



清华大学
Tsinghua University

Advanced Computer Vision
THU×SENSETIME – 80231202



Chapter 2 - Section 6

Image Classification

Dr. Li Hongyang

Friday, March 26, 2021

Acknowledge : Wang Cheng



Outline

- Part 1** **Introduction to Image Classification**

- Part 2** **Supervised Image Classification**

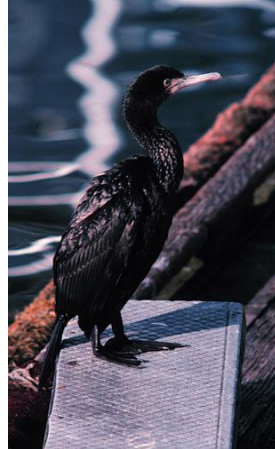
- Part 3** **Semi-Unsupervised Classification Model**

- Part 4** **Further Research Topics**

- Part 5** **Recommended Competitions and Repos**

- Definition

- Attempts to comprehend an entire image as a whole.
- Classify the image by assigning it to a specific label.



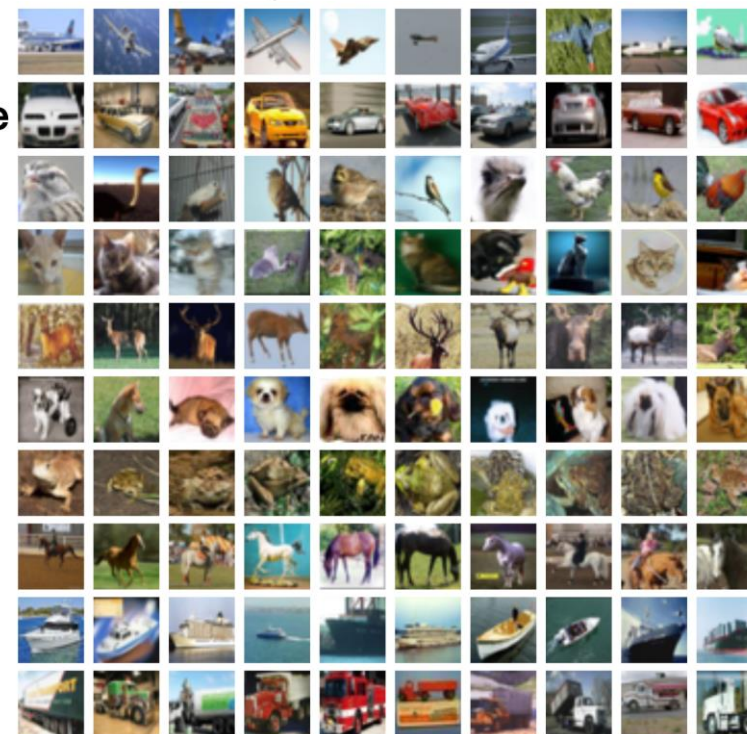
Bird
Car
Dog
Cat
Human
Flower

- Different types of image classification tasks



The MNIST database of handwritten digits

- airplane
- automobile
- bird
- cat
- deer
- dog
- frog
- horse
- ship
- truck



Ten classes in CIFAR-10

LeCun Y. The MNIST database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>. 1998.
Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images.

- Different types of image classification tasks



Traffic sign recognition



Flower recognition

- Challenges
 - Viewpoint variation
 - Illumination
 - Deformation
 - Occlusion
 - Background Clutter
 - Intraclass variation

- Viewpoint variation
 - Images are selected from CUB-100-2011



Brandt Cormorant (Left)



Brandt Cormorant (Right)

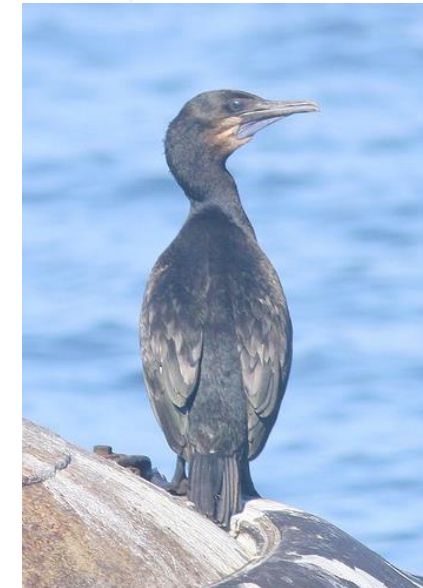
- Illumination



Brandt Cormorant



Brandt Cormorant (Light)



Brandt Cormorant (Lighter)

- Deformation



Brandt Cormorant



Brandt Cormorant (Open)

- Occlusion



Brandt Cormorant



Brandt Cormorant (Occlusion)

- Background Clutter



Brandt Cormorant



Brandt Cormorant (Background Clutter)

- Intra-class variation



Brandt Cormorant



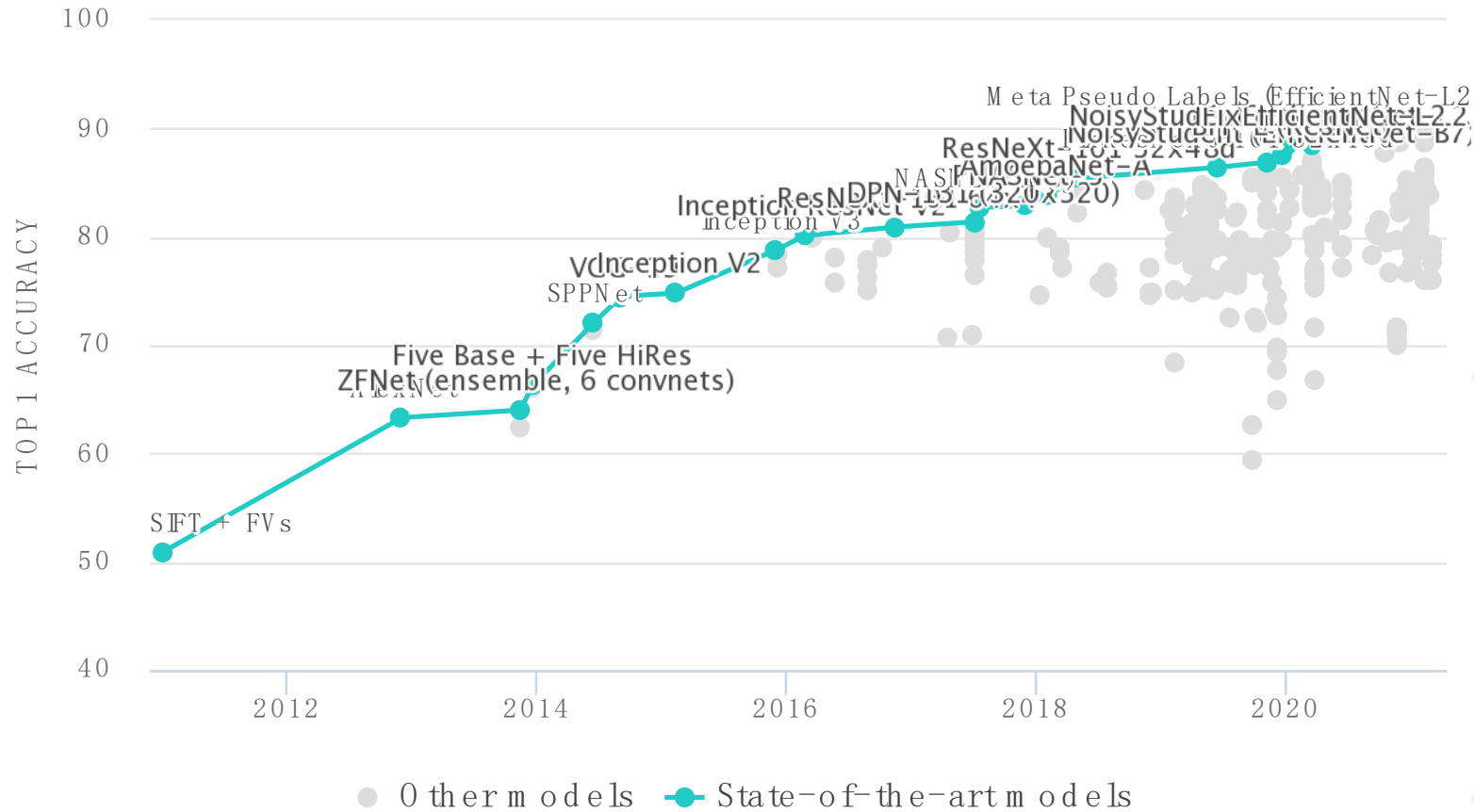
Yellow Throat



Black and White Warbler

- Progress
 - K-Nearest Neighbor
 - Naive Bayes
 - Support Vector Machines
 - Neural network

- Overview of Models in ImageNet Challenge



J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database. CVPR2009.

- ImageNet Datasets
 - The ImageNet dataset contains 14,197,122 annotated images according to the WordNet hierarchy
 - Total number of non-empty WordNet synsets: 21841
 - Total number of images: 14,197,122
 - Number of images with bounding box annotations: 1,034,908
 - Number of synsets with SIFT features: 1000
 - Number of images with SIFT features: 1.2 million

- Several datasets and evaluation metrics

Topic	Dataset Name	Task	Scale
Objects	Cifar-10	Multi-class Classification	60,000 images, 10 classes
	Cifar-100	Fine-grained Classification	60,000 images, 100 classes
	ImageNet	Multi-class Classification	14,000,000 images, 20,000 classes
	PASCAL VOC	Multi-class Classification	17,125 images, 20 classes
Human	SMILES	Binary Classification	13,165 images, 2 classes
	Facial Expression Recognition	Multi-class Classification	35,888, 7 classes
Flowers	Flowers-17	Multi-class Classification	1,360, 17 classes
	Flowers-102	Fine-grained Classification	8,189, 102 classes
Text	MNIST	Multi-class Classification	70,000 images, 10 classes
	USPS	Multi-class Classification	9,900 images, 10 classes

- Several datasets and evaluation metrics
 - Accuracy
 - F1 score
 - Precision-Recall (PR) curves
 - Receiver Operator Characteristic (ROC) curves, AUC

- F1 score
 - Both consider *precision* and *recall*

$$\textit{precision} = \frac{TP}{TP + FP}$$

$$\textit{recall} = \frac{TP}{TP + FN}$$

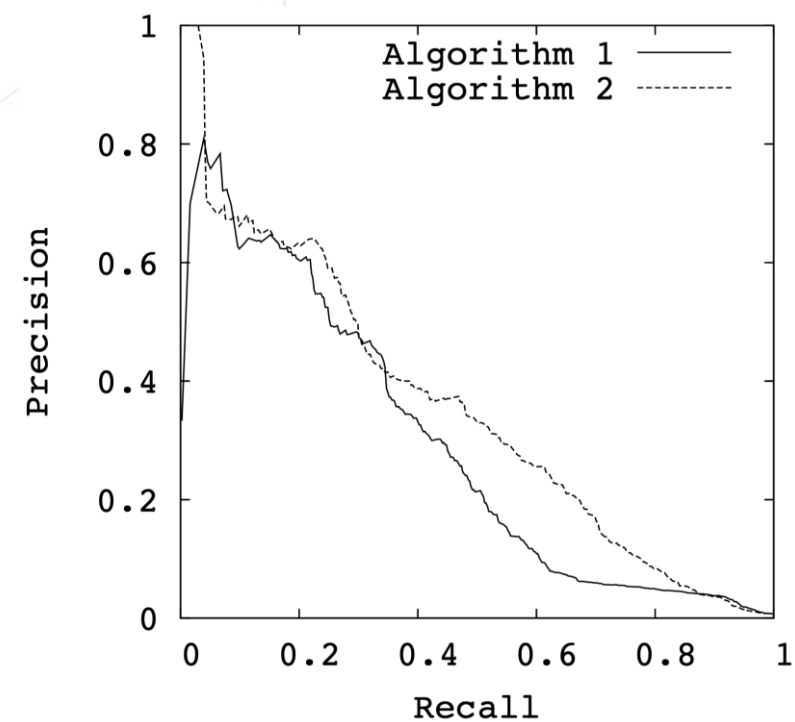
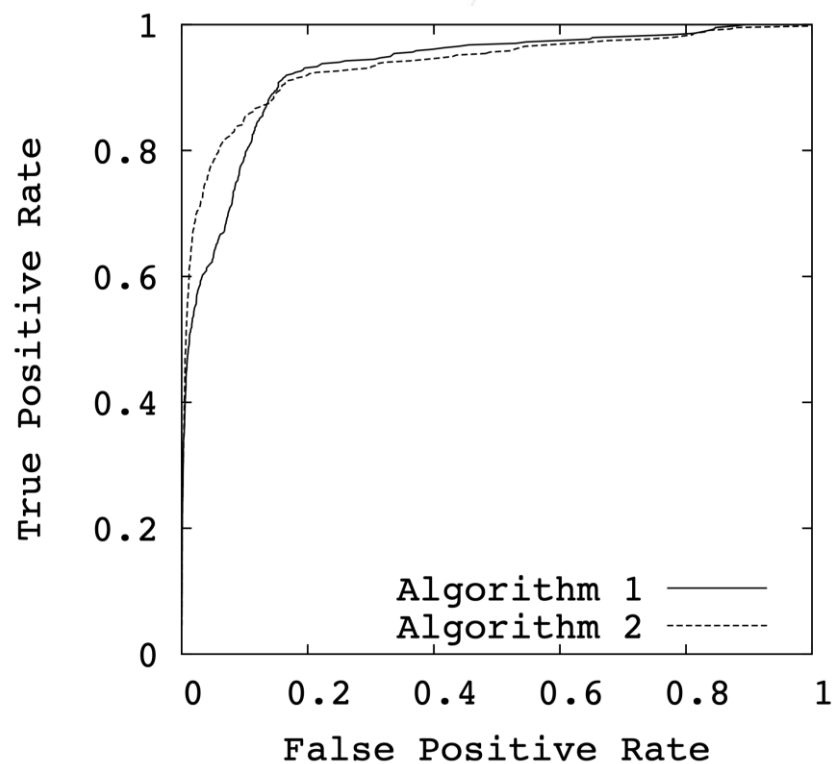
$$F_1 = \frac{2 \times \textit{precision} \times \textit{recall}}{\textit{precision} + \textit{recall}}$$

$$F_\beta = (1 + \beta^2) \frac{\textit{precision} \times \textit{recall}}{(\beta^2 \times \textit{precision}) + \textit{recall}}$$

$$\textit{accuracy} = \frac{TP + TN}{TP + FN + TN + FP}$$

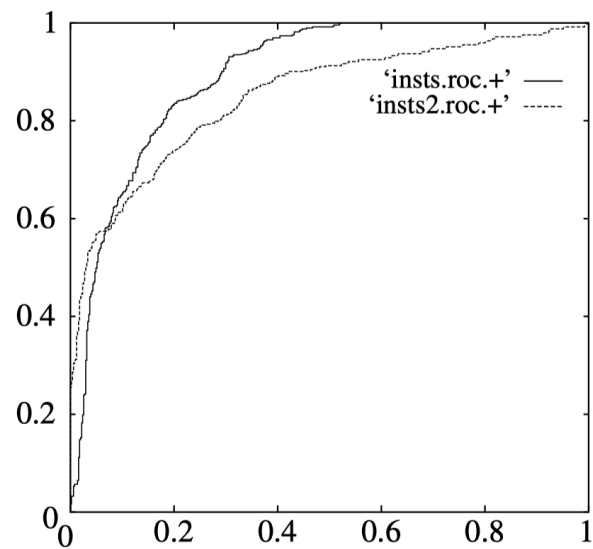
$$\textit{specificity} = \frac{TN}{TN + FP}$$

- PR vs ROC
 - Which algorithm is better?

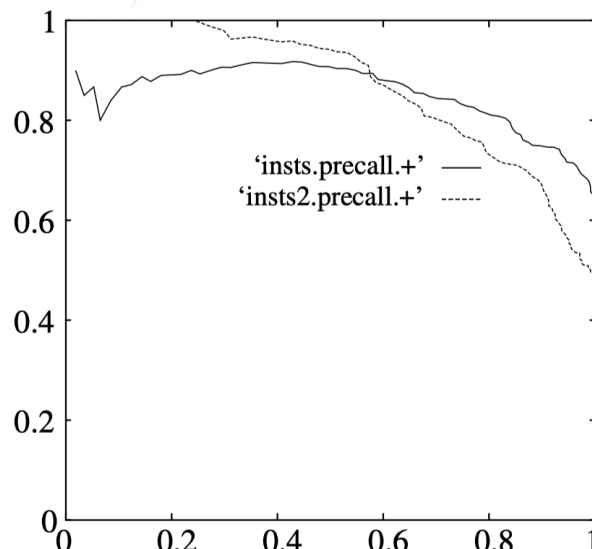


Davis, J. and Goadrich, M., The relationship between Precision-Recall and ROC curves. ICML, 2006.

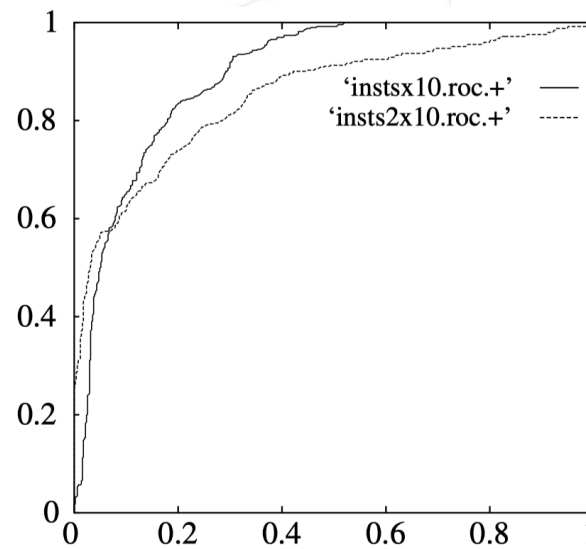
- PR vs ROC
 - Which algorithm is better?



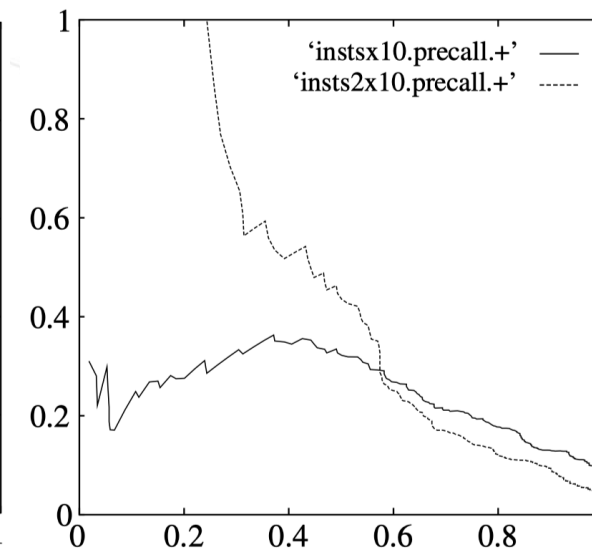
(a)



(b)



(c)



(d)

Fawcett T. An introduction to ROC analysis. Pattern recognition letters. 2006 Jun 1;27(8):861-74.

- PR vs ROC
 - A curve dominates in ROC space if and only if it dominates in PR space
 - The ROC curve and PR curve for a given algorithm contain the same points
 - Optimizing the area under the ROC curve is not guaranteed to optimize the area under the PR curve



Outline

- Part 1 Introduction to Image Classification

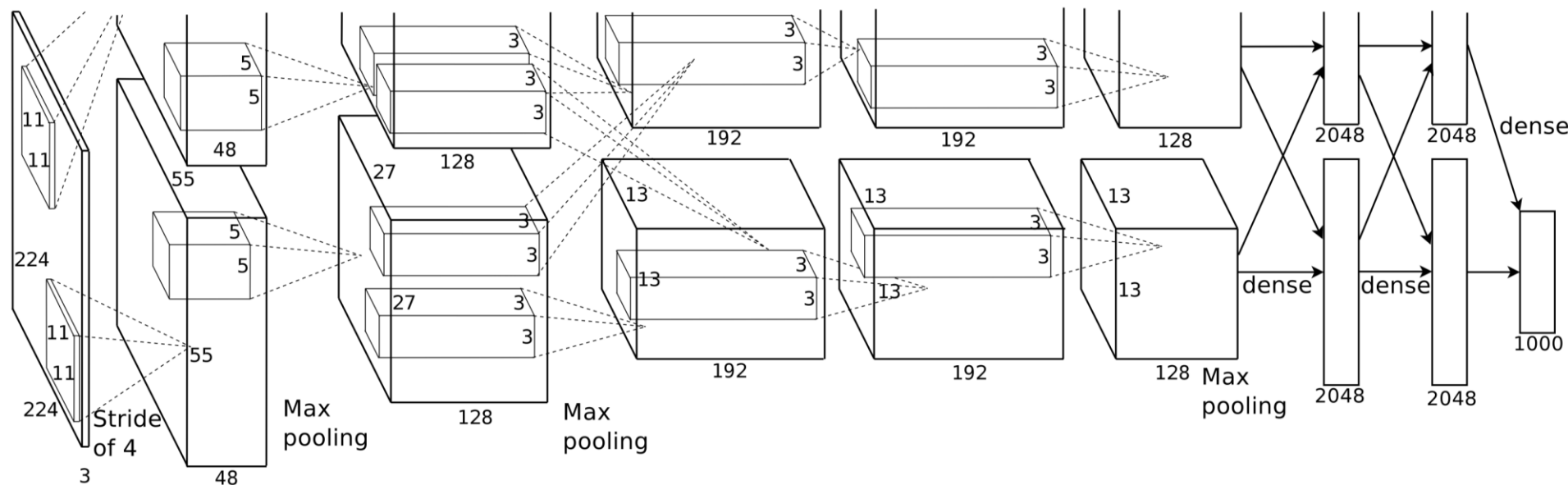
- Part 2 Supervised Image Classification**

- Part 3 Semi-Unsupervised Classification Model

- Part 4 Further Research Topics

- Part 5 Recommended Competitions and Repos

- 2012: AlexNet
 - Top-1 accuracy rate: 62.5%
 - Parameters: 60M
 - Takes between five and six days to train on two GTX 580 3GB GPUs



Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. NIPS. 2012;25:1097-105.

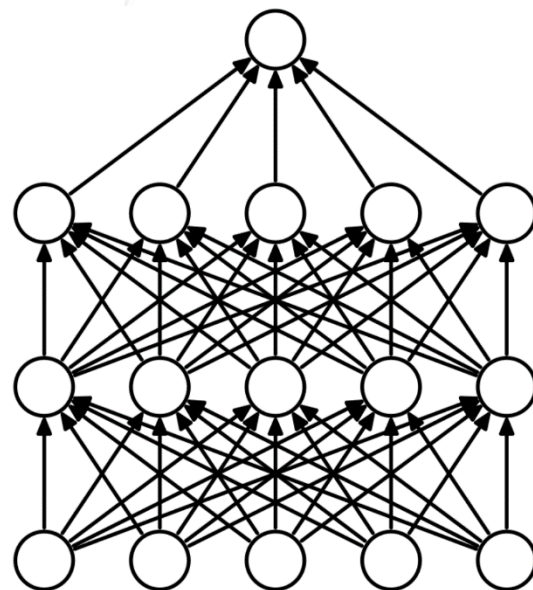
- Rectified linear unit (ReLU)

$$f(x) = x^+ = \max(0, x)$$

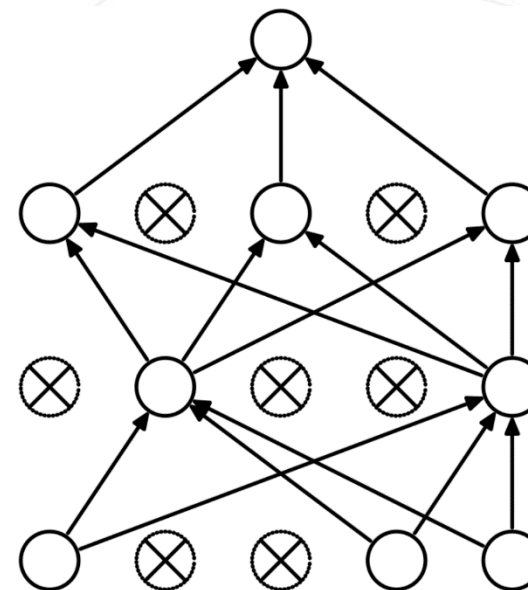
- Biological plausibility: One-sided
- Sparse activation: In a randomly initialized network, only about 50% of hidden units are activated
- Better gradient propagation: Fewer vanishing gradient problems
- Efficient computation: Only comparison, addition and multiplication.
- Scale-invariant: $\max(0, ax) = a \max(0, x)$ for $a \geq 0$

- Dropout in Neural Networks

- Training neural network with dropout can be seen as training a collection of 2^n thinned networks with extensive weight sharing



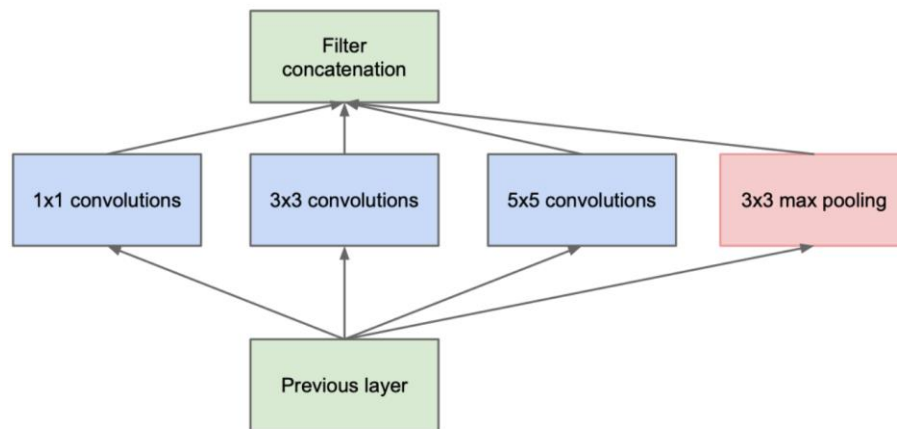
(a) Standard Neural Net



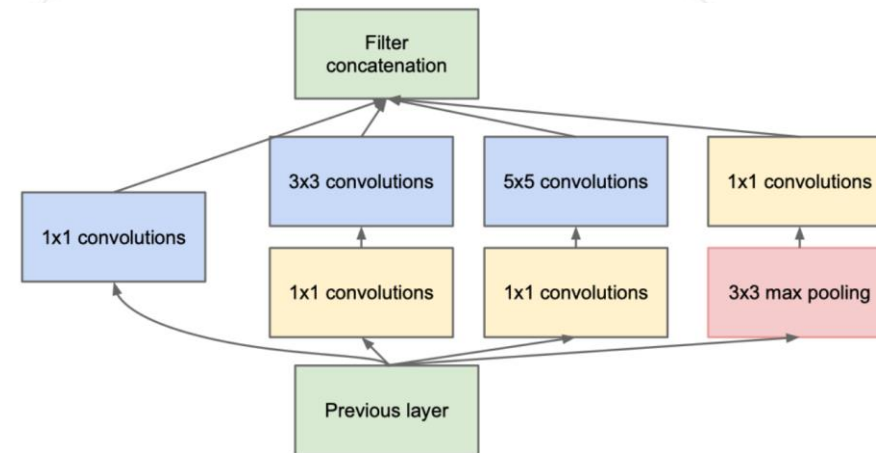
(b) After applying dropout.

Srivastava N, Hinton G, et al. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research. 2014 Jan 1;15(1):1929-58.

- 2014: Inception V1
 - Top-1 accuracy rate: 69.8%
 - Parameters: 5M
 - Introduce a new level of organization in the form of the “Inception module”



(a) Inception module, naïve version



(b) Inception module with dimension reductions

- Inception module and GoogLeNet
 - An optimal local sparse structure can be approximated by dense components
 - 1×1 convolutions are used to compute reductions
 - $2 - 3 \times$ faster than similarly performing networks with non-Inception architecture
 - Auxiliary classifiers
 - Encourage discrimination in the lower stages in the classifier
 - Increase the gradient signal that gets propagated back
 - Provide additional regularization.

- 2015: Inception V3
 - Top-1 accuracy rate: 78.8%
 - Parameters: 23.8M
 - Factorizing Convolutions with Large Filter Size
 - $5*5 \rightarrow 3*3 + 3*3, 3*3 \rightarrow 1*3 + 3*1$
 - Model Regularization via Label Smoothing

$$y_i = \begin{cases} 1 - \varepsilon & \text{if } i = \text{true}, \\ \frac{\varepsilon}{K - 1} & \text{otherwise} \end{cases}$$

- 2015: ResNet 152
 - Top-1 accuracy rate: 78.6%
 - Parameters: 117.4M

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. CVPR2016 (pp. 770-778).

- Is learning better networks as easy as stacking more layers?
 - Vanishing/exploding gradients
 - Batch Normalization; ReLU
 - Overfitting
 - More data; Dropout
 - High resource consumption
 - More GPUs
 - Higher training error, degradation problem
 - Why, and how to solve it?

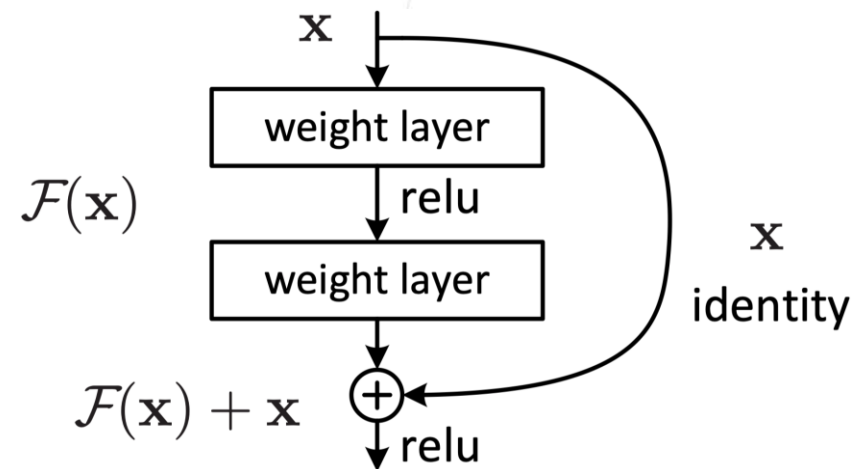
- Deep residual learning framework

- Data Processing Inequality

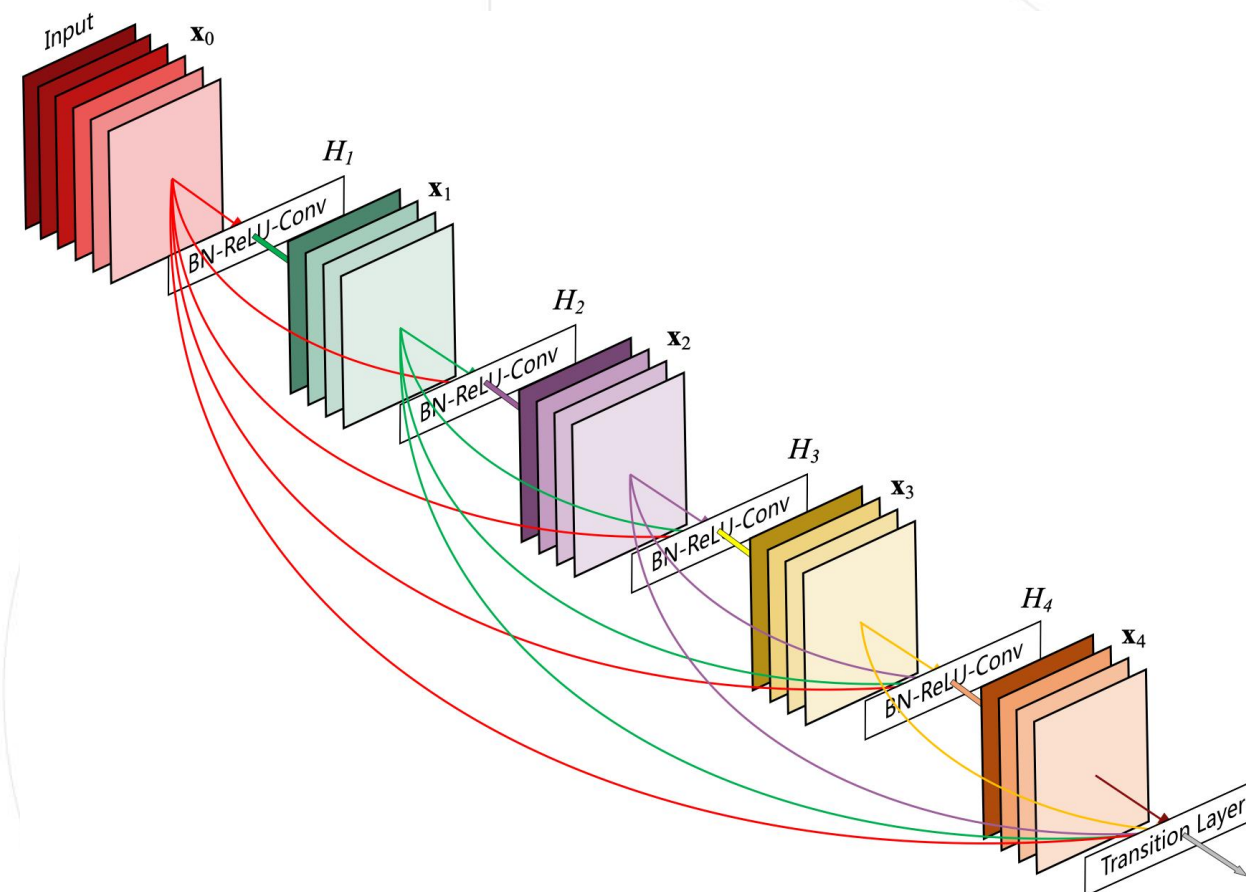
$$I(X; Y) \geq I(X; Z)$$

- Shortcut connections

- a deeper model should have training error no greater than its shallower counterpart



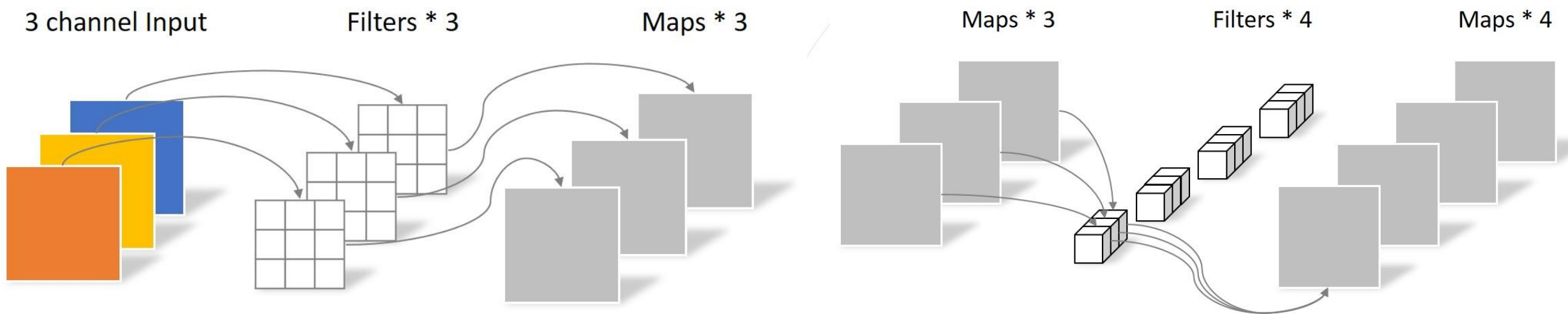
- 2016: DenseNet-264
 - Top-1 accuracy rate: 79.2%
 - Create short paths from early layers to later layers
 - Each layer has direct access to the gradients from the loss function and the original input signal



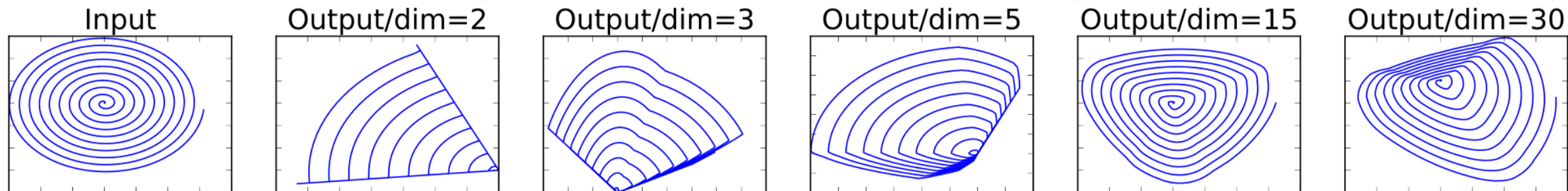
- 2018: MobileNetV2
 - Top-1 accuracy rate: 74.7%
 - Parameters: 6.9M
 - Depthwise separable convolution
 - Linear bottleneck
 - Inverted residuals

Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. Mobilenetv2: Inverted residuals and linear bottlenecks. CVPR2018 (pp. 4510-4520).

- Depthwise Separable Convolution
 - Depthwise Convolution (left) + Pointwise Convolution (right)

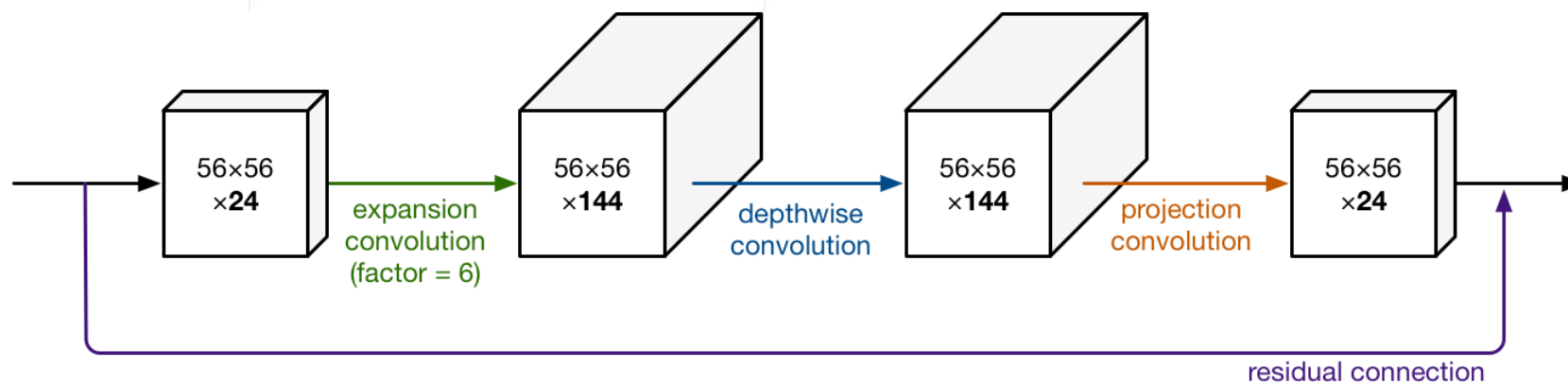


- Linear bottleneck
 - ReLU collapses the channel

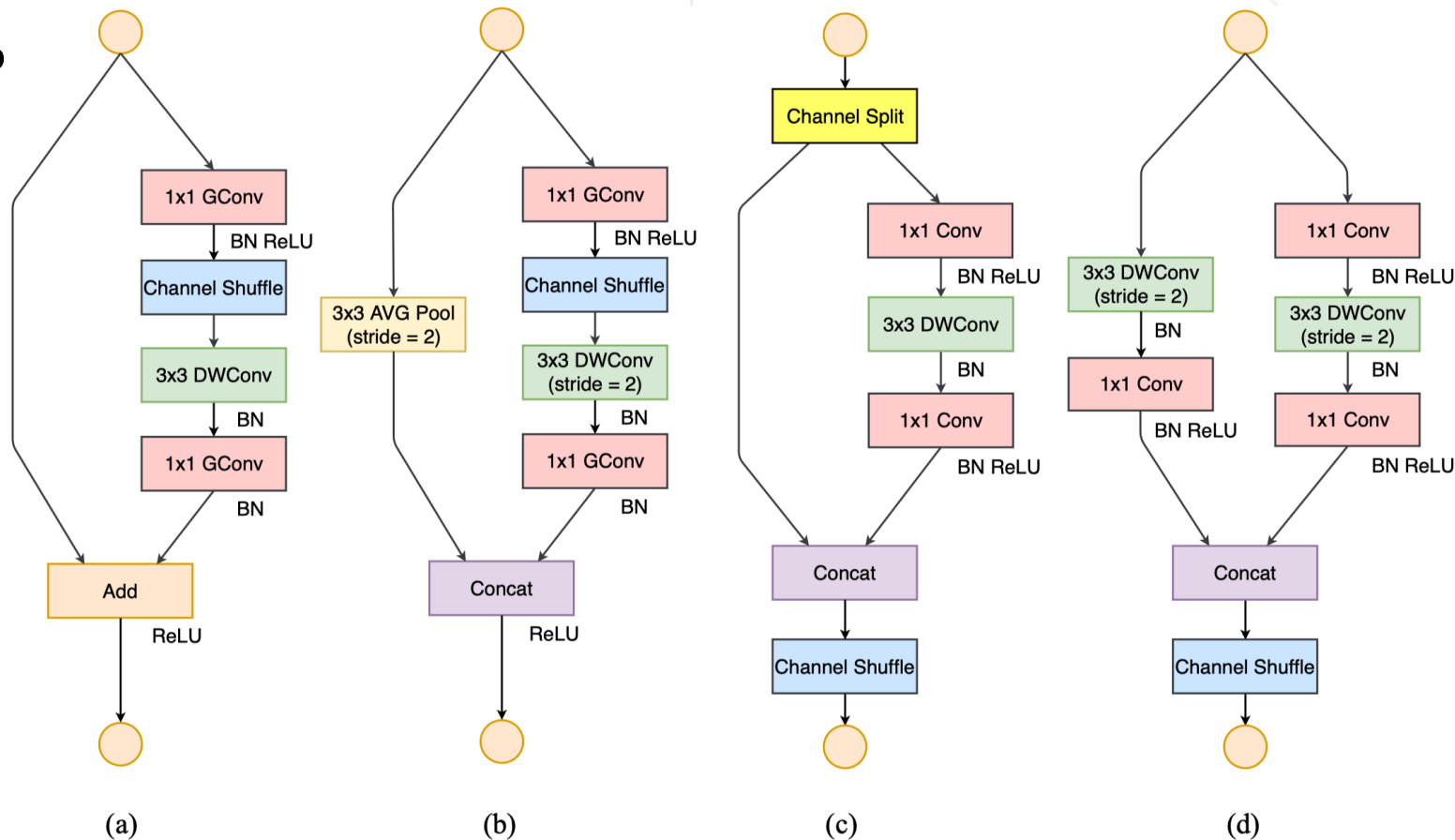


- Using linear layers prevents nonlinearities from destroying too much information
- Expansion ratio
 - The ratio between the size of the input bottleneck and the inner size

- Inverted residuals
 - Expansion Convolution
 - low-dim to high-dim
 - Depthwise Convolution with ReLU
 - Projection layer
 - high-dim to low-dim



- 2018: ShuffleNet V2 2x
 - Top-1 accuracy rate:75.4%
 - Parameters:7.4M
 - Channel Split

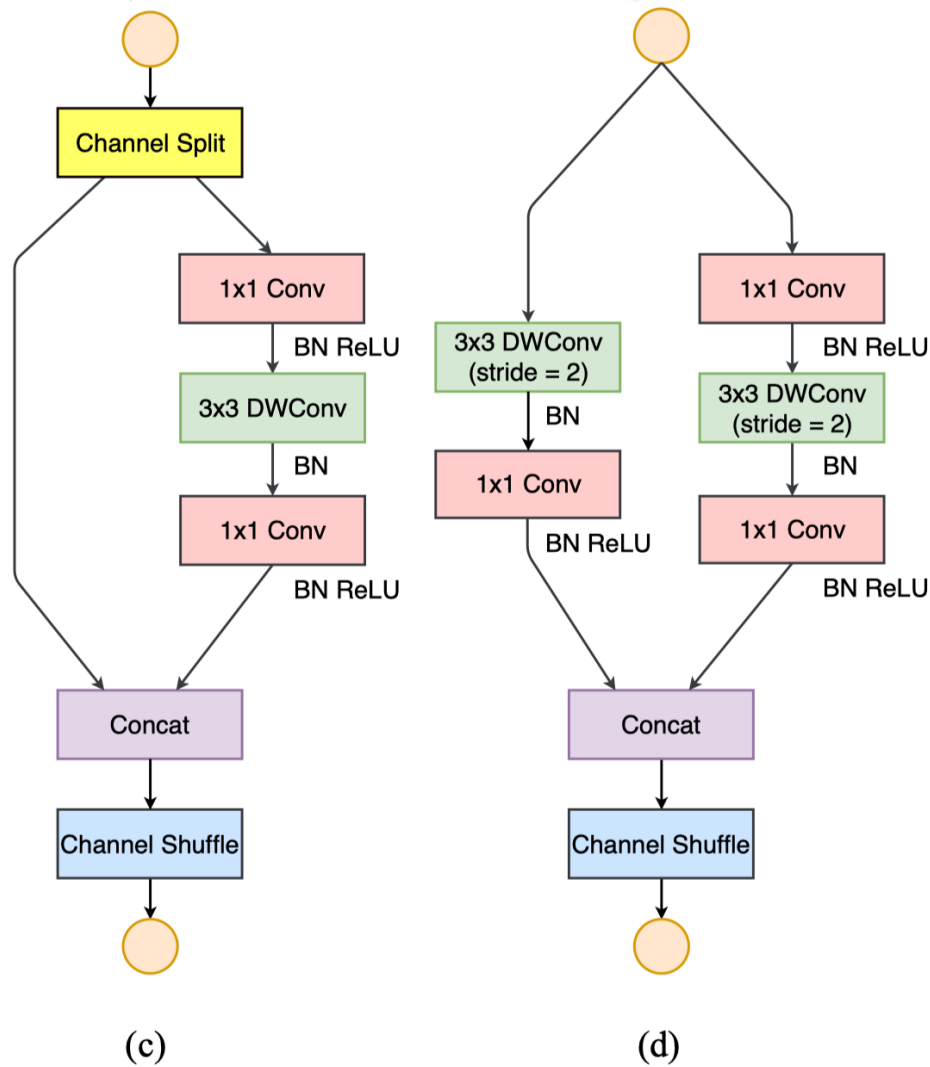


Ma N, Zhang X, Zheng HT, Sun J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. ECCV2018.

- Four practical guidelines for efficient network architecture design
 - G1) Equal channel width minimizes memory access cost (MAC)
 - G2) Excessive group convolution increases MAC
 - G3) Network fragmentation reduces degree of parallelism
 - G4) Element-wise operations are non-negligible

- ShuffleNet V2

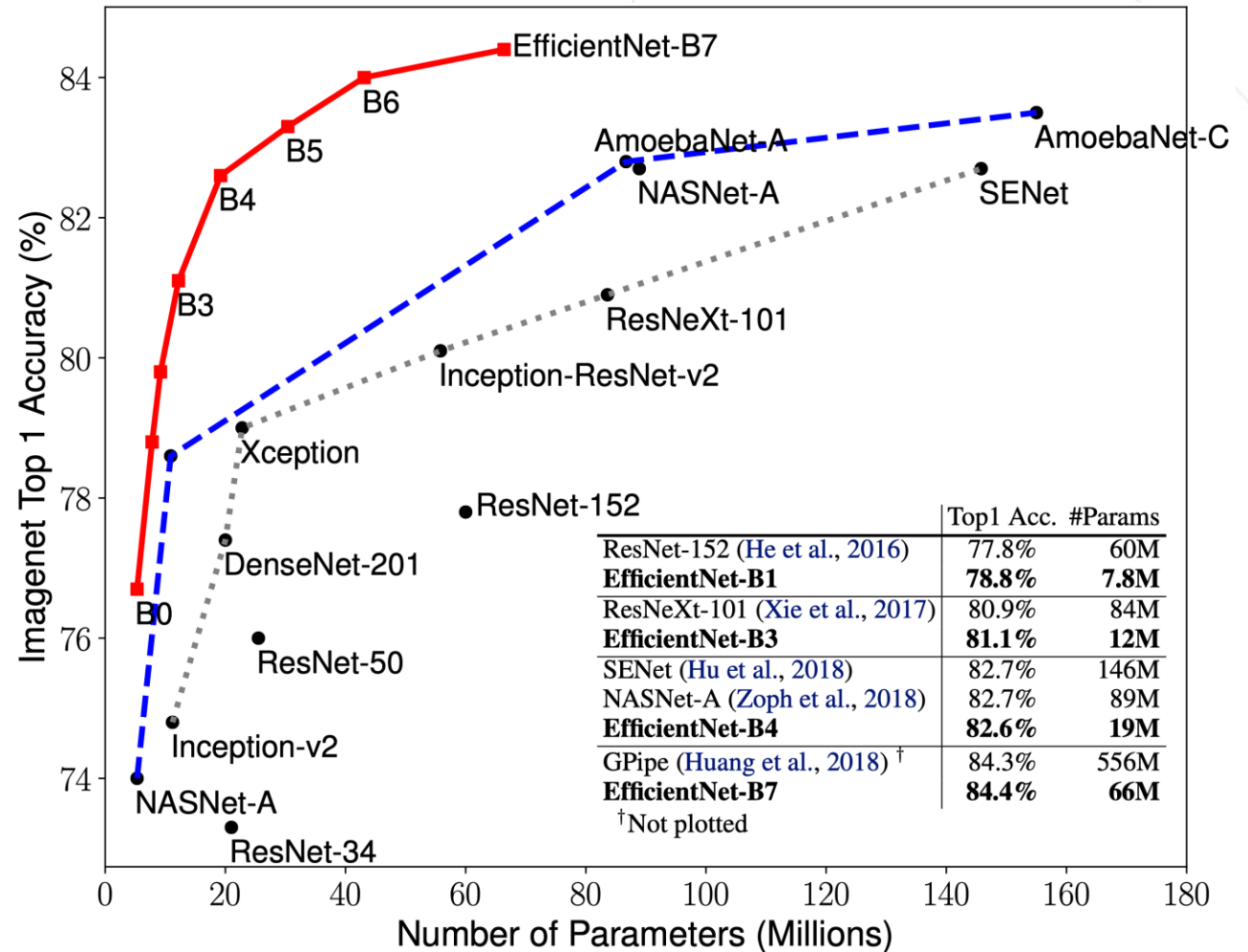
- One branch remains as identity (G3)
- No group convolution (G2)
- The number of channels keeps the same (G1)
- “Concat” , “Channel Shuffle” and “Channel Split” , are merged into a single element-wise operation (G4)



- Some NAS models

ConvNet	ImageNet top1 acc	ImageNet top5 acc	Published In
NASNet	82.7	96.2	CVPR2018
PNASNet	82.9	96.2	ECCV2018
MNASNet	76.13	92.85	CVPR2018
ProxylessNAS	75.1	92.5	ICLR2019
EfficientNet	84.3	97.0	ICML2019

- 2019: EfficientNet
 - Top-1 accuracy rate: 84.4%
 - Parameters: 66M
 - Compound scaling method



Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks. ICML2019



Outline

- Part 1 Introduction to Image Classification

- Part 2 Supervised Image Classification

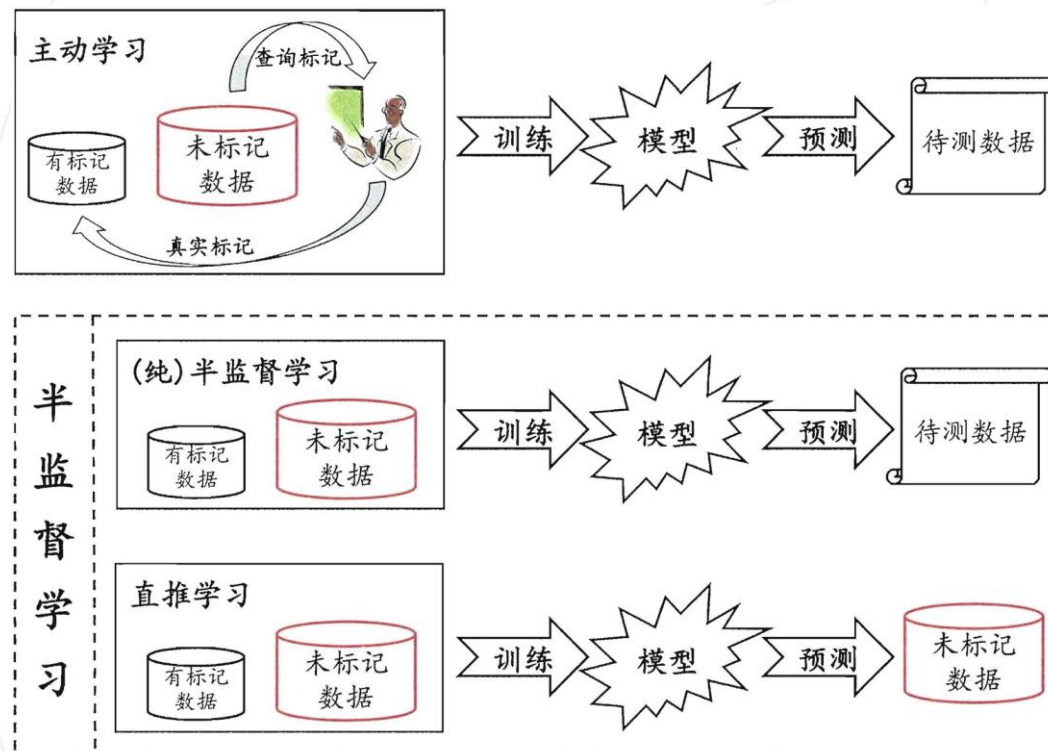
- Part 3 Semi-Unsupervised Classification Model**

- Part 4 Further Research Topics

- Part 5 Recommended Competitions and Repos

- Definition

- Semi-supervised image classification leverages unlabeled data as well as labelled data to increase classification performance



- Background
 - Consistency Regularization

$$\|p_{\text{model}}(y | \text{Augment}(x); \theta) - p_{\text{model}}(y | \text{Augment}(x); \theta)\|_2^2$$

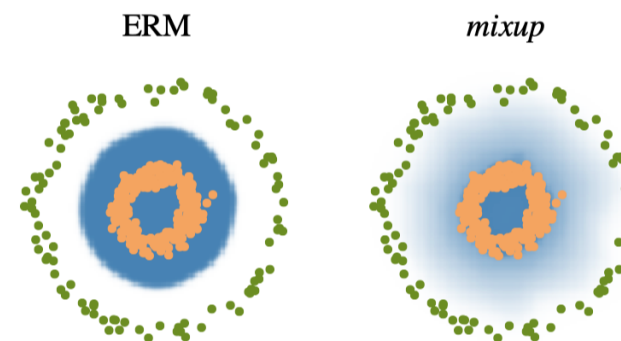
- Entropy Minimization
 - Minimizes the entropy of $p_{\text{model}}(y|x; \theta)$
- Traditional Regularization

- mixup: Beyond Empirical Risk Minimization
 - A simple and data-agnostic data augmentation method

$$\tilde{x} = \lambda x_i + (1 - \lambda)x_j, \quad \text{where } x_i, x_j \text{ are raw input vectors}$$
$$\tilde{y} = \lambda y_i + (1 - \lambda)y_j, \quad \text{where } y_i, y_j \text{ are one-hot label encodings}$$

```
# y1, y2 should be one-hot vectors
for (x1, y1), (x2, y2) in zip(loader1, loader2):
    lam = numpy.random.beta(alpha, alpha)
    x = Variable(lam * x1 + (1. - lam) * x2)
    y = Variable(lam * y1 + (1. - lam) * y2)
    optimizer.zero_grad()
    loss(net(x), y).backward()
    optimizer.step()
```

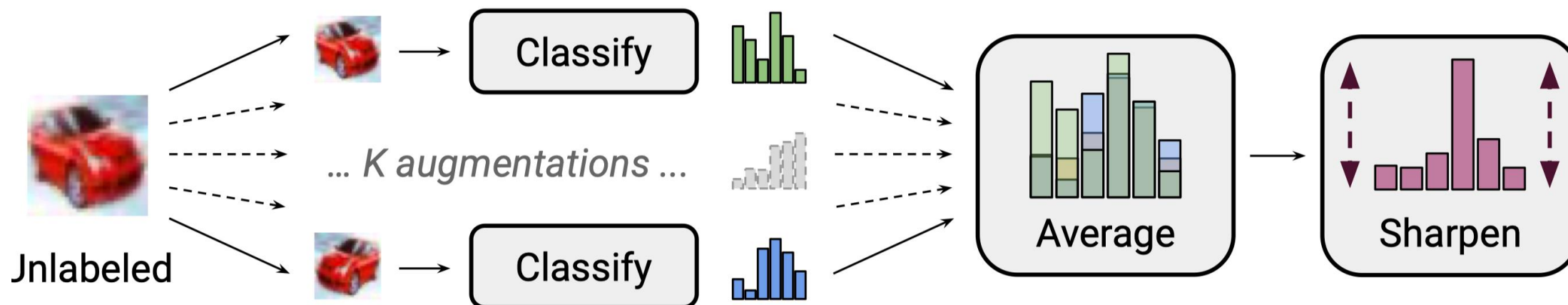
(a) One epoch of *mixup* training in PyTorch.



(b) Effect of *mixup* ($\alpha = 1$) on a toy problem. Green: Class 0. Orange: Class 1. Blue shading indicates $p(y = 1|x)$.

- mixup: Beyond Empirical Risk Minimization
 - Why use beta distribution?
 - Similar to label smooth?
 - Why mixup works? (Generalization gap between training data and real data)

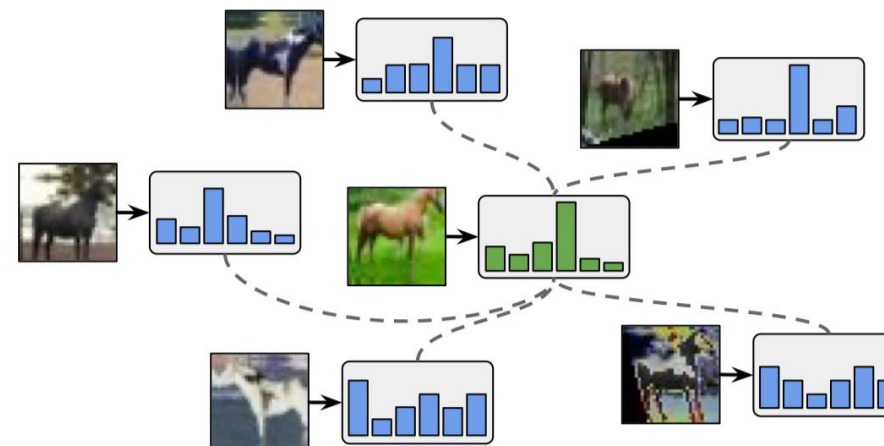
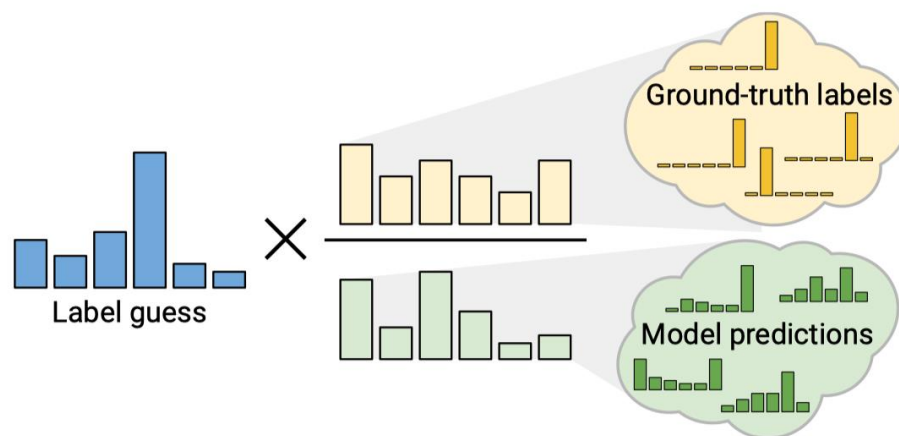
- MixMatch: A Holistic Approach to Semi-Supervised Learning
 - Stochastic data augmentation is applied to an unlabeled image K times
 - The average of these K predictions is “sharpened” by adjusting the distribution’s temperature



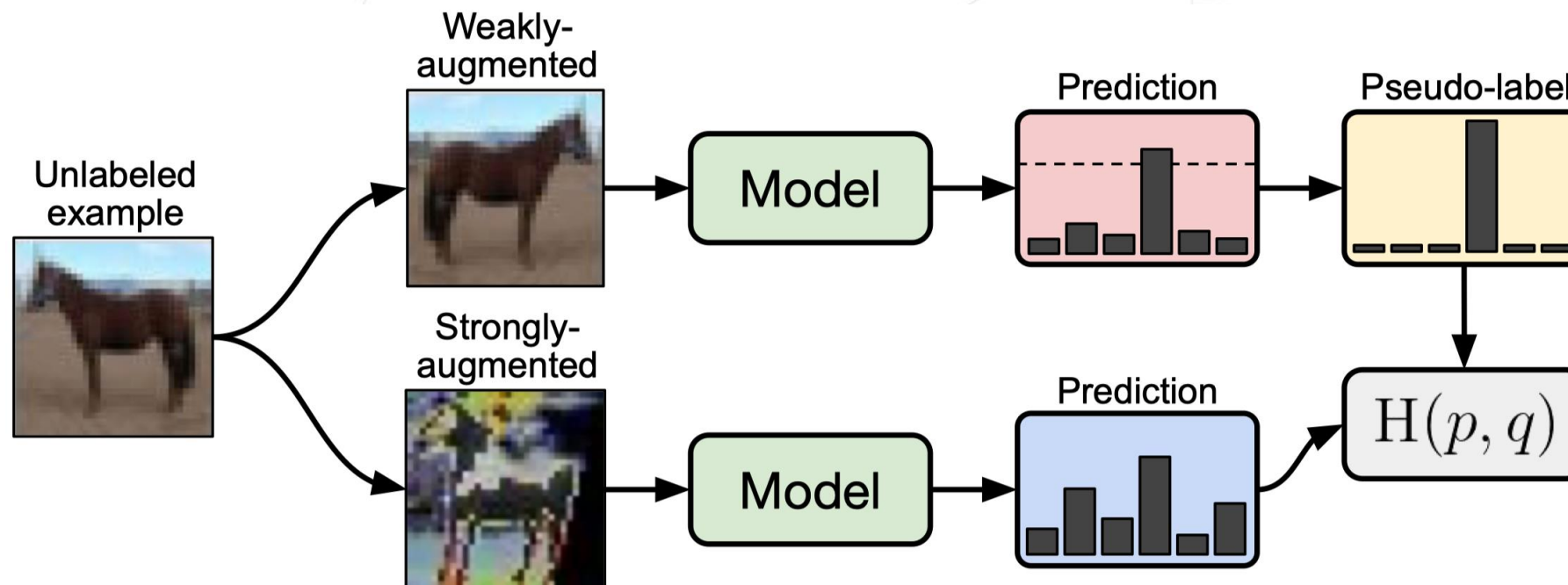
• MixMatch: A Holistic Approach to Semi-Supervised Learning

- 1: **Input:** Batch of labeled examples and their one-hot labels $\mathcal{X} = ((x_b, p_b); b \in (1, \dots, B))$, batch of unlabeled examples $\mathcal{U} = (u_b; b \in (1, \dots, B))$, sharpening temperature T , number of augmentations K , Beta distribution parameter α for MixUp.
- 2: **for** $b = 1$ **to** B **do**
- 3: $\hat{x}_b = \text{Augment}(x_b)$ *// Apply data augmentation to x_b*
- 4: **for** $k = 1$ **to** K **do**
- 5: $\hat{u}_{b,k} = \text{Augment}(u_b)$ *// Apply k^{th} round of data augmentation to u_b*
- 6: **end for**
- 7: $\bar{q}_b = \frac{1}{K} \sum_k P_{\text{model}}(y | \hat{u}_{b,k}; \theta)$ *// Compute average predictions across all augmentations of u_b*
- 8: $q_b = \text{Sharpen}(\bar{q}_b, T)$ *// Apply temperature sharpening to the average prediction (see eq. (7))*
- 9: **end for**
- 10: $\hat{\mathcal{X}} = ((\hat{x}_b, p_b); b \in (1, \dots, B))$ *// Augmented labeled examples and their labels*
- 11: $\hat{\mathcal{U}} = ((\hat{u}_{b,k}, q_b); b \in (1, \dots, B), k \in (1, \dots, K))$ *// Augmented unlabeled examples, guessed labels*
- 12: $\mathcal{W} = \text{Shuffle}(\text{Concat}(\hat{\mathcal{X}}, \hat{\mathcal{U}}))$ *// Combine and shuffle labeled and unlabeled data*
- 13: $\mathcal{X}' = (\text{MixUp}(\hat{\mathcal{X}}_i, \mathcal{W}_i); i \in (1, \dots, |\hat{\mathcal{X}}|))$ *// Apply MixUp to labeled data and entries from \mathcal{W}*
- 14: $\mathcal{U}' = (\text{MixUp}(\hat{\mathcal{U}}_i, \mathcal{W}_{i+|\hat{\mathcal{X}}|}); i \in (1, \dots, |\hat{\mathcal{U}}|))$ *// Apply MixUp to unlabeled data and the rest of \mathcal{W}*
- 15: **return** $\mathcal{X}', \mathcal{U}'$

- ReMixMatch: Semi-Supervised Learning with Distribution Matching and Augmentation Anchoring
 - Improved version of MixMatch
 - Distribution Alignment (left) and Augmentation Anchor (right)



- FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence
 - Combination of Consistency regularization and pseudo-labeling.



Sohn K, Berthelot D, et al. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. arXiv preprint arXiv:2001.07685. 2020 Jan 21.

- FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence
 - FixMatch consists of two loss terms: a supervised loss ℓ_s and an unsupervised loss ℓ_u
 - ℓ_s is the standard cross-entropy loss

$$\ell_s = \frac{1}{B} \sum_{b=1}^B \text{H}(p_b, p_m(y | \alpha(x_b)))$$

- Convert the prediction on the weakly-augmented image to a one-hot pseudo-label
- ℓ_u is the cross-entropy loss against the model' s output for the strongly-augmented image

$$\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}(\max(q_b) \geq \tau) \text{H}(\hat{q}_b, p_m(y | \mathcal{A}(u_b)))$$

- 生成式方法 (generative methods)
 - 假设所有数据(无论是否有标记)都是由同一个潜在的模型 “生成” 的
 - 通过潜在模型的参数将未标记数据与学习目标联系起来, 而未标记数据的标记则可看作模型的缺失参数
 - 通常可基于 EM 算法进行极大似然估计求解

• 生成式方法 (generative methods)

- E 步: 根据当前模型参数计算未标记样本 \mathbf{x}_j 属于各高斯混合成分的概率

$$\gamma_{ji} = \frac{\alpha_i \cdot p(\mathbf{x}_j | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{i=1}^N \alpha_i \cdot p(\mathbf{x}_j | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)} ; \quad (13.5)$$

- M 步: 基于 γ_{ji} 更新模型参数, 其中 l_i 表示第 i 类的有标记样本数目

$$\boldsymbol{\mu}_i = \frac{1}{\sum_{\mathbf{x}_j \in D_u} \gamma_{ji} + l_i} \left(\sum_{\mathbf{x}_j \in D_u} \gamma_{ji} \mathbf{x}_j + \sum_{(\mathbf{x}_j, y_j) \in D_l \wedge y_j = i} \mathbf{x}_j \right) , \quad (13.6)$$

$$\boldsymbol{\Sigma}_i = \frac{1}{\sum_{\mathbf{x}_j \in D_u} \gamma_{ji} + l_i} \left(\sum_{\mathbf{x}_j \in D_u} \gamma_{ji} (\mathbf{x}_j - \boldsymbol{\mu}_i)(\mathbf{x}_j - \boldsymbol{\mu}_i)^T + \sum_{(\mathbf{x}_j, y_j) \in D_l \wedge y_j = i} (\mathbf{x}_j - \boldsymbol{\mu}_i)(\mathbf{x}_j - \boldsymbol{\mu}_i)^T \right) , \quad (13.7)$$

$$\alpha_i = \frac{1}{m} \left(\sum_{\mathbf{x}_j \in D_u} \gamma_{ji} + l_i \right) . \quad (13.8)$$



Outline

- Part 1 Introduction to Image Classification

- Part 2 Supervised Image Classification

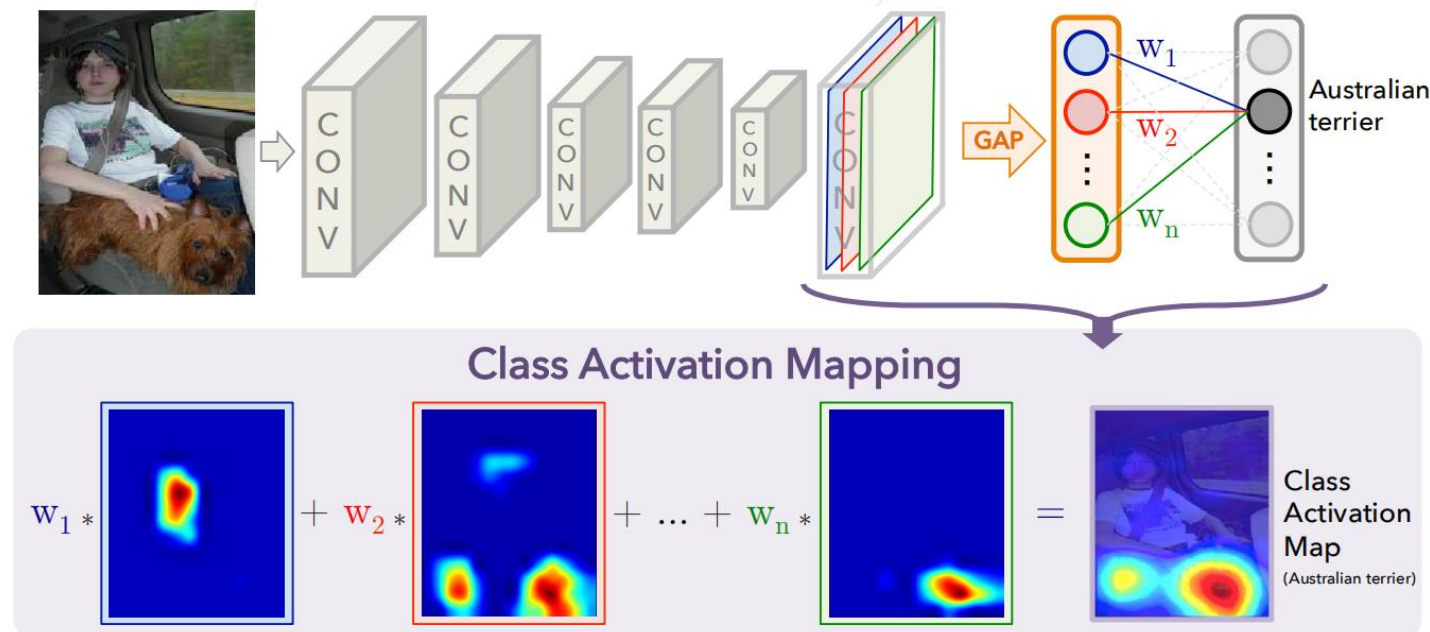
- Part 3 Semi-Unsupervised Classification Model

- Part 4 Further Research Topics**

- Part 5 Recommended Competitions and Repos

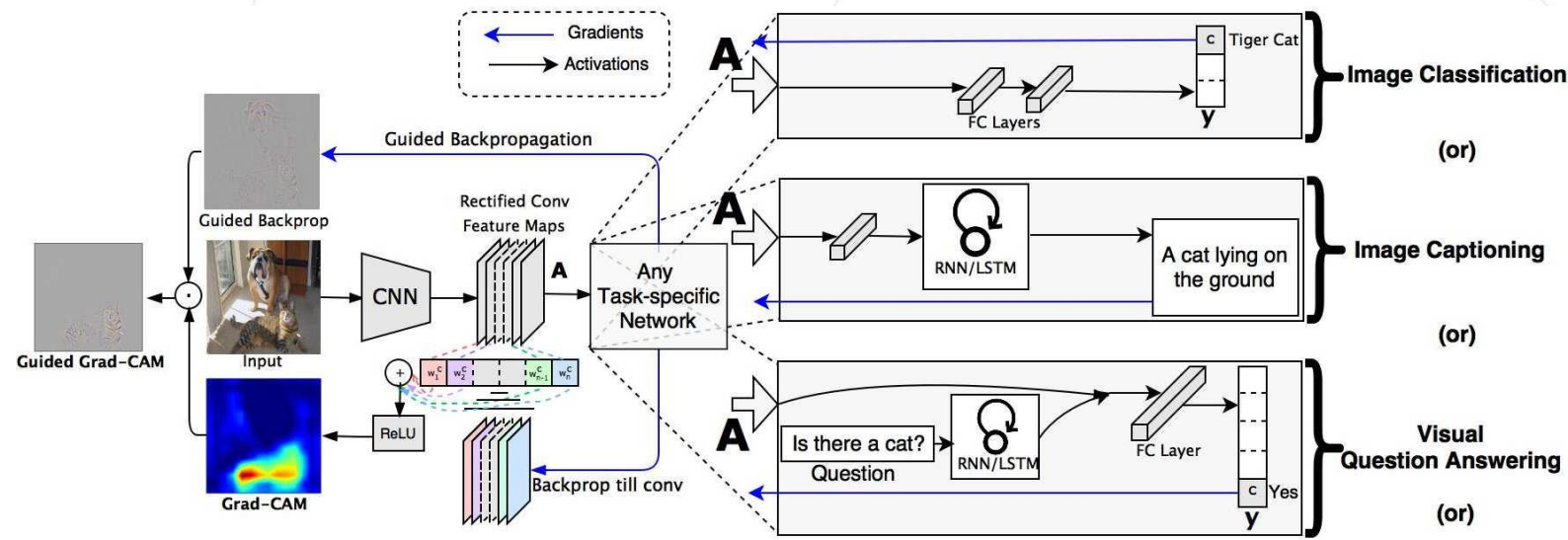
- Adversarial Training
 - Adversarial machine learning is a machine learning technique that attempts to fool models by supplying deceptive input
 - keeping track via this [link](https://en.wikipedia.org/wiki/Adversarial_machine_learning)

- Learning deep features for discriminative localization
 - Replace FC layer with Convs + GAP + Softmax
 - The class activation map is the weighted sum of weights and Convs outputs



Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning deep features for discriminative localization. CVPR2016 (pp. 2921-2929).

- Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization
 - Need no modification on network
 - Grad-CAM is a generalization to CAM



Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-cam: Visual explanations from deep networks via gradient-based localization. CVPR2017 (pp. 618-626).

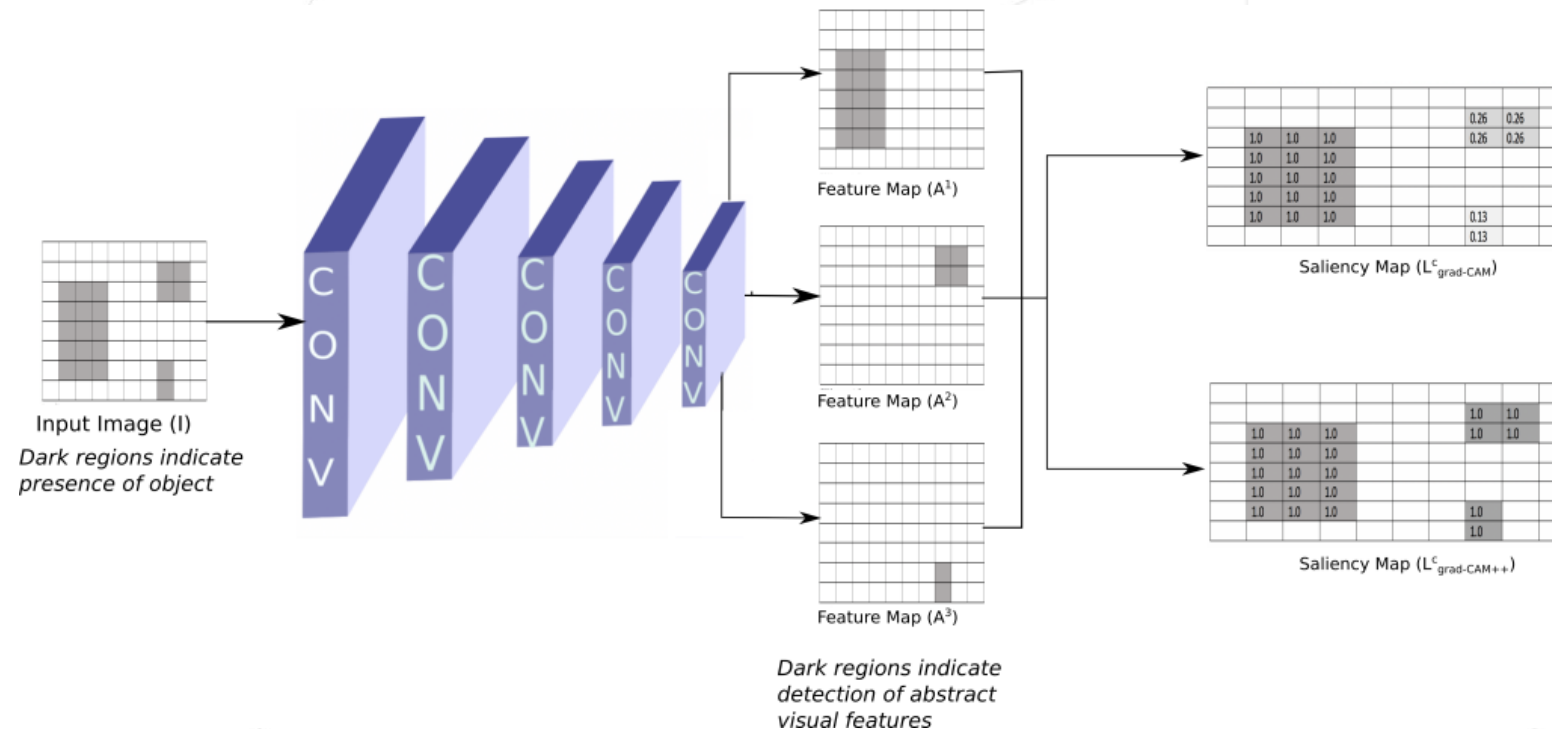
- Obtain the class discriminative localization map
 - Compute the gradient of the score for class c with respect to feature maps of a convolutional layer, i.e. $\partial y^c / A_{ij}^k$
 - Use GAP to obtain the neuron importance weights α_k^c

$$\alpha_k^c = \overbrace{\frac{1}{Z} \sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

- Weighted combination of forward activation maps followed by a ReLU

$$L_{\text{Grad-CAM}}^c = \text{ReLU} \left(\underbrace{\sum_k \alpha_k^c A^k}_{\text{linear combination}} \right)$$

- Grad-CAM++
 - Grad-CAM fails when localizing multiple occurrences of the same class
 - Grad-CAM heatmaps often do not capture the entire object in completeness



Chattopadhyay A, Sarkar A, et al. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. WACV2018 Mar 12 (pp. 839-847). IEEE.

- Grad-CAM++
 - Taking a weighted average of the pixel-wise gradients

$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} \cdot \text{relu}\left(\frac{\partial Y^c}{\partial A_{ij}^k}\right)$$

$$\alpha_k^c = \overbrace{\frac{1}{Z} \sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

- Combining above two equations

$$Y^c = \sum_k \left\{ \sum_a \sum_b \alpha_{ab}^{kc} \cdot \text{relu}\left(\frac{\partial Y^c}{\partial A_{ab}^k}\right) \right\} \left[\sum_i \sum_j A_{ij}^k \right]$$

- Grad-CAM++

- Taking partial derivative w.r.t. A_{ij}^k on both sides:

$$\frac{\partial Y^c}{\partial A_{ij}^k} = \sum_a \sum_b \alpha_{ab}^{kc} \cdot \frac{\partial Y^c}{\partial A_{ab}^k} + \sum_a \sum_b A_{ab}^k \left\{ \alpha_{ij}^{kc} \cdot \frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} \right\}$$

- Taking a further partial derivative w.r.t. A_{ij}^k :

$$\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} = 2 \cdot \alpha_{ij}^{kc} \cdot \frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \left\{ \alpha_{ij}^{kc} \cdot \frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3} \right\}$$

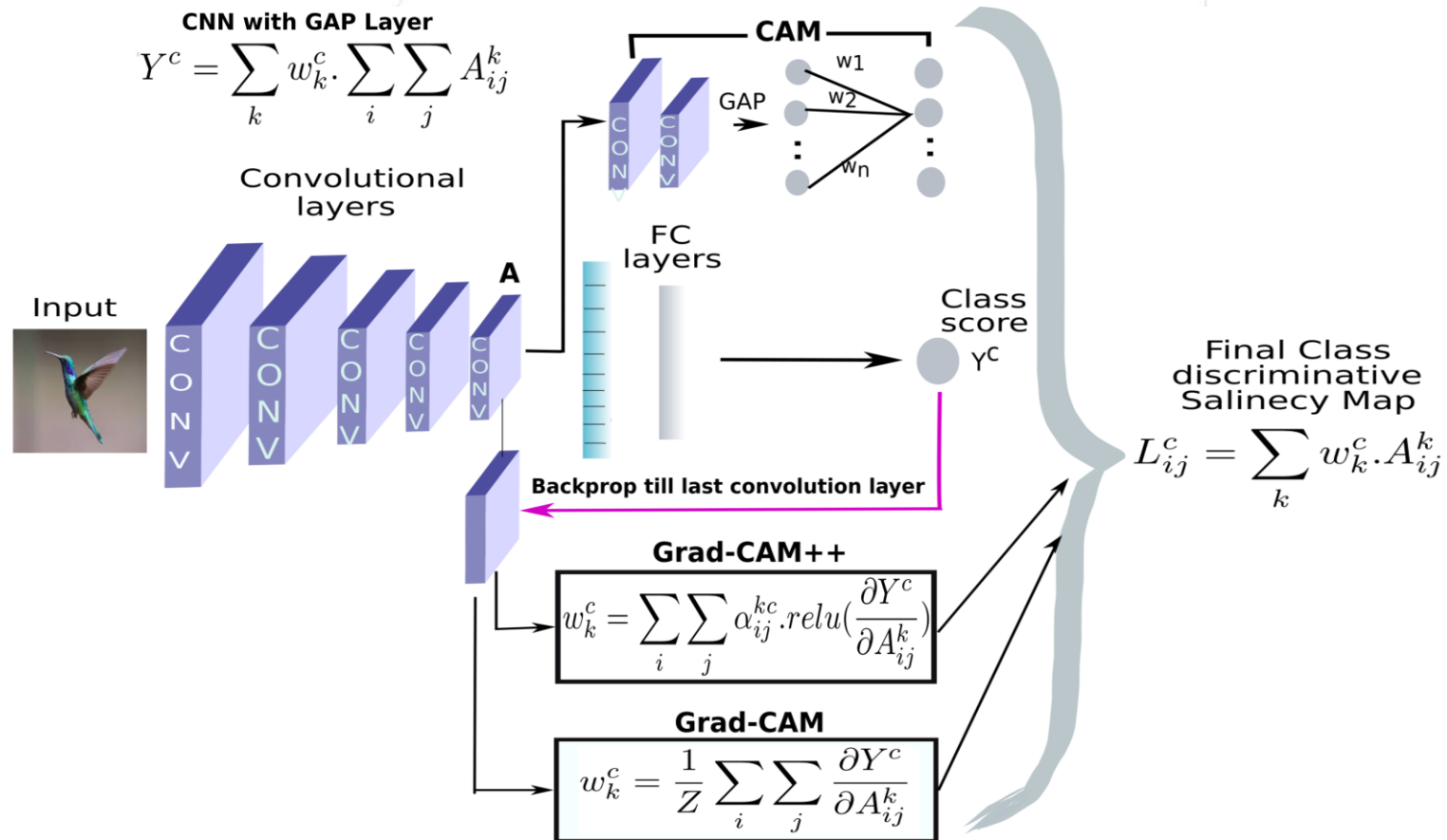
- Rearranging terms, we get:

$$\alpha_{ij}^{kc} = \frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2 \frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \left\{ \frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3} \right\}}$$

- Grad-CAM++
 - Substituting Eqn 10 in Eqn 5, we get the following GradCAM++ weights:

$$w_k^c = \sum_i \sum_j \left[\frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2 \frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \left\{ \frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3} \right\}} \right] \cdot \text{relu} \left(\frac{\partial Y^c}{\partial A_{ij}^k} \right)$$

- Overview of CAM, Grad CAM and Grad-CAM++





Outline

- Part 1** **Introduction to Image Classification**

- Part 2** **Supervised Image Classification**

- Part 3** **Semi-Unsupervised Classification Model**

- Part 4** **Further Research Topics**

- Part 5** **Recommended Competitions and Repos**

- FGVC8 , The Eight Workshop on Fine-Grained Visual Categorization
 - CVPR2021 workshop
 - <https://sites.google.com/view/fgvc8>

Prior FGVC Workshops

- [7th FGVC Workshop](#) @ CVPR 2020, Virtual
- [6th FGVC Workshop](#) @ CVPR 2019, Long Beach, CA
- [5th FGVC Workshop](#) @ CVPR 2018, Salt Lake City, UT
- [4th FGVC Workshop](#) @ CVPR 2017, Honolulu, HI
- [3rd FGVC Workshop](#) @ CVPR 2015, Boston, MA
- [2nd FGVC Workshop](#) @ CVPR 2013, Columbus, OH
- [1st FGVC Workshop](#) @ CVPR 2011, Colorado Springs, CO

- Related Repository

- [Deep Residual Learning for Image Recognition](#)
- [EfficientNet](#)
- [Mixup](#)
- [MixMatch](#)
- [ReMixMatch](#)
- [Learning to Navigate for Fine-grained Classification](#)



清华大学
Tsinghua University



END