CHANGE DETECTION USING A HIERARCHICAL BASED SIAMESE NETWORK IN MULTISPECTRAL REMOTE SENSING IMAGES

Akash Kumar Singh (2023AIY7582)

*Kshitij Kumar Verma (2023CEZ8116) helped in creating report.

ABSTRACT

The study presents a hierarchical based Siamese network architecture approach for Change Detection (CD) from a pair of co-registered remote sensing images along with their corresponding ground truth data. Alternatively, the labels of 10% changed pixels were flipped in the training dataset and performance matrices are evaluated for these noisy labels.

Index Terms— Change Detection, Siamese Network

1. INTRODUCTION

Change Detection (CD) aims to detect relevant changes from a pair of co-registered images acquired at distinct times. The definition of change may usually vary depending on the Application such as urban, vegetation or in the case of damages caused due to disasters. A better CD model is the one that can recognize these relevant changes while avoiding complex irrelevant changes caused by seasonal variations, building shadows, atmospheric variations, and changes in illumination conditions. In this report a variation of Siamese based network is used to leverage it's use for accurate change detection.

2. STUDY AREA AND METHODOLOGY

The training dataset consists of multispectral images over an agricultural area in their first timestamp and it's change to urban/semi-urban areas in the second changed timestamp. The ground truth labels are provided for these changed timestamps.

2.1. Training and Test Dataset

For training we have randomly selected 90 of the images and remaining 10% for the validation set. Each of the training images is 512*512*3 dimensions with each corresponding label of 512*512*1. Later the training set is divided into training and validation with the ratio of 90:10.

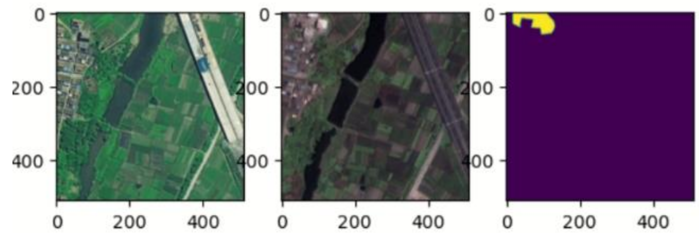

Figure 1: Train Set

The test dataset consists of multispectral images of the same 512*512*3 as the training dataset of agricultural domain.

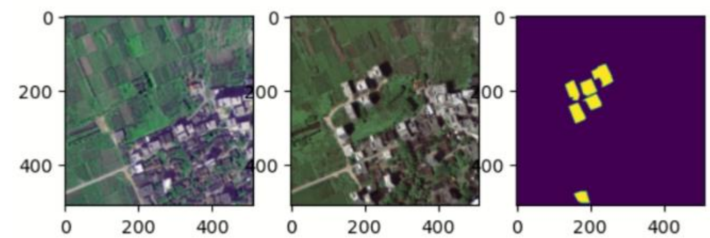

Figure 2: Test Set

2.2. Model

The weights of two pre-trained models namely Resnet34, Seresnet34 are used as initial weights to use as a hierarchical feature extractor. On top of this, we have put our custom layers which outputs change detection label masks. The model is represented in figure 3. A proposed up-sampling approach is used as shown.

2.3. Loss Functions and Metrics

The two classes represented in the study are 0 for unchanged and 255 for changed pixels. We later converted this into 0 and 1. Since it's a binary classification problem Sigmoid function with the classifier is used. Sum of the two losses is Dice loss and Binary focal loss with class weightage (0.4,0.6). For the performance metric while training we took Overall Accuracy (OA) and weighted F1 score, whereas for test set OA, F1 Sensitivity True Positive Rate (TPR) and True Negative Rate (TNR) is evaluated.

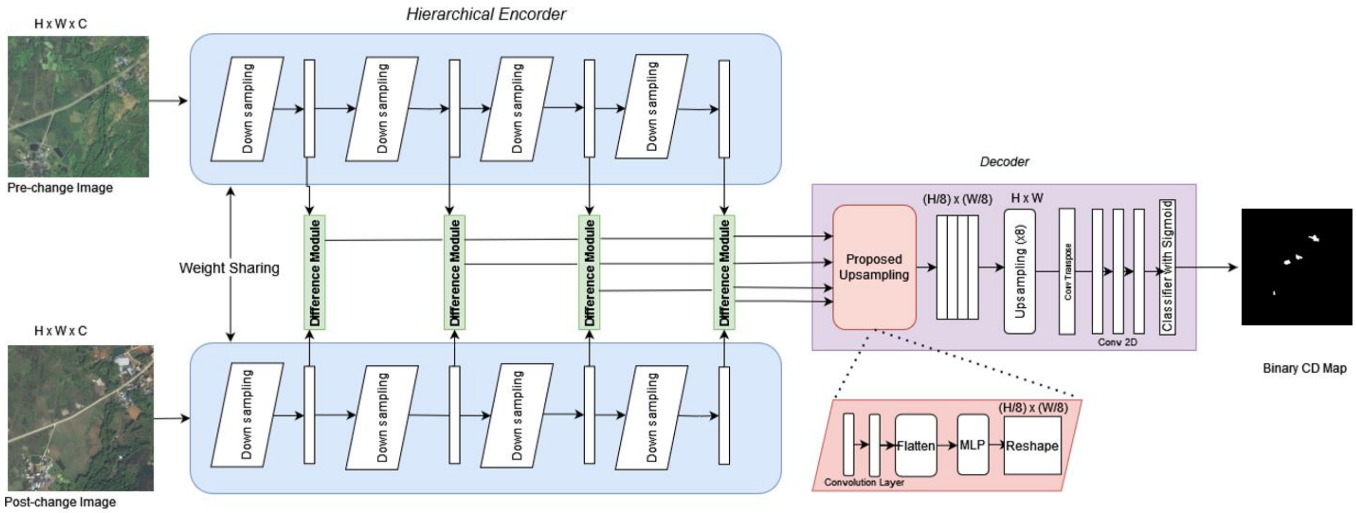

Figure 3: Hierarchical based Siamese Network used in this study

2.4. Training Parameters

Adam optimizer with learning rate 0.001 is used in the study. Each model is trained for 300 Epochs. A batch size of 8 is used throughout the process and the best model is saved.

2.5. Generating Change Detection Maps

For test dataset with known labels, we computed the various performance metrices. For the test dataset without labels, the model successfully generated the Change Detection maps and saved them in the folder 'testLabelNotProvided'. For creating the predicted label as 0 or 1, we took 0.5 as threshold on the output of sigmoid.

3. RESULTS & DISCUSSIONS

The following results are obtained for training the model on the test dataset.

	Resnet34	Seresnet34
OA (%)	91.89	92.12
F1	0.028	0.03
Sensitivity (TPR)	0.0169	0.0176
Sensitivity (TNR)	0.9859	0.9883

The Training and Validation curve using Resnet34 and SeResnet34 are as follows:

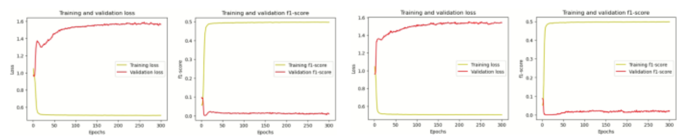

Figure 4: Training and Validation Resnet34 and SeResnet34

3.1. Effect of flipping the labels of training dataset on the performance of the model

To check the robustness of the this, change detection method, we randomly flipped the 10% of changed labels of the training dataset changing 1 to 0 (e.g. 6005 1's to 5405 1's for a random image) and retrained the model which gives us the following results:

	Resnet34	Seresnet34
OA (%)	89.77	91.23
F1	0.0598	0.0582
Sensitivity (TPR)	0.047	0.0391
Sensitivity (TNR)	0.961	0.9772

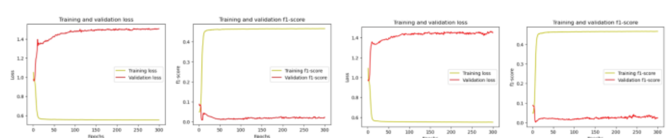

Figure 5: Training and Validation Resnet34 and SeResnet34 for labels flipped dataset

The overall accuracy of the model slightly decreases with the noisy labels which shows the robustness of the method.