

Alexnet

CNNの特徴

- ReLU
- Dropout
- Maxpooling

GoogleNet

CNNの特徴

- BatchNorm
- ReLU
- Maxpooling

Googlenetの構造

Inception



- 3x3 kernel $\times 1 \times 1 \times 3 \times 3 \times 5 \times 3 \times 3$
- 1x1 kernel $\times 1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1$



特徴



- GoogleNetの特徴
- Inception v2の特徴
 - Depthwise Separable Convolution
 - BatchNorm

Resnet

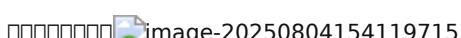
2015年提出された論文で、 $H(x) = x + F(x)$ という特徴的な構造を示す



ResNet Block

BasicBlock ブロック

BottleNeck ブロック



- **BasicBlock**

- **Layer2** \rightarrow layer **Block** \rightarrow Block \rightarrow Block
- $3 \times 3 \times 64$
- **block** \rightarrow **layer**
- - **1** **layer**
 - **2** **layer** \rightarrow **block** \rightarrow **block** \rightarrow **layer**

- **BottleNeck**

- **Layer** \rightarrow **Block** \rightarrow Block \rightarrow Block
- $3 \times 3 \times 1 + 3 \times 3 + 1 \times 1$
- $1 \times 1 \times 4$
- $3 \times 3 \times 3$
- $1 \times 1 \times 4 \times 4$
- - **1** **layer** \rightarrow **block** \rightarrow **block** \rightarrow **layer**
 - $1 \times 1 \times 4$
 - **2** **layer**

ResNet

ResNet

- **层**
- **层** **layer** **block** **stride=2** **层** **block** **stride=1**

ResNet



VGG

卷积层 $3 \times 3 \times 3$

卷积层 $3 \times 3 \times 5$, $5 \times 5 \times 5$, $7 \times 7 \times 7$

- **层** 3×3
- **层** $1 \times 1 \text{ stride} = 1 \text{ padding} = 1$
- **层** 3×3



MobileNet

A horizontal row of fifteen empty square boxes, intended for children to write their names in, likely as part of a classroom activity.

□**□□□□95%**□□□□\$1\times 1\\$□□□

10 / 10

A horizontal row of 24 small, empty rectangular boxes arranged in a single row.

- 亂世の政治家
 - 亂世の政治家
 - 亂世の政治家

1

□□□□\$\\alpha\$□□□□□□□□□□□□□□□□□□

5

□□□□\$\\rho\$□□□□□□□□□□□□

1

ReLU6  $\$ \$ \text{ReLU6}(x) = \min(\text{ReLU}(x), 6) \$ \$$

Inverted Residuals

ResNet

- ResNet \times BottleNeck \times \$1 \times \$3 \times \$3 \times \$1
 - MobileNetv2 \times \$1 \times \$3 \times \$3 \times \$1 \times \$1
 - Inverted residual block
 - Depthwise separable convolution

1

ReLU ReLU

三

- 



Vit|Vision Transformer

Transformer

5 of 5

1. မြန်မာစာပေါ်**patches**မြန်မာစာ
 2. မြန်မာစာ**patches**မြန်မာD⁷⁶⁸မြန်မာစာပေါ်မြန်မာစာပေါ် [CLS] Tokenမြန်မာBert
 3. မြန်မာtransformerencoderမြန်မာ**[CLS] Tokenမြန်မာpatchမြန်မာ**patch**မြန်မာ**
 4. မြန်မာencoderမြန်မာ [CLS] Tokenမြန်မာ

三

1. 『**CNN**による**Instructive Bias**の実験結果
 2. 『**CNN**による**Instructive Bias**の実験結果
 3. 『**CNN**による**Instructive Bias**の実験結果



Deit

www.vit.ac.in

distillation

6

- 
 - 
 - 

1

SEMANTIC SEGMENTATION

- 二〇一〇年九月三十日
 - 二〇一〇年十月一日

Instance Segmentation

-

□□□□□ Panoptic Segmentation □

- 



FPN | Feature Pyramid Network

¶tm¶kaiming¶¶¶¶¶

卷积神经网络

- 卷积神经网络是深度学习中的一种模型，广泛应用于图像识别、自然语言处理等领域。

卷积神经网络的结构

- 卷积神经网络由多层神经元组成，每层包含一个或多个卷积核。
- 卷积神经网络通常包含输入层、卷积层、池化层、全连接层等。

卷积层

- 卷积层是卷积神经网络的基本组成部分，**Backbone**是ResNet。
- 卷积层通过卷积操作将输入特征图转换为输出特征图。
- 卷积层通常包含一个或多个卷积核。
- 卷积层的输出尺寸为 1×1 。



FCN(Fully Convolutional Network)

卷积神经网络

卷积神经网络的结构

- 卷积神经网络由多层神经元组成，每层包含一个或多个卷积核。
- 卷积神经网络通常包含输入层、卷积层、池化层、全连接层等。

卷积神经网络的输出层

- 卷积神经网络的输出层为 1×1 。
- 1×1 的输出经过 softmax 激活函数。
- 卷积层
- 全连接层

FCN 结构

- VGG 和 ResNet 是 Backbone。
 - FCN-32s、FCN-16s、FCN-8s。
- FCN-32s：输入尺寸 32×32 。
 - FCN-16s：输入尺寸 2×2 ，pool4 尺寸 16×16 。
 - FCN-8s：输入尺寸 2×2 ，pool4 尺寸 2×2 ，pool3 尺寸 8×8 。



U-Net

- U-Net 是一种卷积神经网络，广泛应用于医学图像分割。
- U-Net 包含编码器和解码器。
- 编码器负责提取全局上下文信息，解码器负责生成最终的分割结果。



1

- $\text{max}(\$1 \times 572, \times 572) = 572$
 - $\text{conv2d}(\text{in_channels}=1, \text{out_channels}=64, \text{kernel_size}=3, \text{stride}=1, \text{padding}=0)$ \rightarrow $\text{ReLU}(\text{in_shape} - 2) \times \text{out_shape} = 1024$
 - $\text{conv2d}(\text{in_shape}=1024, \text{out_shape}=1024)$
 - $\text{conv2d}(\text{in_shape}=1024, \text{out_shape}=1024)$
 - $\text{conv2d}(\text{in_shape}=1024, \text{out_shape}=1024)$

1

Unet **Unet**

GoogleNet

- 00000000000000000000
 - 000000
 - 00000000000000000000
 - 000000

IoU

$\text{IoU} = \frac{\text{A} \cap \text{B}}{\text{A} \cup \text{B}}$

IoU

- IoU_{bbox}
 - IoU_{cls}



Dice Loss

Dice Loss = $1 - \frac{2|A \cap B|}{|A| + |B|}$

A decorative horizontal bar consisting of a series of small, evenly spaced squares.

BCE Loss

DiceLoss \equiv BCE $= - \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$

- y_i
- p_i
- N

Deeplab v2 v3 backbone

Deeplab v2 v3 backbone

Deeplab

Deeplab

v2 v3 backbone ResNet

Deeplab

1. Deeplab
2. Deeplab

LargeFov

Deeplab

Deeplab backbone

- Deeplab
- Deeplab

VGG

1. VGG FC6 FC7 \$7 \times 7\$
2. \$3 \times 3\$ r=12 \$7 \times 7\$
3. \$1 \times 1\$

ASPP model V3

ASPP model

ASPP model



ASPP model

ASPP model

- \$1 \times 1\$
- \$12 \times 3 \times 3\$
- \$24 \times 3 \times 3\$
- \$36 \times 3 \times 3\$
- \$1 \times 1\$

ASPP model



ဗိုလ်ချုပ်

- ဗိုလ်ချုပ်**16** \$3\times 3\$ 6**12 18**
- ဗိုလ်ချုပ်**8** \$3\times 3\$ 12**24 36** ဗိုလ်ချုပ်

poly

V2 ဗိုလ်ချုပ် \$ poly = lr \times (1 - \frac{iter}{max_iter})^{\text{power}} \$

ဗိုလ်ချုပ်

- \$power\$ ဗိုလ်ချုပ် 0.9-2.0
- \$iter\$ ဗိုလ်ချုပ်
- \$max_iter\$ ဗိုလ်ချုပ်

RCNN

ဗိုလ်ချုပ်

Backbone ဗိုလ်ချုပ် VGG16

ဗိုလ်ချုပ် 4 ဗိုလ်ချုပ် ဗိုလ်ချုပ်

1. ဗိုလ်ချုပ် **1k~2k**
2. ဗိုလ်ချုပ်
3. ဗိုလ်ချုပ် **SVM**
4. ဗိုလ်ချုပ်

ဗိုလ်ချုပ် NMS ဗိုလ်ချုပ်

1. ဗိုလ်ချုပ် **SVM**
2. ဗိုလ်ချုပ် **IoU**
3. ဗိုလ်ချုပ် **IoU**

Fast R-CNN

Backbone ဗိုလ်ချုပ် VGG16

ဗိုလ်ချုပ်

1. ဗိုလ်ချုပ် **1k~2k**
2. ဗိုလ်ချုပ် **SS**
3. ဗိုလ်ချုပ် **ROI pooling** \$7 \times 7 \times 512\$
 - ဗိုလ်ချုပ်
 - ဗိုလ်ချုပ် **N+1** ဗိုလ်ချုပ် **N** ဗိုလ်ချုပ် **1** ဗိုလ်ချုပ်

ဗိုလ်ချုပ် **IoU > 0.5**



ဗိုလ်ချုပ်

ဗိုလ်ချုပ် \$(N + 1) \times 4\$ ဗိုလ်ချုပ်



$\hat{G}_x = P_{wd_x}(P) + P_x \hat{G}_y = P_{hd_y}(P) + P_y \hat{G}_w =$
 $P_{wexp}(d_x(P)) \hat{G}_h = P_{hexp}(d_x(P))$

- P_x, P_y, P_w, P_h
- $\hat{G}_x, \hat{G}_y, \hat{G}_w, \hat{G}_h$

Multi-task loss

$L(p, \mu, t^{\mu}, v) = \underbrace{L_{cls}(p, \mu)}_{1} + \underbrace{\lambda[\mu \geq 1] L_{loc}(t^{\mu}, v)}_{\text{2}}$

- p softmax(p_0, \dots, p_k)
- μ
- t^{μ} **u**
- $(t_x^{\mu}, t_y^{\mu}, t_w^{\mu}, t_h^{\mu})$
- v
- $[\mu \geq 1]$
 - 1
 - 0

$L_{cls}(p, \mu) = -\log(p_{\mu})$

$L_{loc}(t^{\mu}, v) = \sum_i \in \{x, y, w, h\} smooth_{L_1}(t^{\mu} i - v_i) + smooth_{L_1}(x)$
 $x = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$
 $v_x = \frac{G_x - P_x}{P_w}, v_y = \frac{G_y - P_y}{P_h}, v_w = \ln(\frac{G_w}{P_w}), v_h = \ln(\frac{G_h}{P_h})$

- G_x, G_y, G_w, G_h
- P_x, P_y, P_w, P_h

ROI pooling

1. ROI
2. ROI
3. ROI
4. max pooling



Faster R-CNN

tn kaiming

RPN + Fast R-CNN

1. 1k~2k
2. RPN
3. ROI pooling 7×7

RPN Region Proposal Network

Backbone

1. RPN Backbone
2. Feature map

- $k=9$
 - [128, 256, 512] [1:1, 1:2, 2:1]
- \$3 \times 3\$ Backbone 256-dimension
-

3. RPN \$1 \times 1\$

1. $2k$
 - sigmoid softmax
2. $4k$
 - $2w \times 2k$ Fast-RCNN

anchor

RPN anchor 250 1:1

- IoU > 0.7 anchor
- IoU < 0.3

RPN anchor $2w \times 2k$ Fast-RCNN

RPN Multi-task loss

$$L(\{p_i\}, \{t_i\}) = \underbrace{\frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*)}_{\text{cls loss}} + \underbrace{\lambda \frac{1}{N_{reg}} \sum_i L_{reg}(t_i, t_i^*)}_{\text{reg loss}}$$

anchor

- p_i anchor
- p_i^* anchor
- t_i anchor
- t_i^* anchor
- N_{cls} mini-batch
- N_{reg} anchor
- L_{reg} smooth
- L_{cls}

Mask R-CNN

Faster R-CNN mask ROI Pooling ROI Align

Mask Faster R-CNN ROI Align

$$L_{mask} \text{ BCE-Loss } L_{mask_rcnn} = L_{RPN} + L_{Faster_rcnn} + L_{mask}$$

ROI Align

ROI Pooling

ROI Pooling

- ROI pooling
◦ maxpooling
◦ maxpooling + maxpooling

ROI Align

1. ROI Pooling
2. ROI Pooling
3. ROI pooling + ROI Align
4. ROI Pooling

Mask

FPN Mask



Mask

- Mask R-CNN
◦ mask
◦ sigmoid
◦ BCE



- Mask R-CNN
◦ Faster R-CNN
◦ mask



Mask R-CNN
mask
class

- softmax
• mask
◦ sigmoid

YoloV5

Backbone
DarkNet

YoloV7

YoloV8

Clip

Blip

10 of 10

- 亂世の政治家
 - 亂世の軍人

coco

COCO pycocotools.coco □ COCO annotations □ison

□□□□□□□□□□

方法	功能	示例
.getImgIds()	获取所有图像ID	train_coco.getImgIds()
.loadImgs(ids)	加载指定ID的图像	train_coco.loadImgs([1,2])
.getAnnIds(imgIds=[])	获取所有标注ID	train_coco.getAnnIds(imgIds=[1])
.loadAnns(ids)	加载指定ID的标注	train_coco.loadAnns([1,2])
.annToMask(annotation)	将标注转换为掩码	
.showAnns(annotations)	在matplotlib中显示标注	

Focal Loss

1

$\text{L} = -[y_i \ln p_i + (1-y_i) \ln(1-p_i)]$ $L = \frac{1}{100} [2.302 \times 0.98 + 2.302 \times 0.02]$
 $L = -2 \ln 0.5 = 1.386$ $L = 98 \times 0.683 = 67.914$ $L = \frac{1.386 + 67.914}{100} = 0.693$ $L = \frac{2 \times 2.302 + 98 \times 0.105}{100} = 0.325$

Focal Loss

□□□□ \$ L = -\alpha t(1 - p(t))^\gamma \log\{p(t)\} \$\$

- α 代表着单个事件发生的概率
 - p_t 代表了在时间 t 时所有事件发生的概率
 - γ 代表了 α 的累积概率
 - β 代表了 γ 的累积概率