Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 07, July-2020

Extraction of Road Network from Satellite images using Efficient Net

A Deep learning Approach

Nilkamal More, Rishabh Lalla, Rahil Memon,
Palak Shah, Bhagyashree Sawant
Department of Information Technology
K.J. Somaiya College of Engineering
Mumbai, India

V. B. Nikam
Department of Computer Engineering
Veermata Jijabai Technological Institute
Mumbai, India

***Abstract*—Satellite images are great examples of high-resolution images that represent a wide array of information about the surface of the earth. Analyzing such high dimensional images and gaining insights from them is a crucial task and this analysis can provide solution to various kinds of problems. But analyzing and gaining insights from such high-resolution images is an difficult task and requires a lot of computation. The problem that we have considered for this study is identification and extraction of the road networks from the given satellite images using neural network. As observed in the past, the main aim of a supervised learning research is usually higher accuracy and not efficiency. But in this paper, we try to find a solution to this problem that is not only has high accurate but also is efficient in terms of time and space complexity. The reason for this study is to come up with solutions that can be deployed on low compute devices such as edge devices especially during inference time. This problem comes under pixel-wise semantic segmentation and we have done a comparative study on few models that are able to provide potential solutions for the problem.**

Keywords—Road network, Semantic Segmentation, convolution, RESNET18, EfficientNet, Benchmarking

I. INTRODUCTION

Satellite images are able to capture high resolution images of the surface of the earth which creates a diverse set of data which can be useful for various kinds of analysis. High resolution satellite data can be made available for any particular remote region on earth. It can be used in a wide array of useful applications ranging from military surveillance, architecture and building purpose, agricultural purpose and disaster management.

The type of analysis to be performed on these satellite images depends on the application. One great example which we will be focusing on in this paper is road network extraction. Identification of road networks from satellite images is a problem wherein the satellite images are broken down in clear visualization of important information which we need at the moment. Identification of road networks from satellite images is a difficult task and one way to approach this problem is for a human to manually identify and annotate the image but this is a tedious task and not feasible in a long run. A better way is to make someone annotate a few images and train a machine to learn from that data and make appropriate predictions.

Initially, we explain our details of our dataset such as the types of images, their dimensions and the task to be performed using this dataset. The we review some of the existing methodologies for our problem. Then we propose a potential architecture and try out different variants of those architectures and record the results. Then in the end, we derive the conclusions from the observations of our experiment.

II. DATASET

The dataset that we used is taken from DeepGlobe [1, 2] road network extraction challenge. It contains 4484 images for training, 50% of which include satellite images and the rest are their corresponding output masks. Each image has the resolution of 1024 x 1024 pixels and has 3 Color channels i.e. the image is in RGB color format. Each pixel of the image represents 50 cm per pixel of ground resolution. These satellite images as seen in Fig. 1 are captured over Indonesia, Thailand and India. These images are sampled, labelled and segmented by GIS experts. Out of the total 4484 images, we took 4000 images for training and the remaining 484 images were used for validation which is roughly 10% of the training data and a lot of images are provided as test set whose ground truth is not available.

III. LITERATURE SURVEY

In this literature survey, we review few of the top performing architectures in the DeepGlobe challenge. Although these architectures were trained on 4484 images as training data and validated on separate 1243 images, the ground truth of the validation set was not available for our research, so we made use of the training set and sampled a validation set from the 4484 images. The results produced by our research are different from the results achieved in the challenge and so we have produced our own baseline as a comparison metric.

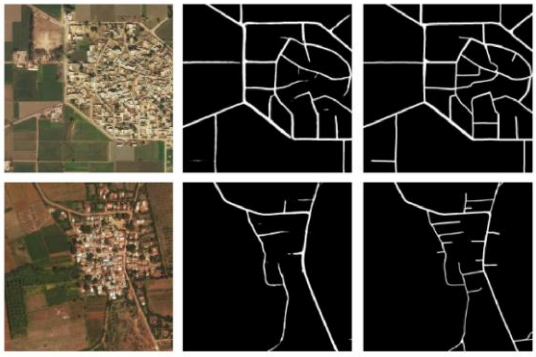

Fig. 1. Road network annotated from satellite images.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 07, July-2020

In this paper [3], The authors have suggested that from a satellite picture, it is troublesome and computationally costly to extract streets because of quality of other road like highlights with straight edges. They have suggested a methodology for automatic road extraction which will be based on a fully CNN of the U-net family. This network comprises of ResNet-34 pre-trained on ImageNet and decoder adjusted from vanilla U-Net. The best open score of their model on the open leaderboard is 0.64.

In this paper [4], stacked U-nets are utilized with various outputs for street network which needs to be extracted. The problem of uneven classes of training data is solved using a hybrid loss function. Post-processing techniques which includes street map vectorization and shortest path search with progressive thresholds which basically helps in improving recall. The overall improvement of mean IoU which when compared to the vanilla VGG network is more than 20%.

In this paper [5], the authors proposed a Linknet architecture that makes use of dilated convolution as its main component. This was the winning solution of the challenge and had an validation and test IoU are 0.6466 and 0.6342 respectively.

IV. PROPOSED SYSTEM

As seen in the Fig. 2, there are 2 aspects to be considered while selection of architecture for semantic segmentation:

- Model: This is essentially the main scaffolding of our architecture. The models used for our problem are U-net [6], LinkNet [7] and Feature Pyramid Network(FPN) [8].
- Backbone: This can be thought of the actual building blocks of our architecture. This actually tells the number of Layers (Convolutions, Pooling etc.) and their flow. For our study, we have used 2 backbones: Resnet18 [9] and Efficientnet b0 [10].

A. *Algorithm*

The table 1 below is a generic format for our algorithm and all our architectures follow this algorithm. This algorithm is divided into two parts the training phase and the test phase.

TABLE I. ALGORITHM

Algorithm
Input:- Satellite Image of dimension 1024x1024x3 Output:- Binary Image of size 1024x1024x1
Training
1. Start. 2. Preprocess the input image and output 3. Create an Encoder – Decoder Architecture. 4. Initialize the Hyperparameters appropriately. 5. Pass the input output pairs in batches to the architecture for training. 6. Measure the performance metrics across each epoch. 7. End.
Testing
1. Start. 2. Preprocess the input image of dimension 3. Pass the input image to the trained neural network architecture. 4. The neural network generates an output of dimension 1024x1024x1 with values between 0 to 255. 5. Apply appropriate thresholding using following condition; 6. If Pixel_Value <=128 then Pixel_Value = 0 else 255 7. This generates a Binary image which will be our ouput. 8. End.

B. *Models*

As seen in the Fig. 2, the model is the frame of the neural network that dictates the flow of the architecture, which depends on the task to be solved. In our case it is an encoder decoder architecture for performing semantic segmentation.

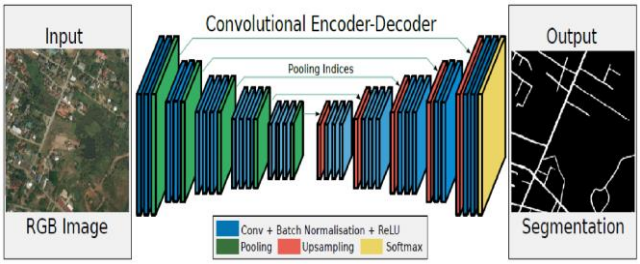

Fig. 2. Proposed System Architecture

1) U-Net: U-net as seen in Fig. 3, is one of the most common architectures for Semantic Segmentation. The U-net model was developed not only as a means to identify or classify the image also to find or localize whether the particular area of interest lies. The way it works is by classifying each pixel on the image and makes sure that the output image has the same size as the input image. The explanation for its name is pretty much evident from the architecture seen in Figure. This architecture can be broken down into 2 parts viz. the contraction path or simply the encoder which essentially captures the Context of the image and the expansion path or commonly known as the decoder which is responsible to convert the acquired context into an output image of appropriate dimensions. The encoder can simply be thought of as a simple CNN consisting of multiple convolutions and pooling layers. The decoder on the hand does contain a convolution layer but does not contain any pooling layers as its goal is to up-sampling the image and pooling layers are usually used to down-samples the image. For up-sampling the image, instead of using some predefined method to up-sample the image, we use transpose convolution where we can learn the weights similar to convolution layer but with the intention of up-sampling the image. This architecture is essentially fully convolutions network which means that it does not contain any dense layers thereby enabling us to use any image of any dimension.

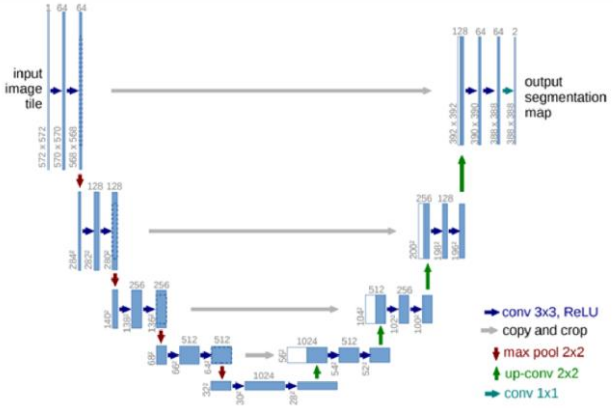

Fig. 3. U-NET architecture. Multi-channel feature maps are represented using blue box. The top of the box corresponds to the number of channels. The Lower left edge of the box represents the x-y-size. Different operations are denoted by the arrows.

2) LinkNet: LinkNet as seen in Fig. 4 is a light weight, fast architecture used for semantic segmentation. This model is

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 07, July-2020

also an encoder-decoder architecture. They are quite similar to ladder networks which contain lateral connection to the decoder blocks. Also, the decoder has fairly a smaller number of parameters due to channel reduction and the decoder output is simply added with the corresponding encoder block in an element wise fashion. The image is passed from a series a decoder which breaks down the image into fundamental patterns which we need to extract and the decoder builds the image back up.

3) Feature Pyramid Network: FPN is shown in Fig. 5, as seen with other models for semantic segmentation has 2 paths, the Bottom-up pathway and the Top-down pathway. The Bottom-up pathway is nothing but a Feed Forward computation of the Backbone CNN, whereas Top Down pathway is responsible for generating the feature maps from low resolution to high resolution. In Top Down pathway, each layer is up-sampled by a factor of 2 and it uses a method known as nearest neighbor. But it also receives a lateral connection from corresponding layer for the Bottom Up pathway on which an 1x1 convolution is applied in order to eliminate the channels dimension, after which a simple element wise addition is done to generate the next layer. Finally, a 3x3 convolution is applied to each feature map and then the output and then these are up-sample according to their respective dimension. The results from all feature maps is up sampled and concatenated and then down sampled to get the final output.

C. Backbones

As seen in the Fig. 2, the backbones are the building blocks of the neural network. These are the actual layers of the network such as convolution, pooling, upsampling etc.

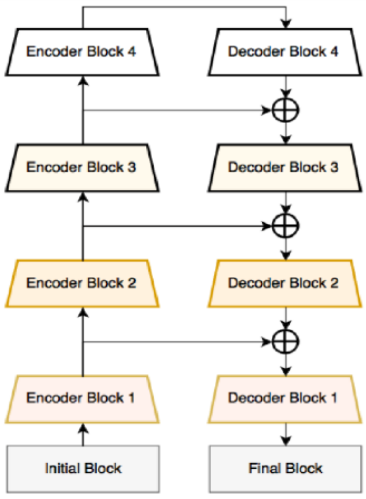

Fig. 4. LinkNet architecture

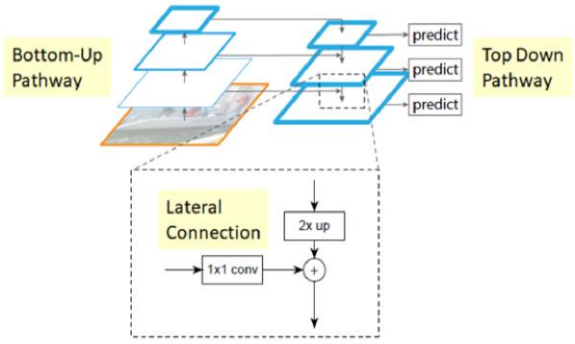

Fig. 5. Feature Pyramid Network architecture with the representation of how the top down pathway and lateral connection are merged by addition

1) ResNet18: As the size and the resolution of images keeps growing with time, there is a need for ways to train more deeper neural networks. But as we keep adding more and more layers, newer problems arise such as vanishing gradients and degradation problem. The degradation problem basically means that the more layers we keep adding, the accuracy will start to saturate and might degrade after a certain number of layers. Resnet basically works on the idea of skip connection as we can see in the given Fig. 6, the left figure shows normal convolution layers and right figure specifies that the input is passed on to the output of the convolution layer. Resnet allows the flow of important information from the initial layers to the later layers ensuring that the later layers performs just as good as the earlier layers on the image. Resnet18 is used as a backbone for various models Faster RCNN, FCN and U-net.

2) Efficient Net: The usual approach of choosing a neural network architecture suitable for the problem is to select some tried and tested architecture for a similar kind of problem as done in case of transfer learning. But if there is a need for creation of custom architecture, then the approach is to vary the components of architecture such as the number of neurons in each layer, the number of layers etc. EfficientNet as seen in Fig.7 proposes a sound approach in scaling the width, depth and resolution in a systematic way which is known as compound scaling. There are 8 variants of Efficientnet ranging from efficientnetb0 to efficientnetb7 wherein efficientnetb0 has least number of parameters and efficientnetb7 has most number of parameters. For our study we have used efficientnetb0 as a backbone.

Fig. 6. Resnet skip connection

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 07, July-2020

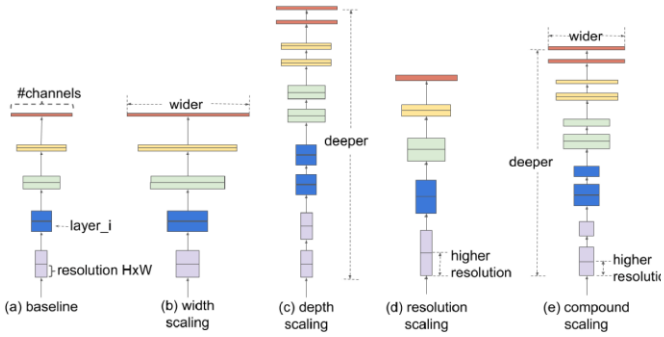

Fig. 7. EfficientNet (a) is a baseline model; (b)-(d) are conventional scaling that only increases one dimension of Model width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

V. RESULT

So, as discussed earlier, we used 3 models with two backbones and ran each of them for 20 epochs. We measured the metrics such as training and validation loss, IOU, F1 score, F2 score, Precision, Recall as well as the trainable and non-trainable Parameters for each of the models. All of these models where run on Google Colab environment. We used Adam as optimizer with a learning rate of 0.0001. Also, all these backbones were initialized with their Imagenet [11] weights to provide a fair starting point in training. The Loss used for training the networks is a combination of Binary Cross-Entropy and Jaccard Loss.

First observation as shown in the Table 2, is the training and validation loss. For both training and validation loss, all the Resnet variants minimized the loss more than in Efficientnet counterpart in 20 epochs.

Second observation as shown in the Table 1, is the training and validation loss. This being a Binary segmentation problem, the IoU score achieved are quite high for both training and validation set. Again the Resnet variants perform slightly better than their Efficientnet counterpart.

Third observation as shown in the Table 1, are the F1, F2, precision and Recall scores. Here again the Resnet variants perform slightly better than their Efficientnet counterparts.

Fourth observation as shown in the Table 1, is the number of trainable parameter for each architecture. It is observed that the Efficientnet variants have less amount of trainable parameters than Resnet. For the U-net architecture, Resnet variant has approximately 42% more trainable parameters than Efficientnet. For the Linknet architecture, Resnet variant has approximately 90% more trainable parameters than Efficientnet. For the Efficientnet architecture, Resnet variant has approximately 97% more trainable parameters than Efficientnet.

Fifth observation as shown in the Table 1, is the total number of parameter (Trainable and Non trainable) for each architecture. For the U-net architecture, Resnet variant has approximately 41% more parameters than Efficientnet. For the Linknet architecture, Resnet variant has approximately 88% more trainable parameters than Efficientnet. For the FPN architecture, Resnet variant has approximately 96% more trainable parameters than Efficientnet.

TABLE II. . OUTPUT METRIC

	U-net		**Linknet**		**FPN**	
Metric	***Resnet18***	***Efficientnetb0***	***Resnet18***	***Efficientnetb0***	***Resnet18***	***Efficientnetb0***
Training Loss	0.046164	0.054573	0.047052	0.0691	0.047205	0.063413
Validation Loss	0.970963	0.845167	0.932245	0.942718	0.82965	1.100368
Training IOU	0.981443	0.978753	0.981145	0.974034	0.981094	0.975841
Validation IOU	0.966089	0.96052	0.965751	0.95883	0.82965	0.959376
Training F1 score	0.990562	0.989168	0.990407	0.986696	0.99038	0.987645
Validation F1 score	0.982502	0.979536	0.982329	0.978602	0.982696	0.978907
Training F2 score	0.990585	0.989258	0.990393	0.986946	0.990407	0.987832
Validation F2 score	0.986827	0.987535	0.987026	0.988122	0.98665	0.986744
Training Precision	0.990527	0.989018	0.99043	0.986292	0.99034	0.987337
Validation Precision	0.975536	0.966847	0.974756	0.963606	0.976309	0.966534
Training Recall	0.990599	0.989321	0.990382	0.987116	0.990424	0.987957
Validation Recall	0.989777	0.993046	0.990227	0.994708	0.98934	0.992158
Trainable Parameters	14,330,644	10,071,501	11,511,668	6,049,549	13,805,124	6,994,173
Non- Trainable Parameters	9,926	44,000	10,022	46,784	10,246	44,320
Total Parameters	14,340,570	10,115,501	11,521,690	6,096,333	13,815,370	7,038,493

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 07, July-2020

VI. CONCLUSION

In this paper, we tried out various encoder decoder approaches of semantic segmentation for our Road Network Extraction dataset. The observations from previous sections help us in deriving the conclusions as seen in Fig. 8:

First Conclusion is that Resnet architecture outperforms all its Efficientnet counterparts by a small margin in the similar settings that is each of them is executed for 20 epochs using Adam as an optimizer with the same learning rates and using their respective ImageNet weights for transfer learning. So in terms of accuracy, Resnet is the idle choice.

Second Conclusion is that Resnet architecture is highly overparameterized than Efficientnet counterparts. Resnet variants have anywhere between 42% to 98% more parameters than Efficientnet variants which depends on the model.

Thus, it can be concluded that Efficientnet outperforms Resnet by a major margin in terms of both time and space complexity making it idle for computation with limited amount of resources.

As we have seen in our dataset, satellite images can have large dimensions not only in terms of the length and breadth, but also in terms of channels (although our images had 3 channels, satellite images can have many more channels), making it difficult to train them on large models. So, it is idle to use architectures such as Efficientnet, that not only give comparable results to architectures like Resnet, but also provide a better space and time complexity, at both training and inference time.

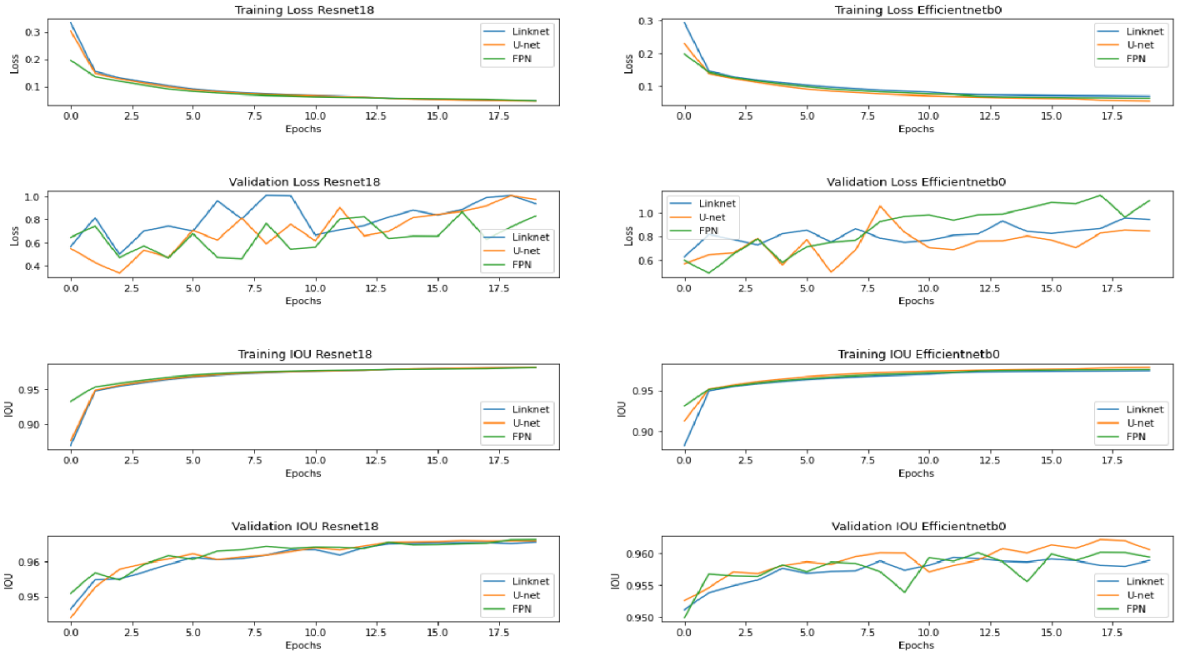

Fig. 8. IoU and Loss plots for training and validation data

REFERENCES

[1] *CVPR18 - Home*. DEEPGLOBE. http://deepglobe.org/.

[2] Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., … Raskar, R. (2018). DeepGlobe 2018: A Challenge to Parse the Earth through Satellite Images. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. https://doi.org/10.1109/cvprw.2018.00031

[3] Buslaev, A., Seferbekov, S., Iglovikov, V., & Shvets, A. (2018). Fully Convolutional Network for Automatic Road Extraction from Satellite Imagery. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. https://doi.org/10.1109/cvprw.2018.00035

[4] Sun, T., Chen, Z., Yang, W., & Wang, Y. (2018). Stacked U-Nets with Multi-output for Road Extraction. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. https://doi.org/10.1109/cvprw.2018.00033

[5] Zhou, L., Zhang, C., & Wu, M. (2018). D-LinkNet: LinkNet with Pretrained Encoder and Dilated Convolution for High Resolution Satellite Imagery Road Extraction. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. https://doi.org/10.1109/cvprw.2018.00034

[6] Ronneberger, O. (2017). Invited Talk: U-Net Convolutional Networks for Biomedical Image Segmentation. *Informatik Aktuell Bildverarbeitung Für Die Medizin 2017*, 3–3. https://doi.org/10.1007/978-3-662-54345-0_3

[7] Chaurasia, A., & Culurciello, E. (2017). LinkNet: Exploiting encoder representations for efficient semantic segmentation. *2017 IEEE Visual Communications and Image Processing (VCIP)*. https://doi.org/10.1109/vcip.2017.8305148

[8] Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2017.106

[9] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2016.90

[10] Tan, M., & Le, Q.V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *ArXiv, abs/1905.11946*.

[11] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., … Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[12] More, N., Nikam, V. B., & Banerjee, B. (2020). Machine learning on high performance computing for urban greenspace change detection: satellite image data fusion approach. *International Journal of Image and Data Fusion*, 1–15. https://doi.org/10.1080/19479832.2020.1749142