++

تمرین 3 یادگیری عمیق

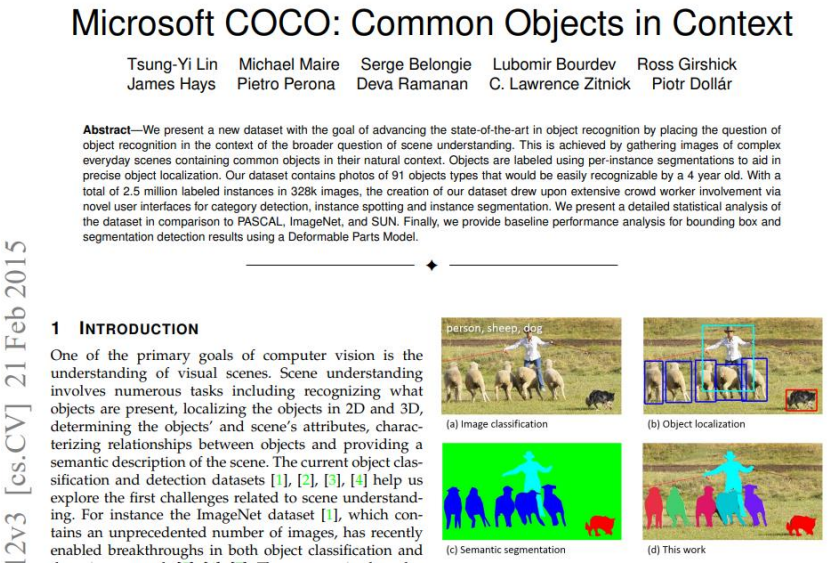
توضیحات سوال 2

امیرعباس افضلی 400100662

توضیحات دیتاست:

COCO dataset

[1]



Microsoft COCO: Common Objects in Context

Tsung-Yi Lin Michael Maire Serge Belongie Lubomir Bourdev Ross Girshick
James Hays Pietro Perona Deva Ramanan C. Lawrence Zitnick Piotr Dollár

Abstract—We present a new dataset with the goal of advancing the state-of-the-art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. This is achieved by gathering images of complex everyday scenes containing common objects in their natural context. Objects are labeled using per-instance segmentations to aid in precise object localization. Our dataset contains photos of 91 objects types that would be easily recognizable by a 4 year old. With a total of 2.5 million labeled instances in 328k images, the creation of our dataset drew upon extensive crowd worker involvement via novel user interfaces for category detection, instance spotting and instance segmentation. We present a detailed statistical analysis of the dataset in comparison to PASCAL, ImageNet, and SUN. Finally, we provide baseline performance analysis for bounding box and segmentation detection results using a Deformable Parts Model.

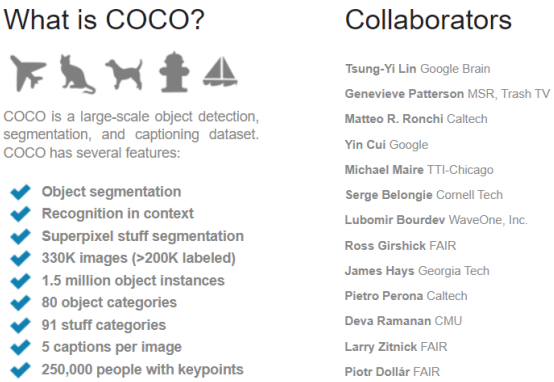
1 INTRODUCTION

One of the primary goals of computer vision is the understanding of visual scenes. Scene understanding involves numerous tasks including recognizing what objects are present, localizing the objects in 2D and 3D, determining the objects' and scene's attributes, characterizing relationships between objects and providing a semantic description of the scene. The current object classification and detection datasets [1], [2], [3], [4] help us explore the first challenges related to scene understanding. For instance the ImageNet dataset [1], which contains an unprecedented number of images, has recently enabled breakthroughs in both object classification and

person, sheep, dog

(a) Image classification (b) Object localization (c) Semantic segmentation (d) This work

این دیتاست که توسط Microsoft معرفی شده ، دارای بیش از 330k تصویر می باشد که برای تسک هایی مانند instance segmentation , semantic segmentation و image captioning و همینطور image classification کاربرد دارد.



What is COCO?

COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- Object segmentation
- Recognition in context
- Superpixel stuff segmentation
- 330K images (>200K labeled)
- 1.5 million object instances
- 80 object categories
- 91 stuff categories
- 5 captions per image
- 250,000 people with keypoints

Collaborators

Tsung-Yi Lin Google Brain
Genevieve Patterson MSR, Trash TV
Matteo R. Ronchi Caltech
Yin Cui Google
Michael Maire TTI-Chicago
Serge Belongie Cornell Tech
Lubomir Bourdev WaveOne, Inc.
Ross Girshick FAIR
James Hays Georgia Tech
Pietro Perona Caltech
Deva Ramanan CMU
Larry Zitnick FAIR
Piotr Dollár FAIR

در این سوال هدف ایمپلمنت کردن deformable convolution و همچنین طراحی یک تسک برای نشان دادن تفاوت این کانولوشن با کانولوشن عادی می باشد.

هر تصویر (sample) از این دیتاست دارای یک دیکشنری با سایز تعداد آبجکت های درون این تصویر می باشد. و هر آبجکت دارای یک category_ID می باشد که درکل 90 کتگوری وجود دارد (در توصیف دیتاست گفته شده است که 80 object

categories وجود دارد ولی 90 تا بود:/) . همچنین بدلیل حجم بالای دیتاست ، درابتدا آن را کوچک کردیم و از تمام آن استفاده نشده است (از 10240 عکس برای train و 2000 عکس برای validation استفاده شده است) .

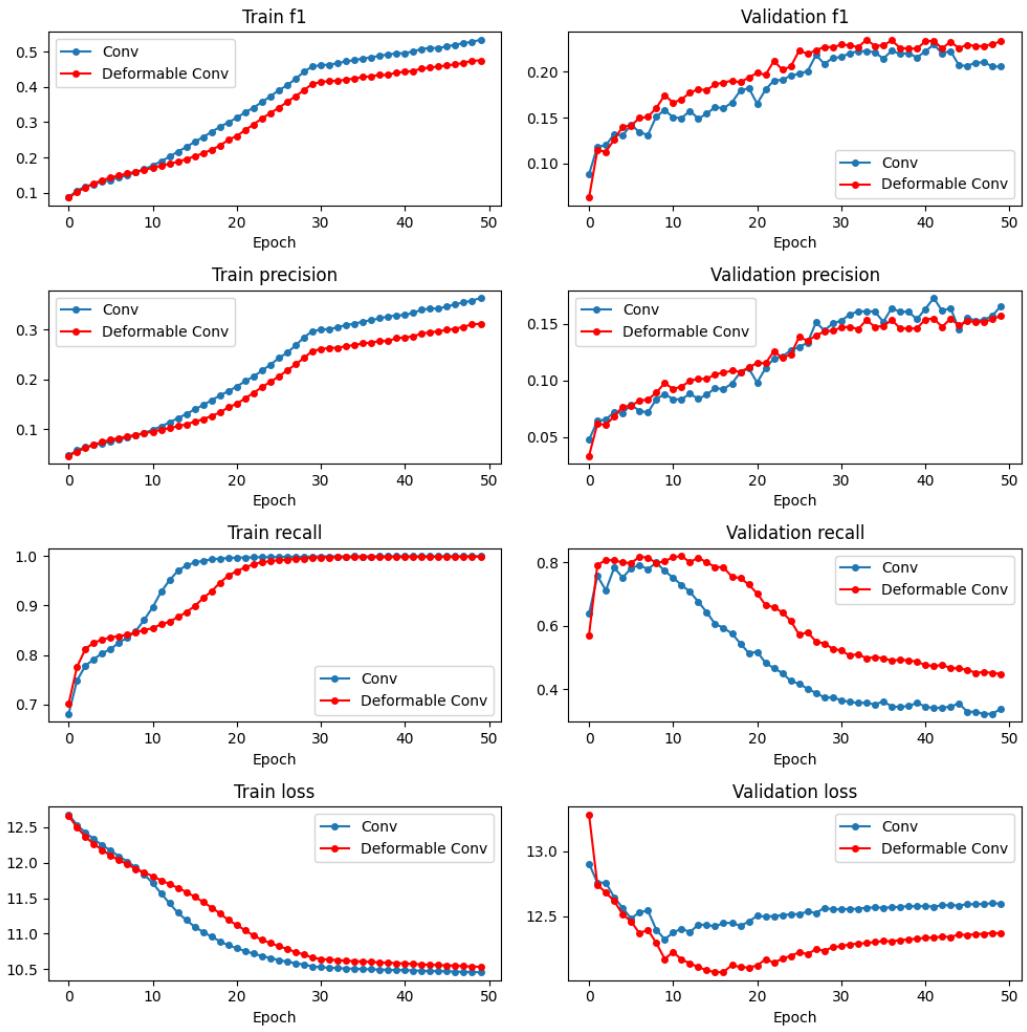
و تسک طراحی شده در این سوال ، به نوعی آبجکت دیتکشن می باشد. یعنی در لایه خروجی شبکه 90 نورون با activation سیگموید وجود دارد که نورون i ام بیان می کند که آبجکت i ام در تصویر وجود دارد و یا نه. برای همین از BinaryCrossEntropy به عنوان loss function استفاده شده است.

برای انجام این تسک ابتدا از یک شبکه کانولوشنی با سه لایه کانولوشن 3*3 و maxpooling استفاده شد . ولی به دلیل ظرفیت پایین شبکه ، نتیجه و convergence مناسبی بر روی این دیتاست نداشت (کد آن در انتهای نوت بوک آمده است) .

شبکه ی استفاده شده در سوال دارای 11 conv 3*3 و همچنین maxpooling و سه لایه dense در خروجی است که تقریبا مشابه معماری انکودر UNET می باشد.

و در لایه های کانلوشنی در شبکه ی بالا، یکبار از nn.conv2d و یکبار از deformable conv استفاده شده است.

نتایج زیر ، تفاوت خروجی شبکه با لایه های کانولوشنی متفاوت را در 50 ایپاک نشان می دهد.

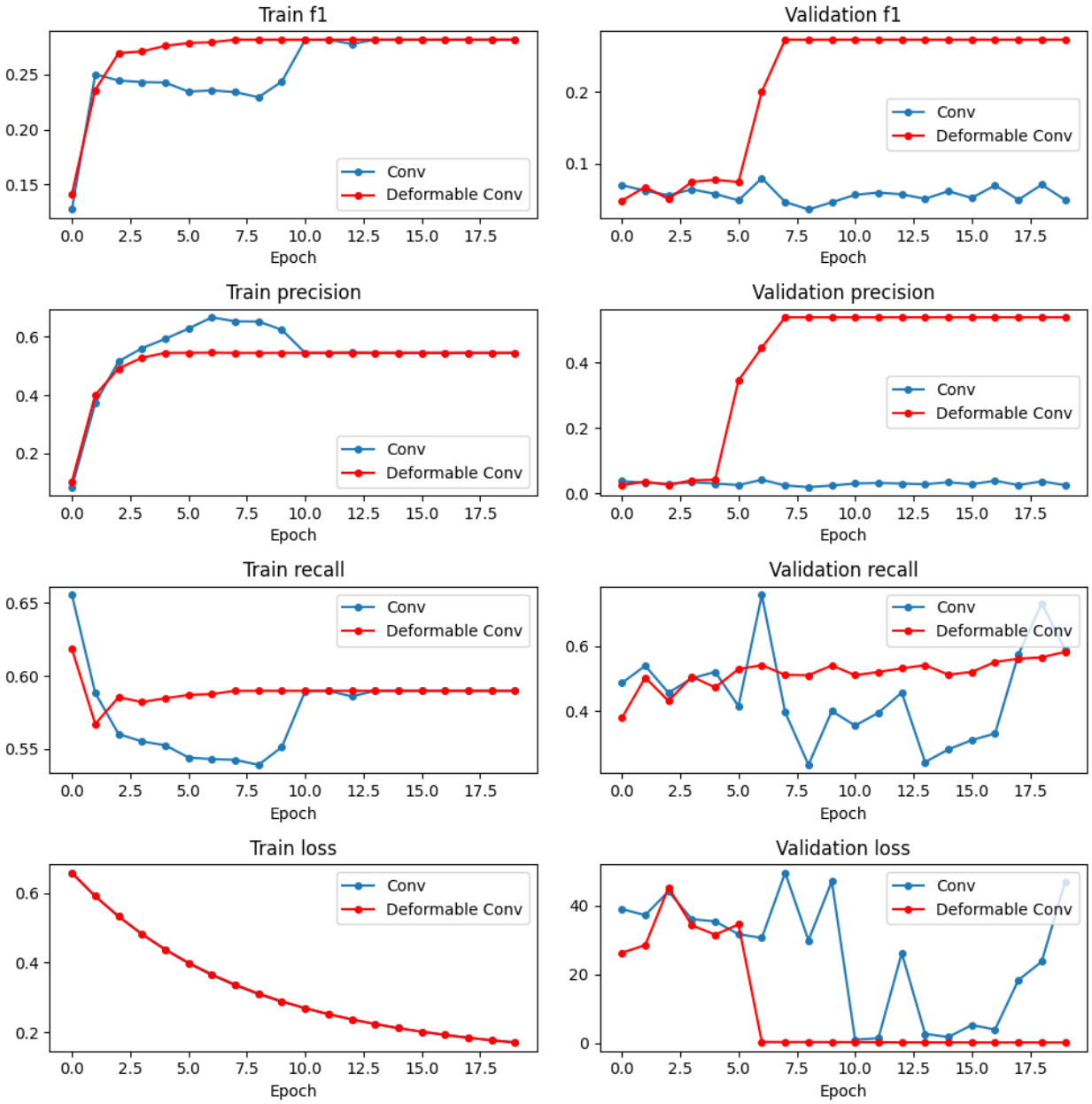


مشاهده می شود که validation precision در دو حالت تقریبا یکسان است. ولی f1 و recall در مود validation برای deformConv بهتر است که یعنی شبکه با این کانولوشن ، توانایی generalization بیشتری دارد. همچنین تفاوت validation loss نیز در دو حالت ، کاملا واضح است.

همچنین یک بار ، آزمایش بر روی دیتای ترین کوچکتر انجام شده است. و با این تفاوت که این بار صرفا دولایه اول کانولوشن را به deformeconv نغییر می دهیم. (در همان معماری قبلی)

(البته نحوه ی محاسبه loss ئ نرمالیزیشن آن در ایت حالت ، کمی با حالت قبلی تفاوت دارد ، که به همین دلیل و البته تغیر مقدار دیتاست تغییرات زیادی در مقادیر دو آزمایش وجود دارد)

نتایج و تفاوت دو کانولوشن مختلف در دو لایه ی اول شبکه :



Train loss برای conv معمولی در اردر 40 تا 50 می باشد.

البته از معایب این کانولوشن ،Computation cost بالاتر نسبت به کانولوشم معمولی می باشد. که می توان این را از تفاوت زمانی هر ایپاک یافت.

```
LR 0.0001
100%|          | 160/160 [03:19<00:00,  1.25s/it]
train: loss: 10.525935
100%|          | 32/32 [00:22<00:00,  1.39it/s]
val: loss: 12.555154
3m 44s
```

an epoch with nn.conv2d

```
LR 0.001
100%|          | 160/160 [05:01<00:00,  1.89s/it]
train: loss: 10.823757
100%|          | 32/32 [00:26<00:00,  1.21it/s]
val: loss: 12.220775
5m 29s
```

an epoch with deform conv2d

سوالات نظری

(آ) تفاوت بین شبکه‌های کانولوشنی عادی و شبکه‌های کانولوشنی Deformable را از نظر grid sampling مقایسه کرده و نتایج مقایسه خود را بنویسید.

(ب) شبکه‌های Deformable چگونه می‌توانند انعطاف‌پذیری را در Geometric transformation در تصاویر را به وجود آوردند.

(ج) به عقیده شما چرا شبکه‌های کانولوشنی ساده در مواجهه با تصاویری که آبجکت‌های تصویر دارای تغییر یا چرخش فضایی زیادی می‌باشد دچار مشکلات جدی‌ای می‌شود؟

(د) چگونه آفست‌های موجود در Deformable Convolution محاسبه می‌شوند؟

آ) شبکه های کانولوشنی عادی و شبکه‌ای کانولوشنی Deformable از دیدگاه sampling grid تفاوت‌های مهمی دارند که عملا دستاورد این کانولوشن بدلیل همین تفاوت می باشد.

در شبکه‌ای کانولوشنی عادی، sampling grid یکنواخت و ثابت است. به معنا که هر بخش از ورودی تنها می‌تواند به نقطه‌های خاصی از فضای ورودی (و یا همان feature space) نگاشت شود، بدون امکان تطبیق با الگوهایی که در نقاط متفاوتی از تصویر وجود دارند.

اما در شبکه‌ای کانولوشنیDeformable ، sampling grid دارای انعطاف پذیری بیشتری است. این به معنای این است که این شبکه قادر است بهترین مکان‌ها برای نگاشت ویژگی‌ها را برای هر ناحیه از تصویر پیدا کرده و از لحاظ مکانی قابل تطبیق است. با استفاده از این انعطاف پذیری، می‌توان تاثیر بخش‌های مختلف تصویر را بر روی **یادگیری** و تشخیص الگوها بهبود داد.

و به طور کلی می توان گفت ، از نظرsampling grid ، شبکه‌ای کانولوشنی عادی sampling grid ثابت دارد در حالی که در شبکه‌ای کانولوشنی Deformable انعطاف پذیری بیشتری دارد که بهبودی در نحوه‌ی نمون‌برداری از تصویر و در نهایت در کارایی مدل می‌تواند داشته باشد.

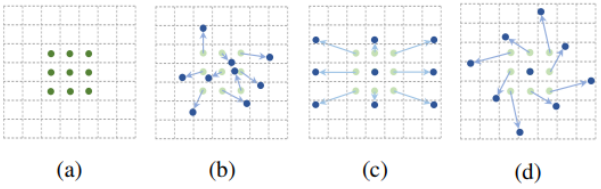


Figure 1: Illustration of the sampling locations in 3×3 standard and deformable convolutions. (a) regular sampling grid (green points) of standard convolution. (b) deformed sampling locations (dark blue points) with augmented offsets (light blue arrows) in deformable convolution. (c)(d) are special cases of (b), showing that the deformable convolution generalizes various transformations for scale, (anisotropic) aspect ratio and rotation.

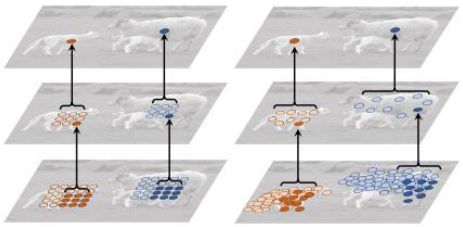


Figure 5: Illustration of the fixed receptive field in standard convolution (a) and the adaptive receptive field in deformable convolution (b), using two layers. Top: two activation units on the top feature map, on two objects of different scales and shapes. The activation is from a 3×3 filter. Middle: the sampling locations of the 3×3 filter on the preceding feature map. Another two activation units are highlighted. Bottom: the sampling locations of two levels of 3×3 filters on the preceding feature map. Two sets of locations are highlighted, corresponding to the highlighted units above.

ب) به گفته ی خود مقاله ، برای این که شبکه نسبت تغییرات جزعی invariant باشد ، دو راه حل وجود دارد که اولی data augmentation می باشد که گاهی اوقات هزینه ی زیادی برای ما دارد و دومی استفاده از transformation-invariant features and algorithms ! و این کانولوشن معرفی شده به نوعی همچین الگوریتمی است که شبکه را نسبت به Geometric transformation ، robust می کند.

و در ناتوانی CNN معمولی در این مورد، در مقاله به اینصورت آمده است :

" There lacks internal mechanisms to handle the geometric transformations. This causes noticeable problems. For one example, the receptive field sizes of all activation units in the same CNN layer are the same. This is undesirable for high level CNN layers that encode the **semantics over spatial locations**. "

برای مثال، اگر یک تصویر حاوی یک شیء باشد که در زوایای مختلف گرفته شده است، شبکه‌های کانولوشنی ساده ممکن است سختی در تشخیص آن شیء داشته باشند چون الگوهای چرخش و انتقال آن شیء را تشخیص نمی‌دهند.

و راه حل این مشکل adaptive determination of scales or receptive field sizes می باشد. به اینصورت که :

It adds two-D **offsets** to the **regular grid sampling** locations in the standard convolution. It enables free form deformation of the sampling grid. The offsets are learned from the preceding feature maps, via additional convolutional layers. Thus, the deformation is conditioned on the input features in a local, dense, and adaptive manner.

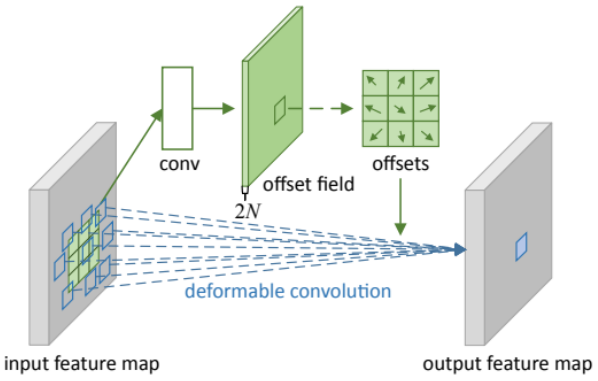


Figure 2: Illustration of 3×3 deformable convolution.

ج) جواب این سوال در بخش قبل داده شده است.

د) در شبکه‌های عصبی کانولوشنال deformable ، offset ها بر اساس input feature map و پارامترهای learnable آفست محاسبه می‌شوند.

محاسبه افست ها توسط یک زیرشبکه اضافی انجام می شود که feature map را به عنوان ورودی می گیرد و مقادیر افست را خروجی می کند. سپس از این مقادیر افست برای محاسبه مکان‌های نمونه‌گیری (regular grid sampling) در عملیات کانولوشن استفاده می‌شود و به شبکه اجازه می‌دهد تا الگوهای پیچیده‌تری را بیابد و عملکرد خود را بهبود بخشد.

عملیات کانولوشن عادی به صورت زیر می باشد:

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n), \qquad (1)$$

که R در اینجا receptive field می باشد. عمل deformable convolution به صورت زیر می باشد :

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n + \Delta\mathbf{p}_n). \qquad (2)$$

که در اینجا Δp_n همان آفست می باشد. و محاسبه ی آن به صورت کلی در مقاله به شکل زیر است (هر چند که خودم چیزی ازش نفهمیدم) :

In the deformable convolution Eq. (2), the gradient w.r.t. the offset $\Delta\mathbf{p}_n$ is computed as

$$\frac{\partial \mathbf{y}(\mathbf{p}_0)}{\partial \Delta\mathbf{p}_n} = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \frac{\partial \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n + \Delta\mathbf{p}_n)}{\partial \Delta\mathbf{p}_n}$$
$$= \sum_{\mathbf{p}_n \in \mathcal{R}} \left[\mathbf{w}(\mathbf{p}_n) \cdot \sum_{\mathbf{q}} \frac{\partial G(\mathbf{q}, \mathbf{p}_0 + \mathbf{p}_n + \Delta\mathbf{p}_n)}{\partial \Delta\mathbf{p}_n} \mathbf{x}(\mathbf{q}) \right], \qquad (7)$$

اما در عمل به اینگونه است که برای محاسبه ی offsets ، از یک کانولوشن استفاده می کنیم که فیچر مپ را به عنوان ورودی می گیرد. این لایه 2 کانال دارد و 2N عدد خروجی می دهد که N ، سایز sampling می باشد (N = |R|) ، پیاده سازی آن به این صورت است که فیچر مپ های تولید شده هم سایز با فیچر مپ ورودی هستند و کانال خروجی اول این کانولوشن نشان دهنده ی میزان جابه جایی در جهت yو کانال دوم ، میزان جابه جایی هر پیکسل در جهت x را نشان می دهد.

(البته در بعضی پیاده سازی ها تعداد کانال offset را برابر 2*kernelsize*kernelsize در نظر می گیرند، که خودم این را هم ایمپلمنت کردم ولی بدلیل کامپوتیشن بسیار بالای آن ، همین معماری قبلی را ترجیح دادم)

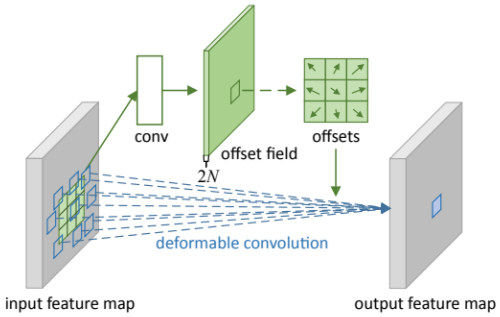


Figure 2: Illustration of 3 × 3 deformable convolution.

```
self.offset_conv = nn.Conv2d(in_channels,deformable_groups * 2,
    kernel_size=self.kernel_size, stride=stride, padding=self.padding,
    dilation=self.dilation, bias=False
)
```