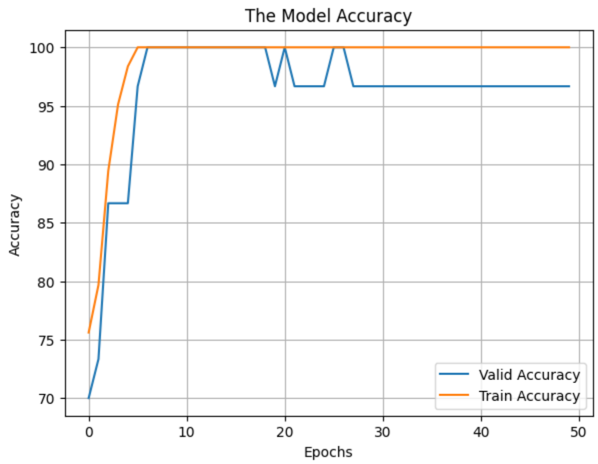
The original dataset contains 152 images (45 field and 107 road images) in different conditions, which means that there are noise datasets with different resolutions and image shapes, and the most important thing is the imbalance between the field and road training images. This may result in the high percentage of field misclassification.

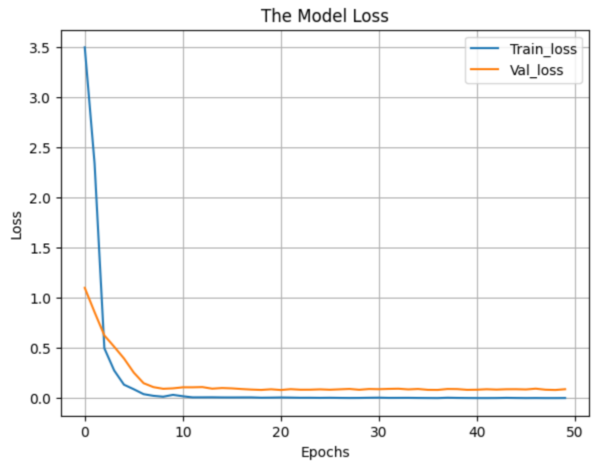
The initializing model comprises of 2 (Conv2d, Batch Normalization and Maxpool layer) for feature extraction and 2 fully connected layer for classifier. The original set (comprising of training and validation sets) and test set are used from the ones sent by Trimble, while arbitrary set (5 – road images and 5-field images) is collected, which is utilized for measuring the generalization of the model.

The input is the tensor (16,3,128,128) where 16: batch size, 3: channels, (128,128): the height, width of the input image, respectively. The output is the tensor (16,1) for binary classification (road-field classification)

Loss function is Binary Cross Entropy between the target and the input probabilities

Optimizer is Adam with learning rate at $1e-4$

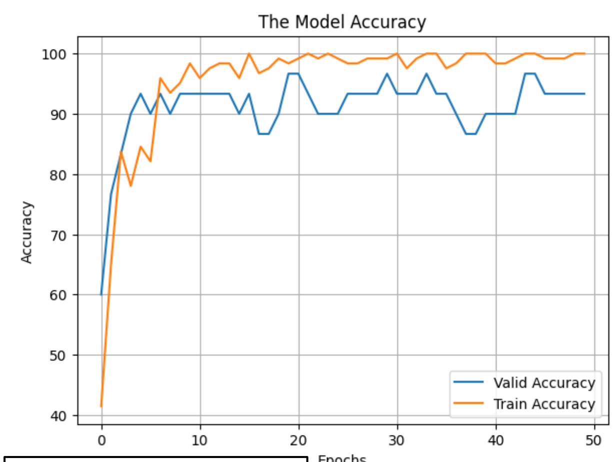




Validation Accuracy: 96.67 % and Train_loss: 0.0018

Model selection is suitable for field-road classification, despite the limit of training data, but the classifier shows that it is good and efficient and it achieves 80% accuracy in the test set. Although it is met the requirements, it is unreliable for the generalization. Therefore, increasing the number of training data is needed to see how it's response.

The next step is using data augmentation to increase the training dataset.



Text Accuracy: 80%

	ID	predictedVal	originalLabel	predictedLabel
0	2.jpeg	1.00	1	1
1	3.jpeg	1.00	1	1
2	5.jpeg	1.00	1	1
3	6.jpeg	0.99	1	1
4	7.jpeg	0.65	1	1
5	8.jpeg	1.00	1	1
6	1.jpeg	0.00	0	0
7	10.jpeg	0.87	0	1
8	4.jpeg	0.99	0	1
9	9.jpeg	0.00	0	0

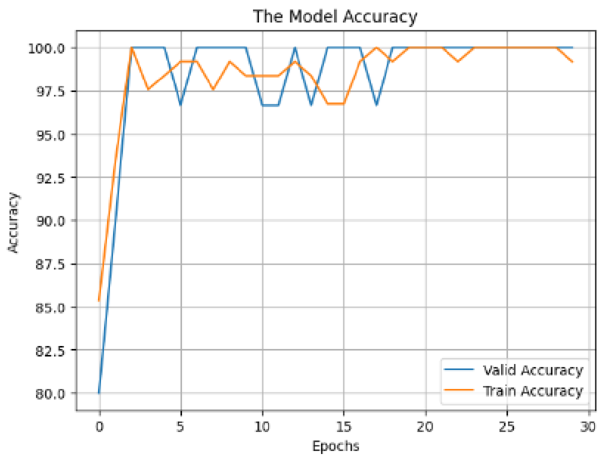
Arb Accuracy: 60%

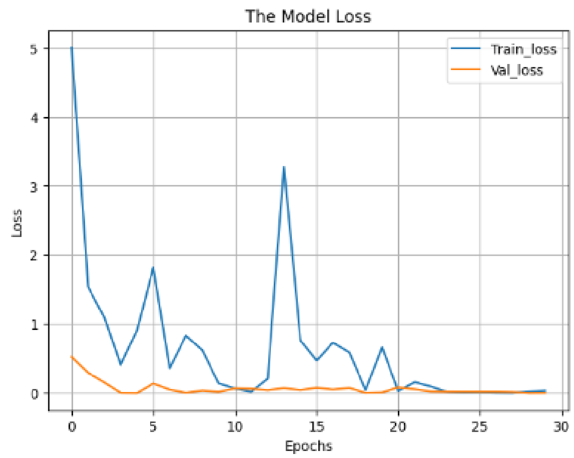
	ID	predictedVal	originalLabel	predictedLabel
0	5.jfif	1.00	1	1
1	10.jpg	1.00	1	1
2	4.jfif	1.00	1	1
3	3.jfif	0.99	1	1
4	2.jfif	1.00	1	1
5	1.jpg	0.99	0	1
6	8.jfif	0.99	0	1
7	7.jfif	0.00	0	0
8	9.jpg	0.98	0	1
9	6.jpg	1.00	0	1

Validation Accuracy: 96.67% - Train_loss: 0.14375

When increasing the number of training images, the validation accuracy decreases a little, but it's acceptable because the generalization is our concern. In text accuracy result, it's showed that all roads (label - 1) are well classified, while in the field class, there are just 2/4 samples that are recognized. In fact, when doing the same tasks in the arbitrary data, the field class is more falsely classified than the road. To deal with this issue, it is necessary to increase the amount of field data. However, according to the paper "WINTER ROAD SURFACE CONDITION RECOGNITION USING A PRE-TRAINED DEEP CONVOLUTIONAL NEURAL NETWORK ", using the pretrained model is much more efficient and time-saving. The specified information is included in the paper summary.

Using model VGG16 as the pretrained model with freezing all parameter in feature extraction networks except the last layer in vgg16 as well as replacing the classifier by 2 fully connected networks.





Validation Accuracy: 100 % - Train_loss: 1e-5

Text Accuracy: 80%

	ID	predictedVal	originalLabel	predictedLabel
0	2.jpeg	0.01	1	0
1	3.jpeg	1.00	1	1
2	5.jpeg	1.00	1	1
3	6.jpeg	0.03	1	0
4	7.jpeg	1.00	1	1
5	8.jpeg	1.00	1	1
6	1.jpeg	0.00	0	0
7	10.jpeg	0.01	0	0
8	4.jpeg	0.00	0	0
9	9.jpeg	0.00	0	0

Arb Accuracy: 80%

	ID	predictedVal	originalLabel	predictedLabel
0	1.jpg	1.00	1	1
1	2.jfif	1.00	1	1
2	3.jfif	1.00	1	1
3	4.jfif	1.00	1	1
4	5.jfif	0.00	1	0
5	6.jpg	1.00	0	1
6	7.jfif	0.00	0	0
7	8.jfif	0.04	0	0
8	9.jpg	0.00	0	0
9	10.jpg	0.01	0	0

The results are shown outstanding, although the accuracies are nearly the name as the previous model, but it is highlighted that the accuracy in the arbitrary data is highly improved and the generalization is satisfied with our expectation (1/10 false prediction for each type of class). In addition, tuning hyper-parameters is necessary for classification problem. However, based on the empirical verification, it showed that this tuning doesn't improve the model much, and thus, the learning rate 0.0001 is kept for the all validations.