، با توجه به محدودیتهای سخت افزاری که در اختیار داشتیم، متأسفانه امکان آموزش این معماری و همچنین معماری Deeplab-v3-plus-CAP میسر نبود، لذا این دو معماری را برای مقایسه منصفانه با منابع و شرایط موجود خودمان یعنی با تصاویری با اندازه 385×385 و اندازه دسته (batch size) برابر ۶ آموزش دادیم. در این شرایط همانطور که انتظار داشتیم، نتیجه بدست آمده برای معماری deeplab-v3-plus از مقداری که در مقاله مربوطه [۴۵] گزارش شده، ضعیف‌تر است. نتایج حاصل از آموزش ۱۰ مرتبه و میانگین‌گیری برای هر یک از معماری‌ها، نشان‌دهنده بهبود ۰/۴ درصدی (mIoU) در معماری پایه Deeplab-v3-plus می‌باشد. در جدول (۳)، نتایج قطعه‌بندی برای این دو معماری آورده شده است.

پایتون نوشته شده است. برای بهینه سازی شبکه طراحی شده از تابع هزینه آنتروپی-متقابل<sup>۱۹</sup> استفاده شده است، رابطه (۱۳).

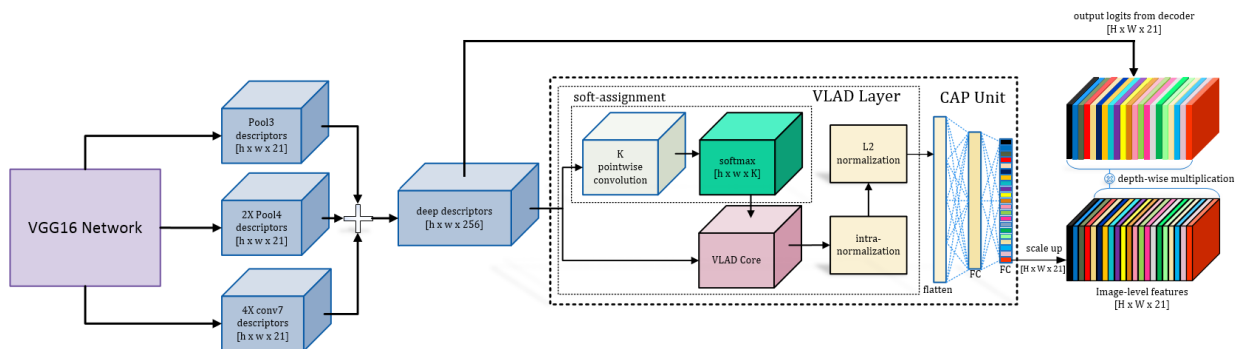
$$loss = - \sum_{i=1}^P \sum_{c=1}^C Y_{ic} \cdot \log(\bar{Y}_{ic}) \quad (13)$$

که در این رابطه، P تعداد پیکسل‌ها، C تعداد کلاس‌ها (در این مجموعه دادگان (۲۱) و  $Y_{ic}$  مقدار مورد انتظار در پیکسل‌ام برای کلاس cام است و  $\bar{Y}_{ic}$  مقدار پیش‌بینی شده توسط شبکه در پیکسل‌ام برای کلاس cام است. به منظور کمینه کردن هزینه، از الگوریتم پس انتشار خطا [۳۷] و بهینه‌ساز Adam Optimizer [۳۸] با نرخ آموزش بسیار کوچک (جهت آموزش بهتر) و برابر با  $10^{-4}$  استفاده شده است.

#### ۵-۴- نتایج آزمایش‌ها

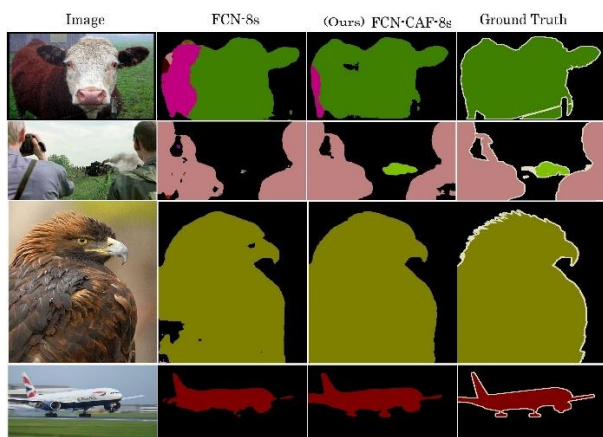
معماری FCN-CAP با قرار دادن واحد CAP در معماری FCN [۲] بدست آمده است. این واحد به تشخیص صحیح پیکسل‌ها و یکنواختی قطعه‌بندی کمک می‌کند. شکل (۵) نمای کلی شبکه FCN-CAP را نشان می‌دهد. پس از آموزش این شبکه، عملکرد آن بر روی تصاویر بخش آزمون از مجموعه دادگان VOC2012 [۳۳] با معیارهای PA و mIoU مورد ارزیابی قرار گرفته و نتایج در جدول (۲) آورده شده است. همچنین در این جدول معماری FCN-CAP با معماری‌های [۴۰، ۴۱، ۴۲] مقایسه شده است.

در الگوریتم [۴۰] ابتدا پنجره‌هایی در تصویر به عنوان نواحی کاندید برای حضور شیء انتخاب می‌شود، سپس با استفاده از شبکه عصبی پیچشی، بطور جداگانه ویژگی‌های کل نواحی کاندید و ویژگی شیء موجود در پنجره استخراج می‌شود. در ادامه، با استفاده از الگوریتم SVM [43] کلاس پنجره مورد نظر بدست آمده و در نهایت با استفاده از ویژگی‌های خاص شیء منتسب شده به پنجره، نواحی شیء موجود در پنجره، قطعه بندی می‌شود. در [۴۱] برای دسته‌بندی هر پیکسل از تمامی ویژگی‌های استخراج شده در لایه های شبکه عصبی پیچشی در راستای پیکسل مورد

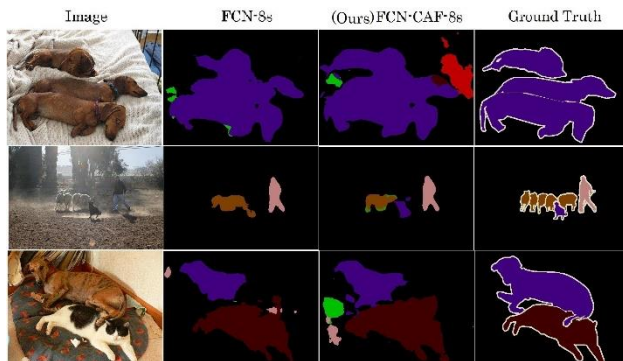


شکل ۵: نمای کلی معماری FCN-CAP که با جاسازی واحد CAP در معماری FCN بدست می آید.

معماری‌های FCN و Deeplab-v3-plus را به ترتیب به اندازه ۱/۹ درصد و ۰/۴ درصد (mIoU) بهبود دهد، استفاده از این واحد در دیگر معماری‌های قطعه‌بندی معنایی تصویر نیز باعث افزایش دقت این معماری‌ها خواهد شد. این نتیجه نشان دهنده اهمیت هر چه بیشتر ویژگی‌های سطح-تصویر در قطعه‌بندی صحیح تصاویر می‌باشد.



شکل ۶: نمونه‌هایی از تصاویر قطعه بندی شده توسط معماری FCN- CAP و معماری پایه FCN.



شکل ۷: نمونه‌هایی از تصاویر قطعه بندی شده توسط معماری FCN- CAP که عملکرد بهتری نسبت به FCN نداشته است.

#### ۷- پیوست‌ها

در مرحله طراحی بخش VLAD از واحد CAP، برای اینکه مراکز خوشه‌ها را با استفاده از الگوریتم k-means بدست آوریم، ابتدا شبکه-پایه را انتخاب می‌کنیم. سپس، تمامی تصاویر بخش آموزش دادگان را یکی پس از دیگری به ورودی شبکه-پایه داده، و بردارهای توصیفگر خروجی را ذخیره می‌کنیم. در نهایت، مجموعه تمامی بردارهای توصیفگر ذخیره شده را با الگوریتم k-means خوشه‌بندی کرده و مراکز خوشه‌ها را بدست می‌آوریم. این مراکز خوشه‌های بدست آمده را بعنوان مقادیر اولیه مراکز خوشه‌ها

جدول ۲: ارزیابی معماری FCN-CAP با معماری پایه FCN

معماری	Pixel Accuracy (PA)	mIoU
SDS [۴۰]	۴۹/۷	۵۱/۶
Hyper column [۴۱]	۶۰	۵۴/۶
CFM [۴۲]	۶۰/۷	۶۱/۸
FCN-8s [۲]	۹۰/۳	۶۲/۲
FCN-CAP-8s	۹۱/۸	۶۴/۱

جدول ۳: مقایسه معماری DeepLab-v3-plus-CAP با معماری پایه DeepLab-v3-plus آموزش دیده شده با تصاویر با ابعاد ۳۸۵×۳۸۵ و اندازه دسته برابر با ۶.

معماری	Pixel Accuracy (PA)	mIoU
Deeplab-v3-plus [۴۵]	۹۲/۰۵	۶۸/۳۳
Deeplab-v3-plus-CAP	۹۲/۲۵	۶۸/۷۳

شکل (۶)، نمونه‌هایی از تصاویر قطعه‌بندی شده توسط روشهای مختلف را نمایش می‌دهد. همانطور که مشاهده می‌شود، در مدل پیشنهادی، به دلیل اضافه شدن ویژگی‌های سطح-تصویر، خروجی‌ها بهبود یافته و قطعه‌بندی به صورت هموارتری در تصاویر انجام شده است. همچنین، در شکل (۷) نمونه‌هایی که در آنها معماری FCN-CAP عملکرد مطلوبی را ارائه نداده، را آورده ایم. همانطور که مشاهده می‌شود در این نمونه‌ها، خروجی بدست آمده شباهت کمتری به خروجی مطلوب دارد. عدم بهبود خروجی در مدل پیشنهادی، بیشتر در تصاویری دیده می‌شود که فضای تصویر بسته بوده و تصویر کاملاً واضح نیست. به عنوان مثال، در سطرهای اول و سوم شکل (۷)، به علت مبهم بودن فضای تصویر، بهبودی حاصل نشده است، درحالیکه در سطر دوم، علیرغم اینکه دقت کلیت تصویر کمتر شده است، ولی به دلیل مشخص بودن فضای تصویر، مشاهده می‌شود که کلاس "سگ" بر خلاف معماری پایه، درست تشخیص داده شده است. لازم به ذکر است که در این نمونه‌ها، کارایی معماری FCN-CAP بر اساس معیارهای mIoU و Pixel Accuracy (PA) از عملکرد معماری پایه FCN پایین‌تر است.

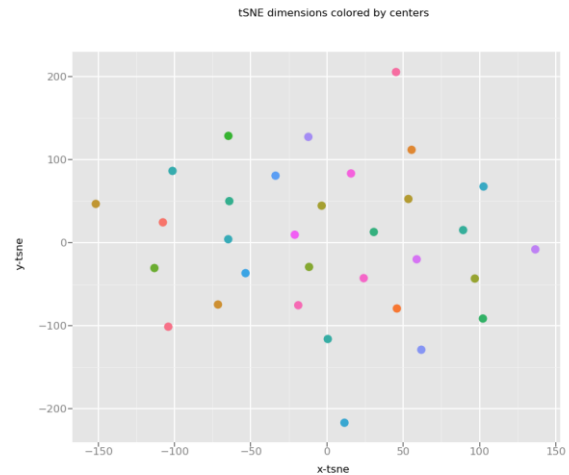
#### ۶- نتیجه‌گیری

در این مقاله، واحدی با نام احتمالات آگاه به محتوا (CAP) برای تقویت ویژگی‌های در سطح تصویر پیشنهاد شد. آزمایش‌های انجام شده و تصاویر قطعه‌بندی شده با قرار دادن این واحد در معماری‌های FCN و Deeplab-v3-plus نشان دهنده عملکرد مفید این واحد می‌باشد. همانطور که این واحد توانسته عملکرد

conference on computer vision and pattern recognition (pp. 3213-3223).

- [9] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (pp. 248-255). Ieee.
- [10] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [11] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- [12] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
- [13] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. *arXiv preprint, 1610-02357*.
- [14] Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., & Lipson, H. (2015). Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*.
- [15] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587).
- [16] Mottaghi, R., Chen, X., Liu, X., Cho, N. G., Lee, S. W., Fidler, S., ... & Yuille, A. (2014). The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 891-898).
- [17] Peng, C., Zhang, X., Yu, G., Luo, G., & Sun, J. (2017, July). Large kernel matters—improve semantic segmentation by global convolutional network. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on* (pp. 1743-1751). IEEE.
- [18] Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017, July). Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (pp. 2881-2890).
- [19] Yang, J., Yu, K., Gong, Y., & Huang, T. (2009, June). Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (pp. 1794-1801). IEEE.
- [20] Bicego, M., Lagorio, A., Grosso, E., & Tistarelli, M. (2006, June). On the use of SIFT features for face authentication. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on* (pp. 35-35). IEEE.
- [21] Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. (2004, May). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV (Vol. 1, No. 1-22, pp. 1-2)*.
- [22] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (Vol. 1, pp. 886-893). IEEE.
- [23] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91-110.
- [24] Sánchez, J., Perronnin, F., Mensink, T., & Verbeek, J. (2013). Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3), 222-245.
- [25] Jegou, H., Perronnin, F., Douze, M., Sánchez, J., Perez, P., & Schmid, C. (2012). Aggregating local image descriptors into compact codes. *IEEE transactions on pattern analysis and machine intelligence*, 34(9), 1704-1716.
- [26] Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1), 100-108.
- [27] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on* (Vol. 2, pp. 1150-1157). Ieee.
- [28] Bicego, M., Lagorio, A., Grosso, E., & Tistarelli, M. (2006, June). On the use of SIFT features for face authentication. In *Computer*

استفاده کرده و حین آموزش سراسری شبکه، دوباره مقادیر این مراکز را دقیق‌تر می‌کنیم. شکل (۸) موقعیت مراکز خوشه‌های بدست آمده را نمایش می‌دهد. این تصویر با اعمال الگوریتم کاهش ابعاد t-SNE [۳۹] بر روی مراکز و کاهش ابعاد آنها از ۲۱ به ۲ بعد ترسیم شده است.



شکل ۸: ترسیم مراکز خوشه‌ها با الگوریتم t-SNE برای ۳۲ خوشه.

## ۸- سپاسگزاری

این پژوهش با حمایت مالی دانشگاه تربیت دبیر شهید رجایی طبق قرارداد شماره ۱۹۹۷۴ مورخ ۹۷/۸/۳۰ انجام گردیده است.

## مراجع

- [1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
- [2] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).
- [3] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6), 82-97.
- [4] Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems* (pp. 568-576).
- [5] Thoma, M. (2016). A survey of semantic segmentation. *arXiv preprint arXiv:1602.06541*.
- [6] Chen, C., Seff, A., Kornhauser, A., & Xiao, J. (2015). Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2722-2730).
- [7] Liang, X., Xu, C., Shen, X., Yang, J., Liu, S., Tang, J., ... & Yan, S. (2015). Human parsing with contextualized convolutional neural network. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1386-1394).
- [8] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., ... & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE*

of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).  
 [50] Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., & Agrawal, A. (2018). Context encoding for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7151-7160).

زیر نویس ها:

- <sup>1</sup>Context-Aware Probability
- <sup>2</sup>Fully Convolutional Network
- <sup>3</sup>Self-Drive
- <sup>4</sup>Human Parsing
- <sup>5</sup>Scene Understanding
- <sup>6</sup>Backend
- <sup>7</sup> Max-pooling
- <sup>8</sup>Low-level Features
- <sup>9</sup>Mid-Level Features
- <sup>10</sup>High-Level Features
- <sup>11</sup>Image-Level
- <sup>12</sup>Kernel
- <sup>13</sup>Region-Level Features
- <sup>14</sup>Visual Words
- <sup>15</sup>Activation Function
- <sup>16</sup>Softmax
- <sup>17</sup>Pixel Accuracy
- <sup>18</sup>Mean Intersection Over Union (mIoU)
- <sup>19</sup>Cross-Entropy

Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on (pp. 35-35). IEEE.

[29] Scovanner, P., Ali, S., & Shah, M. (2007, September). A 3-dimensional sift descriptor and its application to action recognition. In Proceedings of the 15th ACM international conference on Multimedia (pp. 357-360). ACM.

[30] Sánchez, J., Perronnin, F., Mensink, T., & Verbeek, J. (2013). Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3), 222-245.

[31] Liu, W., Rabinovich, A., & Berg, A. C. (2015). Parsenet: Looking wider to see better. arXiv preprint arXiv:1506.04579.

[32] Chen, L. C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587.

[33] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2), 303-338.

[34] Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., & Sivic, J. (2016). NetVLAD: CNN architecture for weakly supervised place recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5297-5307).

[35] Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., & Malik, J. (2011). Semantic contours from inverse detectors.

[36] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016, November). Tensorflow: a system for large-scale machine learning. In OSDI (Vol. 16, pp. 265-283).

[37] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning internal representations by error propagation (No. ICS-8506). California Univ San Diego La Jolla Inst for Cognitive Science.

[38] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[39] Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov), 2579-2605.

[40] Hariharan, B., Arbeláez, P., Girshick, R., & Malik, J. (2014, September). Simultaneous detection and segmentation. In European Conference on Computer Vision (pp. 297-312). Springer, Cham.

[41] Hariharan, B., Arbeláez, P., Girshick, R., & Malik, J. (2015). Hypercolumns for object segmentation and fine-grained localization. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 447-456).

[42] Dai, J., He, K., & Sun, J. (2015). Convolutional feature masking for joint object and stuff segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3992-4000).

[43] Suykens, Johan AK, and Joos Vandewalle. "Least squares support vector machine classifiers." *Neural processing letters* 9.3 (1999): 293-300.

[44] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4), 834-848.

[45] Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European conference on computer vision (ECCV) (pp. 801-818).

[46] Peng, C., Zhang, X., Yu, G., Luo, G., & Sun, J. (2017). Large Kernel Matters--Improve Semantic Segmentation by Global Convolutional Network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4353-4361).

[47] Lin, G., Milan, A., Shen, C., & Reid, I. (2017). Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1925-1934).

[48] Jégou, S., Drozdal, M., Vazquez, D., Romero, A., & Bengio, Y. (2017). The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 11-19).

[49] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings