## Course outline How does an NPTEL Week 3 Week 5 CNNs for Object Detection: Pre-Deep Learning Era and Initial Steps CNNs for Object Detection: Two-Stage Models CNNs for Object Detection: Single-stage Models CNNs for Segmentation • Week 6 Feedback Form : Deep Learning for Computer Practice: Week 6: Assignment 6(Non-Graded) Quiz: Week 6: Assignment 6 **Download Videos** Books

## Week 6: Assignment 6

The due date for submitting this assignment has passed.

Due on 2024-09-04, 23:59 IST.

## Assignment submitted on 2024-09-04, 23:46 IST

1) YOLO v1 i) Uses RepVGG-based backbone 2) YOLO v2 ii) CSPDarknet backbone

3) YOLO v3 iii) Use of Darknet-19 4) YOLO v4 iv) Feature pyramid networks

v) Single grid cell prediction

1→ v, 2→i, 3→iv, 4→ii

1→iv, 2→i, 3→iii, 4→ii

1→v, 2→iii, 3→i, 4→ii

1→v, 2→iii, 3→iv, 4→ii

Yes, the answer is correct. Score: 1

Accepted Answers: 1→v, 2→iii, 3→iv, 4—

2) Match the following

1) VGGNet i) 1 × 1 convolution

 $\begin{array}{lll} 1) \ \mathrm{VGGNet} & & \mathrm{i}) \ 1 \times 1 \ \mathrm{convolution} \\ 2) \ \mathrm{EfficientNet} & & \mathrm{ii}) \ \mathrm{identity} \ \mathrm{mapping} \\ 3) \ \mathrm{GoogleNet} & & \mathrm{iii}) \ 3 \times 3 \ \mathrm{convolution} \\ 4) \ \mathrm{ResNet} & & \mathrm{iv}) \ 7 \times 7 \ \mathrm{convolution} \\ & & \mathrm{v}) \ \mathrm{depth-wise} \ \mathrm{separable} \ \mathrm{convolutions} \end{array}$ 

 $1 \rightarrow \text{iv}, 2 \rightarrow \text{ii}, 3 \rightarrow \text{v}, 4 \rightarrow \text{i}$ 1→iv, 2→iii, 3→v, 4→ii 1→iv, 2→v, 3→ii, 4→iii 1→iii, 2→v, 3→i, 4→ii

Yes, the answer is correct. Score: 1

Accepted Answers: 1→iii, 2→v, 3→i, 4→ii

3) The ConvNEXT model is an evolution of convolutional neural networks (CNNs), designed to bridge the performance gap with vision transformers 1 point (ViTs). One of the key architectural innovations in ConvNEXT is the introduction of depthwise convolution blocks inspired by transformers. Which of the following statements regarding the architectural design of ConvNEXT is incorrect?

ConvNEXT employs depthwise convolutions followed by pointwise convolutions, similar to the MobileNet architecture, but with additional layer normalization and GELU activation

ConvNEXT replaces the traditional ResNet bottleneck block with a modified block that removes the ReLU activation function in favor of more nonlinear operations like Swish.

The ConvNEXT model increases the size of the convolutional kernels to 7x7 to better capture long-range dependencies, mimicking the self-attention

mechanism in transformers. OnvNEXT introduces LayerNorm after the depthwise convolution and before the pointwise convolution to stabilize the training dynamics

Accepted Answers:

ConvNEXT replaces the traditional ResNet bottleneck block with a modified block that removes the ReLU activation function in favor of more non-line operations like Swish.

4) Consider an object detection system evaluated on a dataset consisting of 1000 images. The system makes 1500 predictions across these images, and for each image, there are annotated ground truth bounding boxes. The system's precision-recall curve is calculated, and the precision at different recall levels for one of the classes is as follows:

Recall	Precision
0.1	0.90
0.2	0.85
0.3	0.80
0.4	0.75
0.5	0.70
0.6	0.65
0.7	0.60
0.8	0.55
0.9	0.50
1.0	0.45

Calculate the Average Precision (AP) for this class using the 11-point interpolation method, which averages the precision values at recall levels {0.0, 0.1, 0.2,

..., 1.0). The precision at recall 0.0 can be assumed to be 1.0.
Additionally, the system's AP values for the other two classes are as follows:

AP for class 2: 0.78

AP for class 3: 0.72

Based on these AP values, what is the mean Average Precision (mAP) across all three classes?

0.70

0.73

0.76 Yes, the answer is correct. Score: 1

Accepted Answers: 0.73

5) Match the following computer vision tasks to situations:

1) Instance Segmentation i) There are dogs in these pixels 2) Classification

ii) There are 4 dogs in the image, and here are the pixels with the shape of each of their occurrence

3) Semantic Segmentation iii) There is dog in image 4) Object Detection iv) There are 4 dogs in the image

 $1 \rightarrow iv$ ,  $2 \rightarrow iii$ ,  $3 \rightarrow ii$ ,  $4 \rightarrow i$  $1 \rightarrow \text{ii. } 2 \rightarrow \text{iii. } 3 \rightarrow \text{i. } 4 \rightarrow \text{iv}$ 1 . 1. 0 . 11 0 . 1 4 . 11

$1 \rightarrow ii, 2 \rightarrow iv, 3$	1 → 1, 4 → 11	
	$\rightarrow$ iii, 4 $\rightarrow$ i	
Yes, the answer is		
Score: 1		
Accepted Answers $1 \rightarrow ii$ , $2 \rightarrow iii$ , $3 \rightarrow ii$	$\stackrel{\text{s.}}{\leftarrow} i, 4 \rightarrow iV$	
6) Which one of th	ne following statements is false?	1 p
	mploys a compound scaling method to improve model efficiency and accuracy across different scales.	
	izes depthwise separable convolutions to reduce the computational complexity of the network.	
	y connects each layer to the last layer in a feedforward fashion to address the vanishing gradient problem.	
<ul> <li>SeNet incorpo dependencies.</li> </ul>	orates spatial squeeze-and-excitation blocks to enhance the representation power of the network by explicitly modeling channel	l-wise
Yes, the answer is	correct	
Score: 1		
Accepted Answers	s: Inects each layer to the last layer in a feedforward fashion to address the vanishing gradient problem.	
Deriservet only con	noots saurrays to the hast ayer in a reculorward hashion to dedress the variousing gradient problem.	
7) Which one of th	ne following object detection networks uses an ROI pooling layer?	1 p
Fast R-CNN		
R-CNN		
YOLO		
All of the abov	е	
Yes, the answer is Score: 1	correct.	
Accepted Answers	S:	
Fast R-CNN		
8) Consider two 12	2×12 bounding boxes (one on the upper left and one on the lower right) in an image with an overlapping region of 8 × 8. The	1 p
Intersection over Unio	on (IoU) score between the two boxes is (choose the closest value):	
O 10%		
O 18%		
© 28%		
35%		
Yes, the answer is Score: 1	correct.	
Accepted Answers	s:	
28%		
9)	[3 7 2]	1 p
The integral ima	age of the image 5 4 6	
	[8 1 9]	
[3 6 9]		
12 15 18		
21 24 27		
3 10 12		
8 19 27		
16 28 45		
3 7 12		
8 15 25		
16 28 45		
[9 6 3]		
8 5 2		
7 4 1		
Yes, the answer is	correct.	
Score: 1 Accepted Answers	8	
3 10 12		
8 19 27		
[16 28 45]		
10) Which of the fol	llowing is true? Select all possible answers:	1 p
The size of the	e effective receptive field reduces as we go deeper in Convolution Neural Network.	
	arning, if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task mo	odel's
general layers and	d append a new classifier to it instead of tuning on source task model's specific layers	
	f FLOPS required by EfficientNetB6 is less than the number of FLOPS required by EfficientNetB3	
	× 1 convolution on a feature map, it is good practice to apply padding since 1 × 1 reduces the height and width of feature map.	
	tion, the kernel moves in 3 directions and the input data is 4-dimensional	
Yes, the answer is Score: 1	correct.	
Accepted Answers		
For transfer !	Si gifthe course and torrest detects are disciplined and the torrest detects and the torrest detects.	
For transfer learning	g, if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's g	gener
For transfer learning layers and append	s: gif the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's g a new classifier to it instead of tuning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional	genei
For transfer learning layers and append In 3D convolution, t	g, if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's g a new classifier to it instead of tuning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional	
For transfer learning layers and append in 3D convolution, in the tinput have size D	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ a new classifier to it instead of tuning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $g_i$ $g_i$ $g_i$ where $g_i$	
For transfer learning layers and append in 3D convolution, in the tinput have size D	g, if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's g a new classifier to it instead of tuning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional	
For transfer learning layers and append in 3D convolution, in the size $D$ where $N=32$ . Assuring transfer in the size $D$ where $N=32$ .	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ a new classifier to it instead of tuning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $g_i$ $g_i$ $g_i$ where $g_i$	
For transfer learning layers and append in 3D convolution, in the size $D$ where $N=32$ . Assuring transfer in the size $D$ where $N=32$ .	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ and every expectation of tuning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $g_i$ and $g_i$ are already and $g_i$ and $g_i$ and $g_i$ are already expectation. The square kernel in convolution $g_i$ are $g_i$ and $g_i$ and $g_i$ are $g_i$ and $g_i$ are $g_i$ and $g_i$ and $g_i$ and $g_i$ are $g_i$ and $g_i$ and $g_i$ are $g_i$	
For transfer learning layers and append- In 3D convolution, In 3D con	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ and every expectation of tuning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $g_i$ and $g_i$ are already and $g_i$ and $g_i$ and $g_i$ are already expectation. The square kernel in convolution $g_i$ are $g_i$ and $g_i$ and $g_i$ are $g_i$ and $g_i$ are $g_i$ and $g_i$ and $g_i$ and $g_i$ are $g_i$ and $g_i$ and $g_i$ are $g_i$	
For transfer learning layers and append- $\ln 3D$ convolution, in 3D convolution, in Let input have size $D$ where $N=32$ . Assur Calculate the number 11) Number of Para 12800	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ an ew classifier to it instead of tuning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $g_i$ and $g_i$ are successful and $g_i$ are successful and $g_i$ are successful and $g_i$ and $g_i$ are successful and $g_i$ and $g_i$ are successful and $g_i$ and $g_i$ and $g_i$ are successful and $g_i$ are successful and $g_i$ are successful and $g_i$ are successful and $g_i$ and $g_i$ are successful and $g_i$ and $g_i$ are successful and $g_i$ are successful and $g_i$ and $g_i$ are successful a	
For transfer learning layers and append- In 3D convolution, In 3D con	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ an ew classifier to it instead of tuning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $g_i$ and $g_i$ are successful and $g_i$ are successful and $g_i$ are successful and $g_i$ and $g_i$ are successful and $g_i$ and $g_i$ are successful and $g_i$ and $g_i$ and $g_i$ are successful and $g_i$ are successful and $g_i$ are successful and $g_i$ are successful and $g_i$ and $g_i$ are successful and $g_i$ and $g_i$ are successful and $g_i$ are successful and $g_i$ and $g_i$ are successful a	
For transfer learning layers and append in 3D convolution, 1 MD convolution, 1 Let input have size D where N = 32. Assur Calculate the number 11) Number of Para 12800  Yes, the answer is Score 0.5 Accepted Answers	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ an ew classifier to $h$ instead of uning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $f_i \times D_f \times M$ where $D_f = 128$ and $M = 16$ and output feature map (after passing input through conv layer) has $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_$	
For transfer learning layers and append in 3D convolution, in Let input have size D where N = 32. Assur Calculate the number 11) Number of Para 12800  Yes, the answer is Score: 0.5	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ an ewe classifier to $h$ instead of uning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $f_i \times D_f $	N siz
For transfer learned layers and append. In 3D convolution, it let input have size D where N = 32. Assur Calculate the number 11) Number of Para 12800  Yes, the answer is Score: 0.5  Accepted Answers (Type: Numeric) 120	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ an ew classifier to it instead of uning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $f_i \times D_f \times $	
For transfer learning layers and append in 3D convolution, 1 MD convolution, 1 Let input have size D where N = 32. Assur Calculate the number 11) Number of Para 12800  Yes, the answer is Score 0.5 Accepted Answers	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ an ew classifier to it instead of uning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $f_i \times D_f \times $	N siz
For transfer learned layers and append. In 3D convolution, it let input have size D where N = 32. Assur Calculate the number 11) Number of Para 12800  Yes, the answer is Score: 0.5  Accepted Answers (Type: Numeric) 120	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ an ew classifier to it instead of uning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $f_i \times D_f \times $	N siz
For transfer learnin, layers and append. In 3D convolution, in 3D convolution, in 3D convolution, in 4D conv	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ an ew classifier to it initiated of uniting on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $f_i = f_i$ and $f_i = f_i$ and $f_i = f_i$ and output feature map (after passing input through convolution has $D_f \times D_f $	N siz
For transfer learning layers and append. In 3D convolution, in 3D conv	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ an ew classifier to it instead of uning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $f_i \times D_f \times $	N siz
For transfer learnin, layers and append. In 3D convolution, in 3D convolution, in 3D convolution, in 4D conv	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ and $g_i$ are an ew classifier to it instead of furning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $g_i$ and $g_i$ are the size $g_i$ and $g_i$ and $g_i$ and $g_i$ are the size $g_i$ and $g_i$ and $g_i$ are the size $g_i$ and $g_i$ and $g_i$ are the size $g_i$ and $g_i$ are the six $g_i$ and $g_i$ are the six $g_i$ and $g_i$ are the six $g_i$ and $g_i$ are the	N siz
For transfer learnin, layers and append. In SD convolution, In SD conv	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ and $g_i$ are an ew classifier to it instead of tuning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $g_i$ and $g_i$ are the same padded convolution. Let width of the square kernel in convolution layer $g_i$ and $g_i$ and $g_i$ are the parameters and computational cost for this convolution layer. The same expectation is $g_i$ and $g_i$ are the same expectation $g_i$	<i>N</i> siz
For transfer learning layers and append.  In 3D convolution, 1.  In 3D convolution, 1.  In 3D convolution, 2.  Where N = 32. Assur Calculate the number of Para 12800  Yes, the answer is Score: 0.5  Accepted Answers (Type: Numeric) 12:  12) Computational 12:  1209715200  Yes, the answer is Score: 0.5  Accepted Answers (Type: Numeric) 20:  Score: 0.5  Accepted Answers (Type: Numeric) 20:  Accepted Answers (Type	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ and $g_i$ are an ew classifier to it instead of tuning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $g_i$ and $g_i$ are the input data is 4-dimensional garden passing input through convex layer) has $g_i$ and $g_i$ and $g_i$ are the input data is 4-dimensional garden map (after passing input through convex layer) has $g_i$ and $g_i$ and $g_i$ and $g_i$ and $g_i$ and $g_i$ are the input data is 4-dimensional garden map (after passing input through convex layer) has $g_i$ and $g_i$ and $g_i$ and $g_i$ are the input data is 4-dimensional garden map (after passing input through convex layer) has $g_i$ and $g_i$ and $g_i$ are the input data is 4-dimensional garden map (after passing input through convex layer) has $g_i$ and $g_i$ are the input data is 4-dimensional garden map (after passing input through convex layer) has $g_i$ and $g_i$ are the input data is 4-dimensional garden map (after passing input through convex layer) has $g_i$ and $g_i$ are the input data is 4-dimensional garden map (after passing input through convex layer) has $g_i$ and $g_i$ are the input data is 4-dimensional garden map (after passing input through convex layer) has $g_i$ and $g_i$ and $g_i$ are the input data is 4-dimensional garden map (after passing input through convex layer) has $g_i$ and $g_i$ and $g_i$ are the input data is 4-dimensional garden map (after passing input through convex layer) has $g_i$ and $g_i$ and $g_i$ are the input data is 4-dimensional garden map (after passing input through convex layer) has $g_i$ and $g_i$ are the input data is 4-dimensional garden map (after passing input through convex layer) has $g_i$ and $g_i$ and $g_i$ are the input data is 4-dimensional garden	N siz
For transfer learning layers and append. In 3D convolution, in 3D conv	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ an ew classifier to it instead of uniting on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $f_i \times D_f \times M$ where $D_f = 128$ and $M = 16$ and output feature map (after passing input through conv layer) has $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f \times D_f \times M$ wher	N si:
For transfer learning layers and append. In 3D convolution, in 3D conv	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ an ew classifier to it instead of uniting on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $f_i \times D_f \times M$ where $D_f = 128$ and $M = 16$ and output feature map (after passing input through conv layer) has $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f \times D_f \times$	N siz
For transfer learning layers and append. In 3D convolution, it always are seen as a se	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ an ew classifier to it instead of uniting on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $f_i \times D_f \times M$ where $D_f = 128$ and $M = 16$ and output feature map (after passing input through conv layer) has $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f \times D_f \times M$ wher	N si:
For transfer learning layers and append. In 3D convolution, it always are seen as a se	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ an ew classifier to it instead of uniting on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $f_i \times D_f \times M$ where $D_f = 128$ and $M = 16$ and output feature map (after passing input through conv layer) has $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f \times D_f \times$	N si:
For transfer learning layers and append. In 3D convolution, 1.  Let input have size D where N = 32. Assur Calculate the number 111 Number of Para 12800  Yes, the answer is Score: 0.5 Accepted Answers (Type: Numeric) 12.  12) Computational 1209715200  Yes, the answer is Score: 0.5 Accepted Answers (Type: Numeric) 20.  Accepted Answers (Type: Numeric) 20.  Using the same dimer Seperable convolutional 13 Number of para 200.	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ an ew classifier to it instead of uniting on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $f_i \times D_f \times M$ where $D_f = 128$ and $M = 16$ and output feature map (after passing input through conv layer) has $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f = 128$ and $D_f \times D_f \times M$ where $D_f$	N siz
For transfer learning layers and append. In 3D convolution, in 3D conv	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ an ew classifier to it instead of uniting on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $f_i \times D_f \times M$ where $D_f = 128$ and $M = 16$ and output feature map (after passing input through conv layer) has $D_f \times D_f \times M$ me padded convolution. Let width of the square kernel in conv layer be $k$ where $k = 5$ (ignore the bias term in the calculation). For parameters and computational cost for this convolution layer. The parameters are correct.  Solve the second of the previous question, calculate the number of parameters and computational cost, but make use of $f_i$ $f_$	N siz
For transfer learning layers and append. In 3D convolution, in 3D conv	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ an ewe classifier to it instead of tuning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional and $g_i$ and $g_i$ are supported by $g_i$ and $g_i$ and $g_i$ are supported by $g_i$ and $g_i$ and $g_i$ are supported by $g_i$ and	N siz
For transfer learning layers and append. In 3D convolution, in 3D conv	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ and ew classifier to it instead of tuning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional the kernel moves in 3 directions and the input data is 4-dimensional $g_i = g_i = g_i$ and $g_i =$	N siz
For transfer learning layers and appears in 3D convolution, in 3D conv	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ and ew classifier to it instead of tuning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional the kernel moves in 3 directions and the input data is 4-dimensional $g_i = g_i = g_i$ and $g_i =$	N siz
For transfer learning layers and appears in 3D convolution, in 3D conv	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ an ew classifier to it instead of tuning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional the kernel moves in 3 directions and the input data is 4-dimensional $g_i \times D_f \times M$ where $D_f = 128$ and $M = 16$ and output feature map (after passing input through conv layer) has $D_f \times D_f \times M$ mere apadded convolution. Let width of the square kernel in conv layer be $k$ where $k = 5$ (Ignore the bias term in the calculation). For parameters and computational cost for this convolution layer, ameters:    Correct.   Given the previous question, calculate the number of parameters and computational cost, but make use of $M$ per the on instead of standard convolution.    Given the previous question, calculate the number of parameters and computational cost, but make use of $M$ per the on instead of standard convolution:    Given the previous question, calculate the number of parameters and computational cost, but make use of $M$ per the on instead of standard convolution:    Given the previous question, calculate the number of parameters and computational cost, but make use of $M$ per the one instead of standard convolution:    Given the previous question, calculate the number of parameters and computational cost, but make use of $M$ per the parameters for depthwise convolution:	N siz
For transfer learning layers and append. In 3D convolution, 1 and convolution, 2 where N = 32. Assur Calculate the number of Para 12800  Yes, the answer is Score: 0.5 Accepted Answers (Type: Numeric) 121  12) Computational 1209715200  Yes, the answer is Score: 0.5 Accepted Answers (Type: Numeric) 201  13) Number of para 1400  Yes, the answer is Score: 0.5 Accepted Answers (Type: Numeric) 201  Using the same dimer Seperable convolutional 1400  Yes, the answer is Score: 0.5 Accepted Answers (Type: Numeric) 401  14) Computational 140  1553600	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ and ew classifier to it instead of tuning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional the kernel moves in 3 directions and the input data is 4-dimensional $g_i \times D_f \times M$ where $D_f = 128$ and $M = 16$ and output feature map (after passing input through convilayer) has $D_f \times D_f \times D_f \times M$ where $D_f = 128$ and $M = 16$ and output feature map (after passing input through convilayer) has $D_f \times D_f \times M$ are padded convolution. Let width of the square kernel in convilayer be $k$ where $k = 5$ (Ignore the bias term in the calculation). For parameters and computational cost for this convolution layer, ameters:    Correct.   Given the previous question of the parameters and computational cost, but make use of Depther on instead of standard convolution.   Given the parameters and computational cost, but make use of Depther on instead of standard convolution:    Correct.   Given the previous question of the number of parameters and computational cost, but make use of Depther on instead of standard convolution:    Correct.   Given the parameters and computational cost, but make use of Depther on instead of standard convolution:    Correct.   Given the parameters and computational cost, but make use of Depther on instead of standard convolution:    Correct.   Given the parameters and computational cost, but make use of Depther on instead of standard convolution:    Correct.   Given the parameters and computational cost, but make use of Depther on instead of standard convolution:    Correct.   Given the parameters and computational cost, but make use of Depther on the parameters and computational cost, but make use of Depther on the parameters and computational cost, but make use of Depther on the parameters and computational cost, but make use of Depther on the parameters and computational cost, but make use of De	N siz
For transfer learning layers and appendingly and appendingly and appendingly and appendingly and appendingly appen	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ and ew classifier to it instead of tuning on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional the kernel moves in 3 directions and the input data is 4-dimensional $g_i \times D_f \times M$ where $D_f = 128$ and $M = 16$ and output feature map (after passing input through convilayer) has $D_f \times D_f \times D_f \times M$ where $D_f = 128$ and $M = 16$ and output feature map (after passing input through convilayer) has $D_f \times D_f \times M$ are padded convolution. Let width of the square kernel in convilayer be $k$ where $k = 5$ (Ignore the bias term in the calculation). For parameters and computational cost for this convolution layer, ameters:    Correct.   Given the previous question of the parameters and computational cost, but make use of Depther on instead of standard convolution.   Given the parameters and computational cost, but make use of Depther on instead of standard convolution:    Correct.   Given the previous question of the number of parameters and computational cost, but make use of Depther on instead of standard convolution:    Correct.   Given the parameters and computational cost, but make use of Depther on instead of standard convolution:    Correct.   Given the parameters and computational cost, but make use of Depther on instead of standard convolution:    Correct.   Given the parameters and computational cost, but make use of Depther on instead of standard convolution:    Correct.   Given the parameters and computational cost, but make use of Depther on instead of standard convolution:    Correct.   Given the parameters and computational cost, but make use of Depther on the parameters and computational cost, but make use of Depther on the parameters and computational cost, but make use of Depther on the parameters and computational cost, but make use of Depther on the parameters and computational cost, but make use of De	<i>N</i> si
For transfer learning layers and append. In 3D convolution, in 3D conv	$g_i$ if the source and target datasets are dissimilar and the target dataset size is small, it is better to only use source task model's $g_i$ are new classifier to it instead of uniting on source task model's specific layers the kernel moves in 3 directions and the input data is 4-dimensional $f_i \times D_f \times M$ where $D_f = 128$ and $M = 16$ and output feature map (after passing input through convolution. Let width of the square kernel in convolution and the input data is 4-dimensional or or parameters and computational cost for this convolution layer.  The parameters are computational cost for this convolution layer. The correct is a second or convolution and the previous question, calculate the number of parameters and computational cost, but make use of $f_i = 16$ correct.  The parameters are computational cost, but make use of $f_i = 16$ correct.  The parameters are computational cost, but make use of $f_i = 16$ correct.  The parameters are computational cost, but make use of $f_i = 16$ correct.  The parameters are computational cost, but make use of $f_i = 16$ correct.  The parameters are computational cost, but make use of $f_i = 16$ correct.  The parameters are computational cost, but make use of $f_i = 16$ correct.  The parameters are computational cost, but make use of $f_i = 16$ correct.  The parameters are convolutional cost, but make use of $f_i = 16$ correct.	N si:

	0.25 points
15) Number of parameters for pointwise convolution:	
512	
Yes, the answer is correct. Score: 0.25	
Accepted Answers: (Type: Numeric) 512	
	0.25 points
16) Computational cost for for pointwise convolution:	
8388608	
Yes, the answer is correct. Score: 0.25	
Accepted Answers: (Type: Numeric) 8388608	
(19pe. Humano) occooo	
	0.25 points