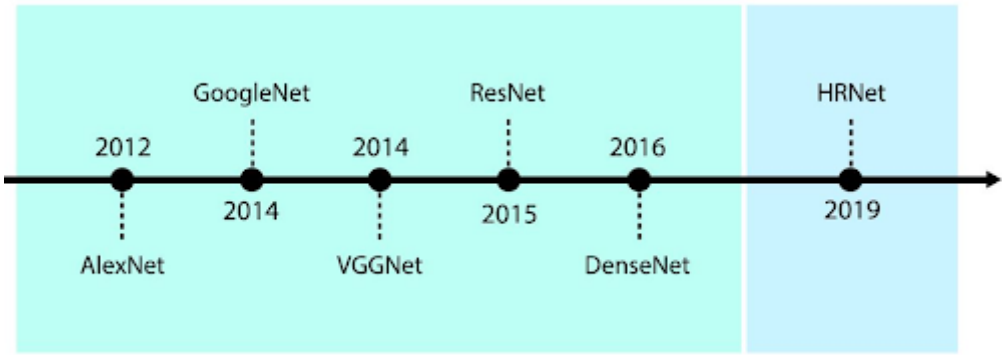
[DEEPLEARNING][CV] HRNET 정리 HIGH-RESOLUTION NETWORK: A UNIVERSAL NEURAL ARCHITECTURE FOR VISUAL RECOGNITION

—

Key Detection에서 너무나 핫한 HRNet에 대해 정리해 보겠다. ㅎㅎ 나도 이건 공부하면서 너무 재미있었다.

HRNet 정리

High-Resolution Network: A universal neural architecture for visual recognition



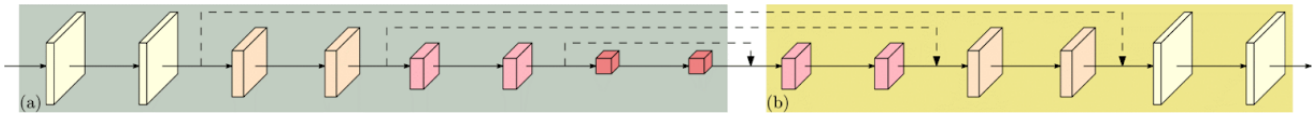
AlexNet, GoogLeNet, VGGNet, ResNet, DenseNet.. 이렇게 계속 CV task가 발전해 왔고, 이런 맥락에서 CV Task는 이제 Classification, 즉 분류를 중심 발전했다. 그래서 **HRNet** 연구자들은 *General* CV task, *Universial* Architecture 등의 표현을 쓰며 분류만의 문제에서 벗어나려는 시도를 한 것 같다. 아무래도 기존의 CV task들과 조금 벗어나 널리.. 여기저기.. 여러 곳에 잘 쓰이는 좀 general한 net을 만들고 싶었구나 하는 게 정리 여기저기서 느껴진다.

HRNet은 Semantic Segmentation, Human Pose Estimation, Object Detection에서 좋은 결과를 거두었다. 일단 1절에서 도대체 예전 것들과 구조가 어떻게 다른지를 알아보기 위해 **아키텍처**에 대해 다루고, 2절에서 HRNet이 **활약하고 있는 task**들(응용들)에 대해 다뤄 보도록 하겠다.

1) 기본 구조 Basic Architecture

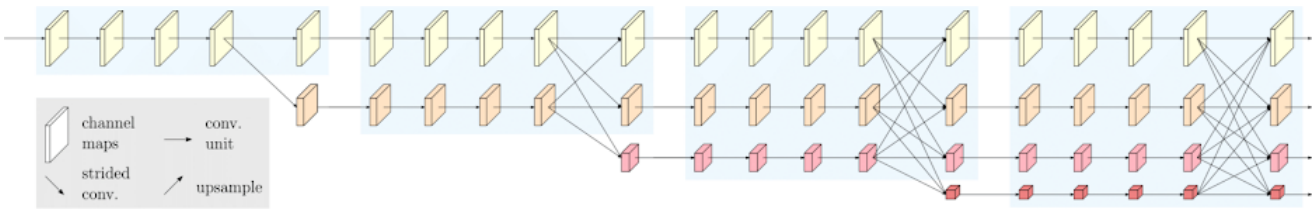
HRNet의 기본 구조를 보도록 하자. HRNet은 HRNet에 대한 여러 논문들*Deep High-Resolution Representation Learning for Visual Recognition, IEEE 2020*에서도 계속 강조했듯 **전 프로세스 과정에서 high-resolution representation을 유지**하기 위한 구조를 채택했다.

이해를 돕기 위해 HRNet 이전의 AlexNet 계통의 구조를 먼저 확인해 보자. 그 류의 구조에서 Input은 High-Resolution Stream이었으나 (Classification이 목표이므로) **작업을 거치며 Low-Resolution으로 변환**한다. 이런 걸 High-to-Low Convolutions가 존재한다고 한다.
(b)의 구조는 SegNet, DeconvNet, encoder-decoder, SimpleBaseline 등이 채택하고 있는 구조라고 하는데, (a)의 단계를 한 후 다시 High Stream으로 변환한다.



그리고 이런 것이 거의 LeNet-5 이후로 굳어진 업계의 표준이라고 한다.

다시 HRNet으로 돌아와서, HRNet은 이렇게 전통적으로 high에서 low로 변환하는 과정을 거치지 않는다. 그저 high로 다 때려박는다.(유지한다.)



위의 전통적인 Net들과 달리 HRNet에서는 **High한 표현을 유지**하려고 했다. high stream으로 시작해서, high-to-low한 표현을 점진적으로 추가하기는 했는데, 이제 그런 multi stream 표현을 **parallel하게 연결**해서 high한 표현을 유지해 준 것이다.

위 그림을 보면 이해가 조금 쉬운데, 위 그림은 이제 4개 part로 이루어졌다. 첫 번째 파트에서는 high stream을 계속 유지했고 두 번째 파트로 넘어갈 때 high-to-low convolution stream을 생성했다. 그리고 그걸 parallel하게 연결한다. 다음 파트로 넘어갈 때도 마찬가지로 동작하고, 그렇게 계속 수행해서 4개의 파트니까 4개 종류의 stream이 만들어진다. 꼭 4단계가 아니라도 상관없는데, n단계에는 n개의 resolution으로 이루어진 n개의 stream이 존재한다.

또 각 단계의 마지막 부분에 **across**하게 정보를 교환하고, 이 과정에서 **multi-resolution** fusion이 수행된다.

이 부분에서 HRNet의 이점을 알 수 있다. HRNet은 semantically하게 강할 뿐 아니라 spatially precise하다고 하는데, 이를 가능하게 해 주는 두 가지 주요 이점이 있다.

(1) Parallel하게 병렬로 연결 → High Resolution을 유지할 수 있었다. (그렇지 않으면 low에서 high로 변환하는 귀찮은 과정을 또 해 줘야 한다.) 그리고 이렇게 연결해 줘서 공간적으로 더 명확한 학습을 할 수 있다.

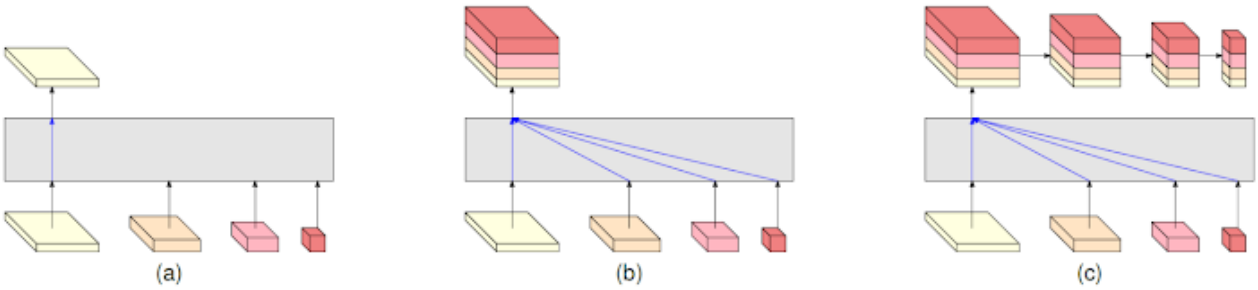
(2) Multi-Resolution fusions 반복 → High Resolution Representation 기능을 향상시키는데, 그 반대도 가능하다. → high-to-low 표현이 더욱 강력해진다. (기존의 방법들은 high-resolution low-level을 aggregate하고 low-resolution high-level representaion을 unsample 하는 식으로 사용했다고 한다.)

2) 응용 Applications

앞서 서두에서 말했듯 HRNet 연구하시는 분들은 이 모델이 **universial** architecture이라는 사실을 굉장히 강조하신다.
HRNet의 시작은 이제 CVPR '19에서 발표된 *Deep High-Resolution Representation Learning for Human Pose Estimation*으로, 제목에서도 알 수 있듯.. 인간의 포즈가 뭔지를 보셨다. 그리고 이건 이제 Human Pose Detection Task 측면에서 표준이 됐다고 한다. (근데 그런 것 같다. 그래서 내가 지금 이걸..)

시작으로 매우 많은 주목을 끈 후에는 이제 Object Detection, Face Detection, 뭐 Facial Landmark Detection 등의 많은 퍼포먼스 류에서 좋은 성능을 거두었다. 2020년 IEEE에 등재된 논문인 *Deep High-Resolution Representation Learning for Visual Recognition*, CVPR '20에서 발표된 *HigherHRNet: Scale-Aware Representation Learning for Bottom-up Human Pose Estimation* 등등.. bottom-up 동작 탐지에서 scale의 다양성을 주기 위한 고해상도 multi scale 표현을 학습시키기 위해 HRNet을 확장하려는 시도를 좀 했고 뭐 좋은 성과를 거두었다고 한다. 이 논문들은 내가 아직 안 읽어 봐서 잘 모르겠고..

그래서 이제 Task별로 version이 갈린다. HRNetV1, HRNetV2, HRNetV2p.



(a)는 Human Pose Estimation에서 쓰이는 HRNetV1. 마지막에 high-resolution conv stream에서만 output 뽑는다.

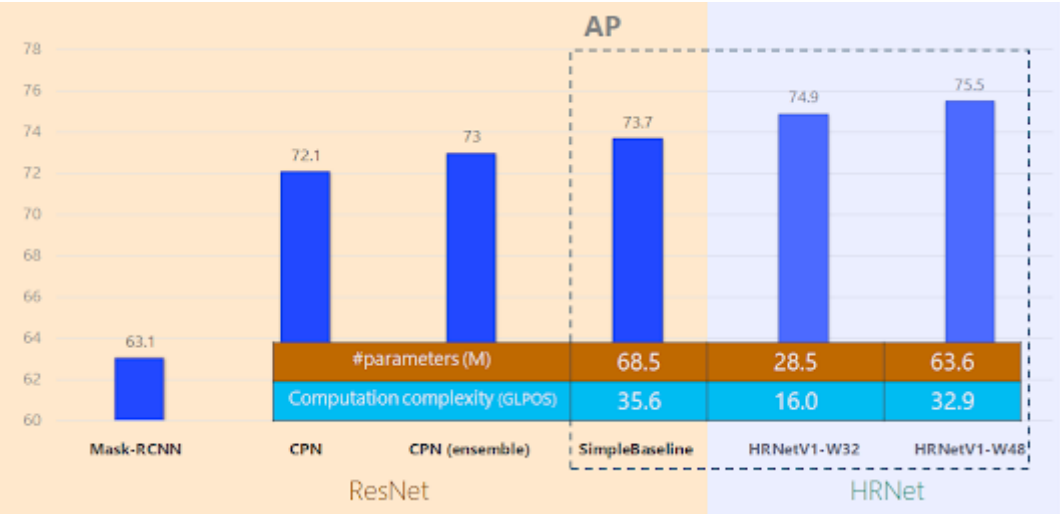
(b)는 Semantic Segmentation에서 쓰이는 HRNetV2. 모든 resolutions에서 다 가져와서 연결해 output 뽑는다.

(c)는 HRNetV2p이고 HRNetV2의 output에서 *feature pyramid* 형성한다. (관련 논문에서 되게 중요하게 설명해 주시며 실재로 중요하지만 일단은 넘어가도록 한다.)

여기서 회색 상자는 출력 표현 얻는 방법을 설명한다. 그럼 이제 Task별로 자세히 보겠다.

HRNetV1: Human Pose Estimation (Key Point Detection)

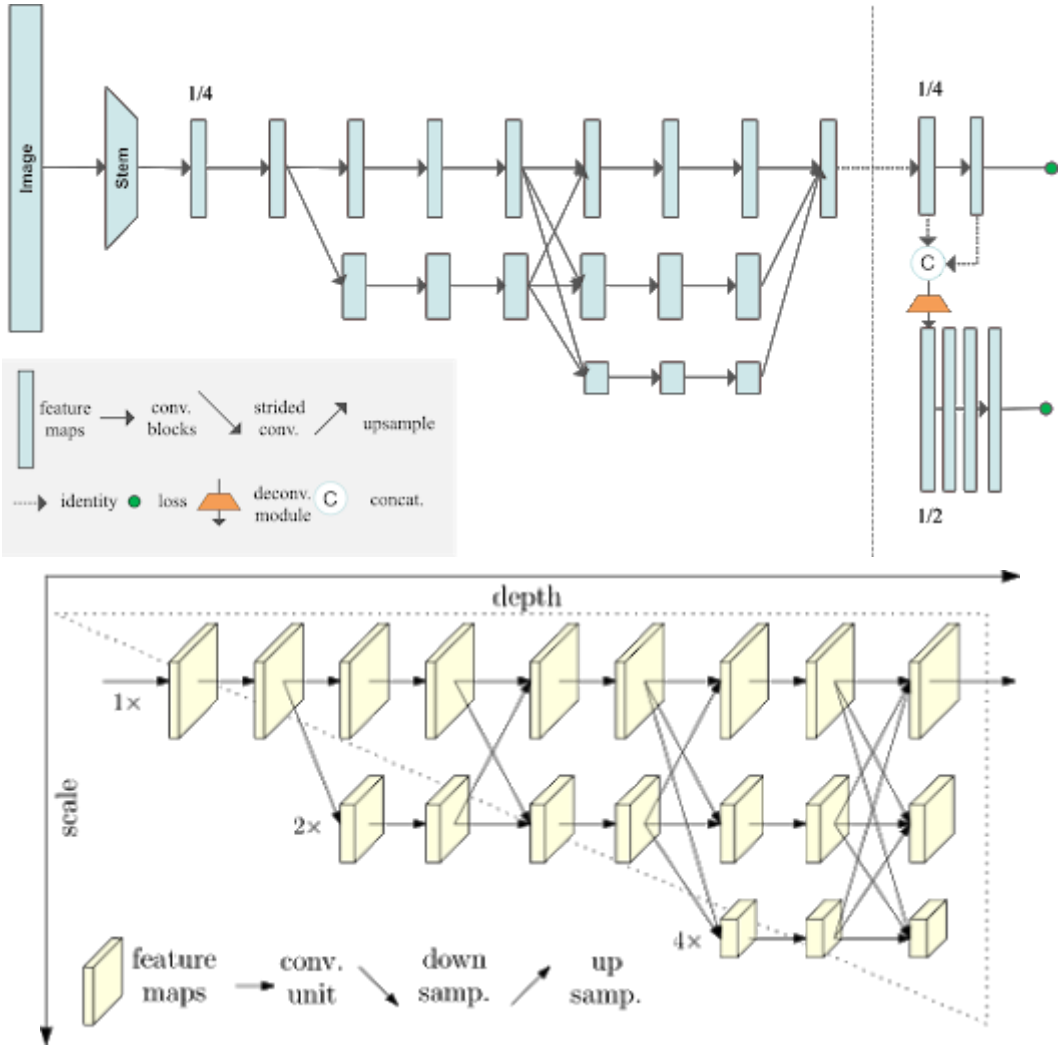
Human Pose Estimation은 주요 지점 몇 개(Keypoint)를 이용해서 포즈를 감지하는 것을 목표로 하는 Task이다. 여기에서는 COCO human pose estimation을 이용했고 밑에도 모두 COCO다.



결과를 보면 ResNet보다 더 좋은 결과를 지닌다. parameters와 computation complexity 모두에서 HRNet이 승리.

method	Backbone	#Params	GFLOPs	AP	AR
Mask-RCNN	ResNet-50-FPN	-	-	63.1	-
CPN (ensemble)	ResNet-Inception	-	-	73.0	79.0
SimpleBaseline	ResNet-152	68.6M	35.6	73.7	79.0
HRNetV1	HRNet-W32	28.5M	16.0	74.9	80.1
HRNetV1	HRNet-W48	63.6M	32.9	**75.5**	**80.5**
HRNetV1+ extra data	HRNet-W48	63.6M	32.9	**77.0**	**82.0**

추정 성능인 AP에서도 꿇리지 않는다.



구조는 이렇게 생김. 완전 기본 net으로 가장 high resolution conv stream에서만 output 출력한다.

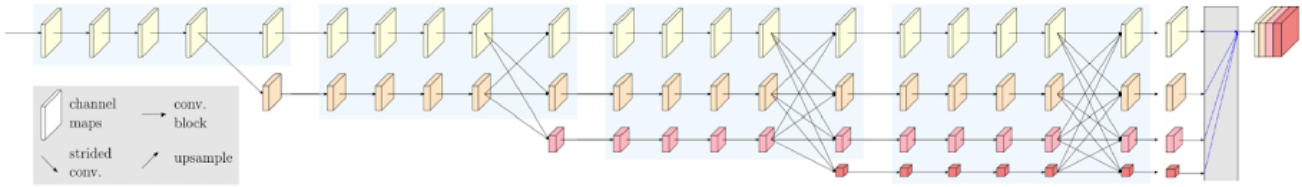
HRNetV2: Semantic Segmentation

Semantic Segmentation은 각 픽셀에다가 class label을 할당하는 task이다.

	backbone	mIoU
DeepLab	Dilated-ResNet-101	70.4
PSANet	Dilated-ResNet-101	80.1
DenseASPP	WDenseNet-161	80.6
HRNetV2	HRNet-W48	**81.6**
HRNetV2 + OCR	HRNet-W48	**82.5**

	Backbone	#Params.	GFLOPs	mIoU
U-Net++	ResNet-101	59.5M	748.5	75.5
DeepLabv3	Dilated-ResNet-101	58.0M	1778.7	78.5
DeepLabv3+	Dilted-Xception-71	43.5M	1444.6	79.6
PSPNet	Dilated-ResNet-101	65.9M	2017.6	79.7
HRNetV2	HRNet-W40	45.2M	493.2	80.2
HRNetV2	HRNet-W48	65.9M	747.3	**81.1**

그 분야에서 좀 핫한 다른 Net들과 비교해 봐도 더 적은 parameter과 더 적은 GFLOPs(complexity)를 이용해서 더 좋은 결과(mIoU)를 얻은 것을 볼 수 있다. 더 많은 정보는 앞서 말한 IEEE 20년 4월자 논문을 확인.



HRNetV2는 이런 식으로 마지막에 모든 stream들을 쌓아 준다.

	Human pose estimation		Semantic segmentation		
	SimpleBaseline-ResNet-152	HRNetV1-W32	PSPNet	DeepLabV3	HRNetV2-W48
train memory	14.8G	5.7.G	14.4G	13.3G	13.9G
inference memory/image	0.29G	0.13G	1.60G	1.15G	1.79G
train seconds/iteration	1.085	1.153	0.837	0.850	0.692
inference seconds/image	0.030 (0.012)	0.057 (0.015)	0.397	0.411	0.150
AP/mIoU	72.0	74.4	79.7	78.5	81.1

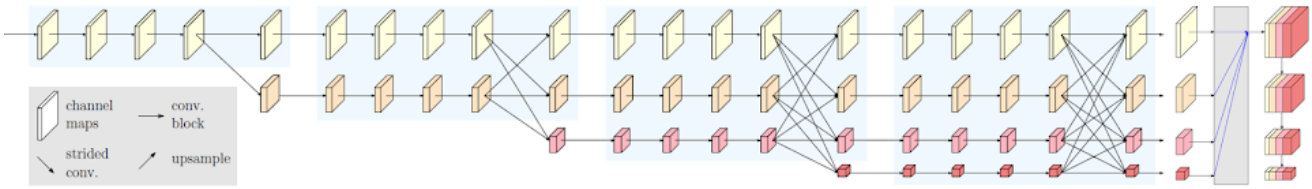
성능이 좋다 보니 이제 Runtime Cost가 많이 들지 않을까? 너무 Expensive한 모델이 아닌가? 싶은 생각이 들 수 있으나 여기에서는 그것조차 반박해 준다. 뭐.. 그냥 다른 모델보다 뛰어나다는 것을 저 표에서 확인할 수 있다.

HRNetV2p: Object Detection and Instance Segmentation

여기서도 COCO 썼다. Object Detection에서는 Object의 bounding box를 식별하는 거고 Instance Segmentation은 Object가 속한 픽셀까지 찾는 Task이다. 자율 주행이 핫하면서 매우 뜨는 분야.. 여기서도 역시 ResNet과 ResNeXt를 이겼다고 하신다.

	Backbone	AP	AP_S	AP_M	AP_L
Faster R-CNN	ResNet-101-FPN	40.3	22.6	43.1	51.0
Faster R-CNN	HRNet2p-W32	41.1	24.0	43.1	51.4
Faster R-CNN	ResNeXt-101-64x4d-FPN	41.1	23.5	44.1	52.3
Faster R-CNN	HRNetV2p-W48	42.4	24.9	44.6	53.0
Cascade R-CNN	ResNet-101-FPN	42.8	23.7	45.5	55.2
Cascade R-CNN	HRNet2p-W32	**43.7**	**25.5**	46.0	**55.3**

backbone	mask				bbox			
	AP	AP_S	AP_M	AP_L	AP	AP_S	AP_M	AP_L
ResNet-50-FPN	35.0	16.0	**37.5**	**52.0**	38.6	21.7	41.6	50.9
HRNet2p-W18	**35.3**	**16.9**	**37.5**	51.8	**39.2**	**23.7**	**41.7**	**51.0**
ResNet-101-FPN	36.7	17.0	39.5	54.8	41.0	23.4	44.4	53.9
HRNet2p-W32	**37.6**	**17.8**	**40.0**	**55.0**	**42.3**	**25.0**	**45.4**	**54.9**

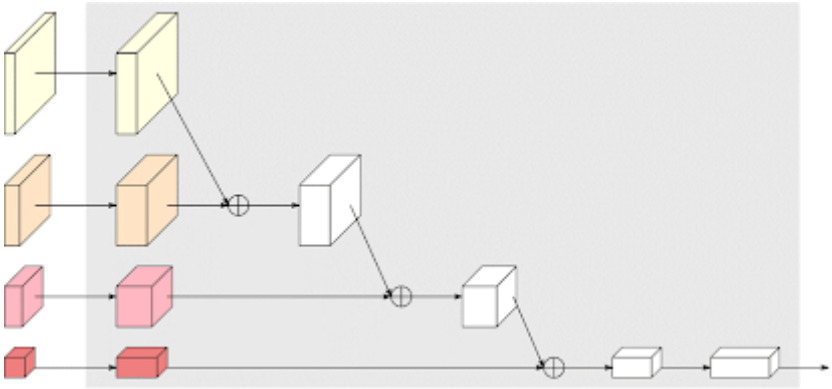


구조는 이렇게 되어 있다. V2에서 마지막에 pyramid가 포함된 형태를 확인할 수 있다.

	Object detection (Faster R-CNN)			
	ResNet-101	ResNext-101	HRNetV2p-W32	HRNetV2p-W48
train memory	5.4G	9.5G	8.5G	11.3G
inference memory/image	0.62G	0.77G	0.51G	0.79G
train seconds/iteration	0.550	1.183	0.690	0.965
inference seconds/image	0.087	0.144	0.101	0.116
AP	39.8	40.8	40.9	41.8

여기서도 이제 cost를 비교해 줬다.

+ ImageNet Pretraining



	#Params.	GFLOPs	Top-1 err.	Top-5 err.
ResNet-50	25.6M	3.82	23.3%	6.6%
HRNet-W44-C	21.9M	3.90	**23.0%**	**6.5%**
ResNet-101	44.6M	7.30	21.6%	5.8%
HRNet-W76-C	40.8M	7.30	**21.5%**	**5.8%**
ResNet-152	60.2M	10.7	21.2%	5.7%
HRNet-W96-C	57.5M	10.2	**21.0%**	**5.7%**

결론적으로 HRNet은 이제 visual recognition에 특화된 universal architecture라고 볼 수 있다. 앞서 설명한 application들이 아니더라도 정말 여러 visual recog에 적용될 수 있다고 한다.

그리고 HRNet+ASPP 모델이 Mapillary panoptic segmentation에서 좋은 결과를 거두었다고 한다. 뭐.. 아무튼 거꾸로 봐도 HRNet이 여러 vis recog task에 적합한 것을 잘 알 수 있었다.

Reference

https://www.microsoft.com/en-us/research/blog/high-resolution-network-a-universal-neural-architecture-for-visual-recognition 번역/정리하여 재구성하였습니다.