# Automatic Change Detection of Physical Model Structures using Deep Convolutional Neural Networks

**Theodoros Karozis**

Snr. 2020843

Anr. 990505

Master Thesis DSBG

Thesis committee:

Dr. Henry Brighton

Dr. Drew Hendrickson

Tilburg University
Tilburg, the Netherlands

# Preface

This thesis has been written as the partial fulfillment of the requirements for the Master track Data Science: Business and Governance and completes my education at Tilburg University. During my studies at Tilburg University, I met very interesting people who kept me motivated in order to finish my studies and fulfill the presented thesis.

First of all, I would like to thank my supervisor Dr. Henry Brighton who with his guidance and recommendations helped me to deliver the current thesis. His guidelines were up to the point to cast every problem which came up during the thesis process.

Furthermore, I would like to thank Deltares who contributed with their dataset to deliver the thesis. Especially, I would like to thank my Deltares' supervisor Joost den Bieman and Menno de Ridder who they supervised my project with great care and for having pushed me to go always a step further.

Moreover, I would like to thank Dr. Drew Hendrickson for evaluating my thesis in its final stage and for the time spent in assessing my work and making it possible for me to graduate.

Finally, the thesis would not be possible without the friends that I made here and the continuous support of my family. I would like to say to all of you and especially to Smaragda and Spyros thank you for doing my life here in Tilburg so enjoyable.

Theodoros Karozis,

Tilburg, December 2018

# Abstract


This thesis addresses the problem of detecting changes in physical model structures used in near-shore hydro-engineering. This thesis implemented in collaboration with Deltares, an institute for applied research in the field of water. Deltares provided us the dataset from their physical model testing. Physical models help coastal engineers to understand the complex hydrodynamic processes happening in the nearshore zone and provide solid and affordable engineering design solutions. Physical model testing is essential because, after the evaluation of these models, it will be decided either to deploy them or redesign.

Until now, the damage is assessed manually by detecting the amount of stones which moved, a technique which called scanning. The assessment of these structures is essential and requires a lot of effort in order to achieve the best outcome. Consequently, it is very time consuming and the reproducibility of the visual inspection over the structure is the only way to minimize the errors. As a result, the creation of an automated tool to detect the changes over the physical models is more than necessary.

Up to now the automatic change detection over the structures is done with photogrammetric methods and digital image processing techniques. Therefore, the main goal of this thesis is to answer the question:

**Research Question 1:** *To what extent can Deep Convolutional Neural Network methods detect changes in physical model structures?*

The physical models consist of small rocks and armour units, which are very small objects and the goal of this thesis is to detect their movement. This goal leads to the following research question:

**Research Question 2:** *To what extent can a Deep Convolutional Encoder-Decoder model detect movements of small objects such as rocks?*

We created a Deep CNN VGG16, a Deep CNN VGG19 and a Deep Convolutional Encoder-Decoder model which we compare them to a baseline model. The Deep CNN VGG16 and the Deep CNN VGG19 are pre-trained models and we use their weights from ImageNet competition to achieve the feature extraction from the images. The baseline model is the PCA k-means approach which used for change detection in remote sensing images. We conducted the experiments by applying the models to 134 pairs of images. The images depict the physical model in the initial state and after a continuous wave impact. When Deltares' coastal engineers captured these images there existed different amount of water into the physical models. As a result, there exist a lot of noise in images due to the combination of flash lighting and different water level. The Deep CNN VGG19 method had the highest F1 score compared to the other methods. However, the models did not perform very well in terms of Precision, because they could not detect and eliminate the noise. Furthermore, the Deep CNN Encoder-Decoder approach with pre-trained weights after training in an artificial dataset could predict the unchanged class with average F1 score equal to 98.44%. Nevertheless, it was not able to give meaningful predictions for the original dataset of physical models.

**Keywords:** Image Change Detection, Physical Model Structureσ, PCA k-means, Deep CNN VGG architectures, Deep CNN Encoder-Decoder

# Contents

# Chapter 1: Introduction

This chapter has been divided into four parts. The first part gives a general introduction to physical model testing and gives some information about Deltares institution. The second part formulates the main research problem. The third part provides the research questions. The fourth section explains briefly the main findings.

## 1.1 Physical Model Testing and Deltares

The stability and the viability of coastal structures are vital for human lives and properties. For this reason, coastal engineers build physical models of these structures and test them in order to do the final evaluation of the possible structure. Thus, the importance of these tests and experiments are high, because, after the evaluation of these models, it will be decided to build them or lead to the redesign of the structure.

Physical models help coastal engineers to understand the complex hydrodynamic processes occurring in the nearshore zone and they provide solid and affordable engineering design solutions. These laboratory experiments are carried out on structures built from rocks, stones and armour units and they are physical models of breakwaters, harbor moles and dike revetments. Physical models created in order to have an estimate of the structure's behavior in actual simulations. These experiments could help coastal engineers to have an estimate of their structure in the real conditions and are necessary when the structure deviates a lot from previous designs (Wolters, van Gent, Allsop, Hamm, & Muhlestein, 2010).

Deltares laboratory is suitable for physical modeling testing and many tests happen there. Deltares is a research institution which specialized in the field of water. The main focus of their research is on deltas, coastal regions, and river basins. The areas of their expertise are flood risk, adaptive delta planning, infrastructure, water & subsurface resources and environment. The physical model tests are taking place and examine many different tasks and problems. Thus, Deltares Hydro facilities are suitable to conduct large-scale tests over extreme conditions. The Hydro facilities include simulations of Atlantic Basin, Pacific Basin, Scheldt Flume, Delta Basin, Intake and Outfall Basin, Delta Flume and Water and Soil Flume.

## 1.2 Project Formulation

Up to now, coastal engineers are facing the issue of damage assessment of their physical models. Damage can be defined as the movement of armour units or rocks after the waves impact (de Almeida Sousa, Van Gent, & Hofland, 2018). Also, the damage can be characterized by the percentage of the displaced units. Later studies proposed some parameters to describe damage as the proportion of removed units from the slope. More specific, parameter S is used and is equal to the eroded area divided by the nominal diameter of the stones. According to the value of this parameter, there are different descriptions in the definition of damage. We have concepts of damage initiation, intermediate damage, and failure. Thus, the assessment of the breakwater structure is very challenging and requires a lot of attention to detail.

Until now, the damage is assessed manually by detecting the amount of stones which moved, a technique which is called scanning. The assessment of these structures is essential and requires a lot of effort in order to achieve the best outcome. The evaluation depends on the accuracy, hence it requires a lot of time and precision. In addition, it is prone to mistakes due to the human component (Hoskere, Narazaki, Hoang, & Spencer Jr, 2018). As a result, the need for an automated tool, able to detect automatically the units and rocks displacements, is essential.

Firstly, it can speed up the process and save a lot of time for coastal engineers, since the counting of all the rock movements is vastly time-consuming. Secondly, the design of such a tool can reduce the errors which are caused by humans. As we mentioned before, the human factor plays a key role to the assessment and the possibility of errors is high; for instance, in case that two different persons conduct the same evaluation of the same structure, then different results are a common scenario. Consequently, the reproducibility of the visual inspection is the only way to have better results, something which is time-consuming. Therefore, the design of the tool is revolutionary and necessary; it will lead to time minimization, reduction of human errors and also the reproducibility of those assessments.

## 1.3 Research Questions

Previous studies used photogrammetric methods (Lemos, Loja, Rodrigues, & Rodrigues, 2016) and digital images processing techniques (Courela, Carvalho, Lemos, Fortes, & Leandro, 2015). For this reason, we will try to implement machine learning techniques to cast the problem on identifying automatically the rocks shifts and movements. The methods which we tried are based on a Deep Convolutional Neural Network model for image feature extraction. Specifically, we created a deep CNN VGG16 and a deep CNN VGG19 method. These models are pre-trained networks which trained with images from the ImageNet dataset (Russakovsky et al., 2015). Furthermore, a Deep Convolutional Encoder-Decoder model with pre-trained weights is proposed. We compare these methods with PCA k-means clustering approach which used for change detection problems on remote sensing images (Celik, 2009). So our first research question is the following:

**Research Question 1:** *To what extent can Deep Convolutional Neural Network methods detect changes in physical model structures?*

Furthermore, we created and trained the Deep Convolutional Encoder-Decoder method in order to determine if it is able to identify any possible rock movement. In order to achieve this we have to answer the following research question:

**Research Question 2:** *To what extent can a Deep Convolutional Encoder-Decoder model detect movements of small objects such as rocks?*

## 1.4 Main Findings

The major objective of this study was to investigate the change detection problem in physical model structures. Part of the aim of this project is to develop methods able to detect these changes. It should be noted that we used cutting-edge techniques to carry out the problem of change detection in physical model structures. So far, Deltares has been used manual visually inspection techniques to detect any possible changes in physical models. Also as it was aforementioned, previous studies in this field suggest photogrammetric methods and digital images processing techniques. As a result, we proposed novel methods in this field using deep learning techniques to cast the problem of change detection.

First of all, the available dataset consists of 134 pairs of images, which depict the physical model structure before and after a wave impact. When Deltares' coastal engineers captured these images there existed different amount of water in the physical models. As a result, there exist a lot of noise in images due to the combination of flash lighting and different water level in physical models. Afterwards, we implemented the methods on the available dataset and the evaluation metrics shown that the highest average F1 score, Precision and Recall value was for the Deep CNN VGG19 model in comparison to VGG16 and the baseline method. However, the models did not perform very well in terms of Precision, because they could not able to detect and eliminate the noise from the water level. Furthermore, the Deep CNN Encoder-Decoder approach with pre-trained weights after additional training with an artificial dataset could predict the changed and the unchanged class with an average F1 score equal to 98.44%. Nevertheless, it was not able to give meaningful predictions for the original dataset of physical models.

# Chapter 2: Related Work

This chapter reviews previous scientific approaches to similar problems. This chapter guides us to find important methods and approaches in order to select an appropriate algorithm. This chapter divided into four sections. The first section (Section 2.1) examines the existing approaches to identify changes in physical models and infrastructures. The second section (Section 2.2) discusses existing studies in the field of moving object detection. Afterwards, we explore the change detection techniques which used for images (Section 2.3). Finally, the last section (Section 2.4) concludes the chapter with an explanation of how the current thesis will contribute to the existing literature.

## 2.1 Existing studies for physical models and structures

The manual assessment of damage in infrastructures concerns all coastal and civil engineers (Lemos et al., 2016). The authors suggest a new and innovative approach to find and define any possible changes in the physical model structures. These changes are rocks' movement or units' displacement. They proposed a photogrammetric method in order to identify these movements by using only two digital photos taken before and after the scale model test. Furthermore, other authors suggest a method which implemented using the intensity difference between pixels (Courela et al., 2015). They developed a Matlab algorithm which tries to identify the armour unit displacement in physical models and more specific models which depict rubble-mound breakwaters. Both studies require the complete drainage of the wave basin at the end of each test.

The automatic assessment of a structure with a Convolutional neural network was implemented first in 2018 (Hoskere et al., 2018). They proposed a CNN to identify any possible cracks, corrosions or fatigues over the structures. They used labeled images of 6 different types of damage. They used these images to train their network and a test dataset to evaluate the accuracy of their proposed method. Until now, there is not a relevant automatic tool which uses machine learning techniques to make automatic change detection in the coastal structures. The research questions are:

**Research Question 1:** *To what extent can Deep Convolutional Neural Network methods detect changes in physical model structures?*

**Research Question 2:** *To what extent can a Deep Convolutional Encoder-Decoder model detect movements of small objects such as rocks?*

For this reason, in order to answer the research questions, it is necessary to examine the existing literature from a moving object detection and image change detection perspective.

## 2.2 Existing studies for moving object detection

Recently many papers published which examine the problem of detection of moving objects on videos or static images. Fragkiadaki, Arbelaez, Felsen, & Malik (2015) proposed a model in which they segment moving objects in videos by sorting spatiotemporal segment proposals calculating the probability to include a moving object. They define that as moving objectness. For every video frame, they calculate the aforementioned video segment proposals using multiple figure-ground segmentations on per frame motion boundaries. The ranking is possible with a Moving Objectness Detector which trained on images and motion fields in order to distinguish the moving objects and set aside over or under segmentations and background parts from the scene.

In addition, many studies suggest contour based moving object detection and tracking (Serby, Meier, & Van Gool, 2004; Yokoyama & Poggio, 2005). At the same time, many authors suggest background subtraction for moving object detection and tracking, such as Gaussian mixture models (Farou, Seridi, & Akdag, 2016) and adaptive distance threshold for background matching (Guang, Wang, & Xi, 2014).

Furthermore, J. Wu, Ye, Chen, & Weng (2018) proposed a Convolutional Neural Network-based object detection model. Their approach takes as an input two images. The first image is the cover page from a book and the second is a captured image. The network is trained by merging the two images into a 6-channel image and considers the differences between these two images as objects to detect. The network fed with the pair of images and the labels “same” or “different”. Their goal was to identify which photos are same or not.

## 2.3 Existing approaches to image change detection

The problem of change detection includes lots of applications in the field of video surveillance and remote sensing (Radke, Andra, Al-Kofahi, & Roysam, 2005). However, different applications require a different approach. For instance, the change detection algorithms for video surveillance examine the problem after calculating the image sequence and the relationship of the images in time-series (Bianco, Ciocca, & Schettini, 2017; Wang et al., 2014).

In remote sensing applications, there are different studies which compare the landscapes in two different situations and conditions and the images captured after a certain time. Most of the studies carried out first segmentation and classification of the images and then, comparison and analysis (Walter, 2004). He suggests a change detection approach based on an object-based classification of remote sensing data and not as a pixel-based. This means that the classification procedure is grouping the pixels which belong to one specific class. This assignment is done based on a supervised maximum likelihood classification among the classified objects.

Furthermore, many studies use post classification techniques for image detection. It is one of the most famous approaches in change detection analysis and a necessary factor is the comparison of the classified image. The general approach of this method is based on the correction of the classified images. El-Hattab (2016) proposes a methodology which divided into two parts to monitor the coastal zone from satellite images. The first part involves digital image processing and the analysis of satellite image to produce a land cover thematic map. The second includes the change detection analysis using post-classification.

In addition, many studies examine the problem of change detection in a pixel-based approach. There are studies which examine the problem through a direct image differencing problem (Coppin & Bauer, 1996). Also, there are several papers for image differencing methods which use different classifiers. Another pixel-based approach considers the image rationing as the main contributing factor to define which pixels belong to the changed or unchanged class (Xu, Zhang, He, & Guo, 2009). Image rationing techniques calculate the image ratios from the two images for every pair of pixels. If the ratio value for each pair of pixels is equal or close to 1, then it belongs to the unchanged class.

Other studies approach the problem using Change Vector Analysis (Chen, Gong, He, Pu, & Shi, 2003; Singh & Talwar, 2014) or CVA. This technique creates two outputs. The first is the magnitude of the change and the second the direction of the change. The approach starts with the elimination of any redundant information, followed by the calculation of changing magnitude among spectral change vectors between the pair of images. Furthermore, another technique is Principal Component Analysis (Celik, 2009; Kumar & Garg, 2013) or PCA. This is a linear transformation technique which takes the pair of images as an input and transforms them linearly in a way such the output images are linearly independent. The data are projected so the data with the highest variance lay on the first axis and the second highest variance to the second axis. This results in data redundancy reduction, however, the results are scene dependent (Hussain, Chen, Cheng, Wei, & Stanley, 2013). Also, a technique which is called Independent Component Analysis or ICA used for change detection (Xiao Benlin, Li Fangfang, Mao Xingliang, & Jin Huazhongb, 2008; Gulati & Pal, 2014). They consider second-order and high order dependencies between variables. The main target is to linearly transform the data so as the transformed data be highly independent from each other.

Moreover, some studies examine the change detection problem using Machine Learning algorithms. Liu & Lathrop (2002) using an artificial neural network (ANN) to detect which areas are more recently urbanized. Furthermore, Support Vector Machines (SVM) were used to identify the pixels as changed or unchanged as a binary classification problem (Bovolo, Bruzzone, & Marconcini, 2008). Also, a Support Vector Machine (SVM) algorithm is used to detect changes using image differencing technique over Kumta region, Karnataka, India (Karthik & Shivakumar, 2017). Other studies which implement machine learning techniques for change detection are using decision tree (Im & Jensen, 2005), random forest (Smith, 2010) and generic programming (Makkeasorn, Chang, & Li, 2009).

Furthermore, Sedaghat & Mahabal (2018) proposed an interesting approach to image differencing. They suggest a deep-learning approach to detect image differences that encloses all the steps of a traditional image subtraction pipeline into a single Convolutional Neural Network. These steps are image registration, background subtraction, noise removal, psf matching, and subtraction.

## 2.4 Contributions of the thesis

The contributions of the thesis are presented as follows:

- Firstly, we present a new model based on image feature extraction using a Deep Convolutional Neural Network (CNN) and more specifically the VGG16 and VGG19. These methods were applied in the physical model structures. To the best of our knowledge, in the existing literature is not examined and applied any deep learning methods to cast the problem of change detection in physical model structures. So, it will be a valuable contribution to the existing studies.
- Secondly, we propose a new Deep CNN Encoder-Decoder method with pre-trained weights in order to determine the changes in the physical models and to what extent can identify very minor changes such as rock displacements.
- Thirdly, Deltares used visual inspection techniques to detect the changes in their physical models. Thus, we provided them a novel approach to detect the changes in the physical model automatically. The proposed methods use cutting-edge techniques and more specific deep learning. Up to now, in the existing literature is not used any machine or deep learning models for the change detection problem in physical models.

# Chapter 3: Experimental Setup

In this section, the dataset and the experimental procedures are described in order to answer the research questions of the thesis. In the subsection 3.1, there is a detailed description of the dataset. Sub-section 3.2 describes the pre-processing techniques which used for this dataset. Sections 3.3-3.5 include the experimental setup and methodology for 3 models which were created and applied to the dataset. In section 3.6 the evaluation metrics are analyzed. Finally, subsection 3.7 presents the software used for the experiments.

## 3.1 Dataset Description

In this project, a dataset which was created by Deltares facilities was used. The complete dataset consists of about 600 images. The images were captured and collected in Deltares laboratory. The images depict various physical models which help coastal engineers to understand the complex hydrodynamic processes occurring in the nearshore zone and they provide solid and affordable engineering design solutions. These laboratory experiments are made of specific structures of rocks, stones and armour units and they need to stay stable, without damage so as to use them as an actual structure. These structures are physical models of breakwaters, harbor moles and dike revetments.

The pictures show the state of the structure before (Figure 1) and after the waves impact (Figure 2). The second image was captured after a long period of wave impacts on the physical models. The tests performed in Deltares laboratory. There are different pictures of structures, which depicts the changes over them. Furthermore, the water level varies and also the lighting is different in the pictures.

There are different physical model tests which performed in Deltares laboratories. For instance, there are physical model tests for breakwater roundheads, trunks and offshore rock series. In addition, every structure consists of different materials. There exist rock based units and stones with different size and shape and armour units.

Every structure consists of different sections and every section contains the aforementioned materials. These materials were painted in different colors. In some structures, the units were painted completely and in other structures, the units were painted only from one side. This happens in order to detect the flipping of stones more easily.

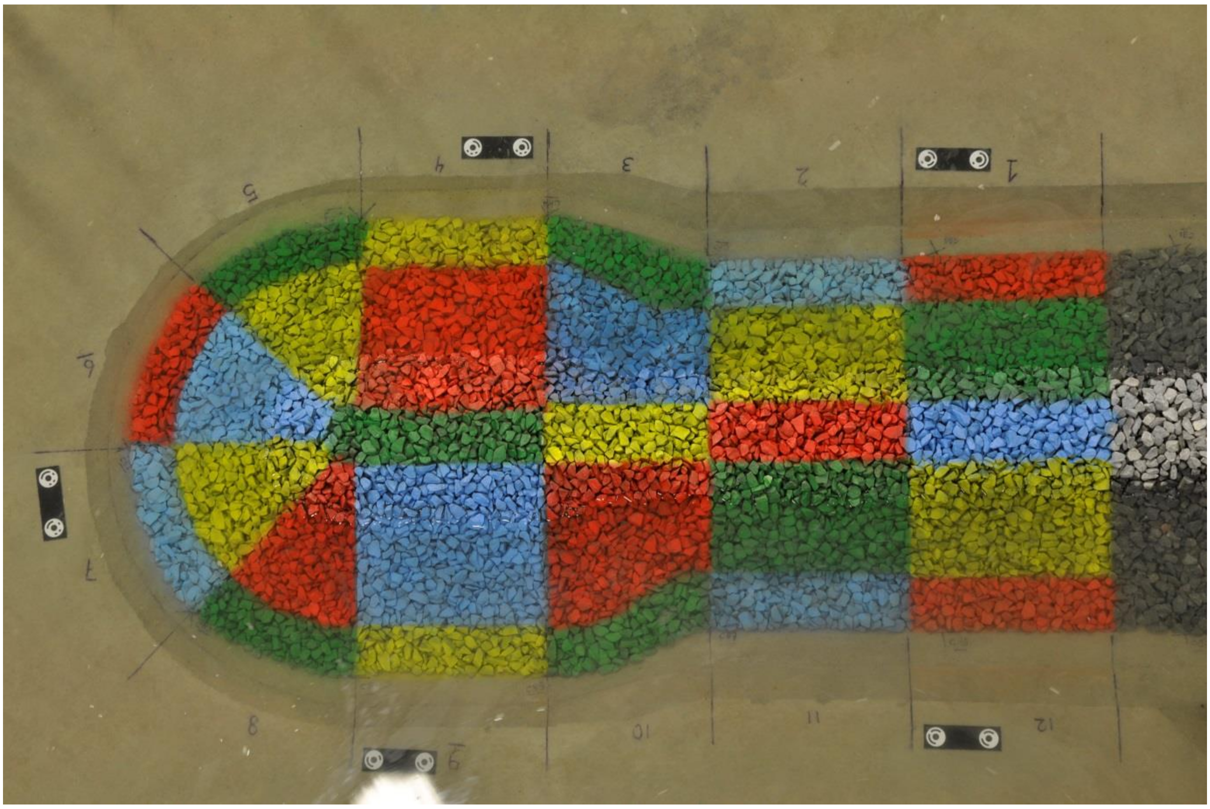
Figure 1: The reference picture, before the wave impact

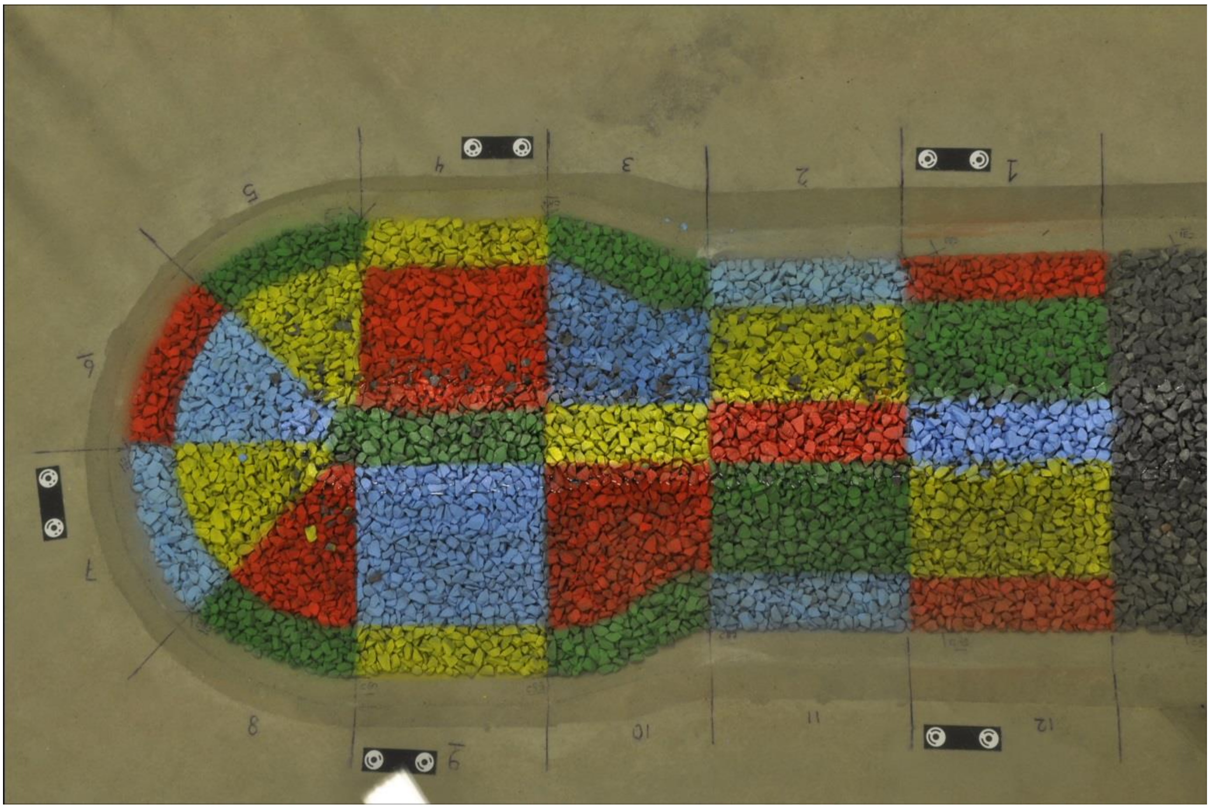
Figure 2: The image after the wave impact

The dataset includes the coordinates of the movements of rocks and stones. These movements were defined by comparing the reference image and the image after the wave impacts and trying to detect visually all the rocks and stones alterations. This technique is called scanning. The reference image is the image which depicts the physical model in its initial state. Furthermore, the dataset contains the name of the reference image and the name of the final image which are compared and also the coordinates of the units' alterations. These files were used in order to determine the pair of images which are going to be processed as well as to create the ground truth data. Figure 3 represents the coordinates on the final image.

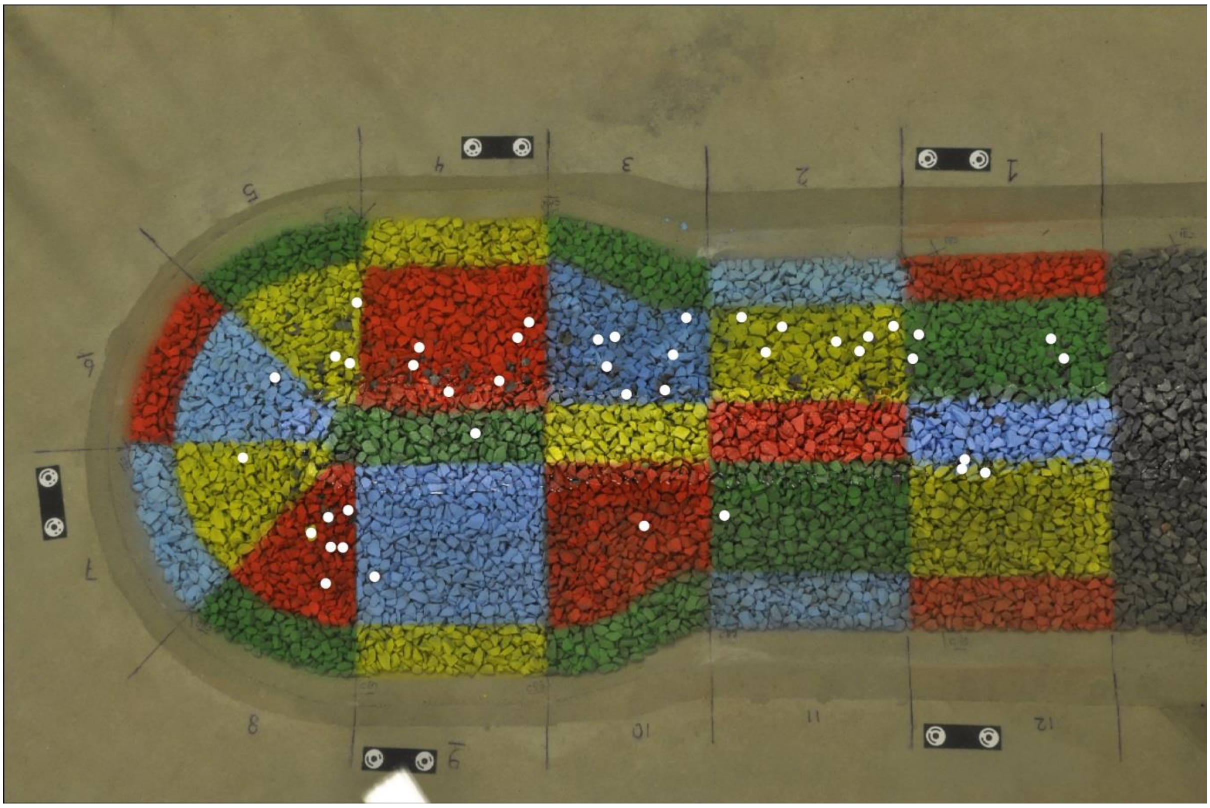

Figure 3: The white dots symbolize the coordinates into which there are rock movements

The dataset consists of 19 different physical model structures with different kinds of noise. For instance, in some images, the water level which exists in the physical models varies for each image and each physical model. For this reason, due to the refraction of light, the image has to be centred and registered in accordance with the reference image in order to compare the images on the same coordinates. Nevertheless, the water level contributes to the noise increase because of the lighting reflection. More specifically, the camera flash light is used in order to capture the images. Consequently, since the water level exists in some physical models the reflection light because of the camera flash light is different for each pair of images.

## 3.2 Data pre-processing

### 3.2.1 Ground Truth

In order to create the ground truth data, we used the available coordinates from the dataset. The coordinates were created after comparing the reference image and the image after the waves' impact. In the places that there were existed rock displacements or flips, the coordinates were pointed out. The scanning of the pair of images was carried out by coastal engineers.

A black image was created in the same dimensions to the original images. Afterwards, we generated the ground truth image for every pair of images (Figure 4).

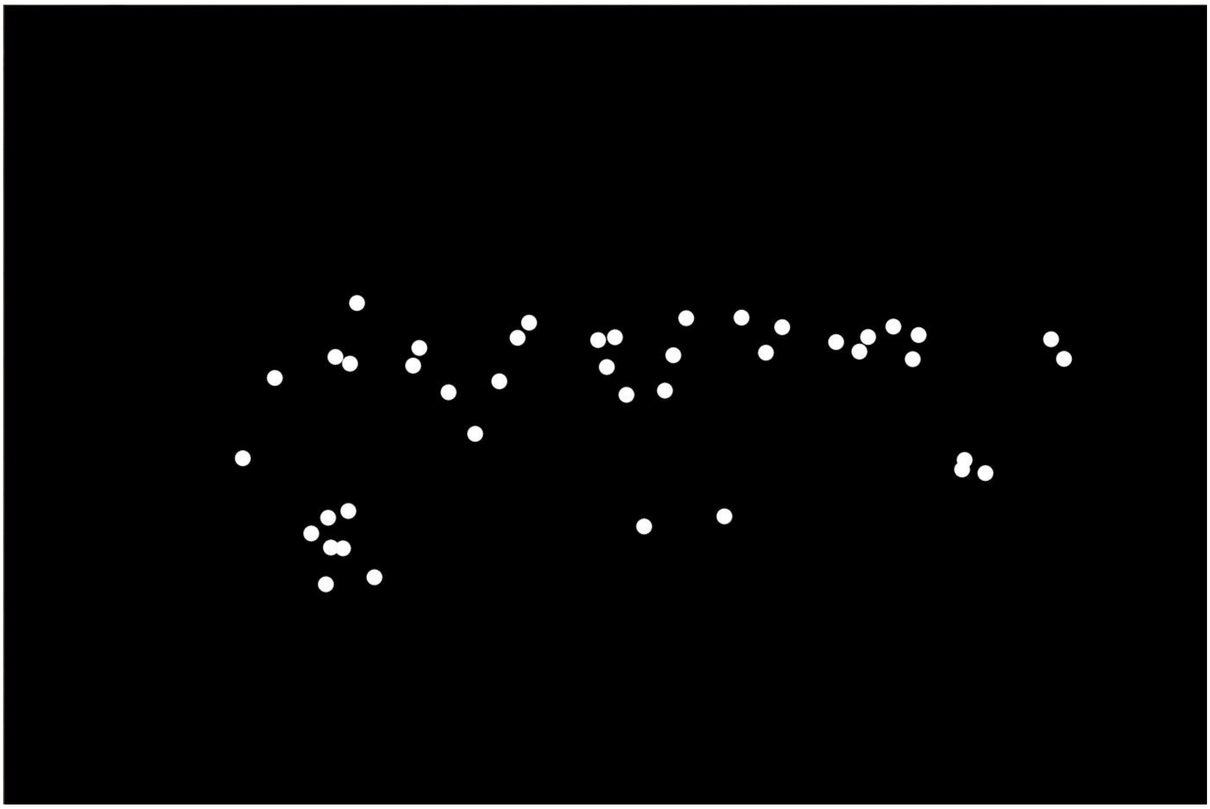

Figure 4: Ground truth image

The white dots symbolize the coordinates into which there are rock displacements. In this example Figure 4 is the outcome of the comparison between Figure 1 and Figure 2. The black colour in the image means that there is no difference between the images which means no change of the physical model. On the other hand, the white colour translated into image differencing between Figure 1 and Figure 2 which means that in those places we have a change in the physical model's structure.

In general, there are 134 pairs of images. Consequently, the total ground truth images are 134 and they will be compared with the models' predicted images so as to take the proper evaluation metrics to address the research questions.

### 3.2.2 Noise minimization

First of all, as it was mentioned in section 3.1, some images needed some adjustments to conform to the coordinates of the reference image. This was because of the water level and the phenomenon of refraction of the light. The adjustments took place only for the final images. The total images which are changed are 56 and depict changes for 8 physical model structures.

The first image (Figure 5) is an example of the aforementioned reference images. The final images need to be registered in order to make more accurate comparisons over the structure.

Figure 5: Reference image

The next image (Figure 6) is the second of the pair of images, which have been shifted so as to be centred according to the coordinates of the reference image. The new image is called registered and image registration is the process of transforming different images into the same coordinate system. The image was registered so as to be in accordance with the red screws which are screwed in the physical model structure. The aforementioned noise is the black colour in the images. The main idea is to keep the registered image in the initial state and the rest of the image which is filled with a black colour to be filled with the reference image (Figure 7). This is going to prevent the noise from registered images to affect the models' performance.

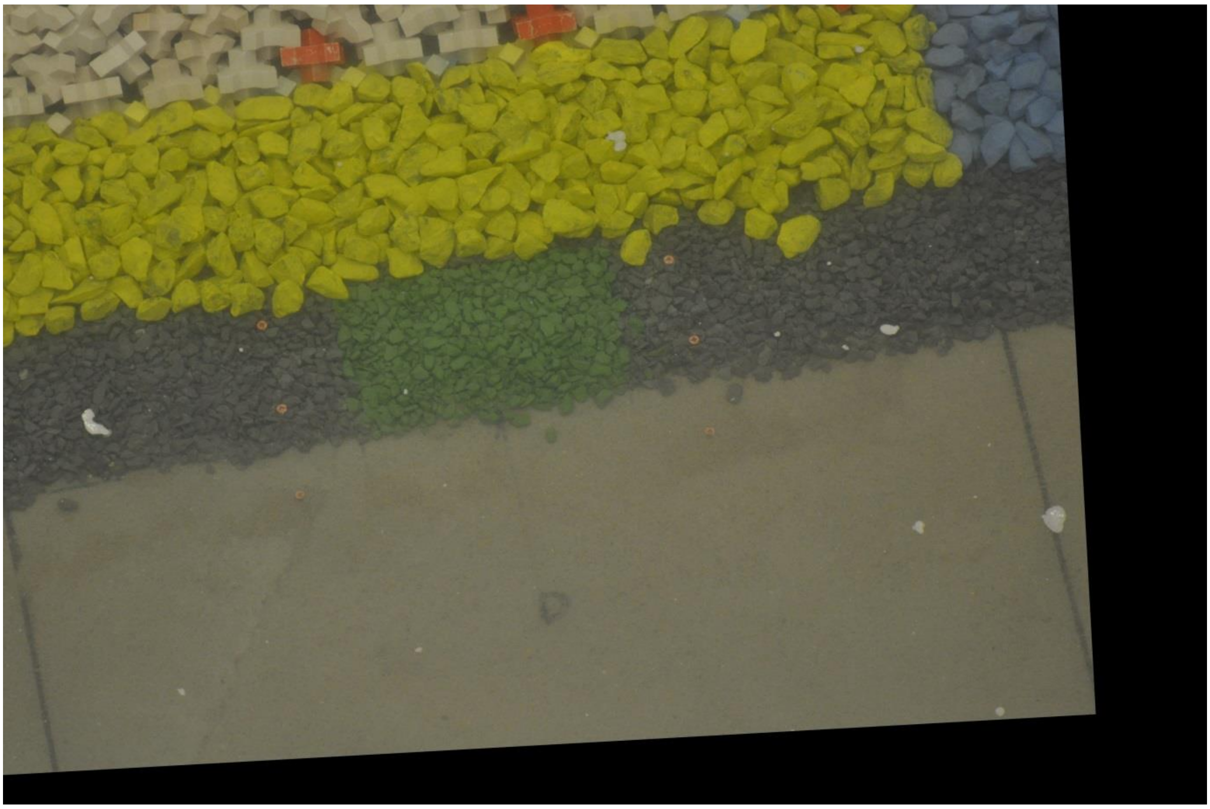
Figure 6: The registered image


Figure 7: Registered image after eliminating the noise

### 3.2.3 Image Resizing

An additional pre-processing technique was to resize the images. The original images have a size of 4288x2848 pixels. As a result, this makes it computationally demanding. Consequently, a possible solution is the image resizing. The final size of images was set at 640x425 pixels. The image resizing helped to apply the models and taking the predictions without memory errors.

## 3.3 PCA k-means Clustering method

Automatic change detection in images of a physical model is essential for coastal engineers. These images are known as multi-temporal images. Thus, change detection requires the analysis of two pairs of physical model images to detect any possible alterations. The PCA k-means algorithm was used previously to detect changes over remote sensing images (Celik, 2009).

This is an unsupervised learning approach to detect changes and differences in a pair of images. This method includes the automatic analysis of the change data. For that purpose, the difference image will be constructed using the pair of images. The difference image is the pixel by pixel subtraction of the two images. The PCA was used to extract the eigenvectors of pixel blocks from the difference image. Afterwards, a feature vector is created for each pixel of difference image by projecting that pixel's neighbourhood onto the eigenvectors. The feature vectors for all pixels form the feature vector space. The feature vector space is used to create two clusters using the k-means algorithm. The first cluster represents the pixels which belong to the changed class. On the other hand, the second cluster represents the pixels which are in the unchanged class. Every pixel will be part of either of the clusters and a change image can be generated. The steps which used in order to achieve this are the following:

1. Creation of the difference image and the eigenvector space (EVS).
2. Generation of the feature vector space (FVS)
3. Clustering of the feature vector space into two clusters.
4. Construction of the change image

### 3.3.1 Difference image and the eigenvector space

The value of each pixel of the difference image is equal to the absolute difference of the values of the corresponding pixels from the pair of images. The main idea is that the value of the pixels which are possible to connect to physical model changes will have values which are greatly different from the pixels associated with no unchanged parts of the physical model structure. The equation of the difference image pixel values is the following:

$$difference\ image[i,j] = |image\ 1[i,j] - image\ 2[i,j]|$$

The next necessary step is the building of the eigenvector space. Prior to that, it is necessary to explain some things about PCA. It is a widely used technique for dimensionality reduction. Principal

Component Analysis is a method to emphasize variation and find out strong patterns in a particular dataset. It transforms a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables. These variables are called principal components.

PCA can extract the covariance matrix of a given dataset after performing mean normalization on it. Subsequently, the eigenvectors and the eigenvalues are computed and then the eigenvectors are sorted in the descending order of eigenvalues. The eigenvector with the highest eigenvalue is the principal component of the dataset. The eigenvectors are sorted in the decreasing order of the eigenvalues. Consequently, by performing the PCA method, it is possible to derive the lines which characterize better the data.

In this approach, blocks of size 5x5 are created from the difference image and then they flattened them into row vectors. The image should be resized in order to make both dimensions a multiple of 5. As it was aforementioned the collection of these row vectors forms a vector set. On this vector set, we applied PCA so as to take the eigenvector space. The size of the eigenvector space is a 25x25 matrix and each column is an eigenvector of 25 dimensions. So, if the size of the difference image is m x n, then the number of rows in the vector set would be $\frac{m\ x\ n}{5\ x\ 5}$.

### 3.3.2 Generation of the feature vector space

In order to create the feature vector space (FVS) we take 5 x 5 blocks from the difference image, flatten them and project them onto the eigenvector space, but this time the blocks will be overlapping. The vector space is created by constructing one vector for each pixel of the difference image in a way that one 5 x 5 block is actually a pixel's 5 x 5 neighbourhood. It should be noted that following this methodology, 4 rows and 4 columns which belong to the boundary will not get any feature vectors because they do not have a 5 x 5 neighbourhood. We made this assumption because any changes are more possible to take place in the middle regions of the images rather than the edges. Thus, there will be m x n – 8 feature vectors in feature vector space with dimensions equal to 25. Below is the flowchart which is used to build the feature vector space (Figure 8).

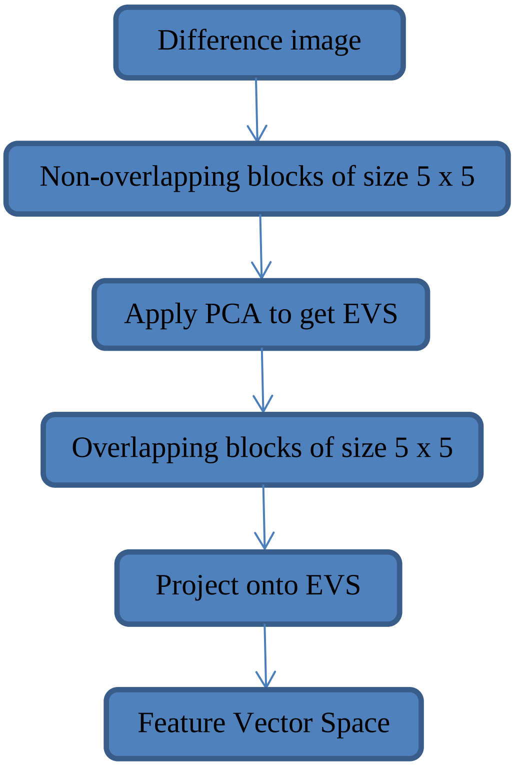


Figure 8: Detailing the process used to build the feature vector space

### 3.3.3 Clustering of the feature vector space

The pixel's feature vectors enclose the relevant information to determine which pixels are changed or unchanged. After the construction of the feature vector space, the next step is to cluster it in order to group the pixels into the two disjoint classes performing the K-means algorithm. It implies that each pixel will be assigned to the cluster in a way that the distance between the cluster's mean vector and the pixel's feature vector is the least. Every pixel will take a label from 1 to N, which represents the cluster number which they belong to.

The next step is to decide which clusters enclose the pixels that belonged to the changed and unchanged class. It can be assumed that the cluster which contains the lowest number of pixels is the cluster which belongs to the changed class. This happens because the general structure remains unchanged with only a very small number of changes. Furthermore, the mean of this cluster will be the highest, because the values of the difference image pixels in the places that there are changes over the structure are significantly higher than the value of pixels in the unchanged regions. Thus, the cluster belongs to the changed class if it encloses the lowest number of pixels and also the highest mean value.

### 3.3.4 Construction of the change image

In this step, the change image will be created. It is a binary image which shows the output of change detection. The regions which are not changed are in black color and the changes are shown in white.

This is an obvious task because it is easier to compare the model's predicted images with the relevant ground truth images.

## 3.4 Deep Convolutional Neural Networks VGG16 and VGG19 methods

The proposed method approaches the change detection problem in physical model structures from a feature learning manner. So, the proposed method is a deep Convolutional Neural Networks feature based change detection model. The main idea is to create a change detection image from the input pair of images by using a pre-trained CNN. The aforementioned features are calculated from different convolutional layers. Afterwards, a concatenation step is implemented for these features, followed by a normalization step. This results in a unique higher dimensional feature map. Lastly, the change detection image is computed using pixel-wise Euclidean distance. Below are the steps which used to implement this method:

1. Features extraction using CNN for the input pair of images.
2. Concatenation step.
3. Definition of changed or unchanged pixels
4. Creation of the change detection image

### 3.4.1 Convolutional Neural Networks

Deep Convolutional Neural Networks (CNN) have been proved that are suitable for image classification and recognition, because of their high accuracy. The ConvNets are able to capture appropriate features from an image (feature learning). In addition, a main feature of CNN is weight sharing. For a totally new problem, CNN is considered to be very good feature extractors. In addition, with back-propagation, CNN can be trained by adjusting the initial values to capture the correct magnitude of a spatial feature on which they are convolving. These filters essentially learn to capture spatial features from the input volumes based on the learned magnitude. As a result, they can achieve to compress a given image into a highly abstracted representation, which is easy to predict.

### 3.4.2 Features extraction using CNN for the input pair of images and concatenation step

Firstly, we read both images and their pixel's values are passed as an input to the model. In this approach, two pre-trained networks are used the VGG16 and the VGG19 models and we used their weights from the ImageNet competition to extract the features from the images.

ImageNet is a project for the purpose of computer vision research which aimed at manually labeling and classifying images into almost 22,000 separate categories. This project in terms of deep learning and Convolutional Neural Networks refers to the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2015). The target of this project is to train a model capable of recognizing and correctly categorizing an image into 1,000 separate object categories. The model was trained on almost 1.2 million training images and 50,000 images for validation and 100,000 images for testing.

The ImageNet challenge is the baseline for computer vision classification algorithms. Convolutional Neural Networks dominate this challenge since 2012. The state of the art pre-trained models which achieved high performance on ImageNet challenge are included in the Keras library. These models are very good as generalized models for images via feature extraction and fine-tuning. VGG16 and VGG19 are such pre-trained models.

The VGG architecture introduced in 2014 (Simonyan & Zisserman, 2014). The network use 3x3 convolutional layers which concentrated on top of each other. There is a max pooling layer to decrease the volume size and then two fully-connected layers followed by a softmax function. The number 16 and 19 denote the number of weight layers in the network.

The authors in order to task the problem of computational power they trained first smaller versions of VGG with fewer weight layers. Then, these smaller networks were used as initializations to the deeper network. This procedure is called pre-training. The sizes of both models are about 500MB, because of their depth and the number of fully connected layers.

In this case, the features retrieved from the intermediate layers using the Keras Backend function for different network layers. Afterwards, these features have to be extracted from the intermediate layers of the network. In this approach, the numbers of features which extracted from the pre-trained neural network are 5 and they are from different intermediate layers.

Lastly, the concatenation function was used to concatenate the features. This is necessary in order to pass the total features into one and use it to make the final decision of changed and unchanged pixels.

### 3.4.3 Definition of changed and unchanged pixels and creation of the change detection image

In this step, the total concatenated features are taken and these values are squared. The reason is to give more meaningful information to values which are different from these two images because the static background has more or less the same value for both images.

Afterwards, the difference from these squared values is calculated and the output is squared again in order to take the more meaningful information for each pixel. In order to define which pixels belong to the changed class and which are not, the Otsu’s thresholding was used.

The Otsu method was used in computer vision and image processing (Zhang & Hu, 2008) and it is an appropriate approach to perform clustering-based image thresholding. The main idea is that the image consists of two clusters of pixels following the bimodal histogram, the foreground (changed) and the background (unchanged) pixels. It calculates the optimum threshold and creates the two clusters in a way that their combined spread is the minimum.

After performing the Otsu's thresholding, the output value was used to create the change detection image. The pixels which their value is lower than the Otsu threshold are defined as no change. On the other hand, the pixels which their value is higher than the Otsu's threshold are the changed pixels. The same as to the previous approach the regions which are not changed are in black color and the changes are shown in white.

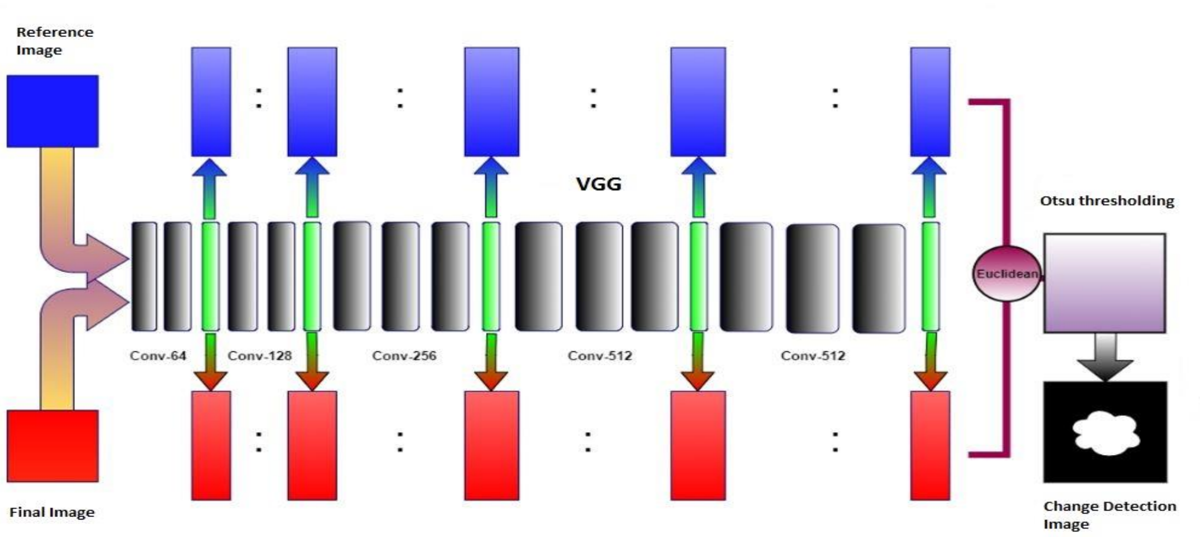


Figure 9: Feature extraction Deep CNN model for change detection

## 3.5 Deep Convolutional Encoder-Decoder Neural Networks Method

The proposed method tasks the change detection problem on physical models' structures after the waves' impact on an Encoder-Decoder manner. The main idea is to construct a change detection image from the input of the difference image. The difference image is the pixel by pixel subtraction of the two images. So, the network has as an input in the Encoder the difference image and the network uses the pre-trained VGG-16 weights trained on ImageNet. These weights used for transfer learning in our network's intermediate layers. The output of our network is the Decoder's output. It is an image with values in the RGB color channels and specific to the blue channel. The values are binary and specifically equal to 0 or 1. Then, this image should be transformed in order to have the final predicted change detection image. The steps which are used for this method are the following:

1. Creation of the difference image
2. Passing arguments to the network
3. Create the change detection image

### 3.5.1 Creation of the difference image

As it was mentioned in Section 3.1, the dataset contains the reference and the final image after the constant wave impacting and the coordinates which depict the changes over the structure. Consequently, that pair of images was selected to create the difference image. The difference image is the pixel by pixel subtraction of the two images. The equation of the difference image pixel values is the following:

$$difference\ image[i,j] = \ |image\ 1[i,j] - image\ 2[i,j]|$$

### 3.5.2 Passing arguments to the network

In this stage, the difference image is the input to the Convolutional Encoder. The Convolutional Encoder-Decoder which used is the Segnet model with VGG pre-trained weights. The Segnet neural network was first introduced in 2015 (Vijay Badrinarayanan, Handa, & Cipolla, 2015) and is a Convolutional Neural Network which used for semantic pixel-wise labeling.

This architecture has encoder and decoder layers. Each encoder uses convolution, batch normalization and a non-linearity. Afterwards, the max pooling has used the output of the previous step. At the same time, it stores the index of the value extracted from each window. The same logic exists for the decoder with the only difference that they do not include the non-linearity and also their input is indices from the encoder’s output.

The output from the last layer of the decoder is an annotated image with pixels’ values 0 or 1 in the RGB channel and more specific to the blue color channel. The pixels with a value equal to 0 belong to the unchanged class and the pixels equal to 1 belong to the changed class.

In our case, we used the total 134 images of the dataset. In order to train the network, we increased the dataset using several data augmentation techniques such as flips. As a result, the total number of images is 536 which separated into a training set, validation set and test set. The training set consists of 400 images, the validation set 100 images and the test set 36 images. The images which used as an input to the network are the difference image as mentioned before. The Decoder’s output images which used to train the network and do the validation and test set are the blue annotated images.

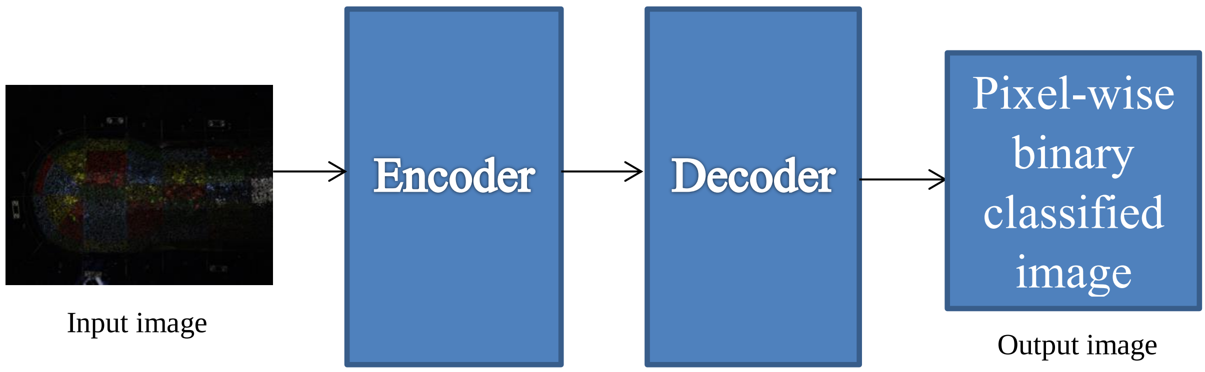

Figure 10: Deep Convolutional Encoder-Decoder architecture

The annotated images created using the ground truth images. As mentioned in Section 3.1, they were created from the available coordinates. The main idea is to define the black color (RGB values equal to 0) as the pixels with no movement and assign the value of the blue channel in the annotated image for those pixels equal to 0. On the other hand, the white color pixels (RGB values equal to 255) means that they belong to the changed class, so the value of blue channel in the annotated image is equal to 1.

### 3.5.3 Creation of change detection image

In this step, we detailed the procedure creating the change detection image. The predicted image from the network as mentioned in the previous paragraph is an annotated image with values in the blue channel 0 or 1. So, in order to convert this image to RGB image, we used a separate function. The main logic is to assign the values of pixels with value 0 as a black color. On the contrary, the values of predicted pixels with value 1 assigned as white.

## 3.6 Evaluation Method

As mentioned in the algorithms section the main idea is to use the coordinates of the rock movements and estimate the ground truth images. For this reason, we used the dataset to estimate these coordinates so as to define the location of the movements in the final image, by comparing it with the reference image and put a circular aperture around.

The most popular evaluation metric for binary classification problem with unbalanced data is the F1 score. Also, the change detection studies used the metrics of Precision, Recall and F1 score (Daudt, Saux, & Boulch, 2018).

| | | **Predicted** | |
|---|---|---|---|
| | | **Negative** | **Positive** |
| **Actual** | **Negative** | True Negative | False Positive |
| | **Positive** | False Negative | True Positive |

Figure 11: Confusion Matrix

In order to estimate the aforementioned metrics, an explanation of the values of the confusion matrix is necessary (Figure 11). **True Positives** are the correctly predicted positive values. That means that the value of predicted value is positive and also the value of the actual class is positive. In this case, true positives are when the number of pixels which correctly predicted to be in the changed class. In addition, **True Negatives** are the correctly predicted negative values and more specifically the number of pixels which correctly predicted to be to the unchanged class.

On the other hand, false positives and false negatives are the values when the actual class values are different from the predicted values. **False Positives** are the values when the actual values are no and the predicted values are yes. **False Negatives** are the values when actual class is positive, but the predicted class is negative.

Furthermore, the sum of the aforementioned metrics is equal to 272,000 for every image which is the number of the total pixels. This is because the size of the image is 640x425. The number of these metrics defined by comparing the predicted images to the ground truth images. So, we will calculate the F1 score for every image and then we will calculate their average value. **F1 Score** is the weighted average of Precision and Recall. Thus, this score takes into account the false positives and false negatives.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

**Precision** is the ratio of correctly predicted positive observation of the total predicted positive observations. Precision is a good measure to determine when the costs of False Positive is high.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} = \frac{True\ Positives}{Total\ Predicted\ Positives}$$

**Recall** is the ratio of correctly predicted positive observations to all observations in actual class – yes.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} = \frac{True\ Positives}{Total\ Actual\ Positive}$$

In Appendix A, we mention the software we used to conduct the experiments.

# Chapter 4: Experiments and Results

This section presents the main results. The first section includes the results for the entire dataset. The second section presents the results for the subset of the registered images. The last section represents the results derived from the Deep Convolutional Encoder-Decoder method for the artificial dataset.

## 4.1 Experiment 1: Original Dataset

In this section, we will derive and analyze the results for the original dataset which consists of 134 pairs of images. For every input pair of images, we applied all the proposed methods and the baseline model. Afterwards, the output is a predicted image for every input pair. Then, the predicted image (Figure 12) is compared with the relevant ground truth image (Figure 13) and we calculated the evaluation metrics for every predicted image and more specifically the precision, recall and f1 score.

Figure 12: Example of predicted image using PCA comparing Figure 1 and Figure 2

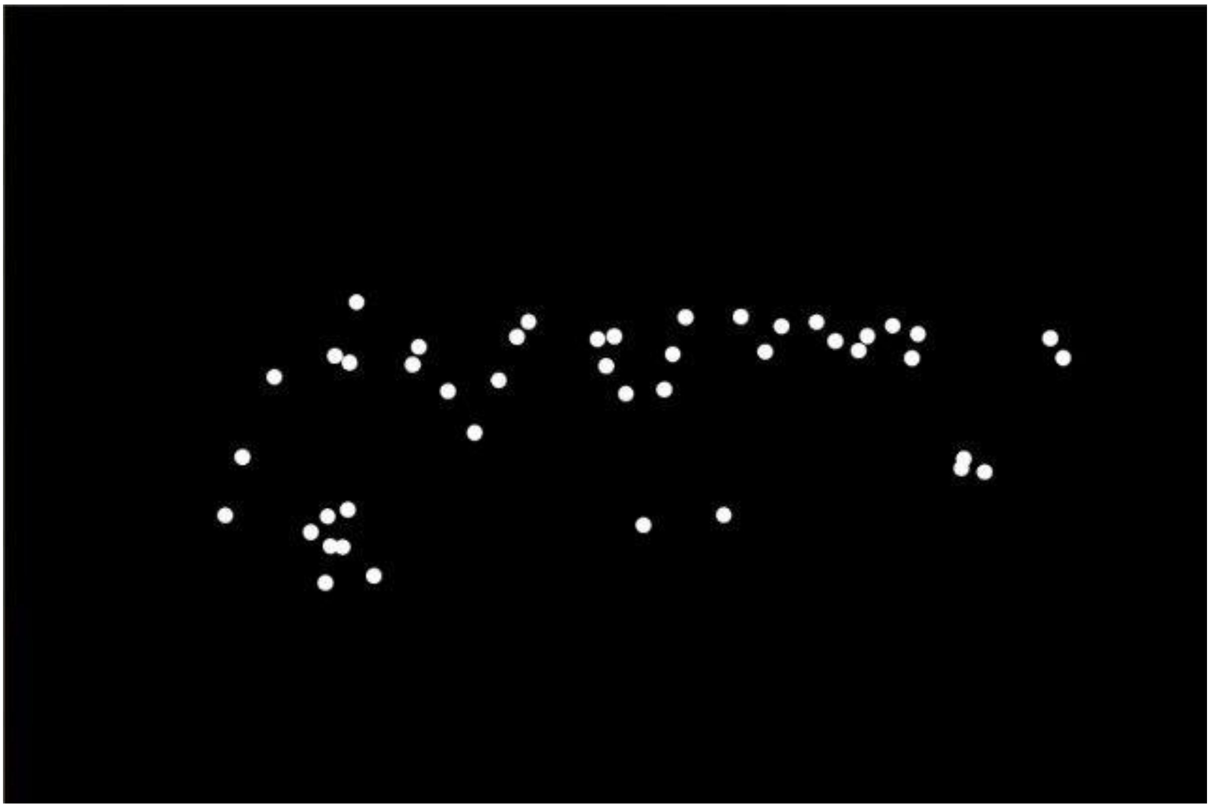

Figure 13: Ground truth image derived from Figure 1 and Figure 2

In the following table (Table 1), the average metrics of Precision, Recall and F1 score are presented. These metrics derived from the total pairs of images.

| ***Method*** | ***Precision*** ***(average)*** | ***Recall*** ***(average)*** | ***F1 score*** ***(average ± st. dev)*** |
|---|---|---|---|
| **PCA k-means** | 12.10% | 56.70% | 18.52% ± 7.05% |
| **VGG16** | 12.37% | 54.39% | 18.58% ± 7.08% |
| **VGG19** | **12.76%** | **61.55%** | **19.41%** ± 7.44% |

Table 1: Evaluation metrics for the total original dataset

According to the results, all the approaches have poor performance for the entire dataset. The first and the second appear to be giving very similar results. The third method gives slightly better results than the other two methods. Furthermore, recall is giving good results for the three approaches, but precision is significantly lower than recall. For this reason, this reduces the F1 rate as well. More specific VGG19 has the highest Precision rate and it differs by 0.39% from VGG16 and 0.66% from PCA k-means method. Also, the Recall value for VGG19 is higher than VGG16 by 7.16% and higher than PCA k-means by 4.85%. In terms of F1 score, VGG19 has the highest values. The standard deviation shows that the results have many fluctuations among the predictions.

## 4.2 Registered Images

Afterwards, we will evaluate the performance of the registered images as we discussed in section 3.2.2. As mentioned in the dataset description section, the total images which have been registered are 56 pairs of images from 8 physical models. For all these images we eliminated the noise and then we applied the methods. The table below (Table 2) includes the average value of Precision, Recall and F1 score for this subset.

| ***Method*** | ***Precision*** ***(average)*** | ***Recall*** ***(average)*** | ***F1 score*** ***(average ± st. dev)*** |
|---|---|---|---|
| **PCA k-means** | 14.09% | 51.91% | 20.87% ± 9.81% |
| **VGG16** | 15.03% | 51.01% | 21.13% ± 9.92% |
| **VGG19** | **15.92%** | **56.08%** | **22.31%** ± 10.34% |

Table 2: Evaluation metrics for the registered images

It is apparent from the results that the VGG19 method performs better than the other two approaches. More specific VGG19 has higher precision rate than VGG16 method by 0.89%. Also, it is higher than the PCA k-means method by 1.83%. In terms of recall rate, the VGG19 model has the highest value among the other two models. It is higher than VGG16 model's recall value by 5.07% and PCA k-means recall value than 4.17%. For that reason, the overall average F1 score of VGG19 model is higher in comparison to the other two methods. It is higher than VGG16's F1 score by 1.18%. Also, it is higher than PCA k-means by 1.18%.

It should be mentioned here that the Deep Convolutional Encoder-Decoder method did not produce any meaningful predictions. The network failed to predict accurately and it predicts the same output for every input. The predicted class is always the unchanged class for every pixel. Firstly, we tried to train the network by feeding it with all the pairs of images from the total original dataset, but it failed to give correct predictions. Also, we tried to train it only with registered images, but again it failed to produce any meaningful predictions even for this dataset.

## 4.3 Deep Convolutional Encoder-Decoder method

As it was mentioned in previous sections the Deep CNN Encoder-Decoder model with pre-trained weights could not produce meaningful predictions. The predicted class is always the same for every input. This class is the unchanged class and the output is a black image, which means that there are no changes over the physical model.

For this reason, we tried to identify any possible improvements to our network and identify the reason why it was not able to give meaningful predictions. In order to achieve this, we created an artificial dataset. This dataset has unbalanced data with 90% belongs to the unchanged class and 10% belongs to the changed class. This proportion even it can be described as unbalanced the difference between the two classes is lower in comparison to the original dataset. From the original dataset, almost 99% belong to the unchanged class and only 1% belongs to the changed class. The following image (Figure 14) is an example of the artificial dataset.

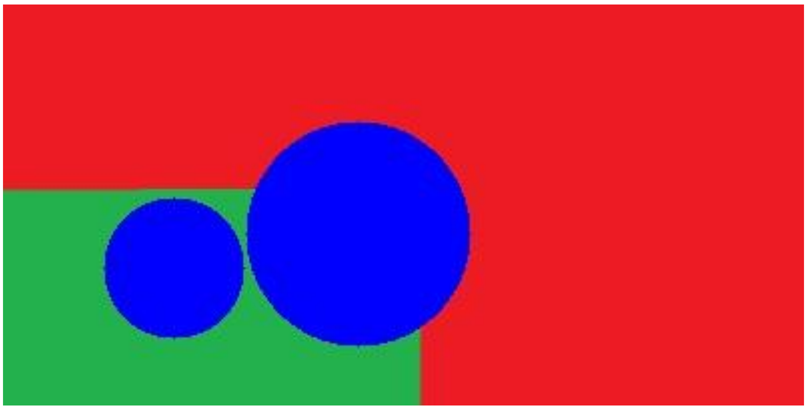

Figure 14: Example image from artificial dataset

The above image is one of the input images to the model. The output is a pixel-wise binary annotated image as described in section 3.5. The blue circles are the pixels which belong to the changed class and the rest of the image belong to the unchanged class. The total dataset consists of 1000 training images, 200 validation images and 200 images as a test set. Each image is 400x200 pixels.

The following table (Table 3) includes the average values of the evaluation metrics of Precision, Recall and F1 score for the test set. The models were trained with different loss functions, different optimizers and different activation functions for the intermediate layers and the last layer. At the first attempt, the categorical cross entropy loss function and adadelta optimizer were used. The activation function was the rectified linear unit for the intermediate layers. The softmax function was the activation function for the last dense layer. In the second attempt, the loss function was binary cross entropy and the optimizer was adam. The activation function which was used for every layer was the sigmoid function. Finally, in the third attempt, the network was trained having binary cross-entropy as a loss function and adam as an optimizer. The activation function which was used to the intermediate layers is the tan function. The last layer has as an activation function the sigmoid function.

| ***Attempts*** | ***Precision*** *(average)* | ***Recall*** *(average)* | ***F1 score*** *(average)* |
|---|---|---|---|
| **1st Attempt** | 98.49% | 71.03% | 81.27% ± 14.16% |
| **2nd Attempt** | **99.91%** | 50.24% | 64.80% ± 18.34% |
| **3rd Attempt** | 99.03% | **97.88%** | **98.44%** ± 0.93% |

Table 3: Evaluation metrics for Deep CNN Encoder-Decoder for artificial dataset

It is apparent from the table that the third attempt has significant higher F1 score and Recall than the other two attempts. The difference between the third and the second attempt is 33.64% and the difference from the first attempt is 17.17%. The difference for the Recall metric between the third and the second attempt is 47.64%, while the difference between the third and the first attempt is 26.85%. The highest average precision is from the second attempt and its difference from the first attempt is by 1.42% and from the third attempt at 0.88%. It is obvious that the precision has very minor differences among the three attempts. On the other hand, the recall value is significantly different for the two attempts. This happens because those attempts cannot detect a considerable amount of pixels which belong to the changed class.

It should be noted that these combinations applied to the original dataset as well. Unfortunately, the network still could not produce meaningful predictions. The predictions once again were the same for every input and it was the unchanged class. Furthermore, in order to detect how the different combinations of loss functions, optimizers and activation functions affect models performance on the artificial dataset we tried different combinations (Table 4). These combinations could not give correct predictions for the artificial dataset. The same as the original dataset these combinations give as predictions the same class and more specific the unchanged class.

| Loss Function | Optimizer | Activation function in the intermediate layers | Activation function in the last dense layer |
|---|---|---|---|
| **Binary Crossentropy** | adadelta | ReLU | softmax |
| **Binary Crossentropy** | adadelta | ReLU | sigmoid |
| **Binary Crossentropy** | adam | ReLU | softmax |
| **Categorical Crossentropy** | adadelta | ReLU | sigmoid |
| **Categorical Crossentropy** | adam | ReLU | sigmoid |

Table 4: Combinations of different training parameters

# Chapter 5: General Discussion and Conclusion

This chapter provides a general discussion on change detection from images of physical models. We present the conclusions and the results of our research and give recommendations and suggestions for future research. The first section of this chapter is dedicated to revisiting the results of our experiments and answer the research questions which formulated in the introduction. In section 5.2, we conclude our results with several directions and suggestions for future research in the field of change detection in physical model structures.

## 5.1 Answers to the research questions

The major objective of this study was to investigate the change detection problem in physical model structures. Part of the aim of this project is to develop methods able to detect these changes. It should be noted that we used cutting-edge techniques to cast the problem of change detection in physical model structures and more specifically deep learning methods. So far, Deltares used manual visually inspection techniques to detect any possible changes in physical models. So, the creation of these methods will contribute to detect the changes in the physical models automatically and faster. Also, as it aforementioned previous studies in this field suggest photogrammetric methods and digital images processing techniques. As a result, we proposed novel methods in this field using deep learning techniques to cast the problem of change detection in physical models.

In this section, we will discuss and conclude upon each of the research questions. The answers will be conducted after taking into account the results and the existing literature for the problem of change detection.

**Research question 1:** *To what extent can Deep Convolutional Neural Network methods detect changes in physical model structures?*

In Chapter 3 all the models were analyzed and presented. In order to answer this research question, we tried different pre-processing techniques and procedures which are analyzed in chapter 3. Furthermore, we applied all the models to the available dataset and we evaluated the predictions by using the evaluation metrics of Precision, Recall and F1 score (Daudt et al., 2018). In addition, we investigated the particular dataset with the registered images and determine how the models performed.

It is apparent that in order to answer this research question, we need to keep into account the evaluation metrics that we present in Chapter 4. According to the results which presented in Table 1, the proposed Deep CNN VGG19 model achieved the highest average scores for all evaluation metrics and more specific for precision, recall and F1 score. The F1 score for this model was higher than the baseline model (PCA k-means) by 0.89%. Furthermore, the average F1 score for the VGG19 model is higher than VGG16 model by 0.83. That means that the differences are really small between the models.

Also, it is obvious that the models did not achieve high precision performance and this had a negative effect on the F1 score as well. This happens because our dataset includes a lot of noise derived from flash lighting, the water level and image registration. Table 2 represents the result of the methods which are applied to the subset of registered images. The average metrics are higher for all methods in compare to the general average scores for all the available dataset. This can lead to the outcome that noise highly affects the models' performance. Different datasets can lead to different results (Daudt et al., 2018).

It should be mentioned here that our proposed model Deep CNN Encoder-Decoder with pre-trained weights did not give any meaningful predictions. The predicted class was always the same for every pixel for every input. The predicted class was the unchanged class. The next research question was dedicated to discover the reason behind this.

In conclusion, the models achieved low scores in terms of precision which have led to a reduced F1 score. That happened because the dataset contains a lot of noise and our models are unable to detect it. So, many predictions which belong to the changed class were wrong.

**Research question 2:** *To what extent can a Deep Convolutional Encoder-Decoder model detect movements of small objects such as rocks?*

The proposed model could not produce meaningful predictions for the original dataset. This happens because of the proportion of the two classes. In the changed class belong almost 99% of the pixels and in the changed class belong only the 1%. In order to prove this argumentation we created an artificial dataset with 90% of the pixels belong to the unchanged class and 10% of the pixels to the changed class.

We tried many combinations of different loss functions, optimizers and activations functions which are explained in detail in subsection 4.3. According to the results (Table 3), the method gives meaningful predictions for three combinations. The best results were the model which was trained by having binary cross-entropy as a loss function and adam as an optimizer. The activation function which was used to the intermediate layers was the tan function. The last layer had as an activation function the sigmoid. It achieved really high performance with 98.44% average F1 score on the test set. That means that the model was able to detect almost every changed or unchanged pixel correctly no matter the size.

In conclusion, the proposed model can detect small changes with the assumption that the training dataset should contain at least 10% from the changed class and 90% from the unchanged class. This was the reason why our approach did not work for the original dataset for every combination we tried.

## 5.2 Limitations and future research

Pointing out the major limitations of this thesis, it will help us suggest directions for future research. The presented thesis and consequently the presented research is limited because of the noise which exists in the available dataset. The proposed models, as well as the baseline, could not achieve high F1 score because they are not able to detect the noise which was in the pictures. So, our first recommendation derived from this fact. The proposed models are suitable to work with databases without unnecessary noise. The water level can influence the lighting because of the flash lighting over the water. Consequently, the images must be captured after the complete drainage of the flume. Furthermore, the second image should be registered according to the reference image.

The second limitation derived from the Deep CNN Encoder-Decoder unavailability to produce meaningful predictions. As we proved with the artificial dataset the network works for small items detection. The reason that it does not work for the original dataset is that the pre-trained weights which we used to train the network are able to detect bigger objects such as humans, traffic lights etc. (V. Badrinarayanan, Kendall, & Cipolla, 2017) and not small items like rock and stones. Thus, in order to use this method and detect the displacements of small objects (Lebedev, Vizilter, Vygolov, Knyaz, & Rubis, 2018) we need to train our network without pre-trained weights. However, in order to achieve this, we need thousands of images to train our network. So, a future recommendation should be in the direction of collecting or creating synthesized images so as to create a dataset big enough to train the network. Furthermore, another suggestion derived from the limited amount of data is to create a new network based on Siamese Networks (Daudt et al., 2018). This new technique for remote sensing change detection managed to achieve high performance using only the available datasets to train their network from scratch.

# Referencies


Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(12), 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615

Badrinarayanan, Vijay, Handa, A., & Cipolla, R. (2015). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling. *ArXiv:1505.07293 [Cs]*. Retrieved from http://arxiv.org/abs/1505.07293

Benlina, X., Fangfang, L., & Huazhongb, J. (2008). Study on Independent Component Analysis ' Application in Classification and Change Detection of Multispectral Images.

Bianco, S., Ciocca, G., & Schettini, R. (2017). How Far Can You Get By Combining Change Detection Algorithms? *ArXiv:1505.02921 [Cs]*, *10484*, 96–107. https://doi.org/10.1007/978-3-319-68560-1_9

Bovolo, F., Bruzzone, L., & Marconcini, M. (2008). A Novel Approach to Unsupervised Change Detection Based on a Semisupervised SVM and a Similarity Measure. *IEEE Transactions on Geoscience and Remote Sensing*, *46*(7), 2070–2082. https://doi.org/10.1109/TGRS.2008.916643

Celik, T. (2009). Unsupervised Change Detection in Satellite Images Using Principal Component Analysis and$k$-Means Clustering. *IEEE Geoscience and Remote Sensing Letters*, *6*(4), 772–776. https://doi.org/10.1109/LGRS.2009.2025059

Chen, J., Gong, P., He, C., Pu, R., & Shi, P. (2003, April). Land-Use/Land-Cover Change Detection Using Improved Change-Vector Analysis [Text]. https://doi.org/info:doi/10.14358/PERS.69.4.369

Coppin, P. R., & Bauer, M. E. (1996). Digital change detection in forest ecosystems with remote sensing imagery. *Remote Sensing Reviews*, *13*(3–4), 207–234. https://doi.org/10.1080/02757259609532305

Courela, J. M., Carvalho, R., Lemos, R., Fortes, C. J. E. M., & Leandro, J. (2015). Rubble-mound breakwater armour units displacement analysis by means of digital images processing methods in scale models. Retrieved from http://repositorio.lnec.pt:8080/xmlui/handle/123456789/1007180

Daudt, R. C., Saux, B. L., & Boulch, A. (2018). Fully Convolutional Siamese Networks for Change Detection. In *2018 25th IEEE International Conference on Image Processing (ICIP)* (pp. 4063–4067). https://doi.org/10.1109/ICIP.2018.8451652

de Almeida Sousa, E., Van Gent, M., & Hofland, B. (2018). Damage Characterisation of Rock Armoured Slopes. *Proceedings of the 7th International Conference on the Application of Physical Modelling in Coastal and Port Engineering and Science (Coastlab18)*. Retrieved from http://resolver.tudelft.nl/uuid:12c99929-db8e-47fb-bad4-4dfad372467f

El-Hattab, M. M. (2016). Applying post classification change detection technique to monitor an Egyptian coastal zone (Abu Qir Bay). *The Egyptian Journal of Remote Sensing and Space Science*, *19*(1), 23–36. https://doi.org/10.1016/j.ejrs.2016.02.002

Farou, B., Seridi, H., & Akdag, H. (2016). Improved gaussian mixture model with background spotter for the extraction of moving objects. *Int. Arab J. Inf. Technol.*, *13*, 807–816.

Fragkiadaki, K., Arbelaez, P., Felsen, P., & Malik, J. (2015). Learning to segment moving objects in videos. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4083–4090). Boston, MA, USA: IEEE. https://doi.org/10.1109/CVPR.2015.7299035

Guang, H., Wang, J., & Xi, C. (2014). Improved visual background extractor using an adaptive distance threshold. *Journal of Electronic Imaging*, *23*(6), 063005. https://doi.org/10.1117/1.JEI.23.6.063005

Gulati, V., & Pal, P. (2014). Enhancement of ICA Algorithm Using Mat lab for Change Detection in Hyper Spectral Images, *1*(5), 9.

Hoskere, V., Narazaki, Y., Hoang, T., & Spencer Jr, B. (2018). Vision-based Structural Inspection using Multiscale Deep Convolutional Neural Networks. *ArXiv:1805.01055 [Cs]*. Retrieved from http://arxiv.org/abs/1805.01055

Hussain, M., Chen, D., Cheng, A., Wei, H., & Stanley, D. (2013). Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS Journal of Photogrammetry and Remote Sensing*, *80*, 91–106. https://doi.org/10.1016/j.isprsjprs.2013.03.006

Im, J., & Jensen, J. R. (2005). A change detection model based on neighborhood correlation image analysis and decision tree classification. *Remote Sensing of Environment*, *99*(3), 326–340. https://doi.org/10.1016/j.rse.2005.09.008

Karthik, & Shivakumar, B. R. (2017). Change detection using image differencing: A study over area surrounding Kumta, India. In *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)* (pp. 1–5). https://doi.org/10.1109/ICECCT.2017.8117851

Kumar, V., & Garg, K. D. (2013). A Valuable approach for Image Processing and Change Detection on Synthetic Aperture Radar Data. *International Journal of Current Engineering and Technology*, 5.

Lebedev, M. A., Vizilter, Y. V., Vygolov, O. V., Knyaz, V. A., & Rubis, A. Y. (2018). CHANGE DETECTION IN REMOTE SENSING IMAGES USING CONDITIONAL ADVERSARIAL NETWORKS. In *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (Vol. XLII–2, pp. 565–571). Copernicus GmbH. https://doi.org/10.5194/isprs-archives-XLII-2-565-2018

Lemos, R., Loja, M. a. R., Rodrigues, J., & Rodrigues, J. A. (2016). Photogrammetric analysis of rubble mound breakwaters scale model tests. *AIMS Environmental Science*, *3*(3), 541–559. https://doi.org/10.3934/environsci.2016.3.541

Liu, X., & Lathrop, R. G. (2002). Urban change detection based on an artificial neural network. *International Journal of Remote Sensing*, *23*(12), 2513–2518. https://doi.org/10.1080/01431160110097240

Makkeasorn, A., Chang, N.-B., & Li, J. (2009). Seasonal change detection of riparian zones with remote sensing images and genetic programming in a semi-arid watershed. *Journal of Environmental Management*, *90*(2), 1069–1080. https://doi.org/10.1016/j.jenvman.2008.04.004

Radke, R. J., Andra, S., Al-Kofahi, O., & Roysam, B. (2005). Image change detection algorithms: a systematic survey. *IEEE Transactions on Image Processing*, *14*(3), 294–307. https://doi.org/10.1109/TIP.2004.838698

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

Sedaghat, N., & Mahabal, A. (2018). Effective Image Differencing with ConvNets for Real-time Transient Hunting. *Monthly Notices of the Royal Astronomical Society*, *476*(4), 5365–5376. https://doi.org/10.1093/mnras/sty613

Serby, D., Meier, E. K., & Van Gool, L. (2004). Probabilistic object tracking using multiple features. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* (pp. 184-187 Vol.2). Cambridge, UK: IEEE. https://doi.org/10.1109/ICPR.2004.1334091

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv:1409.1556 [Cs]*. Retrieved from http://arxiv.org/abs/1409.1556

SINGH, S., & TALWAR, R. (2014). A comparative study on change vector analysis based change detection techniques. *Sadhana*, *39*(6), 1311–1331. https://doi.org/10.1007/s12046-014-0286-x

Smith, A. (2010). Image segmentation scale parameter optimization and land cover classification using the Random Forest algorithm. *Journal of Spatial Science*, *55*(1), 69–79. https://doi.org/10.1080/14498596.2010.487851

Walter, V. (2004). Object-based classification of remote sensing data for change detection. *ISPRS Journal of Photogrammetry and Remote Sensing*, *58*(3), 225–238. https://doi.org/10.1016/j.isprsjprs.2003.09.007

Wang, Y., Jodoin, P., Porikli, F., Konrad, J., Benezeth, Y., & Ishwar, P. (2014). CDnet 2014: An Expanded Change Detection Benchmark Dataset. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 393–400). https://doi.org/10.1109/CVPRW.2014.126

Wolters, G., van Gent, M., Allsop, W., Hamm, L., & Muhlestein, D. (2010). HYDRALAB III: Guidelines for physical model testing of rubble mound breakwaters. In *Coasts, marine structures and breakwaters: Adapting to change* (Vols. 1–0, pp. 2: 559-670). Thomas Telford Ltd. https://doi.org/10.1680/cmsb.41318.0062

Wu, J., Ye, Y., Chen, Y., & Weng, Z. (2018). Spot the Difference by Object Detection. *ArXiv:1801.01051 [Cs]*. Retrieved from http://arxiv.org/abs/1801.01051

Yokoyama, M., & Poggio, T. (2005). A Contour-Based Moving Object Detection and Tracking. In *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance* (pp. 271–276). Beijing, China: IEEE. https://doi.org/10.1109/VSPETS.2005.1570925

Zhang, J., & Hu, J. (2008). Image Segmentation Based on 2D Otsu Method with Histogram Analysis. In *2008 International Conference on Computer Science and Software Engineering* (Vol. 6, pp. 105–108). https://doi.org/10.1109/CSSE.2008.206

# Appendix A

The experiments were conducted using the Python programming language and version 3.5.6. Furthermore, the creation of ground truth data was done with Matlab. We used the following Python libraries to create the methods and do the data analysis:

- opencv
- numpy
- sklearn
- collections
- scipy
- Tensorflow
- keras
- matplotlib
- sys
- argparse
- datetime
- glob
- itertools
- os
- xlsxwriter