International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 11, No.1 (Mar-2022)

https://dx.doi.org/10.12785/ijcds/110180

U-Net Convolutional Networks Performance Based on Software-Hardware Cooperation Parameters: A Review

Ula T. Salim[1], **Fakhrulddin H. Ali**[1] **and Shefa A. Dawwd**[1]

[1]*Computer Engineering Department, Univ. of Mosul, Mosul, Iraq*

Received 13 Feb. 2021, Revised 1 Mar. 2022, Accepted 9 Mar. 2022, Published 31 Mar. 2022

Abstract: In recent years, the continued advances of the deep learning as a part of machine learning produces an accuracy which resembles the people's performance in processing various challenges of the real world. U-Net, as convolutional neural network (CNN), is one of the deep learning architectures that have been utilized to perform segmentation in several applications. The flexible design of the U-Net, utilizing the data augmentation approach, has been contributed in the achievement of successful predictive results for different image sizes particularly with training few datasets implementing efficient computations. However, the accuracy of one application may need adding additional improvement on the basic U-Net, due to the encoding and decoding processes, which causes some information loss. Another challenge is that the training and testing of a large amount of labeled data is a very computation-intensive process which needs to be minimized. Therefore, this review aims to describe the basic building blocks of 2D U-Net architecture, addressing its challenges and then it explains the most important cooperation issue between software and hardware. Finally it introduces important conclusions with considerable remarks that may help in selecting a suitable model.

Keywords: U-Net, Accuracy, Training, Segmentation, Software, Hardware, Performance

1. INTRODUCTION

The adoption of computer vision systems to interpret, analyze, segment and visualize pixels information in a specific application is extremely important. It requires intelligent technologies that are able to adapt with many variables, such as the type of task and environmental conditions for data collection. Some of the deep learning segmentation algorithms are fully convolutional network (FCN)[1], SegNet[2], DeepLabv3[3], but the most popular one is the U-Net which was designed by Ronneberger et al. [4] to segment the biomedical images. U-Net and its variants have been helped in reducing the amount of time required by the experts in numerous medical diagnoses in diseases like liver[5], cardiac[6,7], lung[8], brain[9,10,11,12,13,14,15,16], retina[17,18,19,20]. In addition to that, it became the successful component for other research tasks, such as robotic vision in surgery to segment surgical tool[21,22,23], remote sensing [24,25,26,27], detecting the markings of the road lanes to support autonomous driving [28] etc.

U-Net is an extension of a fully convolutional network (FCN) which includes large feature channels in the up sampling section helping the designed network to transfer context data to layers of higher resolution. Another modification is to use a stack of convolutional layers rather than dense layers. The network of U-Net learns the input images in end-to-end and pixel-to-pixel way through fusing three specific functional components: encoder, decoder, short and long connections generating a Ushaped structure as illustrated in a figure (1). The encoder pathway is the same as in the traditional convolutional network which is composed of repeated blocks. Every block consists of implementing two alternative 3x3 unpadded convolutions and a rectified linear unit (ReLU), and then passes the results to a down sampling layer applying a max pooling operation of a 2×2 size with a stride equils to 2. Through every down sampling level of the encoder, the spatial information is divided by two while the number of feature channels is doubled by two. On the contrary, approximation symmetric decoder path is a repeated series of up sampling level where each one divides the number of feature channels by two and doubles the spatial information by two. The decoder incorporates the operation of both of the features and spatial information via a 2x2 up-convolution then the result is concatenated with cropped features from the corresponding encoder layers, then passes the outcome to two alternative 3x3 convolutions and ReLU. The last layer employs a sigmoid activation function. Finally, each one of 64-component feature vector is mapped to the desired number of classes using a 1x1 convolution layer. The final network includes 23 convolutional layers [4].

E-mail address: ula.tariq@uomosul.edu.iq, fhazaa@uomosul.edu.iq, shefa.dawwd@uomosul.edu.iq

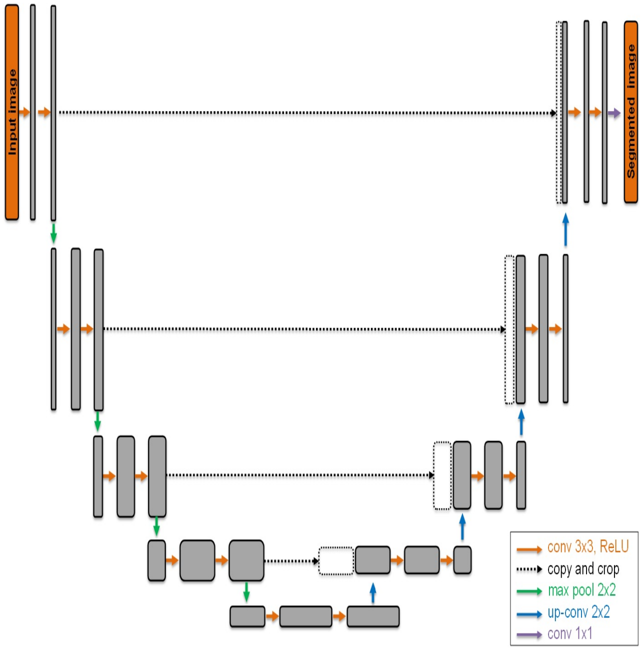

Figure 1. 2D U-Net architecture

Although the U-Net architecture gives a good generalization performance and an accepted training time when executed over GPU device [4], many researchers have noted that it lacks some aspects for example, how the different computations complexity influence on the performance accuracy as well as which component is related to the run time. So to overcome that, they suggested various extensions as in references [5-64]. For both of training and testing phases they exploited software optimization with the development of processor's device technologies such as GPU[5,9,29], FPGA[25,30], Google TPU[31,32], and mobile as embedded systems[33].

In the review proposed in this paper, the performance parameters of U-Net with the fundamental components are explained. Also, suitable suggestions for resolving variant challenges are demonstrated.

The main contribution of this review is as follows:
1. State the main challenges and computations intensive affecting the U-Net performance.
2. Explore the most important developments and updates that have been made to make the 2D U-Net architecture more efficient.

In addition to this introduction, this review paper includes four other sections. Section 2 presents the main challenges that reflect the performance of the deep learning models(U-Net) architecture. Section 3 discusses the performance improvements through the cooperation of one parameter or more of image datasets attributes, model parameters, hyper parameters and implementation stack. Section 4 summarizes the major improvements of the UNet. Finally, some conclusions are made in section 5.

2. CHALLENGES OF DEEP LEARNING MODELS

This section addresses the major challenges imposed by the performance which restrict the implementation of an application using the deep learning models (such as baseline 2D U-Net).

- **Training dataset size:** As a general rule, referring to the large amount of dataset required for training a deep neural models, consequently depends on the application in hand.
- **Accuracy:** The ratio of the number of correctly predicted samples to the total number of input samples is expressed as a percentage. The high accuracy, the better the performance.
- **FLOPs:** Number of floating point operations required by an algorithm/model.
- **Memory size limitation:** The available memory is inadequate storage space for handling the amount of available datasets.
- **Time:** Any deep learning model contains training and testing time. In both, the shorter the time is the better performance especially for real time applications.Training time is the amount of time taken to train the model on datasets to obtain specific accuracy during the training process. Testing time is the amount of time taken when applying one batch dataset on the trained model to produce predication result for the real time applications.
- **Latency:** It is the time interval between the beginning and end of calculation.
- **Throughput:** The number of input or the size of data which can be processed for every unit time.
- **Power consumption:** The energy consumed per unit time.
- **Energy efficiency:** The energy consumed for every data point.
- **Model compression:** It is the reduction of one or more of the followings: in number of filters, number of parameters, number of bits, number of convolutional layers and network depth.
- **Object appearance:** Objects may vary in location, shape, size, and noise level.
- **Class imbalance:** Refers to the irregular distribution of classes within the training dataset.
- **Overfitting:** A state that appears when the training dataset has limited parameters which leads to memorizing the noise instead of learning the data, so the error will be high and the performance will be decreased.
- **Generalization:** The ability of using the architecture with new data collected from different sources.

3. AN OVERVIEW OF U-NET AND ITS VARIANTS ARCHITECTURES

To alleviate U-Net challenges and problems, many variants with a lot of ideas have been proposed for enhancing baseline blocks with different ways as illustrated in figure(2).

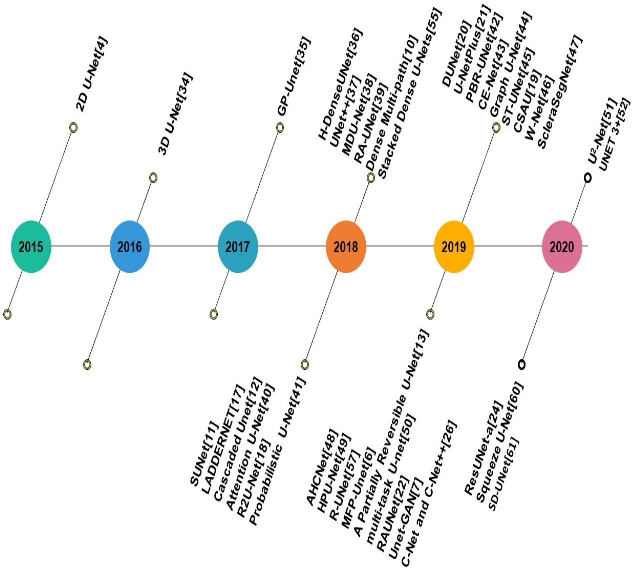

Figure 2. U-Net and some of its evolution variants architectures

Cicek et al. [34] presented 3D U-Net network version to segment volumetric images from some annotations. Semi-automated and fully-automatically are two segmentation setups that are applied on the kidney which has complex and high variable 3D structure. In a Semi-automated, the user annotating a number of slices in each volume required to be segmented, then the model trains on these few annotations, providing a 3D dense segmentation, while in the Fully-automated; The model is trained with annotated-slices from a representative training group and can applied on new annotated volumes. Dubost et al.[35] proposed a GP-Unet to deal with global image-level labels. It trains a 3D regression network with merging a FCN with global pooling. The network employed in brain MRI for detecting enlarged perivascular spaces, achieving high sensitivity. Li et al. [36] tackled the issues of 2D-convolutions that disregard the 3D contexts and the 3D-convolutions that requires high computational cost. They suggested a new hybrid densely connected UNet, named H-DenseUNet. It is used for extracting both of intra-slice and inter-slice features, which are connected and improved by utilizing a hybrid feature fusion layer. The designed model is trained in end-to-end and evaluated to segment Liver and Tumor datasets. Zhou et al. [37] introduced an efficient nested UNet (UNet++) to segment medical images. UNet++ exploits the redesigning of skip-connections to reduce Disparity between feature maps for the sub-blocks of contacting and expanding. Additionally, it utilized the deep supervision to improved the accuracy of segmentation. Zhang et al. [38] proposed a Multi-scale Dense U-Net (MDU-Net) based on the quantization. It enhances the information flow in the encoder, decoder and between them through multi-scale dense pathways. Furthermore, it decreases the overfitting in the architecture. Jin et al. [39] extracted pixel-to-pixel 3D structures using a 3D hybrid residual attention-aware(RAUNet), where attention models adaptively focus on particular parts and the residual to allow for deepest network as well as resolving gradient vanishing problem, while U-Net takes multi-scale attention information then combines low level features with high-level features. Good results achieved when the suggested model is tested on datasets of 3DIRCADb and MICCAI 2017 of segmenting Liver Tumor. Also, when was applied on the BraTS2018 and BraTS2017 to segment brain tumor. Dolz et al. [10] proposed new Dense Multi-path U-Net for multimodality segmentation. It has three main ways including, first each modality processed in a specific path and provides better data utilization, second the network is connected densely to help model which allows model to train the scale in which the modalities must be tackled and merged and finally, the inception modules were extended with dilated convolutions to tackle the variability in size. On the other hand, it handled volumes as series of 2D slides leading to discard 3D context. Cl'erigues et al.[11] proposed Stroke U-Net (SUNet) to segment stroke lesion employing multimodal images, another benefit for SUNet, it as solves the problem of class imbalance utilizing tiny patches with balanced training patch sampling technieque and the dynamically weighted loss function. Zhuang [17] proposed LADDERNET. It is including a chain of several U-Nets and the skip-connections helps LADDERNET in including several paths to passing the information. Another invention is modifying the residual modules with sharing the same weights between two convolutional layers within one module, which produces a reduction in the number of parameters. The proposed method was tested to segment blood vessel in retinal images. Lachinov et al. [12] introduced Cascaded UNet to automatically segmenting brain tumor and handled multimodal MRI inputs while frequently refining segmentation outputs that results from a prior stage. Oktay et al. [40] suggested Attention U-Net with new attention gate (AG) model which focuses on object structures of changing shape and sizes while maintaining computational efficiency. Alom et al. [18] suggested a Recurrent U-Net (RU-Net) and Recurrent Residual U-Net (R2U-Net). The two architectures have advantages of achieving better accuracy compared to U-net with same count of parameters and being suited for deep models. Both models are used to test three segmentation datasets, including blood vessel, lung, and skin cancer lesion.Kohl et al. [41] suggested a probabilistic U-Net for inherent ambiguities. The proposed segmentation model is a generative and depends on a merging of a U-Net network and the conditional variational autoencoder, which is able to efficiently generating an infinite number of reasonable hypotheses. Hasan et al. [21] modified the baseline U-Net and called UNetPlus which is used for tracking the position of the surgical instruments. UNetPlus includes a prior trained encoder unit with batch normalization which accelerates converges. The decoder unit is re-built by using interpolation of type nearest-neighbor (NN) instead of the transposed convolution. Li et al.[42] proposed a probabilistic-map-guided bi-directional recurrent UNet referred to as PBR-UNet, which solves the loss of spatial data in 2D ways and 3D computation-intensive cost. PBRUNet merges intra-slice data with the probabilistic adjacent maps to construct the local 3D hybrid regularization scheme, followed by a bi-directional recurrent unit optimization.

To solve the resulting information losses of series pooling and stride convolution in U-Net architecture, Gu et al. [43] proposed context encoder network (CE-Net) for capturing further high level information while maintaining spatial information for 2D medical images segmentation. CE-Net has three blocks, including a feature-encoder block, a context-extractor and a feature- decoder block. The network was applied on various tasks,including detection of retinal vessel ,segmenting layer of retinal OCT, segmenting of optic disc, segmenting of cell contour, lung segmentation. Gao et al. [44] suggested Graph U-Net for the issue of how to represent the learning of graph data. Graph U-Net includes graph pooling (gPool) and graph unpooling (gUnpool) and achieved better results, as they stated, for both of graph and node classification applications. Yu et al. [45] designed new multi-scale Spatial-Temporal U-Net called ST-UNet for tasks modeling of the graph-structured time series, combining multi-granularity graph convolution(ST-Pool and STUnpool) with dilated recurrent skip path connections via the U-Net design. Additionally, ST-UNet provides the trade-off between capacity and efficiency with regard to flexibility. Li et al. [19] evolved a connection sensitive attention UNet referred to as CSAU to accurately segment retinal vessel. It enhances the accuracy at pixel level as well as the interest of topology structures through both of connection sensitive loss with attention gate. Mehta and Valloli [46] proposed W-Net to count the crowd and estimate Density map. W-Net contains a reinforcement decoding section which helps the model to converge faster and also generates density maps with high value of SSIM index. Wang et al. [47] proposed new sclera segmentation architecture referred to as ScleraSegNet. It is built from enhanced U-Net model with adding attention blocks between the encoding unit and the decoding unit at the final level of U-Net to learn more discriminative features. Jiang et al. [48] designed a cascaded deployment of an Attention Hybrid Connection Network (AHCNet) to segment liver tumor in CT images. It merges the soft with hard attention technique and short and long connections to realize effective feature extraction and fusion. Kohl et al. [49] proposed a Hierarchical Probabilistic U-Net called as HPU-Net to model the Multi-Scale Ambiguities. It integrates with a conditional variational auto-encoder (cVAE) to provide the flexibility for learning the complicate structured distributions via different scales. Jin et al. [20] suggested Deformable UNet termed DUNet which is used for extracting context information and enabling precise position by integrating both of low-level feature maps and high-level ones. DUNet uses deformable convolution module instead of UNet's convolution layers, DUNet is capable of capturing the retinal blood vessels that appears at different scales and shapes. For training and testing the model,DRIVE, $CHASE\_DB1$ DB1 and STARE are employed. Results illustrated the segmentation of retinal vessel using DUNet outperforms with an accuracy of 0.9697/0.9724/0.9722 and 0.9856/0.9863/0.9868 as AUC score. Moradi et al. [6] introduced new multi-feature pyramid U-net architecture called (MFP-Unet) to segment LV (Left Ventricle) in the echocardiography images. MFP-Unet is based on merging the pyramid of feature and the dilated convolutional filters and concatenating the feature maps in all decoder's levels. Ke et al. [50] proposed multi-task U-net with lazy labels, which is applied to perform the segmentation on microscopy images. The model provides accurate results and is applicable on the images that have poor contrast at the boundaries of object. Ni et al. [22] processed specular reflection and class imbalance problems by proposing new Residual Attention U-Net named RAUNet architecture used for segmenting cataract surgical tool. Cata7 dataset was built to evaluate the proposed model, the resulting performance is 97.71% of a mean Dice and the mean IOU is 95.62%. Yan et al. [7] proposed U-Net generative adversarial network term Unet-GAN as a generic framework to handle gathering images from different sources and vendors. Cardiac cine MRI from three main vendors (Philips, GE, and Siemens) is used as an example, showing a significant in enhancement for a segmentation task. Diakogiannis et al. [24] introduced a framework includes new learning architecture ResUNet-a with Tanimoto as new dice loss to label highly resolution images. ResUNet-a employs a baseline U-Net integrated with a connections of residual type, atrous convolutions, and pyramid scene parsing as pooling. The proposed framework was evaluated to segment the dataset of ISPRS 2D Potsdam. Results are a competitive outperforms with 92.9% as an average F1 score over all classes of a best model. Zhou et al. [8] introduced a U-Net incorporating attention technique. The proposed network is used for segmenting a dataset of a CT image of a COVID-19, resulting Dice coefficient of 83.1%, Sensitivity of, 86.7% and 99.3% as Specificity. Furthermore the segmentation time of one CT slice is 0.29 sec. Qin et al. [51] proposed U2-Net for silent objects detection. The designed model with using ReSidual U-modules helps in capturing more information from various scales in addition to rising the depth of the designed network without leading to increase the computation cost. Two models are instantiated, including a large model U2-Net with (176.3 MB and 30 FPS) and a small model U2-Net+ with (4.7 MB, 40 FPS). Huang et al. [52] combined multi-scales features using UNET 3+. This method is useful for the organs which may appear at different scales. Additionally, it enhances the efficiency of the computation due to ability of decreasing the number network parameters. The classification guided block with a hybrid loss function produced an accurate locatization and boundary improved segmentation map. The evalutations on datasets of a liver and a spleen show that the model of UNet 3+ outperforms over related works.

Despite of most exiting U-Net researches are striving to have good accuracy and minimum loss especially with medical applications, however, the selection of training model configuration was imposed by the concerts of simplicity and minimum running time. Commonly, data parallelism and model parallelism approaches are applied to perform parallel training. Data parallelism iterates the network model and operates a divided batch on multiple devices. This type cannot minimize model's memory for each device or handle the memory issue for big models. To resolve this, Model parallelism divides a network model to multiple sections. However, it needs good design to reduce overheads. Therefore, Oyama et al. [5] introduced scalable hybrid-parallel algorithm to train big size 3D U-Net model. It handles the challenges of computation, memory, and I/O achieving a speedup of 1.42 when employing 512 GPUs in compared with 256 GPUs.

Currently, there are other attempts to reduce the complexity of U-Net variants of huge number of parameters and operations through achieving trade-off between various parameters.

Venkataramani et al. [53] decreased the inherent complexity of model through training with file sharing. The method find a small group of filters with merging coefficients to construct filter in each one of convolutional layer during training time, leading to reduce parameter number required to be trained. The method was considered the segmentation problem of 3D lung-nodule in a CT images, producing good experimental results at few number of training data. Hu et al. [54] proposed 2.5D segmentation to estimate cancer's area in MRI images. 2D patches are produced from volumetric MRI images at three orthogonal orientations. The 2.5D network has realized better performance than the basic 2D U-net. Although the accuracy of 2.5D segmentation is less than 3D U-net model, it superiors over 3D U-net with computation efficiency. Imai et al. [14] dealt with the limitation of GPU memory, where 3D U-Net was trained with full resolution images of size 192 × 192 × 192 voxels in the dataset of brain tumor. The output result, included a reduction of 17.1% in communication overhead which is the most important issue. Also, the mean Dice coefficient is improved, by a 4.48% to detect a total tumor sub region and a 5.32 % for detecting tumor core sub region compared to a patch method at patches of size of 128 × 128 × 128 voxels. The overall acceleration time of the training was 3.53x. Guo et al. [55] proposed stack dense U-Nets for localizing facial landmark in the images. The designed model has a channel aggregation module and a scale aggregation network structure to enhance model's capacity without increasing the size of models. Another benefit was exploiting the deformable convolution and coherent loss to make the invariant with random face input images. Heinrich et al. [33] suggested new approach called Ternary Net to accelerate model inference.It uses activations and trainable weights, also it employs sparse and binary kernels as ternary convolutions instead of floating point matrix multiplications. Ternary Net reported 10-fold reduction in memory needs and a speedup of 10x. Mangalam and Salzamann [56] studied the employing of knowledge distillation for compressing U-Net. This was done by modifying the U-net model to include batch normalization as well as a class re-weighting. The resulted model is used to segment a biomedical image, achieving a reduction of 1000x with accuracy closer to baseline UNet. Liu et al. [30] optimized U-Net segmentation as CNN algorithm by sharing the memory and exploration of the design space to made the parallelism parameters optimal and quantizing the data. The model is applied on Cityscapes Dataset of real time scene segmentation and evaluated on Zynq ZC706 kit. The results are 107 GOPS and 0.12 GOPS/DSP at a quantization of 16-bit, which supporting upto 17 fps for an image inputs of size of 512x512 and with a 9.6W power consumption. Isunuri and Kakarla [9] proposed an optimized U-Net (OU-Net) with an adaptive thresholding. OU-Net employes one convolution layer every level, this led to reduce the cost of computation and produces a fast segmentation for the brain. Brügger et al. [13] proposed a partially reversible U-Net architecture which decreases memory requirements significantly for volumetric images. The proposed model is eliminates the requirement for store activations in the backpropagation. The memory savings is demonstrated on the dataset of BraTS challenge. Zhou and Yang [23] focused on the effect of normalization in the training UNet model to use the semantic segmentation in 2D biomedical field. Four types of normalization, including Batch Normalization (BN), Instance Normalization (IN), Layer Normalization (LN), and Group Normalization (GN) are compared and the validated on the Right Ventricle (RV), Left Ventricle (LV) and aorta, datasets. The reported results demonstrated IN or a GN with high group number provides higher accuracy. Liu and Luk [25] proposed a uniform architecture as hardware accelerator to achieve real time RSI segmentation. The developed architecture is implementing both of convolution and de-convolution efficiently and is optimized using different parallelism levels and layer fusion. The architecture is implemented on Intel's Arria 10 platform, producing low latency of 17.4 ms and high throughput of 1578 GOPS. Bahl et al [26] proposed lightweight, adaptable, and high-accuracy architecture, which is suitable to work on low power bounded devices. The proposed solution is tested to perform binary segmentation on remote sensing images, especially to extract clouds and trees from the datasets of RGB satellite images. Peng et al. [27] proposed new end-to-end way built from enhanced UNet++ with a deep supervision (DS) technique used for Change Detection (CD) in a datasets of (VHR) satellite images. Wang et al. [57] presented new recurrent U-Net architecture which maintains the compactness of the baseline U-Net and can work in environments with the training data volume and computational power being bounded. The architecture shows its effectiveness in segmenting various applications, including hand, retina vessel, and road. AskariHemmat et al. [58] introduced quantization model to decrease memory consumption while preserving accuracy. The model with a 4 bits and 6 bits for both of weight and activations respectively is applied to segment three datasets of EM, GM and NIH, achieving a reduction of 8 fold in memory needs at loosing 2:09% , 0:57% , and 2:21%, in dice score, for NIH, GM, and EM. Chiley et al. [59] trained the U-Net neural models by using a robust normalizer named Online normalization to segment images. Pati et al. [29] studied the effect of inference accelerators on the choice of hardware. Various configurations are applied to determine the suitable hardware for processing the images model in healthcare field constraints. For the medical images with a high resolution, Hou et al. [31] used spatial partitioning to deal with memory limitations. The method is helped in training a 3D U-Net on a CT scans of a resolution of 512×512×512, without resulting additional computational overhead. Civit et al. [32] studied the generalization of UNet architectures to resolve the image segmentation issue in the cloud. The generalized model is used for segmenting the Optic Disc and Cup that may be employed as application in glaucoma detection. (RIM-One V3,DRISHTI and DRIONS) are three public image datasets which are combined to obtain a good performance for independent image acquisitions. Ojika et al. [15] analyzed multimodal brain tumor and used a 3D U-Net model to segment a 3D images. Also, they handled the training of memory intensive models by employing a server system with large memory size of 1TB. Niepceron et al. [16] proposed fully-automatic brain tumor segmentation implemented with compressed model and low cost GPU embedded computing such as Nvidia Jetson AGX Xavier (JAX). The compression model coupled depthwise separable as convolution with Independent-Component. On the other hand, the characteristics of the JAX are power consuming, weight as well as it contains a modular scalable architecture appropriate for real time tasks. However, for deep learning applications memory and computation costs are necessary to be low for limited resources devices. For instance, Beheshti and Johnsson [60] proposed an efficient energy and memory architecture named Squeeze U-Net which is suitable for real time mobile utilization. The designed model is a combination of SqueezeNet and U-Net, where the design fire module of SqueezeNet is used in both of the encoder and decoder U-Net's paths. It is evaluated on a datasets of CamVid road Scenes, the results show it preserved the accuracy and it is faster than U-Net by 17% and 52% in both inference and training using GPU. Further, it realized a reduction of 3.2X in MACs and 12X in model size to 32MB. Gadosey et al. [61] introduced Stripped-Down UNet architecture, which is named SD-UNet, as a very fast, small with efficient computation used for devices that have bounded computational resources. The SD-UNet network employed layers with depthwise separable convolution but it degraded accuracy and to solve this a combination of weight standardization with group normalization is used. SD-UNet segments the datasets of neuronal structures(ISBI) challenge and brain tumor challenge (MSD), achieving comparable results. Compared with baseline U-Net, SD-UNet has three main benefits including a reduction of 23x in model size, a fewer parameters less than by 8x, and 8x in FLOPs. Vaze et al. [62] handled the issue of real time ultrasound implementation on CPUs. The proposed method adapting U-Net with thin CNN, separable convolution layers, and knowledge distillation as three efficient train techniques used to reduce the needs for large memory space and accelerates the inference running on the CPUs while preserving accuracy. The proposed network segments nerve in the 2D ultrasound images producing a speed up of 9x over baseline U-Net on a CPU (at a processing rate of 30 fps) and a reduction space of 420x in memory. Joardar et al. [63] merged the benefit of ReRAM and GPU, resulting GRAMARCH architecture with high features, such as accelerating both of a matrix and vector operations that are used in the training, minimize the total communication when mapping the layers of deep neural network (DNN) on it and building 3D NoC to move the data with high throughput. Furthermore, it is able to execute any layer in the DNN and it handles loss accuracy problem that results from low precision calculations using ReRAMs.

Figure (3) shows how U-Net architecture was extended to new types via linking several processes and mathematic operations which are described as major four sub-sections: image datasets attributes, model parameters, hyper parameters and implementation stack, where every section performs a specific function.

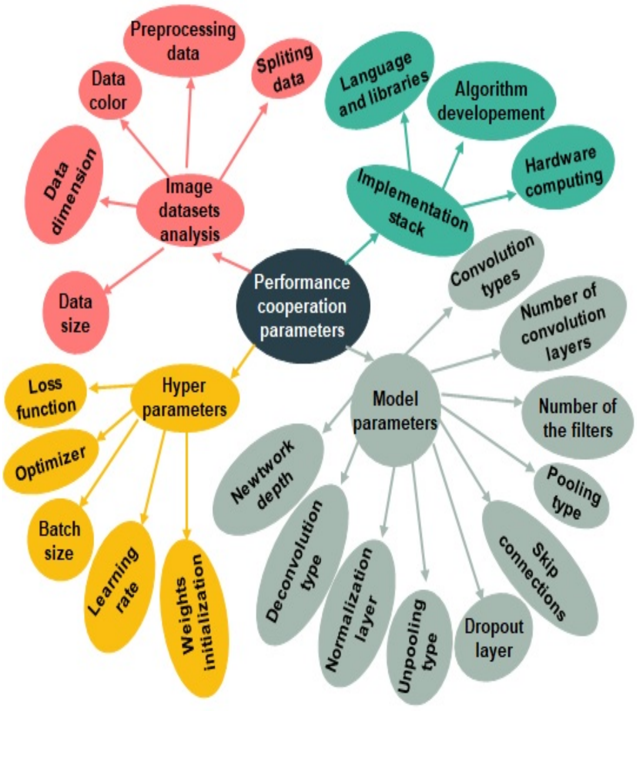

Figure 3. Performance cooperation parameters of U-Net

A. Image datasets attributes

Images are examples of training samples and can be represented via a 2D matrix or m-D matrix as collection of many features. Ronneberger et al. [4] employed 2D image ignoring the spatial data for the third dimension z-axis. Thus Cicek et al. [34] attempted to enhance the accuracy with increasing spatial information using extended dense 3D U-Net architecture that only needs sparse annotated volumetric images for learning but it consumes more time and computations cost. Hu et al. [54] suggested 2.5D U-Net which achieves tradeoff between the computations cost and the accuracy through converting 3D data to 2D patches taken at different depth, width and height directions and training each with separate U-Net model. It then fuses the result by calculating the average of the three prediction probability maps. Li et al. [36] linked 2D Dense UNet network for intra-slice features with 3D network for inter-slice features via a hybrid feature fusion (HFF) and the test time relies on the total number of slices. Li et al. [42] handled spatial loss by fusing intra-slice information and inter-slice probabilistic then it the outcome passes to a bidirectional recurrent network. Oyama et al. [5] introduced strong scalable hybrid-parallel algorithms to apply training pipeline on a large scale 3D U-Net based CNN, the train pipeline covers the computations and the I/O. Hybrid-parallel algorithm is based on adding spatial parallelism with the standard data parallelism where one-sample in the spatial domain is partitioned for achieving strong scaling which is superiors over the mini-batch dimension with greater aggregated memory capacity.

The Images color is another important factor which controls the number of channels. For instance both of black-white and gray images utilize one channel to produce one feature map, in contrast RGB uses three channels (red, green and blue color) when yielding one feature map. The color of the image determines the depth of memory (number of bits), black-white represents each pixel by one bit while gray images represent one pixel by 8-bit and RGB requires 24-bit, 8-bit for each of red, green and blue color. As memory depth becomes high as the training time increases. To enhance training accuracy, images may need some preprocessing using different operations such as resizing, cropping and pixel normalization [61]. Data augmentation is a main preprocessing for increasing the number of dataset using some geometric transformation. Ronneberger et al. [4] were the first to use robust and invariance data augmentation with elastic deformation. Niepceron et al. [16] used elastic transformation, flipping, Gaussian noise or shifting. Imai et al. [14] have been augment the size of the data with applying a random flip of the axis and permutation in different directions implemented in parallel to a void CPU bottleneck so it does not add extra overheads.Gadosey et al. [61] exploited augmentation strategies with height, width shift range, zoom range and horizontal flip. The data augmentation helps in preventing the overfitting problem, further realizes invariant model and generalization capability [54, 27]. Other attempts for enhancing data labels, Ke et al [50] overcame the problem of needing pixel-intensive annotations by proposing a multi-tasking U-net to perform segmentation with lazy labels. Dubost et al. [35] utilized individual global label for one image at the training and was able to determine small structures location. Data splitting ratio determines the ratio of training samples and may be splited randomly as in [8] or using other types. Despite of a significant number of training examples with labels is a key to success for the training of DNN models and getting better accuracy but it is constrained by memory size. Ronneberger et al. [4] employed overlap-tiling strategy for the large size of image. Imai et al. [14] extent the size of effective GPU memory by exploiting data-swap with CPU-GPU communication to trains 3D medical images instead of partitioning a big image into small patches that may influence segment accuracy when the target objects spans to several patches. Hou et al. [31] implemented spatial partitioning with performing halo exchange in tensorflow-TPU, to Mesh-Tensor Flow. The Halo exchange process means exchange data as patch margins between GPU/TPU devices prior to convolution operations.

B. Model parameters

Different model parameters are tackled individually in this article.

1) Model size

The selection of one approach is depending on the network structure. The continued growth in model size will make training large models difficult because it is hard to fit training with the limits of memory of one-GPU. The number of parameters is an essence factor to reflect the topology ability of designed model, thus Zhang et al. [38], compressed the parameters by adopting Incremental Quantization (INQ) as a regularization function faces potential overfitting as huge dense. Vaze et al. [62] developed thin U-Net having little feature channels at each layer, decreasing model size and producing fast inference. Mangalam and Salzamann [56] presented a modified distillation strategy, achieving a compressed Unet architecture with upto 1000x while maintaining an accuracy approaching the baseline U-net. Venkataramani et al. [53] suggested a way depending on finding a small group of filters and the merging parameters in order to derive each filter in every convolutional layer at the same training time, thus decreasing the required number of parameters that will be trained with and assist in avoid overfitting as small annotated data being available. Zhou et al. [37] used deep supervision to allow model pruning and enhancement. Thus, a stronger pruning leads to a further reduction in pridiction time but at the cost of significantly degraded accuracy. Qin et al. [51] increased the depth for the entire structure without rising of computational cost due to the pooling processes employed in the RSU-blocks. Isunuri and Kakarla [9] replaced the two convolution layers by one layer. Cl'erigues et al. [11] used four resolution levels.

2) Convolution operator

A convolution is the essence linear operator in the convolutional layer which extracts the important features of input data channels by repeated sliding the learnable filter of stride number over an input data then applying element-wise multiplication-accumulation outcome with corresponding window of input data and generate neurons that construct an output feature maps. Cicek et al. [34] extended 2D convolution to 3D convolution. To increase several receptive fields, Dolz et al. [10] exploited dilated convolution with different scales to tackle the variability in sizes. Diakogiannis et al. [24] utilized several parallel atrous convolutions expanding receptive field using different scale rate which assist in learning more context. Gu et al. [43] adapted atrous convolution to overcome the result loss from pool layer. Guo et al. [55] exploited deformable convolution for improving transformation-invariant feature learning. Niepceron et al. [16] utilized the depth wise separable convolution that rely on factorizing standard convolutions to a deep convolution then passes output to a standard convolution which has a kernel of 1 × 1 named point wise convolution. This type helped in compressing the model in term of the number of learnable parameters and training time. Beheshti and Johnsson [60] replaced the pooling by convolutions with stride 2 to perform down sampling, increasing the network expressiveness.

3) Activation function

Activation functions are nonlinear layer which apply mathematical operations on the linear output of the previous layer producing a nonlinear output for the next layer. These layers are typically used after each convolution layer or fully connected layer. There are various types of activation functions, for instance, Ronneberger et al. [4] used rectified linear unit abbreviated as ReLU. Cl'erigues et al. [11] employed parametric ReLU(PReLU). Oyama et al. [5] employed leaky ReLU. Peng et al. [27] adopted scaled exponential linear units termed SeLUs that permits stronger regularization schemes and makes learning very robust. Heinrich et al. [33] utilized the parameterized ternary hyperbolic tangent. AskariHemmat et al. [58] achieved a flexible balance between accuracy and memory aspects by using fixed point quantization for both weights and activations, instead of floating point, which reduces the amount of logic required for computational block implementation and leads to reduce the overall system power consumption. As a result the technique is appropriate for both of CPU and GPU hardware.

4) Pooling operator

To reduce the dimension size of the feature maps from the previous layer, pooling operator with some stride number is used. Pooling helps in realizing variance property and reducing computation complexity. Cicek et al. [34] utilized 3D max pool. Average pooling is used in reference number[36]. The global pooling layer is inserted prior the last layer at training phase [35]. Gao et al. [44] used gpool for graphic data. Yu et al. [45] employ ST-Pool for spatio-temporal graph structure.

5) Normalization layers

Batch normalization (BN) considered as the first suggested algorithm to solve the interval covariate shift. It employs the calculated mean and variance within a mini-batch of data to normalize its features within activation. BN is used for training very deep neural networks to standardize the outputs of the previous layer to zero mean and unit variance. This will make the learning process more stable with faster convergence in minimum number of training epochs [23]. In reference[34] batch normalization is inserted before every ReLU. Zhou and Yang [23] compared between four normalization ways BN,IN,LN, and GN. In the IN normalization both of the mean and variance are computed along channel and batch. In the LN normalization both of the mean and variance are computed along batch. In the GN normalization both of the mean and variance are computed along batch and partitioned channel. Inserting normalization will increase the runtime. Generally, BN is faster than IN, LN, and GN. GN with large group number and IN achieved better accuracy than other methods. To solve BN theoretical constraints, Chiley et al. [59] introduced online normalization which offers training without batches leading to reduced activation memory size. It processes with automatic differentiation by adding statistical normalization as a primitive. Online Normalization realizes the top Jaccard similarity coefficient compared to none, BN, LN normalization types. Gadosey[61] applied a weight standardization (WS) which standardizes gradients through backpropagation and applied on the input for the convolution layer counter to BN and GN that are applied either on the activations or the output layer. Further, the researchers illustrated through experiments a fusing of WS and GN realizes BN-comparable performance at large batch sizes. Niepceron et al. [16] employed GN help in training models on smaller batches, accelerating training and avoiding memory restrictions.

6) Dropout layer

Dropout indicates neglecting units such as neurons through the training stage of a particular group of neurons that is selected at random. It is a regularization mechanism which blocks neural network models from overfitting as in [4]. AskariHemmat et al. [58] explained that when this mechanism is used in conjunction with quantification, the accuracy drops much.

7) Deconvolution operator

Deconvolution and un pooling are revers operations of the convolution and they are necessary to restore the image space from the trained feature maps. Cicek et al.

[34] extended 2D deconvolution to 3D deconvolution, but it is computationally costly and demands acceleration. Liu et al. [30] designed hardware to accelerate deconvolution algorithms using FPGA compared with using CPU and GPU software implementations. The FPGA realized speedup of 4.14x to 9.48x over fully utilized 8-core CPU also with much less power consumption. On the other hand, FPGA consumes less power and better energy efficiency than a GPU but takes more times due to fewer numbers of DSPs (900). Hasan et al. [21] employed up- sampling operation relying on nearest-neighbor (NN) instead of transposed convolution. Niepceron et al.[16] used up sampling layer with bilinear interpolation as an alternative to transposed convolution.

8) Connections for Multi-scale transmission

Multi-scale features fusion and transmission are influential factor in segmentation accuracy, Ronneberger et al. [4] employed plain skip-connection to copy the context data from the encoder blocks to the decoder blocks at the same level. Zhou et al. [37] attempted an improvement by introducing UN++ which is nested and dense skip-connection and not just between the encoder and decoder, but still it is lacking enough information from the full scales. Then Peng et al. [27] enhanced it using multiple side-output fusion (MSOF) as deep supervision followed by a sigmoid function in order to create the final change map (CM). Huang et al. [52] proposed full scale skip-connections that could find the feature maps of one decoder layer by copying the information from both of the smaller and same inter-connection at the encoder unit as well as from intra-connection among the large scales at the decoder. This approach could captures fine-grained details as well as coarse-grained semantics from full scales achieving better accurate position-aware with few number of parameters. Oktay et al. [40] added attention gates (AGs) to the plain skip connection to help in detecting salient features, also enabled detecting target with different shapes and sizes. Bahl et al [26] proposed CUNet then extended to C-UNet++ which has less number of parameters, both does not contain skip-connection due to no introduction of more improvement in performance metrics when used as cloud segmentation . Li et al. [36], densely connected the layers within micro block to maximize the information flow. Zhang et al. [38], utilized multi-scales dense links for U-Net encoder unit, decoder unit and cross them to enhance the flow of information. Dolz et al. [10] divided the encoding path into N streams, every input for one image modality, then fuses with hyper-dense links through the same and between several paths. Guo et al. [55] employed stack dense U-Nets with scale aggregation network topology structure with a channel aggregation building. Li et al. [19], improved the accuracy and vessel boundary, developing a connection sensitive attention UNet (CSAU) by combining the connection sensitive loss with the attention gates as well as learns attention weights then concatenating it at the output of the network. Mehta and Valloli [46] proposed a reinforcement decoding branch to speed up the network convergence as well as assisting the network to predict density maps and with high SSIM index.

C. *Hyper parameters*

To deal with one application, five main parameters are necessary to be chosen which are: weight initializations, learning rate, batch sizes, optimization approaches and loss function.An appropirate initialization for the weights is the main aspect, else some portions of the neural network might results in excessive activations and the others never participate. Ideally, the initial weights must be adapted, so that every feature map within the designed network possess approximately unit variance. For instance, Gaussian distribution is used in [4] and glorot normal initialization is employed in [16,58]. The learning rate is the most significant hyper parameter which is controlled by how much variation in the model as a response for the evaluated error every time of updating model weights. Selecting the suitable learning rate is a big challenge, where a very small value may lead to elongate training process that may be disrupted, while a very large value may lead to learning a set of suboptimal weights too quickly or an unstable training process. Batch size is the number of training samples used per iteration. The large batch will improve the accuracy estimation of parameter gradients but at the expense of memory, while the reverse happens with small batch size [59]. For instance, Niepceron et al. [16] trained the model in tiny mini-batch with batches of size equal 2. Optimizers perform an indispensable function in reducing the loss incurred by the process of network training as well as in the neural network model during training. There are various algorithms for optimization such as stochastic gradient descent (SGD)[4][47], Adam[10], Adadelta[35], Nesterov-accelerated Adaptive Moment Estimation (Nadam)[55], Adafactor[31]. Loss layer is considered the final fully connected layer which describes the model's prediction for a single example by calculating the loss or error between desired and actual outcomes. A weighted cross-entropy loss function is utilized in reference[36]. In reference[35], a mean sequare loss is applied to optimize the weights. Diakogiannis et al. [24] presented a new dice loss termed Tanimoto which achieved better convergence, and performs well even in very unbalanced classes. Niepceron et al. [16] merged two losses, binary cross-entropy that abbreviation as (BCE) and dice. Imai et al. [14] used dice loss. Guo et al. [55] designed a coherent loss to outside transformed data.

D. *Implementation stack*

The developers have introduced several stacks of an efficient frameworks and hardware computing to facilitate and optimize performance of the U-Net learning algorithm. Each stack has a special structure and gives a different quality features for one application problem. For instance, Imai et al. [14] used a stack implementation of TensorFlow1.8 framework with a feature TensorFlow Large Model Support (TFLMS) dictated to perform data swapping as a solution of GPU limitation memory while offering parameters to reduce the communication overhead. The TensorFlow1.8 support Keras APIs running on RHEL 7.3 as an operating system (OS), then both of CUDA9.1, and cuDNN7.0.2 are used as driver and library. The implementation was done on IBM® Power Systems™ S822LC that includes two POWER 8-CPU (ten cores operates at 3.54 GHz) and a CPU memory with 512 GB. The system includes four NVIDIA® Tesla® P100s, each one with GPU memory of 16-GB. Both of CPU and GPU are integrated using NVLink 1.0 with a bidirectional bandwidth of 80 GB/sec.

The most frameworks used with U-Net and its variants are Caffe[4][47], Theano[35], PyTorch[10] and MXNet[55]. Hou et al. [31] have participated in a new Mesh-TensorFlow based framework that is able to handle images of various volumes with a model parallelism in different mappings. In general, frameworks supports various APIs of several languages such as Python[35], C++[30], Verilog [30] and others. Pati et al. [29] compared three inference accelerators as OpenVINO (CPU), TensorRT (GPU), and WinML (CPU, GPU), the results show that WinML should be employed. Variant compilers operate with frameworks such as cu/DNN. Ojika et al. [15] achieved a 3.4x speedup using TensorFlow with deep neural network library (DNNL) compared to stock TensorFlow without DNNL at similar training batch size. The frameworks are executed over one of the operating systems like Linux, Ubuntu 14.04 [54] and windows [29]. Pati et al. [29] achieved faster inference speed over Linux compared to window. However, the configuration of the frameworks dictates how the workload is distributed on the available hardware processing units via driver and interconnection technologies like AXILite Bus [30]. On the other hand, the choice of a suitable processor is one way to evaluate the influence of hardware on performance baseline U-Net models at training and inference phases. Main hardware platform types cover CPUs [9], GPUs [5,9,29], TPUs [31,32], FPGAs [25,30], servers [15] and heterogeneous [63]. Implementation of the U-Net concepts on the general purpose processors, as CPU is not suitable due to performance bottlenecks which lack a parallel property. To solve memory bottleneck without employing down scaling or tiling/patch tricks when fitting the images into memory, Ojika et al. [15] assisted data scientists in training approximately 1TB full scale healthcare images on 3D U-Net and implemented over CPU-based server using single node Dell EMC PowerEdge server that has a 4-socket 2nd Generation Intel Xeon Scalable processor and supporting a large system memory of 1.5 TB. GPU is more efficient to accelerate the implementation than CPU, Isunuri and Kakarla [9] employed U-Net for performance analysis. The resulted training time is 467sec and 33sec for CPU and GPU, respectively. Pati et al. [29] analyzed the inference time over NVIDIA K80 GPU, NVIDIA V100 GPU, and Intel Xeon CPU, then recommend employing NVIDIA V100 as faster computing hardware. Oyama et al. [5] implemented strong scalable hybrid-parallel algorithms based training pipeline on a GPU supercomputer, a speed up of 1.42 is obtained for the 3D U-Net model over a 512 of GPUs which is faster than 256 GPUs. TPUs are modern processors that employ a systolic array into multiplication. Google TPUs consider efficient resources to accelerate the learning time for cloud-based service; this may lead in pruning the network to be lighter with little impacts in prediction efficiency. Also, TPUs helps in testing wider and deeper architectures to solve the memory limitation of numerous existing one GPU systems [32]. Civit et al. [32] applied generalized U-Net segmentation on images from different acquisition sources as a cloud-based service. To reduce the execution time, the training and prediction are done with employing cooperative iPython notebook environment Google Colaboratory, that support Keras framework and implemented on cloud architectures. The Google tensor processing units (TPU) processors and the GPU used for a small group of trials. The TPUs are faster than cloud GPU solutions by a range of 2 to 3 times. Hou et al. [31] trained models over a cluster of TPUs. Each one contains 2 cores. On the other hand, the reconfigurability and customization features of FPGA emerge to be a good platform to power-efficient and higher performance for neural networks based CNN. Liu et al. [30], designed a CNN accelerator applied on the real time image segmentation. The accelerator was built on U-Net and the tailored architecture is implemented over FPGA with a sharing input buffer and non-linear optimization to construct space exploration. The results show, that the FPGA is superior on a fully exploiting 8-threads CPU by 10x speedup and 110x energy efficiency, respectively. On the other hand when compared to GPU, FPGA is slower but gives an enhancement of 8x energy efficiency. Liu and Luk [25] proposed a uniform design to run convolution and deconvolution operations with random kernel size using one vector multiplication module. The implementation was done with Intel's Arria 10 SOC FPGAs that includes an A10-SX 660 device (20nm), dual core ARM based on CPU of 1.5 GHz and DDR4 memory of 2GB. The new directions are using heterogeneous computing (HGC) which combines the capability of more than one type of sophisticated computing processors. Joardar et al. [63] accelerated DNN segmentation based U-Net through the proposed heterogeneous architecture called GRAMARCH that merges the advantages of Resistive Random-Access Memory (ReRAM) with in-memory computation while GPUs have inherent parallelism simultaneously by employing a high-throughput 3D Network-on-Chip(NoC).

4. U-NET IMPROVEMENTS

The main improvements produced from the reviewed studies are summarized in the table I as a fast referencing:

TABLE I. PERFORMANCE CHALLENGES WITH THE MOST IMPROVEMENTS

Challenge	Improvements
Training dataset size	Data augmentation:[4],[6],[12],[13],[14],[16],[19],[21],[22],[23],[24],[26],[27],[28],[31],[32],[34],[35],[36],[40], [41],[42],[43],[46],[47],[49],[50],[51],[54],[55],[61]
Accuracy	Dilated convolution: [6],[8] Attention technique:[8],[19],[22],[39],[40] Residual technique: [8],[11],[12],[17],[18],[22],[24],[27],[39],[43] Dense technique: [10],[27],[36],[37],[38],[51],[52],[55] Recurrent technique: [18] Cascade:[12],[17],[55] Deformable convolution:[20],[55] Multipath fusion:[6],[10] high-resolution 3D U-Net:[14],[31] Depthwise separable convolution:[26]
FLOPs	Compressed model: C-UNet++ and C-UNet[26], SD-UNet [61]
Overfitting	Data augmentation: [4],[14],[16],[21],[22],[24],[27],[31],[32],[36],[42],[43],[47],[50],[54],[61] Dropout:[4],[17],[32],[44],[58] BN:[16],[32] Regularization: Sparsity[33], Incremental quantization (INQ)[38], loss(Lreg or weight decay)[43], L2[44] Early stopping:[7],[8],[14],[16] Others: Dense connection[10], Small patch[20], Stack multiple similar structure[51], Fliter sharing[53]
Memory size limitation	Tiling[4],Hybrid parallelism[5], Partially reversible U-Net[13], Data exchange[14], Big memory[15], Memory sharing[30], Spatial partitioning[31], Parameters sharing[57], Small batch and online normalization[59] Compressed model:[16],[22],[26],[33],[58],[60],[62]
Time	Model compression:[16],[58] GPU:[4],[16],[29], TPU [32], Heterogeneous[63] Compressed model: TernaryNet[33], Squeeze U-Net[60], SD-UNet [61], knowledge distillation [62]
Latency	Hardware accelerators: [25],[29]
Throughput	Hardware accelerators: [25],[5]
Power	Model compression:[16],[26],[58], Hardware accelerators [30]
Energy	Hardware accelerators[30], Model compression[60]
Model compression	Decrease number of convolution layer[9], Shared-weights residual unit[17], Ternary quantisation[33], Incremental quantization [38], UNET3+[52],Filter sharing [53], Knowledge distillation[57], U-Net Fixed-Point Quantization[58], Squeeze U-Net [60], Depth wise separable convolution:[26],[61],[62]
Object appearance	MFP-Unet [6], Dense Multi-path U-Net[10], Deformable convolution blocks [20], U-NetPlus [21], Attention gate (AG)[40],UNET3+ [52]
Class imbalance	Hybrid sampling [11], Hybrid loss [22], A variant of the Dice loss: [24],[27]
Generalization	Unet-GAN[7],DUNet[20],U-Net with GN or IN[23], Generalized U-Net[32],H-DenseUNet[36], CE-Net[43], ScleraSegNet[47], AHCNet[48]

5. CONCLUSIONS

The studies and works that have been made on the performance analysis of the U-Net architecture designs show a significant performance advances over the traditional approaches. The developments in U-Nets can be classified regarding to different approaches, especially, the design feature of the structural units, and the fast implementation. Therefore, this review highlights the efforts of researchers through the main details of U-Net and its application to different tasks. It is demonstrated also the challenges and recent directions in the field of technology implementation. Out of the results from prior paper works, it is concluded that the performance of U-Net networks depends on the amount of data for the application and the implementation of both the training and inference. The good management of a variety of hardware's and improved frameworks can accelerate performance and achieves better results. Some points that can be helpful for future works are: data augmentation for increasing training examples. Attention technique for silent detection features. Residual technique allow deepest network and tackle gradient vanishing. Dense technique is used to enhance information flow and deep supervision. The suitable performance in term of time, memory space and energy will be either using software based on data swapping with heterogeneous cooperation CPU-GPU or using hardware computing with FPGA which is faster than CPU but slower than GPU. FPGA is more energy efficient than both CPU and GPU. On the other hand, TPU is for cloud based service. However, despite of U-Net and its variants have shown significant results in analyzing many tasks, there are some open challenges. First there is a need for a design that matches with the power and heat of the platform battery trying to use advanced software optimization techniques to accelerate training at an affordable cost and save battery power for the platform. Second, attempt to delete some layers to achieve a simple design while maintaining accuracy. Third, try to design an adaptive model which can be able to work at real-time. Finally, there is a necessity for designing a hybrid optimizer.

REFERENCES

[1] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” Proceedings of the IEEE conference on computer vision and pattern recognition. Boston, MA, USA, pp. 3431–3440, 2015.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 12, pp. 2481–2495, 2017.

[3] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking Atrous Convolution for Semantic Image Segmentation,” arXiv preprint arxiv:1706.05587, 2017.

[4] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” Proceedings of the International Conference on Medical image computing and computer-assisted intervention, Springer, Cham, pp. 234–241, 2015.

[5] Y. Oyama, N. Maruyama, N. Dryden, E. McCarthy, P. Harrington, J. Balewski, S. Matsuoka, P. Nugent, and B. Van Essen, "The Case for Strong Scaling in Deep Learning: Training Large 3D CNNs with Hybrid Parallelism," arXiv preprint arxiv:2007.12856, 2020.

[6] S. Moradi, M. G. Oghli, A. Alizadehasl, I. Shiri, N. Oveisi, M. Oveisi, M. Maleki, and J. Dhooge, "A Novel Deep Learning Based Approach for Left Ventricle Segmentation in Echocardiography: MFP-Unet," arXiv preprint arXiv: 1906.10486, 2019.

[7] W. Yan, Y. Wang, S. Gu, L. Huang, F. Yan, L. Xia, g and Q. Tao, "The Domain Shift Problem of Medical Image Segmentation and Vendor-Adaptation by Unet-GAN," International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Cham, pp. 623–631, 2019.

[8] T. Zhou, S. Canu, and S. Ruan, "An automatic COVID-19 CT segmentation based on U-Net with attention mechanism," arXiv preprint arXiv:2004.06673, 2020.

[9] B. V. Isunuri, and J. Kakarla, "Fast brain tumor segmentation using optimized U-Net and adaptive thresholding," Automatika, vol. 61, no. 3, pp. 352-360, 2020.

[10] J. Dolz, I. Ben Ayed, and C. Desrosiers, "Dense multi-path U-Net for ischemic stroke lesion segmentation in multiple image modalities," International MICCAI Brainlesion Workshop, pp. 271–282, 2018.

[11] A. Clèrigues, S. Valverde, J. Bernal, J. Freixenet, A. Oliver, and X. Lladó, "SUNet: a deep learning architecture for acute stroke lesion segmentation and outcome prediction in multimodal MRI," arXiv preprint arXiv:1810.13304, 2018.

[12] D. Lachinov, E. Vasiliev, and V. Turlapov, "Glioma segmentation with cascaded UNet," International MICCAI Brainlesion Workshop, pp. 189–198, 2018.

[13] R. Brügger, C. F. Baumgartner, and E. Konukoglu, "A Partially Reversible U-Net for Memory-Efficient Volumetric Image Segmentation," International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Cham, pp. 429–437, 2019.

[14] H. Imai, S. Matzek, T. D. Le, Y. Negishi, and K. Kawachiya, "Fast and Accurate 3D Medical Image Segmentation with Data-swapping Method," arXiv preprint arxiv:1812.07816, 2018.

[15] D. Ojika, B. Patel, G. A. Reina, T. Boyer, C. Martin, and P. Shah, "Addressing the Memory Bottleneck in AI Model Training," arXiv preprint arXiv:2003.08732, 2020.

[16] B. Niepceron, A. Nait-Sidi-Moh, and F. Grassia, "Moving Medical Image Analysis to GPU Embedded Systems: Application to Brain Tumor Segmentation," Applied Artificial Intelligence, vol. 34, no. 12, pp. 866–879, 2020.

[17] J. Zhuang, "LADDERNET: Multi-path networks based on U-Net for medical image segmentation," arXiv preprint arxiv:1810.07810, 2018.

[18] M. Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari, "Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation," arXiv preprint arxiv:1802.06955, 2018.

[19] R. Li, M. Li, J. Li, and Y. Zhou, "Connection Sensitive Attention U-NET for Accurate Retinal Vessel Segmentation," arXiv preprint arxiv:1903.05558, 2019.

[20] Q. Jin, Z. Meng, T. D. Pham, Q. Chen, L. Wei, and R. Su, "DUNet: A deformable network for retinal vessel segmentation," Knowledge-Based Systems, vol. 178, pp. 149–162, 2019.

[21] S. M. K. Hasan, C. A. Linte, and S. Member, "U-NetPlus : A Modified Encoder-Decoder U-Net Architecture for Semantic and Instance Segmentation of Surgical Instrument," arXiv preprint arXiv:1902.08994, 2019.

[22] Z.-L. Ni, G.-B. Bian, X.-H. Zhou, Z.-G. Hou, X.-L. Xie, C. Wang, Y.-J. Zhou, R.-Q. Li, and Z. Li, "RAUNet: Residual Attention U- Net for Semantic Segmentation of Cataract Surgical Instruments," International Conference on Neural Information Processing. Springer, Cham, pp. 139–149, 2019.

[23] X. Y. Zhou and G. Z. Yang, "Normalization in training U-Net for 2-D biomedical semantic segmentation," IEEE Robot. Automation Letters, vol. 4, no. 2, pp. 1792–1799, 2019.

[24] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, "ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 162, pp. 94–114, 2020.

[25] S. Liu and W. Luk, "Towards an efficient accelerator for DNN-based remote sensing image segmentation on FPGAs," In: 29th International Conference on Field Programmable Logic and Applications (FPL). IEEE, pp. 187–193, 2019.

[26] G. Bahl, L. Daniel, M. Moretti, and F. Lafarge, "Low-power neural networks for semantic segmentation of satellite images," Proceedings of the IEEE International Conference on Computer Vision Workshops(ICCVW), Seoul, Korea (South),pp. 2469-2476, 2019.

[27] D. Peng, Y. Zhang, and H. Guan, "End-to-end change detection for high resolution satellite images using improved UNet++," Remote Sensing, vol. 11, no. 11, 2019.

[28] L. A. Tran and M. H. Le, "Robust u-net-based road lane markings detection for autonomous driving," International Conference on System Science and Engineering (ICSSE). IEEE, Dong Hoi, Vietnam, pp. 62–66, 2019.

[29] D. Pati, C. Favart, P. Bahl, V. Soni, Y.-C. Tsai, M. Potter, J. Guan, X. Dong, and V. R. Saripalli, "Impact of Inference Accelerators on hardware selection," arXiv preprint arXiv:1910.03060, 2019.

[30] S. Liu, H. Fan, X. Niu, H. C. Ng, Y. Chu, and W. Luk, "Optimizing CNN-based Segmentation with Deeply customized convolutional and deconvolutional architectures on FPGA," ACM Transactions on Reconfigurable Technology and Systems (TRETS), vol. 11, no. 3, pp. 1-22, 2018.

[31] L. Hou, Y. Cheng, N. Shazeer, N. Parmar, Y. Li, P. Korfiatis, T. M. Drucker, D. J. Blezek, and X. Song, "High Resolution Medical Image Analysis with Spatial Partitioning," arXiv preprint arXiv:1909.03108, 2019.

[32] J. Civit-Masot, F. Luna-Perejon, S. Vicente-Diaz, J. M. Rodriguez Corral, and A. Civit, "TPU cloud-based generalized U-Net for eye fundus image segmentation," IEEE Access, vol. 7, pp. 142379–142387, 2019.

[33] M. P. Heinrich, M. Blendowski, and O. Oktay, "TernaryNet: faster deep model inference without GPUs for medical 3D segmentation using sparse and binary convolutions," International journal of computer assisted radiology and surgery, vol. 13, no.

9, pp. 1311–1320, 2018.

[34] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-net: Learning dense volumetric segmentation from sparse annotation,” International conference on medical image computing and computer-assisted intervention. Springer, Cham, pp. 424–432, 2016.

[35] F. Dubost, G. Bortsova, H. Adams, A. Ikram, W. J. Niessen, M. Vernooij, M. De Bruijne, “Gp-Unet: Lesion detection from weak labels with a 3D regression network,” International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Cham, pp. 214-221, 2017.

[36] X. Li, H. Chen, X. Qi, Q. Dou, C. W. Fu, and P. A. Heng, “H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation from CT Volumes,” IEEE transactions on medical imaging, vol. 37, no. 12, pp. 2663–2674, 2018.

[37] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, “Unet++: A nested u-net architecture for medical image segmentation,” Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer, Cham, pp. 3–11, 2018.

[38] J. Zhang, Y. Jin, J. Xu, X. Xu, and Y. Zhang, “MDU-Net: Multi-scale Densely Connected U-Net for biomedical image segmentation,” arXiv preprint arXiv: 1812.00352, 2018.

[39] Q. Jin, Z. Meng, C. Sun, L. Wei, and R. Su, “RA-UNet: A hybrid deep attention-aware network to extract liver and tumor in CT scans,” arXiv preprint arXiv:1811.01328, 2018.

[40] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, and others, “Attention U-Net: Learning Where to Look for the Pancreas,” arXiv preprint arXiv:1804.03999, 2018.

[41] S. Kohl, B. Romera-Paredes, C. Meyer, J. De Fauw, J. R. Ledsam, K. Maier-Hein, S. A. Eslami, D. J. Rezende, and O. Ronneberger, “A probabilistic U-net for segmentation of ambiguous images,” Advances in Neural Information Processing Systems, pp. 6965–6975, 2018.

[42] J. Li, X. Lin, H. Che, H. Li, and X. Qian, “Probability Map Guided Bi-directional Recurrent UNet for Pancreas Segmentation,” arXiv preprint arXiv:1903.00923, 2019.

[43] Z. Gu, J. Cheng, H. Fu, K. Zhou, H. Hao, Y. Zhao, T. Zhang, S. Gao, and J. Liu, “CE-Net: Context Encoder Network for 2D Medical Image Segmentation,” IEEE transactions on medical imaging, vol. 38, no. 10, pp. 2281–2292, 2019.

[44] H. Gao and S. Ji, “Graph u-nets,” arXiv preprint arXiv:1905.05178, 2019.

[45] B. Yu, H. Yin, and Z. Zhu, “ST-UNet: A Spatio-Temporal U-Network for Graph-structured Time Series Modeling,” arXiv preprint arXiv:1903.05631, 2019.

[46] K. Mehta and V. K. Valloli, “W-Net: Reinforced U-Net for Density Map Estimation,” arXiv preprint arXiv: 1903.11249, 2019.

[47] C. Wang, Y. He, Y. Liu, Z. He, R. He, and Z. Sun, “ScleraSegNet: An Improved U-Net Model with Attention for Accurate Sclera Segmentation,” International Conference on Biometrics (ICB). IEEE, Crete, Greece, pp. 1-8, 2019.

[48] H. Jiang, T. Shi, Z. Bai, and L. Huang, “AHCNet: An Application of Attention Mechanism and Hybrid Connection for Liver Tumor Segmentation in CT Volumes,” IEEE Access, vol. 7, pp. 24898–24909, 2019.

[49] S. A. A. Kohl, B. Romera-Paredes, K. H. Maier-Hein, D. J. Rezende, S. Eslami, P. Kohli, A. Zisserman, and O. Ronneberger, “A Hierarchical Probabilistic U-Net for Modeling Multi-Scale Ambiguities,” arXiv preprint arXiv: 1905.13077, 2019.

[50] R. Ke, A. Bugeau, N. Papadakis, P. Schuetz, and C.-B. Schönlieb, “A multi-task U-net for segmentation with lazy labels,” arXiv preprint arXiv: 1906.12177, 2019.

[51] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, “U2-Net: Going deeper with nested U-structure for salient object detection,” Pattern Recognition, vol. 106, 2020.

[52] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu, “UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation,” IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, Barcelona, Spain, pp. 1055-1059, 2020.

[53] R. Venkataramani, S. Thiruvenkadam, P. Sudhakar, H. Ravishankar, and V. Vaidya, “Filter sharing: Efficient learning of parameters for volumetric convolutions,” arXiv preprint arXiv: 1612.02575, 2016.

[54] K. Hu, C. Liu, X. Yu, J. Zhang, Y. He, and H. Zhu, “A 2.5 D Cancer Segmentation for MRI Images Based on U-Net,” In: 2018 5th International Conference on Information Science and Control Engineering (ICISCE). IEEE, pp. 6–10, 2018.

[55] J. Guo, J. Deng, N. Xue, and S. Zafeiriou, “Stacked dense u-nets with dual transformers for robust face alignment,” arXiv preprint arXiv:1812.01936, 2018.

[56] K. Mangalam and M. Salzamann, “On Compressing U-net Using Knowledge Distillation,” arXiv preprint arXiv:1812.00249, 2018.

[57] W. Wang, K. Yu, J. Hugonot, P. Fua, and M. Salzmann, “Recurrent U-Net for resource-constrained segmentation,” Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea (South), pp. 2142–2151, 2019.

[58] M. AskariHemmat, S. Honari, L. Rouhier, C. S. Perone, J. Cohen-Adad, Y. Savaria, and J.-P. David, “U-Net Fixed-Point Quantization for Medical Image Segmentation,” Large-Scale Annotation of Biomedical Data and Expert Label Synthesis and Hardware Aware Learning for Medical Imaging and Computer Assisted Intervention. Springer, Cham, pp. 115–124, 2019.

[59] V. Chiley, I. Sharapov, A. Kosson, U. Koster, R. Reece, S. S. de la Fuente, V. Subbiah, and M. James, “Online normalization for training neural networks,” Advances in Neural Information Processing Systems, pp. 8433-8443, 2019.

[60] N. Beheshti and L. Johnsson, “Squeeze U-net: A memory and energy efficient image segmentation network,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops(CVPRW), Seattle, WA, USA, pp. 1495-1504, 2020.

[61] P. K. Gadosey, Y. Li, E. A. Agyekum, T. Zhang, Z. Liu, P. T. Yamak, and F. Essaf, “SD-UNET: Stripping down U-net for segmentation of biomedical images on platforms with low computational budgets,” Diagnostics, vol. 10, no. 2, 2020.

[62] S. Vaze, W. Xie, and A. I. L. Namburete, “Low-Memory CNNs Enabling Real-Time Ultrasound Segmentation towards Mobile Deployment,” IEEE Journal of Biomedical and Health

Informatics, vol. 24, no. 4, pp. 1059–1069, 2020.

[63] B. K. Joardar, N. K. Jayakodi, J. R. Doppa, H. Li, P. P. Pande, and K. Chakrabarty, "GRAMARCH: A GPU-ReRAM based Heterogeneous Architecture for Neural Image Segmentation," Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, Grenoble, France, pp. 228–233, 2020.

Ula T. Salim received the BSc and MSc degree in Computer Engineering in 2007 and 2013. She is work as assistant lecturer with Computer Engineering Department, Mosul University. Currently, she is PhD student at research stage. Her research interests include image processing, deep learning and parallel processing.

Fakhrulddin H. Ali is assistant professor at the computer engineering Department-University of Mosul. He received B.Sc in Electronic and Communication Engineering-Department of Electrical Engineering-University of Mosul. He received P.G. Diploma and M.Sc from the same Department at 1977-1979. He graduated from university of Bradford-U.K. with a PhD degree at 1989. He has more than 30 scientific papers in journals and conferences. He supervised more than 25 postgraduate M.Sc and PhD. Thesises and dissertations. His field of interest is 3D computer graphics and real time systems.

Shefa A. Dawwd is a professor of computer engineering at the Computer Engineering Department-University of Mosul. He received the B.Sc in Communication Engineering, the M.Sc and the Ph.D in Computer Engineering. He has authored about 40 international journal, conference papers and one chapter book. His research focus is on the processing acceleration of 1D, 2D and 3D signals, real time applications, deep learning, Convolutional Neural Networks, and hetrogeneous computing. He is a regular reviewer of IEE, Elsevier and other Scopus journals.