

Building detection based on searching of the optimal kernel shapes pruning method on Res2-Unet

Arulappan Amala Arul Reji¹, Sathiyamoorthy Muruganantham²

¹S. T. Hindu College, Nagercoil, Affiliated to Manonmaniam Sundaranar University, Tirunelveli, India

²Department of computer Application, S. T. Hindu College, Nagercoil, Affiliated to Manonmaniam Sundaranar University, Tirunelveli, India

Article Info

Article history:

Received Jan 13, 2024

Revised Apr 4, 2024

Accepted Apr 30, 2024

Keywords:

CNN

Data quantification

Deep learning networks

Res2-Unet

SOKS

ABSTRACT

In recent years, advances in remote sensing technology have made it feasible to use satellite data for large-scale building detection. Moreover, the building detection from multispectral satellite photography data is necessary; however, it is difficult to recover the accurate building footprint from the high-resolution pictures. Because the deep learning networks contains high computational cost and over-parameterized. Therefore, network pruning has been used to reduce the storage and computations of convolutional neural network (CNN) models. In this article, we proposed the pruning technique to prune the CNN network from Res2-Unet model for accurately detecting the buildings. Initially, the CNN network is pruned by utilizing the searching of the optimal kernel shapes technique. It is employed to carry out stripe-wise pruning and automatically find the ideal kernel shapes. Then the data quantification is applied to enhance the proposed model and also reduce the complexity. Finally, the enhanced Res2-Unet model is used for the building detection. Moreover, WHU East Asia Satellite and the Massachusetts building dataset are the two available datasets used to access the suggested framework. Compare to the existing models, the proposed model gives better performance.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Arulappan Amala Arul Reji

S. T. Hindu College, Nagercoil, Affiliated to Manonmaniam Sundaranar University

Tirunelveli, Tamil Nadu, India

Email: amalasweet.reji05@gmail.com

1. INTRODUCTION

Buildings have a crucial role in the formation of cities and are necessary for urban mapping. Moreover, object detection is a crucial component of computer vision tasks like instance segmentation, image captioning, and object tracking [1], with applications in building detection, plant and crops disease detection [2], [3], number-plate recognition and pedestrian detection. In recent years, the field of obtaining exact building objects from the remote sensing data has gained significant interest. Additionally, environmental sciences, urban planning, 3D city modelling, risk and damage assessment of natural hazards and other geospatial applications all depend on the building information. Many data sources, including satellite images, radar images, laser scanning data and aerial photos that are used to define the building objects [4]. Then the building shapes, sizes, colours, and backdrops are diverse and complicated, making building detection difficult. Furthermore, in many urban applications, such as infrastructure development, state cadastral inspection, city planning, the municipal services providing, the detection from satellite imagery becomes the real challenge [5]. Therefore, the deep learning techniques are employed by various researchers for detecting the buildings from satellite images [6], [7].

There are several methods for detecting buildings, including segmentation-based, polygon-based, active contour-based, footprint regression-based and topology-aware loss-based. These methods identify buildings from images using various techniques like fuzzy logic, machine learning, deep learning, and optimisation. Moreover, the rapid growth of artificial intelligence technology has the potential to enable building detection using high-resolution satellite images. In order to segment remote sensing images for building detection, [8] attempts to apply the U-Net deep neural network (DNN) model with ResNet encoder. Aghayari [9] assess UNet and Inception ResNet UNet are the two deep network architectures utilized for automated building detection from aerial photography. This is due to the fact that, in contrast to UNet, the Inception ResNet UNet model has a greater range of parameters and is deeper [10].

Saidi *et al.* [11] pruning involves removing certain weights or neurons from a neural network, resulting in a smaller model size. The authors pruned convolutional neural network (CNN) models (VGG-16 and ResNet-50) to decrease computational complexity and memory requirements, making them more suitable for resource-constrained platforms. Furthermore, the pruning reduces the computational cost of inference by eliminating unnecessary connections. Gong *et al.* [12] propose reparameterized fusion convolution (RFConv) as a CNN building block to address challenges in object detection from unmanned aerial vehicles (UAV), including information losses and high computational costs. This RFConv utilizes the multiscale convolution branches to increase the receptive field and capture tiny object information. Moreover, through pruning, reparameterization lowers the computational burden during inference. Recent research suggests large convolutional kernel design enhances CNN performance in vision transformers, but it may cause incompatible operators with different hardware platforms, making it unsuitable for large kernel sizes. Therefore, Li *et al.* [13] demonstrate that the closure effects of high kernel sizes can be achieved using small convolutional kernels and convolution procedures, and also propose a shift-wise operator that ensures CNNs capture long-range dependencies using a sparse mechanism while remaining hardware-friendly. As neural networks become deeper and wider, their performance improves with more layers and neurons. However, this also increases computational and memory costs. Therefore, Hu *et al.* [14] suggested the network pruning technique, which optimizes the network repeatedly by removing unnecessary neurons. Then the large networks often have a significant portion of neurons with zero outputs, which can be safely removed without affecting the network's overall accuracy.

Pruning reduces the number of weight connections in a network to achieve various objectives, such as a smaller model footprint and faster inference. Additionally, pruning is an efficient way to condense models while maintaining the accuracy. The pruning techniques are, applied during or after the training process, to regularize models and reduce the overfitting risk. By eliminating unnecessary connections and parameters from a model, a technique known as post-training pruning can increase a model's performance and execution speed [15]. Furthermore, three primary approaches are mostly applied in the existing research such as weight pruning, neuron pruning, and structured pruning. The weight pruning approach is utilized to remove the connections or weights from a model [16], then the neural pruning eliminates individual neurons from a model according to a predetermined significance level [17], and the structured pruning operates at a higher granularity level, targeting entire channels or groups of neurons. It removes entire structures depending on a predetermined significance threshold, rather than individual weights or neurons [16]. Among the three methods, structured pruning stands out as the best choice for achieving efficient neural networks. It strikes a balance by reducing both parameters and computational resources while maintaining model performance. Therefore, this paper, proposes the structure pruning technique SOKS.

In previous works, many deep learning approaches such as Unet, Segnet, Res2-Unet, and Resnet50 are proposed to detect the buildings from various image sources. Moreover, these deep learning approaches are given their best performance for building detection. However, the deep learning methods has some issues like, it takes more time for training and testing process, it needs large memory footprint and also it is very expensive [18]. Mostly the deep learning techniques are based on the CNNs, so to minimize the complexities of deep learning by employing the pruning technique in CNN layer [19], [20]. This study investigated the effects of the SOKS pruning method on building detection from satellite images. While earlier studies have explored the impact of various techniques, they have not explicitly addressed the influence of the SOKS pruning method on this specific task. Therefore, the SOKS pruning method is proposed in this article to enhance the deep learning approach for the building detection from satellite images. First the pruning technique SOKS is applied in CNN layer from Res2-Unet model, it is used to minimize the number of parameters and also it reduces the computational cost. Then secondly, the network is applied in the quantitative process. The quantification data is one of the speed and efficiency methods; therefore, it is used to save the networks time and also analysing the huge dataset. Finally, the enhanced Res2-Unet model trained by the datasets and it shows the best detection result.

The main contributions of this work are detailed as:

- This paper presents structure pruning technique SOKS, it is utilized to reduce the complexity of the Res2-Unet model. Then the SOKS refers to the system that is developed to automatically find the best kernel shapes and perform stripe-wise pruning.
- In addition to offering suitable receptive fields for every convolution layer, the ideal kernel shapes also eliminate unnecessary parameters from convolution kernels. By using these irregular kernels, SWP is also accomplished, and real GPU inference speedups are realized.
- In order to further reduce the neural network's computing load, the reorganized neural network is further trained by including quantitative processing and fine-tuning the network. Finally, the enhanced Res2-Unet model is used to accurately detect the buildings from the satellite images.
- Finally in the experimental results, we found that the SOKS pruning technique correlates with reducing the parameters and enhancing model accuracy when compared to existing techniques such as PFEC, FPGM, GAL, Hinge, and HRank. Moreover, the suggested enhanced Res2-Unet exhibits better outcomes for detecting buildings compared to previous networks, including CT-Unet and Unet-Resnet50 and Res2-Unet.

The remaining part of this paper is explained in the below section. Section 2 explains the related work. The background of this article is detailed in section 3. Then section 4 discusses about the proposed methodology. The experimental result section is explained in section 5. Section 6 details the conclusion of this article.

2. METHOD

Figure 1 shows the proposed Res2-Unet framework. The $3 \times 512 \times 512$ pixels image is fed as input into the framework. The different scales features of images are extracted through encoding and decoding stages. Res2-Unet is designed for building detection from high spatial resolution images, which tackles issues with complicated object properties and visual similarity. However, the Res2-Unet cannot accurately detect the buildings because the network contains more layers so it causes the over fitting and high computational cost. Especially in CNN layer has so many parameters, model deployment can occasionally be expensive. The proposed Res2-Unet model is pruned by using the SOKS pruning technique for building detection. First, the SOKS pruning technique is applied in CNN layer to reduce the over-parameterized and computational cost. Then the quantification of data process is applied to fine-tune the network. The model is enhanced by completing these processes. Then the enhanced Res2-Unet model is accurately detecting the buildings from satellite images.

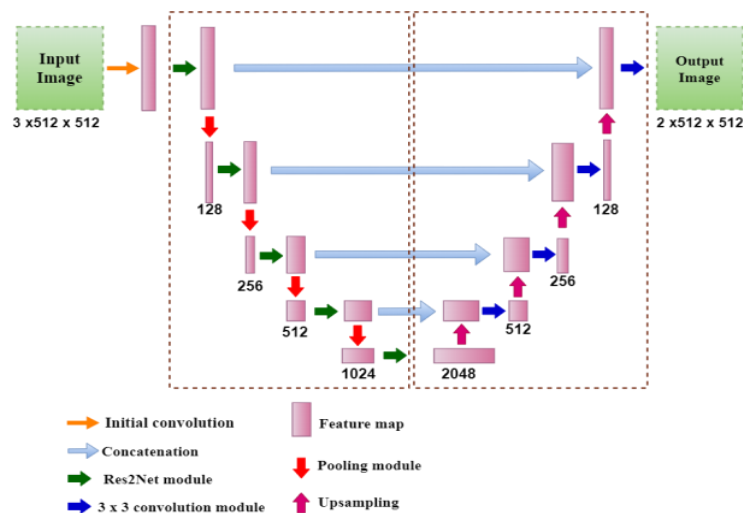


Figure 1. Framework of Res2-Unet

2.1. Res2-Unet

The architecture of the Res2-Unet is shown in Figure 1. This Res2-Unet model is almost similar structure to the Unet [21]. The $1 \times 3 \times 512 \times 512$ pixels input image is first convoluted utilizing an early convolution module, and then encoded using continuous convolution stages to create a feature map with 2,048 channels. After decoding, the map yields an output image measuring $1 \times 2 \times 512 \times 512$ pixels. With a

stride of 2, 1, and 1, and a kernel of a 3×3 element, the first convolution module consists of three convolution operations. The final image is a binary image with a 0–1 intensity scale, where 1 identifies buildings and 0 specifies background items.

As shown in Figure 2, the encoder section uses the Res2Net module in place of the conventional layer wise 3×3 convolution process in the bottle-neck layer by employing a group of filters on distinct subset of channels. After a 1×1 convolution the feature map F2 is divided into 4 subset feature maps such as F2_1, F2_2, F2_3, and F2_4. In the Res2Net module, F2_1 is used straight for concatenating with 3 additional output feature maps produced in the remaining framework. After F2_2 is convolution on by a 3×3 element kernel, F2_2o is added with F2_3. A 3×3 element kernel further convolves the summary feature map to produce F2_3o. In order to increase the scale variability, feature map F2_4o is created by convolving the real scale feature map F2_4 with a 3×3 element kernel, resulting in the synthesis of F2_3o. Then the augmentation of feature maps with distinct convolution repetitions expands the model's scale inconsistency and receptive field sizes for learning. The four distinct scale information is synthesised and feature map F3 is created by concatenating the output feature maps of each subbranch, F2_1o, F2_2o, F2_3o, and F2_4o. Then finally the output feature map F4 of the Res2Net module is generated by further convolving it using a 1×1 element kernel.

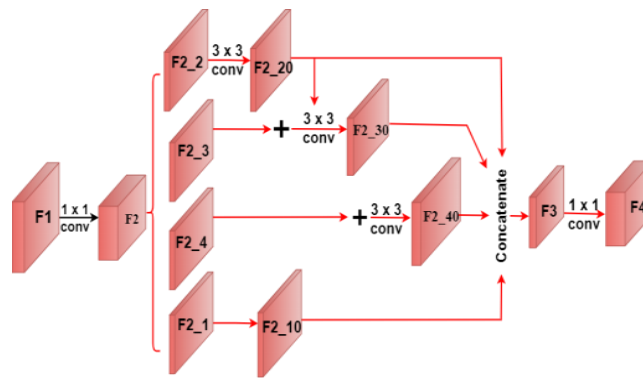


Figure 2. Architecture of Res2Net model

2.2. Pruning method based on SOKS

The SOKS contains two phases they are retraining and searching phase [22]. Coefficient matrices limited by several regularisation terms are used in the searching phase to find significant locations in the convolution kernels. Let $X \in \mathbb{R}^{c \times h \times w}$ represent the input tensor for a convolution layer, which has c channels and a size of $h \times w$. After convolution the outcomes are shown in (1).

$$Y = X * w \quad (1)$$

Here, $Y \in \mathbb{R}^{n \times h' \times w'}$ is the outcome with n channels and $h' \times w'$ in size, the convolution operator is denoted by $*$, and the convolution weight is represented by $w \in \mathbb{R}^{c \times k \times k \times n}$. Moreover, filter weights w is multiplied by utilizing a coefficient matrix F prior to convolution in order to investigate the best kernel shapes, then Y becomes:

$$Y = X * (F \odot w) \quad (2)$$

where, the element wise multiplication is denoted by \odot .

Then assign distinct coefficient matrices to every channel or use a single matrix for all channels. So, employ the similar coefficient matrix within each group after dividing these n filters equally into d ($1 \leq d \leq n$) groups. If $d = 1$, then every filter in the set of n filters learns the same kernel shapes and the same coefficient matrix. Every filter in the set of n is independent of the others and develops its own kernel shape if $d = n$. Every n/d filter learns the same kernel shapes and shares a single coefficient matrix if $1 < d < n$. Then $d \times k \times k$ is the size of F , and every 2-D matrix in F is referred as $F \in \mathbb{R}^{k \times k}$, where $i = 1, \dots, d$. Figure 3 illustrates the convolution process involving F .

To compress a CNN layer with L convolution layers, each 2-D coefficient matrix collected and obtain:

$$\mathcal{F} = \{F_1^1, \dots, F_d^1, \dots, F_i^l, \dots, F_1^L, \dots, F_d^L\} \tag{3}$$

where, the coefficient matrix for the i^{th} filter collection in the l^{th} convolution layer is represented by F_i^l . During training, several regularisation constraints are applied and all components in \mathcal{F} are initialised to 1. The network parameters F , will converge after the quantity of training rounds. Then the \mathcal{F} is utilized to identify significant kernel locations that are then maintained to create the ideal kernel shapes. Figure 4 illustrate the convolution operation with uneven kernel shapes. In order to attain high precision in the retraining phase, stripe-wise pruning is carried out after getting the ideal kernel shapes for every convolution layer, and from scratch, the compressed model is trained.

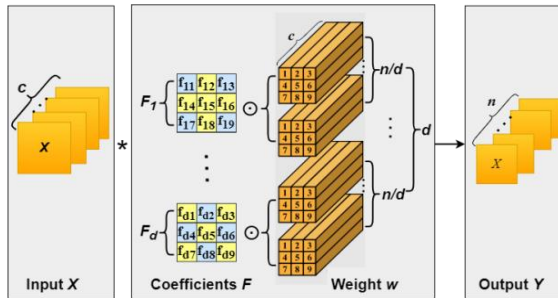


Figure 3. The coefficient matrix F is involved in the convolution process

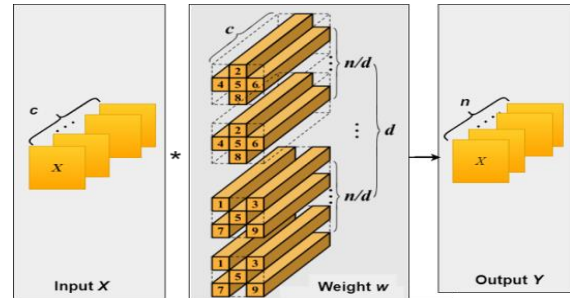


Figure 4. Irregularly shaped kernels in the convolution process

2.2.1. Adjustments made to the coefficient matrices

For image classification, the conventional training loss function can be expressed as (4):

$$L_0 = L_{classification} + \lambda L_2 \tag{4}$$

where, the classification loss is represented by $L_{classification}$, the penalty level is controlled by utilizing λ and L_2 is utilized to reduce the over fitting.

The regularization constraints L_s is added on the coefficient matrices \mathcal{F} in order to perform the automatic finding of the optimal kernel shapes. The entire training loss function is (5).

$$L = L_0 + L_s \tag{5}$$

To achieve model compression, it is necessary to train each coefficient matrix in \mathcal{F} to be sparse. This serves as a detector to locate significant kernel points. The application of sparse regularisations is a typical solution. Nevertheless, the recent research indicates that the contributions of the kernel parameters at various places differ. Furthermore, other concerns like the pixel shift problem need to be taken into account. Then L_s can be formulated as (6).

$$L_s = \lambda_1 L_{sparse} + \frac{\lambda_2}{2} L_{dir} + \frac{\lambda_3}{2} L_{group} \tag{6}$$

Where, sparse regularization term is denoted by L_{sparse} , then the group and direction wise regularization term is represented by L_{group} and L_{dir} . Moreover, these terms strengths are balanced by using λ_1, λ_2 and λ_3 . With regard to f_{ij} , the partial derivative of L_{sparse} is given by (7).

$$\frac{\partial L_{sparse}}{\partial f_{ij}} = k_j \cdot \text{sgn}(f_{ij}) \tag{7}$$

Where, the sign function is denoted by (\cdot)

Then the partial of L_{dir} with reverence to f_{ij} can be expressed as following using the chain rule (8).

$$\frac{\partial L_{dir}}{\partial f_{ij}} = \frac{L_{dir}}{\partial \bar{F}_j} \cdot \frac{\partial \bar{F}_j}{\partial f_{ij}} \tag{8}$$

The partial derivative of L_{group} with regard to f_{ij} can be expressed as in (9).

$$\frac{L_{group}}{\partial f_{ij}} = \begin{cases} \frac{\partial L_{group}}{\partial \bar{F}_i^{corner}} \cdot \frac{\partial \bar{F}_i^{corner}}{\partial f_{ij}}, & \text{if } j \in S_{corner} \\ \frac{\partial L_{group}}{\partial \bar{F}_i^{edge}} \cdot \frac{\partial \bar{F}_i^{edge}}{\partial f_{ij}}, & \text{if } j \in S_{edge} \\ \frac{\partial L_{group}}{\partial \bar{F}_i^{center}} \cdot \frac{\partial \bar{F}_i^{center}}{\partial f_{ij}}, & \text{if } j \in S_{center} \end{cases} \quad (9)$$

2.2.2. Pruning for the optimal kernel shapes

Every F_i^l coefficient matrix in \mathcal{F} undergoes training to become sparse, with many parameters near 0. Then the binary search algorithm (BSA) is suggested to determine the proper pruning threshold in order to achieve the essential compression rate. In order to prune unnecessary kernel places, the BSA is used to determine a threshold τ^* . The maximum absolute value of each parameter f_{ij} in F_i^l must first be determined the given a threshold τ . This value is represented as $|f|_{max}$, where i ($i = 1, \dots, d$) denotes the kernel group's index and j ($j = 1, \dots, 9$) denotes the kernel position's index. Moreover, if $|f_{ij}| < \tau|f|_{max}$, then prune the position j . After obtaining the best kernel shapes, stripe-wise pruning can be accomplished by removing irrelevant kernel parameters. Additionally, the binary search technique achieves the appropriate compression ratio. Then, in order to achieve acceptable performance, the pruned model is trained from scratch.

2.3. Quantification of data

CNN models for image recognition use 32-bit floating-point parameters [23]. To reduce computational cost, quantization methods compress the remaining weight parameters. Quantization is implemented by adding quantization operations prior to and subsequent to operations, minimising detection loss. Then the incremental network quantification technique was employed to retrain three parts of a floating-point network, achieving an 8-bit network that exceeded the weights. This method gradually removes unimportant weights from trained grids, decreasing the quantified network loss. The process involves matrix multiplication, pooling layer calculations, convolution, activation function calculations, and splicing operations to convert high-precision data into low-precision data. The packet quantization process increases, allowing filter-strip recombination results to be used for quantitative grouping. This process is interspersed into convolutional calculations, reducing computational costs and accelerating model efficiency. Quantifying data after grouping allows uninvolved data to be directly output, simplifying the calculation process.

3. RESULTS AND DISCUSSION

3.1. Dataset description

In this article, there are two datasets such as WHU and Massachusetts are utilized to training and testing the proposed model. First the WHU dataset is taken from WHO building dataset [24]. It contains satellite and aerial images. There are two subsets in the satellite imagery dataset. One of them is gathered from remote sensing resources such as QuickBird, Worldview series, IKONOS, ZY-3, and other sources, as well as from cities all over the world. Six nearby satellite photos with a 2.7 m ground resolution over 550 km² in East Asia make up the other satellite building sub-dataset. Then the second dataset Massachusetts is presented in Massachusetts buildings dataset [25]. Each of the 151 aerial images in the massachusetts buildings dataset covers an area of 2.25 square kilometres and has a pixel size of 1,500×1,500. Therefore, the whole dataset is around 340 square kilometres in size.

3.2. Result

The enhanced Res2-Unet model's accuracy at various thresholds is shown in Figure 5. From the Figure, the pruning is does not occur when the threshold is set to 0. The entire model was the smallest accurate when the threshold was set at 0.05, but the concurrent accuracy loss was also kept under control at roughly 0.01. Simultaneously, the 0.04 criterion was similarities to the 0.03 threshold. Moreover, the model's calculation quantity and number of parameters both are fairly dropped. Then the accuracy, flops, and parameters of a model at various threshold levels are displayed in the Table 1. The accuracy column shows the proportion of the model's correct detections. The amount floating-point processes the model does during inference, expressed in millions (M), is shown in the flop's column. The amount learnable parameter in the model is expressed in millions in the params column.

The performance of the proposed and existing approaches accuracy is shown in Figure 6. Compare to the existing methods such as Res2-Unet, Unet-Resnet50, and CT-Unet, the proposed technique accuracy value is very high. Moreover, the accuracy values are increases with increasing the epochs. Then all models exhibit increasing accuracy but the highest accuracy is given by the proposed approach. The proposed method has high efficiency compared with other approaches because it contains fewer parameters, therefore it achieves high accuracy. Hence, this outcome details the propose approach is the good one for accurately detecting the buildings. Then Table 2 displays the performance of the training and testing time of the Res2-Unet model. The amount of time needed to train a model on a dataset is called the training time, and the amount of time needed to test the model on a new dataset is called the testing time. From the Table, before pruning the Res2-Unet model took 587.48 ms to train and 195.53 ms to test, while after pruning the Res2-Unet model took 479.92 ms to train and 99.76 ms to test. In comparison to the before pruning model, the after pruning model requires less time for testing and training, suggesting that it is a more effective model. Because, the pruning technique utilized to reduce the parameters so the model takes less time for processing. Moreover, the pruning model is known as Enhanced Res2-Unet. The Table 3 shows the experimental result of various pruning techniques on deep learning models. There are six pruning techniques such as PFEC, FPGM, GAL, Hinge, HRank, and SOKS are compared with each other. These pruning methods are evaluated based on their trade-offs between model complexity (parameters and FLOPs) and accuracy. Finally, the SOKS pruning technique gives the better outcomes compare to the all methods. Because this pruning technique is utilized to reduce the maximum complexity of the model by decreasing the parameters and also it improves the detection accuracy by decreasing the overfitting. This result shows that the SOKS is best approach for reduce the parameters and enhance the accuracy.

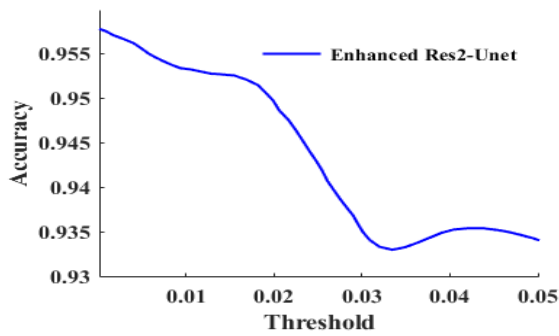


Figure 5. Accuracy of the proposed model at various thresholds

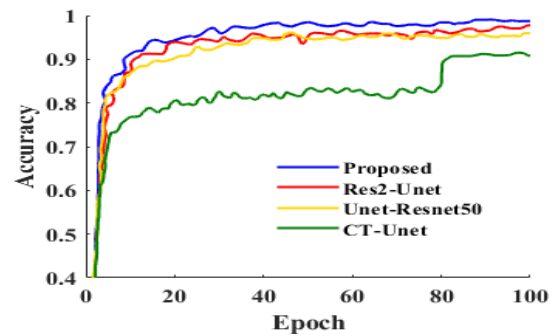


Figure 6. Comparison of accuracy values

Table 1. Pruning result of the various thresholds of the enhanced Res2-Unet model

Threshold	Accuracy	Params (M)	FLOPs (M)
0	0.9584	3.2	314
0.01	0.9532	2.38	256
0.02	0.9497	1.65	192
0.03	0.9350	1.1	147
0.04	0.9352	0.82	95
0.05	0.9343	0.62	43

Table 2. Comparison of before and after pruning the model

Model	Training time	Testing time
Res2-Unet (before pruning)	587.48 ms	195.53 ms
Enhanced Res2-Unet (after pruning)	479.92 ms	99.76 ms

Table 3. Performance of the various pruning methods

Pruning method	Params.	FLOPs	Accuracy
PFEC	5.4M	206M	93.40%
FPGM	5.4M	206M	93.54%
GAL	3.36M	189M	93.77%
Hinge	2.99M	191M	93.59%
HRank	2.51M	146M	93.43%
SOKS	1.09M	89.6M	96.36%

The performance of the Enhanced Res2-Unet model with various pruning ratios is displayed in the Table 4. The percentage of parameters that were pruned from the original model is shown in the pruning ratio column. SOKS retrains the compressed model and does searches from scratch that values are shown in SOKS column. Then before and after fine-tuning, the accuracy of the model is displayed in the pre-trained and Fine-tuning columns, respectively. Based on the table, the model's accuracy declines as the pruning ratio increases. However, the SOKS value falls as the pruning ratio rises, suggesting that the model gets more effective and less complicated. The values for pre-trained and fine-tuning indicate that adjusting the pruned model can increase its accuracy.

Table 4. Performance of SOKS, pre-trained, and fine-tuning

Network	Pruning ratio	SOKS	Pre-trained	Fine-tuning
Enhanced Res2-Unet	70%	94.65	94.58	94.08
	60%	95.27	94.25	93.77
	50%	91.25	90.68	90.98
	40%	92.44	90.89	90.48

Figure 7 illustrates the WHU dataset with various models perform in terms of building detection. From the Figure the input image is called the original image, and the intended outcome is called the ground truth. The output of several models, including CT-Unet, Unet-Resnet 50, Res2-Unet, and proposed, is displayed in the other columns. Then these models are used for building detection from satellite images in WHU dataset. In terms of building detection accuracy, the suggested model outperforms the Res2-Unet model, as seen in the Figure. The Res2-Unet model has more complexity because it contains more parameters so it causes the overfitting but the proposed model has very less and relevant parameters, therefore it gives the high building detection accuracy. In comparison to the other models, the accuracy of the CT-Unet and Unet-Resnet 50 models is lower.

The WHU dataset overall performance such as precision, recall, IOU (intersection over union), and F1-score are shown in Figure 8. The proposed model is compared with other three existing approaches such as CT-Unet, Unet-Resnet 50, and Res2-Unet. When it comes to precision, recall, IOU, and F1-scores, the suggested model performs better than any other model. Moreover, the high precision value is indicated that the model minimizes the false positive detections; the high recall value indicates that the model correctly identifies the positive instances; then the high IOU values are indicating that the model's detections align well with the ground truth; and the high F1-score value is indicating that the models attain a good trade-off between precision and recall. Then, the existing models are achieving the lower scores in all categories. Table 5 compares models on the WHU building dataset, showing the suggested model yielding 89.95% precision, 96.36% recall, 90.43% IOU, and 85.05% F1-score. Table 6 displays experimental results of various networks using SOKS pruning approach on WHU dataset, showing Res2-UNet network with maximum accuracy of 96.36% with 1,432 ms latency, and 89.6M FLOPs. Moreover, the SOKS pruning approach in the VGG16 network achieved high accuracy with 87.3 million FLOPs and low latency, while Res2-Unet with SOKS provided the highest accuracy.

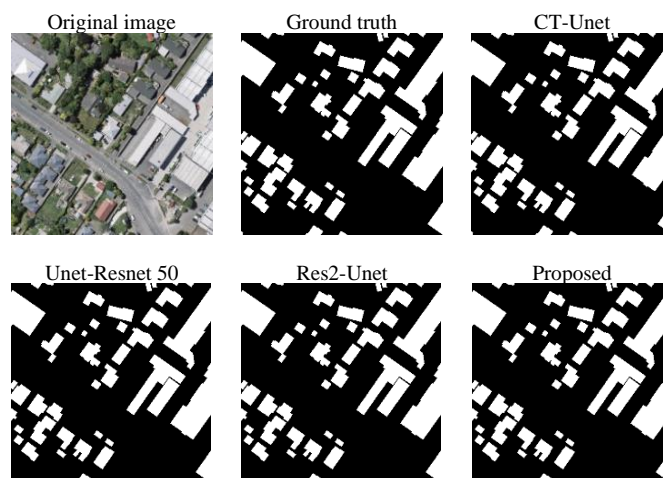


Figure 7. Visual comparisons of WHU dataset

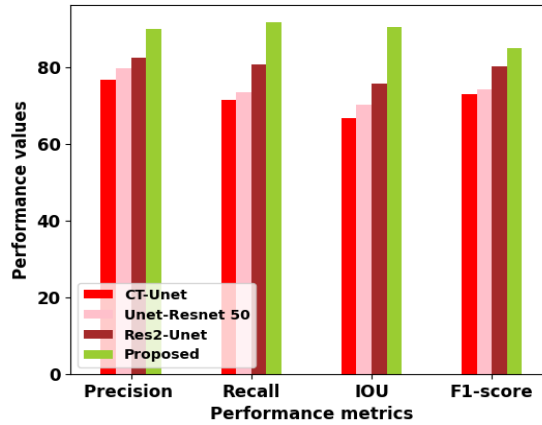


Figure 8. Performance of the WHU building dataset

Table 5. Performance of other models on the WHU building dataset

Method	Recall	Precision	F1	IOU
CT-Unet	71.35	76.58	72.97	66.58
Unet-Resnet 50	73.48	79.64	74.29	70.14
Res2-Unet	80.59	82.49	80.14	75.67
Proposed	91.68	96.36	85.05	90.43

Table 6. The SOKS performed with different networks on WHU dataset

Network with method	FLOPs	Latency	Accuracy
Res2-UNet with SOKS	89.6M	1.432 ms	96.36%
VGG16 with SOKS	87.3M	1.647 ms	94.11%
ResNet-20 with SOKS	15.6M	2.326 ms	91.83%
ResNet-32 with SOKS	31.7M	3.403M	92.44%

The performance of the four various models in building detection is demonstrated in Figure 9, which shows the Massachusetts building dataset. Visual comparisons between the ground truth, each model’s output, and the original image are displayed in this figure. The proposed model detection is almost similar to the ground truth; it signifies that the model is performing accurately. Compare to the all four models proposed technique achieve the high accuracy and it accurately detect the buildings. Therefore, as a result the proposed enhanced Res2-Unet is the best technique for detecting the building.

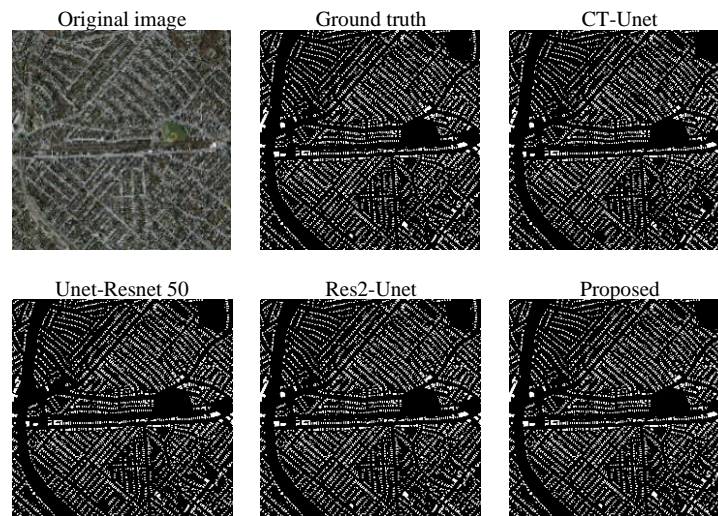


Figure 9. Visual comparisons of Massachusetts building dataset

The overall performance of the Massachusetts building dataset is shown in Figure 10. From the Figure shows, there are four metrics are calculated in this work such as recall, F1-score, precision and IOU and also four models are comparing each other. Furthermore, compare to the existing models, the proposed model achieves the high F1-score, precision, recall, and IOU values. This better performance indicates that the model achieves a good trade-off between precision and recall, minimizing false positives, accurately identifying positive instances and detections align well with ground truth. Then the value of the various metrics in massachusetts building dataset is detailed in Table 7. The presented model attains the 92.45% recall, 90.13% F1-score, 95.87% precision, and 94.46% IOU. Table 8 shows the SOKS pruning technique is performed with different networks on massachusetts dataset. From the table the number of floating-point operations needed to run the model is indicated by FLOPs; latency is a measure of how long the model takes to run; the model accuracy is represented in accuracy column. With 94.8M FLOPs and 1,524 ms latency, the Res2-UNet network using the SOKS pruning approach obtained the maximum accuracy of 95.87%, as shown in this table. Achieving an accuracy of 94.03% with 89.6M FLOPs and 1,656 ms of delay, the VGG16 network with the SOKS pruning method 1. 91.18% and 93.42%, respectively, were the accuracy rates attained by the ResNet-20 and ResNet-32 networks utilising the SOKS pruning technique. Additionally, Tables 9 and 10 shows the state-of-art models of WHU and Massachusetts building datasets. In Table 9 there are four state-of-art approaches are compared to the suggested model and in Table 10, eight state-of-art models compare to the suggested method. Finally, from these two tables the proposed model gives the better performance.

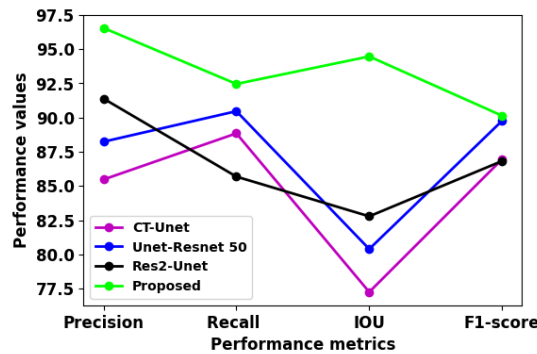


Figure 10. Performance of the Massachusetts building dataset

Table 7. Performance of other models on the massachusetts building dataset

Approaches	Precision	Recall	IOU	F1
CT-Unet	85.47	88.84	77.25	86.95
Unet-Resnet 50	88.23	90.46	80.39	89.76
Res2-Unet	91.38	85.68	82.78	86.82
Proposed	95.87	92.45	94.46	90.13

Table 8. The various networks performed with SOKS on massachusetts dataset

Network with method	FLOPs	Latency	Accuracy
Res2-UNet with SOKS	94.8M	1,524 ms	95.87%
VGG16 with SOKS	89.6M	1,656 ms	94.03%
ResNet-20 with SOKS	23.8M	2,331 ms	91.18%
ResNet-32 with SOKS	37.1M	3,410M	93.42%

Table 9. Performance of the published data (%) in WHU dataset

Approaches	Accuracy	Recall	F1-measure	Iou
SiU-Net [26]	65.3	86.9	74.56	59.4
U-Net [26]	72.5	79.6	75.88	61.1
SR-FCN [27]	79.0	77.0	77.99	64.0
Res2-UNet [21]	81.29	78.64	79.99	64.0
Proposed	95.87	91.68	90.43	85.05

Table 10. Performance of the published data (%) on massachusetts building dataset

Approaches	Accuracy	Recall	F1-measure	IOU
SegNet [28]	88.2	82.2	85.1	74.0
FCN [28]	89.5	86.7	88.1	78.8
U-Net [28]	89.9	86.9	88.4	80.3
FRRN [28]	92.8	79.6	85.7	74.9
Deeplab-v3 [29]	-	-	81.34	68.55
ENRU-Net [29]	-	-	84.41	73.02
MSCRF [30]	89.93	80.14	84.75	71.19
Res2-Unet [21]	92.12	89.27	90.67	82.93
Proposed	96.36	92.45	94.46	86.82

4. CONCLUSION

In the context of urban development and digital city mapping, the accurate building detection is essential. Recent observations suggest that the deep learning approaches significantly impacts building detection. Our findings provide conclusive evidence that this phenomenon is associated with a positive change in the accuracy and efficiency of building by enhancing the deep learning approach. Therefore, the SOKS pruning technique is proposed in this paper for prune the CNN from Res2-Unet model. Recently, network pruning is an efficient way of compressing networks by eliminating unnecessary or redundant parameters. The main aim of this paper is to detect the buildings accurately by using Res2-Unet. Therefore, first the pruning technique is applied to the CNN layer for decreasing the complexity and parameters of the model. After the CNN pruning the Res2-Unet model is applied in the quantitative process, which is utilized to fine-tune the network. Moreover, this improved model is accurately detecting the buildings from the satellite images. Compare to the current building models our proposed method achieves the better performance. However, the satellite imagery shows a significant class imbalance, potentially leading to biased predictions favouring non-building pixels, potentially affecting the model's performance. In the future work to address the class imbalance issue in building detection by developing the new robust methods.




REFERENCES

- [1] K. I. Rufai, "Attack detection in a rule-based system using fuzzy spiking neural P system," *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 5, no. 1, p. 11, Apr. 2016, doi: 10.11591/ijict.v5i1.pp11-20.
- [2] A. F. H. Mahomodally, G. Suddul, and S. Armoogum, "Machine learning techniques for plant disease detection: an evaluation with a customized dataset," *International Journal of Informatics and Communication Technology*, vol. 12, no. 2, pp. 127–139, Aug. 2023, doi: 10.11591/ijict.v12i2.pp127-139.
- [3] M. J. Alam, M. A. Awal, and M. N. Mustafa, "Crops diseases detection and solution system," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 3, pp. 2112–2120, Jun. 2019, doi: 10.11591/ijece.v9i3.pp2112-2120.
- [4] F. H. Nahhas, H. Z. M. Shafri, M. I. Sameen, B. Pradhan, and S. Mansor, "Deep learning approach for building detection using LiDAR-Orthophoto Fusion," *Journal of Sensors*, vol. 2018, pp. 1–12, Aug. 2018, doi: 10.1155/2018/7212307.
- [5] G. Prathap and I. Afanasyev, "Deep learning approach for building detection in satellite multispectral imagery," in *9th International Conference on Intelligent Systems 2018: Theory, Research and Innovation in Applications, IS 2018 - Proceedings*, Sep. 2018, pp. 461–465, doi: 10.1109/IS.2018.8710471.
- [6] J. H. Jeppesen, R. H. Jacobsen, F. Inceoglu, and T. S. Toftegaard, "A cloud detection algorithm for satellite imagery based on deep learning," *Remote Sensing of Environment*, vol. 229, pp. 247–259, Aug. 2019, doi: 10.1016/j.rse.2019.03.039.
- [7] H. Perez, J. H. M. Tah, and A. Mosavi, "Deep learning for detecting building defects using convolutional neural networks," *Sensors (Switzerland)*, vol. 19, no. 16, p. 3556, Aug. 2019, doi: 10.3390/s19163556.
- [8] Z. Liu, B. Chen, and A. Zhang, "Building segmentation from satellite imagery using U-Net with ResNet encoder," in *Proceedings - 2020 5th International Conference on Mechanical, Control and Computer Engineering, ICMCC 2020*, Dec. 2020, pp. 1967–1971, doi: 10.1109/ICMCC51767.2020.00431.
- [9] S. Aghayari, A. Hadavand, S. M. Niazi, and M. Omidalizari, "Building detection from aerial imagery using Inception Resnet Unet and Unet architectures," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 10, no. 4/W1-2022, pp. 9–17, Jan. 2023, doi: 10.5194/isprs-annals-X-4-W1-2022-9-2023.
- [10] C. Chen, W. Gong, Y. Chen, and W. Li, "Learning a two-stage CNN model for multi-sized building detection in remote sensing images," *Remote Sensing Letters*, vol. 10, no. 2, pp. 103–110, Feb. 2019, doi: 10.1080/2150704X.2018.1528398.
- [11] A. Saidi, S. Ben Othman, M. Dhoubi, and S. Ben Saoud, "CNN inference acceleration on limited resources FPGA platforms epilepsy detection case study," *International Journal of Informatics and Communication Technology*, vol. 12, no. 3, pp. 251–260, Dec. 2023, doi: 10.11591/ijict.v12i3.pp251-260.
- [12] L. Gong, X. Huang, J. Chen, M. Xiao, and Y. Chao, "Multiscale leapfrog structure: an efficient object detector architecture designed for unmanned aerial vehicles," *Engineering Applications of Artificial Intelligence*, vol. 127, p. 107270, Jan. 2024, doi: 10.1016/j.engappai.2023.107270.
- [13] D. Li, L. Li, Z. Chen, and J. Li, "Shift-ConvNets: small convolutional kernel with large kernel effects," *arXiv*, 2024, [Online]. Available: <http://arxiv.org/abs/2401.12736>.
- [14] H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang, "Network trimming: a data-driven neuron pruning approach towards efficient deep architectures," *arxiv*, Jul. 2016, [Online]. Available: <http://arxiv.org/abs/1607.03250>.
- [15] L. Capogrosso, F. Cunico, D. S. Cheng, F. Fummi, and M. Cristani, "A machine learning-oriented survey on tiny machine learning," *IEEE Access*, vol. 12, pp. 23406–23426, 2024, doi: 10.1109/ACCESS.2024.3365349.
- [16] S. Vadera and S. Ameen, "Methods for pruning deep neural networks," *IEEE Access*, vol. 10, pp. 63280–63300, 2022, doi: 10.1109/ACCESS.2022.3182659.




- [17] D. Wu and Y. Wang, "Adversarial neuron pruning purifies backdoored deep models," *Advances in Neural Information Processing Systems*, vol. 20, pp. 16913–16925, 2021.
- [18] W. Zhang, N. Wang, K. Chen, Y. Liu, and T. Zhao, "A pruning method for deep convolutional network based on heat map generation metrics," *Sensors*, vol. 22, no. 5, p. 2022, Mar. 2022, doi: 10.3390/s22052022.
- [19] M. Li, M. Zhao, T. Luo, Y. Yang, and S. L. Peng, "A compact parallel pruning scheme for deep learning model and its mobile instrument deployment," *Mathematics*, vol. 10, no. 12, p. 2126, Jun. 2022, doi: 10.3390/math10122126.
- [20] N. Qin, L. Liu, D. Huang, B. Wu, and Z. Zhang, "Leannet: an efficient convolutional neural network for digital number recognition in industrial products," *Sensors*, vol. 21, no. 11, p. 3620, May 2021, doi: 10.3390/s21113620.
- [21] F. Chen, N. Wang, B. Yu, and L. Wang, "Res2-Unet, a new deep architecture for building detection from high spatial resolution images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 1494–1501, 2022, doi: 10.1109/JSTARS.2022.3146430.
- [22] G. Liu, K. Zhang, and M. Lv, "SOKS: automatic searching of the optimal kernel shapes for stripe-wise network pruning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 12, pp. 9912–9924, Dec. 2023, doi: 10.1109/TNNLS.2022.3162067.
- [23] M. Zhao, X. Tong, W. Wu, Z. Wang, B. Zhou, and X. Huang, "A novel deep-learning model compression based on filter-stripe group pruning and Its IoT application," *Sensors*, vol. 22, no. 15, p. 5623, Jul. 2022, doi: 10.3390/s22155623.
- [24] WHO Building Dataset, available: <https://www.kaggle.com/datasets/xiaoqian970429/whu-building-dataset/data>
- [25] The massachusetts buildings dataset, available: <https://www.kaggle.com/datasets/balraj98/massachusetts-buildings-dataset>
- [26] S. Ji, S. Wei, and M. Lu, "Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 1, pp. 574–586, Jan. 2019, doi: 10.1109/TGRS.2018.2858817.
- [27] S. Ji, S. Wei, and M. Lu, "A scale robust convolutional neural network for automatic building extraction from aerial and satellite imagery," *International Journal of Remote Sensing*, vol. 40, no. 9, pp. 3308–3322, May 2019, doi: 10.1080/01431161.2018.1528024.
- [28] Y. Liu, L. Gross, Z. Li, X. Li, X. Fan, and W. Qi, "Automatic building extraction on high-resolution remote sensing imagery using deep convolutional encoder-decoder with spatial pyramid pooling," *IEEE Access*, vol. 7, pp. 128774–128786, 2019, doi: 10.1109/ACCESS.2019.2940527.
- [29] S. Wang, X. Hou, and X. Zhao, "Automatic building extraction from high-resolution aerial imagery via fully convolutional encoder-decoder network with non-local block," *IEEE Access*, vol. 8, pp. 7313–7322, 2020, doi: 10.1109/ACCESS.2020.2964043.
- [30] Q. Zhu, Z. Li, Y. Zhang, and Q. Guan, "Building extraction from high spatial resolution remote sensing images via multiscale-aware and segmentation-prior conditional random fields," *Remote Sensing*, vol. 12, no. 23, pp. 1–18, Dec. 2020, doi: 10.3390/rs12233983.

BIOGRAPHIES OF AUTHORS



Arulappan Amala Arul Reji    received his first degree from lekshmpuram college of arts and science in Manonmaniam Sundaranar University, Computer Information Technology in 2004, Master degree from Vivekananda college of arts and science in 2006, M.phil. in Manonmaniam Sundaranar University in 2009, she also done her B.Ed. in 2013. Doing research in image processing. She can be contacted at email: amala.arul.reji@yahoo.com.



Sathiyamoorthy Muruganantham    received his first degree from Madurai Kamaraj university, Computer Science, Madurai in 1992 He has also received M.C.A from Manonmaniam Sundaranar university, Tirunelveli in 1996 and Ph.D. degree Manonmaniam Sundaranar university, computer science. Information Technology and Engineering in 2013. He is currently working as an assistant professor in the department of M.C.A. at S.T. Hindu College, Nagercoil, Tamil nadu, India-629002 since 1996. He can be contacted at email: smuru2013@gmail.com.