Efficient High-Resolution Deep Learning: A Survey

ARIAN BAKHTIARNIA, QI ZHANG, and ALEXANDROS IOSIFIDIS, DIGIT, Aarhus University, Aarhus, Denmark

Cameras in modern devices such as smartphones, satellites and medical equipment are capable of capturing very high resolution images and videos. Such high-resolution data often need to be processed by deep learning models for cancer detection, automated road navigation, weather prediction, surveillance, optimizing agricultural processes and many other applications. Using high-resolution images and videos as direct inputs for deep learning models creates many challenges due to their high number of parameters, computation cost, inference latency and GPU memory consumption. Simple approaches such as resizing the images to a lower resolution are common in the literature, however, they typically significantly decrease accuracy. Several works in the literature propose better alternatives in order to deal with the challenges of high-resolution data and improve accuracy and speed while complying with hardware limitations and time restrictions. This survey describes such efficient high-resolution deep learning methods, summarizes real-world applications of high-resolution deep learning, and provides comprehensive information about available high-resolution datasets.

CCS Concepts: • **Computing methodologies** → **Computer vision**; **Neural networks;**

Additional Key Words and Phrases: High-resolution deep learning, efficient deep learning, vision transformer

ACM Reference Format:
Arian Bakhtiarnia, Qi Zhang, and Alexandros Iosifidis. 2024. Efficient High-Resolution Deep Learning: A Survey. *ACM Comput. Surv.* 56, 7, Article 181 (April 2024), 35 pages. https://doi.org/10.1145/3645107

1 INTRODUCTION

Many modern devices such as smartphones, drones, augmented reality headsets, vehicles and other **Internet of Things (IoT)** devices are equipped with high-quality cameras that can capture high-resolution images and videos. With the help of image stitching techniques, camera arrays [126, 157], gigapixel acquisition robots [110] and whole-slide scanners [41], capture resolutions can be increased to billions of pixels (commonly referred to as *gigapixels*), such as the image depicted in Figure 1. One could attempt to define *high-resolution* based on the capabilities of human visual system. However, many deep learning tasks rely on data captured by equipment which behaves very differently compared to the human eye, such as microscopes, satellite imagery and infrared cameras. Furthermore, utilizing more detail than the eye can sense is beneficial in many deep learning tasks, such as in the applications discussed in Section 2. The amount of detail that can be captured and is useful if processed varies greatly from task to task. Therefore, the definition of high-resolution is *task-dependent*. For instance, in image classification and **computed tomography**

This work was funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 957337, and by the Danish Council for Independent Research under Grant No. 9131-00119B..
Authors' address: A. Bakhtiarnia, Q. Zhang, and A. Iosiidis, DIGIT, Aarhus University, 5125 Edison, Finlandsgade 22, Aarhus, Midtjylland, Denmark, 8200; e-mails: arianbakh@ece.au.dk, qz@ece.au.dk, ai@ece.au.dk.

This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2024 Copyright held by the owner/author(s).
ACM 0360-0300/2024/04-ART181
https://doi.org/10.1145/3645107

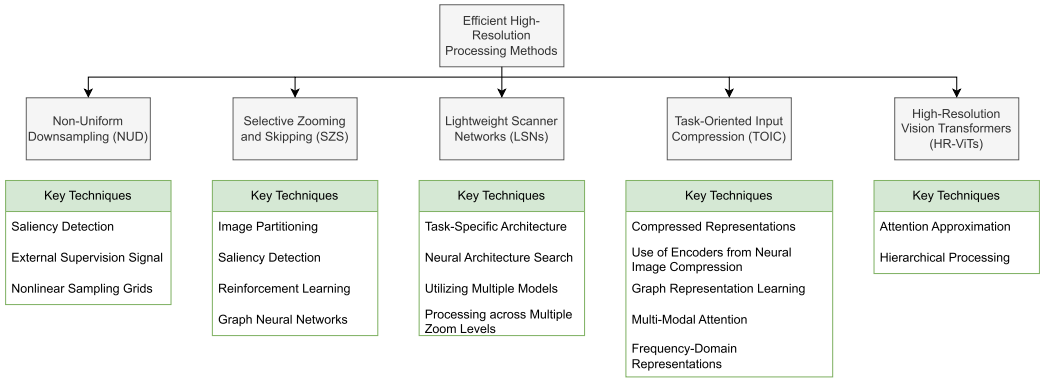

Fig. 1. Example of a gigapixel image, taken from the PANDA-Crowd dataset [144], captured using an array of micro-cameras; (a) original image with a size of 26,558 × 14,828 pixels, and (b) zoomed in to the location specified by the red rectangle in the original image, with a size of 2,516 × 1,347 pixels, which is more than 100 times smaller than the original image, yet still approximately 5 times larger than the image size processed by state-of-the-art deep learning models for crowd counting such as SASNet [116], which is 1024 × 768, and around 50 times larger than the standard image size processed by image classification models, which is 224 × 224.

(CT) scan processing, a resolution of 512 × 512 pixels is considered to be high [17, 37]. In visual crowd counting, datasets with **High-Definition (HD)** resolutions or higher are common [45], and **whole-slide images (WSIs)** in histopathology, which is the study of diseases of the tissues, or remote sensing data, which are captured by aircrafts or satellites, can easily reach gigapixel resolutions [134, 135].

Moreover, with the constant advancement of hardware and methodologies, what deep learning literature considers high-resolution has shifted over time. For instance, in the late 1990s, processing the 32×32-pixel MNIST images with neural networks was an accomplishment [78], whereas in early 2010s, the 256×256-pixel images in ImageNet were considered high-resolution [76]. This trend can also be seen in the consistent increase of the average resolution of images in popular deep learning datasets, such as crowd counting [45] and anomaly detection [101] datasets. Therefore, the definition of high-resolution is also *period-dependent*. Based on the task- and period-dependence properties, it is clear that the term "high-resolution" is technical, not fundamental or universal. Therefore, instead of trying to derive such a definition, we shift our focus to resolutions that create technical challenges in deep learning at the time of this writing.

Using high-resolution images and videos directly as inputs to deep learning models creates challenges during both training and inference phases. With the exception of **fully-convolutional networks (FCNs)**, the number of parameters in deep learning models typically increases with larger input sizes. Moreover, the amount of computation, which is commonly measured in terms of **floating point operations (FLOPs)**, and therefore inference/training time, as well as GPU memory consumption increase with higher-resolution inputs, as shown in Figure 2. This issue is especially problematic in **Vision Transformer (ViT)** architectures, which use the self-attention mechanism, where the inference speed and number of parameters scale quadratically with input size [37, 122]. These issues are exacerbated when the training or inference needs to be done on resource-constrained devices, such as smartphones, that have limited computational capabilities compared to high-end computing equipment, such as workstations or servers.

Even though methods such as *model parallelism* can be used to split the model between multiple GPUs during both the training [113, 146] and inference [39] phases, and thus avoid memory and

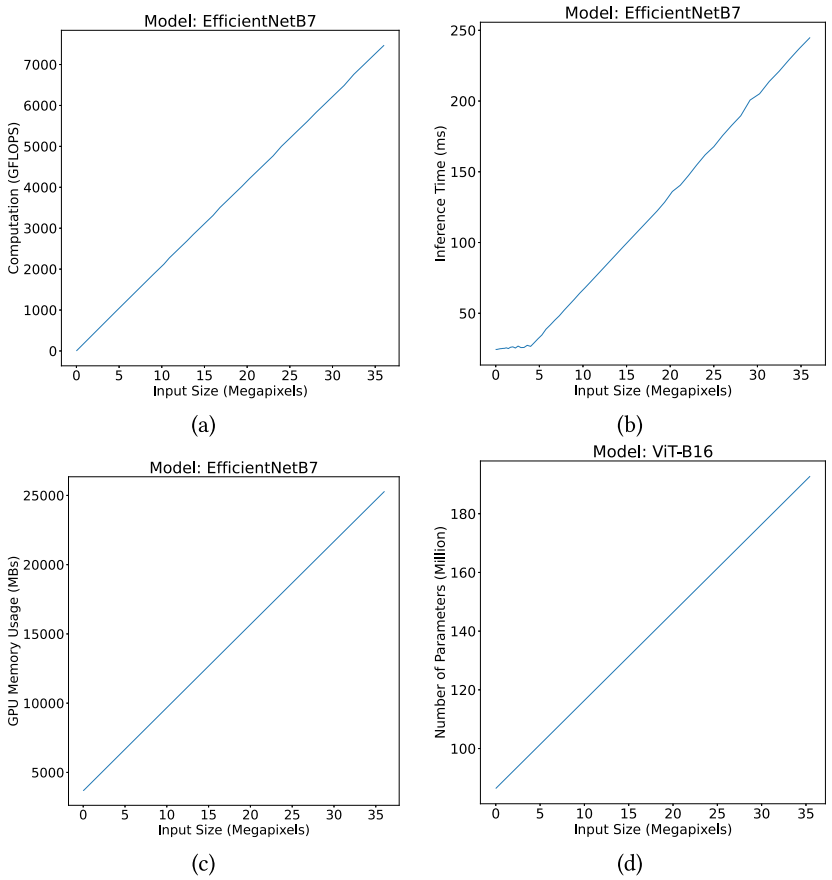

Fig. 2. As the resolution of the input image increases, so does (a) the amount of computation, (b) inference time, (c) GPU memory usage in the EfficientNet-B7 [121], and (d) the number of parameters in the ViT-B16 [37] architecture. The last layer of EfficientNet-B7 was removed to form a fully-convolutional feature extractor. Since accuracy is not considered in these figures, there is no need to use real images, thus randomly generated images are given to the models as input. All experiments were conducted on an Nvidia A6000 GPU.

latency issues, these methods require a large amount of resources, such as a large number of GPUs and servers, which can incur high costs, especially when working with extreme resolutions such as gigapixel images. Furthermore, in many applications, such as self-driving cars and drone image processing, there is a limit for the hardware that can be mounted, and offloading the computation to external servers is not always possible because of unreliability of the network connection due to movement and the time-critical nature of the application. Therefore, the most common approach for deep learning training and inference is to load the full model on each single GPU instance. Multi-GPU setups are instead typically used to speed up the training by increasing the overall batch size, to test multiple sets of hyper-parameters in parallel or to distribute the inference load. Consequently, in many cases, there is an effective maximum resolution that can be processed by deep learning models. As an example, the maximum resolution for inference using SASNet [116], which is the state-of-the-art model for crowd counting on the Shanghai Tech dataset [162] at the time of this writing, is around 1024×768 (less than HD) on Nvidia 2080 Ti GPUs which have 11 GBs of video memory.

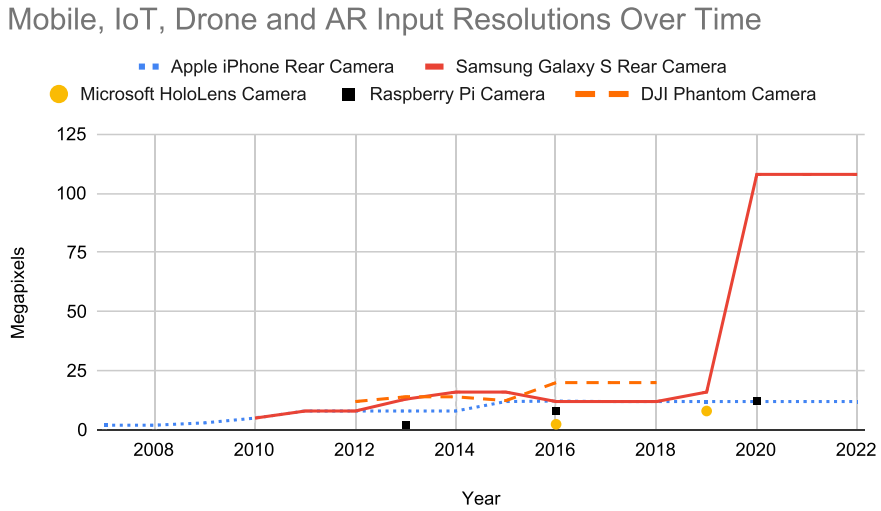

Fig. 3. Trend of the maximum resolutions captured by smartphones (Apple iPhone and Samsung Galaxy S), drones (DJI Phantom), augmented reality headsets (Microsoft HoloLens) and IoT devices (Raspberry Pi) over time. Details and data sources are available in Appendix A.

Although newer generations of GPUs are getting faster and have more memory available, the resolution of images and videos captured by devices is also increasing. Figure 3 shows this trend across recent years for multiple types of devices. Therefore, the aforementioned issues will likely persist even with advances in computation hardware technology. Furthermore, current imaging technologies are nowhere near the physical limits of image resolutions, which is estimated to be in petapixels [11].

Whether or not capturing and processing a higher resolution leads to improvements depends on the particular problem at hand. For instance, in image classification, it is unlikely that increasing the resolution for images of objects or animals to gigapixels would reveal more beneficial details and improve the accuracy. On the other hand, if the goal is to count the total number of people in scenes such as the one presented in Figure 1, using an HD resolution instead of gigapixels would mean that several people could be represented by a single pixel, which significantly increases the error. Similarly, it has been shown that using higher resolutions in histopathology can lead to better results [89].

Assuming there is an effective maximum resolution for a particular problem due to hardware limitations or latency requirements, there are two simple baseline approaches for processing the original captured inputs which are commonly used in deep learning literature [21, 30, 102]. The popularity of these baselines can be attributed to the simplicity of their implementation. The first one is to resize (downsample) the original input to the desired resolution, however, this will lead to a lower accuracy if any important details for the problem at hand are lost. This approach is called ***uniform downsampling (UD)*** since the quality is reduced uniformly throughout the image. The second approach is to cut up the original input into smaller patches that each have a maximum resolution, process the patches independently, and aggregate the results, for instance, by summing them up for regression problems and majority voting for classification problems. We call this approach ***cutting into patches (CIP)***. There are two issues with this approach. First, many deep learning models rely on global features which will be lost since features extracted from each patch will not be shared with other patches, leading to decreased accuracy. For instance, crowd counting methods typically heavily rely on global information such as perspective or illumination [45, 116],

Table 1. Performance of Baseline Approaches on the Shanghai Tech Part B Dataset

Input Size	Shanghai Tech Part B				PANDA			
	Uniform Downsampling		Cutting Into Patches		Uniform Downsampling		Cutting Into Patches	
	MAE	Time (ms)	MAE	Time (ms)	MAE	Time (ms)	MAE	Time (ms)
Original	6.31	7.02	6.31	7.02	262.21	30.91	262.21	30.91
Reduced 4×	9.01	2.00	6.40	7.11	335.81	8.16	203.51	31.68
Reduced 16×	16.06	1.14	6.67	7.48	440.46	2.21	193.36	31.99

and in object detection, objects near the boundaries may be split between multiple patches. Secondly, since multiple passes of inference are performed, that is, one pass for each patch, inference will take much longer. This issue is worse when patches overlap.

To highlight these issues, we test the two baseline approaches (UD and CIP) on the Shanghai Tech Part B dataset [162] for crowd counting, which contains images of size 1024 × 768 pixels, as well as the PANDA dataset [144], which contains gigapixel images. However, we resize the gigapixel images to 2,560×1,440 in order to comply with our hardware limitations. We reduce the original image size by factors of 4 and 16 and measure the **mean absolute error (MAE)** for both baselines. To test UD, we take pre-trained a SASNet model [116] and fine-tune it for the target input size using the AdamW optimizer [88]. Note that the original SASNet paper uses the Adam optimizer [71]. We train the model for 100 epochs with batch size of 12 per GPU instance using 3×Nvidia A6000 GPUs for Shanghai Tech Part B experiments, and a batch size of 1 for PANDA experiments. We empirically found that fine-tuning does not improve the accuracy of cutting into patches, therefore, we cut the original image into 4 and 16 patches, and obtain the count for each patch using the pre-trained SASNet mentioned above, then aggregate the results by summing up the predicted count for each patch.

The results of these experiments are shown in Table 1. It can be observed that uniform downsampling significantly increases the error compared to processing the original input size. Keep in mind that even though the increase in error is not as drastic with cutting into patches, and there are even improvements in some cases, the inference time of this approach is increased by the same factor (i.e., 4 and 16) when using the effective maximum resolution possible for hardware. This is due to the fact that patches cannot be processed in parallel, as the entire hardware is required to process a single patch. Indeed, with the PANDA experiments, which are close to the maximum effective resolution of our hardware, we can see this drastic increase in computation time when using CIP compared to UD.

Since these baseline approaches are far from ideal, in recent years, several alternative methods have been proposed in the literature in order to improve accuracy and speed while complying with the maximum resolution limitation caused either by memory limitations or speed requirements. The goal of this survey is to summarize and categorize these contributions. To the best of our knowledge, no other survey on the topic of high-resolution deep learning exists. However, there are some surveys that include aspects relevant to this topic. A survey on methods for reducing the computational complexity of Transformer architectures is provided in [122], which discusses the issues related to the quadratic time and memory complexity of self-attention and analyzes various aspects of efficiency including memory footprint and computational cost. While reducing the computational complexity of Transformer models can contribute to efficient processing of high-resolution inputs, in this survey, we only include Vision Transformer methods that explicitly focus on high-resolution images. Some application-specific surveys include high-resolution datasets and methods that operate on such data. For instance, a survey on deep learning for histopathology, which mentions challenges with processing the giga-resolution of WSIs, is provided in [118]; a survey of methods that achieve greater spatial resolution in **computed tomography (CT)** is

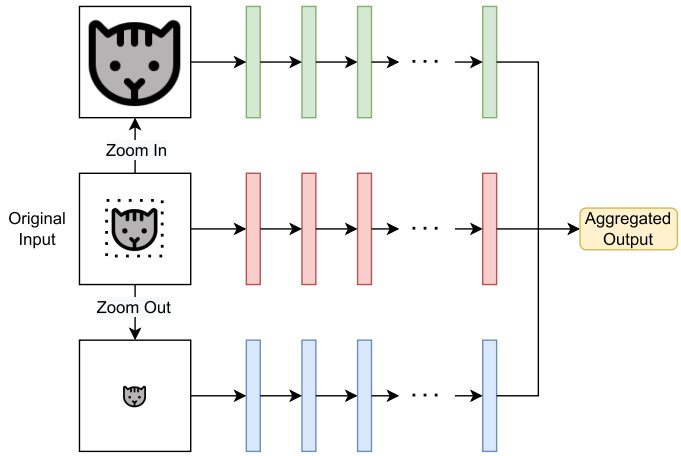

Fig. 4. Schematic illustration of a multi-column architecture. If the original input to the DNN is a patch taken from a larger image, such as in [167], in addition to zooming in, it is also possible to zoom out.

provided in [111], which highlights improved diagnostic accuracy with ultra high-resolution CT, and briefly discusses deep learning methods for noise reduction and reconstruction; a survey on crowd counting where many of the available datasets are high-resolution is provided in [45]; a survey on deep learning methods for land cover classification and object detection in high-resolution remote sensing imagery is provided in [161]; and a survey on deep learning-based change detection in high-resolution remote sensing images is provided in [66].

It is important to mention that some methods operate on high-resolution inputs, yet do not make any effort to address the aforementioned challenges. For instance, *multi-column* (also known as *multi-scale*) networks [45, 116] incorporate multiple columns of layers in their architecture, where each column is responsible for processing a specific scale as shown in Figure 4. However, since the columns process the same resolution as the original input, most of these methods in fact require even more memory and computation compared to the case where only the original scale is processed. The primary goal of these methods is instead to increase the accuracy by taking into account the scale variances that occur in high-resolution images, although there are some multi-scale methods that improve both accuracy and efficiency [15, 138, 164]. Therefore, these methods do not fall within the scope of this survey, unless they explicitly address the efficiency aspect for high-resolution inputs. ZoomCount [109], Locality-Aware Crowd Counting [167], RAZ-Net [86] and Learn to Scale [149] are all examples of multi-scale methods in crowd counting, and DMMN [57] and KGZNet [139] in medical image processing.

The primary purpose of this survey is to collect and describe methods that exist in deep learning literature, which can be used in situations where the high resolution of input images and videos creates the aforementioned technical challenges regarding memory, computation and time. The rest of this paper is organized as follows: Section 2 lists applications where high-resolution images and videos are processed using deep learning. Section 3 categorizes efficient methods for high-resolution deep learning into five general categories and provides several examples for each category. This section also briefly discusses alternative approaches for solving the memory and processing time issues caused by high-resolution inputs. Section 4 lists existing high-resolution datasets for various deep learning problems and provides details for each of them. Section 5 discusses the advantages and disadvantages of using efficient high-resolution methods belonging to different categories and provides recommendations about which method to use in different

situations. Finally, Section 6 concludes the paper by summarizing the current state and trends in high-resolution deep learning as well as suggestions for future research. The code for experiments conducted in this survey is available at https://gitlab.au.dk/maleci/high-resolution-deep-learning.

2 APPLICATIONS OF HIGH-RESOLUTION DEEP LEARNING

In this section, we list some real-world applications where high-resolution images are processed with deep learning. Most of these methods do not focus on the efficiency angle, however, some of the methods address issues encountered with high-resolution images. For instance, [91] mentions that "it was not possible to train the model with the original 6,000 × 4,000 pixel images because of GPU memory limitation" and [151], which uses the cutting into patches approach, states that "a raw remote image has millions of pixels and is difficult to process directly".

2.1 Medical and Biomedical Image Analysis

Multi-gigapixel whole-slide pathology images can be processed with deep learning in order to detect breast cancer [87], skin cancer [140, 147], prostate cancer [147], lung cancer [147], cervical cancer [22], and cancer in the digestive tract [128]. Some methods are even able to detect the cancer subtypes [147] or detect the spread of cancer to lymph nodes (metastasis) [83]. Semantic segmentation of such images can be useful in neuropathology [77], which is the study of diseases of the nervous system, and identifying tissue components such as tumor, muscle, and debris in medical images [65].

Moreover, the processing of high-resolution computed tomography (CT) scans with deep learning is becoming more prevalent. The studies in [153] and [17] detect COVID-19 in high-resolution CT scans of the lung, and [3] uses deep learning to improve the quality of captured ultra-high-resolution CT scans. In addition, the study in [70] performs semantic segmentation on high-resolution electron microscopy images from hearts and brains of mice, which is useful for fundamental biomedical research. Additionally, high-resolution deep learning can be used for reconstruction of CT images and reduction of image noise, which has been shown to obtain results similar to other conventional methods with clinically feasible speed [43, 95].

Even though medical image analysis methods primarily focus on improving the accuracy of particular tasks, inference speed can be crucial in some applications, for instance, speed might be a requirement in clinical practice [83]. Furthermore, real-time augmented reality under microscopes can provide suitable human–computer interaction for AI-assisted slide screening [22]. Finally, there might be situations where the speed for processing a single input is acceptable, however, the sheer number of input data is so high that inputs collectively cannot be processed within a deadline. For instance, 55,000 high-resolution images are taken during the examination of a single patient using wireless capsule endoscopy, where a tiny wireless camera is swallowed to take pictures of the digestive tract, which can be used to detect lesions and inflammation [148].

2.2 Remote Sensing

Processing high-resolution aerial and satellite imagery with deep learning has various applications [7], such as detecting buildings [133], which is useful for urban planning and monitoring; detecting airplanes [4], which can be used for defense and military applications as well as airport surveillance; extracting road networks [151], which has applications in automated road navigation with unmanned vehicles, urban planning and real-time updating of geospatial databases; detecting areas in a forest that are damaged due to natural disasters such as storms [52]; identifying weed plants, which can be used for targeted spraying of pesticides in agricultural fields; semantic segmentation of satellite data which can help with crop monitoring, natural resource management and digital mapping [31]; and remote sensing image captioning which is useful for applications

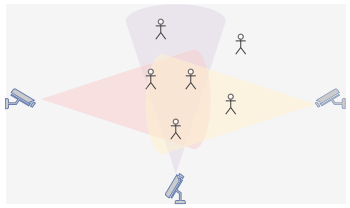

Fig. 5. Overlap in the field of view for multi-camera setups, which can result in duplicates in tasks such as crowd counting.

such as image retrieval and military intelligence generation [165]. Moreover, significant accuracy improvements can be obtained by taking low-resolution weather data as input and interpolating high-resolution data using super-resolution [106]. The motivation behind this approach is that high-resolution data are only available with a few days delay, and this method can be used to more accurately process low-resolution but up-to-date data.

2.3 Surveillance

Capturing and processing gigapixel images for surveillance is becoming increasingly widespread, and such images can be processed with deep learning for searching and identifying people [42, 117] as well as detecting pedestrians [26, 80] which can be used for human behavior analysis and intelligent video surveillance such as enforcing social distancing restrictions during a pandemic [1, 2]. It should be noted that capturing gigapixel images for surveillance has several advantages over capturing lower resolutions with multiple cameras at different locations of the scene. First, cameras in a multi-camera setup typically have some overlap in their fields of view to avoid blindspots. This may result in errors for many applications, such as crowd counting, due to duplicates, as shown in Figure 5. Reducing this error is not an easy task, since it requires information about the geometry of the scene and the use of re-identification methods for identifying and deduplicating people in multiple views of the same scene. Secondly, tracking the trajectory of people, vehicles and other moving objects is difficult with multiple cameras, since it also requires identifying them in multiple views of the scene. Finally, in many deep learning applications such as crowd counting, incorporating global information from the entire scene, such as illumination and perspective, improves the accuracy of the task [45, 116]. Note that images captured from drastically different locations and perspectives, such as the ones in in Figure 5, cannot be stitched together to form a single image.

2.4 Other Applications

High-resolution deep learning can be beneficial in many other applications and various domains of science. For instance, the study in [91] estimates the density of wheat ears, which are the grain-bearing parts of the plant, from high-resolution images taken from grain fields, which aids plant breeders in optimizing their yield; and the study in [59] introduces a deep learning method for segmentation of high-resolution electron microscopy images, which has applications in material science such as understanding the degradation process of industrial catalysts. [84] proposes a method for real-time high-resolution background replacement, which is useful in video calls and conferencing.

3 METHODS FOR EFFICIENT PROCESSING OF HIGH-RESOLUTION INPUTS WITH DEEP LEARNING

We classify deep learning methods for efficient processing of high-resolution inputs into five categories, as summarized in Figure 6. First, non-uniform downsampling methods use the result

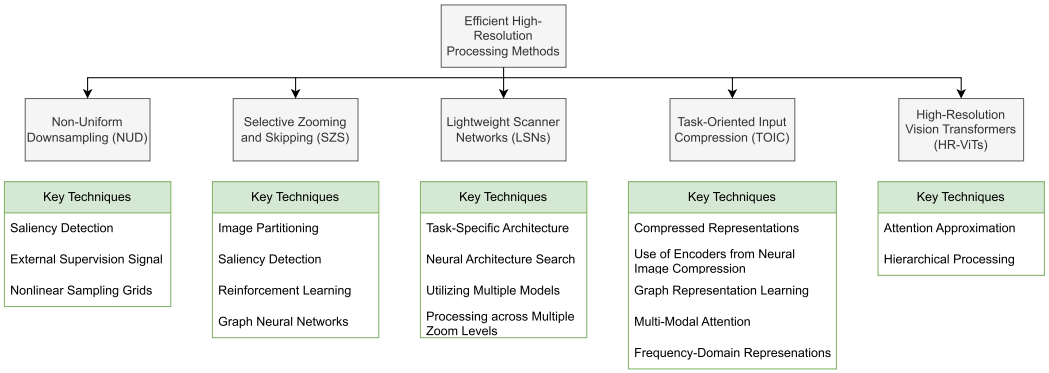

Fig. 6. Overview of methods for efficient processing of high-resolution inputs.

of saliency detection methods to define a nonlinear sampling grid, and downsample the image in a non-uniform fashion. These methods often rely on external supervision functions and custom loss functions for optimal training. Second, selective zooming and skipping methods partition the high-resolution image into several patches. These patches are then prioritized using saliency detection or reinforcement learning. Alternatively, the relationship between the patches can be modeled using graph neural networks, which can help determine patch priority. High-priority patches are then processed using computationally expensive high-performance DNNs, whereas low-priority patches are either processed using lightweight DNNs or discarded entirely. Third, lightweight scanner networks design one or more ultra-lightweight architectures tailored to the specific task at hand. Neural architecture search can be used to aid the design of such architectures. Furthermore, multiple models may be designed to process the image across multiple scales and resolutions, which are then combined to produce a final result. Fourth, task-oriented input compression methods use encoders, graph representation learning or frequency-domain transforms to obtain compressed representations for high-resolution images, which require less computation to process. Multi-modal attention can also be used to reduce the size of representations for high-resolution modalities. Finally, high-resolution Vision Transformers reduce the quadratic cost of the attention operation by various approximation approaches. High-resolution images can also be processed with ViTs in a hierarchical manner to alleviate the quadratic cost imposed as a result of large input sizes.

3.1 Non-Uniform Downsampling

Non-uniform downsampling (NUD) is based on the idea that for any deep learning task, some locations of an input image are more important than others. For instance, in gaze estimation, where the goal is to detect where a person is looking given an image including the person's face, the image locations depicting the person's eyes are much more important than other parts of the image. Therefore, when reducing the resolution of the image, it might be beneficial to sample more pixels from salient areas and less pixels from non-salient locations, resulting in a warped and distorted image. This operation requires salient areas to be determined before introducing the downsampled image to the task DNN. Therefore, a small saliency detection network is utilized in order to obtain this saliency map. Figure 7 provides a schematic illustration of the non-uniform downsampling approach. Note that non-uniform downsampling is a broad process that encompasses any method that downsamples the input image in any manner other than uniform. [102] further subdivides non-uniform downsampling into three categories: attention mechanisms, saliency-based methods

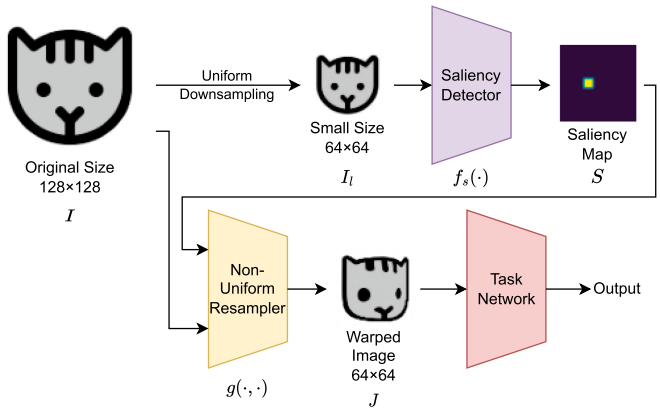

Fig. 7. Schematic illustration of the non-uniform downsampling approach. The saliency detector detects the cat's right eye as a salient area, therefore, the non-uniform resampler samples more pixels from that area.

and adaptive image sampling methods. However, as the authors point out, there is a lot of overlap between these categories and it is difficult to draw a clear border between them.

Formally, the saliency map S can be obtained by applying saliency detection network $f_s(\cdot)$ on a uniformly downsampled image I_l, that is, $S = f_s(I_l)$. The input to the saliency detection network is downsampled in order to keep the overhead of the saliency detection process low. The non-uniformly downsampled image J can then be obtained based on $J = g(I, S)$, where $g(\cdot)$ is the non-uniform resampler and I is the original image. Essentially, the resampler should compute a mapping $J(x, y) = I(u_c(x, y), v_c(x, y))$ from the original image to the downsampled one. Functions $u_c(\cdot)$ and $v_c(\cdot)$ need to map pixels proportionally to the weight assigned to them in the saliency map. Assuming the saliency map is normalized and $\forall x, y : 0 \leq u_c(x, y) \leq 1$ and $\forall x, y : 0 \leq v_c(x, y) \leq 1$, this problem can be written as

$$\int_0^{u_c(x,y)} \int_0^{v_c(x,y)} S(x', y')dx'dy' = xy. \tag{1}$$

However, methods for determining this transformation based on Equation (1) are not efficient [102]. An alternative approach is to presume each pixel (x', y') is pulling all other pixels with a force proportional to its saliency $S(x', y')$, which can be formulated as

$$u_c(x, y) = \frac{\sum_{x',y'} S(x', y')k((x, y), (x', y'))x'}{\sum_{x',y'} S(x', y')k((x, y), (x', y'))}, \tag{2}$$

$$v_c(x, y) = \frac{\sum_{x',y'} S(x', y')k((x, y), (x', y'))y'}{\sum_{x',y'} S(x', y')k((x, y), (x', y'))}, \tag{3}$$

where $k((x, y), (x', y'))$ is a distance kernel, for instance, the Gaussian kernel. Using this formulation, salient areas will be sampled more, since they attract more pixels. Moreover, based on this formulation, $u_c(\cdot)$ and $v_c(\cdot)$ can be computed with simple convolutions. Therefore, this operation can be easily plugged into neural network architectures as a layer, and has the added benefit of preserving the differentiability, which is a requirement for training neural networks with the back-propagation algorithm. The overall result is that the entire module, including the saliency detection network and the task network, can be trained end-to-end. The method in [102] uses this approach to improve the performance of gaze estimation as well as fine-grained classification, which is the task of differentiating between hard-to-distinguish objects such as different species of animals.

The method in [92] applies the idea of non-uniform downsampling to semantic segmentation. If the input image $I = I_{ij}$ has a size $H \times W$ and must be downsampled to size $h \times w$, the first step is to generate ideal sampling tensors from **ground truth (GT)** labels based on

$$E(\phi) = \sum_{i,j} \|\phi_{ij} - b(u_{ij})\|^2 + \lambda \sum_{|i-i'|+|j-j'|=1} \|\phi_{ij} - \phi_{i'j'}\|^2, \tag{4}$$

where $\phi \in [0,1]^{h \times w \times 2}$ is the sampling tensor to be determined, $E(\phi)$ is the (energy) cost function to minimize, $u \in [0,1]^{h \times w \times 2}$ is the uniform downsampling tensor and $b(u_{ij})$ is the coordinates of the closest point to pixel u_{ij} on semantic boundaries in the GT labels. Equation (4) corresponds to a least squares problem with convex constraints that can be efficiently solved using a set of sparse linear equations. The first term in Equation (4) ensures the sampling locations are close to the semantic boundaries, and the second term ensures that the distortion is not excessive by forcing the transformations of adjacent pixels to be similar. Equation (4) is also subject to covering constraints that ensure the sampled locations cover the whole image. The contribution of the second term is controlled by a parameter λ which is empirically set to 1. The next step is to train a neural network to generate sampling tensors from input images. The images are then downsampled based on the output of this neural network and introduced to the task network. Finally, the segmentation output is upsampled to remove distortions and match the original resolution.

Similarly, the method in [68] utilizes non-uniform downsampling for semantic segmentation. However, in contrast with the previous method, the saliency detector in this method is optimized based on the performance of semantic segmentation rather than external supervision signals. This method is similar to [102], however, applying a straightforward adaptation of [102] to semantic segmentation does not perform well. To improve the performance, an *edge loss* is added as a regularization term, which is calculated by using the **mean squared error (MSE)** between the deformation map d obtained by the saliency detector and target deformation map d_t calculated based on segmentation labels. To combat trivial solutions, the target deformation map has denser sampling around object boundaries and is formulated by $d_t = f_{\text{edge}}(f_{\text{gauss}}(Y_{lr}))$, where Y_{lr} is the uniformly downsampled segmentation label, f_{edge} is an edge detection filter by convolution with a specific 3×3 kernel, and f_{gauss} is Gaussian blur with $\sigma = 1$.

Since the distortions caused by the *customized grids* defined in Equations (2) and (3) can be severe, the method in [148] introduces *structured grids* that can be combined with customized grids to obtain a more subtle spatial distortion effect for **wireless capsule endoscopy (WCE)** image classification. These structured grids ensure that pixels that were in the same row/column in the input image are also in the same row/column in the output image, and can be obtained by

$$u(x) = \frac{\sum_{x'} S(x')k(x,x')x'}{\sum_{x'} S(x')k(x,x')}, \tag{5}$$

$$v(y) = \frac{\sum_{y'} S(y')k(y,y')x'}{\sum_{y'} S(y')k(y,y')}, \tag{6}$$

where $S(x) = \max_y S(x,y)$ and $S(y) = \max_x S(x,y)$. $u(x)$ and $v(y)$ are then copied and stacked to form $u_s(x,y) = u(x)$ and $v_s(x,y) = v(y)$. Finally, the combined deformation grids can be computed by

$$u(x,y) = \lambda u_s(x,y) + (1-\lambda)u_c(x,y), \tag{7}$$

$$v(x,y) = \lambda v_s(x,y) + (1-\lambda)v_c(x,y), \tag{8}$$

where parameter λ is empirically set to 0.5.

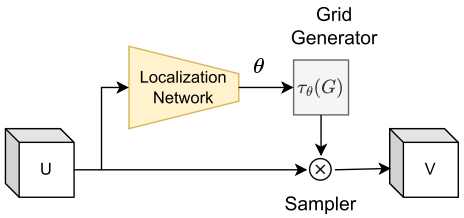

Fig. 8. Architecture of the spatial transformer module [64].

Similarly, FOVEA [124] discards custom grids and solely relies on structured grids for object detection in autonomous driving use cases. It also introduces *anti-cropping regularization* to combat cropping which may result in missing objects, by using reflect padding on the saliency map. In [102], the saliency detector is trained end-to-end along with the task network, however, as mentioned, finding saliency maps in object detection is more difficult. Therefore FOVEA uses intermediate supervision to train the saliency detection network.

Even though the primary goal of the *spatial transformer* module in **spatial transformer networks (STNs)** [64] is to learn invariance to translation, scale, rotation and warping in order to improve performance, in the special case where the module is the first layer of the network, it can learn to crop the raw high-resolution input to a lower resolution and increase computational efficiency, thus it could be considered a form of NUD. Figure 8 shows the architecture of the spatial transformer module, where the localization network determines the parameters θ for the transformation τ_θ from input features U. $\tau_\theta(\cdot)$ can be a 2D affine transformation, a more constrained transformation such as

$$A_\theta = \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \end{bmatrix}, \tag{9}$$

which only allows cropping, translation and scaling, or a more general transformation such as plane projective transformation with 8 parameters, piecewise affine, thin plate spline [40], or any transformation as long as it is differentiable with respect to its parameters.

SALISA [8] uses spatial transformer modules to perform non-uniform downsampling for object detection in high-resolution videos. In SALISA, the output of a video frame is used to determine the saliency map for the next frame. Figure 9 shows this method, where the first frame is introduced to a high-performing detector without any downsampling. The detected objects are subsequently used to create a saliency map, which is then given to the resampling module. The resampling module contains a spatial transformer module with a thin plate spline transformation, where the localization network receives the saliency map as input. The downsampled image provided by the resampling module is then introduced to a lightweight detector. Since the lightweight detector detects objects in the warped image, the detected bounding boxes need to be transformed back into the original grid. Therefore, an inverse transformation is applied before generating the saliency map. To prevent cascading errors, the method is reset to use the original high-resolution frame and high-performing detector every few frames.

3.2 Selective Zooming and Skipping

Selective zooming and skipping (SZS) methods take a more efficient approach to cutting into patches by only zooming into regions of the input image that are important. The zoom level may differ across different patches, and some patches may be entirely skipped. ***Reinforced Auto-Zoom Net (RAZN)*** [35] uses reinforcement learning to determine where to zoom in WSIs for the task of breast cancer segmentation. RAZN assumes the zoom-in action can be performed at most m

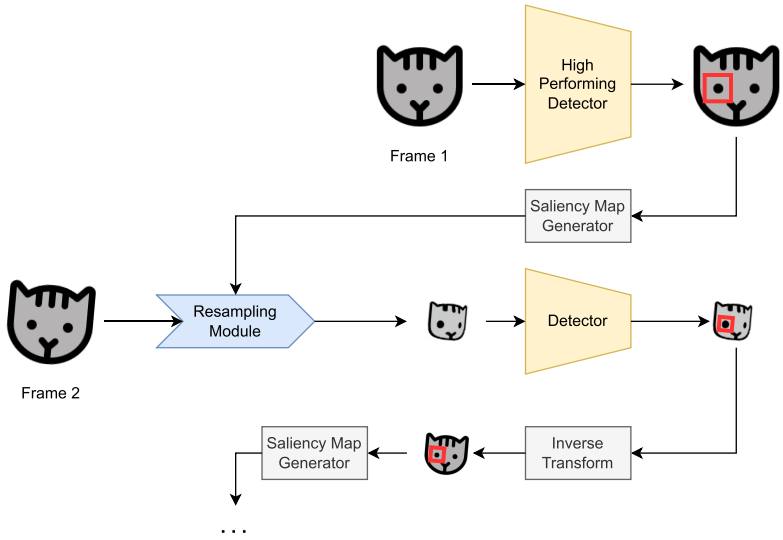

Fig. 9. Overview of SALISA [8]. The second frame is slightly different from the first frame (in this case, slightly rotated clockwise), therefore, the detection result obtained from the first frame can be used to estimate the saliency of objects in the second frame.

times and the zooming rate is a constant r. At each zoom level i, there is a different segmentation network f_{θ_i} and a different policy network g_{θ_i}. Initially, policy network g_{θ_0} takes a cropped image $x_0 \in \mathbb{R}^{H \times W \times 3}$ as input and determines whether to zoom-in or to break. If there is no need to zoom in, x_0 is given as input to segmentation network f_{θ_0} which produces the output, otherwise, a higher-resolution image $\hat{x}_0 \in \mathbb{R}^{rH \times rW \times 3}$ is sampled from the same area and will be cut into r^2 patches of size $H \times W \times 3$. Each patch is then given to policy network g_{θ_1} and this process is recursively repeated until all policy networks break or the maximum zoom level is reached. RAZN achieves an improved performance over other state-of-the-art methods while reducing the inference time by a factor of ~2. Similarly, the methods in [46] and [132] use reinforcement learning for efficient object detection and aerial image classification, respectively.

Instead of reinforcement learning, the method in [38] uses a hierarchical graph neural network to classify whether a mammogram (X-ray image of a breast) is normal/benign (contains a tumor that is not cancerous) or malignant (contains a tumor that is cancerous). At each zoom level i, the graph G^i is defined by the adjacency matrix $A^i \in \mathbb{R}^{N_i \times N_i}$ where there is an edge between each zoomed-in patch and its original image. The feature matrix of the graph is defined as $X_i \in \mathbb{R}^{N_i \times D \times D}$, and the maximum zoom level is R. The features on the nodes are zoomed-in regions of the input image, resized to $D \times D$. A pre-trained CNN is used to extract feature vectors $H_i \in \mathbb{R}^{N_i \times H}$ from X_i. $\text{GAT}_{\text{node}}(\cdot)$ is a graph attention network [136] used to classify whether to zoom in for each node. Therefore, the output of the i-th level in the hierarchical graph is

$$P_i = \begin{cases} 1, & i = 1, \\ \text{softmax}(\text{GAT}_{\text{node}}(A_i, H_i)), & 1 < i < R, \end{cases} \tag{10}$$

where $P_i \in \mathbb{R}^{N_i \times 2}$ represents the decision to zoom or not for each node of the i-th level. At the final zoom level R, another graph attention network $\text{GAT}_{\text{graph}}(\cdot)$ is used to perform the final classification for the entire mammogram based on $\hat{Y} = \text{softmax}(\text{GAT}_{\text{graph}}(A_R, H_R)W)$, where W is a trainable weight matrix. The loss function contains both node losses and graph losses, with the zoom

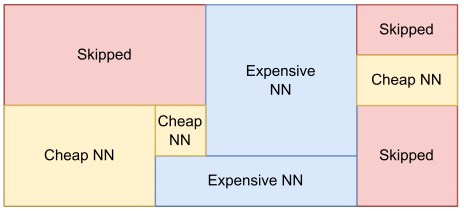

Fig. 10. Partitioning in REMIX [67]. Some parts of the image are skipped, some processed by computationally cheap DNNs, and some by computationally expensive DNNs.

labels for nodes being obtained from lesion segmentation labels. This method achieves an accuracy comparable to the state of the art, however, it is unclear how much it improves the inference speed.

GigaDet [18] achieves near real-time object detection in gigapixel videos. At the core of GigaDet is the ***Patch Generation Network (PGN)***. PGN takes a uniformly downsampled image as input and outputs a dense regression map which counts the number of objects that are completely contained within the corresponding area in the image, referred to as the *patch candidate*. PGN is applied at different scales in order to obtain patch candidates of varying scales. The patch candidates selected by the PGN go through post-processing which includes **non-maximum suppression (NMS)**, and are subsequently sorted based on their count. The top K patch candidates are then selected to be processed by the ***Decorated Detector (DecDet)*** to detect objects. VGG [114] and YOLO [103] are used for the PGN and DecDet networks, respectively. Given gigapixel videos, GigaDet is capable of running 5 FPS on a single Nvidia 2080 Ti GPU, which is 50× faster than Faster RCNN [104], yet obtains a comparable performance in terms of average precision.

REMIX [67] detects pedestrians in high-resolution videos within a latency budget given by the user. The input frame is partitioned into several blocks, where more salient blocks are processed using a computationally expensive but accurate network, whereas less salient blocks are processed using a computationally cheap network or even skipped, as shown in Figure 10. REMIX uses historical frames to determine the object distribution, and establishes the optimal partition using a dynamic programming algorithm that takes into account the given latency budget, the estimated object distribution, as well as the accuracy and speed of available neural networks for object detection. REMIX achieves up to 8.1× inference speedup with an accuracy comparable to state-of-the-art methods.

3.3 Lightweight Scanner Networks

Lightweight scanner networks (LSNs) are lightweight **fully convolutional neural networks (FCNs)** that efficiently scan the entire high-resolution input. To achieve a lightweight architecture, LSNs are typically designed and trained for very specific tasks. Moreover, as opposed to the cutting into patches approach, FCNs are inherently efficient in a sliding-window setting since they share the computation in overlapping regions [112].

VGG-720p and VGG-1080p [129, 130] are LSNs capable of running in real-time on drones and provide heatmaps for input images of size 1280×720 and 1920×1080 pixels, respectively, that specify whether or not there are people, faces, or bicycles at each location in the input image. Both models take patches of size 32×32 or 64×64 pixels as input. The architectures of VGG-720p and VGG-1080, shown in Tables 2 and 3, respectively, contain only 5 convolutional layers with only 2 to 24 output channels. In contrast, the original VGG architectures have 11 to 19 layers with up to 512 output channels in some layers [114].

Similarly, the study in [131] proposes an architecture with 6 convolutional layers for the same problem of generating a crowd heatmap from high-resolution images. The study in [127] proposes

Table 2. Architecture of VGG-720p

Layer	Kernel	Stride	Pad† (X/Y)*	Max Pool (X/Y)	Channels
conv1_1	3×3	1/1	1/1	– / –	16
conv1_2	3×3	1/1	1/1	✓/ –	16
conv2_1	3×3	1/1	1/1	– / –	24
conv2_2	3×3	1/4	1/1	✓/ ✓	16
conv_last	8×8	1/1	0/0	– / –	2

†Zero padding.
*X and Y represent the horizontal and vertical axes.

Table 3. Architecture of VGG-1080p

Layer	Kernel	Stride	Pad† (X/Y)*	Max Pool (X/Y)	Channels
conv1_1	3×3	2/1	0/0	– / –	8
conv1_2	3×3	1/2	0/0	✓/ –	8
conv2_1	3×3	1/1	0/0	– / –	6
conv2_2	3×3	1/2	0/0	– / –	6
conv_last	8×8	1/1	0/0	– / –	2

†Zero padding.
*X and Y represent the horizontal and vertical axes.

lightweight FCNs for face detection with 7 convolutional layers and 76K parameters, for facial parts detection (such as eyes, nose and mouth) with 4 convolutional layers and 20K parameters, and for combined face and parts detection with 9 convolutional layers and 101K parameters.

You only look twice (YOLT) [135] is a method that detects objects of different scales in DigitalGlobe satellite images which have a size of over 250 megapixels. The architecture of YOLT is based on the YOLO architecture [103], however, it reduces the number of layers from the original 30 down to 22. Furthermore, YOLT trains two separate models: one which processes images that correspond to areas of $200 \times 200\text{m}^2$ for detecting relatively small objects such as cars, airplanes, boats and buildings, and another which processes images that correspond to areas of $2500 \times 2500\text{m}^2$ for detecting large objects such as airports. YOLT has an inference speed of 32km^2/min for the former model and 6000km^2/min for the latter on an Nvidia Titan X GPU.

Fast ScanNet [83] converts VGG16 [114] to a fully convolutional network by replacing the last fully-connected layers in VGG16 with convolutional layers of kernel size 1×1. Fast ScanNet is applied to patches of size 2800×2800 pixels, a size which is dictated by GPU memory limitations, taken from WSIs, which have ∼400 patches on average. It takes about one minute for Fast ScanNet to process a WSI on a workstation with 8×Nvidia Titan X GPUs.

ICNet [164] takes advantage of both the efficiency of processing lower resolutions and the accuracy of processing higher ones by uniformly downsampling the input image to two smaller scales, processing each scale separately, and fusing the result of processing lower resolutions with higher ones. Lower resolutions are processed with more convolution layers and higher resolutions with less, which makes the entire architecture efficient, as shown in Figure 11. In addition, some of the layers share weights in order to increase the efficiency. ICNet is able to perform semantic segmentation on 2048×1024 images at 30 frames per second with high accuracy on a Titan X GPU. Even though ICNet does not obtain state-of-the-art accuracy, it is ∼ 15× faster than methods with similar performance.

ESPNet [96] relies on ***efficient spatial pyramid (ESP)*** modules which reduce the amount of computation by decomposing standard convolutions with $n \times n$ kernels into two steps. The first

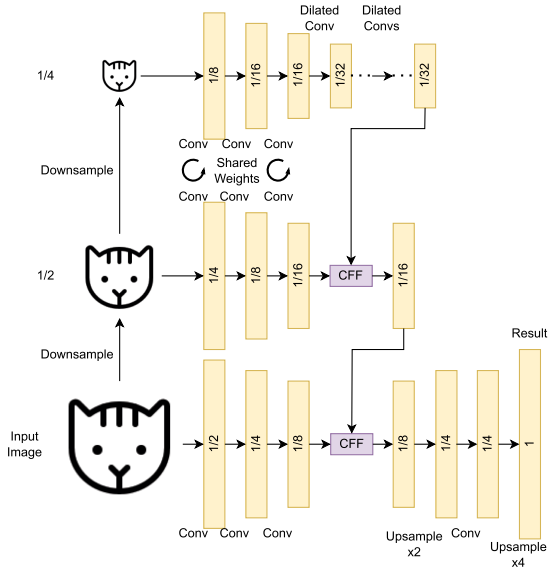

Fig. 11. ICNet architecture. CFF blocks perform the fusion operation and consist of convolution and upsample layers. CFF blocks get supervision signals using downsampled annotations during the training process.

step applies a 1×1 convolution to project feature maps with dimension N to feature maps with dimension $\frac{N}{K}$. The second step applies K dilated convolutions with kernel size $n \times n$ and dilation rates $2^{k-1}, k \in \{1, \ldots, K\}$ to the new feature maps simultaneously, and combines the results. Concatenating the outputs of dilated convolutions creates checkerboard artifacts, therefore, a simple solution is used where the outputs of dilated convolutions are hierarchically added to each other before concatenation. ESPNet can perform semantic segmentation on 2048×1024 images at 54 frames per second with an accuracy comparable to the state of the art.

Neural architecture search (NAS) techniques can be used for designing better LSNs. Since LSNs need to be lightweight and contain few layers and parameters, the search space is relatively small, making NAS easier. HR-NAS [32] is one such method that searches for network architectures that can contain both convolutions and lightweight Transformers, and may have parallel branches. HR-NAS obtains state-of-the-art results in the trade-off between efficiency and accuracy in semantic segmentation, human pose estimation and 3D object detection tasks with high-resolution inputs.

3.4 Task-Oriented Input Compression

Task-oriented input compression (TOIC) methods compress high-resolution inputs into lightweight representations. These representations are then given to the task DNN as input instead of the high-resolution images or videos. The exact nature of the lightweight representations and the compression procedure varies from method to method and is often highly dependent on the underlying task.

There is an important distinction between this approach and *neural image compression* methods such as SlimCAE [152]. The goal of neural image compression is to learn optimal compression algorithms for the task at hand, in order to reduce the size of stored or transmitted data. Therefore, the network that compresses and decompresses this data may be very large and inefficient. Moreover, neural image compression aims to reconstruct the input from the compressed representations,

whereas TOIC does not reconstruct the input data and strives to extract compact representations that are suitable for the second part of the network, which is responsible for performing the task.

Slide Graph [89] recognizes the loss of visual context that comes with using the cutting into patches method, and fixes this issue by building and processing a compact graph representation of the cellular architecture in breast cancer WSIs in order to predict the status of **human epidermal growth factor receptor 2 (HER2)** and **progesterone receptor (PR)**, which are proteins that promote the growth of cancer cells. Slide Graph has four stages: The first stage uses a HoVer-Net [48], which is a CNN for segmentation and classification of cellular nuclei, trained on the PanNuke dataset [44] to extract features of the tissue cells. The second stage uses agglomerative clustering [99] to group neighboring nuclei to further reduce the computational cost. The third stage constructs a graph where each vertex corresponds to a cluster and contains features extracted in the previous stage. Graph edges are constructed based on Delauney triangulation where vertices are represented by the geometric center of their corresponding cluster, which results in a planar graph. In the final stage, HER2 and PR status predictions are obtained from the constructed graph using a **graph convolutional network (GCN)** [73]. Slide graph is more accurate than state-of-the-art methods and reduces the average inference time from 1.2 seconds of the baseline down to 0.4 milliseconds. However, these measurements do not include the graph construction phase. Therefore, the end-to-end improvement in efficiency obtained by Slide Graph is unclear.

The method in [123], shown in Figure 12, compresses gigapixel histopathology WSIs down to a size that can be processed with a CNN on a single GPU. This compression is obtained by training an autoencoder (either VAE [72] or bidirectional GAN [34]) on image patches of size $P \times P \times 3$. The WSI image of size $M \times N \times 3$ is then cut into patches of the aforementioned size, and compressed embeddings of size $1 \times 1 \times C$ are obtained from the patches using the encoder part of the autoencoder. These embeddings are then concatenated to form a compressed image of size $\lceil \frac{M}{P} \rceil \times \lceil \frac{N}{P} \rceil \times C$, which can be given as input to the CNN. In experiments where $M = N = 50,000$ and $P = C = 128$, the input size is reduced by a factor of ∼43.

MCAT [20] uses a combination of WSIs and genomics data for cancer survival outcome prediction. At the core of MCAT is the ***Genomic-Guided Co-Attention (GCA)*** layer which reduces the spatial complexity of processing WSIs. MCAT processes the input in data structures known as *bags*, which are unordered sets of objects of varying size without individual labels. MCAT constructs one bag (H_{bag}) from multiple WSIs in order to utilize the entire tissue microenvironment, and another bag (G_{bag}) from genomic features. H_{bag} is constructed by cutting the WSIs into non-overlapping 256×256 pixel patches and processing each patch with a ResNet50 CNN [55] pre-trained on the ImageNet dataset [29] to obtain d_k-dimensional feature embeddings. G_{bag} is constructed by categorizing genes into N different sets based on similarity and applying a fully-connected (FC) layer to obtain genomic embeddings. GCA then takes these two bags as input and performs the co-attention operation

$$\begin{aligned} \text{CoAttn}_{G \to H}(G, H) &= \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \\ &= \text{softmax}\left(\frac{W_q G H^T W_k^T}{\sqrt{d_k}}\right) W_v H, \end{aligned} \tag{11}$$

where $Q = W_q G$ is the query matrix, $K = W_k H$ is the key matrix, $V = W_v H$ is the value matrix, and $W_q, W_k, W_v \in \mathbb{R}^{d_k \times d_k}$ are trainable weights. The output of this operation, as shown in Figure 13, has a dimension of $N \times d_k$. Therefore, the subsequent self-attention layers in the MCAT network are quadratic with respect to N instead of M. Since on average $M = 15,231$ and $N = 6$, this results in a massive reduction in complexity.

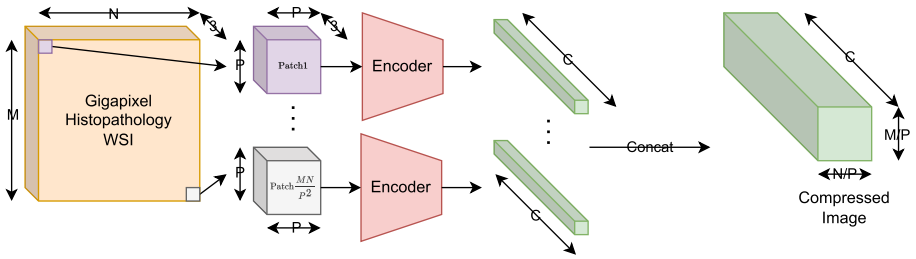

Fig. 12. A method based on neural image compression for gigapixel histopathology images.

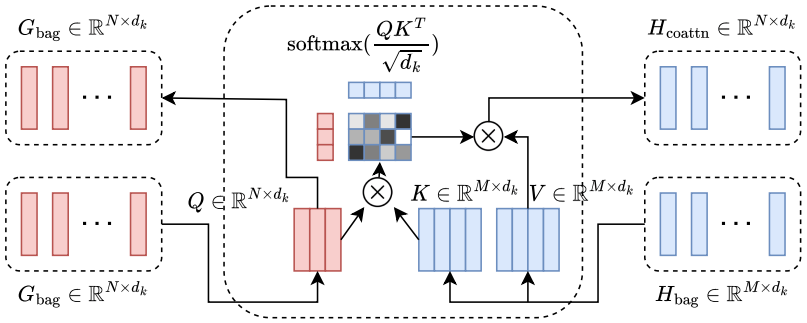

Fig. 13. Genomic-Guided Co-Attention (GCA) layer.

A subcategory of TOIC methods are *frequency-domain DNNs*, which convert input RGB pixels to frequency domain representations with the help of operations such as **discrete cosine transform (DCT)** or wavelet transform. The intuition behind this approach is that the first few layers in CNNs often learn filters that resemble such transforms. Therefore, not only are image representations more compact in the frequency domain, but also a lower number of layers is required for processing such representations.

The method in [50] uses the DCT coefficients obtained in the middle of JPEG encoding as inputs to a modified ResNet50 CNN [55] for the image classification task. JPEG encoding consists of three stages. The first stage converts the input 3-channel 24-bit RGB image to the YCbCr color space by

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168935 & -0.331665 & 0.50059 \\ 0.499813 & -0.418531 & -0.081282 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{12}$$

The *luma* component (Y) represents the brightness, and the *chroma* components (Cb and Cr) represent color. The resolution of chroma components is then reduced by a factor of 2 due to the fact that the human eye is less sensitive to fine color detail than fine brightness. Figure 14 shows an example image and its corresponding Y, Cb and Cr components. The second stage is a blockwise DCT, where each of the three components is partitioned into 8×8 blocks that undergo a 2D DCT. The amplitude values of the frequency domain are the input representations used by this method. The DCT representations of Cb and Cr are upsampled by a factor of two and concatenated with the DCT representation of Y before being given as input to the task DNN, as shown in Figure 15. The rest of the JPEG encoding process contains the quantization of these representations as well as lossless compression techniques such as Huffman coding. However, this method uses the representations obtained before quantization and lossless compression.

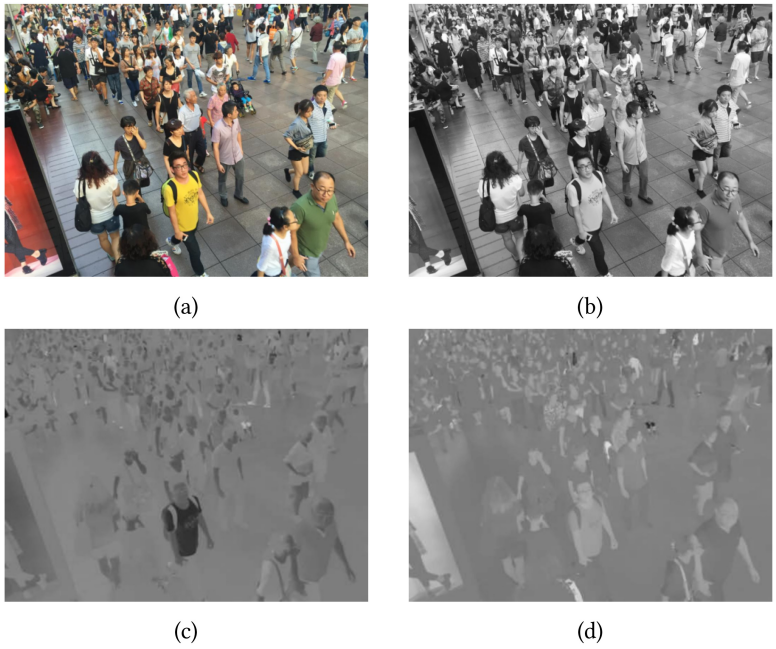

Fig. 14. (a) Original color image, taken from the Shanghai Tech Part B dataset [162]; (b) luma component Y, which is essentially a grayscale version of the color image; (c) chroma component Cb; and (d) chroma component Cr.

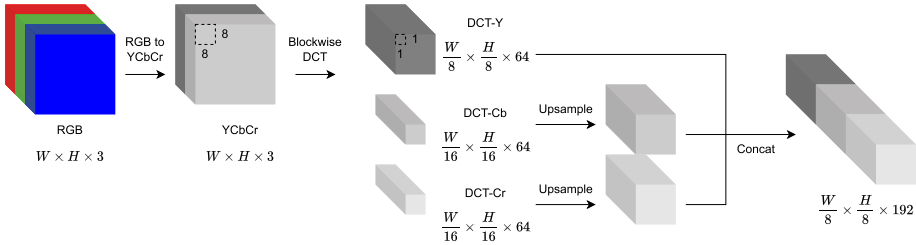

Fig. 15. Initial stages of JPEG encoding, used by [50] to obtain frequency-domains representations of the RGB input.

With the help of these input representations, this method obtains DNNs that are both more accurate and up to 1.77× faster than ResNet50. Moreover, [50] includes experiments attempting to learn convolutions behaving like DCT, however, they find that this learned DCT transform leads to higher error compared to the conventional DCT transform.

The method in [150] uses the same idea for image classification and semantic segmentation tasks using ResNet50 and MobileNetV2 architectures. However, this method also prunes the 192 DCT channels with the help of a gating module that generates a binary decision for each channel. Furthermore, this study discovers that some channels are consistently pruned regardless of the particular task, and develops a static frequency channel selection scheme based on these results. This scheme prunes up to 87.5% of the channels with little accuracy drop, if any. The method in [141] uses the same approach for image classification, however, it uses several variants of ***discrete wavelet transform (DWT)*** instead of DCT. The advantage of DWT over DCT is that it can

obtain a better compression ratio without loss of information, however, it is more computationally expensive [69]. Experiments show that using DWT instead of DCT can lead to higher accuracy, however, the impact of DWT on inference time is unclear.

Finally, similar to images, DNNs can directly process the compressed representations obtained by video compression formats. MMNet [143] performs efficient object detection on H.264/MPEG-4 Part 10 compressed videos [105], one of the most commonly used video compression formats, by taking advantage of the motion information already embedded in the video compression format. It only runs the complete feature extractor DNN on a few reference frames in the video and aggregates the visual information from the subsequent frames with the help of an LSTM [58]. H.264 has two types of frames: *I-frames* which contain a complete image, and *P-frames*, also known as delta frames, which store the offset to previous frames using *motion vectors* and *residual errors*. In MMNet, the extracted motion vectors and residual errors for each P-frame following an I-frame are passed on to the LSTM. MMNet is 3× to 10× faster than competing models with minor loss in accuracy.

3.5 High-Resolution Vision Transformers

As previously mentioned, the self-attention operation in Transformers has a high complexity that increases in a quadratic fashion with respect to the number of input tokens. This operation is formulated by

$$Z = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{13}$$

where query $Q = XW^Q \in \mathbb{R}^{n\times d_q}$, key $K = XW^K \in \mathbb{R}^{n\times d_k}$ and value $V = XW^V \in \mathbb{R}^{n\times d_v}$ are obtained from a sequence of input tokens $X = (x_1, \ldots, x_n) \in \mathbb{R}^{n\times d}$, and W^Q, W^K and W^V are learnable weight matrices. Due to this quadratic complexity, naive approaches, such as ViT [37], that create a long sequence of input tokens from a high-resolution image will lead to massive complexity. On the other hand, if X contains few tokens, each input token represents a large area of the original image, leading to loss of detailed information that might be crucial to some applications.

Vision Longformer (ViL) [160] is a variant of Longformer [9] which has a linear complexity with respect to the number of input tokens, and is capable of processing high-resolution images. This linear complexity is achieved by adding n_g global tokens, which include the classification token *cls*, that serve as global memory by attending to all input tokens. Input tokens are only allowed to attend to the global tokens as well as their neighbors within a 2D window. If the number of input tokens are n_l and the 2D window size is w, then the memory complexity is $O(n_g(n_g + n_l) + n_l w^2)$. When $n_g \ll n_l$, the complexity is significantly reduced from the original n_l^2 in ViT. By using ViL in a multi-scale architecture, multi-scale Vision Longformer is able to obtain superior performance compared to the state-of-the-art in image classification, object detection and semantic segmentation while requiring less computation in terms of FLOPs in some cases.

High-Resolution Transformer (HRFormer) [158] reduces the computational complexity of self-attention by partitioning the input representations into non-overlapping patches, and performing the self-attention only within each patch. Figure 16 shows the building block of HRFormer, which contains a depth-wise convolution that facilitates information exchange between patches. By utilizing this augmented self-attention in a multi-scale architecture, HRFormer obtains superior performance in human pose estimation and semantic segmentation with fewer parameters and FLOPs.

Multi-Scale High-Resolution Vision Transformer (HRViT) [49] uses cross-shaped self-attention [36] and parameter sharing to decrease the computational cost of self-attention. Cross-shaped self-attention, shown in Figure 17, splits the K self-attention heads present in multi-head

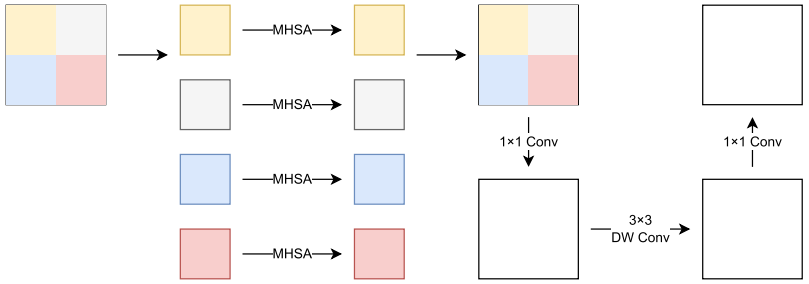

Fig. 16. HRFormer block. **Multi-head self-attention (MHSA)** is applied only within each patch. The patches are then concatenated and followed by a **depth-wise (DW)** convolution.

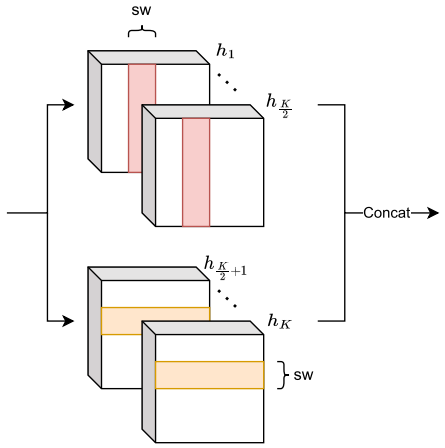

Fig. 17. Cross-shaped self-attention.

attention into two groups: $\{h_1, \ldots, h_{\frac{K}{2}}\}$ and $\{h_{\frac{K}{2}+1}, \ldots, h_K\}$. These groups perform self-attention in horizontal and vertical strips in parallel. Strip width sw can be adjusted to achieve a trade-off between efficiency and performance. The linear projections for key and value tensors are shared in HRViT's blocks to save computation and parameters. In addition to efficient self-attention, HRViT employs a convolutional stem to reduce the spatial dimension of the input. HRViT achieves the best performance-efficiency trade-off compared to state-of-the-art models for semantic segmentation.

Instead of restricting self-attention to patches that are neighbors in the 2D grid, ***Glance and Gaze Transformer (GG-Transformer)*** [156], shown in Figure 18, performs the self-attention within dilated partitions. Since these dilations create holes in the receptive field, a parallel branch containing depth-wise convolution is added to compensate for the local interactions with negligible cost. GG-Transformer achieves superior performance in image classification, object detection and semantic segmentation and reduces the parameters or FLOPs in some cases.

Hierarchical Image Pyramid Transformer (HIPT) [19] processes gigapixel WSIs for the task of cancer subtyping and survival prediction. Since the input WSIs are as large as 150,000 × 150,000 pixels, processing them with a normal ViT and small patch size, such as 16 × 16, results in a massive number of parameters and computational cost requirements, and using large patch sizes such as 4,096 × 4,096 pixels directly would result in loss of cellular information. Therefore, HIPT takes a hierarchical approach, shown in Figure 19, where an initial ViT processes patches of 16 × 16 in an area of size 256 × 256 pixels. A second ViT then takes the aggregated tokens from the previous

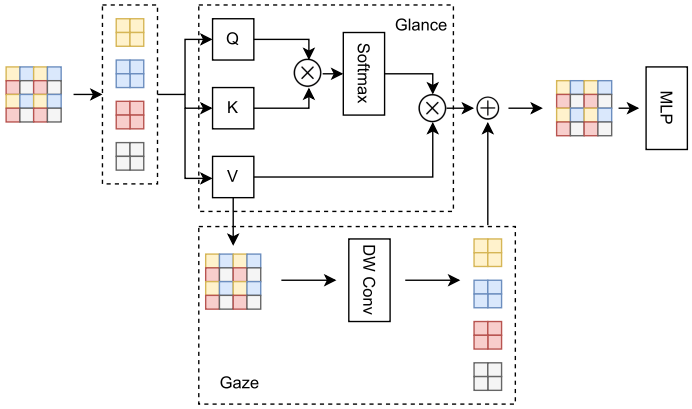

Fig. 18. GG-Transformer block.

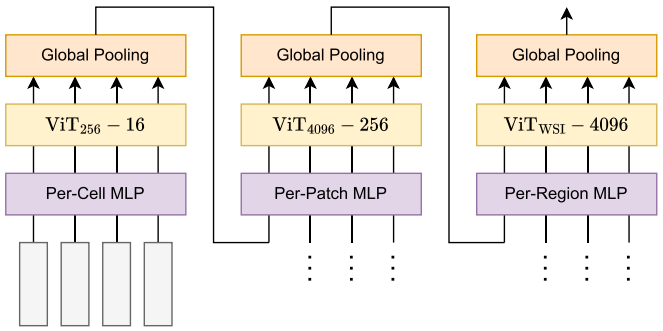

Fig. 19. Hierarchical Image Pyramid Transformer (HIPT). The notation $\mathrm{ViT}_L - l$ means a Vision Transformer that operates on size $L \times L$ with patch size of $l \times l$. $\mathrm{ViT}_{\mathrm{WSI}}$ operates on the entire WSI.

ViT and processes an area of size 4096×4096 pixels. A final ViT takes the aggregated tokens from the second ViT and processes the entire image.

Recent works on efficient ViTs and Transformers reduce memory consumption as well as latency, allowing for more efficient processing of high-resolution images. Even though most of these developments do not explicitly include experiments on high-resolution images, benefits obtained on low-resolution images are likely to be useful for high-resolution images as well. Furthermore, most works on efficient Transformers include experiments on long sequences of text, therefore, the memory and computation improvements are probably beneficial for high-resolution images as well, which are processed as long sequences of image patches.

Conventional model compression techniques have been successfully applied to Vision Transformers. For instance, Q-ViT [82] quantizes Vision Transformers down to 3-bytes without significant reduction in performance, MiniViT [159] applies knowledge distillation to compress the parameters of Vision Transformers by up to 9.7×, and SPViT [54] prunes the Vision Transformer architecture to achieve a 52% reduction in terms of FLOPs, while slightly increasing the performance.

FlashAttention [27] enhances the attention process by incorporating IO-awareness, that is, taking into account the total number of read and write operations between different levels of GPU memory. By reducing the number of read and write operations in GPU memory using tiling, which is the incremental application of softmax reduction, FlashAttention is able to speed up the computation by up to 7.6× and reduce the memory requirement to linear with respect to the input size.

Table 4. List of Popular High-Resolution Datasets

Name	Applications	Resolution (Pixels)	# of Samples	Annotations	Splits	Year	Availability
Supervisely Persons[‡]	Person Segmentation	800 × 1116 to 9933 × 6622	5,711 images	Pixel Mask	None	2018	Public
PANDA [144]	Person Detection	>25K × 14K	555 frames[§]	Person Bounding Box	None	2020	Upon Request
UCF_CC_50 [62]	Crowd Counting	2888 × 2101 on average	50 images	Head Annotations[*]	None	2013	Public
Shanghai Tech Part A [162]	Crowd Counting	868 × 589	482 images	Head Annotations	Train & Test	2016	Public
Shanghai Tech Part B [162]	Crowd Counting	1024 × 768	716 images	Head Annotations	Train & Test	2016	Public
UCF-QNRF [63]	Crowd Counting	2902 × 2013 on average	1,535 images	Head Annotations	Train & Test	2018	Public
PANDA Crowd [144]	Crowd Counting	25,151 × 14,151 to 26,908 × 15,024	45 images	Person Bounding Box	None	2020	Upon Request
JHU-CROWD++ [115]	Crowd Counting	1430 × 910 on average	4,372 images	Head Annotations	Train, Val & Test	2020	Public
NWPU-Crowd [142]	Crowd Counting	3209 × 2191 on average	5,109 images	Head Annotations	Train, Val & Test	2020	Public
DISCO [60]	Audio-Visual Crowd Counting	1920 × 1080 (Full HD)	1,935 images	Head Annotations	Train & Test	2020	Public
CityScapes [25]	Autonomous Driving	2048 × 1024	5K images	Pixel Mask	Train, Val & Test	2016	Upon Request
SYNTHIA-RAND [107]	Autonomous Driving	1280 × 720 (HD)	~13K images	Pixel Mask	Train & Test	2016	Public
ApolloScape [61]	Autonomous Driving	3384 × 2710	~113K images	Pixel Mask	Train & Test	2020	Upon Request
Argoverse-HD [81]	Autonomous Driving	1920 × 1200	89 videos	Bounding Box	Train, Val & Test	2020	Public
BDD100K [155]	Autonomous Driving	1280 × 720 (HD)	100K videos	Bounding Box	Train, Val & Test	2020	Upon Request
nuScenes [13]	Autonomous Driving	1600 × 900	1,000 videos	3D Bounding Box	Train, Val & Test	2020	Upon Request
Waymo Open [119]	Autonomous Driving	1920 × 886 to 1920 × 1280	1,150 videos	2D & 3D Bounding Box	Train, Val & Test	2020	Upon Request
PASCAL-Context [98]	Scene Understanding	500 × 375 to 500 × 500	10,103 images	Pixel Mask	Train & Test	2014	Public
ADE20K [166]	Scene Understanding	683 × 512 to 2100 × 2100	27,574 images	Pixel Mask	Train & Test	2017	Upon Request
COCO-Stuff 10K [14]	Scene Understanding	~640 × 480	10K images	Pixel Mask	Train & Test	2018	Public
DeepGlobe [28]	Land Cover Classification	2448 × 2448	1,146 images	Pixel Mask	Train, Val & Test	2018	Public
Copernicus [12]	Land Cover Classification	20,160 × 20,160	94 images	Pixel Mask	None	2015-2019	Public
fMoW [24]	Aerial Image Classification	up to 16,032 × 14,840	1,047,691 images	Classes	Train, Val & Test	2018	Public
KID [75]	Capsule Endoscopy	360 × 360	~2,500 frames	Pixel Mask	None	2017	Public (N/A)
CAD-CAP [79]	Capsule Endoscopy	576 × 576	25,124 frames	Pixel Mask	Train & Test	2020	Upon Request
CAMELYON16 [154]	Pathology	up to 200,000 × 100,000	400 images	Pixel Mask	Train & Test	2016	Public
TUPAC16 [137]	Pathology	~50,000 × 50,000	821 images	Proliferation Score[†]	Train & Test	2016	Public
BACH Part B [5]	Pathology	(39,980-62,952) × (27,972-44,889)	40 images	Pixel Mask	Train & Test	2019	Public
TCGA-BRCA [74]	Pathology	up to 150,000 × 100,000	709 images	Classes	None	2020	Public
PCa-Histo [68]	Pathology	(1968±216) × (9392±4794)	266 images	Pixel Mask	Train, Val & Test	2021	Private
INbreast [97]	Breast Cancer Detection	2560 × 3328 to 3328 × 4084	410 images	Pixel Mask	Train & Test	2012	Public
UA-DETRAC [145]	Video Object Detection	960 × 540	140K frames	Bounding Box	Train & Test	2015	Public
ImageNet-VID [108]	Video Object Detection	176 × 132 to 1280 × 720 (HD)	5,354 videos	Bounding Box	Train, Val & Test	2015	Public
FAIR1M [120]	Fine-Grained Object Detection	600 × 600 to 10,000 × 10,000	40,000 images	Bounding Box	Train & Test	2021	Public (N/A)
COCO [85]	Object Detection Human Pose Estimation	~640 × 480	>200K images	Pixel Mask Keypoints	Train, Val & Test	2014	Public

[§]A frame is a single image in a sequence representing a video.
[*]The location for the center of each human head in the image is specified.
[†]A measure of the number of cells in a tumor that are dividing.
[‡]https://github.com/supervisely-ecosystem/persons

The authors also introduce *block-sparse FlashAttention*, an approximate extension of FlashAttention, which is up to 4× faster than the original FlashAttention, while obtaining competitive results on several tasks.

In practice, the implementation of dynamic sparse attention algorithms typically leads to slower inference times compared to the full attention algorithm using the FlashAttention framework. Therefore, [100] modifies FlashAttention to facilitate various attention sparsity patterns such as hash-based attention mechanisms as well as query/key dropping attention. Their method obtains speedups for both inference and training on long sequences of text by up to 3.3×.

4 HIGH-RESOLUTION DATASETS

Table 4 lists popular datasets used in high-resolution deep learning literature and provides information about their attributes, such as the deep learning application they are primarily used for, the number of images/videos in the dataset and their resolution, the type of available annotations, whether they specify training/validation/test set splits, the year of publication, and whether they are publicly available. It is important to note that studies reported in some papers create customized datasets. For instance, [46] constructs a dataset from YFCC100M [125]; [129] constructs datasets from AFLW [93], MTFL [163] and WIDER FACE [154]; and [135] constructs datasets from DigitalGlobe satellites, Planet satellites, and aerial platforms.

The Cancer Genome Atlas (TCGA) program is a collaboration between **National Cancer Institute (NCI)** and **National Human Genome Research (NHGRI)**[1]. Since 2006, TCGA has generated over 2.5 petabytes of publicly available data which has led to improvements in cancer

[1]https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga

Table 5. Summary of Applications and Limitations of Efficient High-resolution Methods

Method	Applications	Limitations
NUD	— Tasks with small salient areas (e.g., gaze estimation, object detection, hand gesture recognition, facial expression recognition, cancer tumor detection)	— Not applicable to tasks with large salient areas (e.g., crowd counting, monocular depth estimation) — Leads to severe distortion and low performance in case of massive reduction in image resolution — Requires high-quality saliency detection — May require design of custom loss or regularization — Not suitable for videos with frequent cuts
SZS	— More efficient on scenes with small salient areas	— Less efficient on scenes with large salient areas
LSNs	— Suitable for dense tasks where output is a map (e.g., crowd counting, monocular depth estimation) — Suitable for scenes without perspective (e.g., WSIs, remote sensing)	— Requires custom architecture design; difficult and time-consuming
TOIC	— Frequency-domain methods are general-purpose and applicable to a wide variety of tasks — Can lead to massive speedup compared to other methods	— Requires domain-knowledge and expertise to extract proper representations
HR-ViTs	— General-purpose and applicable to a wide variety of tasks	— Not as efficient as CNNs and other methods

diagnosis, treatment, and prevention. Among efficient high-resolution deep learning methods, the most widely used subset of this data is the **breast invasive carcinoma (BRCA)**, which is outlined in Table 4. However, TCGA provides data for many other types of cancer, such as **bladder urothelial carcinoma (BLCA)**, **glioblastoma and lower grade glioma (GBMLGG)**, **lung adenocarcinoma (LUAD)**, and **uterine corpus endometrial carcinoma (UCEC)**. These are used in some studies, and have properties similar to that of BRCA.

5 DISCUSSION AND OPEN ISSUES

Each of the approaches introduced in Section 3 has its advantages and disadvantages and is useful in certain situations, which are summarized in Table 5. NUD (Section 3.1) works well in cases where the salient area is small compared to the entire image, and thus, it is possible to sample many pixels from such areas. This requirement is satisfied in gaze estimation or object detection problems. Our conjecture is that it would also work well in problems such as hand gesture detection and non-cropped facial expression recognition, although these tasks are not yet explored in the literature in combination with NUD. However, when the salient area is large, for instance, densely populated

scenes in crowd counting or a scene fully covered with objects in object detection, the quality gain obtained by sampling from salient areas will be negligible, and the result of NUD will be similar that of uniform downsampling [8].

Similarly, SZS methods (Section 3.2) require the salient area to be small, otherwise they zoom everywhere and save little time and computation. This also means that the effectiveness of NUD and SZS methods may vary based on the specific input. For instance, the more people there are in an image processed for crowd counting, or the more tumors there are in cancer detection, the less efficient such methods will be, unless there are specific safeguards that prevent them from performing an enormous number of computations, such as GigaDet [18] which processes at most K patch candidates.

Furthermore, NUD methods are not effective when the resulting resolution is extremely smaller compared to the input resolution, for instance, when gigapixel inputs need to be resized down to HD, as this would result in highly distorted images, which makes it difficult for the task DNN to perform well. Even when the gap between the two resolutions is not extremely large, NUD can lead to high distortions in some cases, for instance, it may completely distort and change the shape of the edges of a gastrointestinal lesion, making it difficult for the task network to detect useful features. This may reduce accuracy despite the fact that more pixels are sampled from salient areas. As explained in Section 3.1, some methods try to mitigate the distortion by using structured grids. However, this may limit the benefits obtained by NUD.

In addition, since NUD is enlarging some parts of the image compared to uniform downsampling, some areas of the resulting image will be smaller than they would be with uniform downsampling. Thus, if the saliency map is not of high quality, unimportant areas will be enlarged and the ones important for the final task will shrink, resulting in accuracy loss. This is directly at odds with the requirement that the saliency detection method should be low-overhead, creating another trade-off that needs to be carefully balanced. Moreover, as explained in Section 3.1, some variations of NUD require an external supervision signal or regularization term to train the saliency detection network, which can be difficult to design. In NUD or SZS methods that detect saliency in videos based on the results obtained from previous frames, such as SALISA [8] and REMIX [67], when the difference between subsequent frames is high, the method needs to be reset to processing the entire high-resolution image. When this occurs frequently, the obtained benefits are diminished.

As mentioned in Section 3.3, LSNs need to be designed, trained and well optimized for the specific problem at hand, which is not an easy task. Furthermore, since LSNs produce an output for each scanned area of the input, they are suitable for tasks where the output has the form of a map, such as dense classification or dense regression problems. Moreover, the scanning nature of LSNs means that all areas of the image are treated similarly, therefore, they are better suited for situations where there is no perspective and objects of the same type have the same size regardless of their location, such as WSIs and remote sensing, as opposed to surveillance and crowd counting where people close to the camera are larger than people far away.

Since TOIC methods extract representations that are both compressed and suitable for the task at hand, they often need to be tailored to the specific problem, which requires high domain knowledge. Both Slide Graph [89] and MCAT [20] presented in Section 3.4 are based on domain knowledge about cellular structure of tissues and biological function of genes, respectively. Almost all frequency-domain DNNs try to preserve the architecture of the CNNs they are based on. However, since the interpretation of features in frequency-domain is different, and they have certain properties such as being non-negative, it might be better to customize the architectural elements for the frequency domain, as CS-Fnet [90] does.

Most high-resolution Vision Transformer methods try to reduce the quadratic cost of self-attention to linear, and then compensate the accuracy loss by learning data transformations using

Table 6. Datasets used in Experiments of Various Methods

Task	Method Category	Method	Dataset
Object Detection	NUD	FOVEA [124]	ArgoverseHD [81] & BDD100K [155]
Object Detection	NUD	SALISA [8]	ImageNet VID [108] & UA-DETRAC [145]
Object Detection	SZS	[46]	Caltech Pedestrian [33]
Object Detection	SZS	[132]	fMoW [24]
Object Detection	SZS	GigaDet [18]	PANDA [144]
Object Detection	SZS	REMIX [67]	PANDA [144]
Object Detection	LSNs	HR-NAS [32]	KITTI (3D) [47]
Object Detection	TOIC	MMNet [143]	ImageNet VID [108]
Object Detection	HR-VITs	ViL [160]	COCO [85]
Object Detection	HR-VITs	GG-Transformer [156]	COCO [85]
Histopathology	SZS	RAZN [35]	BACH [5]
Histopathology	LSNs	Fast ScanNet [83]	CAMELYON16 [154]
Histopathology	TOIC	Slide Graph [89]	TCGA-BRCA [74]
Histopathology	TOIC	[123]	CAMELYON16 [154]
Histopathology	TOIC	MCAT [20]	TCGA-BRCA [74]
Histopathology	HR-ViTs	HIPT [19]	TCGA-BRCA [74]

convolutions. To keep the overhead of convolutions low, depth-wise convolution is typically used. Additionally, most high-resolution ViTs utilize a multi-scale architecture in order to capture features of various scales. High-resolution ViTs are more general purpose than other high-resolution deep learning methods and are often used for a large variety of tasks.

Quantitative comparison of various methods is a serious challenge in efficient high-resolution deep learning. As methods available in the literature rarely provide code, in order to compare them against the same benchmark, they need to be reproduced from scratch, which requires massive effort. The next best approach is to compare these methods based on results reported on the same benchmark. However, methods rarely use the same datasets and metrics in their experiments. To shed some light on these challenges, consider Table 6 as an example. Although a single common benchmark among these methods does not exist, several pairs include experiments on the same dataset. However, upon further inspection, it is not possible to make fair comparisons. GigaDet and REMIX both use the PANDA dataset, and ViT and GG-Transformer both use COCO. However, both pairs belong to the same category of methods, therefore, there is little benefit in comparing them. SALISA and MMNet both use ImageNet VID, and they do not belong to the same category of methods. However, SALISA uses GFLOPs as efficiency metric, which is hardware agnostic, while MMNet evaluates efficiency using **frames-per-second (FPS)**, which is hardware dependent. Slide Graph, MCAT and HIPT all use TCGA-BRCA, however, neither MCAT nor HIPT report any efficiency metrics. Finally, Fast ScanNet and [123] both use CAMELYON16, however, Fast ScanNet reports performance using the AUC and FROC metrics, while MCAT reports performance in terms of c-Index, and does not measure efficiency. Due to the trade-off between efficiency and performance, both metrics must be taken into account to properly compare methods and draw meaningful conclusions.

6 CONCLUSION AND OUTLOOK

Processing high-resolution images and videos with deep learning is crucial in various domains of science and technology. However, few methods exist that address the computational challenges. Among existing methods, the trend of designing solutions specifically for the problem at hand is clearly visible. This can be an issue in tasks for which high-resolution datasets are not available.

Similar to model compression approaches, both modifying existing methods and designing an efficient high-resolution method from scratch are viable approaches.

Efficient high-resolution deep learning is in its infancy and there is a lot of room for improvement. For instance, a number of attention-free MLP-based methods have been recently proposed as lightweight alternatives for Transformers [51], which try to mimic the global receptive field of Transformers without the self-attention mechanism. Exploiting such architectures for efficient processing of high-resolution inputs would be an interesting research direction. Furthermore, the multimodal co-attention in MCAT [20] can be applied to many other multimodal tasks, especially the ones with audio, vision and language modalities. Moreover, frequency-domain representations can be explored as inputs to ViTs, which can lead to more efficiency compared to frequency-domain CNNs. For instance, ViTs can take separate patches from DCT-Cb, DCT-Cr and DCT-Y components, bypassing the need to upsample DCT-Cb and DCT-Cr to match the dimensions of DCT-Y.

The combination of efficient high-resolution deep learning with other efficient deep learning methods, such as model compression [23], dynamic inference [53], collaborative inference [16] and continual inference [56], is an unexplored area of research. For instance, if the saliency detection network is a lightweight version of the task network, NUD can be combined with early exiting, where the output of the saliency detection network would be a fast, but less accurate, early result. This is simple to implement in dense regression problems such as depth estimation and crowd counting, where the output of the task can be interpreted as a form of saliency.

Moreover, with the adoption of edge and cloud computing, transmission of high-resolution inputs to servers for processing is a real challenge. As a solution, efficient high-resolution deep learning methods can be combined with edge computing paradigms. For instance, the downsampled images in NUD and compressed representation in TOIC can be transmitted instead of the original inputs. This would be a form of split computing (also known as collaborative intelligence) [6, 94], where the initial portion of computation is performed on a resource-constrained end-device, and the compact intermediate representation is then transmitted to a server where the rest of the computation is carried out. A study using this idea for high-resolution images captured by drones is reported in [10].

Finally, we strongly recommend that future research on high-resolution deep learning methods begin by examining the datasets employed in previous approaches and incorporate relevant datasets into their experimental evaluation. This approach facilitates a more accurate comparison among different methods. Furthermore, it is essential to employ evaluation metrics consistent with relevant literature. Additionally, to facilitate a thorough comparison of methods and determine their positions on the accuracy-efficiency spectrum, it is crucial to report both efficiency and performance metrics. Moreover, metrics that are independent of hardware, such as FLOPs are preferred for evaluation of efficiency, whereas efficiency metrics tied to specific hardware, such as FPS, are challenging to reproduce consistently.

APPENDIX

A DATA SOURCES

Data Sources and details for device camera resolutions are shown in Table 7.

Table 7. Details for Device Camera Resolutions

Device Camera	Year	Resolution (MP)	Source
Apple iPhone Rear Camera	2007	2	link
	2008	2	link
	2009	3	link
	2010	5	link
	2011	8	link
	2012	8	link
	2013	8	link
	2014	8	link
	2015	12	link
	2016	12.2	link
	2017	12	link
	2018	12	link
	2019	12	link
	2020	12	link
	2021	12	link
	2022	12	link
Samsung Galaxy S Rear Camera	2010	5	link
	2011	8	link
	2012	8	link
	2013	13	link
	2014	16	link
	2015	16	link
	2016	12	link
	2017	12	link
	2018	12	link
	2019	16	link
	2020	108	link
	2021	108	link
	2022	108	link
Microsoft HoloLens Camera	2016	2.4	link
	2019	8	link
Raspberry Pi Camera	2013	2.1	link
	2016	8	link
	2020	12.3	link
DJI Phantom Camera	2012	12	link
	2013	14	link
	2014	14	link
	2015	12.4	link
	2016	20	link
	2017	20	link
	2018	20	link

All links were accessed on 26 July 2022.

REFERENCES

[1] Maya Aghaei, Matteo Bustreo, Yiming Wang, Gian Luca Bailo, Pietro Morerio, et al. 2021. Single image human proxemics estimation for visual social distancing. In *IEEE Winter Conference on Applications of Computer Vision*.

[2] Imran Ahmed, Misbah Ahmad, Joel J. P. C. Rodrigues, Gwanggil Jeon, and Sadia Din. 2021. A deep learning-based social distance monitoring framework for COVID-19. *Sustainable Cities and Society* 65 (2021), 102571.

[3] Motonori Akagi, Yuko Nakamura, Toru Higaki, Keigo Narita, Yukiko Honda, et al. 2019. Deep learning reconstruction improves image quality of abdominal ultra-high-resolution CT. *European Radiology* 29, 11 (2019), 6163–6171.

[4] Ugur Alganci, Mehmet Soydas, and Elif Sertel. 2020. Comparative research on deep learning approaches for airplane detection from very high-resolution satellite images. *Remote Sensing* 12, 3 (2020), 458.

[5] Guilherme Aresta, Teresa Araújo, Scotty Kwok, Sai Saketh Chennamsetty, Mohammed Safwan, et al. 2019. BACH: Grand challenge on breast cancer histology images. *Medical Image Analysis* 56 (2019), 122–139.

[6] Arian Bakhtiarnia, Nemanja Milošević, Qi Zhang, Dragana Bajović, and Alexandros Iosifidis. 2022. Dynamic split computing for efficient deep edge intelligence. *arXiv preprint arXiv:2205.11269* (2022).

[7] John E. Ball, Derek T. Anderson, and Chee Seng Chan Sr. 2017. Comprehensive survey of deep learning in remote sensing: Theories, tools, and challenges for the community. *Journal of Applied Remote Sensing* 11, 4 (2017), 042609.

[8] Babak Ehteshami Bejnordi, Amirhossein Habibian, Fatih Porikli, and Amir Ghodrati. 2022. SALISA: Saliency-based input sampling for efficient video object detection. *arXiv preprint arXiv:2204.02397* (2022).

[9] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[10] N. Boehrer, A. Gabriel, A. Brandt, W. Uijens, L. Kampmeijer, et al. 2020. Onboard ROI selection for aerial surveillance using a high resolution, high framerate camera. In *Mobile Multimedia/Image Processing, Security, and Applications*, Vol. 11399. 76 – 95.

[11] David J. Brady, Daniel L. Marks, Steven Feller, Michael Gehm, Dathon Golish, et al. 2013. Petapixel photography and the limits of camera information capacity. In *Computational Imaging XI*. 87–93.

[12] Marcel Buchhorn, Bruno Smets, Luc Bertels, Bert De Roo, Myroslava Lesiv, et al. 2020. Copernicus global land service: Land cover 100m: collection 3: Epoch 2019: Globe. *Version V3. 0.1) [Data set]* (2020).

[13] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, et al. 2020. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[14] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. 2018. COCO-stuff: Thing and stuff classes in context. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1209–1218.

[15] Zhaowei Cai, Quanfu Fan, Rogerio S. Feris, and Nuno Vasconcelos. 2016. A unified multi-scale deep convolutional neural network for fast object detection. In *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer International Publishing, Cham, 354–370.

[16] Joao Carreira, Viorica Patraucean, Laurent Mazare, Andrew Zisserman, and Simon Osindero. 2018. Massively parallel video networks. In *European Conference on Computer Vision*.

[17] Jun Chen, Lianlian Wu, Jun Zhang, Liang Zhang, Dexin Gong, et al. 2020. Deep learning-based model for detecting 2019 novel coronavirus pneumonia on high-resolution computed tomography. *Scientific Reports* 10, 1 (2020), 19196.

[18] Kai Chen, Zerun Wang, Xueyang Wang, Dahan Gong, Longlong Yu, et al. 2022. Towards real-time object detection in GigaPixel-level video. *Neurocomputing* 477 (2022), 14–24.

[19] Richard J. Chen, Chengkuan Chen, Yicong Li, Tiffany Y. Chen, Andrew D. Trister, et al. 2022. Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16144–16155.

[20] Richard J. Chen, Ming Y. Lu, Wei-Hung Weng, Tiffany Y. Chen, Drew F. K. Williamson, et al. 2021. Multimodal co-attention transformer for survival prediction in gigapixel whole slide images. In *IEEE/CVF International Conference on Computer Vision*. 4015–4025.

[21] Gong Cheng, Xingxing Xie, Junwei Han, Lei Guo, and Gui-Song Xia. 2020. Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13 (2020), 3735–3756. https://doi.org/10.1109/JSTARS.2020.3005403

[22] Shenghua Cheng, Sibo Liu, Jingya Yu, Gong Rao, Yuwei Xiao, et al. 2021. Robust whole slide image analysis for cervical cancer screening using deep learning. *Nature Communications* 12, 1 (2021), 5639.

[23] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. 2018. Model compression and acceleration for deep neural networks: The principles, progress, and challenges. *IEEE Signal Processing Magazine* 35, 1 (2018), 126–136.

[24] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. 2018. Functional map of the world. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[25] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, et al. 2016. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[26] Mickael Cormier, Stefan Wolf, Lars Sommer, Arne Schumann, and Jürgen Beyerer. 2021. Fast pedestrian detection for real-world crowded scenarios on embedded GPU. In *IEEE International Conference on Smart Technologies*. 40–44.
[27] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, et al. (Eds.), Vol. 35. Curran Associates, Inc., 16344–16359. https://proceedings.neurips.cc/paper_files/paper/2022/file/67d57c32e20fd0a7a302cb81d36e40d5-Paper-Conference.pdf
[28] Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, et al. 2018. DeepGlobe 2018: A challenge to parse the earth through satellite images. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 172–181.
[29] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, et al. 2009. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*. 248–255.
[30] Neofytos Dimitriou, Ognjen Arandjelović, and Peter D. Caie. 2019. Deep learning for whole slide image analysis: An overview. *Frontiers in Medicine* 6 (2019). https://doi.org/10.3389/fmed.2019.00264
[31] Lei Ding, Dong Lin, Shaofu Lin, Jing Zhang, Xiaojie Cui, et al. 2022. Looking outside the window: Wide-context transformer for the semantic segmentation of high-resolution remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), 1–13. https://doi.org/10.1109/TGRS.2022.3168697
[32] Mingyu Ding, Xiaochen Lian, Linjie Yang, Peng Wang, Xiaojie Jin, et al. 2021. HR-NAS: Searching efficient high-resolution neural architectures with lightweight transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2982–2992.
[33] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. 2009. Caltech Pedestrians. https://doi.org/10.1109/CVPR.2009.5206631
[34] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. 2016. Adversarial feature learning. *arXiv preprint arXiv:1605.09782* (2016).
[35] Nanqing Dong, Michael Kampffmeyer, Xiaodan Liang, Zeya Wang, Wei Dai, et al. 2018. Reinforced auto-zoom net: Towards accurate and fast breast cancer segmentation in whole-slide images. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. 317–325.
[36] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, et al. 2022. CSWin transformer: A general vision transformer backbone with cross-shaped windows. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12124–12134.
[37] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*. https://openreview.net/forum?id=YicbFdNTTy
[38] Hao Du, Jiashi Feng, and Mengling Feng. 2019. Zoom in to where it matters: A hierarchical graph based model for mammogram analysis. *arXiv preprint arXiv:1912.07517* (2019).
[39] Jiangsu Du, Xin Zhu, Minghua Shen, Yunfei Du, Yutong Lu, et al. 2020. Model parallelism optimization for distributed inference via decoupled CNN structure. *IEEE Transactions on Parallel and Distributed Systems* 32, 7 (2020), 1665–1676.
[40] Jean Duchon. 1977. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive Theory of Functions of Several Variables*. 85–100.
[41] Navid Farahani, Anil V. Parwani, Liron Pantanowitz, et al. 2015. Whole slide imaging in pathology: Advantages, limitations, and emerging perspectives. *Pathology and Laboratory Medicine International* 7, 23-33 (2015), 4321.
[42] Cristiane B. R. Ferreira, Helio Pedrini, Wanderley de Souza Alencar, William D. Ferreira, Thyago Peres Carvalho, et al. 2020. Where's Wally: A gigapixel image study for face recognition in crowds. In *Advances in Visual Computing*. 386–397.
[43] Yasuhiro Fukushima, Yasutaka Fushimi, Takeshi Funaki, Akihiko Sakata, Takuya Hinoda, et al. 2022. Evaluation of moyamoya disease in CT angiography using ultra-high-resolution computed tomography: Application of deep learning reconstruction. *European Journal of Radiology* 151 (2022), 110294. https://doi.org/10.1016/j.ejrad.2022.110294
[44] Jevgenij Gamper, Navid Alemi Koohbanani, Ksenija Benet, Ali Khuram, and Nasir Rajpoot. 2019. PanNuke: An open pan-cancer histology dataset for nuclei instance segmentation and classification. In *European Congress on Digital Pathology*. 11–19.
[45] Guangshuai Gao, Junyu Gao, Qingjie Liu, Qi Wang, and Yunhong Wang. 2020. CNN-based density estimation and crowd counting: A survey. *arXiv preprint arXiv:2003.12783* (2020).
[46] Mingfei Gao, Ruichi Yu, Ang Li, Vlad I. Morariu, and Larry S. Davis. 2018. Dynamic zoom-in network for fast object detection in large images. In *IEEE Conference on Computer Vision and Pattern Recognition*.
[47] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? The KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
[48] Simon Graham, Quoc Dang Vu, Shan E. Ahmed Raza, Ayesha Azam, Yee Wah Tsang, et al. 2019. HoVer-Net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Medical Image Analysis* 58 (2019), 101563.

[49] Jiaqi Gu, Hyoukjun Kwon, Dilin Wang, Wei Ye, Meng Li, et al. 2022. Multi-scale high-resolution vision transformer for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 12094–12103.
[50] Lionel Gueguen, Alex Sergeev, Ben Kadlec, Rosanne Liu, and Jason Yosinski. 2018. Faster neural networks straight from JPEG. *Advances in Neural Information Processing Systems* 31 (2018).
[51] Meng-Hao Guo, Zheng-Ning Liu, Tai-Jiang Mu, Dun Liang, Ralph R. Martin, et al. 2021. Can attention enable MLPs to catch up with CNNs? *Computational Visual Media* 7, 3 (2021), 283–288.
[52] Zayd Mahmoud Hamdi, Melanie Brandmeier, and Christoph Straub. 2019. Forest damage assessment using deep learning on high resolution remote sensing data. *Remote Sensing* 11, 17 (2019), 1976.
[53] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, et al. 2021. Dynamic neural networks: A survey. *arXiv preprint arXiv:2102.04906* (2021).
[54] Haoyu He, Jing Liu, Zizheng Pan, Jianfei Cai, Jing Zhang, et al. 2021. Pruning self-attentions into convolutional layers in single path. *arXiv preprint arXiv:2111.11802* (2021).
[55] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.
[56] Lukas Hedegaard and Alexandros Iosifidis. 2022. Continual inference: A library for efficient online inference with deep neural networks in PyTorch. *arXiv preprint: arXiv:2204.03418* (2022).
[57] David Joon Ho, Dig V. K. Yarlagadda, Timothy M. D'Alfonso, Matthew G. Hanna, Anne Grabenstetter, et al. 2021. Deep multi-magnification networks for multi-class breast cancer image segmentation. *Computerized Medical Imaging and Graphics* 88 (2021), 101866.
[58] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.
[59] James P. Horwath, Dmitri N. Zakharov, Rémi Mégret, and Eric A. Stach. 2020. Understanding important features of deep learning models for segmentation of high-resolution transmission electron microscopy images. *NPJ Computational Materials* 6, 1 (2020), 108.
[60] Di Hu, Lichao Mou, Qingzhong Wang, Junyu Gao, Yuansheng Hua, et al. 2020. Ambient sound helps: Audiovisual crowd counting in extreme conditions. *arXiv preprint arXiv:2005.07097* (2020).
[61] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, et al. 2020. The ApolloScape open dataset for autonomous driving and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 10 (2020), 2702–2719.
[62] Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. 2013. Multi-source multi-scale counting in extremely dense crowd images. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2547–2554.
[63] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, et al. 2018. Composition loss for counting, density map estimation and localization in dense crowds. In *European Conference on Computer Vision*. 532–546.
[64] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. 2015. Spatial transformer networks. *Advances in Neural Information Processing Systems* 28 (2015).
[65] Sajid Javed, Arif Mahmood, Naoufel Werghi, and Nasir Rajpoot. 2019. Deep multiresolution cellular communities for semantic segmentation of multi-gigapixel histology images. In *IEEE/CVF International Conference on Computer Vision Workshops*.
[66] Huiwei Jiang, Min Peng, Yuanjun Zhong, Haofeng Xie, Zemin Hao, et al. 2022. A survey on deep learning-based change detection from high-resolution remote sensing images. *Remote Sensing* 14, 7 (2022).
[67] Shiqi Jiang, Zhiqi Lin, Yuanchun Li, Yuanchao Shu, and Yunxin Liu. 2021. Flexible high-resolution object detection on edge devices with tunable latency. In *Annual International Conference on Mobile Computing and Networking*. 559–572.
[68] Chen Jin, Ryutaro Tanno, Thomy Mertzanidou, Eleftheria Panagiotaki, and Daniel C. Alexander. 2021. Learning to downsample for segmentation of ultra-high resolution images. *arXiv preprint arXiv:2109.11071* (2021).
[69] Anilkumar Katharotiya, Swati Patel, and Mahesh Goyani. 2011. Comparative analysis between DCT & DWT techniques of image compression. *Journal of Information Engineering and Applications* 1, 2 (2011), 9–17.
[70] Afshin Khadangi, Thomas Boudier, and Vijay Rajagopal. 2021. EM-stellar: Benchmarking deep learning for electron microscopy image segmentation. *Bioinformatics* 37, 1 (2021), 97–106.
[71] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.
[72] Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114* (2013).
[73] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.
[74] Daniel C. Koboldt, Robert S. Fulton, Michael D. McLellan, Heather Schmidt, Joelle Kalicki-Veizer, et al. 2012. Comprehensive molecular portraits of human breast tumours. *Nature* 490, 7418 (2012), 61–70.

[75] Anastasios Koulaouzidis, Dimitris Iakovidis, Diana Yung, Emanuele Rondonotti, Uri Kopylov, et al. 2017. KID project: An internet-based digital video atlas of capsule endoscopy for research purposes. *Endoscopy International Open* 5, 6 (2017), E477–E483.

[76] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'12)*. Curran Associates Inc., Red Hook, NY, USA, 1097–1105.

[77] Zhengfeng Lai, Chao Wang, Luca Cerny Oliveira, Brittany N. Dugger, Sen-Ching Cheung, et al. 2021. Joint semi-supervised and active learning for segmentation of gigapixel pathology images with cost-effective labeling. In *IEEE/CVF International Conference on Computer Vision Workshops*. 591–600.

[78] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324. https://doi.org/10.1109/5.726791

[79] Romain Leenhardt, Cynthia Li, Jean-Philippe Le Mouel, Gabriel Rahmi, Jean Christophe Saurin, et al. 2020. CAD-CAP: A 25,000-image database serving the development of artificial intelligence for capsule endoscopy. *Endosc. Int. Open* 8, 3 (2020), E415–E420.

[80] Lingling Li, Xiaohui Guo, Yan Wang, Jingjing Ma, Licheng Jiao, et al. 2022. Region NMS-based deep network for gigapixel level pedestrian detection with two-step cropping. *Neurocomputing* 468 (2022), 482–491.

[81] Mengtian Li, Yu-Xiong Wang, and Deva Ramanan. 2020. Towards streaming perception. In *European Conference on Computer Vision*. 473–488.

[82] Zhexin Li, Tong Yang, Peisong Wang, and Jian Cheng. 2022. Q-ViT: Fully differentiable quantization for vision transformer. *arXiv preprint arXiv:2201.07703* (2022).

[83] Huangjing Lin, Hao Chen, Simon Graham, Qi Dou, Nasir Rajpoot, et al. 2019. Fast ScanNet: Fast and dense analysis of multi-gigapixel whole-slide images for cancer metastasis detection. *IEEE Transactions on Medical Imaging* 38, 8 (2019), 1948–1958.

[84] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L. Curless, Steven M. Seitz, et al. 2021. Real-time high-resolution background matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8762–8771.

[85] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, et al. 2014. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*. 740–755.

[86] Chenchen Liu, Xinyu Weng, and Yadong Mu. 2019. Recurrent attentive zooming for joint crowd counting and precise localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[87] Yun Liu, Krishna Gadepalli, Mohammad Norouzi, George E. Dahl, Timo Kohlberger, et al. 2017. Detecting cancer metastases on gigapixel pathology images. *arXiv preprint arXiv:1703.02442* (2017).

[88] Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

[89] Wenqi Lu, Simon Graham, Mohsin Bilal, Nasir Rajpoot, and Fayyaz Minhas. 2020. Capturing cellular topology in multi-gigapixel pathology images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 260–261.

[90] Rui Ma and Qi Hao. 2021. CS-Fnet: A compressive sampling frequency neural network for simultaneous image compression and recognition. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. 1–6.

[91] Simon Madec, Xiuliang Jin, Hao Lu, Benoit De Solan, Shouyang Liu, et al. 2019. Ear density estimation from high resolution RGB imagery using deep learning technique. *Agricultural and Forest Meteorology* 264 (2019), 225–234.

[92] Dmitrii Marin, Zijian He, Peter Vajda, Priyam Chatterjee, Sam Tsai, et al. 2019. Efficient segmentation: Learning downsampling near semantic boundaries. In *IEEE/CVF International Conference on Computer Vision*. 2131–2141.

[93] Peter M. Roth Martin Koestinger, Paul Wohlhart and Horst Bischof. 2011. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*.

[94] Yoshitomo Matsubara, Marco Levorato, and Francesco Restuccia. 2021. Split computing and early exiting for deep learning applications: Survey and research challenges. *Comput. Surveys* (2021).

[95] C. M. McLeavy, M. H. Chunara, R. J. Gravell, A. Rauf, A. Cushnie, et al. 2021. The future of CT: Deep learning reconstruction. *Clinical Radiology* 76, 6 (2021), 407–415. https://doi.org/10.1016/j.crad.2021.01.010

[96] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. 2018. ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[97] Inês C. Moreira, Igor Amaral, Inês Domingues, António Cardoso, Maria João Cardoso, et al. 2012. INbreast. *Academic Radiology* 19, 2 (2012), 236–248.

[98] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, et al. 2014. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[99] Daniel Müllner. 2011. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378* (2011).

[100] Matteo Pagliardini, Daniele Paliotta, Martin Jaggi, and François Fleuret. 2023. Faster causal attention over large sequences through sparse flash attention. *arXiv preprint arXiv:2306.01160* (2023).

[101] Bharathkumar Ramachandra, Michael J. Jones, and Ranga Raju Vatsavai. 2022. A survey of single-scene video anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 5 (2022), 2293–2312. https://doi.org/10.1109/TPAMI.2020.3040591

[102] Adria Recasens, Petr Kellnhofer, Simon Stent, Wojciech Matusik, and Antonio Torralba. 2018. Learning to zoom: A saliency-based sampling layer for neural networks. In *European Conference on Computer Vision*. 51–66.

[103] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. 779–788.

[104] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems* 28 (2015).

[105] Iain E. Richardson. 2004. *H. 264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. John Wiley & Sons.

[106] Eduardo Rocha Rodrigues, Igor Oliveira, Renato Cunha, and Marco Netto. 2018. DeepDownscale: A deep learning strategy for high-resolution weather forecast. In *IEEE International Conference on e-Science*. 415–422.

[107] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. 2016. The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[108] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, et al. 2015. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252.

[109] Usman Sajid, Hasan Sajid, Hongcheng Wang, and Guanghui Wang. 2020. ZoomCount: A zooming mechanism for crowd counting in static images. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 10 (2020), 3499–3512.

[110] Randy Sargent, Chris Bartley, Paul Dille, Jeff Keller, Illah Nourbakhsh, et al. 2010. Timelapse GigaPan: Capturing, sharing, and exploring timelapse gigapixel imagery. In *Fine International Conference on Gigapixel Imaging for Science*.

[111] Joanne D. Schuijf, João A. C. Lima, Kirsten L. Boedeker, Hidenobu Takagi, Ryoichi Tanaka, et al. 2022. CT imaging with ultra-high-resolution: Opportunities for cardiovascular imaging in clinical practice. *Journal of Cardiovascular Computed Tomography* 16, 5 (2022), 388–396. https://doi.org/10.1016/j.jcct.2022.02.003

[112] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, et al. 2013. OverFeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229* (2013).

[113] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, et al. 2019. Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053* (2019).

[114] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[115] Vishwanath A. Sindagi, Rajeev Yasarla, and Vishal M. Patel. 2020. JHU-CROWD++: Large-scale crowd counting dataset and a benchmark method. *Technical Report* (2020).

[116] Qingyu Song, Changan Wang, Yabiao Wang, Ying Tai, Chengjie Wang, et al. 2021. To choose or to fuse? Scale selection for crowd counting. In *AAAI Conference on Artificial Intelligence*, Vol. 35. 2576–2583.

[117] Andreas Specker, Lennart Moritz, Mickael Cormier, and Jürgen Beyerer. 2022. Fast and lightweight online person search for large-scale surveillance systems. In *IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*. 570–580.

[118] Chetan L. Srinidhi, Ozan Ciga, and Anne L. Martel. 2021. Deep neural network models for computational histopathology: A survey. *Medical Image Analysis* 67 (2021), 101813.

[119] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[120] Xian Sun, Peijin Wang, Zhiyuan Yan, Feng Xu, Ruiping Wang, et al. 2022. FAIR1M: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing* 184 (2022), 116–130. https://doi.org/10.1016/j.isprsjprs.2021.12.004

[121] Mingxing Tan and Quoc Le. 2019. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*. 6105–6114.

[122] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732* (2020).
[123] David Tellez, Geert Litjens, Jeroen van der Laak, and Francesco Ciompi. 2021. Neural image compression for gigapixel histopathology image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 2 (2021), 567–578.
[124] Chittesh Thavamani, Mengtian Li, Nicolas Cebron, and Deva Ramanan. 2021. FOVEA: Foveated image magnification for autonomous navigation. In *IEEE/CVF International Conference on Computer Vision*. 15539–15548.
[125] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, et al. 2016. YFCC100M: The new data in multimedia research. *Commun. ACM* 59, 2 (2016), 64–73.
[126] Eric Thomson, Mark Harfouche, Pavan Konda, Catherine W. Seitz, Kanghyun Kim, et al. 2021. Gigapixel behavioral and neural activity imaging with a novel multi-camera array microscope. *bioRxiv* (2021).
[127] Danai Triantafyllidou, Paraskevi Nousi, and Anastasios Tefas. 2018. Fast deep convolutional face detection in the wild exploiting hard sample mining. *Big Data Research* 11 (2018), 65–76.
[128] Nelson Zange Tsaku, Sai Chandra Kosaraju, Tasmia Aqila, Mohammad Masum, Dae Hyun Song, et al. 2019. Texture-based deep learning for effective histopathological cancer image classification. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 973–977.
[129] Maria Tzelepi and Anastasios Tefas. 2020. Class-specific discriminant regularization in real-time deep CNN models for binary classification problems. *Neural Processing Letters* 51, 2 (2020), 1989–2005.
[130] Maria Tzelepi and Anastasios Tefas. 2020. Improving the performance of lightweight CNNs for binary classification using quadratic mutual information regularization. *Pattern Recognition* 106 (2020), 107407.
[131] Maria Tzelepi and Anastasios Tefas. 2021. Graph embedded convolutional neural networks in human crowd detection for drone flight safety. *IEEE Transactions on Emerging Topics in Computational Intelligence* 5, 2 (2021), 191–204.
[132] Burak Uzkent and Stefano Ermon. 2020. Learning when and where to zoom with deep reinforcement learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
[133] M. Vakalopoulou, K. Karantzalos, N. Komodakis, and N. Paragios. 2015. Building detection in very high resolution multispectral data with deep learning features. In *IEEE International Geoscience and Remote Sensing Symposium*. 1873–1876.
[134] Jeroen van der Laak, Geert Litjens, and Francesco Ciompi. 2021. Deep learning in histopathology: The path to the clinic. *Nature Medicine* 27, 5 (2021), 775–784.
[135] Adam Van Etten. 2018. You only look twice: Rapid multi-scale object detection in satellite imagery. *arXiv preprint arXiv:1805.09512* (2018).
[136] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, et al. 2018. Graph attention networks. In *International Conference on Learning Representations*.
[137] Mitko Veta, Yujing J. Heng, Nikolas Stathonikos, Babak Ehteshami Bejnordi, Francisco Beca, et al. 2019. Predicting breast tumor proliferation from whole-slide images: The TUPAC16 challenge. *Medical Image Analysis* 54 (2019), 111–121.
[138] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, et al. 2021. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 10 (2021), 3349–3364. https://doi.org/10.1109/TPAMI.2020.2983686
[139] Kun Wang, Xiaohong Zhang, and Sheng Huang. 2019. KGZNet: Knowledge-guided deep zoom neural networks for thoracic disease classification. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 1396–1401.
[140] Linyan Wang, Longqian Ding, Zhifang Liu, Lingling Sun, Lirong Chen, et al. 2020. Automated identification of malignancy in whole-slide pathological images: Identification of eyelid malignant melanoma in gigapixel pathological slides using deep learning. *British Journal of Ophthalmology* 104, 3 (2020), 318–323.
[141] Luyuan Wang and Yankui Sun. 2022. Image classification using convolutional neural network with wavelet domain inputs. *IET Image Processing* 16, 8 (2022), 2037–2048.
[142] Qi Wang, Junyu Gao, Wei Lin, and Xuelong Li. 2020. NWPU-crowd: A large-scale benchmark for crowd counting and localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 6 (2020), 2141–2149.
[143] Shiyao Wang, Hongchao Lu, and Zhidong Deng. 2019. Fast object detection in compressed video. In *IEEE/CVF International Conference on Computer Vision*.
[144] Xueyang Wang, Xiya Zhang, Yinheng Zhu, Yuchen Guo, Xiaoyun Yuan, et al. 2020. PANDA: A gigapixel-level human-centric video dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3268–3278.
[145] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, et al. 2020. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *Computer Vision and Image Understanding* 193 (2020), 102907.
[146] Lilian Weng and Greg Brockman. 2022. Techniques for Training Large Neural Networks. https://openai.com/blog/techniques-for-training-large-neural-networks/
[147] Chensu Xie, Hassan Muhammad, Chad M. Vanderbilt, Raul Caso, Dig Vijay Kumar Yarlagadda, et al. 2020. Beyond classification: Whole slide tissue histopathology analysis by end-to-end part learning. In *Conference on Medical Imaging with Deep Learning*. 843–856.

[148] Xiaohan Xing, Yixuan Yuan, and Max Q.-H. Meng. 2020. Zoom in lesions for better diagnosis: Attention guided deformation network for WCE image classification. *IEEE Transactions on Medical Imaging* 39, 12 (2020), 4047–4059.
[149] Chenfeng Xu, Kai Qiu, Jianlong Fu, Song Bai, Yongchao Xu, et al. 2019. Learn to scale: Generating multipolar normalized density maps for crowd counting. In *IEEE/CVF International Conference on Computer Vision*.
[150] Kai Xu, Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, et al. 2020. Learning in the frequency domain. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1740–1749.
[151] Yongyang Xu, Zhong Xie, Yaxing Feng, and Zhanlong Chen. 2018. Road extraction from high-resolution remote sensing imagery using deep learning. *Remote Sensing* 10, 9 (2018), 1461.
[152] Fei Yang, Luis Herranz, Yongmei Cheng, and Mikhail G. Mozerov. 2021. Slimmable compressive autoencoders for practical neural image compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4998–5007.
[153] Shuyi Yang, Longquan Jiang, Zhuoqun Cao, Liya Wang, Jiawang Cao, et al. 2020. Deep learning for detecting corona virus disease 2019 (COVID-19) on high-resolution computed tomography: A pilot study. *Annals of Translational Medicine* 8, 7 (2020), 450–450.
[154] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. 2016. WIDER FACE: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition*.
[155] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, et al. 2020. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
[156] Qihang Yu, Yingda Xia, Yutong Bai, Yongyi Lu, Alan L. Yuille, et al. 2021. Glance-and-gaze vision transformer. In *Advances in Neural Information Processing Systems*, Vol. 34. 12992–13003.
[157] Xiaoyun Yuan, Lu Fang, Qionghai Dai, David J. Brady, and Yebin Liu. 2017. Multiscale gigapixel video: A cross resolution image matching and warping approach. In *IEEE International Conference on Computational Photography*. 1–9.
[158] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, et al. 2021. HRFormer: High-resolution vision transformer for dense predict. In *Advances in Neural Information Processing Systems*, Vol. 34. 7281–7293.
[159] Jinnian Zhang, Houwen Peng, Kan Wu, Mengchen Liu, Bin Xiao, et al. 2022. MiniViT: Compressing vision transformers with weight multiplexing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 12145–12154.
[160] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, et al. 2021. Multi-scale vision Longformer: A new vision transformer for high-resolution image encoding. In *IEEE/CVF International Conference on Computer Vision*. 2998–3008.
[161] Xin Zhang, Liangxiu Han, Lianghao Han, and Liang Zhu. 2020. How well do deep learning-based methods for land cover classification and object detection perform on high resolution remote sensing imagery? *Remote Sensing* 12, 3 (2020).
[162] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. 2016. Single-image crowd counting via multi-column convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*. 589–597.
[163] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. 2014. Facial landmark detection by deep multi-task learning. In *European Conference on Computer Vision*. 94–108.
[164] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. 2018. ICNet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
[165] Rui Zhao, Zhenwei Shi, and Zhengxia Zou. 2022. High-resolution remote sensing image captioning based on structured attention. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), 1–14. https://doi.org/10.1109/TGRS.2021.3070383
[166] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, et al. 2017. Scene parsing through ADE20K dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*. 633–641.
[167] Joey Tianyi Zhou, Le Zhang, Du Jiawei, Xi Peng, Zhiwen Fang, et al. 2021. Locality-aware crowd counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 7 (2021), 3602–3613.

Received 6 December 2022; revised 19 December 2023; accepted 31 January 2024