Performance Comparison of Hybrid CNN-SVM and CNN-XGBoost models in Concrete Crack Detection

Sahana Thiyagarajan

Technological University Dublin

Performance Comparison of Hybrid CNN-SVM and CNN-XGBoost models in Concrete Crack Detection

OLLSCOIL TEICNEOLAÍOCHTA
BHAILE ÁTHA CLIATH
DUBLIN
TECHNOLOGICAL
UNIVERSITY DUBLIN

Sahana Thiyagarajan

Technological University Dublin

A dissertation submitted in partial fulfillment of the requirements of
Technological University Dublin for the degree of
M.Sc. in Computer Science (Data Analytics)

2019

DECLARATION

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Data Analytics), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Technological University Dublin and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: ______________________________

Sahana Thiyagarajan

Date: **01 September, 2019**

ABSTRACT

Detection of cracks mainly has been a sort of essential step in visual inspection involved in construction engineering as it is the commonly used building material and cracks in them is an early sign of de-basement. It is hard to find cracks by a visual check for the massive structures. So, the development of crack detecting systems generally has been a critical issue. The utilization of contextual image processing in crack detection is constrained, as image data usually taken under real-world situations vary widely and also includes the complex modelling of cracks and the extraction of handcrafted features. Therefore the intent of this study is to address the above problem using two-hybrid machine learning models and classify the concrete digital images into having cracks or non-cracks. The Convolutional Neural Network is used in this study to extract features from concrete pictures and use the extracted features as inputs for other machine learning models, namely Support Vector Machines (SVMs) and Extreme Gradient Boosting (XGBoost). The proposed method is evaluated on a collection of 40000 real concrete images, and the experimental results show that application of XGBoost classifier to CNN extracted image features include an advantage over SVM approach in accuracy and achieve a relatively better performance than a few existing methods.

Keywords: *Image Classification, CNN, Feature Extraction, Support Vector Machine, Extreme Gradient Boosting, Crack Detection, Machine Learning*

ACKNOWLEDGEMENTS

Firstly, I would like to express my gratitude to my Supervisor **Dr.Bojan Bozic,** for providing invaluable guidance throughout this research. It was a great honor to work and study under his supervision.

I submit my heartiest gratitude to **Dr.Luca Longo** for the encouragement and insightful comments given. Without his guidance and diligent help, this dissertation would not have been conceivable.

I'm deeply indebted to all the TUD staffs of the School of Computing for their sincere help for completing this thesis.

My special appreciation to my classmates and friends in providing relevant assistance and help to complete this study.

Last but not least, I also want to thank my beloved parents and sister for being my inspiration. You're the reason why I keep pushing.

Special thanks to all the people who along the way believed in me.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

List of Acronyms

AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Program Interface
AUC	Area Under the Curve
CIFAR-10	Canadian Institute For Advanced Research
CMOS	Complementary Metal-Oxide Semiconductor
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CRISP-DM	Cross InduStry Process for Data Mining
CV	Cross-Validation
ELM	Extreme Learning Machine
GB	GigaByte
GPU	Graphics Processing Unit
IPT	Image Processing Techniques

KNN	K-Nearest Neighbor
METU	Middle East Turkey University
ML	Machine Learning
MLP	Multi-Layer Perceptron
MNIST	Modified National Institute of Standards and Technology database
NN	Neural Network
PC	Personal Computer
PDE	Partial Differential Equations
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RGB	Red, Green, Blue
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
UAV	Unmanned Aerial Vehicle
VEDAI	VEhicle-Detection Aerial-Imagery
XGBOOST	eXtreme Gradient BOOST

1. INTRODUCTION

This section introduces the background scope for the study, the research question, the hypothesis, and research objectives used for this study. The data source used will also be stated, followed by a rundown of this thesis and its substances.

1.1 Background

Civil structures, including bridges, dams, and high rises, are becoming vulnerable to losing their premeditated capacities as they debilitate from use. Within the final two decades, a lot of studies have been committed to develop an effective method for finding cracks in buildings. On the other hand, many studies involving deep learning approaches have claimed state-of-the-art performances in a considerable number of tasks. These include, but are not limited to, image classification (Krizhevsky, Sutskever & Hinton, 2017). Many studies which included CNN alongside different classifiers replacing Softmax Classifier have asserted state-of-the-art performances in image classification, and they are the fundamental motivation behind this hybrid machine learning-based crack detection approach which can be applied to evaluate stable structures during building inspection to improve the quality and user acceptance.

This study is to develop a classification model that recognizes cracks and non-cracks (other noises) from photos of concrete structures employing a combination of Convolutional Neural Network with Support Vector Machine and with Extreme Gradient Boosting classifiers.

Traditional manual design feature selection is a cumbersome and time-consuming mission that cannot process the original image, while an automatic extraction method by CNN can detect features directly from the original image (Bernard, Adam & Heutte, 2007). A CNN is a feed-forward network that extracts topological highlights from pictures. It pulls together elements from the original image in the first layer and uses its last layer to classify the array. At the classification stage, the SVM makes the most excellent separation hyperplane in the high dimensional characteristic space. Also, the XGBoost algorithm is one of a standard ensemble classification strategy that provides an effective solution to combine the predictive power of various learners within the classification task.[1] The subsequent is a single model which gives the combined output from several models.

1.2 Research Problem

The utilization of intensity thresholding methods, edge detection based methods and wavelet-transform based methods in crack detection may have difficulty in detecting the full crack curves: they usually detect a set of disjoint crack fragments with many false positives (Zou, Cao, Li, Mao & Wang, 2012). As a result of the non-uniform illuminations and diverse background clutters, the gray values of the crack alter broadly, and the corresponding detection results based on edge analysis may be defective. Moreover, other standard image-based crack detection approaches require handcrafted feature extraction techniques to obtain unique crack features from images. Those methods are perplexing and tedious, unlike the Convolutional Neural Network (CNN). In any case, a drawback of CNN as a classifier is that it finds only a local optimum since it uses the similar backpropagation technique as MLP.

The point of this research is to identify the cracks in concrete structures that influence the solidness of structures and develop and compare hybrid machine learning models which can classify the images into having cracks and non-cracks way better than the existing methods which are right now being used or proposed in the field of structural engineering and computing. Since this research centers on a classification problem, the Receiver Operating Curve (ROC) and the model accuracy and other measures computed from confusion matrix are used to assess the performance of the models developed in the research. Currently, the highest accuracy achieved by conventional image processing techniques and machine learning models already used in other studies is 98.32%. This accuracy was

[1] *https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/*

produced in a recent study by (Li & Zhao, 2019) where Convolutional Neural Network and Exhaustive Search Technique was used to detect cracks in concrete images.

The objective of this study is to assess the performance of the two-hybrid Machine Learning models in crack detection.

To direct the study, the research question has been formalized as: “**Can a hybrid ‘Convolutional Neural Network - eXtreme Gradient Boosting’ model statistically out-perform the hybrid ‘Convolutional Neural Network - Support Vector Machine’ model in classifying concrete images into having and not having cracks?”**

1.3 Research Objectives

The research objective of this study can be outlined from the hypothesis:

H0 (Null Hypothesis): The application of hybrid CNN-XGBoost model in crack detection will result in no performance increase over hybrid CNN-SVM model.

The principal objective is to plan and execute experiments that seek to reject the null hypothesis.

The research is performed as follows:

- Investigate and document the state of the art in detecting cracks and also the current application of hybrid machine learning models in the field of image classification.
- Make a combined dataset by putting together the images from both ‘Positive (Crack)’ and Negative (‘No-Crack’) classes into a single data frame along with their respective labels.
- Automatically extract crack features from the images using CNN architecture and giving the output as an input to two machine learning models, namely SVM and XGBoost, to classify the concrete pictures.
- Evaluate the model for its ability to detect the cracks using various performance metrics.

1.4 Research Methodologies

The focus of this study is Crack Detection (a classification problem) which needs optimal features of the crack to be extracted from an existing concrete image dataset, subsequently falling under secondary research. As part of this current research, a literature review was carried out for the traditional image processing techniques, image classification algorithms, feature extraction techniques, and hybrid machine learning models that were used to identify cracks to get the general idea of the project. This research utilized quantitative approaches for checking which of the research hypotheses is valid. It empirically builds the best model for crack detection based on image classification. Data was generated by combining both the classes and was split into a training and a test set. A comparative study between two hybrid machine learning algorithms for classification problem was carried out using these sets. The two models were assessed using classification accuracy, classification error, precision, sensitivity, and cross-validation scores for ten folds. The ROC characteristic was plotted on a graph for comparison. The results clearly showed the difference in the performance of the two models.

1.5 Scope and Limitations

The scope of this paper is Crack Detection in concrete images where SVM and XGBoost classifiers are applied and compared to decide whether cracks appear in the images, in which images are pre-processed and fed into Convolutional Neural Network to extract the crack features automatically. Additionally, the purpose is to improve the current classification performance in image classification problem. This thesis has the following limitations:

- The developed model is constrained to a binary classification at a fixed level of 128 x 128 pixels.

.• All articulations allude to the data set that’s primarily used for this research and are by no means commonly valid. A significant amount of preprocessing of the raw data must be done for each diverse data set.

- This research is limited to basic CNN model architecture along with hyperparameter variations to SVM and XGBoost models to evaluate the best possible way to identify cracks in the images.

1.6 Organization of the Dissertation

The rest of the thesis is organized as follows:

● **Chapter 2 ("Literature Review")** is dedicated to an exploration of the previous research in crack detection, image classification, inclusive of viewpoints in image processing, hand feature extraction, and machine learning. There is also a run-down of equivalent studies utilizing the CNN extracted features to feed the Machine Learning classifiers, which this researcher finds as a persuading base to the current research work.

● **Chapter 3 ("Design / Methodology")** discusses the selection of the dataset in more detail. Information about the design of this research, steps that are included to carry out the study are stated.

● **Chapter 4 ("Implementation / Results")** conveys a run-down of the convolutional neural network, support vector machine, and extreme gradient boosting implementation and associated results with each of the model.

● Assessment of the research is explained in **Chapter 5 ("Evaluation / Analysis")** which offers a breakdown of the experimental results, the model evaluation and the discussion of the results in light of the research question.

● Chapter 6 **("Conclusions and Future Work")** gives a summary of the research project. This section moreover outlines the future work that may be attempted in this regard.

2. LITERATURE REVIEW

This section gives a review of the literature presented in a few Image Classification problems, crack detection methods, different handcrafted feature extraction methods, application of machine learning and hybrid models in image classification and the evaluation metrics used for assessing those models. The chapter concludes with the gaps in the research, which shapes the objective of the study.

2.1 Image Classification

The Image Classification method has advanced a lot in the last few years from simple edge detection algorithms to pixel-level object detection. Most of the work in image classification is motivated for crack, damage, or object detection in structures. One such work has been done by Lin & Liu, 2010, who found potholes in pavement distress with Support Vector Machine. Later in 2016, fuzzy logic based crack detection in concrete was suggested by Bose Samir Kumar Bandyopadhyay, 2016 in which feature extraction was done by using morphological image processing technique and was fed to fuzzy logic to identify cracks. Even inside the specific problem of image classification, state of the art was achieved over several years of dedicated analysis by hundreds of researchers. It is thus not shocking that in recent years, techniques to automatically discover these architectures have been gaining popularity (Real et al., 2017).

2.2 Traditional IPT and Crack Detection Methods

Identification of cracks is critical because they provide the initial indication of a structure being deteriorated. Earlier methods of crack detection involved inspection by experts where the sketch is prepared manually, and then the depth, shape, and impact of the cracks are analyzed. Image processing was an advancement which reduced the burden involved in the manual inspection.

In research by (Mohan & Poobal, 2018), different works of literature involving crack detection using image processing was done. It was found that the majority of researchers were interested in analyzing the length, width, and the direction of propagation of the cracks by using real-time datasets. The analysis also involved the accuracy, error rate, method of capturing the images using different cameras, and finally, the image processing algorithm used was also discussed.

(Valença, Puente, Júlio, González-Jorge, & Arias-Sánchez, 2017) Focused on crack detection of concrete bridges by combining image processing and laser scanning. The process involves capturing of images by both UAV and terrestrial photography. Orth rectification was considered as a mandate for image processing, and hence, terrestrial laser scanning was used to obtain point clouds from which surface discontinuities were analyzed. A local analysis of the images that provided details of the cracks such as length, width, surface, etc. was later combined with a global analysis to procure a global region of interest. From this region, a 3D model was developed which contained crack pattern data. This combination yielded good results compared to traditional methods; however, the terrestrial photography required access for positioning the various equipment and also a detailed survey of the bridge to be analyzed was necessary to obtain the crack pattern output accurately.

Another model for automatic detection of cracks was proposed by (Wang, Gopalakrishnan, Smadi, & Somani, 2018), which was based on shape-based crack detection. The approach followed in this research was to obtain the pavement images initially, and from these images, potential crack components were extracted. A polynomial curve was developed to fit all the pixels within these components. Finally, a shape metric was developed to distinguish the crack blocks and the background blocks. This study was focused on the fact that the spatial distribution of cracks plays a vital role in determining the types of cracks. The process further involved initial filtering also called as local filtering followed by minor component removal and maximum component extraction, which was done to extract the final potential crack components from the images. Minor component removal was done to remove the non-crack parts, that left only the crack and grooves that are larger. The potential crack components (cracks) with high severity was more significant in size when compared to the gutters, and hence, these were taken and treated as possible crack components. These are then identified and distinguished by using the shape metrics. It was found that this method provided more accuracy and less false alarms.

(Medina, Llamas, Gómez-García-Bermejo, Zalama, & Segarra, 2017) Proposed a way to detect cracks in concrete tunnels. This method utilizes a Gabor filter invariant to rotation as a single crack can exhibit various orientations along its length. Hence a filter which had the same response irrespective of the crack orientation was necessary. This method incorporated proper lighting conditions and linear cameras to capture the images. One of the advantages of using this Gabor filter is that it can detect the cracks in any direction. The parameters that were set for this Gabor filter was done by an algorithm called Differential Evolution optimization method. In the tunnels, the cracks were classified into various types depending on their severity ranging from cracks that were not considered dangerous to cracks that were in the risk of detachment. It was found that this method yielded an accuracy of 95.27%.

The research that was done by (W. Zhang, Zhang, Qi, & Liu, 2014) analyzed the automatic crack detection and classification methodology for subway tunnel safety monitoring with the application of high-speed complementary metal-oxide-semiconductor (CMOS) industrial cameras. The new practical crack detecting and classifying approach proposed by the author was tested on the safety monitoring for Beijing Subway Line 1 and seems to have an excellent performance in detection rate, detection accuracy, and efficiency. An effective way to describe the degree of irregularity of a spatial shape, specifically for crack classification, was also found. The experimental results revealed the rules of parameter settings, which are significant in practical crack detection and classification applications. It was also proved that the proposed approach is effective and efficient for automatic crack detection and classification. A detailed description of the image processing techniques and the optimal parameter settings was given in the paper. The research claimed that the image processing technique for crack detection and the distance-based shape descriptor proposed by the authors might be suitable for other state monitoring and pattern recognition applications. However, it was emphasized that all the experimental results were obtained from images with a resolution of 0.3 mm. Hence it has been advised that the parameter settings may need to be adjusted in other situations.

2.3 Machine Learning Models in Image Classification Problems

2.3.1 Support Vector Machine

SVM, a supervised Machine Learning algorithm for performing regression as well as classification tasks. But in recent times, SVM is majorly used for classification. SVM works by finding a hyperplane which will classify the different classes. The challenge faced here is finding the appropriate hyperplane to differentiate the classes as there can be more than one hyperplane to a particular problem as determined by margins. A margin is a distance from the hyperplane to the support vectors of each class. A support vector is the data point that is closest to the hyperplane. An appropriate hyperplane is the one in which this margin is maximum. It is also particularly easy to use SVM in linearly separable datasets. Since this will be a limitation to use SVM, a method to use SVM's for non-linearly separable datasets was also found by using kernels. The kernels would convert non-linearly separable datasets into linearly separable data and perform the classification. The kernel does this by introducing another dimension such that a non-linearly separable data in one dimension can be linearly separable in a higher dimension. After labeling, using mathematical transformations, the changed decision boundaries can be projected back into original dimensions. It should also be noted that SVM is robust to outliers. SVM's performance also largely depends on the parameters given while the model is built. It is highly essential to carefully determine both C and Gamma parameters while designing the model. SVM is also referred to as the black box because of the unpredictable nature of the data once it is transformed by the kernels.

Some of the advantages of SVM is that it is highly memory efficient. It also performs reasonably well when the number of dimensions is higher. It is because of these advantages that SVM is mainly preferred for image classification. However, a con of SVM is that it does take a lot of time for training if the dataset is big.

SVM is used as a binary classifier, but it also used to classify more than three classes at times. One such study where SVM was used for image classification was proposed by (Srunitha & Padmavathi, 2017). This study involved the classification of 7 different types of soil by incorporating SVM. The study took soil characteristics identification and classification very important as it would reduce agricultural product quantity loss. The process involved image collection also called as data acquisition, image pre-processing, feature extraction, and classification of those images. The features of the soil images were extracted using a low pass filter and Gabor filter. Statistical parameters like mean, standard deviation, and histogram were also considered. The process involved four steps starting with an SVM followed by segmentation, transformation, and then the statistical parameters. The working of the SVM was the same as discussed above, and the segmentation involved splitting of the region of interest from noise. The transformation step included three techniques, namely color quantization, usage of a low-pass filter, and utilization of a Gabor filter. The quantization was done to represent the new image similar to that of the old image. The low pass filter passed only low frequencies through it and attenuated high ones. The Gabor filter was used as an edge detector to extract features. Mean, and the standard deviation was the statistical parameters considered. After all the above steps, the Euclidian distance was calculated, and the soil images were classified. The dataset consisted of 175 pictures of diverse soil data. The results of this study showed that SVM performed better in classifying two classes namely sandy and non-sandy soils, whereas its performance was not satisfactory in predicting three categories and also seven classes which involved seven soil types. Three class classification did not achieve the required accuracy, mainly because of the high overlapping of clayey soil with the other two categories. The 7-class classifier did not yield the expected accuracy as there were no optimization methods used in the built SVM.

(Liang, Jianchun, & Xun, 2018) Proposed a plan for crack detection in civil structures which was based on machine vision. It involved both the extraction of crack images in concrete structures and also classifying them by the use of Support Vector Machines. In this method, the adaptive non-linear grayscale transformation was used initially to expand the gray difference that existed between the crack image and the background image. After this, the cracks were extracted by using an improved OTSU algorithm which employs threshold segmentation. Once the cracks were obtained the fracture points that exist in the cracks was connected by a combination of the extensive fracture skeleton crack line and the gray features that exist in the crack edge. Once this was done, a complete image of the crack was obtained. The process which followed this was the extraction of classification characteristics (features) of the cracks, which included peak ratio of the gray histogram, distribution ratio, and the mean square deviation of the gray histogram. These features were used as input for the training of an SVM classifier which performed the crack classification and gave a result as to whether the image belongs to a crack class or not. It was interesting to note that the OTSU algorithm used in this research was capable of handling a maximum between-class variance method that enables them to do both extraction and classification. The features used in this research played an important role as they were fed as inputs to the SVM. Although the features were extracted manually in this research, better feature extraction methods using CNN and ELM also exist.

While implementing the SVM algorithm, it becomes highly essential to parameterize the SVM properly. The two most important parameters include kernel function and the penalty factor. SVM’s often suffer from the problem of overfitting when there are very few data points. This problem can be overcome by the proper selection of the penalty factor. The other parameter kernel function represents a degree of correlation between the support vectors. Hence in this research, a grid-search cross-validation method was used to optimize the above-given parameters. The type of SVM used in this research was C-SVC, and the type of kernel used was RBF kernel.

2.3.2 Extreme Gradient Boosting

(Chen & He, 2015) Proposed an algorithm using XGBoost for approximate tree learning of sparse data with theoretical justification. This research provides deep insights about data compression and sharding for building a scalable tree boosting system. Highly scalable end-to-end tree boosting system was designed, and a weighted quantile sketch was also developed for proposal calculation and evaluated. The sparsity-aware algorithm was introduced for parallel tree learning, and a cache-aware block structure was proposed for out-of-core tree learning, and it claims that this end-to-end system can be used to solve real-world use cases. Varieties of datasets have been used for this research with single machine parallel setting and distributed & out-of-core settings. It was stated that although the basic exact greedy algorithm is mighty, it is inefficient when the data doesn’t fit into memory entirely or in a distributed setting and an approximate algorithm is summarised for overcoming the shortcomings and support efficient gradient boosting. Storage of data in in-memory units called Blocks was proposed by storing in Compressed Column (CSC) format where columns were sorted by corresponding feature value to reduce the time consumption during sorting of data. The results indicated that XGBoost and scikit-learn produced better results in comparison with R’s GBM and also XGBoost runs 10x faster than sci-kit-learn, which fails to handle non-sparse input. This research proves the efficiency of XGBoost by building a scalable tree boosting system and establishes the capability of XGBoost in solving with limited resources.

(Ayumi, 2017) Studied the performance of XGBoost in action recognition and compared with SVM (Support Vector Machines) and Naïve Bayes in terms of classification accuracy. The datasets used for this analysis were from Kinect database that consisted of 594 sequences of human skeletal body movements for 12 different gestures and another dataset containing ten different types of human action in an indoor setting was also used. 10-fold cross-validation was performed, and XGBoost was implemented along with SVM and Naïve Bayes algorithms. The performance was compared based on accuracy, computational time, and F1 score. Also, the confusion matrix was plotted, where XGBoost was more balanced and robust in predicting the classes. It was concluded that XGBoost performs well on all the datasets used for the study based on the outcomes. However, computational time was more when using XGBoost. This research elaborates the efficiency of XGBoost in Image classification. However, the performance of XGBoost technique in very high-Resolution images is to be explored which was done in the below study.

(Georganos et al., 2018) Investigated the sensitivity of XGBoost to various sample sizes of high-resolution images in object-based land classification. Correlation-based Feature Selection technique was used in this research, and XGBoost classifier was compared with Random Forest and SVM (Support Vector Machines) models. High-resolution images of three cities, Ouagadougou, Dakar, and Vaihingen, were used for analysis. All these cities contain planned and unplanned residential buildings, commercial structures, etc. Image segmentation was done in Python environment. Correlation-based Feature Selection was done for extracting the features as it provides high computational efficiency in CART classifiers. The Bayesian Optimisation procedure was done for optimizing the parameters of XGBoost classifier. The results indicate XGBoost performed better than RF & SVM; however, it became computationally inefficient with a higher number of trees and recommended a lower number of iterations for optimal results using XGBoost.

These researches establish the efficiency of XGBoost classifier in image classification and provide insights on the pre-processing of data to obtain maximum results. It is to be noted that XGBoost has high computational time, which can be improvised by appropriate Feature Selection.

2.4 Application of CNN’s in Crack Detection / Image Classification

Deep learning model, namely, Convolutional Neural Networks (CNN), offered means to overcome the limitations in crack detection based on image processing. Precisely, CNN has successfully been applied to image classification, while featuring a significant level of abstraction (generalization) and learning capabilities. These features are a key to detecting damages such as cracks in concrete in a reliable manner; modern CNN based automatic crack detection system under development for pavements is proof to that. The basis for CNN development relies on transfer learning. Considering the analysis carried out in the research by (Maeda, Sekimoto, Seto, Kashiyama, & Omata, 2018), the use of the transfer-learning demonstrated the potential to train a model with limited data. The authors developed a CNN model that was limited to a binary classification at a patch level of 256 × 256 pixels, and the accuracy and runtime speed on a GPU server and a smartphone were measured. The research demonstrated that the type of damage could be classified into eight with high accuracy by applying the proposed object detection method. This research proved that although many models have been submitted for image classification, the Convolutional Neural Networks (CNN) outperforms others for higher accurate predictions.

A model proposed by (Bhaskar, n.d.) Explores a similar approach that used CNN to predict the probability that an image uploaded in Instagram will get more likes. The authors had modified a pre-trained AlexNet ImageNet CNN model using Caffe on a set of pictures uploaded by users. For the dataset, the images uploaded with 'hashtag me' was downloaded with the required information from Instagram through the available API. Authors had considered taking the ratio of no. of likes an image has to the no. of followers to train the CNN model.

The dataset was normalized by finding the Median and the images without user's face were removed using CCV (a modern computer vision library) face recognition algorithm was used to detect human face/eye. The data was divided into two sets and assigned labels 1 (for popular images with above 50 percentile ratio) and 0 (for rest) for each group by using the ratio of likes to followers. The AlexNet ImageNet CNN model was applied to the dataset and extracted FC6, and FC7 activation features each of 4096 dimensions. Authors had used Linear SVM Classifier, Random Forest Classifier, and then Mlib's – SVM, and cross-validated the results for the optimum regularization parameter and step size. After finalizing it, the Alexnet CNN model was fine-tuned with Caffe to the data set. Authors had considered an additional benefit of using pre-trained networks that were trained on a broad collection of images so that the intermediate layers captured the 'semantics' of general visual appearance. The dataset was divided into the ratio 3:1 to train and test. The model was trained for 35000 iterations with a starting learning rate of 0.001 and reducing it by a factor of 0.1 every 5000 iterations to decrease learning rate after loss stagnates after many iterations. With the input of 7.5 GB, still, the RAM used didn't cross 12 GB, proving to be resource-efficient. After training the model, the test images were passed, and the probable top 25 most and least popular photos were considered. The model predicted the most famous images as Caucasian women, with either close-up shots or pictures taken in conventionally considered as beautiful background (like famous landmarks or in the natural origin). The bad images were mostly memes which were not considered to be that funny, and also had three pictures of a single person for unknown reasons to the authors. The results have shown that even with noisy data and lesser accuracies, the model worked.

In Image Classification, when the dataset is large, the variables required to describe data is also significant. More complex the data is, the higher computation power, and memory is needed to process the data. There is also a possibility of having redundant information in the data set. It could also cause a classification algorithm to overfit to training samples. Thus, by using feature extraction, the complicated and substantial data sets can be reduced to non-redundant informative features which facilitates subsequent learning and better generalization steps.

In a paper exploring the road crack detection using Deep Convolutional Neural Networks (CNN), the authors (L. Zhang, Yang, Daniel Zhang, & Zhu, 2016), have outlined the benefits of using CNN when compared to hand-craft methods that have been used till now. The authors have used a dataset of 500 images clicked on a low-cost smartphone, thus avoiding any specially designed or high-cost optical types of equipment to gather the data. The authors have primarily compared various popular traditional methods used for road crack detection and the constraints of it, which were overcome by using CNN.

Some of the critical issues in hand-craft methods are that they are not discriminative enough to differentiate crack and complex background in low-level image cues (ex, in shadows). Since the number of images with cracks were lesser than the rest, to train with a vast dataset for better predictability, the images were rotated by a random angle and thus, 640000 samples were used as a training set. The images were analyzed considering a patch whose center is within 5 pixels of crack centroid regarded as positive patch else negative patch. All convolutional filter kernel elements were trained from the data in a supervised fashion. In each convolutional layer, the ConvNet performs max-pooling operations to summarize feature responses across neighboring pixels, thereby allowing it to learn features that do not change concerning the location of objects in the images. Finally, fully-connected layers were used for classification. The dropout method was used between two fully connected layers to reduce overfitting by preventing complex co-adaptations on training data. The ConvNet was constructed using the Caffe framework and trained by using fivefold cross-validation.

The proposed method was compared against the Support Vector Machine (SVM) trained with LIBSVM and the Boosting technique via the OpenCV toolkit. The results clearly show that CNN had detected the cracks in the images better than the other hand-craft methods. The probability maps show that the higher brightness (greener lines) represents, the higher confidence the model describes the cracks. Thus, CNN represented the cracks with the highest intensity and accuracy compared to the other two methods.

2.5 Use of Hybrid Models in Image Classification

Although deep learning models like CNN, ANN performs well on the classification of images, these technologies have specific limitations such as overfitting of data and an increase in use cost. Since the number of parameters in the fully connected layer accounts for almost 80% of the total number of model parameters, this results in a considerable increase in training and thereby leads to the mentioned shortcomings. To overcome this limitation (Jiang, Zhao, Wu & Tan, 2018) presented a framework for HRRS images of scene classification, using XGBoost classifier instead of Softmax layer.

Similarly, Notley & Magdon-Ismail, 2018 has been working with image and numeric data, where he used CNN for extracting features and used the extracted features as inputs for another machine learning models, namely Support Vector Machines (SVMs) and K-Nearest Neighbor classifiers (KNNs), in order to see if neural-network-extracted features enhance the capabilities of these models on 4 images and 3 numeric data. CNN extracted features being fed to other classifiers gained popularity when (Ren, Guo, Li, Wang & Li, 2017) proposed a novel CNN-XGBoost model and implemented it on the well-known MNIST and CIFAR-10 databases. According to Ren et al., this new method performed better compared with other processes on the same databases, which verifies the effectiveness of the proposed method in image classification problem.

Classical Machine Learning algorithms require image features as input for Image Classification, and those features can be effectively extracted by using CNN as mentioned in above researches. The research by (Copur, Melisozyildirim, & Ibrikci, 2018) is also one such example of having CNN as feature extractor and SVM as classifier. In this research, aerial images were classified based on the presence of vehicles using CNN for learning the features and SVM for classification. The VEDAI dataset was used in this research since it contains aerial images with different types of vehicles and backgrounds. From each image, small parts containing vehicles were extracted and used as a positive sample, and random parts of the image which do not hold vehicles were used as negative samples. Data augmentation was done by adding rotated and sharpened images of original samples to obtain more training data. In total, over 9000 positive and negative samples each were used for training, and around 1500 positive and negative samples each were used for testing. In the training phase, after the CNN was trained with the training data, its last fully connected layer was removed, and the output of the previous segment was fed to the SVM for training. In the testing phase, CNN was used to extract the features, and SVM was used for classification. The results of this research indicate that combining CNN and SVM provides better classification accuracy than using only CNN for both extraction and classification. It also outperforms traditional approaches with feature extracts such as Histogram of Oriented Gradients and SVM as classifier. The authors argued that CNN is a useful feature extractor as it tunes the kernel based on the training data, and SVM is better at classification than a traditional feed-forward neural network. So, combining these two techniques resulted in better classification. However, since both CNN and SVM had to be trained, this method was computationally expensive and took longer to train. Thus, hybrid models have shown promising results at analyzing and classifying datasets with increased accuracy and performance.

Another research was done by (Chaiyasarn et al., 2018), which made use of a hybrid model CNN-SVM to detect and classify cracks in concrete structures. Digital camera images of concrete cracks from various locations acted as their primary dataset for both classification and validation in the proposed research. These RGB images were divided into image patches with the help of Adobe® Photoshop software package and labeled as either 1 or 0 depicting the presence of a crack or non-cracks, respectively. The raw RGB images were divided into training, testing, and validation data. They act as the input to the CNN model, which makes use of fully connected layers to extract in-depth convolutional features. These features were utilized for training the classifier and the weights adjusted using the backpropagation algorithm. The model used an SVM classifier in place of a Softmax classifier so that it provides the probability scores along with the output class labels, i.e., whether a crack is present or not. These probability scores were used to create the ROC curve, which was used for evaluation purpose. Results show that the combined CNN-SVM model has outperformed the CNN model with an accuracy of 90.76%. Since such hybrid models have become the dominant method for feature extraction and classification of image-based classification problems, our research follows a similar approach.

2.6 Summary

2.6.1 Summary of Literature

This chapter has reviewed the existing literature relevant to the research. Notably, it highlighted many operational techniques that have to be considered, namely image processing, feature extraction, and pertinent classifiers of the machine and deep learning. These factors should be addressed to achieve the highest crack detection accuracy. To sum up, an ample but not thorough list of studies which used machine learning or deep learning techniques in correspondence with feature extraction approaches in damage/crack detection are described in Table 2.1.

Work Reference	**Application Area**	**Feature extraction/ Image Segmentation**	**Classifier**
(Lin & Liu, 2010)	Pavement Pothole Detection	Partial Differential Equations (PDE) Model	Support Vector Machine (SVM)
(W. Zhang et al., 2014)	Subway Tunnel-Crack detection	Morphological Image Processing Techniques and Thresholding Operations.	ELM, RBF, SVM, KNN
(Bose Samir Kumar Bandyopadhyay, 2016)	Concrete Crack Detection	Morphological Image Processing Technique	Fuzzy Logic
(L. Zhang et al., 2016)	Road Crack Detection	Convolutional Neural Network (CNN)	
(Yokoyama & Matsumoto, 2017)	Concrete Crack Detection		
(Cha, Choi & Büyüköztürk, 2017)			

(Dorafshan, Thomas & Maguire, 2018)	Concrete Crack Detection	Convolutional Neural Network (CNN)	
(Silva & Lucena, 2018)			
(Maeda, Sekimoto, Seto, Kashiyama & Omata, 2018)	Road Damage Detection		
(Liang et al., 2018)	Concrete Crack Detection	OTSU Threshold Segmentation	Support Vector Machine (SVM)
(Chaiyasarn et al., 2018)	Masonry Structures- Crack Detection	Convolutional Neural Network (CNN)	
(Sharma, Anotaipaiboon, & Chaiyasarn, 2018)	Concrete Crack Detection		

Table 2.1- provides a list of studies which used machine learning or deep learning techniques in correspondence with feature extraction approaches in damage/crack detection.

2.6.2 Gaps in Research

From the analysis by (Cha, Choi, & Büyüköztürk, 2017), the Sobel and Canny edge detection strategies used for image classification failed to give any meaningful crack information. Though Conventional machine learning classifiers remained powerful and robust, they needed to leverage the power of Deep Learning. CNN has been recognized as the most powerful and effective mechanism for feature extraction, yet traditional classifiers associated with CNN did not fully understand the extracted features as per (Ren, Guo, Li, Wang, & Li, 2017) thereby creating scope for optimization by using hybrid machine learning models for image classification. Several studies have used such kinds of hybrid models for crack detection. But XGBoost was never used as a classifier in the particular database used for this research and also in crack detection despite its achievements in other image classification problems. Also, an assessment on the level to which hybrid models can outperform each other in automatically detecting cracks in a concrete structure has not been discussed.

2.6.3 The Research Question

The downsides and research gaps presented in this section can be addressed by the research question given as:

"Can a hybrid 'Convolutional Neural Network- eXtreme Gradient Boosting' model statistically out-perform the hybrid 'Convolutional Neural Network- Support Vector Machine' model in classifying concrete images into having and not having cracks?"

The following sections will contain the research design, execution of models, and assessment of the models to address the research question.

3. DESIGN / METHODOLOGY

3.1 Introduction

The following part will portray the information that's utilized to fulfill the research and experimentation directed by the principal research question. In addition to that, this chapter clarifies the subtleties of data treatment -- this is a significant consideration as it impacts the features to be extracted, which in turn influences the classification accuracy.

The details of the model development, the tools employed to evaluate model performance and the confinements and strengths of the design will be provided at the end of this section.

Figure 3.1 offers a general idea of the subsequent sections, each outlining the design, methodologies, and considerations significant to the implementation of this research endeavor.

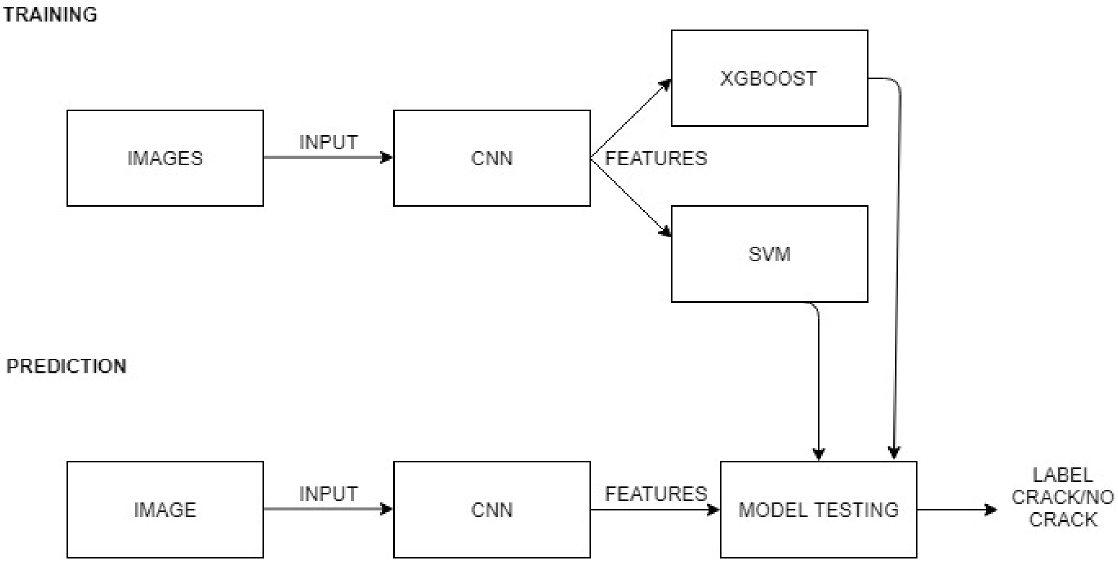


Figure 3.1 provides the project workflow diagram. This diagram highlights the different phases in experimental design used throughout

The thesis takes after the CRISP-DM methodology, and each of the stages of it is described in detail underneath.

3.2 Problem Definition

The study focusses on classifying concrete images into having ‘cracks’ and ‘no cracks,’ which was accomplished by training a Deep Learning-based Convolutional Neural Network model on the images collected from a digital camera. The features extricated by CNN were then fed to Classical Support Vector Machine and Extreme Gradient Boosting model to be compared. The performance of the hybrid models was evaluated based on classification accuracy, classification error, cross-validation scores, recall, precision, and ROC-AUC Curve.

3.3 Data Understanding

3.3.1 Dataset

The dataset consists of concrete images which were collected from walls and floors of several buildings in METU Campus from approximately 1 meter away from the surface and camera facing directly to the target surface various. The dataset was generated from 458 high-resolution images (4032x3024 pixel) with the method proposed by Zhang et al. (2016) and was divided into two folders as negative(‘no crack’) and positive(‘crack’) for image classification. Each class has 20000 images with a total of 40000 images. The photos are of size 227 x 227 pixels with RGB channels. The input images are high-quality fragments, and the RGB values are used as features in input vectors to the CNN. The dataset has variance in terms of surface finishes, e.g., exposed concrete, plastering, and paint. This dataset is publicly available on the Mendeley site. (Özgenel, 2018).

3.3.2 Data Pre-processing

Initially, raw image data may have diverse issues such as distortion or skewing and so can likely not deliver optimal results in image classification. That is why careful consideration of image preprocessing is vital.

Moreover, due to the utilization of Convolutional neural networks with tensor flow backend for feature extraction in this research, the images being fed to the system will be required of a fixed size and shape[2]. For this reason, before the feature extraction, the images were preprocessed to the size and shape which the network needed. With the fixed-sized image, benefits of handling them in batches can be obtained.

Having a differently scaled object of interest in the images is the supreme facet of image diversity. When the network is in the hands of real users, the object in the image can be small or big. Likewise, at times, the object can cover the whole image and yet will not exist entirely in the image (i.e., cropped at boundaries). So, this research emphasized on merging the images from both the classes, converting them into a grayscale format, resizing and appending them with corresponding labels, reshaping and scaling them in the preprocessing stage.

3.3.3 Data Encoding

Classifiers used in this research cannot work with categorical data directly. They assume that the variables used are numeric. For this reason, the categorical variables in the data have to be converted to numeric type before feeding them to the classifiers. The class labels were converted to the numerical values ‘1’ and ‘0’ for cracks and no-cracks, respectively. 60% of the data was used to train the model, and the remaining 40% was used for testing the model’s performance.

3.4 Modeling

The research aims at building and comparing two hybrid models to predict the condition of the concrete images. Python programming language and Tensor flow library will be used to implement the models.

3.4.1 Feature Extractor

The primary stage uses the CNN model to extract features from the concrete images. The architecture of feature extractor has 3 Convolutional layers, the first with 16 filters and the other two with 32 nodes, each one followed by Max Pooling

[2] *https://medium.com/ymedialabs-innovation/data-augmentation-techniques-in-cnn-using-tensorflow-371ae43d5be9*

and Dropout layers. Then, it has a Flatten layer followed by a Fully Connected layer with 64 nodes and finally the Output layer.

A convolutional layer acquires a feature map by calculating the dot product between the receptive field and kernel. Over-all, an activation function is added behind each convolutional layer, such as the sigmoid function, Rectified Linear Unit (ReLU). This part uses the ReLU (Rectified Linear Unit) for all the layers, except for the output layer where the sigmoid function was used. Figure 3.2 shows the plot of ReLU function, the formula for which is given below[3]:

$$F(x) = \max(0, x)$$

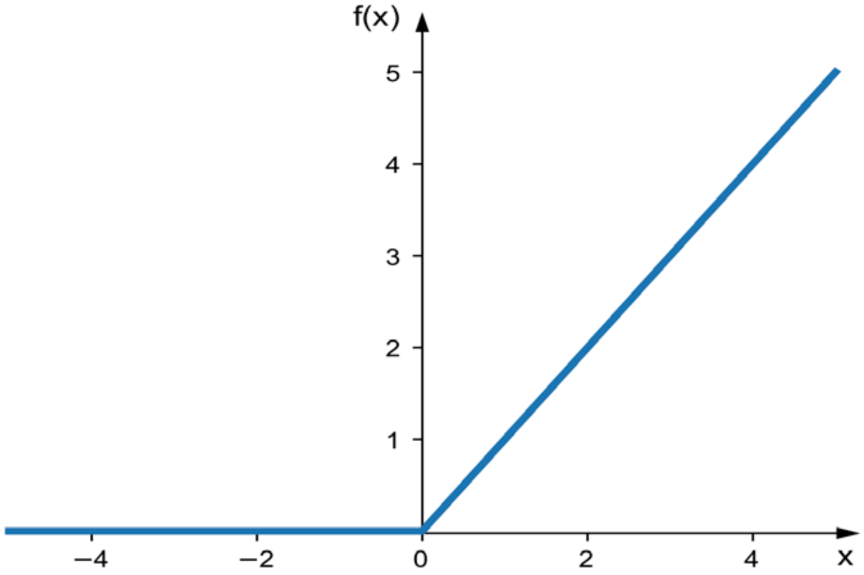


Figure 3.2- Plot of Rectified Linear Unit Function

The plot and formula for the sigmoid function are depicted in Figure 3.3.

Pooling layers are utilized to downsample the image feature maps. There are two broadly used pooling layers, the average pooling layer, and the maximum pooling layer. The max-pooling layers used in this model will yield the max value from each sub-area, and the pictures are down-sampled by max-pooling layers, causing 1/2 lessening in each image’s height and weight.

[3] *https://sebastianraschka.com/faq/docs/relu-derivative.html*

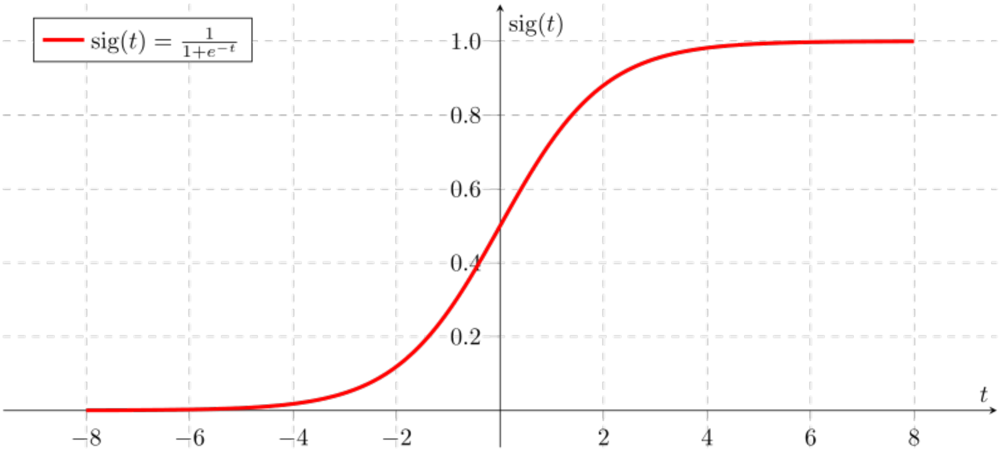


Figure 3.3 Sigmoid Function of CNN

Then, the trained CNN is applied to the data to extract the features of concrete images in a better way. The global characteristics of concrete images are obtained through this stage.

3.4.2 Classification

(a) CNN- Support Vector Machine

Support Vector Machine has been extensively used in various fields to achieve state-of-the-art results on a lot of experiments. The idea behind an SVM is to find an ideal linear hyperplane decision boundary such that the anticipated classification error for testing data is minimalized. An SVM finds a hyperplane that splits the most substantial portion of a categorized data set for binary classification, the training data is a set of training sample pairs $\{(x_1, y_1),\ldots, (x_i, y_i)\}$ where x_i is the observation for the ith sample and $y_i \in \{1,0\}$ is the associated target label. The SVM classifier is a discriminant function mapping an input vector space x into a class label.

$$F(x) = (w.x) + b,$$

Where w is the weight of the linear decision boundary, and b is the bias added, which maximizes a margin between each class. This SVM classifier is added to the fully connected layer of CNN to produce outcomes for image classification in the research, as shown in Figure 3.1. A Grid search CV was employed to attain the optimal parameter values for SVM. Different values of C and gamma have been experimented, and the values that provided the best accuracy was chosen finally to test the model on unseen data.

(b) CNN- Extreme Gradient Boosting (XGBoost)

The XGBoost algorithm was implemented using XGBoost library in Python. The tree model is usually used as a primary classifier in XGBoost System. The features extracted from CNN (Fully Connected Layer) was fed to train and test the XGBoost classifier in this study. The optimal parameters for XGBoost were also found using Grid Search. The objective function of the model can be defined as:

$$\mathrm{Obj}(\Theta)=L(\theta)+\Omega(\Theta)$$

Where L is loss function and Ω is the regularization term which most algorithms fail to include in the objective function. But, XGBoost consists of regularization, consequently controlling the intricacy of the model and avoiding overfitting. The final classifier was obtained by optimizing this function.[4]

Finally, a model with the best test accuracy and AUC Score is selected from the experiments and is used for hypothesis evaluation and deployment.

3.5 Evaluation

This research uses the Accuracy, ten-fold cross-validation scores, classification error, AUC value, and other performance metrics calculated from the confusion matrix to evaluate the performances of the models. Confusion matrix conveys the volume of instances that are correctly classified and misclassified.

	Actual Positive	**Actual Negative**
Predicted Positive	True Positive	False Positive
Predicted Negative	False Negative	True Negative

Table 3.1 Sample confusion matrix for a binary classification problem.

Table 3.1 provides the components of a fundamental 2- class confusion matrix,

Where **TP (True Positive)** denotes the positive cases that are correctly classified as positive,

[4] *https://www.kdnuggets.com/2017/10/xgboost-concise-technical-overview.html*

TN (True Negative) is the negative instances that are correctly classified,

FP (False Positive) represents the negative cases that are incorrectly classified as positive,

FN (False Negative) denotes the positive instances that are incorrectly classified as negative.

3.5.1 Accuracy

It is the fraction of the right predictions to the total prediction in the test data:

$$Accuracy = Correct\ Prediction / Total\ Prediction$$

It's a good measure for this research as the data is symmetric and the classes are not imbalanced.

3.5.2 Classification error

The classification error of the system can be expected from the confusion matrix as follows:

$$Error = (FP + FN) / (FN + TN + TP + FP)$$

3.5.3 Cross-validation

K fold cross-validation (k =10) was also used to assess the models for over-fitting by dividing the original sample into 10 equal-sized subsamples. Among the 10 subsamples, a single subsample was reserved as a validation data for testing models performance to guarantee that the model is capable of generalizing to new data and the other nine subsamples were combined into a training set. The cross-validation procedure was then repeated 10 times, with each of the 10 subsamples adopted precisely once as the validation set. The cross-validation scores for each fold were then obtained.

3.5.4 AUROC

AUROC Curve can be relied upon when it comes to a classification problem. It conveys how much model is proficient in distinguishing the classes. Higher AUC value depicts that the model is better at classifying 0s as 0s and 1s as 1s[5]. The ROC curve was plotted with True Positive Rate on the y-axis against the False Positive Rate on the x-axis for this research for both the models and compared.

3.6 Strengths and Limitations

This section summaries the strength and restrictions of the design and methodology used in the study. Tenfold cross-validation was performed to help attain shorter computing time and avoid overfitting. Another significant advantage of the research is that the features were treated as a black box as it was automatically extracted by CNN that are relevant to the prediction of cracks.

The prime limitation of the research is the similar illumination condition and clear evident patterns of the data which make the model biased to such terms.

3.7 Summary

This section gave an overview of the CNN-SVM and CNN-XGBoost design specifications and provided a short explanation of the data and its source. A brief outline was given to both the models which will be extended upon in the following part. The next chapter will assess how these methods were implemented and the results of the model designs that have been applied.

[5] *https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5*

4. IMPLEMENTATION / RESULTS

4.1 Software

The research was conducted using Python. The project depends profoundly upon Numpy, Pandas, and Scikit Learn, three very frequently used open-source libraries intended for machine learning and data analysis. The Convolutional Neural Network used Keras with tensor flow backend, another open-source library for Python built as an extension to Theano implementation. The results were then analyzed by visualizations created using Microsoft Excel and Matplotlib package of Python.

4.2 Data Collection

The dataset used for this research consists of a set of images that were created by taking pictures of different concrete structures of METU campus buildings with the use of a digital camera. The dataset has features in the form of an image and corresponding labels to indicate whether it has a crack on it or not. The photos are put up into two separate folders named ‘Positive’ and ‘Negative’ by the contributor. The data from both the folders were appended and shuffled to maintain distribution of the data when splitting into test and training dataset. A numeric label was used to denote the present condition of the images, consisting of values 1 and 0, which represents whether the image has a crack on it or not, respectively. Figure 4.1 provides a few pictures from both the classes.

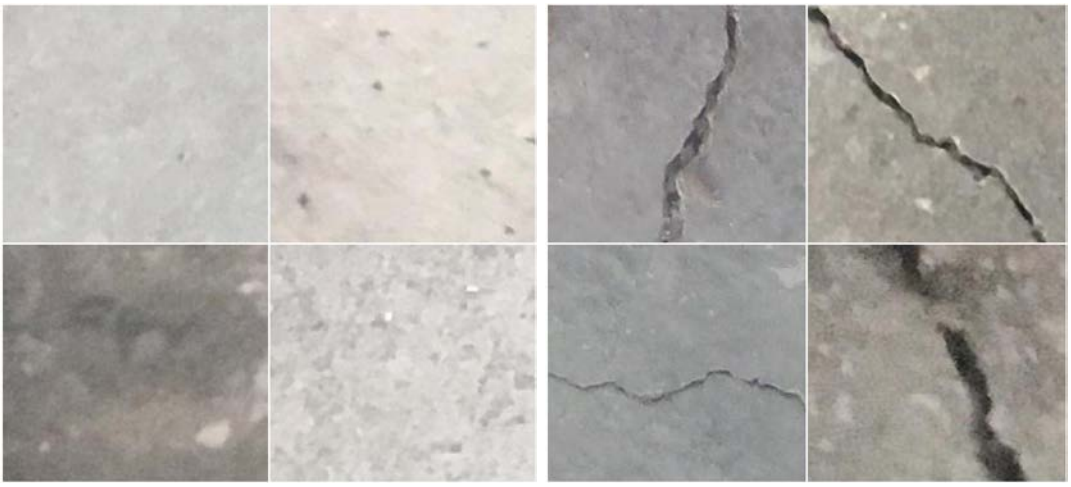


Figure 4.1 Images on the left are from Negative class and on the right are from Positive class

4.3 Data Preprocessing

It is crucial to preprocess the data to use them as input in the CNN model, which includes resizing, reshaping and scaling of images, followed by splitting of the dataset.

Images are required to be in similar dimension as filters in convolutional neural networks are used to extract features from them by sliding over. Due to this reason, all images were loaded in gray-scale format, resized to 128 x 128, as described in Figure 4.2 below.

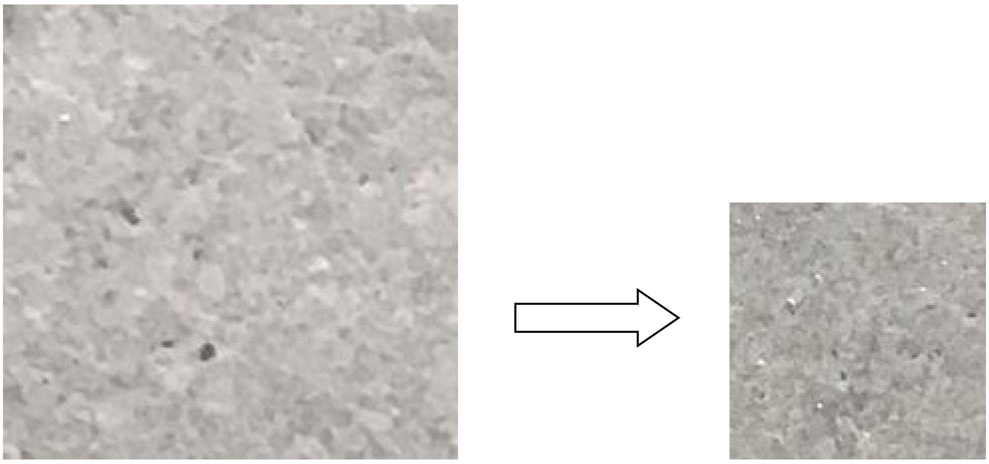


Figure 4.2 Original raw image re-sized to 128x128 pixels to feed into NN

Finally, before images were fed into NN, all images were scaled by 255and reshaped into Tensor Flow format (nº images, width, height, channels),

Where nº is the total number of images- X data,

Width * height = 128 x 128,

Channel = 1.

4.4 Saving and Loading Data

There are various means to save and load the data, models to make predictions in Python using Scikit-learn. In this experiment, Joblib, a part of the SciPy system that offers utilities for pipelining Python jobs was used for saving the pre-processed data into the drive and load them again while modeling to reduce the runtime.

4.5 Data Sampling

In the experiment, a basic sampling strategy has been used initially, as 40% of data was considered as test data; remaining data has been taken for training. In later implementations, the count of instances varied as per the technique used.

Tenfold cross-validation method was used in the research to evaluate the performance of both models other than the test set. In each iteration, nine folds of data were used for training, and one fold for validation and the average, individual scores for each fold were obtained and compared with the test accuracy to check if the model was generalized and not over-fitted.

4.6 Data Modeling

In the proposed crack detection system, Convolutional Neural Network (CNN) was applied on training data with a cross-validation technique and the test and training features were extracted from its dense layer and fed to the classical ML algorithms namely SVM and XGBoost. The architecture of hybrid models used in this paper is shown in Figure 4.3.

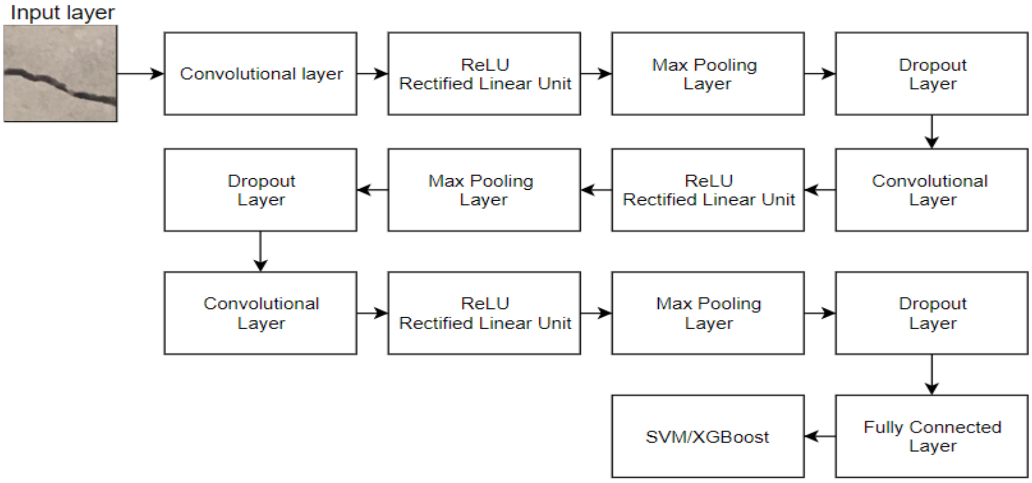


Figure 4.3 Architecture of Hybrid models used in this research (CNN-SVM & CNN- XGBoost)

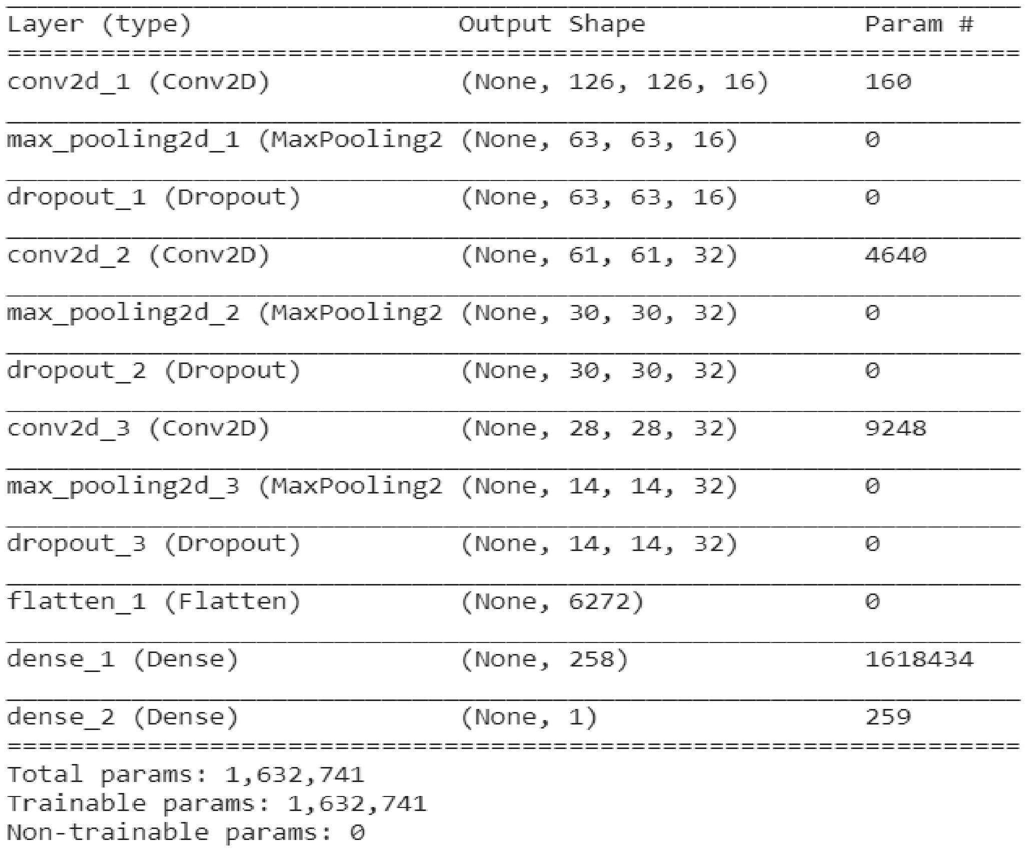
The batch size for each iteration in training was 64, the number of epochs was 3(after which the validation loss started to increase), and the validation split was 20 %. Dropout layer and this validation split method were used to avoid overfitting.

The image features were then fed to the dense layer with the dropout of 0.30. The optimizer used was Adam, and the cost or error loss has been calculated using binary cross-entropy since it’s a binary classification problem. Figure 4.4 provides a summary of the model created and the number of trainable parameters obtained from CNN.

The program was implemented on a PC with 2.50GHz i5-7300HQ CPU, 8GB memory and an NVIDIA GEFORCE GTX GPU for acceleration.

The features extracted from the dense layer were applied on testing and training dataset to obtain testing and training features separately. These features were then respectively applied for training and testing the Machine Learning Classifiers – SVM and XGBoost.

For the selection of parameters in the ML classifiers, Grid Search CV was used. It works by fitting the models on a dataset and evaluating all the possible combinations of parameter values that were specified in the parameter grid and retaining the best combination for tweaking the models again to achieve optimal performance.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 126, 126, 16)      160
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 63, 63, 16)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 63, 63, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 61, 61, 32)        4640
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 30, 30, 32)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 30, 30, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 28, 28, 32)        9248
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
dropout_3 (Dropout)          (None, 14, 14, 32)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 6272)              0
_________________________________________________________________
dense_1 (Dense)              (None, 258)               1618434
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 259
=================================================================
Total params: 1,632,741
Trainable params: 1,632,741
Non-trainable params: 0
```

Figure 4.4 Summary of CNN model built to extract features

Below are the parameters' value specified for selecting best hyperparameters using Grid search CV for SVM and XGBoost Model.

SVM = {'C': [1, 10, 100, 1000], 'kernel': [linear, rbf]}

XGBoost = {'min_child_weight': [1, 5, 10], 'gamma': [0.5, 1, 1.5, 2, 5], 'subsample': [0.6, 0.8, 1.0], 'colsample_bytree': [0.6, 0.8, 1.0], 'max_depth': [3, 4, 5]}[6]. The best hyper parameters obtained for both the models finally is described in the Table 4.1 below.

Model	CNN-SVM	CNN-xgboost
Parameters	C: 1 Kernel: RBF	Subsample: 0.8 Min_child_weight: 1 Max_depth: 5 Gamma: 0.5 Colsample_bytree: 0.8
Best Score	0.98975768	0.997

Table 4.1 Best Parameters obtained through Grid Search CV for CNN-SVM and CNN-XGBoost models

24018 images were used to train the model, and 16012 images were used to test the classification accuracy. The training and testing accuracy of the three models are mentioned in Table 4.2 below.

Model	CNN-SVM	CNN-XGBoost
Training Accuracy	98.8%	99.25%
Testing Accuracy	98.76%	98.79%

Table 4.2 Training and Testing Accuracy of CNN-SVM and CNN-XGBoost models

4.7 Summary

This section has reexamined the datasets adopted in this research at the start, and the resizing, reshaping, scaling methods in the preprocessing stage have been described. Then, the implementation particulars of the two models- CNN-SVM and CNN-XGBoost, such as the software used, selection of the packages, and the tuning

[6] *https://stackoverflow.com/questions/51671058/different-roc-auc-with-xgboost-gridsearch-scoring-roc-auc-and-roc-auc-score*

of the parameters have been considered. Finally, the outcome of the models was compared, and the research question was answered. As can be seen from Table 4.2, the joint performances of the CNN with XGBoost model is higher than with SVM in crack detection. Notably, the combination of CNN and XGBoost reached the highest efficiency of 99.25% during training. The following part will be having an analysis of the experimental results, as depicted in Table 4.2. Additionally, model evaluation and assessment will also be shown in detail. To conclude, comparison and discussion will be provided in line with the literature review; the novelty of this research will also be stated.

5. EVALUATION/ANALYSIS

This section evaluates the outcomes obtained from the experiments done in the context of the research question and hypothesis. Similarly, the performance comparison of the classifiers will be discussed in this section. The section is concluded by conferring the limits and strengths of the study.

5.1 Introduction

In these experiments, two hybrid models were studied, and the performance of each of them was measured by using the cross-validation scores, classification accuracy and other measures from confusion-matrix like precision, recall, AUROC curve and classification error as evaluation metrics. Table 4.2 displays a summary of the results. In particular, K-fold cross-validation with k=10 was applied to training sets to avoid over-fitting. Further, to evaluate the model, the confusion matrix, error rate, classification reports, and some errors which are the difference between predicted labels and correct labels will be illustrated. Finally, concrete images that have never been seen by the systems will be used to test the models and show the anticipated figures.

5.2 Evaluation of the Results

The first experiment was to train CNN on 19214 samples and evaluate the performance of the CNN on a sample of 4804 records from the dataset using cross-validation split to extract features. The features from the dense layer were extracted and predicted on test and train datasets each containing 24018 and 16012 to obtain test and training features respectively. From the results of this experiment, it was evident that the model was not overfitted as the validation loss and training loss were similar like the accuracy. The validation loss seemed to increase after three epochs, and the accuracy was saturated. Hence the model was trained only for three epochs. Figure 5.1 represents the performance of CNN for four epochs.

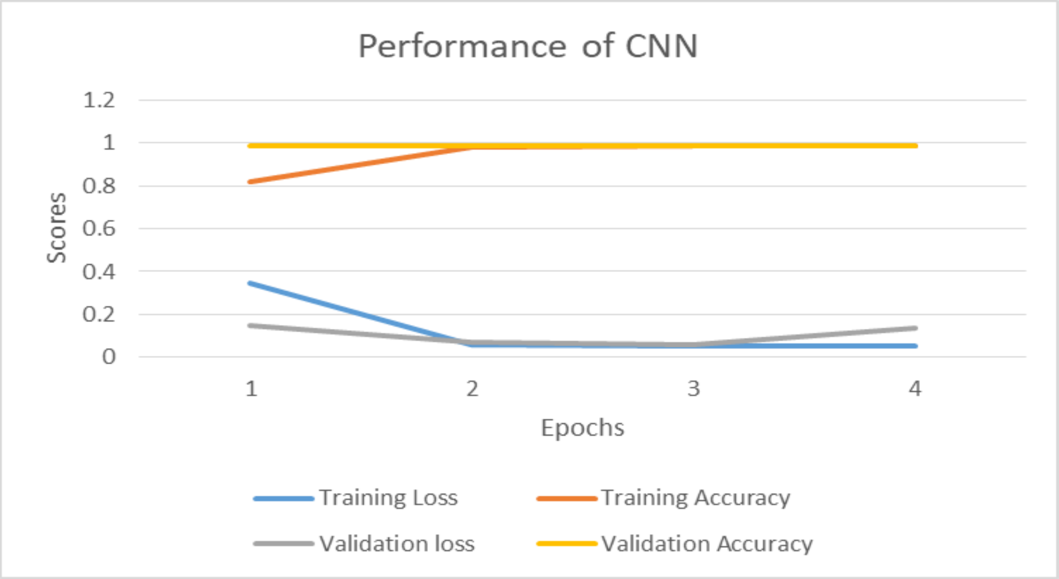


Figure 5.1 Performance of CNN for four epochs

5.3 Evaluating the model to predict the crack in Concrete Images

The Classification Accuracy of 0.99 indicated that the CNN-XGBoost model was able to classify the crack images as a crack and non-crack images as non-crack better than CNN-SVM. Even though some of the images contained background noise along with the crack, the model was successful in identifying and classifying the images correctly. With an average accuracy of 98.83% for XGBoost model, the null hypothesis can be rejected, and it is concluded that the combination of Convolutional neural networks and XGBoost can detect the crack patterns from the images of concrete in a better way by outperforming CNN-SVM model. The out-performance can also be seen from Figure 5.2

5.3.1 Model Accuracy Measures

The optimal design of both models was assessed using a confusion matrix to determine the best hybrid model for crack detection. Shown in Figure 5.3, are the confusion matrices. Both the CNN-SVM model and CNN-XGBoost model presented the highest accuracy of 98.76% and 98.79% respectively.

As can be seen from Fig.5.3, CNN-XGBoost performed considerably well on the images with fewer errors when compared with CNN-SVM.

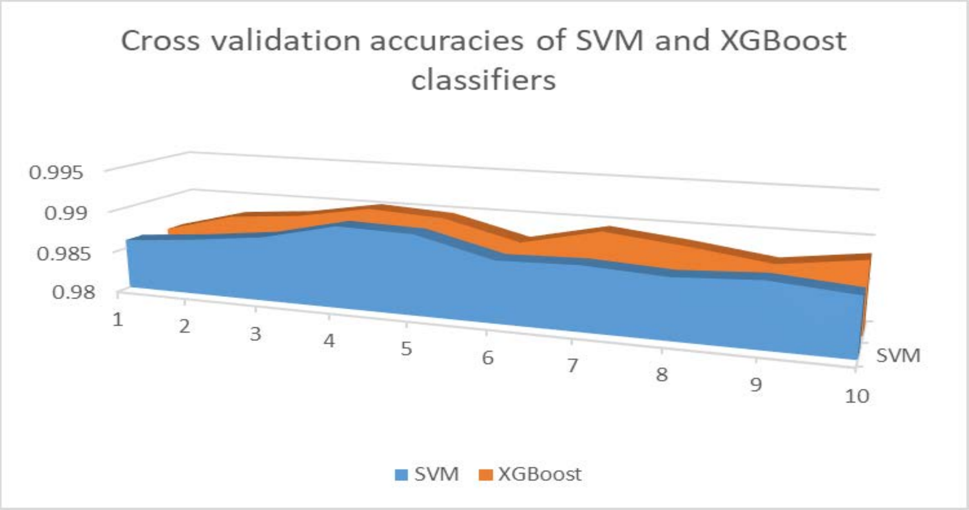


Figure 5.2 10 Fold Cross Validation Accuracies of CNN-SVM and CNN-XGBoost

Figure 5.4 depicts a few misclassified samples of both the models with their prediction.

The classification report is often used to check the quality of classification model predictions. Table 5.1 shows the primary classification metrics like precision, recall, and classification error for both models. These are defined as the True Positive, True Negatives, False Positives, and False negatives. The CNN-XGBoost model had a slight difference in all metrics and seemed better than CNN-SVM.

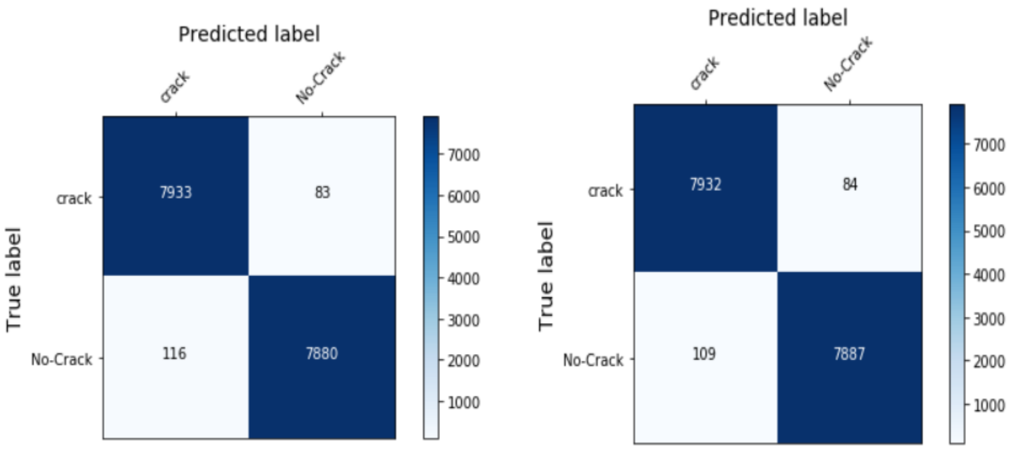


Figure 5.3 (Left) Confusion Matrix for CNN-SVM model, (Right) Confusion Matrix for CNN-XGBoost model

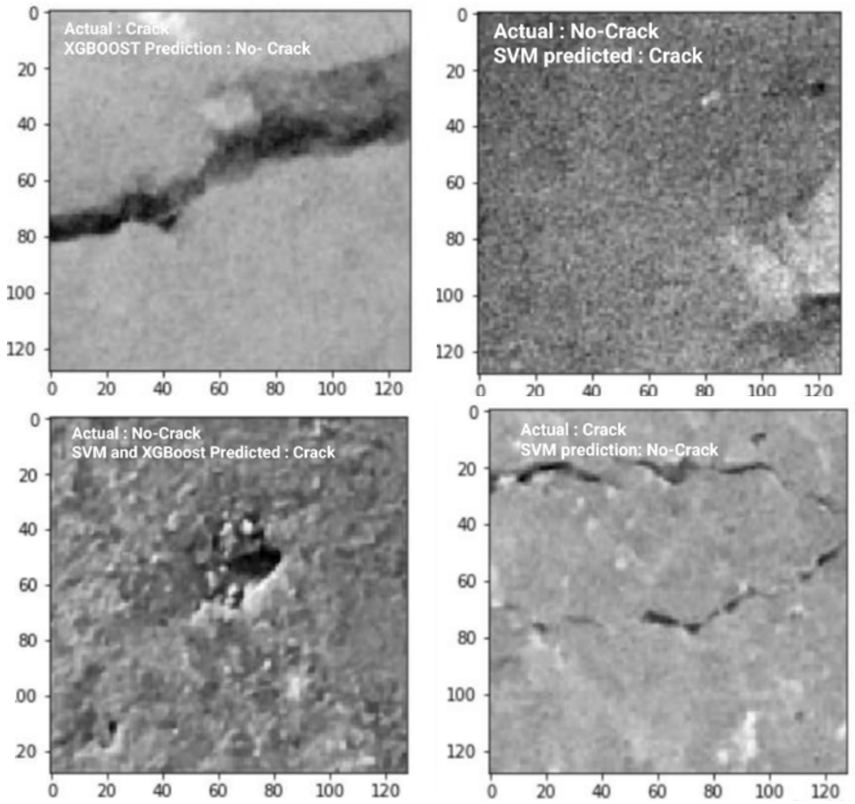


Figure 5.4 Misclassified samples

5.4 Hypothesis Evaluation

The objective of the research was to build a hybrid CNN-XGBoost model that can deliver a better performance in terms of crack detection, compared to the CNN-SVM model.

The hypothesis coined to achieve the objective is as follows:

H0: "The application of hybrid CNN-XGBoost model in crack detection will result in no performance increase over hybrid CNN-SVM model."

From the research, Combination of CNN with two machine learning models, XGBoost and SVM could be built with following AUC scores and mean cross-validation accuracy.

Metrics	CNN-SVM	CNN-XGBoost
True Positive	7933	7932
False Positive	116	109
True Negative	7880	7887
False Negative	83	84
Classification error	0.0124	0.0121
Recall/Sensitivity	0.985	0.986
Precision	0.99	0.989

Table 5.1 Performance metrics of CNN-SVM and CNN-XGBoost calculated from respective confusion matrices

From Table 5.2, it could be seen that the CNN-XGBoost model built with the same dataset size of 40,000 instances has an AUC Score of 0.999. The mean difference between the models is 8 x10-4. A paired t-test was implemented on cross-validation scores of both the models to check if the difference was due to chance or if it was a statistically significant difference. The test exhibited a statistically significant difference between the mean scores with a p-value of 0.013 (<0.05), thereby providing evidence to reject the null hypothesis. Therefore, it can be concluded that CNN-XGBoost model outperforms CNN-SVM model in classifying the concrete images into having or not having cracks.

Model	Data Size	AUC Scores	Mean Accuracy
CNN-SVM	40000	0.988	0.987584
CNN-XGBoost	40000	0.999	0.988384

Table 5.2 AUC Score and Cross-validation Mean Accuracy scores of the models

5.5 Strengths and Limitations of the Results

The ability of hybrid machine learning models, namely, CNN-SVM and CNN-XGBoost, to process images of concrete surfaces and predict their condition is studied as a part of this research study. The key strength of the study was its ability to accurately identify the concrete cracks when compared to all other existing algorithms. The images used for the experiment have been initially resized into a standard size, which makes it easier to feed them into a CNN model. The architecture of the CNN model used for the experiment was selected based on the previous work done on the same dataset to extract features. The same architecture can be used for various other datasets in the same domain to predict the condition of different surfaces or different structures. The model also has a reasonable convergence rate for a small number of epochs of training.

One of the limitations of the research is that the model is trained on images which were captured under the same illumination condition. The patterns in the dataset were clearly evident, which led to high accuracies. A varied and challenging dataset with different background illuminations can be used to make the model more robust. The pre-processing steps carried out in the research is data specific. A different dataset should be differently pre-processed to fit the models built as part of this research.

6. CONCLUSION

This last chapter of the research will present a short explanation of the research outcomes and stress about the novelty of the current study, including the problems that were addressed, and the limitations of the study. This section will also provide suggestions for future work.

6.1 Problem Definition

The problem of this research study was: Can a hybrid ‘Convolutional Neural Network- eXtreme Gradient Boosting’ model statistically out-perform the hybrid ‘Convolutional Neural Network-Support Vector Machine’ model in classifying concrete images into having or not having cracks?”

As described in the first chapter, the null hypothesis (H_0) of this research is that the application of hybrid CNN-XGBoost model in crack detection will result in no performance increase over hybrid CNN-SVM model.

Based on the evaluation and comparison of the two models, the results have clearly shown the difference in the performance of the CNN-XGBoost classifier. Finally, the classification scores of the classifiers were assessed to reject the null hypothesis.

6.2 Design, Experimentation, Evaluation and Results

This study attempted to recognize the cracks in concrete images by using tools from Machine Learning to train the classifier. The models were evaluated through the Mendeley Concrete Crack dataset which had 20,000 instances of Positive (Crack) and Negative (Non-Crack) data each. It is a brilliant data for machine learning-based crack detection while taking minimal efforts in preprocessing. According to literature analysis, few research studies have made some achievements in the field of crack detection. For instance, (Kim, Ahn, Shin, & Sim, 2019) used a classification framework based on the CCRs for identifying cracks in the presence of non-crack objects that share similar image characteristics (i.e., shape and color)., as well as highlighting the proposed method by achieving a 98% detection accuracy. Recently, (Ni, Zhang, & Chen, 2019) applied the Google net Architecture of CNN to classify crack images with a maximum efficiency of 89.5%. Overall, the Convolutional Neural Networks have been well verified by previous researchers and got good results. But, too many parameters in the fully connected layers of CNN limits the training using the deep learning model. The framework proposed in this research aims at solving this problem. SVM and XGBoost systems are used instead of the fully connected layer in this framework to complete the classification task. This framework integrates feature extraction capabilities of convolutional neural network and advantages of the XGBoost and SVM classifier. The frameworks took less training time but achieved higher accuracy. The CNN-XGBOOST algorithm won with a classification accuracy of 98.79%, followed by CNN-SVM with 98.76%. Finally, both the models attained the goal of successfully improving the accuracy to over 98%. So, both frameworks have proven to be valid for concrete or other composite materials like steel images' classification.

Two models were analyzed and evaluated by a series of tools such as confusion matrices, k-fold cross-validation, error rates, and classification reports. Each modification produced changes in the results mostly improved accuracy and widely varying performance times. The objective of this study was to decrease the misclassification rate in crack detection using XGBoost model over SVM model on CNN extracted features which was achieved. To conclude, the performance of the models have been verified again with some images never seen by the systems using the classifier and achieved satisfactory results.

6.3 Contribution and Impact

Image Classification and analysis is an entertaining research area in Artificial Intelligence, and also significant to a variety of present open research problems. Concrete crack detection is a well-researched subarea and one of the vital benchmark task within the field because it is directly related to the threat of human beings. Examining damages at concrete constructions due to physical, chemical, and mechanical contacts requires the use of advanced non-destructive testing approaches that can trace dimensional variations of microstructures. Hence, concrete crack detection is still an active area of research. With this vision in mind, the current research focussed on developing machine learning-based hybrid models to discover cracks on concrete surfaces, thereby providing innovative archetypes for the assessment of structures. Currently, the study is restricted to identifying concrete cracks through a binary classification method; i.e., the model classifies whether or not a crack is present on the concrete surface. The aim of increasing crack detection accuracy was concentrated throughout this study. Therefore CNN was used for automatic feature extraction. Finally, XGBoost applied to CNN improved the efficiency of the crack detection system to more than 98.5%. Overall, the implementation and completion of this project have a series of advantages. One clear example is that the system can identify almost all superficial defects (e.g., cracks and corrosion) with a less computational cost and by saving a lot of human efforts. Besides, the benefits of applying this system are substantial, as a lot of components of current civil structures, such as pavements, bridges, and erections, are suffered from quick aging and involves a massive amount of nation’s resources from federal and state agencies to check and preserve them.

6.4 Future Work and Recommendations:

This research has introduced the hybrid machine learning method to predict the cracks from the concrete structures. The machine learning model was built only using the available data which has been taken in similar illumination conditions. In the future, a better prediction model could be developed by including data with varied illumination conditions. Also, fine-tuning of the classifiers and experimenting with new hybrid algorithms is recommended for future work. The primary objective of this study is to classify concrete images into having cracks and non-cracks using hybrid models. But, the classification of type and severity of cracks is not included in this research. Future research can focus on it by using hybrid CNN-XGBOOST model, thereby helping the authorities to prioritize and take necessary actions accordingly. Processing time of the models can be considered as an evaluation metric in future researches.

BIBLIOGRAPHY

Agarap, A. F. (2017). An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification. Retrieved from http://arxiv.org/abs/1712.03541

Akbari, H., Halig, L. V., Schuster, D. M., Osunkoya, A., Master, V., Nieh, P. T., … Fei, B. (2012). Hyperspectral imaging and quantitative analysis for prostate cancer detection. Journal of Biomedical Optics, 17(7), 0760051. https://doi.org/10.1117/1.JBO.17.7.076005

Ayumi, V. (2017). Pose-based human action recognition with Extreme Gradient Boosting. Proceedings - 14th IEEE Student Conference on Research and Development: Advancing Technology for Humanity, SCOReD 2016, 1–5. https://doi.org/10.1109/SCORED.2016.7810099

Bose Samir Kumar Bandyopadhyay, K. (2016). Crack Detection and Classification in Concrete Structure. Journal for Research, 02(04), 29–38.

Cha, Y. J., Choi, W., Suh, G., Mahmoudkhani, S., & Büyüköztürk, O. (2018). Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types. Computer-Aided Civil and Infrastructure Engineering, 33(9), 731–747. https://doi.org/10.1111/mice.12334

Chaiyasarn, K., Khan, W., Ali, L., Sharma, M., Brackenbury, D., & DeJong, M. (2018). Crack detection in masonry structures using convolutional neural networks and support vector machines. ISARC 2018 - 35th International Symposium on Automation and Robotics in Construction and International AEC/FM Hackathon: The Future of Building Things, (November). https://doi.org/10.22260/ISARC2018/0016

Chatterjee, S., Saeedfar, P., Tofangchi, S., & Kolbe, L. (2018). Intelligent Road Maintenance: a Machine Learning Approach for surface Defect Detection. In ECIS (p. 194).

Chen, T., He, T., Benesty, M., Khotilovich, V., & Tang, Y. (2015). Xgboost: extreme gradient boosting. R package version 0.4-2, 1-4.

Copur, M., Melisozyildirim, B., & Ibrikci, T. (2018). Image Classification of Aerial Images Using CNN-SVM. Proceedings - 2018 Innovations in Intelligent Systems and Applications Conference, ASYU 2018. https://doi.org/10.1109/ASYU.2018.8554008

Dorafshan, S., Thomas, R. J., & Maguire, M. (2018). Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. Construction and Building Materials, 186, 1031–1045. https://doi.org/10.1016/j.conbuildmat.2018.08.011

Georganos, S., Grippa, T., Vanhuysse, S., Lennert, M., Shimoni, M., & Wolff, E. (2018). Very High Resolution Object-Based Land Use-Land Cover Urban Classification Using Extreme Gradient Boosting. IEEE Geoscience and Remote Sensing Letters, 15(4), 607–611. https://doi.org/10.1109/LGRS.2018.2803259

Hoang, N.-D., & Nguyen, Q.-L. (2018). Automatic Recognition of Asphalt Pavement Cracks Based on Image Processing and Machine Learning Approaches: A Comparative Study on Classifier Performance. Mathematical Problems in Engineering, 2018. https://doi.org/10.1155/2018/6290498

Jiang, S., Zhao, H., Wu, W., & Tan, Q. (2018). A novel framework for remote sensing image scene classification. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, 42(3), 657–663. https://doi.org/10.5194/isprs-archives-XLII-3-657-2018

Kim, B., & Cho, S. (2018). Automated vision-based detection of cracks on concrete surfaces using a deep learning technique. Sensors (Switzerland), 18(10). https://doi.org/10.3390/s18103452

Kim, H., Ahn, E., Shin, M., & Sim, S. H. (2019). Crack and Noncrack Classification from Concrete Surface Images Using Machine Learning. Structural Health Monitoring, 18(3). https://doi.org/10.1177/1475921718768747

Liang, S., Jianchun, X., & Xun, Z. (2018). An extraction and classification algorithm for concrete cracks based on machine vision. IEEE Access, 6, 45051–45061. https://doi.org/10.1109/ACCESS.2018.2856806

Lin, J., & Liu, Y. (2010). Potholes detection based on SVM in the pavement distress image. Proceedings - 9th International Symposium on Distributed Computing and Applications to Business, Engineering and Science, DCABES 2010, 544–547. https://doi.org/10.1109/DCABES.2010.115

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., & Omata, H. (2018). Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone, 2000, 4–6. Retrieved from http://arxiv.org/abs/1801.09454

Mao, Q., Zou, Q., Wang, S., Li, Q., & Cao, Y. (2011). CrackTree: Automatic crack detection from pavement images. Pattern Recognition Letters, 33(3), 227–238. https://doi.org/10.1016/j.patrec.2011.11.004

Mohan, A., & Poobal, S. (2018). Crack detection using image processing: A critical review and analysis. Alexandria Engineering Journal, 57(2), 787–798. https://doi.org/10.1016/j.aej.2017.01.020

Ni, F. T., Zhang, J., & Chen, Z. Q. (2019). Pixel-level crack delineation in images with convolutional feature fusion. Structural Control and Health Monitoring, 26(1). https://doi.org/10.1002/stc.2286

Özgenel, Çağlar Fırat (2018), "Concrete Crack Images for Classification", Mendeley Data, v1http://dx.doi.org/10.17632/5y9wdsg2zt.1

Silva, W. R. L. da, & Lucena, D. S. de. (2018). Concrete Cracks Detection Based on Deep Learning Image Classification. Proceedings, 2(8), 489. https://doi.org/10.3390/icem18-05387

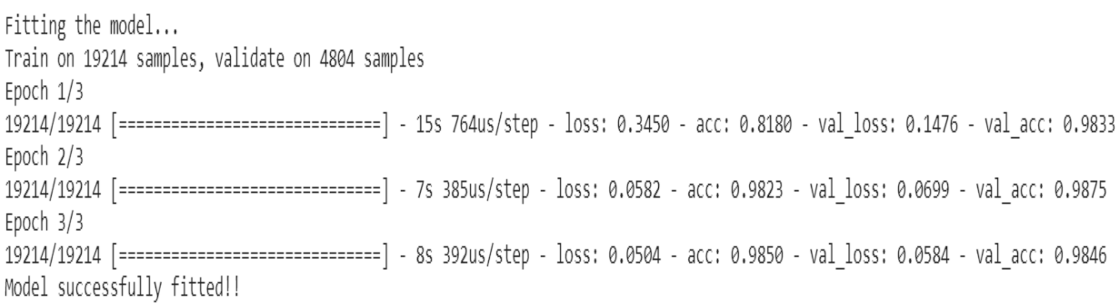
Sharma, M., Anotaipaiboon, W., & Chaiyasarn, K. (2018). Concrete Crack Detection using the integration of Convolutional Neural Network and Support Vector Machine, (June), 1–9. https://doi.org/10.14456/scitechasia.2018.11

Zhang, L., Yang, F., Daniel Zhang, Y., & Zhu, Y. J. (2016). Road crack detection using deep convolutional neural network. Proceedings - International Conference on Image Processing, ICIP, 2016-Augus(October 2017), 3708–3712. https://doi.org/10.1109/ICIP.2016.7533052

Zou, Q., Cao, Y., Li, Q., Mao, Q., & Wang, S. (2012). CrackTree: Automatic crack detection from pavement images. Pattern Recognition Letters, 33(3), 227–238. https://doi.org/10.1016/j.patrec.2011.11.004

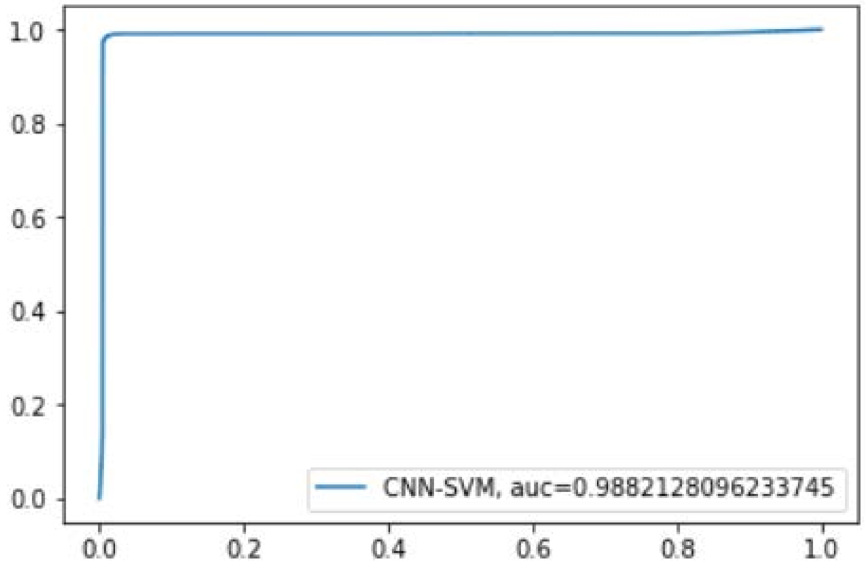
APPENDIX A

Performance of CNN

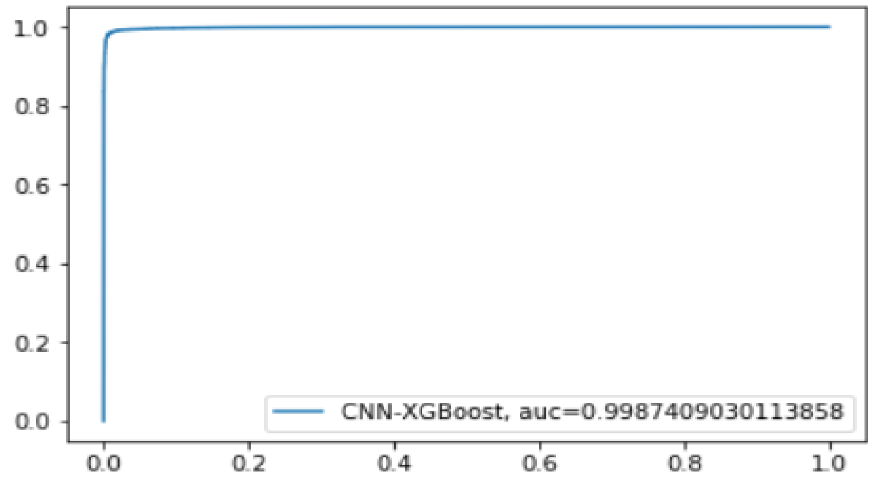


```
Fitting the model...
Train on 19214 samples, validate on 4804 samples
Epoch 1/3
19214/19214 [==============================] - 15s 764us/step - loss: 0.3450 - acc: 0.8180 - val_loss: 0.1476 - val_acc: 0.9833
Epoch 2/3
19214/19214 [==============================] - 7s 385us/step - loss: 0.0582 - acc: 0.9823 - val_loss: 0.0699 - val_acc: 0.9875
Epoch 3/3
19214/19214 [==============================] - 8s 392us/step - loss: 0.0504 - acc: 0.9850 - val_loss: 0.0584 - val_acc: 0.9846
Model successfully fitted!!
```

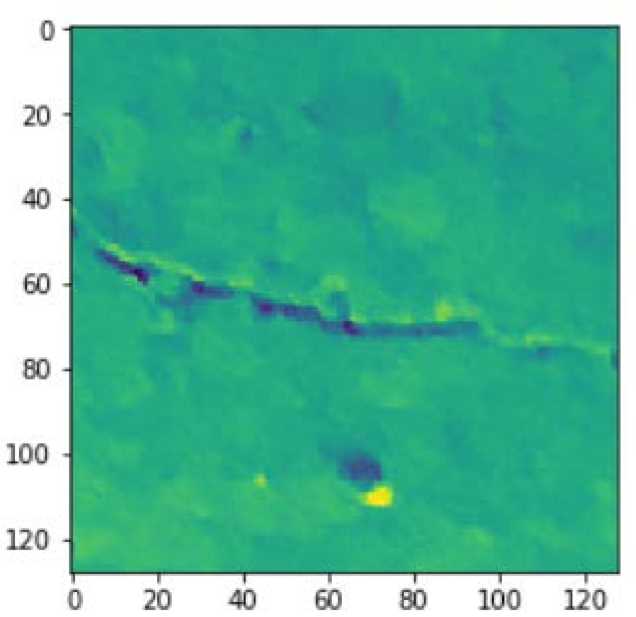
ROC Curve of CNN-SVM Model



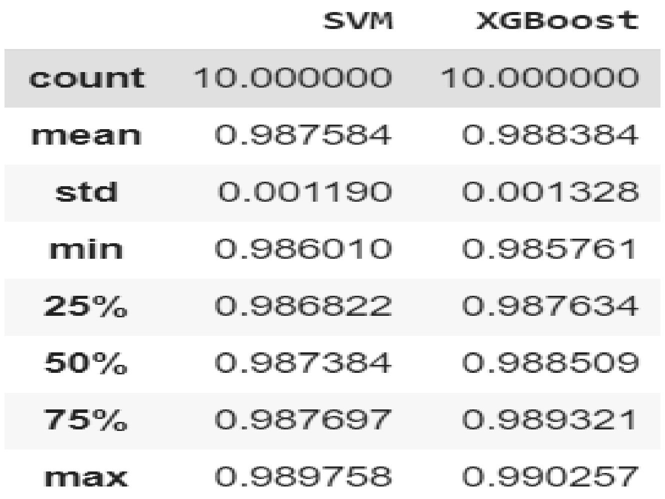
ROC Curve of CNN-XGBoost Model



One of the misclassified sample

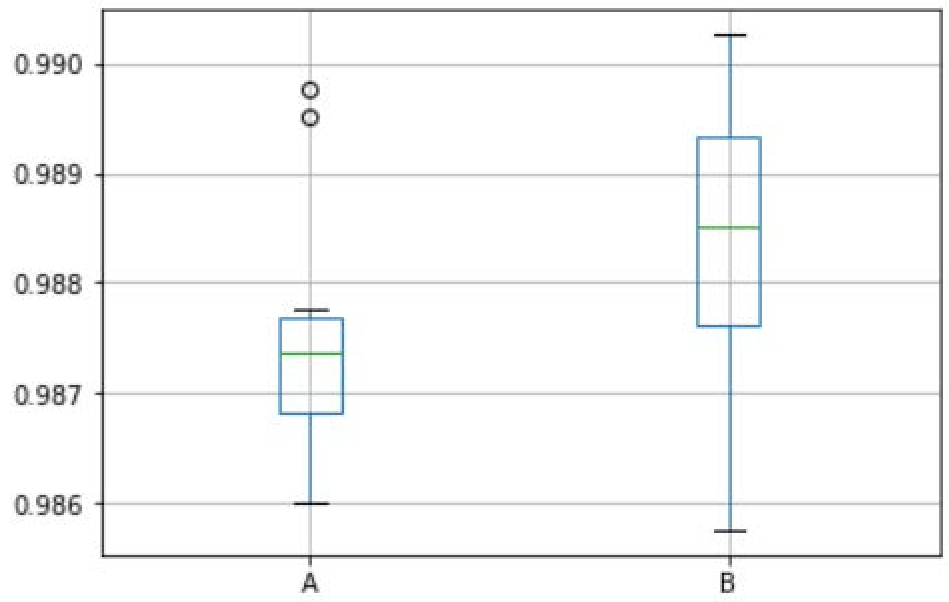


Descriptive Statistics of 10 fold Cross-Validation Scores of hybrid models built

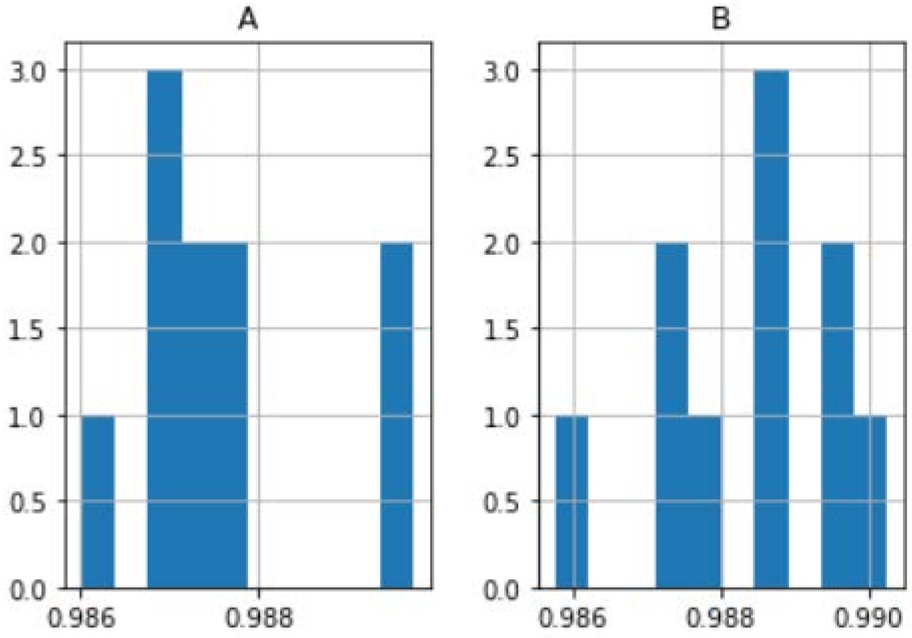


	SVM	XGBoost
count	10.000000	10.000000
mean	0.987584	0.988384
std	0.001190	0.001328
min	0.986010	0.985761
25%	0.986822	0.987634
50%	0.987384	0.988509
75%	0.987697	0.989321
max	0.989758	0.990257

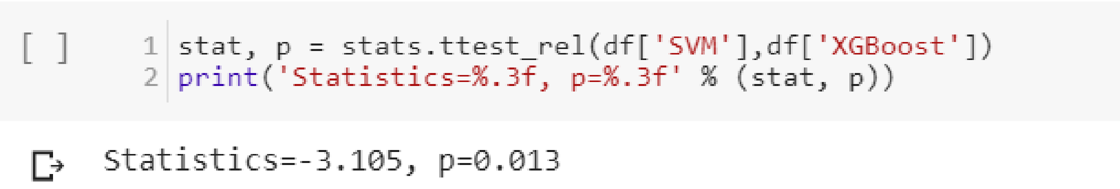
Boxplot for Performance of Model A(CNN-SVM) and B(CNN-XGBoost)



Distribution of cross-validation scores of model A and B



Results of paired t-test



```
stat, p = stats.ttest_rel(df['SVM'],df['XGBoost'])
print('Statistics=%.3f, p=%.3f' % (stat, p))
```

```
Statistics=-3.105, p=0.013
```