DenseNetx: Efficient DenseNets for Remote Scene Classification without Pretraining

Russo Mohammad Ashraf Uddin
Department of Electrical, Electronic and Computer Engineering
University of Ulsan
Ulsan, Korea
ashrafrusso@islab.ulsan.ac.kr

Tien-Dat Tran
Department of Electrical, Electronic and Computer Engineering
University of Ulsan
Ulsan, Korea
tdat@islab.ulsan.ac.kr

Ge Cao
Department of Electrical, Electronic and Computer Engineering
University of Ulsan
Ulsan, Korea
caoge9706@gmail.com

Kang-Hyun Jo
Department of Electrical, Electronic and Computer Engineering
University of Ulsan
Ulsan, Korea
acejo@ulsan.ac.kr

***Abstract*— Remote sensing scene classification is growing fast in demand and application within the Earth Observation domain. Satellite Image data are usually high resolution but low in number. DenseNet architectures are quite powerful and achieve good accuracy in this task even without large-scale pretraining from ImageNet-like datasets. But, DenseNet lacks efficiency and is considered a quite heavy model by modern standards. We propose DenseNetx, a family of efficient densenet architecture which can dramatically reduce computation costs while outperforming the baseline model. In short, we use a larger input size while aggressively downsampling in the stem block using two 3x3 convolutions of stride 2, and use large-kernel depthwise-separable convolution in the denselayer to achieve higher efficiency. Our results on the WHU-RS19 and Optimal-31 scene classification datasets show that our model can outperform the baseline at 20% reduced parameters and 53% fewer flops, while achieving up to 4.5% increased accuracy with a larger input while retaining efficiency.**

Keywords—remote sensing, scene classification, densenet, efficient

I. INTRODUCTION

Recent developments in machine learning have led to remarkable performance in image-data analysis, particularly in computer vision tasks across various domains. Deep learning approaches have played a significant role in this advancement by using modular and scalable deep neural network architectures to process large amounts of data. These approaches have also shown immense potential in remote sensing, especially in Earth Observation, where they can analyze various types of large-scale satellite data. Most of the contributions in this area focus on image-scene classification tasks, such as land-use and land-cover identification, which involve the analysis, characterization, and classification of changes in the landscape caused by either human activities or natural elements [1]. Traditionally, these tasks have been addressed through either pixel-level or object-level classification paradigms, but these methods have limitations. Pixel-level approaches are not scalable for high-resolution images, while object-level methods struggle with images containing diverse and indistinguishable objects. Scene-level classification, a relatively new paradigm, has shown significant improvements in performance by leveraging the capabilities of deep learning to learn semantically meaningful representations of more sophisticated patterns in an image. These developments have the potential to advance image-data analysis in various fields.

ImageNet Pretraining is a popular technique for improving image classification accuracy in smaller datasets with fewer annotated samples. Proposed first in [2] and later in detail in [3], ImageNet pretraining increases the deep learning model's capability to generalize as the ImageNet dataset has 1000 different image classes. Large-Scale pretraining has become a very useful technique since then[4] and is used widely in downstream tasks such as object detection[5], semantic or instance segmentation [6], pose estimation [7] as well as image classification in other smaller datasets. But ImageNet is a large dataset and pretraining any new model on it takes a considerable amount of time depending on the available hardware, it is quite difficult for many small-scale researchers to pretrain their proposed model on ImageNet. On the other hand, in [8] authors argue that with sufficient data and training iterations even random initialization of model weights can lead to comparable performance to ImageNet pretrained model. Also, ImageNet pretraining can sometimes lead the model to overfit to ImageNet classes. As the remote sensing datasets are captured mainly using satellites or drones, we only have the aerial view of the objects, which can be vastly different from their regular view counterpart images. Since ImageNet consists of images captured in regular camera view, we consider the best approach to develop our model from scratch, for our remote sensing image classification task.

Authors in [1] present "AITLAS", a benchmark arena for Earth Observation (EO), which compares the state-of-the-art deep learning architectures in 22 different remote sensing datasets in multi-class and multi-label classification tasks. Among their results, DenseNet161 [9] performs best in most of the remote sensing datasets, in the trained-from-scratch scenario. While the dense connection in the DenseNet architecture leads to good performance in the remote sensing image classification tasks, it contains around 28M parameters and has 7.82GFlops for an input image size of 224x224. Which cannot compete with other recent models in efficiency and model size. Deep learning models are integrated into many edge devices in today's world, and the efficiency of these models is one of the major concerns as well as accuracy. In the case of remote sensing, embedded mini or microcomputers inside a drone can be used to analyze the landscape using a model which has high accuracy and high

efficiency. Thus, we develop "DenseNetx", a modified family of DenseNet architectures that are highly efficient and can outperform the original architecture in accuracy as well. We use WHU-RS19 and Optimal-31 remote sensing datasets for evaluation.

We design DenseNetx aiming to reduce redundant information from the feature maps and increase computational efficiency while not sacrificing much of the image classification capability of the model. Inspired by [10], we employ large kernel (5x5) depth-wise separable convolutions to replace the regular 3x3 convolution in the dense layers, which reduces the model parameters and Flops by 20% and 26% respectively. This helps the model capture global information much more effectively, and for remote sensing images it is very crucial. Since, the aerial view usually contains much more objects and thus information than a regular image, understanding the global context amplifies the chances for the input to be accurately classified. Afterward, we modify the stem block of the baseline DenseNet161, to further reduce the Flops by ~53%, while still managing to capture the important features from the input and maintain the model accuracy. Our contribution can be summarized below-

1. An efficient Stem Block which reduces Flops by >50% than baseline while still capturing critical features from the input and thus retaining accuracy

2. A Large-Kernel Depthwise-Separable conv. based DenseLayer to capture better global information while increasing efficiency in parameters and flops.

3. Densenetx Architecture for remote sensing image classification, which uses 20% fewer parameters and 79% fewer Flops than the baseline with only 1.7% accuracy drop, and outperforms the baseline at 1.5x input while still having 53% fewer Flops and 20% fewer parameters.

The rest of the paper is organized as- Section II contains related research about the topics of discussion. Section III described the methodologies used. Section IV provides implementation details and dataset information. Section V elucidates the experimental analysis and ablation studies. Finally, Section VI concludes the paper.

II. Related Research

A. Earth Observation

Earth observation through satellite imagery has become an essential tool for understanding and monitoring global environmental changes, such as deforestation, urbanization, and climate change [11]. Satellite image classification plays a crucial role in many applications, including land use and land cover mapping, agriculture monitoring, disaster management, and urban planning [12], [13]. Another area of application for satellite image classification is disaster management, where satellite imagery can provide a rapid assessment of damage and support disaster response efforts [14]. Multi-source data fusion, including satellite imagery, climate data, and ground observations, can enhance the accuracy of satellite image classification [15]. Authors in [1] detail a large-scale study on 22 datasets with numerous combinations of deep learning models and compare their effectiveness.

B. DenseNet and Variants

DenseNets have been widely popular for various image-processing tasks since their inception. There have also been many variants proposed by other researchers trying to improve the baseline model for specific or general tasks. [16] introduced binary connect convolutional layers to the DenseNet architecture, which reduced the memory requirements and improved the network's accuracy. In [17], authors modified the DenseNet architecture by introducing parallel connections between different DenseNet blocks, which improved the network's accuracy and reduced the training time. [18] proposed scale-invariant convolutional layers to the DenseNet architecture, which improved the network's performance on datasets with varying scales. [19] modified the DenseNet architecture by introducing dual attention modules, which improved the network's ability to capture long-range dependencies and spatial context in scene segmentation tasks. An energy and computation-efficient architecture called VoVNet comprised of One-Shot Aggregation (OSA) was proposed by the authors in [20] from baseline DenseNet architectures.

C. Efficient CNNs for Classification

Model efficiency has seen major interest from researchers in recent years. Depthwise-Separable Convolution was first proposed by [21] introducing the architecture Xception, which greatly reduces parameters and flops of a regular convolution operation, while still retaining good feature-capturing capabilities. MobileNets [22] are based on this idea and introduced a family of efficient CNN architectures that are designed to be fast and lightweight for mobile and embedded devices. [23] introduced ShuffleNet, a CNN architecture that uses channel shuffling and pointwise group convolutions to achieve high accuracy with low computational cost. EfficientNets proposed by [24] are a family of CNN architectures that use a novel compound scaling method to achieve state-of-the-art performance with significantly fewer parameters and less computational cost. SqueezeNet from [25] is a CNN architecture that uses a combination of 1x1 and 3x3 convolutions to reduce the number of parameters while maintaining high accuracy. [26] introduced ProxylessNAS, a neural architecture search method that can directly optimize CNN architectures for specific hardware and tasks, resulting in highly efficient and accurate models. RTM-Det [10] introduced the modification of the famous darknet-53 architecture using large-kernel depthwise-separable convolutions.

III. Methodology

A. Baseline

DenseNet161 is a subset of the DenseNet architecture, which Huang et al. proposed in [9]. DenseNet161 is made up of numerous dense blocks, each with several convolutional layers and a set number of output channels. The feature maps from all preceding layers are concatenated and supplied as input to each subsequent layer inside each dense block. This fosters feature reuse while also lowering the possibility of vanishing gradients. DenseNet incorporates transition layers, which use a combination of average pooling and convolutional layers to minimize the number of feature mappings. This helps to manage the number of feature maps

Layers	Output Size	DenseNet161	Output Size	DenseNetx161
Convolution(Stem Block)	112 x 112	7 x 7 conv, stride 2	112 x 112	3 x 3 conv, stride 2
			56 x 56	3 x 3 conv, stride 2
Pooling	56 x 56	3 x 3 max pool, stride 2	28 x 28	3 x 3 max pool, stride 2
Dense Block (1)	56 x 56	[1 x 1 conv 3 x 3 conv] x 6	28 x 28	[1 x 1 conv 5 x 5 depthwise conv 1 x 1 pointwise conv] x 6
Transition Layer (1)	56 x 56	1 x 1 conv	28 x 28	1 x 1 conv
	28 x 28	2 x 2 average pool, stride 2	14 x 14	2 x 2 average pool, stride 2
Dense Block (2)	28 x 28	[1 x 1 conv 3 x 3 conv] x 12	14 x 14	[1 x 1 conv 5 x 5 depthwise conv 1 x 1 pointwise conv] x 12
Transition Layer (2)	28 x 28	1 x 1 conv	14 x 14	1 x 1 conv
	14 x 14	2 x 2 average pool, stride 2	7 x 7	2 x 2 average pool, stride 2
Dense Block (3)	14 x 14	[1 x 1 conv 3 x 3 conv] x 36	7 x 7	[1 x 1 conv 5 x 5 depthwise conv 1 x 1 pointwise conv] x 36
Transition Layer (3)	14 x 14	1 x 1 conv	7 x 7	1 x 1 conv
	7 x 7	2 x 2 average pool, stride 2	4 x 4	2 x 2 average pool, stride 2
Dense Block (4)	7 x 7	[1 x 1 conv 3 x 3 conv] x 24	4 x 4	[1 x 1 conv 5 x 5 depthwise conv 1 x 1 pointwise conv] x 24
Classification Layer	1 x 1	7 x 7 global average pool	1 x 1	7 x 7 global average pool
		N-D fully-connected, softmax		

Fig. 1. DenseNetx161 Architecture with comparison of the Baseline DenseNet161

as the network grows deeper, while also giving a mechanism to downsample the feature maps and minimize the network's computational cost. We choose DenseNet161 as our baseline model since it performs best, in most of the RSI datasets in [1].

B. DensexLayer

A Depthwise Separable Convolution is made up of two major operations: depthwise and pointwise convolution. Depthwise convolution employs a single convolutional filter per input channel, with no channel mixing. This procedure generates a collection of output feature maps with the same number of channels as the input but a lower spatial resolution. This operation's purpose is to extract low-level information from the input. The output of the depthwise convolution is sent through a 1x1 convolutional filter in pointwise convolution (also known as 1x1 convolution). This technique combines the feature map channels and generates a collection of output feature maps with possibly higher dimensionality than the input. The purpose of this operation is to obtain higher-level features that are more discriminative for the specified task by performing a linear combination of the low-level features recovered via depthwise convolution. Depthwise Separable Convolution, which combines depthwise and pointwise convolutions, can achieve equivalent or greater performance than classic convolutional layers while requiring fewer parameters and processes. As a result, it is especially helpful for mobile and embedded devices with low processing resources. In our densexLayer, we replace the 3x3 convolution with a 5x5 depthwise convolution followed by a 1x1 pointwise convolution in each dense layer. Which reduces the parameters of the baseline model by ~20% and GFlops by ~26%.

C. Efficient Stem Block

The choice of using a 7x7 convolution in the Stem block of VGG and other architectures is based on the idea that a larger receptive field can capture more global information from the input images, which may be useful for some tasks such as object recognition. However, this comes at the cost of increased computational requirements and may not always be necessary or optimal for all tasks [20]. Two 3x3 convolutions usually produce a similar feature map as one 5x5 convolution. But in operations, are much more efficient. Replacing the conv. 7x7, stride 2 operations in the baseline stem block, with two conv. 3x3, stride 2 operations followed by a maxpool 3x3, stride 2 will downsample the input 3 times instead of 2. While this may seem counterintuitive, in reality, this helps remove unwanted redundancy from large input sizes and trains the model to achieve higher accuracy.

D. DenseNetx161 Architecture

Finally, by replacing the regular denselayer with the proposed densexLayer, and using the efficient stem block we construct our DenseNetx architecture. Fig. 1 shows our model architecture in contrast with the baseline. We use the 161-layer version to make the comparison fair and showcase the effectiveness of our ideas. By aggressively downsampling in the stem block we can reduce the flops drastically, while the large kernel depthwise-separable convolution brings large receptive field in the denselayer, capturing more global context than before while reducing parameters and flops simultaneously. The results are discussed in the experiments section in detail. In our architecture we use growth-rate = 48 same as baseline.

TABLE I. Evaluation of WHU-RS19 using DenseNetx161 architecture. Each model is trained for 300 epochs.

Model	Image Size	Params. (M)	GFlops	Accuracy(%)	Precision(%)	F1_score(%)	Training Time(s)
DenseNet161	1x	26.51	7.82	86.56	87.63	86.52	2196
ResNet152	1x	58.18	11.58	71.14	71.72	70.84	2026
ResNet50	1x	23.55	4.12	76.61	78.25	76.38	883
DenseNetx161 (ours)	1x	21.21	1.65	85.07	86.19	84.7	2091
DenseNetx161 (ours)	1.5x	21.21	3.65	87.06	87.13	86.82	2230
DenseNetx161 (ours)	2x	21.21	6.76	90.04	90.85	89.94	2747

TABLE II. Evaluation of Optimal-31 using DenseNetx161 architecture.

Model	Image Size	Params. (M)	GFlops	Accuracy (%)	F1_score (%)	mIoU (%)	Training Time(s)
DenseNet161	1x	26.51	7.82	69.62	69.91	55.1	1338
ResNet152	1x	58.18	11.58	65.59	65.83	49.93	2730
ResNet50	1x	23.55	4.12	68.54	68.68	53.05	1202
DenseNetx161 (ours)	1x	21.21	1.65	70.43	69.9	54.9	1346
DenseNetx161 (ours)	1.14x	21.21	2.2	71.5	70.85	55.77	1589
DenseNetx161 (ours)	2x	21.21	6.76	74.19	73.56	59.71	3598

IV. Implementation and Dataset Details

We evaluate our proposed model in two remote sensing scene classification datasets- WHU-RS19 and Optimal-31. WHU-RS19 contains 19 classes of satellite images of 600x600 dimension, each class containing at least 50 images and in a total of 1005 images. Optimal-31 is also a scene classification dataset, but it's more difficult as it contains 31 classes and an image dimension of 256x256. Each class has at least 60 images and the total number of images is 1860. Evaluation metrics for WHU-RS19 are accuracy, precision, and F1 score, and for Optimal-31 accuracy, F1 score and mean IoU is used. All the ablation experiments are done in the WHU-RS19 dataset.

We utilize the AITLAS toolbox for Earth Observation from [1] to train and evaluate our models. Training Split is 60% for training, 20% for validation, and 20% for testing, for both datasets. All models are trained on one NVIDIA Tesla V-100 GPU with 32 GB of memory. The batch size was set to 64 for training. Rectified Adam or RAdam [27] is used as the optimizer. We use learning rate .0001 for WHU-RS19 and .001 for Optimal-31, learning is reduced by factor of 0.1 when validation loss stops improving. Each model is trained for 300 epochs, as we train models from scratch higher iterations were necessary. We use input size of 224x224 by default as "1x" in Table 1 and 2. Inputs are first resized to 256x256 and then center-cropped to 224x224, horizontal and vertical flips are used as data augmentations.

TABLE III. Study of effect of Proposed Ideas

Model	Params. (M)	GFlops	Accuracy (%)	Training Time(s)
Baseline (DenseNet161)	26.51	7.82	86.56	2196
Baseline w. densexLayer	21.21 (-20%)	5.77 (-26%)	84.57 (-2.2%)	2486 (+13%)
Baseline w. e.stem-block	26.59 (+0.3%)	2.17 (-72%)	85.56 (-1.1%)	1911 (-13%)
DenseNetx161	21.21 (-20%)	1.65 (-79%)	85.07 (-1.7%)	2091 (-4.7%)
DenseNetx161 (1.5x)	21.21 (-20%)	3.65 (-53%)	87.06 (+0.6%)	2230 (+1.5%)

V. Experiments

A. Evaluation on WHU-RS19

We use the baseline DenseNet161, ResNet152, ResNet50[28], and the proposed DenseNetx161 in three input settings to evaluate the WHU-RS19 dataset. The experimental results are shown in Table. 1, DenseNet161 contains 26.51m parameters and 7.82 GFlops at input size 224x224. The baseline has much better accuracy than ResNet152 and ResNet50, while ResNet152 has much higher parameters and Flops, but it suffers from overfitting in the ResNet50 architectures. In this case, our DensNetx161 already outperforms the baseline at 1x input by increasing the accuracy by almost 1% while having similar F1, mIoU scores, and training time. At 1.14x input accuracy improves by around 2% and at 2x accuracy is improved by 4.5%, F1 and mIoU also improve by a similar amount. One disadvantage of the larger input size is the training time

TABLE IV. Ablation study of densexLayer Configurations.

conv1	conv2	Params (M)	GFlops	Accuracy (%)	Training Time(s)
1x1	dws 5x5	21.21	5.77	84.57	2486
dws 3x3	dws 5x5	21.94	9.66	84.57	2943
dws 5x5	dws 7x7	22.3	11.87	87.56	3325

TABLE V. Ablation Study on the Efficient Stem Block

Stem Block	Params. (M)	GFlops	Accuracy (%)	Training Time(s)
3x3 3x3	21.21	1.65	85.07	2091
5x5 5x5	21.36	2.17	83.58	2058
7x7 7x7	21.59	2.95	80	2026
3x3 3x3 3x3	21.29	3.68	81.09	2275

TABLE VI. Ablation study on Dilated Convolutions in Stem Block

Conv Kernel	Dilation Rate	Params. (M)	GFlops	Accuracy (%)	Training Time(s)
3x3	1	21.21	1.65	85.07	2091
3x3	2	21.21	1.65	83.58	2072
3x3	3	21.21	1.64	78.1	1971

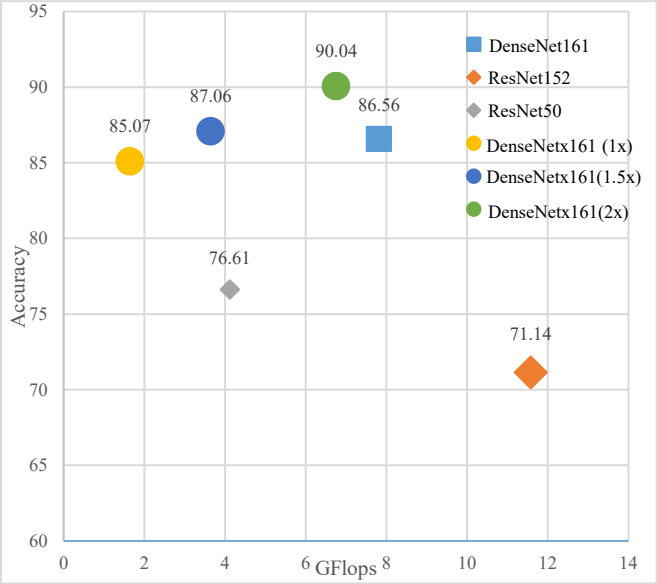

Fig. 2. GFlops VS Accuracy, comparison of the baseline DenseNet161, ResNet152, ResNet50 and proposed DenseNetx161 architectures at 3 input scales.

small data available, ResNet50 the smaller variant performs better. At 1x input, the proposed DenseNetx161 has 20% fewer parameters and 79% fewer GFlops, but it performs very close to the baseline only decreasing the accuracy by 1.7%. The training time is also lower than baseline. At 1.5x input size while still having 53% fewer GFlops our model can already outperform the baseline. And, at 2x input, it improves the accuracy by more than 4% while having 13% fewer GFlops and similar parameters as before, Fig. 2. showcasing the clear superiority of our model while maintaining efficiency. At a higher input size, the training time is increased from the baseline model.

B. Evaluation on Optimal-31

Table.2 shows the evaluation details on the Optimal-31 dataset. In a similar fashion as the previous dataset, DensNet161 the baseline performs better than ResNet152 an increase, which is a much bigger jump in this case than in the previous dataset.

C. Ablation Studies: Impacts of Proposed Ideas

We study in depth the effect our proposed contributions to the baseline bring evaluating the WHU-RS19 dataset. Table.3 shows cases of the effect of the desnexLayer and efficient stem-block in parameters, GFlops, Accuracy, and Training Time. Changing the baseline configuration from regular 3x3 conv. to densexLayer configuration decreases parameters by 20% and flops by 26% from baseline, it also decreases accuracy by 2.2% and adds 13% extra training time. Baseline with only efficient stem block increases parameters by 0.3% as it contains two 3x3 conv. operations, reduces flops by 72%, decreased accuracy by 1.1%, and reduces training time by 13%. With both proposed modifications DenseNetx161 reduces parameters by 20%, flops by 79%, training time by 4.7%, and accuracy by 1.7%. At 1.5x input it outperforms the baseline as stated above.

D. Ablation Studies: densexLayer

able 4. shows the ablation study with three different configurations of densexLayer. First, 1x1 conv. followed by 5x5 depthwise separable, second, 3x3 depthwise separable followed by 5x5 depthwise separable and third, 5x5 depthwise separable followed by 7.x7 depthwise separable convolutions. Increasing the kernel size for dw-separable convolutions increases the parameters, flops, and training time, but the accuracy gain is insignificant.

E. Ablation Studies: efficient Stem-Block

In Table. 5 we show 4 configurations of stem block evaluated on the WHU-RS19 dataset. It is notable, that densexLayer is applied in all of these models. For the last configuration of three 3x3 convolutions, the input size is 2x. But, the clear winner is two 3x3 convolutions with stride 2 to replace the one 7x7 conv. stride 2 of the DenseNet architecture. It comfortably beats the other models in efficiency as well as accuracy.

F. Ablation Studies: dilated convolutions

We also study the dilated convolutions from [29], which can give large receptive fields using smaller kernels. We use dilation rates 2 and 3 for the 3x3 conv. in the efficient stem block and show the comparative study in Table. 6. As it is apparent from the results, dilated convolutions don't show much effectiveness for our proposed model.

VI. Conclusion

Remote scene classification tasks usually have smaller datasets with a high-resolution image, while high-resolution input gives accurate results, it is computationally expensive. To that end, we propose an efficient variation of DenseNets, DenseNetx which can take in a large input size and give accurate results while being computationally efficient. Even without any kind of pretraining it manages to achieve over 90% accuracy in the WHU-RS19 dataset. We detail our evaluation in multiple ablation studies to show the effectiveness of our idea. In the future, we aim to develop a fast-pretraining method to further improve the accuracy while considering more techniques to reduce computation costs as well.

REFERENCES

[1] I. Dimitrovski, I. Kitanovski, D. Kocev, and N. Simidjievski, "Current trends in deep learning for Earth Observation: An open-source benchmark arena for image classification," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 197, pp. 18-35, March 2023.

[2] H. Lee, R. Grosse, R. Ranganath, and A. Ng, "Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks," in Proceedings of the *26th International Conference on Machine Learning, Montreal*, Canada, Jun. 2009.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, Lake Tahoe, NV, USA, Dec. 2012, pp. 1097-1105.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proceedings of the *International Conference on Learning Representations*, San Diego, CA, USA, May 2015.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems 28*, Montreal, Canada, Dec. 2015, pp. 91-99.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, Jun. 2015, pp. 3431-3440.

[7] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, Jul. 2017, pp. 1302-1310.

[8] K. He, R. Girshick, and P. Dollár, "Rethinking ImageNet Pre-training," in Proceedings of the *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, Jun. 2019, pp. 4918-4927.

[9] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely Connected Convolutional Networks," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, Jul. 2017, pp. 4700-4708.

[10] C. Lyu, W. Zhang, H. Huang, Y. Zhou, Y. Wang, Y. Liu, S. Zhang, and K. Chen, "RTMDet: An Empirical Study of Designing Real-Time Object Detectors," *arXiv preprint arXiv:2212.07784,* Dec. 2022

[11] M. Hansen, P. Potapov, R. Moore, M. Hancher, S. A. Turubanova, A. Tyukavina, D. Thau, S. V. Stehman, S. J. Goetz, T. R. Loveland, A. Kommareddy, A. Egorov, L. Chini, C. O. Justice, and J. R. G. Townshend, "High-Resolution Global Maps of 21st-Century Forest Cover Change," Science, vol. 342, no. 6160, pp. 850-853, Nov. 2013.

[12] X.-Y. Tong, G.-S. Xia, Q. Lu, H. Shen, S. Li, S. You, and L. Zhang, "Land-cover classification with high-resolution remote sensing images using transferable deep models," *Remote Sensing of Environment*, vol. 237, Feb. 2020.

[13] S. Fei, M.A. Hassan, Y. Xiao, et al., "UAV-based multi-sensor data fusion and machine learning algorithm for yield prediction in wheat," *Precision Agriculture*, vol. 24, pp. 187-212, Jan. 2023.

[14] S. Dotel, A. Shrestha, A. Bhusal, R. Pathak, A. Shakya, and S. P. Panday, "Disaster Assessment from Satellite Imagery by Analysing Topographical Features Using Deep Learning," in Proceedings of the *2nd International Conference on Image, Video and Signal Processing (IVSP)*, Singapore, Jan. 2020, pp. 86-92.

[15] X. Liu, L. Jiao, L. Li, X. Tang, and Y. Guo, "Deep multi-level fusion network for multi-source image pixel-wise classification," *Knowledge-Based Systems*, vol. 221, pp. 106921, Jun. 2021.

[16] X. Ding, H. Xiong, and X. Liu, "DenseNet-BCNN: DenseNet with Bottleneck Channels and Binary Connect Convolutional Layers," in Proceedings of the *2nd International Conference on Image, Vision and Computing (ICIVC),* Chongqing, China, Jun. 2018, pp. 540-544.

[17] Z. Huang, X. Liu, L. Ma, and H. Zhang, "Parallel DenseNet: A Deep Convolutional Neural Network with Parallel Connections," in Proceedings of the *IEEE International Conference on Multimedia and Expo (ICME)*, San Diego, CA, Jul. 2018, pp. 1-6.

[18] X. Liu, L. Jiao, L. Li, X. Tang, and Y. Guo, "DenseNet-SI: DenseNet with Scale Invariant Convolutional Layers," in Proceedings of the *IEEE International Conference on Image Processing (ICIP)*, Anchorage, AK, Sep. 2021, pp. 1577-1581.

[19] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual Attention Network for Scene Segmentation," in Proceedings of the *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA*, Jun. 2019, pp. 3146-3154.

[20] Y. Lee, J. -w. Hwang, S. Lee, Y. Bae and J. Park, "An Energy and GPU-Computation Efficient Backbone Network for Real-Time Object Detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Long Beach, CA, USA, 2019, pp. 752-760.

[21] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1251-1258.

[22] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, pp. 1387-1396.

[23] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 6848-6856.

[24] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 6105-6114.

[25] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," *arXiv preprint* arXiv:1602.07360 [cs.CV], 2016.

[26] H. Cai, L. Zhu, and S. Han, "ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware," in Proceedings of the *International Conference on Learning Representations*, New Orleans, LA, USA, 2019.

[27] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the Variance of the Adaptive Learning Rate and Beyond," in *International Conference on Learning Representations*, 2020.

[28] He, K., Zhang, X., Ren, S., & Sun, J. (2016). "Deep Residual Learning for Image Recognition." In Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778).

[29] Fisher, Matthew, et al. "Multi-scale context aggregation by dilated convolutions." arXiv preprint arXiv:1511.07122 (2015)