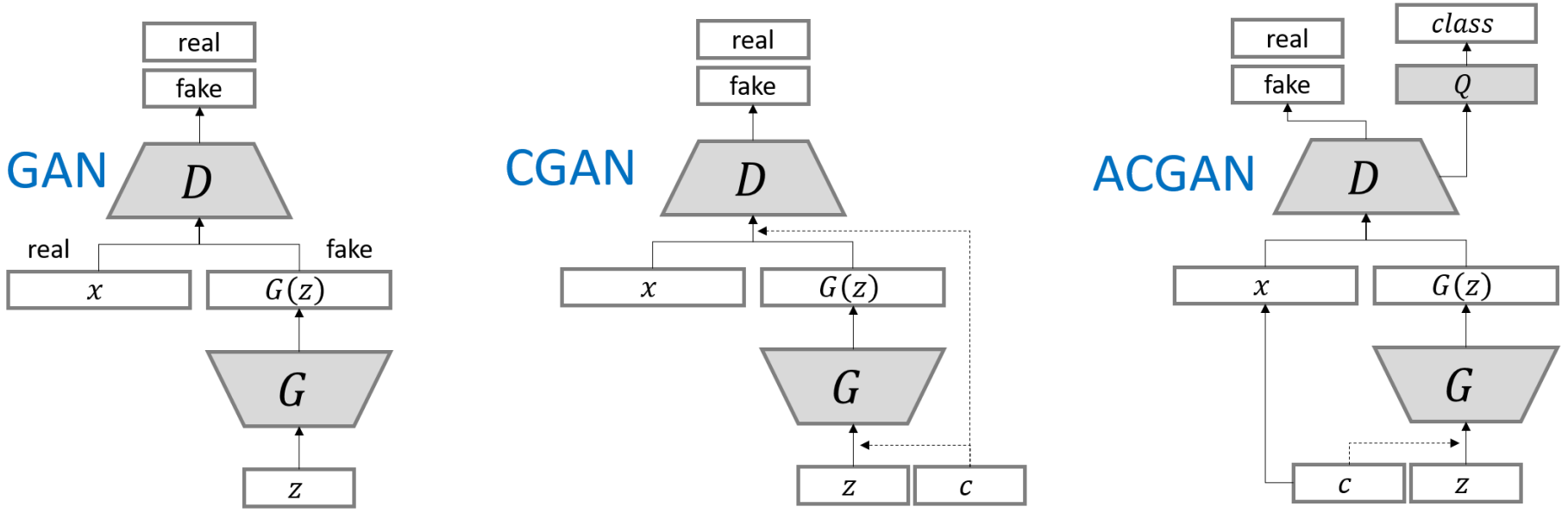
Deep Learning

Mohammad Reza Mohammadi

2021

Conditional GAN

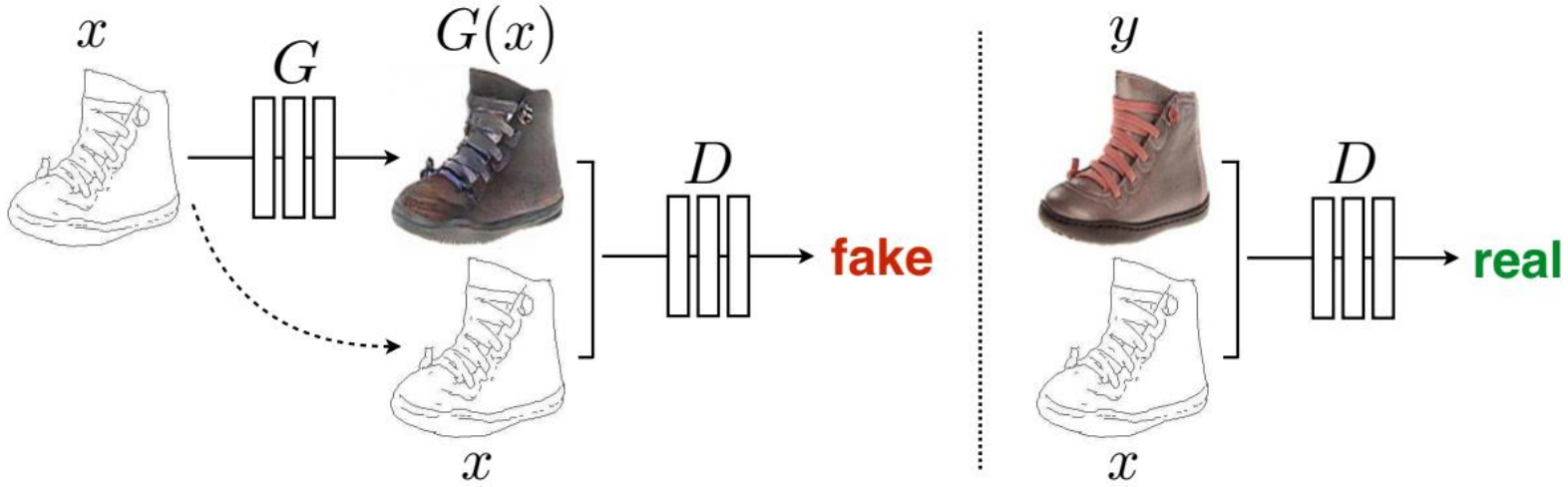
- In a CGAN, the generator and the discriminator are conditioned on c, which could be a class label or some data from another modality
- In Auxiliary Classifier GANs, the discriminator is forced to identify fake and real images, as well as the class of the image, irrespective of whether it is fake or real



A. Odena, C. Olah, and J. Shlens. "Conditional image synthesis with auxiliary classifier gans." *International conference on machine learning*, 2017.

Conditional GAN

- To train such models, we need pairs of images
- The advantage of this method compared to pixel to pixel comparison is that the contents are compared



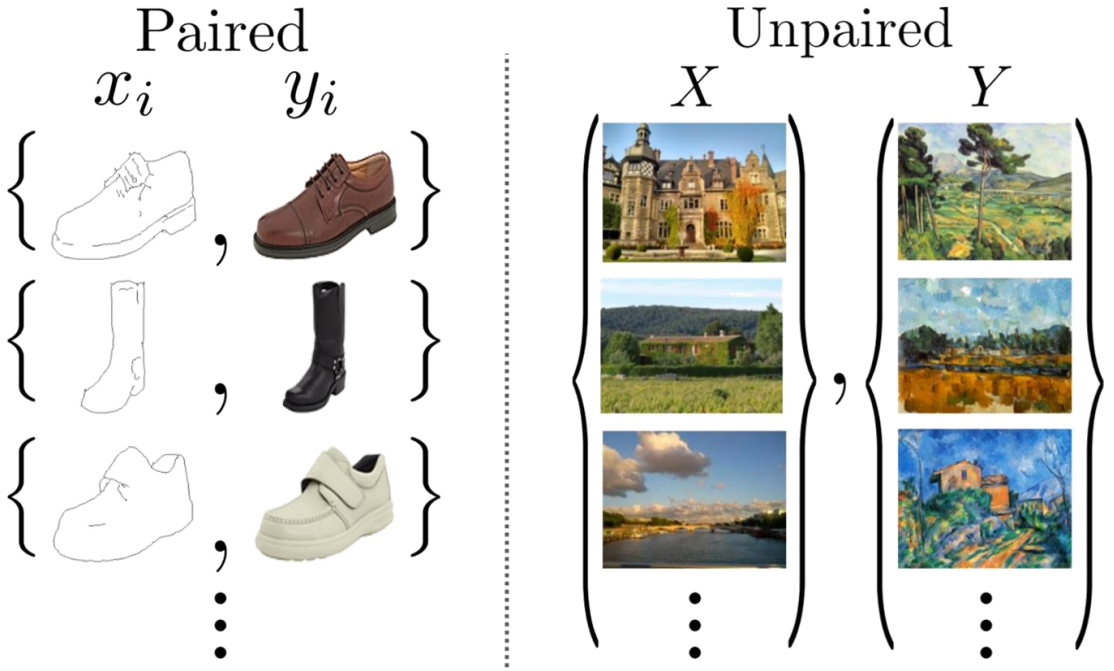
Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." *IEEE conference on computer vision and pattern recognition*. 2017.

CycleGAN

- Collecting pair images is costly or even impossible in many applications
- CycleGAN is proposed to use non-pair images for training
  - How can such a model be trained?

Winter

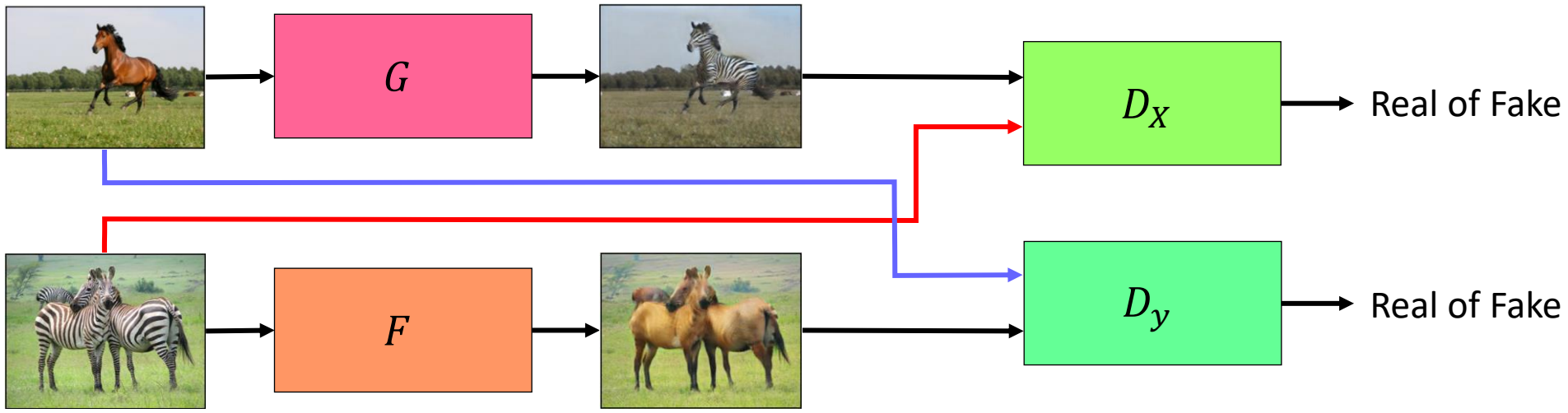
Summer



Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." *IEEE international conference on computer vision*. 2017.

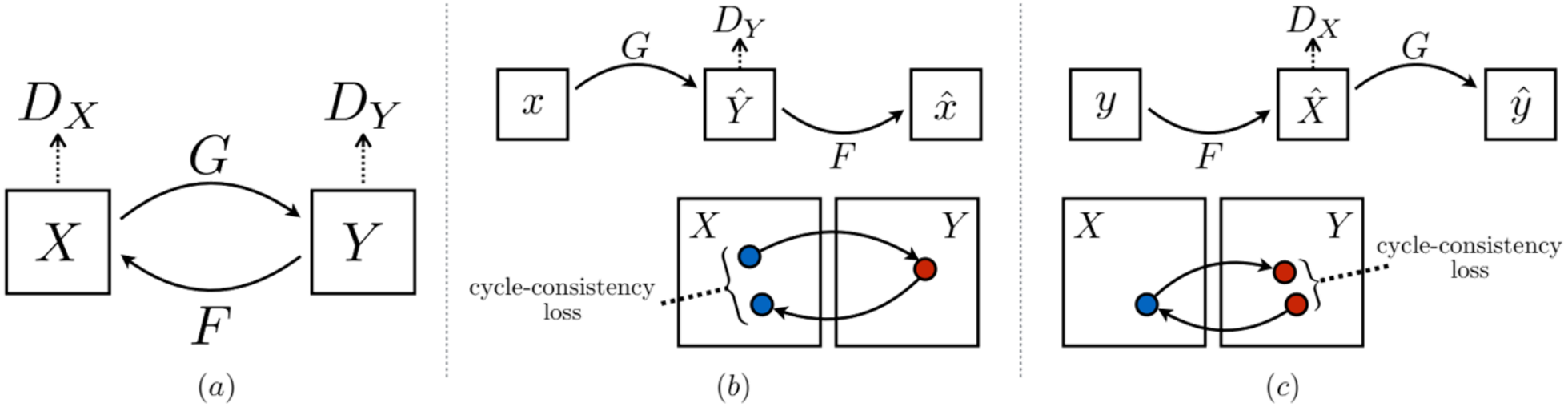
CycleGAN

- CycleGAN idea:
  - Two generators (network G for X to Y and network F for Y to X)
  - Two discriminators (network D_X for domain X and network D_Y for domain Y)
  - The generators should be inverse



CycleGAN

- CycleGAN idea:
  - Two generators (network G for X to Y and network F for Y to X)
  - Two discriminators (network D_X for domain X and network D_Y for domain Y)
  - The generators should be inverse

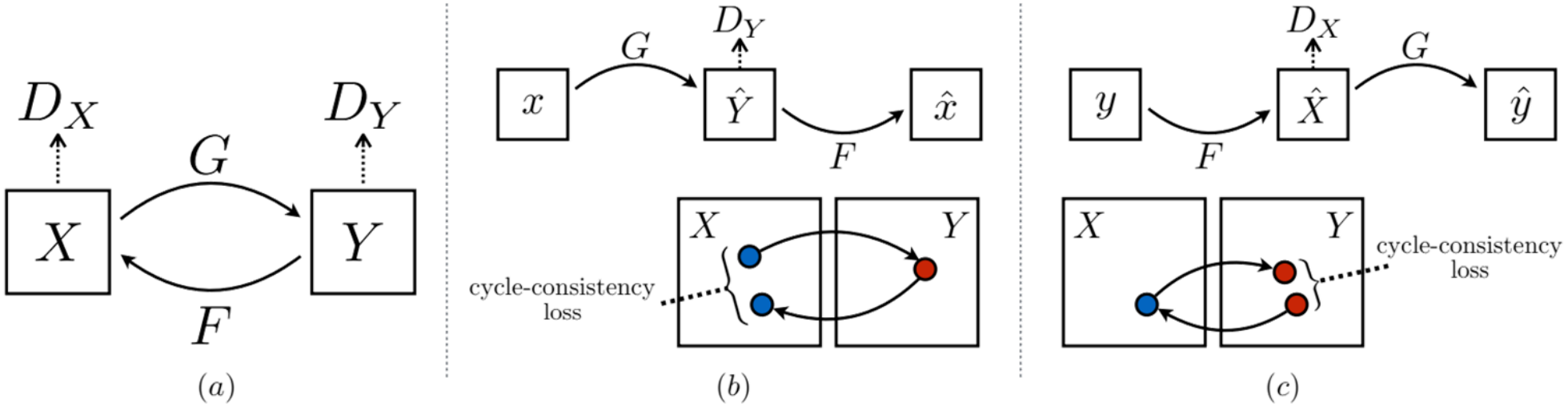


CycleGAN

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))]$$

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]$$

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F)$$



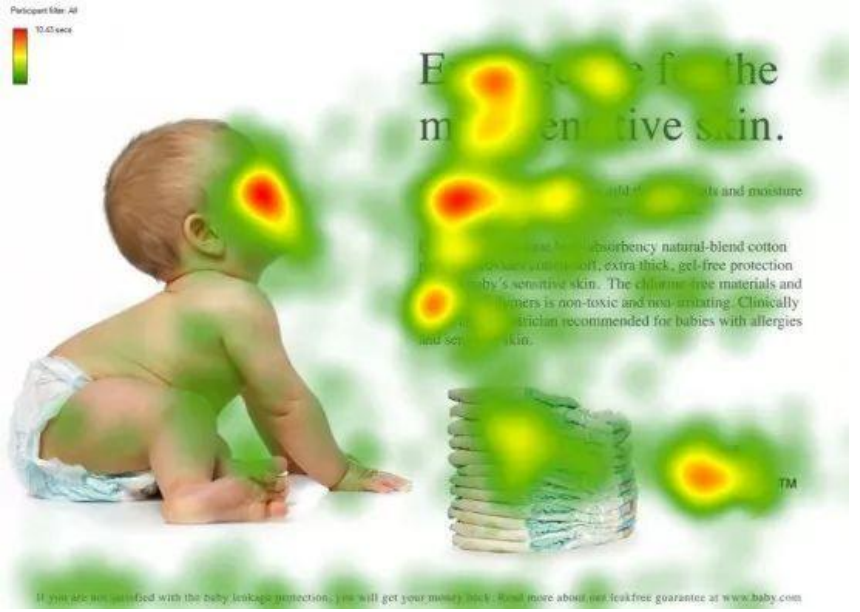
https://github.com/eriklindernoren/Keras-GAN/blob/master/cyclegan/cyclegan.py

Attention

https://shairozsohail.medium.com/a-survey-of-visual-attention-mechanisms-in-deep-learning-1043eb25f343

Visual attention

- Visual attention is so much a part of everyday life
  - Focusing on the road while you're driving
  - Glancing at the food on your plate before you take a bite
  - Looking at the text instead of the sidebars or screen bezel when you're reading
- We should make such a mechanism trainable, so that given a task and a set of images relevant to it, the network starts to learn to "filter out" irrelevant sections of the image to make clearer and more robust judgements for the task



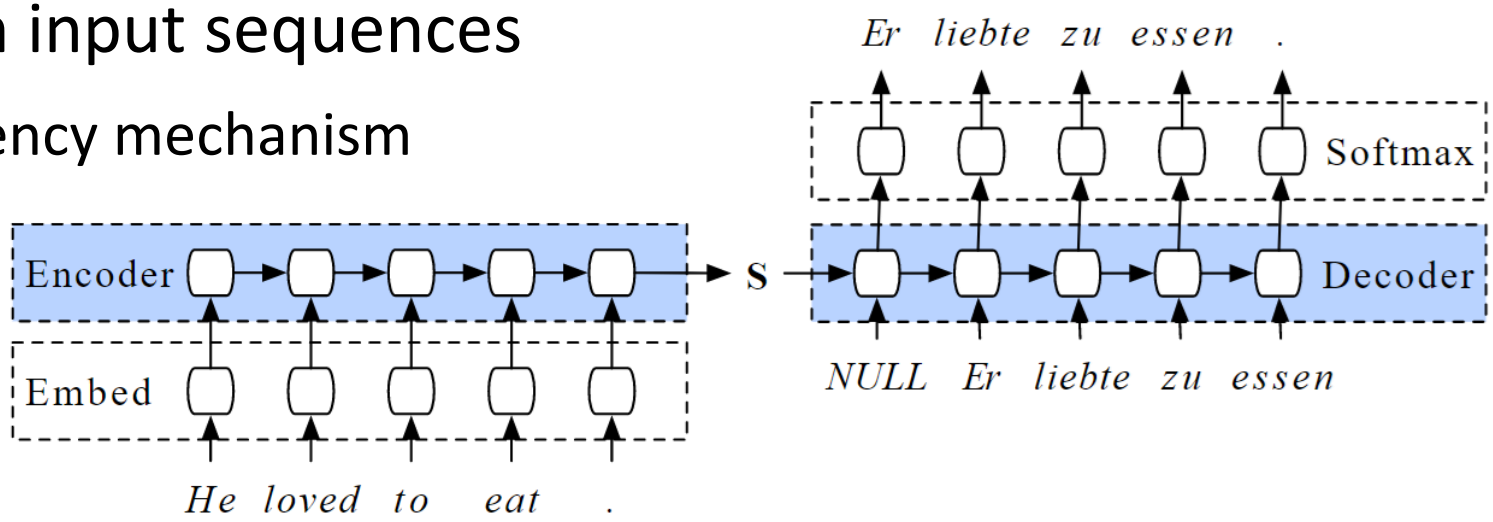
What is attention?

- A mechanism by which a network can weigh features by the level of importance to a task, and use this weighting to help achieve the task
  - Class activation maps generally don't serve to improve performance or reduce computation like trainable attention methods do
- Attention models are widespread among multiple areas of deep learning, and the learned weighting schemes can apply to features as diverse as pixels in an image, words in a sentence, nodes in a graph, or even points in a 3D point cloud

Class activation maps for one object class

Attention

- The idea of attention was born in the area of seq2seq modeling
  - Models are trained to consume a sequence of arbitrary length (such as an English sentence) and output another sequence (such as the same sentence in German)
- The issue with such tasks is that there is often a complicated dependency that ranges far beyond the last sequence element seen, and such a dependency can vary between input sequences
  - Need to learn a flexible dependency mechanism

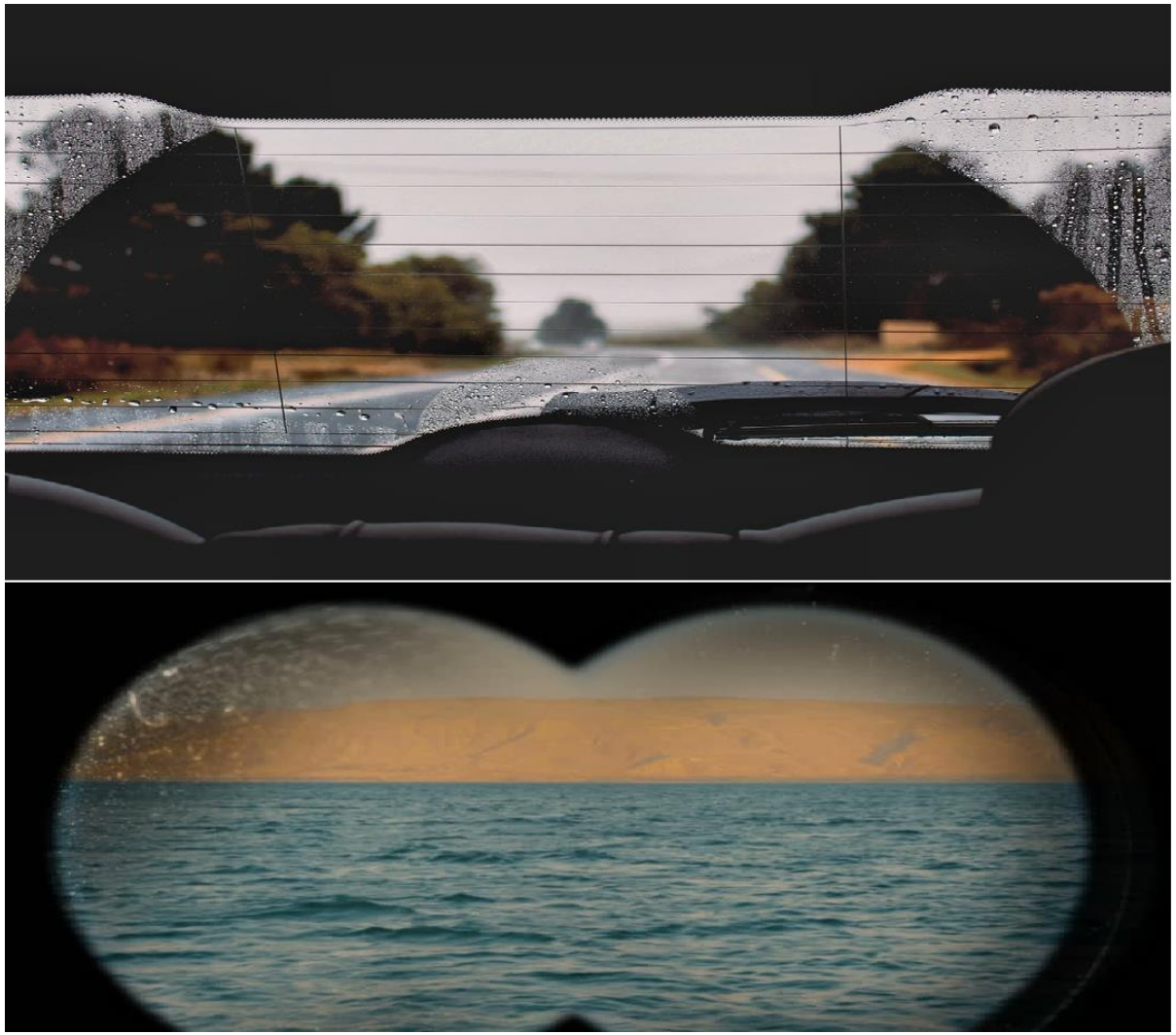


Attention

- In computer vision tasks, the dependency being along the spatial domain
  - The same texture may be present in multiple areas of an image
  - Multiple disjoint semantic cues may give clues to the overall classification of an image
  - An object may have multiple complex and obscured parts throughout an image
- Learn these dependencies beyond the limited receptive field of a convolutional filter is important in capturing maximum performance and allowing our models to build a wider intuition

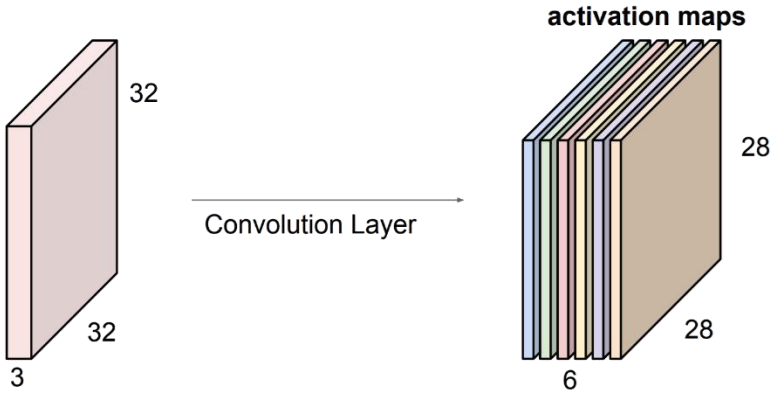
Soft vs. Hard attention

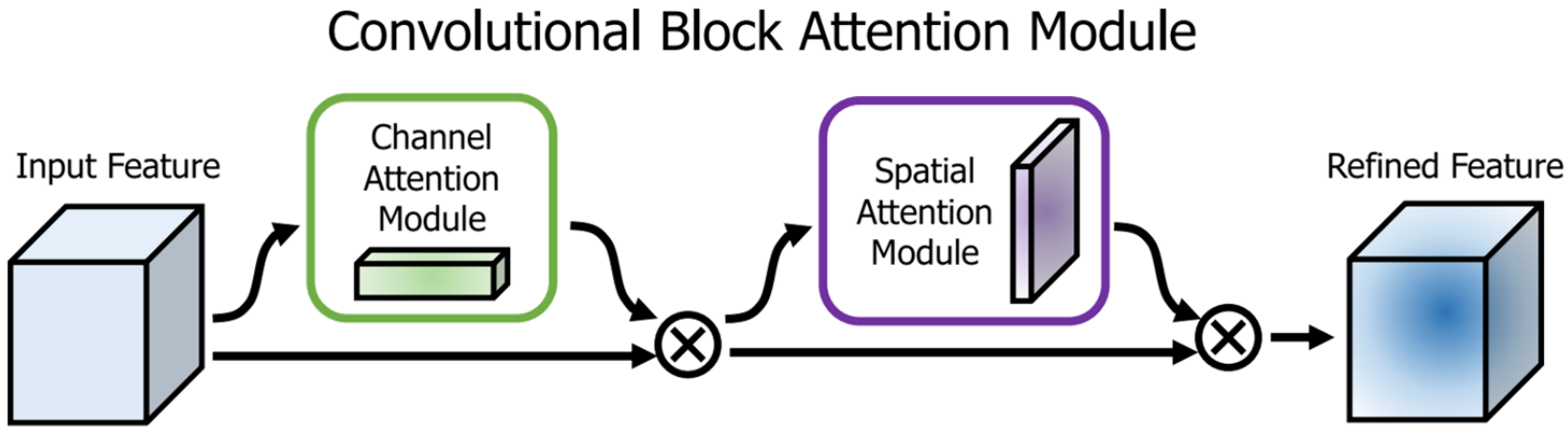
- Looking through a foggy pane of glass represents soft attention, where the entire image is still being “seen”, but certain areas are being attended to more
- The binoculars represent hard attention, where we are only seeing a subset of the image, hopefully the part most relevant to our task



Convolutional block attention module

- Given an intermediate feature map, CBAM sequentially infers attention maps along two separate dimensions, channel and spatial
  - Then, the attention maps are multiplied to the input feature map for adaptive feature refinement
- Focus on important features and suppressing unnecessary ones.



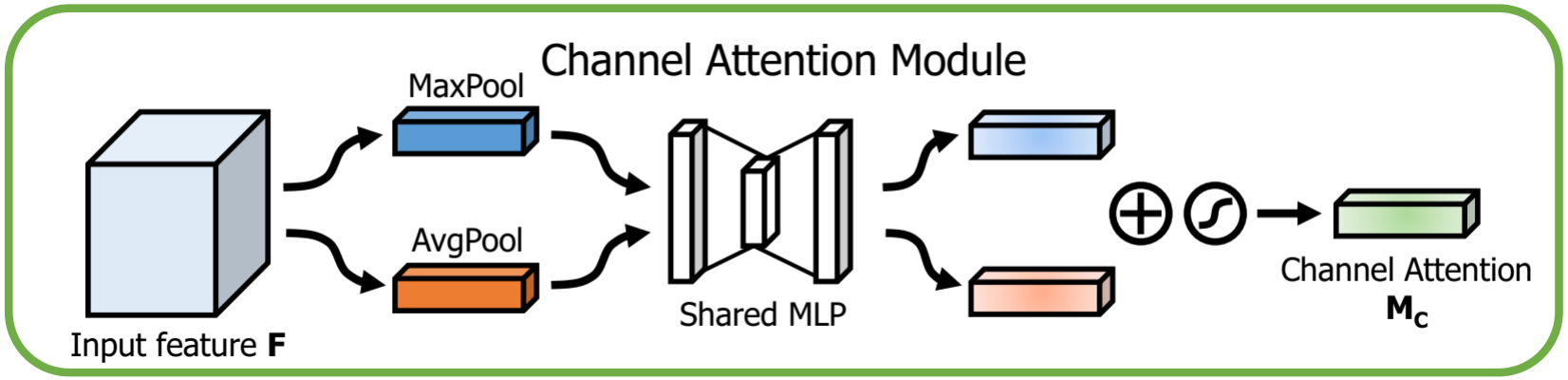


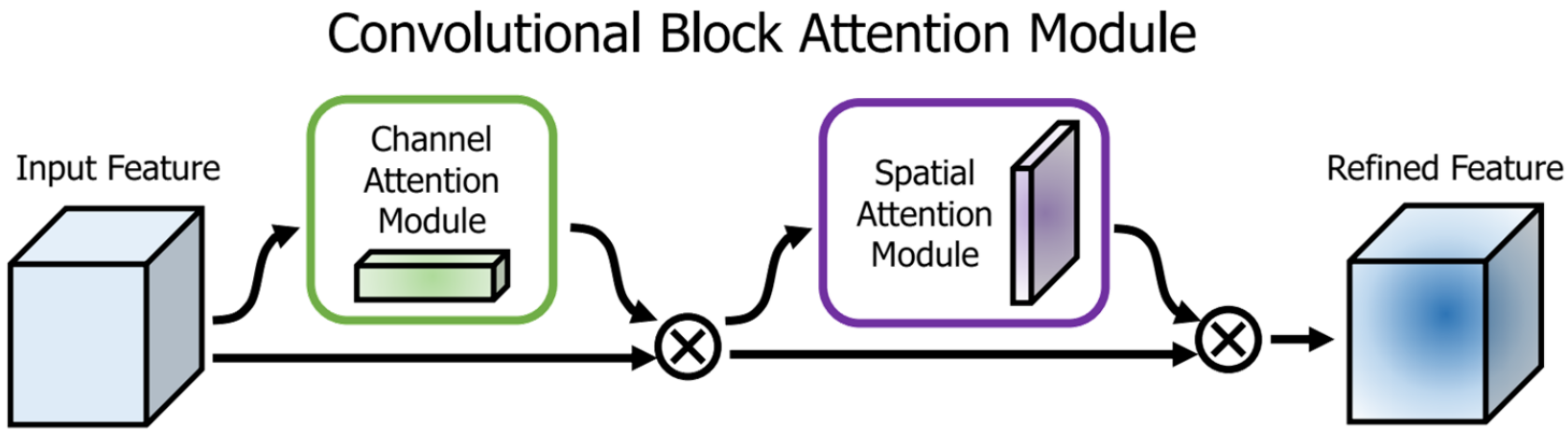
Woo, Sanghyun, et al. "Cbam: Convolutional block attention module." *Proceedings of the European conference on computer vision (ECCV)*. 2018.

Channel attention

- Channel attention focuses on ‘what’ is meaningful given an input image

$$\mathbf{M_c}(\mathbf{F}) = \sigma(\mathbf{W_1}(\mathbf{W_0}(\mathbf{F^c_{avg}})) + \mathbf{W_1}(\mathbf{W_0}(\mathbf{F^c_{max}})))$$

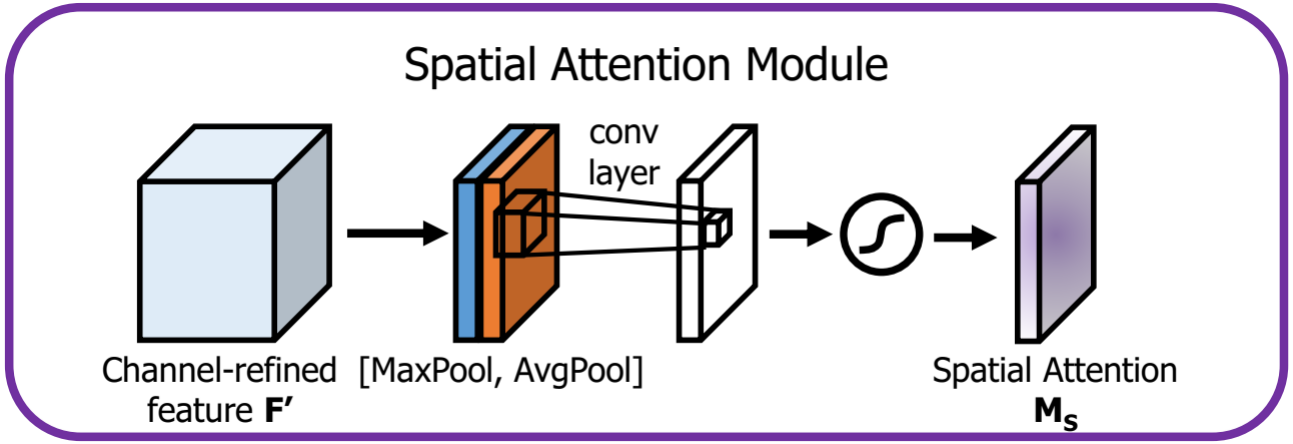


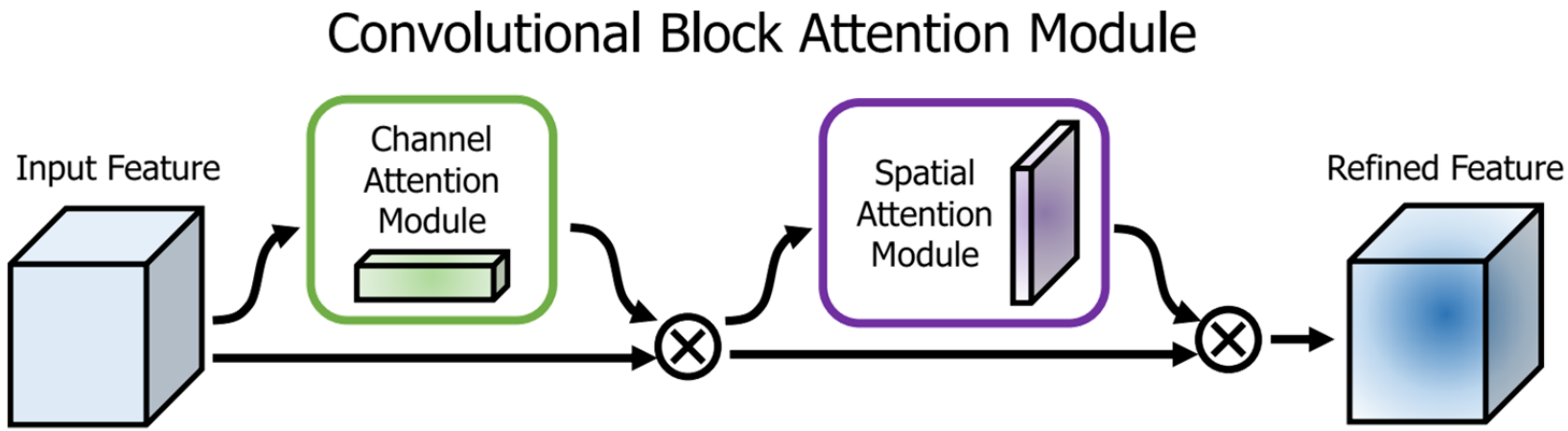


Spatial attention

- Spatial attention focuses on ‘where’ is an informative part

$$\mathbf{M_s}(\mathbf{F}) = \sigma(f^{7\times7}([\mathbf{F^s_{avg}}; \mathbf{F^s_{max}}]))$$





CBAM

Description	Parameters	GFLOPs	Top-1 Error(%)	Top-5 Error(%)
ResNet50 (baseline)	25.56M	3.86	24.56	7.50
ResNet50 + AvgPool (SE [28])	25.92M	3.94	23.14	6.70
ResNet50 + MaxPool	25.92M	3.94	23.20	6.83
ResNet50 + AvgPool & MaxPool	25.92M	4.02	**22.80**	**6.52**

Table 1: **Comparison of different channel attention methods**. We observe that using our proposed method outperforms recently suggested Squeeze and Excitation method [28].

Description	Top-1 Error(%)	Top-5 Error(%)
ResNet50 + channel (SE [28])	23.14	6.70
ResNet50 + channel + spatial	**22.66**	**6.31**
ResNet50 + spatial + channel	22.78	6.42
ResNet50 + channel & spatial in parallel	22.95	6.59

Table 3: **Combining methods of channel and spatial attention**. Using both attention is critical while the best-combining strategy (*i.e.* sequential, channel-first) further improves the accuracy.

CBAM

Architecture	Param.	GFLOPs	Top-1 Error (%)	Top-5 Error (%)
ResNet18 [5]	11.69M	1.814	29.60	10.55
ResNet18 [5] + SE [28]	11.78M	1.814	29.41	10.22
ResNet18 [5] + CBAM	11.78M	1.815	**29.27**	**10.09**
ResNet34 [5]	21.80M	3.664	26.69	8.60
ResNet34 [5] + SE [28]	21.96M	3.664	26.13	8.35
ResNet34 [5] + CBAM	21.96M	3.665	**25.99**	**8.24**
ResNet50 [5]	25.56M	3.858	24.56	7.50
ResNet50 [5] + SE [28]	28.09M	3.860	23.14	6.70
ResNet50 [5] + CBAM	28.09M	3.864	**22.66**	**6.31**
ResNet101 [5]	44.55M	7.570	23.38	6.88
ResNet101 [5] + SE [28]	49.33M	7.575	22.35	6.19
ResNet101 [5] + CBAM	49.33M	7.581	**21.51**	**5.69**
WideResNet18 [6] (widen=1.5)	25.88M	3.866	26.85	8.88
WideResNet18 [6] (widen=1.5) + SE [28]	26.07M	3.867	26.21	8.47
WideResNet18 [6] (widen=1.5) + CBAM	26.08M	3.868	**26.10**	**8.43**
WideResNet18 [6] (widen=2.0)	45.62M	6.696	25.63	8.20
WideResNet18 [6] (widen=2.0) + SE [28]	45.97M	6.696	24.93	7.65
WideResNet18 [6] (widen=2.0) + CBAM	45.97M	6.697	**24.84**	**7.63**
ResNeXt50 [7] (32x4d)	25.03M	3.768	22.85	6.48
ResNeXt50 [7] (32x4d) + SE [28]	27.56M	3.771	**21.91**	6.04
ResNeXt50 [7] (32x4d) + CBAM	27.56M	3.774	21.92	**5.91**
ResNeXt101 [7] (32x4d)	44.18M	7.508	21.54	5.75
ResNeXt101 [7] (32x4d) + SE [28]	48.96M	7.512	21.17	5.66
ResNeXt101 [7] (32x4d) + CBAM	48.96M	7.519	**21.07**	**5.59**

https://github.com/kobiso/CBAM-keras/blob/master/models/attention_module.py

بار درخت علم ندانم مگر عمل

با علم اگر عمل نکنی شاخ بی‌بری

(سعدی)