The Ikshana Hypothesis of Human Scene Understanding

Venkata Satya Sai Ajay Daliparthi

1 Introduction

The human brain can seamlessly perceive diverse perceptual and semantic information regarding the natural scene/image during a glance [21, 30, 58, 62]. The visual scene information perceived during/after a glance refers to the gist (a summary) of the scene/image. The gist includes all the visual information from the low-level (e.g., colors and contours) to the high-level (e.g., shapes and activation). Due to this reason, [55] suggested that the gist can be investigated at both the perceptual and conceptual levels. The structural representation of the image refers to the perceptual gist, and the semantic information of the image refers to the conceptual gist. However, the conceptual gist is more refined and modified than the perceptual gist [55]. Several works [4, 19, 20, 26, 57, 58, 65] in neuroscience have addressed the fundamental question, i.e.,"how does the human brain performs several visual tasks?" by investigating through conceptual and perceptual gist. They conducted several experiments and proposed various theories to explain how modeling of the scene occurs in the human brain. However, there was no general principle that explains the functioning of the human brain. Even though there is a general principle, we expect that to be different from human-to-human. Depending on the situation and the environment, the human brain can seamlessly grasp the information by recognizing the objects and observing their structure. On the other hand, for a computer to do the same is the fundamental goal of the computer vision field.

In recent years, deep learning methods have shown a significant improvement over traditional handcrafted techniques on several computer vision tasks. Though these deep neural networks (DNNs) achieved state-of-the-art performance in many cases, the one major drawback is the requirement of massive labeled data. The collection of a huge amount of labeled data is an expensive and time taking process. Even though

V. S. S. A. Daliparthi (✉)
Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden
e-mail: veda18@student.bth.se

U. Mudenagudi et al. (eds.), *Proceedings of the Satellite Workshops of ICVGIP 2021*,
Lecture Notes in Electrical Engineering 924,
https://doi.org/10.1007/978-981-19-4136-8_12

these DNNs are said to be inspired by the functioning of the human brain, is this how the human brain learns to perform any visual task? **NO**. Because the human brain does not require massive labeled data to perform any visual task, and it can perform with few data samples. However, we cannot observe a similar phenomenon in the case of many DNNs.

Semantic segmentation is the task of assigning a class label to every pixel in the given image, which has applications in various fields such as medical, autonomous driving, robotic navigation, localization, and scene understanding. The prominent work FCN [48] adopted the image-classification networks [35, 75, 81] for semantic segmentation. Later on, several works [2, 12, 67, 78, 83, 98, 100, 107] improved the FCN [48] architecture, and proven to be successful in diverse semantic segmentation benchmarks [5, 15, 108]. However, these methods mainly focus on achieving state-of-the-art performance by using the entire and additional datasets [16] (for pre-training). Due to this reason, even though various methods [12, 78] outperformed U-Net [67] in terms of accuracy and computational complexity, the U-Net [67] architecture is still exploited in several medical image segmentation methods due to its ability to perform with few data samples [74]. Although several few-shot semantic segmentation (FSS) methods are introduced to address this problem, they often use techniques such as meta-learning [18, 59, 64, 85, 93] and metric learning [73, 89, 89, 90, 95, 101, 102, 106] on top of the existing architectures.

Unlike FSS methods, we tackle the formerly mentioned drawback of the DNNs, i.e., the requirement of massive labeled data, from a neuroscience perspective. In this work, we propose a hypothesis of human scene understanding mechanism named Ikshana. The idea is that, "to understand the conceptual gist of a given image; humans look at the image multiple times recurrently at different scales". Following the Ikshana hypothesis, we propose a novel neural-inspired CNN architecture named IkshanaNet, a multi-scale architecture that learns representations at full image resolution. In contrast to the existing CNN architectures that pass the input image only to the initial layer (stem module), our method feeds the input image to every module in the network and to the best of our knowledge, this is the first work to propose the same.

To evaluate the performance of IkshanaNet, we conduct extensive experiments on the entire and subsets of the Cityscapes and Camvid benchmarks. Moreover, we conduct multiple ablation studies to verify the effect of image scales in IkshanaNet. The empirical results illustrate that our method outperforms several baselines on the entire and few data samples. Furthermore, the ablation studies shows the importance of multi-scale information in achieving considerable performance. We hope that our hypothesis sparks future research in neural network architectures for vision tasks.

2 Related Work

In **Neurological** terms, all the low-level and high-level computer vision tasks come under a single term called human scene understanding. A scene is a view of a real-world environment that contains multiple surfaces and objects organized in a mean-

ingful way. In neuroscience, the perceptual gist is more investigated compared to the conceptual gist. The early works on the conceptual gist [30, 61] explained that a typical scene fixation of 275 to 300 ms is often sufficient to understand the gist of the image. Several works on the perceptual gist [4, 19, 20, 26, 56–58, 65, 72] provided insight into how the modelling of the scene occurs in the human brain through perceiving boundaries, blobs, scales, texture, contours, openness, depth, and so on. The information perceived through the perpetual gist is refined and extracted into the conceptual gist (the semantic meaning) during the cognitive process. Thus, the conceptual gist is highly dependent upon the perceptual gist. In many cases [15, 16, 108], we do not explicitly encode the perceptual process in DNNs, and the CNN learns various representations regarding the image during the training process. Thus, our hypothesis focuses on the conceptual gist rather than the perceptual gist.

Neural networks exist from a long time [50, 68, 70] and some prominent works [14, 16, 25, 35, 43, 75, 81, 82] made them popular during recent years. In our work, we use the convolutional neural network (CNN) architecture [36, 88] to learn representations from the images, which itself is inspired by [23, 29]. The architecture of IkshanaNet is inspired by [28, 75] and related to [27, 39].

The first seminal work on **Semantic segmentation (SS)** using deep learning is the fully convolutional networks (FCN) [48]. Later on, many semantic segmentation networks followed the FCN [48] architecture. The total prominent works on deep learning-based semantic segmentation methods can be roughly classified into five categories. They are (i) Encoder-decoder based methods (DeconvNet [54], SegNet [2], U-Net [67], RefineNet [41, 42], FC-DenseNet [33], and GFR-Net [1]), (ii) Regional proposal methods (MaskRCNN [24], FPN [44], and PANet [46]), (iii) Increased resolution of feature map methods (DeepLab series [8–10, 12], PSPNet [107], DenseASPP [96], and HRNet [78]), (iv) Context information methods (ParseNet [47], ATS [11], DANet [22], OCNet [99], OCR [98], EncNet [104], Non-local [91], ZigZagNet [40], ACFNet [103], CoCurNet [105], GLAD [38], and HANet [13]) (v) Boundary refinement methods ([3, 7, 17, 49], Gated-SCNN [83], and SegFix [100]). The IkshanaNet uses the dilated convolutions, interpolation of feature maps, and skip connections from different layers in the network. Therefore, our work is related to the formerly mentioned encoder-decoder and increased resolution of feature map methods.

Few-shot segmentation (FSS) methods [6, 18, 45, 59, 64, 73, 85, 86, 89, 90, 93, 95, 101, 102, 106] are introduced to handle limited training data. They use meta-learning (knowledge distillation), metric-learning (similarity learning), and a combination of both the techniques on top of FCN [48] based architectures, which often involve multistage training. The metric-learning techniques can be further classified into the prototypical feature learning [18, 37, 87, 90, 102, 106] and the affinity learning [89, 97, 101] techniques. Unlike general SS methods, FSS methods are evaluated on different benchmarks and handle novel class categories during testing. Since the IkshanaNet does not use any of the formerly mentioned FSS techniques and only handles the classes seen in the training data, our method is more closely related to the general SS methods than the FSS methods.

3 Method

3.1 Ikshana (the Eye) Hypothesis

In her prominent work [61], professor Mary C. Potter found that an average human can understand the gist of the image between the time interval of 125 to 300 ms. Furthermore, through several works [19, 20, 26, 30, 57, 58, 65, 72] in neuroscience, it is evident that humans understand the gist of the image in a certain time interval. During that time interval, the Ikshana hypothesis approximates the functioning of the human brain. The Ikshana hypothesis states that "**To understand the conceptual gist of a given image, humans look at the image multiple times recurrently, at different scales**." The word Ikshana is derived from the Sanskrit language, which has many synonyms such as the eye, sight, look, and so on.

We present an example to explain the Ikshana hypothesis in Fig. 1, where there is an image (x) on the left side and the human brain mechanism on the right side. According to the Ikshana hypothesis, for a human to understand the conceptual gist of the given image, the following process occurs in the human brain:

At a time step (t), during the first glance (Φ_1), the brain learns the first representation $(f(x))$ from the image (x) and stores that representation in the memory (M), as shown in the Eq. 1.

$$f(x) = \Phi_1(x); \quad M = f(x) \tag{1}$$

At a time step $(t+1)$, during the second glance (Φ_2), the brain holds the first representation $(f(x))$ in the memory and learns the second representation $(g(x))$ from the image and the first representation $(x, f(x))$. Then the brain stores the representation $(g(x))$ along with $(f(x))$ in the memory (M), as shown in the Eq. 2.

$$g(x) = \Phi_2(x, f(x)); \quad M = f(x), g(x) \tag{2}$$

At a time step $(t+2)$, during the third glance (Φ_3), the brain holds the first and the second representations $(f(x), g(x))$ in the memory and learns the third representation $(h(x))$ from the image and the previous representations $(x, f(x), g(x))$. Then the brain stores the representation $(h(x))$ along with $(f(x), g(x))$ in the memory (M), as shown in the Eq. 3.

$$h(x) = \Phi_3(x, f(x), g(x)); \quad M = f(x), g(x), h(x) \tag{3}$$

From Eqs. 1, 2, and 3, this kind of recurrent process occurs at $(t+n)$ times at a single image scale. Depending upon the given task (T), by combing all the information stored in the memory until the $(t+n)$th time step, the brain understands the conceptual gist (Y_1) of the image at a single scale, as shown in the Eq. 4.

$$Y_1 = T(f(x), g(x), h(x).....n(x)) \tag{4}$$

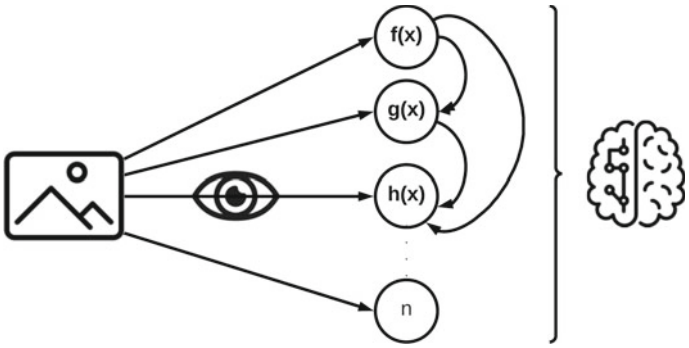

Fig. 1 The Ikshana Hypothesis at single scale

This process occurs at N different scales and generates N different outputs (Y_1, Y_2, Y_3, ..., Y_n). By considering all the outputs, the brain selects some of those representations and forgets the remaining representations. In this way, the brain learns (Δ) the final output (Y) of the given visual task (T), as shown in the Eq. 5.

$$Y = \Delta(Y_1, Y_2, Y_3, ...Y_N) \tag{5}$$

From the Eqs. 1, 2, 3, 4, and 5, this is how Ikshana hypothesis approximates the functioning of the human brain, while human understands the conceptual gist of the image. The time required (or the number of glances required) by an average human to understand the gist of the image may depend upon several factors such as the given task, age, intelligence, memory, and so on.
The existing CNN architectures such as VGG [75], Resnet [25], DenseNet [28], and so on learns a representation (say $f(x)$) with 32/64 filters from the input image and learns further representations on top of the $f(x)$ until the network achieves adequate performance. In contrast, the network designed by following the Ikshana hypothesis learns representations from the input image and previous outputs at each glance/layer.

3.2 IkshanaNet Architecture

In this section, we introduce a novel neural-inspired encoder-decoder CNN architecture named IkshanaNet, designed by following the Ikshana hypothesis. Humans can look at the image and seamlessly learn various useful representations regarding it [21, 30, 58, 62]. On the other hand, for a computer to do the same, we use the convolutional neural network [23, 29, 36] architecture to learn representations. The IkshanaNet architecture uses three image scales and consists of 4M parameters. The entire architecture is made of three building blocks, and they are: (1) the glance

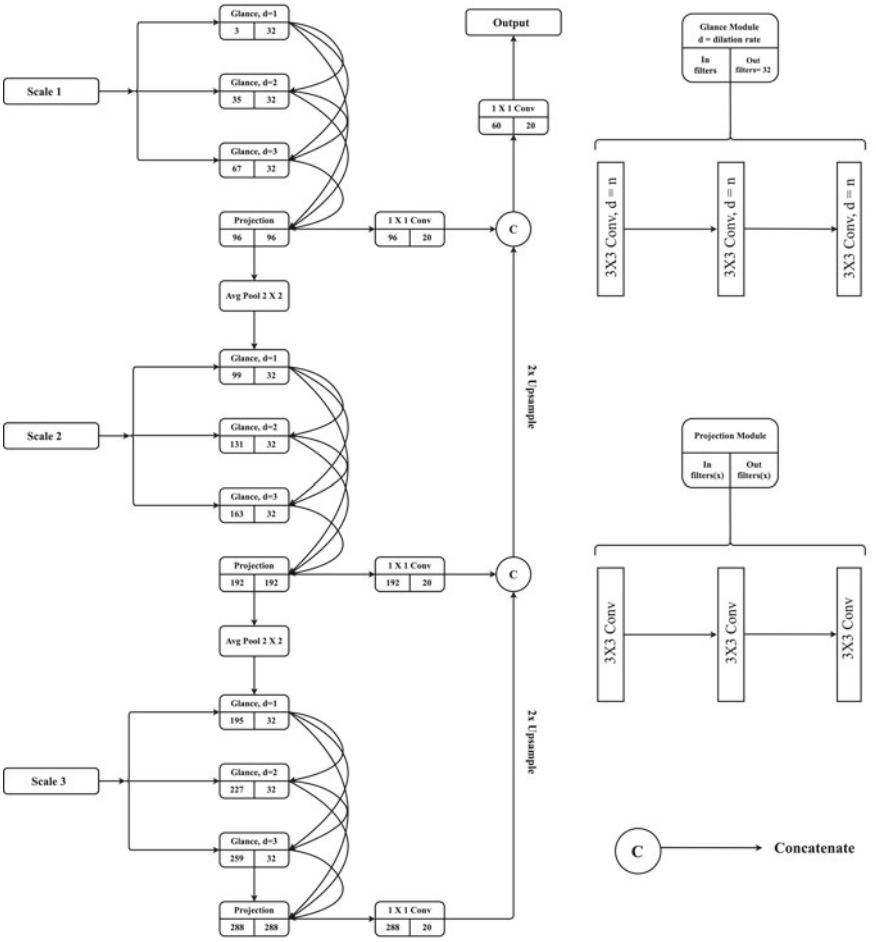

Fig. 2 IkshanaNet-main architecture

module, (2) the projection module, and (3) a 1 × 1 convolutional layer, as illustrated in Fig. 2.

The **glance module** consists of three 3 × 3 convolutional layers (with the same dilation rates), and we use it to learn representations from the given image (or a feature map). The number of input filters passed into the glance module varies several times in the architecture; however, it always returns a feature map with 32 filters. The **projection module** consists of three 3 × 3 convolutional layers, and we use it to refine the representations learned from the glance modules. The input and output filters are always the same for the projection module. We use the **1 × 1 convolution layers** to reduce the number of filters in a given future map. Except for the last 1 × 1 convolutional layer that returns the final output, every convolutional layer in the

architecture is followed by a batch normalization [31] and a ReLU [52] activation layer.

In **the encoder** part, the IkshanaNet learns representations at three image scales. At scale 1, we pass the input image through a glance module with a dilation rate (d = 1), which returns a feature map with 32 filters. Then we concatenate the input image with the previously learned feature map ($32 + 3 = 35$). The concatenation of the input image with the feature map is essential to ensure that we are learning representations from the input image. Then we pass the feature map through another glance module with a dilation rate (d = 2) and concatenate the resulting feature map with the feature maps from the preceding layers ($32 + 32 + 3 = 67$). We pass the resulting feature map through another glance module with a dilation rate (d = 3), which takes in 67 filters and returns 32 filters. Again, we concatenate the resulting feature map with feature maps from the preceding layers ($32 + 32 + 32 + 3 = 99$). At this point, we remove the input image from the feature map through tensor slicing ($99 - 3 = 96$), and the resulting feature map consists of ($32 + 32 + 32 = 96$) filters learned from three glances modules. In this way, the network followed the Ikshana hypothesis had three glances recurrently at the full resolution. Then we pass the feature map through a projection module to refine the representations ($96 = 96$). Here, we pass the refined feature map through a 1×1 convolutional layer that reduces 96 filters into 20 filters and name it the side one output (Y_1). Simultaneously, we pass the feature map through an average pooling layer, which reduces the size of the feature map by a factor of two.

At scale 2, we down-sample the input image by a factor of two and concatenate with the pooled feature map from the scale 1 ($96 + 3 = 99$). We pass the resulting feature map with 99 filters through three glance modules with different dilation rates (d = 1, 2, 3) and concatenate all the outputs as follows ($99 + 32 + 32 + 32 = 195$). Then we remove the image from the feature map ($195 - 3 = 192$) and pass it through a projection module to refine the representations ($192 = 192$). Then we pass the refined feature map through a 1x1 convolutional layer that reduces 192 filters into 20 filters and name it the side two output (Y_2). Then, we pass the refined feature map through an average pooling layer that reduces the size by a factor of two.

At scale 3, we down-sample the input image by a factor of four and concatenate with the pooled feature map from the scale 2 ($192 + 3 = 195$). Here, we follow the same process ($195 + 32 + 32 + 32 = 291$); ($291 - 3 = 288$); ($288 == 288$) as the scale 2 part, which returns a feature map with 20 filters, and name it the side three output (Y_3).

In **the decoder** part, we bi-linearly interpolate the outputs from two scales (Y_2 and Y_3) to match with the output of scale 1 Y_1, i.e., the input image size. Then we concatenate all the three outputs ($20 + 20 + 20 = 60$) and pass it through a 1x1 convolutional layer, which returns a feature map with 20 filters, that is the final output of the network [$Y = \Delta(Y_1, Y_2, Y_3)$].

Depth Architectures: Here, we introduce three variants of the IkshanaNet named IkshanaNet-3G, IkshanaNet-6G, and IkshanaNet-12G. If we remove the projection layers in IkshanaNet-main, then it will remain with three scales and three glances at each scale; it is IkshanaNet-3G (which consists of 514 K parameters). If we increase

the number of glances per scale, from three to six, then it is IkshanaNet-6G (which consists of 1.8M parameters), and from three to twelve, then it is IkshanaNet-12G (which consists of 6.5M parameters).

Multi-scale Architectures: Here, we introduce three variants of IkshanaNet named IkshanaNet 1S-6G, 2S-3G, and 3S-2G. In IkshanaNet 1S-6G, there are no pooling layers and contain six glances at full-scale resolution (which consists of 257K parameters). In IkshaNet 2S-3G, there are two scales and three glances at each scale (which consists of 259K parameters). In IkshanaNet 3S-2G, there are three scales and two glances at each scale (which consists of 260 K parameters).

4 Experiments

4.1 Experimental Setup

GPU: 1 X NVIDIA Tesla T-4 (16 GB VRAM)
Framework: PyTorch 1.8 [60]
Epochs: 180 ; **Batch Size**: 2
Criterion: Pixel-wise cross-entropy loss
Learning Rate Scheduler: ReduceLROnPlateau (decrease factor = 0.5 and patience = 20 epochs) with an initial learning rate of $1e-06$.
Optimizer: Stochastic gradient descent [66] with Nesterov momentum [53][1]
Random Seed: To ensure that data splits are reproducible, we set the random seed 42 in the function torch.utils.data.random-split.
Pre-Processing: We normalize all the images with mean and standard deviation values of ImageNet [16] dataset. We did not use any data augmentation techniques.
Baselines: We use the open-source implementations for networks DeepLabV3+ (ResNet-101) [32], DeepLabV3 (DenseNet-161) [77], HRNet-V2 [79], and U-Net [51]. We import DeeplabV3+ with encoder networks such as ResNet [25], MobileNet-V2 [71], ResNext [92], EfficientNet [84], and RegNet [63] from the segmentation models library [94].

4.2 Experiments on Cityscapes

The Cityscapes [15] semantic segmentation dataset consists of 5, 000 finely annotated high-quality images, which are further divided into $2,975/500/1,525$ images for training, validation, and testing. During the evaluation, only 19 classes are considered

[1] For all the baselines, we use the Nesterov momentum of 0.9 for the SGD [66] optimizer by following [12, 25, 28, 63, 71, 84]. For the IkshanaNet and its variants, we use the Nesterov momentum of 0.7 for the SGD [66] optimizer by tuning with several values such as 0.5, 0.6, 0.7, 0.8, and 0.9, i.e., the only hyper-parameter tuning step in this work. In our preliminary experiments, we observe that the training of IkshanaNet is unstable with 0.9 momentum. We hypothesize that this phenomenon is due to the small size of IkshanaNet compared to baseline networks.

out of the 35 classes. Therefore, by using the cityscapes-scripts, we convert the 35 classes into 20 classes (including background). We resize all the images from the resolution of 1024×2048 to 512×1024.

4.2.1 Baseline Experiments

Here, we use the networks DeeplabV3+ (ResNet-101 [25]), DeeplabV3 (DenseNet-161 [28]), HRNet-V2 [80], and U-Net [67] as the baselines [2] to compare with IkshanaNet-main.

We train all the networks on the entire dataset T_{2975} and provide the mean class IoU results evaluated on the validation-set in Table 1, where we observe the following:

(i) U-Net [67] (49.3) shown top performance within the baseline networks followed by HRNet-V2 [80] (48.0).
(ii) IkshanaNet outperformed U-Net by 5.2 % and HRNet-V2 [80] by 6.5 %.
(iii) IkshanaNet outperformed baselines by a huge margin in classes such as fence, pole, traffic light, traffic sign, rider, bus, motorcycle, and bicycle.
(iv) Even though U-Net [67] and IkshanaNet learn representations at full-scale resolution before reducing the spatial resolution, the IkshanaNet still outperforms U-Net [67] in the formerly mentioned classes.

4.2.2 Data Ablation Study

While trained on few data samples, the network size might strongly influence the performance. The networks ResNet-101 [25] (59.3 M), DenseNet-161 [28]) (43.2 M), HRNet-V2 [80] (65.9), and U-Net [67] (31.0) consists more number of parameters compared to IkshanaNet-main (4 M). To make it a fair comparison, we include DeeplabV3+ [12] with several light-weight encoder networks (such as ResNet-18 [25], MobileNet-V2 [71], EfficientNet-b1 [84], and RegNetY-08 [63]) along with the networks from the baseline experiments.

Here, we conduct a data ablation study on five different subsets of the training data, T_{1487}, T_{743}, T_{371}, T_{185}, and T_{92} (suffix number represents the number of training samples in the subset) by using the same validation set (500 images).

In Table 2, we provide the mean class IoU results evaluated on the validation set, the average M.IoU score, the number of parameters (in million)), and the GFLOPs [76] (calculated with an input resolution of 1x512x1024x3).

From Table 2, we observe the following:

[2] For the baselines, ResNet-101 [25], DenseNet-161 [28], and HRNet-V2 [80], we use the ImageNet [16] pre-trained weights. Because in the existing literature, the architectures [12, 80, 98, 107] used an ImageNet pertained network as a feature extractor and reported the results by using pre-trained weights only. However, in the case of IkshanaNet and U-Net [67] no pre-training is done. Since this work addresses the requirement of massive data, this provides strong motivation against pre-training.

Table 1 Class-wise IoU results of the Cityscapes baseline experiments

Method	Road	Sidewalk	Building	Wall	Fence	Pole	Traffic light	Traffic sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle	Average
ResNet101 [25]	95.0	66.1	81.9	15.0	13.5	26.7	20.7	29.5	86.7	55.4	89.3	48.5	6.3	85.5	6.8	26.1	19.0	9.8	32.0	42.8
DenseNet161 [28]	94.8	64.5	81.3	20.1	13.0	15.8	15.6	28.7	84.6	**58.7**	86.1	44.1	0.6	84.7	17.0	19.7	**23.1**	4.3	31.4	41.5
HRNet-V2 [80]	94.9	68.6	84.2	24.0	24.5	39.0	23.2	42.3	86.9	51.5	90.2	55.6	15.3	86.1	**19.9**	36.1	21.2	2.2	46.1	48.0
U-Net [67]	94.9	69.4	85.3	**27.3**	28.7	41.0	32.2	49.0	88.6	46.3	90.4	59.1	14.5	86.5	12.4	28.4	15.5	10.9	55.6	49.3
IkshanaNet-Main	**95.6**	**72.8**	**85.9**	22.6	**35.3**	**49.6**	**47.0**	**60.7**	**89.2**	48.9	**91.6**	**63.3**	**28.8**	**87.1**	18.4	**40.3**	21.8	**16.5**	**60.8**	**54.5**

Table 2 Cityscapes data ablation experiments evaluated on the validation set

Backbone	T_{1487}	T_{743}	T_{371}	T_{185}	T_{92}	T_{avg}	Param(M)	GFLOPs
ResNet-18 [25]	42.6	35.6	27.9	22.4	21.0	29.9	12.3	36.8
MobileNet-V2 [71]	38.5	32.2	30.6	22.5	19.2	28.6	4.4	12.3
EfficientNet-b1 [84]	37.8	32.5	26.9	24.6	19.8	28.3	7.4	**4.6**
RegNetY-08 [63]	28.5	31.9	29.4	27.4	22.1	27.9	7.0	17.2
ResNet-101 [25]	29.3	28.8	28.6	21.6	19.4	25.5	59.3	177.8
DenseNet-161 [28]	33.3	30.1	26.0	24.9	20.8	27.0	43.2	129.4
HRNet-V2 [80]	27.8	18.8	23.3	18.3	15.4	20.7	65.9	187.8
U-Net[67]	42.8	34.2	30.2	27.8	25.0	32.0	31.0	387.1
IkshanaNet-Main	**43.4**	**40.2**	**31.7**	**29.9**	**25.8**	**34.2**	**4.0**	413.3

(i) U-Net [67] (T_{avg}–32.2) achieves top average performance within the baselines.
(ii) Even though U-Net [67] consists of 31M parameters, it still managed to outperform its lightweight counterparts.
(iii) IkshanaNet outperformed all other baselines in the M.IoU score and the average M.IoU score in all five subsets.
(iv) IkshanaNet consists of fewer parameters, and EfficientNet-b1 [84] consists of fewer GFLOPs than other networks.

4.2.3 Multi-scale Ablation Study

In Sect. 3.1, the Ikshana hypothesis stated that "humans often require multi-scale information to understand the gist of an image". Therefore, to verify the requirement of multi-scale information, we conduct a multi-scale ablation study.

Here, we train three different variants of IkshanaNet, such as the 1S-6G, 2S-3G, and 3S-2G (explained in Sect. 3.2) on the five different subsets of the training data (same as Sect. 4.2.2). In Table 3, we provide the results of the multi-scale ablation study evaluated on the validation set.

From Table 3, we observe that:

(i) IkshanaNet-3S-2G network outperforms other networks in the M.IoU score, the average M.IoU score, and requires fewer GFLOPs, while requiring the same number of parameters.

Table 3 Cityscapes multi-scale ablation experiments results

Backbone	T_{1487}	T_{743}	T_{371}	T_{185}	T_{92}	T_{avg}	Param(M)	GFLOPs
1S-6Glances	29.2	24.9	23.3	20.2	18.1	23.1	0.26	136.0
2S-3Glances	37.3	34.9	33.2	25.7	24.0	31.0	0.26	70.0
3S-2Glances	**43.5**	**36.9**	**34.4**	**27.5**	**26.5**	**33.8**	0.26	**42.4**

(ii) The multi-scale information improved the performance and decreased the computational complexity (GFLOPs) of the network and vice-versa.
(iii) From Tables 2 and 3, we observe that IkshanaNet 3S-2G network (with only 260K parameters) outperforms all the baselines in the data ablation study by occupying approximately 10x few GFLOPs and 15x few parameters than IkshanaNet-main.

The above observations suggest that, the multi-scale architectures can achieve superior performance than an isometric architecture.

4.3 Experiments on Camvid

The Cambridge-driving labeled video dataset [5] for semantic segmentation consists of 700 images, which are further divided into 367 training, 101 validation, and 233 testing sets. We convert the 32 classes to 12 classes (including background) by following [2, 34] and resize the images from the resolution of 720 $\times$ 960 to 368 $\times$ 480.

4.3.1 Baseline Experiments

Here, according to the size of the networks, we classify the total networks into three different sets.

Set-1 consists of DeeplabV3+ [12] with the encoder networks such as Resnet-18 [25], EfficientNet-b1 [84], RegNetY-08 [63], MobileNet-V2 [71], and IkshanaNet-3G (see Sect. 3.2).

Set-2 consists of DeeplabV3+ [12] with the encoder networks such as Resnet-50 [25], EfficientNet-b4 [84], RegNetY-40 [63], and ResNext-50 [92], and IkshanaNet-6G (see Sect. 3.2).

Table 4 Camvid baseline experiments results

Backbone	T_{367}		T_{183}		T_{91}		T_{avg}		Param(M)	GFLOPs
	Val	Test	Val	Test	Val	Test	Val	Test		
ResNet-18 [25]	83.3	64.9	79.7	**63.7**	70.0	56.6	77.7	61.7	12.3	12.4
EfficientNet-b1 [84]	**84.4**	**68.4**	75.0	61.3	77.0	58.8	78.8	62.8	7.4	**1.5**
RegNetY-08 [63]	80.4	64.3	77.7	61.4	70.9	57.8	76.3	61.2	7.0	5.8
MobileNet-V2 [71]	80.8	63.9	77.3	56.1	66.1	54.6	74.7	58.2	4.4	4.1
IkshanaNet-3G	81.6	65.7	**80.0**	62.5	**78.0**	**61.2**	**79.9**	**63.1**	**0.5**	26.0
ResNet-50 [25]	78.6	61.6	80.0	60.3	**78.3**	55.9	80.0	59.3	26.7	25.0
EfficientNet-b4 [84]	82.7	64.1	77.7	62.2	75.6	**60.5**	78.7	62.3	18.6	**1.7**
RegNetY-40 [63]	80.8	62.0	76.4	61.0	74.9	59.2	77.4	60.7	21.5	18.8
ResNext-50 [92]	80.1	62.6	77.3	56.1	66.1	54.6	74.5	57.8	26.2	25.0
IkshanaNet-6G	**83.3**	**67.8**	**81.4**	**65.9**	76.0	60.0	**80.2**	**64.6**	**1.8**	82.0
ResNet-101 [25]	81.6	63.8	75.6	56.4	70.1	55.7	75.8	58.6	59.3	59.9
EfficientNet-b6 [84]	80.6	65.0	80.3	57.8	**77.4**	60.4	79.4	61.0	42.0	**1.9**
RegNetY-80 [63]	78.5	62.0	78.2	63.8	66.2	53.8	74.3	59.9	40.3	34.4
DenseNet-161 [28]	77.8	58.6	75.7	57.8	73.0	53.8	75.5	56.7	43.2	43.6
HRNet-V2 [80]	81.1	63.6	79.1	62.9	72.9	55.0	77.7	60.5	65.9	63.5
U-Net [67]	83.0	69.5	78.0	62.8	76.8	**61.6**	79.3	64.6	31.0	130.0
IkshanaNet-12G	**83.9**	**70.0**	**83.3**	**67.1**	76.5	60.6	**81.2**	**65.9**	**6.5**	285.0
IkshanaNet-M	83.2	68.5	79.9	62.9	72.2	58.8	78.4	63.4	4.0	139.0

Set-3 consists of DeeplabV3+ [12] with the encoder networks such as Resnet-101 [25], EfficientNet-b6 [84], RegNetY-80 [63], DeepLabV3 (DenseNet-161 [28]), HRNet-V2 [80], U-Net [67], and IkshanaNet-12G (see Sect. 3.2) [3].

Additionally, we include IkshanaNet-main and did not compare it with other networks. By using the same validation, we train each network on three different subsets of the training data, T_{367}, T_{183}, and T_{91}.

In Table 4, we provide the mean IoU results evaluated on the validation set, the test set, the average M.IoU score of all the variants, the number of parameters (in Million), and the GFLOPs [76] (calculated the GFLOPs with an input resolution of $1 \times 368 \times 480 \times 3$). From Table 4, we observe the following:

In **Set-**1: (i) IkshanaNet-3G outperforms all other networks in the subsets T_{91}, T_{avg}, and requires fewer parameters. (ii) EfficientNet-b1 [84] outperforms other networks in the T_{367} and requires fewer GFLOPs.
In **Set-**2: (i) IkshanaNet-6G outperforms all other networks in the subsets T_{367}, T_{183}, T_{avg}, and requires fewer parameters.

[3] Same as Sect. 4.2.1, except for U-Net [67] and IkshanaNet-12G, we use the ImageNet [16] pre-trained weights for all the networks in the Set-3.

Table 5 Camvid multi-scale ablation experiments results

Backbone	T_{367}		T_{183}		T_{91}		T_{avg}		Param(M)	GFLOPs
	Val	Test	Val	Test	Val	Test	Val	Test		
1S-6Glances	79.2	60.0	77.8	58.8	66.7	50.9	74.6	56.6	0.26	45.6
2S-3Glances	80.1	65.6	79.5	60.1	77.2	59.5	78.9	61.7	0.26	23.1
3S-2Glances	**82.9**	**66.5**	**80.9**	**62.8**	**77.5**	**60.8**	**80.4**	**63.4**	0.26	**14.0**

(ii) EfficientNet-b4 [84] outperforms all other networks in the subset T_{91} and requires fewer GFLOPs.
In **Set-**3: (i) IkshanaNet-12G outperforms all other networks in the subsets T_{367}, T_{183}, T_{avg}, and requires fewer parameters.
(ii) U-Net [67] outperformed other networks in the subset T_{91} and EfficientNet-b6 [84] requires fewer GFLOPs than other networks.

4.3.2 Multi-scale Ablation Study

Same as Sect. 4.2.3, by using the same validation set, we train three different variants of IkshanaNet such as 1S-6G, 2S-3G, and 3S-2G (explained in Sect. 3.2) on three subsets of the training data (T_{367}, T_{183}, and T_{91}).

In Table 5, we provide the mean IoU results evaluated on the validation set, the test set, the average score of all variants, the parameters, and the GFLOPs. We calculate the GFLOPs with an input resolution of 1x368x480x3.

From Table 5, we observe that, the IkshanaNet-3S-2G network outperforms all other networks in all the subsets (T_{367}, T_{183}, T_{91}, T_{avg}), and requires fewer GFLOPs. The results are similar to the Sect. 4.2.3 (Table 3), demonstrating the importance of multi-scale information.

5 Validity Threats

(i) Most of the existing works [12, 98, 107] used a mini-batch size of 8 and SyncBN [69, 104] for training. However, due to the limited availability of the computing resources, we train all the networks with a mini-batch size of 2. Due to this reason, we cannot directly compare the performance of our method with the state-of-the-art methods.
(ii) In this work, even though the training data splits are reproducible, the performance of the networks trained on subsets of the training data might depend upon the fact that "how well the subset represents the whole dataset?". If we use a

different random seed to generate the splits, then the exact behavior may or may not be expected.

(iii) In this work, through multi-scale ablation experiments, we observe that multi-scale information is often necessary to improve the performance of the networks. By observing the images in Cityscapes, and CamVid datasets, it is evident that the images consist of multi-scale objects. However, this phenomenon might not be valid to other datasets, where there exist no multi-scale objects.

6 Conclusion

In this work, we attempt to bridge the gap between the current vision DNNs and the human visual system by proposing a novel hypothesis of human scene understanding and a neural-inspired CNN architecture that learns representations at full-scale resolution.

The empirical results illustrate the effectiveness of our method on entire and few data samples compared to the baselines. Also, through multi-scale ablation studies, we observe that using multi-scale information improves the performance of IkshanaNet by reducing the computational complexity.

Moreover, we observe that our method is just an improvement over the baselines, and it is still dependent on the data. Hence, it is nowhere close to the human visual system. Therefore, a better-performing and computationally efficient architectures based on the Ikshana hypothesis will be studied in the future work.

Furthermore, we hope that our hypothesis inspires future generation of neural inspired vision architectures.

6.1 Code

https://github.com/dvssajay/The-Ikshana-Hypothesis-of-Human-Scene-Understanding.

Acknowledgements I would like to thank all the anonymous reviewers and area chairs for their constructive feedback. I would like to thank Shri Daliparthi Sathi Raju Garu for providing computational resources to conduct the main experiments of this work. Also, I would like to thank Mr. Florian Westphal for proving GPUs to conduct the preliminary experiments of this work.

References

1. Amirul Islam M, Rochan M, Bruce NDB, Wang Y (2017) Gated feedback refinement network for dense image labeling. In: Computer vision and pattern recognition (CVPR). https://doi.org/10.1109/CVPR.2017.518
2. Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39(12):2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615
3. Bertasius G, Shi J, Torresani L (2015) High-for-low and low-for-high: efficient boundary detection from deep object features and its applications to high-level vision. In: Proceedings of the IEEE international conference on computer vision (ICCV). https://doi.org/10.1109/ICCV.2015.65
4. Biederman I (1987) Recognition-by-components: a theory of human image understanding. Psychol Rev 94(2):115. https://doi.org/10.1037/0033-295X.94.2.115
5. Brostow GJ, Shotton J, Fauqueur J, Cipolla R (2008) Segmentation and recognition using structure from motion point clouds. In: ECCV (1), pp 44–57. https://doi.org/10.1007/978-3-540-88682-2_5
6. Cao Z, Zhang T, Diao W, Zhang Y, Lyu X, Fu K, Sun X (2019) Meta-seg: a generalized meta-learning framework for multi-class few-shot semantic segmentation. IEEE Access 7:166109–166121. https://doi.org/10.1109/ACCESS.2019.2953465
7. Chen LC, Barron JT, Papandreou G, Murphy K, Yuille AL (2016) Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In: CVPR. https://doi.org/10.1109/CVPR.2016.492
8. Chen L, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2015) Semantic image segmentation with deep convolutional nets and fully connected crfs. In: 3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, conference track proceedings. arXiv:1412.7062
9. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2018) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans Pattern Anal Mach Intell 40(4):834–848. https://doi.org/10.1109/TPAMI.2017.2699184
10. Chen L, Papandreou G, Schroff F, Adam H (2017) Rethinking atrous convolution for semantic image segmentation. arXiv:1706.05587
11. Chen LC, Yang Y, Wang J, Xu W, Yuille AL (2016) Attention to scale: Scale-aware semantic image segmentation. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 3640–3649. https://doi.org/10.1109/CVPR.2016.396
12. Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV), pp 801–818 (2018). https://doi.org/10.1007/978-3-030-01234-2_49
13. Choi S, Kim JT, Choo J (2020) Cars can't fly up in the sky: improving urban-scene segmentation via height-driven attention networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9373–9383. https://doi.org/10.1109/CVPR42600.2020.00939
14. Chollet F (2017) Xception: seep learning with depthwise separable convolutions. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 1800–1807. https://doi.org/10.1109/CVPR.2017.195
15. Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B (2016) The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3213–3223. https://doi.org/10.1109/CVPR.2016.350
16. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp 248–255. IEEE. https://doi.org/10.1109/cvprw.2009.5206848

17. Ding H, Jiang X, Liu AQ, Thalmann NM, Wang G (2019) Boundary-aware feature propagation for scene segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV). https://doi.org/10.1109/ICCV.2019.00692
18. Dong N, Xing EP (2018) Few-shot semantic segmentation with prototype learning. In: BMVC, vol 3
19. Evans KK, Treisman A (2005) Perception of objects in natural scenes: is it really attention free? J Exp Psychol Human Percept Perform 31(6):1476. https://doi.org/10.1037/0096-1523.31.6.1476
20. Fei-Fei L, Iyer A, Koch C, Perona P (2007) What do we perceive in a glance of a real-world scene? J Vision 7(1):10–10. https://doi.org/10.1167/7.1.10
21. Friedman A (1979) Framing pictures: The role of knowledge in automatized encoding and memory for gist. J Exp Psychol Gen 108(3):316. https://doi.org/10.1037//0096-3445.108.3.316
22. Fu J, Liu J, Tian H, Li Y, Bao Y, Fang Z, Lu H (2019) Dual attention network for scene segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3146–3154. https://doi.org/10.1109/CVPR.2019.00326
23. Fukushima K, Miyake S, Ito T (1983) Neocognitron: a neural network model for a mechanism of visual pattern recognition. IEEE Trans Syst Man Cybern 45(5):826–834. https://doi.org/10.1109/TSMC.1983.6313076
24. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969. https://doi.org/10.1109/ICCV.2017.322
25. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778. https://doi.org/10.1109/CVPR.2016.90
26. Henderson JM (2003) Human gaze control during real-world scene perception. Trends Cognit Sci 7(11):498–504. https://doi.org/10.1016/j.tics.2003.09.006
27. Huang G, Chen D, Li T, Wu F, van der Maaten L, Weinberger K (2018) Multi-scale dense networks for resource efficient image classification. In: International conference on learning representations. https://openreview.net/forum?id=Hk2aImxAb
28. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708. https://doi.org/10.1109/CVPR.2017.243
29. Hubel DH, Wiesel TN (1962) Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J Physiol 160(1):106. https://doi.org/10.1113/jphysiol.1962.sp006837
30. Intraub H (1981) Rapid conceptual identification of sequentially presented pictures. J Exp Psychol Human Percept Perform 7(3):604. https://doi.org/10.1037/0096-1523.7.3.604
31. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, pp 448–456. PMLR
32. jfzhang95: pytorch-deeplab-xception (deeplabv3+ with resnet-101 backbone). https://github.com/jfzhang95/pytorch-deeplab-xception
33. Jégou S, Drozdzal M, Vazquez D, Romero A, Bengio Y (2017) The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In: 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW), pp 1175–1183. https://doi.org/10.1109/CVPRW.2017.156
34. Kendall A: Segnet-tutorial. https://github.com/alexgkendall/SegNet-Tutorial
35. Krizhevsky A, Sutskever I: E. hinton, geoffrey (2012) imagenet classification with deep convolutional neural networks. Neural Inf Process Syst 25(10.1145), 3065386. https://doi.org/10.5555/2999134.2999257
36. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324. https://doi.org/10.1109/5.726791

37. Li G, Jampani V, Sevilla-Lara L, Sun D, Kim J, Kim J (2021) Adaptive prototype learning and allocation for few-shot segmentation. CoRR arXiv:2104.01893
38. Li X, Zhang L, You A, Yang M, Yang K, Tong Y (2019) Global aggregation then local distribution in fully convolutional networks. In: BMVC
39. Liao Q, Poggio T (2016) Bridging the gaps between residual learning, recurrent neural networks and visual cortex. arXiv preprint arXiv:1604.03640
40. Lin D, Shen D, Shen S, Ji Y, Lischinski D, Cohen-Or D, Huang H (2019) Zigzagnet: fusing top-down and bottom-up context for object segmentation. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 7482–7491. https://doi.org/10.1109/CVPR.2019.00767
41. Lin G, Milan A, Shen C, Reid I (2017) RefineNet: multi-path refinement networks for high-resolution semantic segmentation. In: CVPR. https://doi.org/10.1109/CVPR.2017.549
42. Lin G, Liu F, Milan A, Shen C, Reid I (2019) Refinenet: multi-path refinement networks for dense prediction. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/TPAMI.2019.2893630
43. Lin M, Chen Q, Yan S (2014) Network in network. In: 2nd international conference on learning representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, conference track proceedings. arXiv:1312.4400
44. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2117–2125. https://doi.org/10.1109/CVPR.2017.106
45. Liu L, Cao J, Liu M, Guo Y, Chen Q, Tan M (2020) Dynamic extension nets for few-shot semantic segmentation. In: Proceedings of the 28th ACM international conference on multimedia, pp 1441–1449. MM '20, Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3394171.3413915
46. Liu S, Qi L, Qin H, Shi J, Jia J (2018) Path aggregation network for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8759–8768. https://doi.org/10.1109/CVPR.2018.00913
47. Liu W, Rabinovich A, Berg AC (2015) Parsenet: looking wider to see better. CoRR arXiv:1506.04579
48. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440. https://doi.org/10.1109/TPAMI.2016.2572683
49. Marin D, He Z, Vajda P, Chatterjee P, Tsai SS, Yang F, Boykov Y (2019) Efficient segmentation: learning downsampling near semantic boundaries. In: 2019 IEEE/CVF international conference on computer vision, ICCV 2019, Seoul, Korea (South), October 27–November 2, 2019, pp 2131–2141. IEEE. https://doi.org/10.1109/ICCV.2019.00222
50. McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bullet Math Biophys 5(4):115–133. https://doi.org/10.1007/BF02478259
51. milesial: Pytorch-unet. https://github.com/milesial/Pytorch-UNet
52. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: Icml
53. Nesterov YE (1983) A method for solving the convex programming problem with convergence rate o $(1/\hat{k}2)$. In: Dokl. akad. nauk Sssr. vol 269, pp 543–547
54. Noh H, Hong S, Han B (2015) Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE international conference on computer vision, pp 1520–1528. https://doi.org/10.1109/ICCV.2015.178
55. Oliva A (2005) Gist of the scene. In: Neurobiology of attention, pp 251–256. Elsevier. https://doi.org/10.1016/B978-012375731-9/50045-8
56. Oliva A, Schyns PG (1997) Coarse blobs or fine edges? evidence that information diagnosticity changes the perception of complex visual stimuli. Cognit Psychol 34(1):72–107. https://doi.org/10.1006/cogp.1997.0667
57. Oliva A, Schyns PG (2000) Diagnostic colors mediate scene recognition. Cognit Psychol 41(2):176–210. https://doi.org/10.1006/cogp.1999.0728

58. Oliva A, Torralba A (2001) Modeling the shape of the scene: a holistic representation of the spatial envelope. Int J Comput Vis 42(3):145–175. https://doi.org/10.1023/A:1011139631724
59. Pambala AK, Dutta T, Biswas S (2020) SML: semantic meta-learning for few-shot semantic segmentation. CoRR arXiv:2009.06680
60. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S (2019) Pytorch: an imperative style, high-performance deep learning library. In: Advances in neural information processing systems 32, pp 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
61. Potter MC (1975) Meaning in visual search. Science 187(4180):965–966. https://doi.org/10.1126/science.1145183
62. Potter MC (1976) Short-term conceptual memory for pictures. J Expe Psychol Human Learn Memory 2(5):509. https://doi.org/10.1037/0278-7393.2.5.509
63. Radosavovic I, Kosaraju RP, Girshick R, He K, Dollár P (2020) Designing network design spaces. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10428–10436. https://doi.org/10.1109/cvpr42600.2020.01044
64. Rakelly K, Shelhamer E, Darrell T, Efros AA, Levine S (2018) Conditional networks for few-shot semantic segmentation. In: 6th International Conference on Learning Representations, ICLR, Vancouver, BC, Canada, Workshop Track Proceedings. https://openreview.net/forum?id=SkMjFKJwG
65. Rayner K (1998) Eye movements in reading and information processing: 20 years of research. Psychol Bulletin 124(3):372. https://doi.org/10.1037/0033-2909.124.3.372
66. Robbins H, Monro S (1951) A stochastic approximation method. The annals of mathematical statistics, pp 400–407. https://doi.org/10.1214/aoms/1177729586
67. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, pp 234–241. Springer. https://doi.org/10.1007/978-3-319-24574-4_28
68. Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 65(6):386. https://doi.org/10.1037/H0042519
69. Rota Bulò S, Porzi L, Kontschieder P (2018) In-place activated batchnorm for memory-optimized training of dnns. In: Proceedings of the IEEE conference on computer vision and pattern recognition. https://doi.org/10.1109/CVPR.2018.00591
70. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323(6088):533–536. https://doi.org/10.1038/323533a0
71. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) Mobilenetv2: inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, pp 4510–4520. https://doi.org/10.1109/CVPR.2018.00474
72. Schyns PG, Oliva A (1994) From blobs to boundary edges: evidence for time-and spatial-scale-dependent scene recognition. Psychol Sci 5(4):195–200. https://doi.org/10.1111/j.1467-9280.1994.tb00500.x
73. Shaban A, Bansal S, Liu Z, Essa I, Boots B (2017) One-shot learning for semantic segmentation. Proceedings of the British machine vision conference (BMVC), pp 167.1–167.13. https://doi.org/10.5244/C.31.167
74. Siddique N, Paheding S, Elkin CP, Devabhaktuni V (2021) U-net and its variants for medical image segmentation: a review of theory and applications. IEEE Access 9:82031–82057. https://doi.org/10.1109/ACCESS.2021.3086020
75. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: 3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings. arXiv:1409.1556
76. Sovrasov V: Fops-counter.pytorch. https://github.com/sovrasov/flops-counter.pytorch
77. stigma0617: Vovnet-deeplabv3 (deeplabv3 with densenet161 backbone). https://github.com/stigma0617/VoVNet-DeepLabV3

78. Sun K, Xiao B, Liu D, Wang J (2019) Deep high-resolution representation learning for human pose estimation. In: CVPR. https://doi.org/10.1109/CVPR.2019.00584
79. Sun K, Zhao Y, Jiang B, Cheng T, Xiao B, Liu D, Mu Y, Wang X, Liu W, Wang J, Hrnet-semantic-segmentation. https://github.com/HRNet/HRNet-Semantic-Segmentation
80. Sun K, Zhao Y, Jiang B, Cheng T, Xiao B, Liu D, Mu Y, Wang X, Liu W, Wang J (2019) High-resolution representations for labeling pixels and regions. CoRR arXiv:1904.04514
81. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR), pp 1–9. https://doi.org/10.1109/CVPR.2015.7298594
82. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of IEEE conference on computer vision and pattern recognition. https://doi.org/10.1109/CVPR.2016.308
83. Takikawa T, Acuna D, Jampani V, Fidler S (2019) Gated-scnn: Gated shape cnns for semantic segmentation. ICCV. https://doi.org/10.1109/ICCV.2019.00533
84. Tan M, Le QV (2019) Efficientnet: rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th international conference on machine learning, ICML 2019, 9–15 June 2019, Long Beach, California, USA. Proceedings of Machine Learning Research, vol 97, pp 6105–6114. PMLR. http://proceedings.mlr.press/v97/tan19a.html
85. Tian P, Wu Z, Qi L, Wang L, Shi Y, Gao Y (2020) Differentiable meta-learning model for few-shot semantic segmentation. In: The Thirty-Fourth AAAI conference on artificial intelligence, AAAI, pp 12087–12094. AAAI Press. https://aaai.org/ojs/index.php/AAAI/article/view/6887
86. Tian Z, Lai X, Jiang L, Shu M, Zhao H, Jia J (2020) Generalized few-shot semantic segmentation. CoRR arXiv:2010.05210
87. Tian Z, Zhao H, Shu M, Yang Z, Li R, Jia J (2020) Prior guided feature enrichment network for few-shot segmentation. IEEE transactions on pattern analysis and machine intelligence, pp 1–1. https://doi.org/10.1109/TPAMI.2020.3013717
88. Waibel A, Hanazawa T, Hinton G, Shikano K, Lang K (1989) Phoneme recognition using time-delay neural networks. IEEE Trans Acoust Speech Signal Process 37(3):328–339. https://doi.org/10.1109/29.21701
89. Wang H, Zhang X, Hu Y, Yang Y, Cao X, Zhen X (2020) Few-shot semantic segmentation with democratic attention networks. In: Computer Vision - ECCV 2020—16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII. Lecture Notes in Computer Science, vol 12358, pp 730–746. Springer. https://doi.org/10.1007/978-3-030-58601-0_43
90. Wang K, Liew JH, Zou Y, Zhou D, Feng J (2019) Panet: few-shot image semantic segmentation with prototype alignment. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV) (2019). https://doi.org/10.1109/ICCV.2019.00929
91. Wang X, Girshick R, Gupta A, He K (2018) Non-local neural networks. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, pp 7794–7803. https://doi.org/10.1109/CVPR.2018.00813
92. Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 5987–5995. https://doi.org/10.1109/CVPR.2017.634
93. Xu N, Price B, Cohen S, Yang J, Huang TS (2016) Deep interactive object selection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 373–381. https://doi.org/10.1109/CVPR.2016.47
94. Yakubovskiy P (2020) Segmentation models pytorch. https://github.com/qubvel/segmentation-models.pytorch
95. Yang B, Liu C, Li B, Jiao J, Ye Q (2020) Prototype mixture models for few-shot semantic segmentation. In: Computer Vision—ECCV 2020—16th European conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII. Lecture Notes in Computer Science, vol. 12353, pp. 763–778. Springer. https://doi.org/10.1007/978-3-030-58598-3_45

96. Yang M, Yu K, Zhang C, Li Z, Yang K (2018) Denseaspp for semantic segmentation in street scenes. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, pp 3684–3692. https://doi.org/10.1109/CVPR.2018.00388
97. Yang X, Wang B, Zhou X, Chen K, Yi S, Ouyang W, Zhou L (2020) Brinet: towards bridging the intra-class and inter-class gaps in one-shot segmentation. In: 31st British Machine Vision Conference 2020, BMVC 2020, Virtual Event, UK, September 7–10, 2020. BMVA Press. https://www.bmvc2020-conference.com/assets/papers/0139.pdf
98. Yuan Y, Chen X, Wang J (2020) Object-contextual representations for semantic segmentation. In: Vedaldi A, Bischof H, Brox T, Frahm JM (eds) Computer vision—ECCV 2020, pp 173–190. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-58539-6_11
99. Yuan Y, Wang J (2018) Ocnet: object context network for scene parsing. CoRR arXiv:1809.00916
100. Yuan Y, Xie J, Chen X, Wang J (2020) Segfix: model-agnostic boundary refinement for segmentation. In: European conference on computer vision, pp 489–506. Springer. https://doi.org/10.1007/978-3-030-58610-2_29
101. Zhang C, Lin G, Liu F, Guo J, Wu Q, Yao R (2019) Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV) (October 2019). https://doi.org/10.1109/ICCV.2019.00968
102. Zhang C, Lin G, Liu F, Yao R, Shen C (2019) Canet: class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) (June 2019). https://doi.org/10.1109/CVPR.2019.00536
103. Zhang F, Chen Y, Li Z, Hong Z, Liu J, Ma F, Han J, Ding E (2019) Acfnet: attentional class feature network for semantic segmentation. In: 2019 IEEE/CVF international conference on computer vision (ICCV), pp 6797–6806. https://doi.org/10.1109/ICCV.2019.00690
104. Zhang H, Dana K, Shi J, Zhang Z, Wang X, Tyagi A, Agrawal A (2018) Context encoding for semantic segmentation. In: The IEEE conference on computer vision and pattern recognition (CVPR) (June 2018). https://doi.org/10.1109/CVPR.2018.00747
105. Zhang H, Zhang H, Wang C, Xie J (2019) Co-occurrent features in semantic segmentation. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 548–557. https://doi.org/10.1109/CVPR.2019.00064
106. Zhang X, Wei Y, Yang Y, Huang TS (2020) Sg-one: similarity guidance network for one-shot semantic segmentation. IEEE Trans Cybern 50(9):3855–3865. https://doi.org/10.1109/TCYB.2020.2992433
107. Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2881–2890. https://doi.org/10.1109/CVPR.2017.660
108. Zhou B, Zhao H, Puig X, Fidler S, Barriuso A, Torralba A (2017) Scene parsing through ade20k dataset. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 5122–5130. https://doi.org/10.1109/CVPR.2017.544