



Investigating Transfer Learning Performances of Deep Learning Models for Classification of GPR B-Scan Images

Mehmet Dikmen

Dept. of Computer Engineering, Başkent University, Ankara 06790, Turkey

Corresponding Author Email: mdikmen@baskent.edu.tr

<https://doi.org/10.18280/ts.390534>

ABSTRACT

Received: 17 June 2022

Accepted: 26 September 2022

Keywords:

ground penetrating radar, image classification, deep learning, transfer learning

Recent advances in deep learning models have made them the state-of-the-art method for image classification. Due to this success, they have been applied to many areas, such as satellite image processing, medical image interpretation, video processing, etc. Recently, deep learning models have been utilized for processing Ground Penetrating Radar (GPR) data as well. However, studies generally focus on building new Convolutional Neural Network (CNN) models instead of utilizing baseline ones. This paper investigates the usefulness of existing baseline CNN models for classifying GPR B-scan images and aims to determine how well pre-trained models perform. To that end, a real bridge deck GPR data, DECKGPRHv1.0 dataset was used to evaluate the transfer learning performances of various CNN models. Different variants of the models in terms of varying depths and number of parameters were also considered and evaluated in a comparative manner. Although it is an older model, ResNet achieved the best results with 0.998 accuracy. The experimental results showed that there is generally a direct correlation between the simplicity of the model and its success. Overall, it is concluded that near perfect results are possible by just adapting pre-trained models to the problem without fine-tuning.

1. INTRODUCTION

Ground Penetrating Radar (GPR) is a fast and efficient way to sense information about subsurface without drilling or digging. The visualization mode of a GPR signal can be done in 1D, 2D or 3D, which are named as A-scan, B-scan, and C-scan, respectively. Buried objects are typically represented as hyperbolas in GPR B-scan images. However, due to the challenging nature of processing such data and its difficult interpretation, automated approaches were needed and developed over time [1]. Although, the number of studies using conventional machine learning techniques were decreased in the past 3-4 years, Support Vector Machines [2], Artificial Neural Networks (ANN) [3, 4], boosting algorithms [5, 6], Hidden Markov Models [7] were utilized for detection and classification tasks on GPR images. With the improvement in computation power of GPUs, deep learning techniques which proved their superiority on image classification started to replace them.

Recently, deep learning techniques have been successfully studied for processing GPR images, including land mine classification [8, 9], classification of soil types [10], subsurface target detection and classification [11-14], object size prediction [15], recognition of tunnel lining elements [16], rebar detection [17, 18], detection of moisture damages [19] and recognition of subgrade defects [20, 21]. All these studies applied deep learning techniques in one of the two ways. The first way is to build a user-defined Convolutional Neural Network (CNN) for the solution of the given problem. Many researchers preferred this way for their specific studies [9, 10, 14, 15, 20, 22, 23]. However, this strategy requires lots of effort and time for deciding an optimal network structure and

its optimal hyper parameters. Therefore, some researchers preferred to utilize existing CNN architectures as an alternative and 2nd way. In this strategy, researchers are required to make a choice of using an existing model with or without its pre-trained weights. Using pre-trained weights obtained from ImageNet is referred as transfer learning. An alternative way is ignoring those weights and train the model from scratch. While the former makes it possible to get faster results, usually higher performance is achieved with the latter if a large dataset is provided. Therefore, seeking a good enough performance with the least effort, which is possible when existing CNN models are used with transfer learning, forms the motivation of this study.

In this paper, a comparative analysis is performed to investigate various existing CNN models using transfer learning. Their classification performances are evaluated on a real bridge deck GPR dataset containing B-scan images. The rest of the paper is organized as follows: Section 2 presents the literature review of the previous studies that utilized existing pre-trained models. In Section 3, a general information about CNNs is given. Section 4 describes the general workflow of the experiments done. In Section 5, information about the dataset, experiments, and the discussion are presented. Finally, Section 6 contains the conclusion.

2. THE LITERATURE REVIEW

The studies that utilize existing CNN models can be categorized in two: studies that directly use existing models or use them as a backbone model for Region Based CNN (R-CNN) or similar frameworks. Studies in the first category

utilized AlexNet [12, 13, 18] and ResNet-50 [19] directly. Kim et al. utilized both B-scan and C-scan images to classify underground objects with AlexNet model as cavity, pipe, manhole and subsoils background [12]. Their study showed that it is difficult to classify underground objects using only B-scan images and including C-scan images can yield better results. The authors also classified hyperbolas on cropped B-scan images in a different study [13]. Although, the overall classification accuracy of the trained AlexNet model was 96.2%, the authors stated that the model fails for the images where the shape of the hyperbolas are not visually distinguishable. Another study using AlexNet dealt with rebar detection [18]. The classification accuracy of their model ranged from 70% to 90%, depending on the window sizes. The final study in this category successfully utilized ResNet-50 model to extract image features, and then use them in a YOLO v2 model for moisture damage detection [19]. The highest recall and precision achieved in experiments were 94.53% and 91%, respectively.

In the second category, where existing CNN models were used as a backbone model for frameworks, ResNeXt-101 [11], ResNet-101 [16] and Vgg16 [17, 21] were successfully applied. Hou et al. incorporated ResNeXt-101 into a Mask Scoring R-CNN (MS R-CNN) architecture to improve the object signature detection performance in GPR scans [11]. Although, the average precision and recall were generally below 50%, the authors showed that their architecture outperforms various versions of R-CNN framework. In a study where a similar MS R-CNN framework is built to recognize tunnel lining elements from GPR images, ResNet-101 model is used for feature extraction [16]. Their architecture achieved recognition accuracies of 96.02%, 91.17%, and 95.45% in a field GPR survey experiment with three targets. In one of the studies that applied Vgg16 model, a Single Shot Multibox Detector (SSD) model was constructed for rebar detection and localization [17]. The performance of the proposed model was compared with a Faster R-CNN model and proved to be slightly better. Xu et. al also chose Vgg16 as the base network for their Improved Faster R-CNN framework for automatic identification of railway subgrade defects [21]. The authors compared their results with a traditional SVM+HOG method and the baseline Faster R-CNN. The proposed model outperformed them with a mean Average Precision (mAP) of 83.6%.

All these past studies above chose an existing model and stick with it during their experimentation. However, there are many existing CNN models with different architectures and surely, the success of those studies would have been different if another CNN model had been used. Therefore, the performance of different models for processing GPR images needs to be investigated. So far, it has been observed that there is just one study which compares performances of different CNN architectures on analyzing GPR B-scan images [24]. However, the number of CNN models used in that study is rather few; only AlexNet, VGG-16, GoogleNet (i.e., Inception), ResNet-50 and SqueezeNet were considered. Their transfer learning performances were evaluated on both simulated and real GPR images. Material type classification performances was found poor, since at most 56.78% accuracy has been achieved with ResNet50. Shape and soil type performances were notably better, again ResNet50 was the best with accuracies of 95.32% and 98.27%, respectively. Shape type classification accuracy of the models in real data couldn't achieve better than 66.67% accuracy, including the

new model presented by the authors which obtained 74.07% accuracy. In addition to the lack of sufficient number of existing models in that study, all existing models considered in this study were rather old. The newest model among them was ResNet50 which was presented in 2015. As new CNN models are introduced every day that outperform their predecessors in the ImageNet competition, their performances are needed to be assessed to guide researchers who are aiming to improve deep learning solutions that analyze GPR images. This is the main contribution of this paper; investigating the existing CNN models, identify their capabilities for the given problem and determine the best one. Additionally, to our knowledge, this paper is the only study that compares different variants of a CNN model. Therefore, as another contribution, the effect of model depth and parameters are also investigated and the possible reasons of performance differences between these variants are discussed.

3. CONVOLUTIONAL NEURAL NETWORKS

CNN is a type of deep learning model specifically built for analyzing images. The most notable difference from a conventional feed forward ANN is that CNNs initially include a series of layers which are responsible for feature extraction. It is the design of this part that makes a CNN model superior to others. Typically, it contains some convolutional layers where a set of filters are convolved with the input to obtain feature maps, and some pooling layers for down sampling these maps to reduce the computation. Generally, the output obtained from this part is first flattened and then fed into several fully connected (FC) layers usually designed as a multi-layer Perceptron (MLP) structure. An illustration of such an architecture is given in Figure 1.

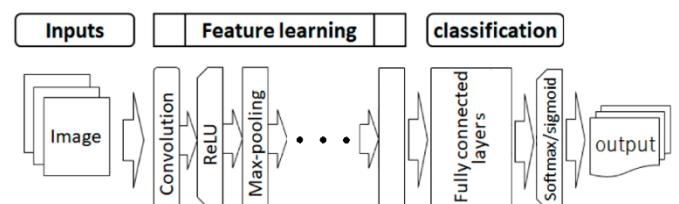


Figure 1. General structure of a CNN

Most CNN configurations use a dropout layer where some neurons are temporarily removed to help overcome the overfitting problem. Another important parameter is the activation function which is responsible for forwarding the amount of information through layers. The most common ones are ReLU, Softmax, tanH and the Sigmoid. Finally, loss functions are used to calculate the predicted error which is optimized in training stage. Cross-Entropy is the most common one, used in classification problems. On the other hand, Euclidean loss function is generally preferred in regression tasks.

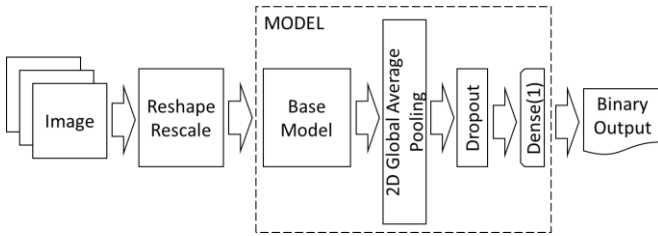
Table 1 shows the CNN models used in this study present in Keras API which are publicly available with pre-trained weights [25]. As can be seen from Table 1, the publication year of the CNN models investigated in this study varies from 2015 to 2021. The top-1 accuracy refers to their performance on ImageNet validation dataset. The term depth corresponds to the topological depth of the network, including activation layers, batch normalization layers, etc. The M in number of parameters corresponds to millions.

Table 1. Pretrained CNN models on ImageNet

Model	Size (MB)	Top-1 Accuracy	Parameters	Depth	Publish year
Vgg16	528	71.3%	138.4M	16	2015
ResNet50	98	74.9%	25.6M	107	2015
ResNet101	171	76.4%	44.7M	209	2015
InceptionV3	92	77.9%	23.9M	189	2016
Xception	88	79.0%	22.9M	81	2017
MobileNet	16	70.4%	4.3M	55	2017
MobileNetV2	14	71.3%	3.5M	105	2017
DenseNet121	33	75.0%	8.1M	242	2017
DenseNet201	80	77.3%	20.2M	402	2017
NASNetMobile	23	74.4%	5.3M	389	2018
EfficientNetB0	29	77.1%	5.3M	132	2019
EfficientNetB4	75	82.9%	19.5M	258	2019
EfficientNetV2B0	29	78.7%	7.2M	-	2021
EfficientNetV2S	88	83.9%	21.6M	-	2021

4. METHODOLOGY

Figure 2 shows the workflow of the experiments. The input image is the GPR B-scan image which is a grayscale 2D image. Since the CNN models considered in this study (Table 1) requires a 3-channel input, this 2D image is first converted to an RGB image by giving the original 2D image to all channels. In addition, CNN models don't necessarily have the same input size and pixel value ranges (i.e., scales). Therefore, for each model, input images are reshaped using Bilinear Interpolation to match the input size required for that model, and then pixel values are rescaled accordingly.

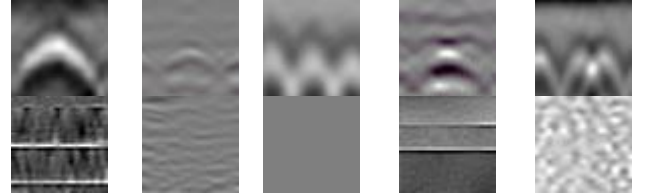
**Figure 2.** The workflow of the experiments

In this study, transfer learning was applied to use those models given in Table 1. The idea of this technique is keeping the layers that are responsible for feature learning (Figure 1) and then replace the last layers with new ones depending on the problem [26]. These new layers are fixed for each base model chosen. This is because, they determine the output of the problem. In this study, a 2D global average pooling, dropout and a dense layer with a single unit are inserted to replace the removed layers (Figure 2). Dropout rate was chosen as 0.2. The activation function in Dense layer is selected as Sigmoid, since the number of classes in the dataset used in experiments is 2 (i.e., positive and negative samples). In this architecture represented in Figure 2, weights between the feature learning layers of the base model are directly taken and not changed during the training step. Only weights of the newly inserted layers are updated in training. This layout is constructed for each model given in Table 1.

5. RESULTS AND DISCUSSION

In experiments, the DECKGPRHv1.0 dataset [27] was used

that contains 17,260 GPR B-scan images which are cropped from real bridge deck GPR field data. There are 2 reasons for choosing this dataset. First, it contains cropped images so that a preprocessing is not needed to identify all hyperbolas from a single scan and crop them to feed the proposed model. Possible inaccuracies in this procedure may lead to false alarms in the final classification, thus, these are avoided as well. Second, it is a reliable dataset since it is already used in a past journal study [28]. The dataset contains 8,664 positive and 8,596 negative samples with sizes 50x50 and 48x48, respectively. Some samples from the dataset are given in Figure 3.

**Figure 3.** Sample positive (top) and negative (bottom) images

To evaluate the generalization ability of the models 10-fold stratified cross validation was used. The reason of implementing cross-validation using stratified sampling is to maintain the ratio of positive to negative samples in each fold. All models were trained with Adam optimizer where learning rate was set to 10^{-3} and the loss function was chosen as Binary Cross Entropy. The training was performed until the loss value converges with a maximum of 100 epochs. To avoid overfitting, early stopping technique was used with a tolerance of 10 epochs. Therefore, if validation loss fails to improve after 10 epochs, the training stops. Among all experiments, the maximum number of epochs reached with this configuration was 41. On the other hand, the shortest training lasted for 17 epochs. A checkpoint mechanism was also included in this configuration. Therefore, when training ends due to early stopping, best weights of the model were loaded back.

Model performances are evaluated by 3 metrics which use the primitives of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN):

- Accuracy: The ratio of the number of correctly classified samples to all samples calculated by Eq. (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

- Recall: The ratio of the number of correctly classified positive samples to all positive samples calculated by Eq. (2).

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

- Precision: The ratio of the number of correctly classified positive samples to all samples which are predicted as positive, calculated by Eq. (3).

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

The results for 10-fold stratified cross validation is given in Table 2. The values are the average of the folds. All experiments were done in Google Collab. Since the GPU assigned for the run varies in each session and model runs were

resumed from checkpoints when internet connection fails; a comparison between training times would not be fair, thus, not given. Therefore, models that are too heavy on computation are not considered in this study. However, a general opinion can be formed by examining the inference times of their training on ImageNet. A summary of such information can be found in Keras API where their inference times for both CPU and GPU are presented [25].

Table 2 presents some interesting results. Overall, all models except NasNetMobile achieved 94% or better accuracy even without using data augmentation or training from scratch. The performance of NasNetMobile is not a disappointment though since the classification accuracy is almost 89% with nearly approximately recall and precision. Surprisingly, the best results were obtained by the oldest models ResNet50 and ResNet101 which are introduced in 2015. On the other hand, two variants of the newest model EfficientNetV2 (2021) in Table 1 performed almost identical results (i.e., less than 0.01 accuracy) compared to ResNet variants, thus, missed the first place narrowly. Despite their slightly worse performance, it should be noted that EffcientNet models are the smallest in terms of size except the mobile CNN models (Table 1), and they are usually trained faster while still achieving good results [29]. Comparing its variants, the first versions of EfficientNet (B0 and B4) performed slightly worse compared to V2 versions, as excepted. The next model in top ranking was another old model, Vgg16 (2015), that slightly outperformed EfficientNetB4, however, it fell behind EfficientNetB0.

Table 2. Results of 10-fold stratified cross validation

Model	Loss	Accuracy	Recall	Precision
DenseNet121	0.080	0.977	0.981	0.975
DenseNet201	0.084	0.973	0.978	0.968
EfficientNetB0	0.032	0.991	0.994	0.989
EfficientNetB4	0.050	0.984	0.987	0.982
EfficientNetV2B0	0.017	0.996	0.997	0.995
EfficientNetV2S	0.024	0.992	0.993	0.990
InceptionV3	0.396	0.941	0.946	0.940
MobileNet	0.038	0.989	0.993	0.984
MobileNetV2	0.175	0.943	0.965	0.924
NasNetMobile	0.294	0.889	0.901	0.887
ResNet50	0.009	0.998	0.998	0.998
ResNet101	0.008	0.998	0.999	0.997
Vgg16	0.047	0.986	0.989	0.983
Xception	0.144	0.952	0.953	0.956

Another surprising result is that the smallest model (i.e., MobileNet) in terms of both depth and number of parameters in Table 1 was able to outclass popular models like DenseNet, Xception and Inception. Considering this together with ResNet’s superiority, it can be concluded that smaller models generally perform better when compared to larger models. This result might not be so surprising after visually examining the GPR B-scan images. Recall that all models in Table 1 were previously trained on RGB images of 1000 categories and their weights, which are obtained after this heavy training, are used to classify GPR images. Some examples of those images are presented in Figure 4 which are taken from CIFAR-100 dataset to give an idea how complicated patterns these RGB images have.

As seen in Figure 3, GPR images have less textural complexity than the images which these models are pre-trained on (Figure 4). Examining the textures in GPR images in Figure 3, it can be derived that those images have circular patterns which are either hyperbola, line, or noise like small

granules. The background is generally homogenous or slightly speckled. On the other hand, as seen in Figure 4, RGB images corresponding to real world objects often have much detail due to their variety. Their backgrounds might also include some level of detail if they do not correspond to smooth regions like sky and road. Therefore, instinctively, there is no need to extract complex features to classify GPR B-scan images as opposed to RGB images which contain numerous objects and backgrounds, both usually have detailed textures. Since deeper models with more parameters tend to extract more details compared to smaller models, it is expected that they perform better. However, this is not the case for GPR images, thus, simpler models also have a potential to achieve good results.

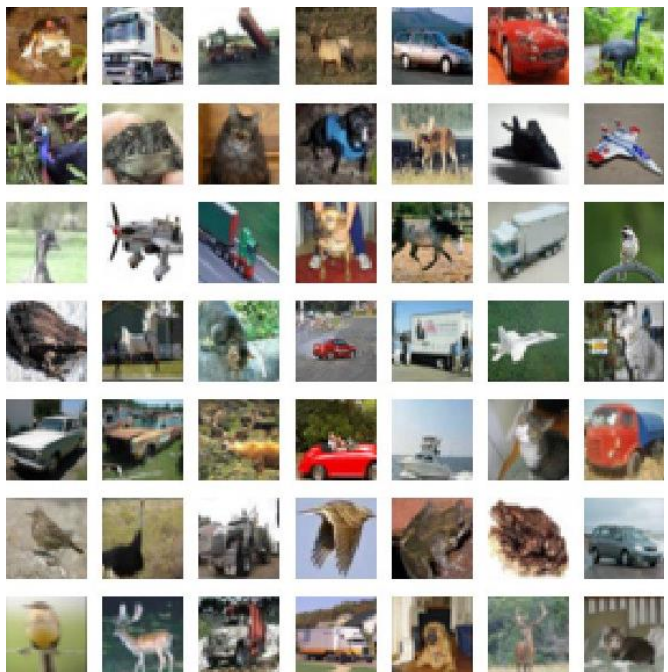


Figure 4. Sample RGB images from CIFAR-100 dataset

In addition to the textural detail, another factor might be the input image size (50x50) which is rather small compared to the input sizes of CNN models that generally have a few hundreds of pixels in height and weight. For instance, ResNet requires a 3-channel input with each having a size of 224x224. Since the level of detail gradually degrades when image size is decreased, number of features to be extracted will also be limited when compared to images with higher resolutions (i.e., sizes). For this reason, smaller models might have gotten the upper hand.

Comparing CNN models with their variants, a similar deduction to the one mentioned about image complexity level can be made. Inspecting the results given in Table 2 together with model sizes (number of parameters and the depth) shown in Table 1, it is seen that many models with the smaller variant outperformed the bigger one, even though the performance difference is rather small. The only exception is ResNet where the larger ResNet101 model achieved almost equal results. The biggest performance difference is observed between MobileNetV2 and MobileNet in which the former obtained more than 4% accuracy. Although, MobileNetV2 has somewhat less parameters compared to MobileNet, it is twice deeper. This also explains the poor performance of NasNetMobile. Although, the number of parameters is not much more than the MobileNet models, NasNetMobile is more than 6 times deeper than MobileNet and more than 3

times deeper than MobileNetV2. Deeper models offer more non-linearity which might be too much for simple cases. Hence, that should have affected those model's classification ability adversely.

6. CONCLUSIONS

In this paper, an investigation was made to determine the usefulness of pre-trained deep learning architectures for classifying GPR B-scan images. Using transfer learning, a total of 14 different CNN models were considered with varying depths and number of parameters, including new models like EfficientNet. The best result was obtained by ResNet variants that explains why it is still popular in many deep learning applications for image analysis. Still, it should be noted that the accuracy difference between ResNet and EfficientNetV2B0 model was just 0.002 and it is a lighter model than ResNet. For this reason, EfficientNet might be more practical for use. On the other hand, 6 of these models achieved better than 0.98 classification accuracy. Therefore, it can be concluded that it is possible to have almost perfect results in GPR image classification problem using existing models with pre-trained models.

Another conclusion derived in this study is that smaller models can compete with even newer models which have more layers and parameters, although the general performance of these smaller models on ImageNet is worse. It has been suspected that their success lies on the textural complexity of input images which are generally less detailed when compared to colored photographs of real-world objects. Therefore, analyzing GPR B-scan images with deep learning models do not necessarily require a deeper and larger model which generally provides better results compared to smaller models on other image processing tasks. Comparing variants of the same model also supports this conclusion since, a variant with more layers and/or parameters generally didn't improve the classification performance.

Using deep learning models for the solution of problems is generally costly since they require too much time and effort to setup and train. Therefore, it is believed that findings of this paper can also guide the researchers who aims to create new CNN models for analyzing GPR images when considering the architecture of their model. As a future work, releases of new CNN models should always be examined and compared with the earlier models. In addition, it will be enlightening to use different datasets and make similar studies on other problems and that require processing GPR images.

REFERENCES

- [1] Travassos, X.L., Avila, S.L., Ida, N. (2021). Artificial neural networks and machine learning techniques applied to ground penetrating radar: A review. *Applied Computing and Informatics*, 17(2): 296-308. <https://doi.org/10.1016/j.aci.2018.10.001>
- [2] El-Mahallawy, M.S., Hashim, M. (2013). Material classification of underground utilities from GPR images using DCT-based SVM approach. *IEEE Geoscience and Remote Sensing Letters*, 10(6): 1542-1546. <https://doi.org/10.1109/LGRS.2013.2261796>
- [3] Harkat, H., Ruano, A.E., Ruano, M.G., Bennani, S.D. (2019). GPR target detection using a neural network classifier designed by a multi-objective genetic algorithm. *Applied Soft Computing*, 79: 310-325. <https://doi.org/10.1016/j.asoc.2019.03.030>
- [4] Zhang, Y., Huston, D., Xia, T. (2016). Underground object characterization based on neural networks for ground penetrating radar data. *Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, and Civil Infrastructure*, SPIE, 9804, pp. 10-18. <https://doi.org/10.1117/12.2219345>
- [5] Sakaguchi, R., Morton, K.D., Collins, L.M., Torrione P.A. (2017). A comparison of feature representations for explosive threat detection in ground penetrating radar data. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12): 6736-6745. <https://doi.org/10.1109/TGRS.2017.2732226>
- [6] Aggarwal, B., Acharya, A., Laghari A. (2019). An efficient multi-stage adaptive boosting approach for landmine detection in ground penetrating radar data. *Robotic Technologies for NDT*. <https://doi.org/10.13140/RG.2.2.35813.22246>
- [7] Yuksel S.E., Gader P.D. (2016). Context-based classification via mixture of hidden Markov model experts with applications in landmine detection. *IET Computer Vision*, 10(8): 873-883. <https://doi.org/10.1049/iet-cvi.2016.0138>
- [8] Kafedziski, V., Pecov, S., Tanevski, D. (2018). Detection and classification of land mines from ground penetrating radar data using faster R-CNN. *26th Telecommunications Forum (TELFOR)*, pp. 1-4. <https://doi.org/10.1109/TELFOR.2018.8612117>
- [9] Lameri, S., Lombardi, F., Bestagini, P., Lualdi, M., Tubaro, S. (2017). Landmine detection from GPR data using convolutional neural networks. *25th European Signal Processing Conference (EUSIPCO)*, pp. 508-512. <https://doi.org/10.23919/EUSIPCO.2017.8081259>
- [10] Barkataki, N., Mazumdar, S., Singha, P.B.D., Kumari, J., Tiru, B., Sarma, U. (2021). Classification of soil types from GPR B scans using deep learning techniques. *International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, pp. 840-844. <https://doi.org/10.1109/RTEICT52294.2021.9573702>
- [11] Hou, F., Lei, W., Li, S., Xi, J. (2021). Deep learning-based subsurface target detection from GPR scans. *IEEE Sensors Journal*, 21(6): 8161-8171. <https://doi.org/10.1109/JSEN.2021.3050262>
- [12] Kim, N., Kim, S., An, Y.K., Lee, J.J. (2021). A novel 3D GPR image arrangement for deep learning-based underground object classification. *International Journal of Pavement Engineering*, 22(6): 740-751. <https://doi.org/10.1080/10298436.2019.1645846>
- [13] Kim, N., Kim, K., An, Y.K., Lee, H.J., Lee, J.J. (2020). Deep learning-based underground object detection for urban road pavement. *International Journal of Pavement Engineering*, 21(13): 1638-1650. <http://dx.doi.org/10.1080/10298436.2018.1559317>
- [14] Ishitsuka, K., Iso, S., Onishi, K., Matsuoka, T. (2018). Object detection in ground-penetrating radar images using a deep convolutional neural network and image set preparation by migration. *International Journal of Geophysics*, 2018: 1-8. <https://doi.org/10.1155/2018/9365184>
- [15] Barkataki, N., Tiru, B., Sarma, U. (2022). A CNN model for predicting size of buried objects from GPR B-Scans.

- Journal of Applied Geophysics, 200: 104620. <https://doi.org/10.1016/j.jappgeo.2022.104620>
- [16] Qin, H., Zhang, D., Tang, Y., Wang, Y. (2021). Automatic recognition of tunnel lining elements from GPR images using deep convolutional networks with data augmentation. *Automation in Construction*, 130: 103830. <https://doi.org/10.1016/j.autcon.2021.103830>
- [17] Liu, H., Lin, C., Cui, J., Fan, L., Xie, X., Spencer, B.F. (2020). Detection and localization of rebar in concrete by deep learning using ground penetrating radar. *Automation in Construction*, 118: 103279. <https://doi.org/10.1016/j.autcon.2020.103279>
- [18] Xiang, Z., Rashidi, A., Ou, G. (2019). An improved convolutional neural network system for automatically detecting rebar in GPR data. *ASCE International Conference on Computing in Civil Engineering 2019*. <https://doi.org/10.1061/9780784482438.054>
- [19] Zhang, J., Yang, X., Li, W., Zhang, S., Jia, Y. (2020). Automatic detection of moisture damages in asphalt pavements from GPR data with deep CNN and IRS method. *Automation in Construction*, 113: 103119. <https://doi.org/10.1016/j.autcon.2020.103119>
- [20] Tong, Z., Gao, J., Zhang, H. (2018). Innovative method for recognizing subgrade defects based on a convolutional neural network. *Construction and Building Materials*, 169: 69-82. <https://doi.org/10.1016/j.conbuildmat.2018.02.081>
- [21] Xu, X., Lei, Y., Yang, F. (2018). Railway subgrade defect automatic recognition method based on improved faster R-CNN. *Scientific Programming*, 2018. <https://doi.org/10.1155/2018/4832972>
- [22] Barkataki, N., Tiru, B., Sarma, U. (2022). A CNN model for predicting size of buried objects from GPR B-Scans. *Journal of Applied Geophysics*, 200: 104620. <https://doi.org/10.1016/j.jappgeo.2022.104620>
- [23] Özkaya, U., Öztürk, Ş., Melgani, F., Seyfi, L. (2021). Residual CNN+ Bi-LSTM model to analyze GPR B scan images. *Automation in Construction*, 123: 103525. <https://doi.org/10.1016/j.autcon.2020.103525>
- [24] Ozkaya, U., Melgani, F., Bejiga, M.B., Seyfi, L., Donelli, M. (2020). GPR B scan image analysis with deep learning methods. *Measurement*, 165: 107770. <https://doi.org/10.1016/j.measurement.2020.107770>
- [25] Keras Applications. <https://keras.io/api/applications/>, accessed on June 10, 2022.
- [26] Nogay, H.S. (2021). Comparative experimental investigation of deep convolutional neural networks for latent fingerprint pattern classification. *Traitement du Signal*, 38(5): 1319-1326. <https://doi.org/10.18280/ts.380506>
- [27] Asadi, P., Gindy, M. (2019). DECKGPRHv1.0 dataset. <https://github.com/PouriaAI/GPR-Detection/>, accessed on April 14, 2022.
- [28] Asadi, P., Gindy, M., Alvarez, M. (2019). A machine learning based approach for automatic rebar detection and quantification of deterioration in concrete bridge deck ground penetrating radar B-scan images. *KSCE Journal of Civil Engineering*, 23(6): 2618-2627. <https://doi.org/10.1007/s12205-019-2012-z>
- [29] Tan, M., Le, Q. (2021). Efficientnetv2: Smaller models and faster training. *International Conference on Machine Learning*. PMLR, pp. 10096-10106.