ISTANBUL TECHNICAL UNIVERSITY ★ INFORMATICS INSTITUTE

DEEP LEARNING BASED CRACK DETECTION WITH APPLICATIONS TO STRUCTURAL HEALTH MONITORING

M.Sc. THESIS

Mahtab MOHTASHAM KHANI

Department of Computer Sciences

Computer Science Programme

JUNE 2019

ISTANBUL TECHNICAL UNIVERSITY ★ INFORMATICS INSTITUTE

DEEP LEARNING BASED CRACK DETECTION WITH APPLICATIONS TO STRUCTURAL HEALTH MONITORING

M.Sc. THESIS

Mahtab MOHTASHAM KHANI
(704171003)

Department of Computer Sciences

Computer Science Programme

Thesis Advisor: Asst. Prof. Dr. Nazım Kemal ÜRE

JUNE 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

YAPISAL SAĞLIK İZLENMESİNDE DERİN ÖĞRENME TEMELLİ ÇATLAK TESPİTİ

YÜKSEK LİSANS TEZİ

Mahtab MOHTASHAM KHANI
(704171003)

Bilgisayar Bilimleri Anabilim Dalı

Bilgisayar Bilimleri Programı

Tez Danışmanı: Asst. Prof. Dr. Nazım Kemal ÜRE

HAZİRAN 2019

Mahtab MOHTASHAM KHANI, a M.Sc. student of ITU Informatics Institute 704171003 successfully defended the thesis entitled “DEEP LEARNING BASED CRACK DETECTION WITH APPLICATIONS TO STRUCTURAL HEALTH MONITORING”, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Asst. Prof. Dr. Nazım Kemal ÜRE** Istanbul Technical University	....
Jury Members :	**Assoc. Prof. Dr. Behçet Uğur TÖREYIN** Istanbul Technical University	....
	Dr. Umut GENÇ University of Cambridge, Eatron Technologies	....

Date of Submission :3 June 2019
Date of Defense :13 June 2019

To my family,

FOREWORD

This thesis is a framework of crack detection based on deep learning as a completion for M.S thesis in the computer science program, at Istanbul Technical University. The research was challenging but joyful and allowed me to learn all I need to do in the detection issues. Fortunately, my advisor, Prof. Nazim Kemal Ure was always available for all my questions and queries and he influenced me by his knowledge on how to convey and perform research. I would like to express my sincere gratitude to him for the excellent guidance and support during this process. Also, I am so grateful for the lessons I took with Professor B.Ugur Toreyin that provided me well in computer vision and many other kinds of stuff. I also wish to thank my spouse Sahand Vahidnia an excellent mentor for me during this work. I'm appreciating his patience and motivation for me to learning how to work in a professional way. Also, my friend Leila Ghasemzadeh who have been not only great colleagues but also her wonderful cooperation during this project provided me with great insights into the team working. I'm also so grateful of all my friends for their kind support. Finally, I would like to thank my family for their love and support. They always encouraged me to pursue my dreams, which I'm so grateful to them.

June 2019 Mahtab MOHTASHAM KHANI
(Computer Vision Expert)

TABLE OF CONTENTS

Page

ABBREVIATIONS

ML	: Machine Learning
ANN	: Artificial Neural Network
App	: Appendix
CNN	: Convolutional Neural Network
CDAT	: Crack Detection Annotation Tool
CDTT	: Crack detection Training Tool
GE	: General Electric
ITU	:Istanbul Technical University
SGD	: Stochastic Gradient Descent
HPF	: High Pass Filters
LPF	: Low Pass Filters
resMax	: Maximum Resolution
resMin	: Minimum Resolution
MLP	: Multi-Layer Perceptrons
ReLU	: Rectified Linear Unit
TP	: True Positive
TN	: True Negative
FP	: False Positive
FN	: False Negative
VI	: Visual Inspection

SYMBOLS

c(x,y) : Closeness Function
s(x,y) : Similarity Function
f(x) : Input Image
k(x) : Normalization Value
h(x) : Bilateral Filter

LIST OF TABLES

LIST OF FIGURES

DEEP LEARNING BASED CRACK DETECTION WITH APPLICATIONS TO STRUCTURAL HEALTH MONITORING

SUMMARY

Computer vision has been a hot research topic for years with broad applications. Various detection applications and detection techniques exist in the literature. Crack detection from images is a popular problem since it applies to different structures like bridges, dams, pavements, concretes or metals structures and etc. Manual visual inspection (VI) of structural defects and cracks is very time-consuming, and sometime unfeasible due to the volume of data and the size of the structure. Manual inspection of cracks and defects in structures is prone to human error for a range of reasons like fatigue, irresponsible inspection, weak eye sight and even sabotage. In addition, some structural defects like the ones with low contrast between cracks and the surrounding areas, is challenging to the naked eye to detect. Structural and safety maintenance require a consistent inspection of cracks and other anomalies. The inspections provide information regarding the life condition of the structures and yield information for estimating structural health and repair costs. Automation of such inspections, can reduce the reliance on manual inspections and reduce the error. Incorporating automatic inspection can increase the frequency of inspections and assist human inspector in variety of ways.

Hence, incorporating Image processing techniques (IPTs) and computer vision based algorithms, provides a good opportunity and a viable solution to deal with these challenges. Various image processing techniques have been used in the past to address the problem of automated visual crack detection, with varying degrees of success. In this work a novel crack detection framework is proposed, which utilizes techniques from both classical image processing and deep learning methodologies. The main contribution of this work is demonstrating that applying filters to image data in the pre-processing phase can significantly boost the classification performance of a convolutional neural network based model. Wide utilization of IPTs, especially for image pre-processing, helped us in achieving the results that have outperformed the prior best methods.

The proposed vision-based method, utilizes convolutional neural networks (ConvNets or CNNs) as its main mechanism for detecting cracks in the structures. The designed CNN architecture has been trained on 650 images, which was labeled by a custom annotation tool and a classification accuracy of 96.26 % was achieved on image blocks, on a dataset of cracked surface images. Through grid searching among a range of CNN parameters, a CNN network was nominated as the best network for training our model. The developed custom grid search accepts variety of parameters, including number of layers, activation functions, number of neurons, number of classes, input dimension, value of dropout and max pooling layers. Although the parameter search techniques (grid search in this case) can provide the best possible combination of parameters, there are other factors contributing to the success of a model. In this work, the accuracy has

been achieved due to the novel technique proposed, which is the incorporation of IPTs. IPTs can improve the accuracy and among them, bilateral filtering has been observed to perform very well as a pre-processing and smoothing technique for detection with CNNs.

In addition to the CNN structure and pre-processing techniques, there are other factors contributing to the success of a model. The research findings shows that the quality of the dataset is one of the most important factors of model generalization success. Dataset quality means a the balance of classes, quality of images and their properties. Although the pre-processing techniques are introduced to improve the quality of CNN input, annotation and data acquisition should be performed with care and precision. In this work, the custom data annotation tool has been developed for this reason.

In this work, further and complete analysis of CNN architectures, parameters, and their contribution to failure or success of the model has been discussed in detail. Additionally, incorporation of IPTs and the impact of such pre-processing techniques have been studied and the results have been discussed in detail.

YAPISAL SAĞLIK İZLENMESİNDE DERİN ÖĞRENME TEMELLİ ÇATLAK TESPİTİ

ÖZET

Bilgisayar görmesi, son yıllarda geniş uygulama alanları ile birlikte popülaritesi artan bir araştırma konusu olmuştur. Literatürde çeşitli uygulamalarda farklı tespit teknikleri bulunmaktadır. Bilgisayar görmesi uygulamalarından biri olan, köprüler, barajlar, kaldırımlar, betonlar veya metal yapılar gibi farklı yapılar üzerinde görüntülerden çatlak tespiti, oldukça popüler olan bir araştırma alanıdır. veri hacmi ve yapının boyutu nedeni ile yapılardaki çatlakların ve kusurların manuel olarak incelenmesi zaman alıcı ve hatta bazen imkansızdır. yorgunluk, sorumsuz muayene, zayıf göz görme ve hatta sabotaj gibi çeşitli nedenlerden dolayı çatlak ve kusurların yapılarda incelenmesi insan hatalarına açıktır. Ek olarak, çatlaklar ve çevresindeki alanlar arasında düşük kontrastlı olanlar gibi bazı yapısal kusurlar, tespit etmek için çıplak göze zordur. Yapısal ve güvenlik bakımı, çatlakların ve diğer anomalilerin tutarlı bir şekilde incelenmesini gerektirir. Denetimler, yapıların yaşam durumuna ilişkin bilgi sağlamakta ve yapısal sağlık ve onarım maliyetlerini tahmin etmek için bilgi vermektedir. Bu denetimlerin otomasyonu, manuel denetimlere olan güvensizliği ve hatayı azaltabilir. Otomatik muayenenin yapılması, denetimlerin sıklığını artırabilir ve insan denetçisine çeşitli şekillerde yardımcı olabilir.

Bu nedenle, Bu zorlukların üstesinden gelmek için görüntü işleme tekniklerini (IPT'ler) ve bilgisayarlı görü tabanlı algoritmaları kullanmak iyi bir fırsat ve uygun bir çözüm sunar. Geçmişte, çeşitli başarı derecelerinde otomatik görsel çatlak tespiti sorununu ele almak için çeşitli görüntü işleme teknikleri kullanılmıştır. Bu çalışmada, hem klasik görüntü işleme hem de derin öğrenme metodolojilerinden teknikleri kullanan yeni bir çatlak tespit çerçevesi önerilmiştir. Bu çalışmanın ana katkısı, işleme öncesi aşamadaki görüntü verilerine filtre uygulanmasının, convolutional bir sinir ağı temelli modelin sınıflandırma performansını önemli ölçüde artırabileceğini göstermesidir. Özellikle ön işleme görüntü için IPT'lerin geniş kullanımı, önceki en iyi yöntemleri geride bırakan sonuçlara ulaşmamıza yardımcı oldu.

Önerilen vizyon temelli yöntem, yapılardaki çatlakları tespit etmek için ana mekanizma olarak convolutional sinir ağlarını (ConvNets veya CNN'ler) kullanır. Tasarlanan CNN mimarisi, 650 görüntü üzerinde eğitildi; ki, özel bir açıklama aracıyla etiketlenmiştir ve kırık yüzey görüntülerinin bir veri kümesi üzerinde,görüntü bloklarında, % 96,26 sınıflandırma doğruluğu elde edildi. Bir dizi CNN parametreleri arasında şebeke arama yoluyla, CNN ağı modelimizi eğitmek için en iyi ağ olarak belirlenmiştir. Geliştirilmiş özel şebeke araştırması, katman sayısı, etkinleştirme fonksiyonları, nöron sayısı, sınıf sayısı, giriş boyutu, bırakma değeri ve max pooling katmanları dahil olmak üzere çeşitli parametreleri kabul eder. Parametre arama teknikleri (bu çalışmada şebeke arama) parametrelerin mümkün olan en iyi kombinasyonunu sağlayabilmesine rağmen, modelin başarısına katkıda bulunan başka faktörler de vardır. Bu çalışmada, önerilen yeni teknik nedeniyle IPT'lerin dahil

edilmesi ile doğruluk sağlanmıştır. IPT'ler doğruluğu artırabilirler ve aralarında, bilateral filtering, ön işleme ve kolaylaştırma tekniği olarak CNN'lerle tespit etmek için çok iyi performans gösterdiği gözlenmiştir.

Modelin başannsında, CNN grapısı ve ön işleme tekniklerine ilave başka faktörlerde. Katkı sağlamaktadır. Araştrmalara göre,veristının doğruluğu,modelin genelliştirme başarısında önemlı faktörlerden birdir. Verisetinim kalitesi ve özellikleri anlamındır. Ğerçi CNN giriş kalıtesinin artmas için, ön işleme teknikleri önerilmiştir, yineke veri toplama ve işaretleme işlemlerı dikkatli ve doğru birşekilde yaplımalıdır. Bu çalışmada özel veri işaretleme aracı, yukarıda anlatılan amaç için geliştirilmiştirç.

Bu çalışmada, CNN mimarileri, parametreleri ve anların modelın başarı ya da başarısızlıgında olan etkileri daha ayrıntılı ve eksızce anlatılmıştır. Ek olarak, IPT'lerin dahil edilmesi ve bu tür ön işleme tekniklerinin etkisi araştırılmış ve sonuçlar ayrıntılı olarak tartışılmıştır.

bir katmanın amacı görüntüdeki özelliklerin varlığını tespit etmek için kullanılan filtrelemektir. Karmaşık yapıların ve şekillerin özellikleri. Filtre bir görüntü üzerinde hareket ederken. Filtreyi seçtikten sonra, adım ve dolguyu da seçmek zorundayız. Hem doldurma hem de adımlama veri boyutunu etkiler. Adım, filtrenin giriş hacminin etrafında nasıl büküleceğini kontrol eder. Dolgu, daha derin ağlar tasarlamamıza olanak tanır ve özellik haritamızın büzülmesini önler. Ayrıca, dolgular aslında bilgileri sınırlarda tutarak performansı artırır. Daha önce belirtildiği gibi Havuzlama Katmanı, CNN katmanlarında gerçekleşir. Çekirdeği alıp çekirdeği görüntünün üzerine götürdüğümüz kıvrımlı sesler gibi çalışır. Havuzlamanın işlevi, ağdaki parametre ve hesaplama sayısını sürekli olarak azaltmaktır.

Maksimum havuzlama, minimum havuzlama ve ortalama havuzlama gibi birkaç tür havuz vardır. Max havuzu sizi bir sonraki seviyeye götürür. Diğer havuzlama yöntemleri olsa da, maksimum havuzlama genellikle bu çalışmada kullandıklarımızdan daha etkilidir.

CNN tabanlı çatlak tespit teknikleri temel olarak iki gruba ayrılır. İlk grup (blok seviyesi tespiti), çatlak yamalarının tespiti ve üzerlerinde sınırlayıcı kutular temin edilmesine dayanır. İkinci grup (piksel seviyesi tespiti), piksel seviyesinde çatlak tespiti sağlayan çatlak segmentasyonuna (delineation) dayanmaktadır.

Daha sonraki çalışmalarda somut görüntülerde çatlakların tespit edilmesine yönelik derin bir öğrenme yaklaşımı önerilmiştir. Bu çalışmada, sonuçlar, önerilen CNN'nin doğruluğunu arttırdığını gösteren Canny ve Sobel kenar algılama algoritmaları ile karşılaştırılmıştır. CNN'lerin kullanılması genellemeyi arttırır ve verideki gürültünün etkisini azaltır. Yazar, veri setinde %97'den fazla test doğruluğu olduğunu iddia ediyor.

Önerilen CNN mimarisi dört katlamalı katmanı, bir makspooling katmanı ve iki yoğun katmanı içerir. Destek Vektör Makineleri (SVM) ve Yükseltme yöntemleri gibi klasik makine öğrenme yöntemleriyle karşılaştırıldığında, önerilen CNN'ler, kaldırım çatlağı tespit veri setinde çok daha iyi doğruluk ve geri çağırma puanları (0.925) sağlar.

Önerilen yöntem, ilgi alanı üzerinde bir sınırlama kutusu sağlayarak sekiz sınıfı tespit edebilmektedir. Yöntem çok güçlü yöntemler kullanmasına ve özellik çıkarıcıları kullanmasına rağmen, bu yöntemler çatlak alanlarının hassas şekilde algılanması için uygun değildir.

Sivil altyapıların incelenmesi ile ilgili bir araştırmada, bilgisayarlı görme teknikleri ve CNN'ler kullanılarak somut tünel çatlakları üzerine bir araştırma yapılmıştır. Bu çalışmada, CNN'ler için gürültü giderme, düz çizgiler kaldırma, eğri algılama, Hough dönüşümü boyunca şekil filtreleme ve morfolojik rekonstrüksiyon da dahil olmak üzere ön işleme adımları olarak birkaç IPT kullanılmıştır. Çalışma, IPT'lerin CNN'lerin kombinasyonunda kullanılmasının daha yüksek sınıflandırma doğruluğu sağladığını kanıtlıyor.

1. INTRODUCTION

Monitoring structural health, such as detection of surface cracks is essential in identifying the anomalies in very first stages of the defect formation. Different structures like bridges, dams, pavements and power station components can benefit from this process, by reducing the risk of possible failures. A precise assessment of defects, including cracks and other anomalies can help for predicting and preventing damages which has brought the matter to researchers' attention, in order to automate this process. Automating this process will increase the inspection speed and reduce the inspection costs and possible human errors.

This project focuses on building deep learning based classification and prediction models for detecting cracks and structural failures in gas turbine components. In the long term, algorithms are planned to be deployed on a robotic arm that performs autonomous inspection on the components.

Practitioners used to rely on traditional Image processing techniques (IPTs) for detecting and extracting infrastructure defects, such as cracks in concrete or steel surfaces [1] [2] [3]. Although IPTs are powerful tools for feature identification, they still suffer from distinguishing between samples with similar features, such as cracks versus lighting spots, shadows, and edges. Hence, relying solely on traditional IPTs, limits the capability of the feature extractor and the detector. In addition, it takes a considerable amount of effort to manually extract and model the features in images, which most likely will be limited to specific features and image types.

Deep learning methods aim to address the restrictions of IPTs in extracting and learning high quality features. Using deep learning based methods are motivated by the increasing popularity of neural network based approaches in the machine learning community, which has largely replaced classical vision processing approaches that rely on hand-crafted features.

Deep Learning is a branch of Machine Learning that is based on Deep Neural Networks (with multiple hidden layers). Convolutonal Neural Networks (CNNs) are also a family of neural networks that are tailored towards processing spatial information and are deep and feed-forward (Multi-Layer Perceptrons; MLPs). CNNs are known to perform much better than traditional computer vision methods and are very useful in applications such as Image Classification, Object Detection, Segmentation, etc. [4] [5]. With the development of CNNs, detection of surface cracks can be performed without further image processing [6] [7] [8] due to the fact that they are capable of learning image features automatically. Deep learning models exploit image features in different resolutions through stacked convolutional layers, which leads toward an improved detection and classification performance.

A convolution Layer is a convolution operation, which is performed by sliding a filter or a kernel over an image and summing through element-wise multiplication of entries. A CNN consists of several different layers and parts. Pooling or mostly Max-Pooling layers are also very common in convolutional neural networks. Dropout is an option to apply regularization which will be described in detail in the following.

CNNs totally have three main parts which are input, hidden layers, and output. Input is a matrix of values which is fed to the network. Hidden Layers which are set of operations and contain two components. The first one is feature extraction and the second one is classification. Feature extraction contains convolutions, pooling and activation layers and the classification contains fully connected layers; this layers assign a probability for the object on the image being what the algorithm predicts it is. Output is a full classified form of input image. The output of the convolution is usually smaller (in width and height) than the original image.

The goal of a convolutional layer is filtering which are used to detect the presence of features within an image. Features could be anything from simple edges and curves to more complex structures and shapes. As the filter move over an image the network can effectively check for patterns in that section of the image. After we choose the filter size, we also have to choose the stride and the padding. Both the padding and stride impacts the data size. Stride controls how the filter convolves around the input volume then pads the input volume with zeros around the border. Padding allows us to design deeper networks and prevents our feature map from shrinking.Also padding actually improves performance by keeping information at the borders. Pooling Layer as mentioned before takes place in between the CNN layers. it works very much like convoluting, where we take a kernel and move the kernel over the image, the only difference is the function that is applied to the kernel and the image window which isn't linear. The function of pooling is to continuously reduce the dimensionality to reduce the number of parameters and computation in the network.

There are several types of pooling like max pooling, min pooling, and average pooling. Max pooling takes the largest value from the window of the image currently covered by the kernel and it is a cheap way of extracting feature and passing them to next layers. Although there are other pooling methods, max-pooling is generally more effective than them in object classification tasks.Min pooling takes the smallest value and average pooling as it's name shown takes the average of all values in the window.

After each convolution layer, it is convention to apply an activation layer immediately afterward. It is possible to use different activation functions for finding the best result regarding a specific task. There are two types of activation Layer, linear activation function and non-linear activation function. The most commonly used type is non-linear one. In the past, nonlinear functions like tanh and sigmoid functions were used, but researchers found out that Rectified Linear Unit (ReLU) layers work far better because the network is able to train a lot faster and be efficient in computational without making a significant difference to the accuracy. Activation function of choice for forward feeding layers of our proposed CNN is ReLU. ReLU is a function that its gradient is either 1 or 0. Moreover, Softmax function is the final activation which has been used for predicting the output. Additionally, neural networks may benefit from different types of optimizers, including but not limited to stochastic gradient descent (SGD) or ADAMAX, which has been our choice of optimizer for this research [4], [5]. For further information regarding CNN parameters, reader is referred to textbooks and literature on deep learning, such as [9].

Dropout is a regularization technique for reducing over-fitting in neural networks by preventing complex co-adaptations on training data.The term "dropout" refers to dropping out units (both hidden and visible) in a neural network. So it drops out the nodes that their probabilities are less than a specific value in each iteration. This regularization technique was not used in this research.

Data Augmentation is a regularization technique that is used to avoid over-fitting when training machine Learning models. Adjustments are made to the original images in the training dataset before being used in training. Data augmentation can be applied to any form of data to Increasing the number of images in data set by translating, cropping, scaling, rotating, changing brightness and contrast of the original image.

When the data is trained on a CNN, training and testing error and accuracy values can be measured and monitored. Training error and accuracy values demonstrates the success of model on the training data. It shouldn't been assumed that this error will be zero just because network have seen the exact data before. Test error and accuracy values are acquired from the data which have been included in training data. Thus, test accuracy and error values are more reliable for evaluation of generalization purpose. However, the third portion of the dataset, also known as validation set, should only be relied to validate the model, as the test set have been indirectly used for tuning the CNN and cannot provide us with valid error and accuracy values.

Here are some tips which can help in tuning the CNN model to achieve a better performance

- If both the training error and test error is high then our model is probably under-fitting so we should increase model capacity (more neurons and layers in grid search) or in the other case increase training data.

- If training error is low but test error is high so model is probably over-fitting and should be regularized and/or decrease model capacity or in the other case increase training data or maybe the test and training images are similar.

- If both training and test error is low we could achieve a successful network.

Confusion Matrix also is another method for monitoring the errors and accuracy of networks. It determines the false positive, false negative, true positive and true negative.

- True Positive (TP) means when the network predicts the positive condition correctly.

- True Negative (TN) means when the network predicts the negative condition correctly.

- False Positive (FP) means when the network predicts the negative condition incorrectly.
- False Negative (FN) means when the network predicts the positive condition incorrectly.

1.1 Purpose of Thesis

In this work, we propose a method utilizing CNNs in combination with IPTs for detecting cracks in gas turbine component surfaces. To the best of our knowledge, our utilization of IPTs as a pre-processing technique for CNNs is being done for the first time in the structural health monitoring. This approach provides more robust and accurate results, regardless of the dataset quality, in reasonable computational time. utilizing IPTs as pre-processing step of CNNs improves learning speed. Handpicking some features through IPTs, will enable the CNNs to learn faster by reducing the amount of features to learn.

1.2 Literature Review

The works on crack detection can be divided into two major groups, methods that rely on image processing and convolutional neural network models.

1.2.1 Image processing techniques

In a study by Qin Zou et al. [2], shadow removal was exploited in detecting pavement cracks. In their method, shadow removal has been implemented through balancing the illumination of shadow region and the other parts. Likelihood of having shadows similar to pavement shadows is near-zero and there are other geometric and surface complexities such as having 3D surface areas, shapes and textures in the structures of interest in our work. Thus, implementation of such technique in this work will distort crack features instead.

Thresholding, which is considered as one the most widely used methods of segmentation, was utilized in [1]. Although thresholding is one of the most widely used IPTs, it has its own drawbacks when dealing with more complex surfaces. The study also implements median filtering, after conversion of color image to gray-scale, in order to reduce the surface complexity. However, the suggested techniques in this study for elimination of unwanted elements are insufficient to deal with more complex and large elements.

In another study [10], steel cracks were detected using a vision based detector. Despite the using crack-like features in the dataset of the study, the authors propose a detector based on Frangi filter and Hessian matrix edge detection. The proposed detector is able to successfully detect bolts and cracks in bridge images. In general, utilization of vision based techniques requires huge amount of feature engineering and are difficult to re-implement.

1.2.2 CNN based techniques

The CNN based crack detection techniques are mainly divided into two groups. The first group (block-level detection) is based on detection of crack patches and providing bounding boxes over them. The second group (pixel-level detection) is based on crack segmentation (delineation), which provides pixel level crack detection of cracks. There have been very recent studies in pixel-level detection category, including [11] and [12] and there also have been prior works like [13].

A deep learning approach for detecting cracks on concrete images was proposed in [6]. In this study, the results are compered to Canny and Sobel edge detection algorithms, which suggests that the proposed CNN improves the accuracy. Utilization of CNNs increases the generalization, and reduces the effect of noise in data. The author claims over 97% test accuracy in their dataset.

Likewise, a CNN model was implemented in [14]. The proposed CNN architecture has four convolution layers, a maxpooling layer and two dense layers. Compared to classical machine learning methods such as Support Vector Machines (SVM) and Boosting methods, the proposed CNNs provide much better accuracy and recall scores (0.925) on a pavement crack detection dataset.

In a study of road damage inspection [15], smart phones were used for gathering the data from road surface. The proposed method is able to detect eight classes through providing a bounding box over the area of interest. Although the method utilizes very powerful methods and feature extractors, these methods are not suitable to precise detection of crack areas.

In a study [16], a different implementation of CNNs is proposed. In this implementation, extracting and forward feeding of low level features as the first layer of the CNN is suggested. This implementation, yields 88.6% accuracy.

In an investigation on inspection of civil infrastructures [17], Stentoumis et al. conducted a research on concrete tunnel cracks, using computer vision techniques and CNNs. In this study, several IPTs have been used as pre-processing steps for CNNs, including noise removal, straight lines removal, curve detection, shape filtering through Hough transform and morphological reconstruction. The study proves that utilization of IPTs in combination of CNNs yields higher classification accuracy.

In the study by Futao Ni [12], a fine-tuned GoogLeNet is used with combination of a crack delineation network (CDN). GoogLeNet, as a successful feature extractor, provides the feature extraction and detection in this pipeline. The dataset image size in this research is 4000×6000 pixels, which is considered a high resolution dataset as opposed to our dataset which has the maximum resolution of 1200×2000 pixels. citeYang2018 and [13] also utilize similar approaches of pixel level crack delineation. These studies provides segmentation for crack patches, which is not the focus in our study.

The focus of this study is detection of image patches or grids, which fits into the first group (block-level detection) of studies. Distinguishing cracks from non-cracks is done using a CNN based classifier and a sliding window, which marks the detected patches as demonstrated in the visualization section, in Figure 2.1. The provided framework focuses on achieving high accuracy by a flexible model which is capable The works on crack detection can be divided into two major groups, methods that rely on image processing and convolutional neural network models.

1.2.3 Image processing techniques

In a study by Qin Zou et al. [2], shadow removal was exploited in detecting pavement cracks. In their method, shadow removal has been implemented through balancing the illumination of shadow region and the other parts. Likelihood of having shadows similar to pavement shadows is near-zero and there are other geometric and surface complexities such as having 3D surface areas, shapes and textures in the structures of interest in our work. Thus, implementation of such technique in this work will distort crack features instead.

Thresholding, which is considered as one the most widely used methods of segmentation, was utilized in [1]. Although thresholding is one of the most widely used IPTs, it has its own drawbacks when dealing with more complex surfaces. The study also implements median filtering, after conversion of color image to gray-scale, in order to reduce the surface complexity. However, the suggested techniques in this study for elimination of unwanted elements are insufficient to deal with more complex and large elements.

In another study [10], steel cracks were detected using a vision based detector. Despite the using crack-like features in the dataset of the study, the authors propose a detector based on Frangi filter and Hessian matrix edge detection. The proposed detector is able to successfully detect bolts and cracks in bridge images. In general, utilization of vision based techniques requires huge amount of feature engineering and are difficult to re-implement.

1.2.4 CNN based techniques

The CNN based crack detection techniques are mainly divided into two groups. The first group (block-level detection) is based on detection of crack patches and providing bounding boxes over them. The second group (pixel-level detection) is based on crack segmentation (delineation), which provides pixel level crack detection of cracks. There have been very recent studies in pixel-level detection category, including [11] and [12] and there also have been prior works like [13].

A deep learning approach for detecting cracks on concrete images was proposed in [6]. In this study, the results are compered to Canny and Sobel edge detection algorithms, which suggests that the proposed CNN improves the accuracy. Utilization of CNNs increases the generalization, and reduces the effect of noise in data. The author claims over 97% test accuracy in their dataset.

Likewise, a CNN model was implemented in [14]. The proposed CNN architecture has four convolution layers, a maxpooling layer and two dense layers. Compared to classical machine learning methods such as Support Vector Machines (SVM) and Boosting methods, the proposed CNNs provide much better accuracy and recall scores (0.925) on a pavement crack detection dataset.

In a study of road damage inspection [15], smart phones were used for gathering the data from road surface. The proposed method is able to detect eight classes through providing a bounding box over the area of interest. Although the method utilizes very powerful methods and feature extractors, these methods are not suitable to precise detection of crack areas.

In a study [16], a different implementation of CNNs is proposed. In this implementation, extracting and forward feeding of low level features as the first layer of the CNN is suggested. This implementation, yields 88.6% accuracy.

In an investigation on inspection of civil infrastructures [17], Stentoumis et al. conducted a research on concrete tunnel cracks, using computer vision techniques and CNNs. In this study, several IPTs have been used as pre-processing steps for CNNs, including noise removal, straight lines removal, curve detection, shape filtering through Hough transform and morphological reconstruction. The study proves that utilization of IPTs in combination of CNNs yields higher classification accuracy.

In the study by Futao Ni [12], a fine-tuned GoogLeNet is used with combination of a crack delineation network (CDN). GoogLeNet, as a successful feature extractor, provides the feature extraction and detection in this pipeline. The dataset image size in this research is 4000×6000 pixels, which is considered a high resolution dataset as opposed to our dataset which has the maximum resolution of 1200×2000 pixels. citeYang2018 and [13] also utilize similar approaches of pixel level crack delineation. These studies provides segmentation for crack patches, which is not the focus in our study.

The focus of this study is detection of image patches or grids, which fits into the first group (block-level detection) of studies. Distinguishing cracks from non-cracks is done using a CNN based classifier and a sliding window, which marks the detected patches as demonstrated in the visualization section, in Figure 2.1. The provided framework focuses on achieving high accuracy by a flexible model which is capable of performing, regardless of image size and resolution. In addition, data preparation for block-level detection is relatively an easy task. In contrast, segmentation needs high quality images and more complicated annotation process. Therefore, pixel-level detection is not the focus and priority in this study.

Utilization of CNNs in crack detection brings a great improvement over IPTs. However, these implementations do not provide necessary precautions to reduce surface noises and do not sharpen particular features. Our implementation, provides solutions to overcome the existing shortcomings and improve generalization accuracy and training speed.

2. INTEGRATED DATA

2.1 Data Preparation

The research dataset has initially been provided in form of batches of images from defected parts and structures of gas turbines. Dataset images have varying resolutions and different crack sizes and types. Hence, a custom software was designed to further automate the annotation and data preparation. The software is designed in a way to handle different resolutions and crack sizes. The software takes block samples on mouse track line as user input, in every 5 to 10 pixels of mouse movement, as illustrated in Figure 2.1. This ensures that the proportion of a crack size to the bounding block size are consistent. Additionally, this data annotation method brings the flexibility of having different block sizes and class balances. The dataset is acquired from about 700 images, resulting in 250k labeled image blocks.

2.2 Pre-processing

2.2.1 Data augmentation

Data Augmentation is a regularization technique that is used to avoid over fitting by producing variations in the training dataset. Data augmentation can be applied to any form of data and increase the number of images in dataset by translating, cropping, scaling, rotating, changing brightness and contrast of the original image. This technique improves generalization of the model. The list of applied transformations on images have been provided in Table 2.1.

2.2.2 Smoothing method

Image processing operations can help distinguish the features easier and also eliminate noise from images. Thus, some image processing techniques were implemented in this work to investigate and potentially improve the results. Due to the nature of inspected

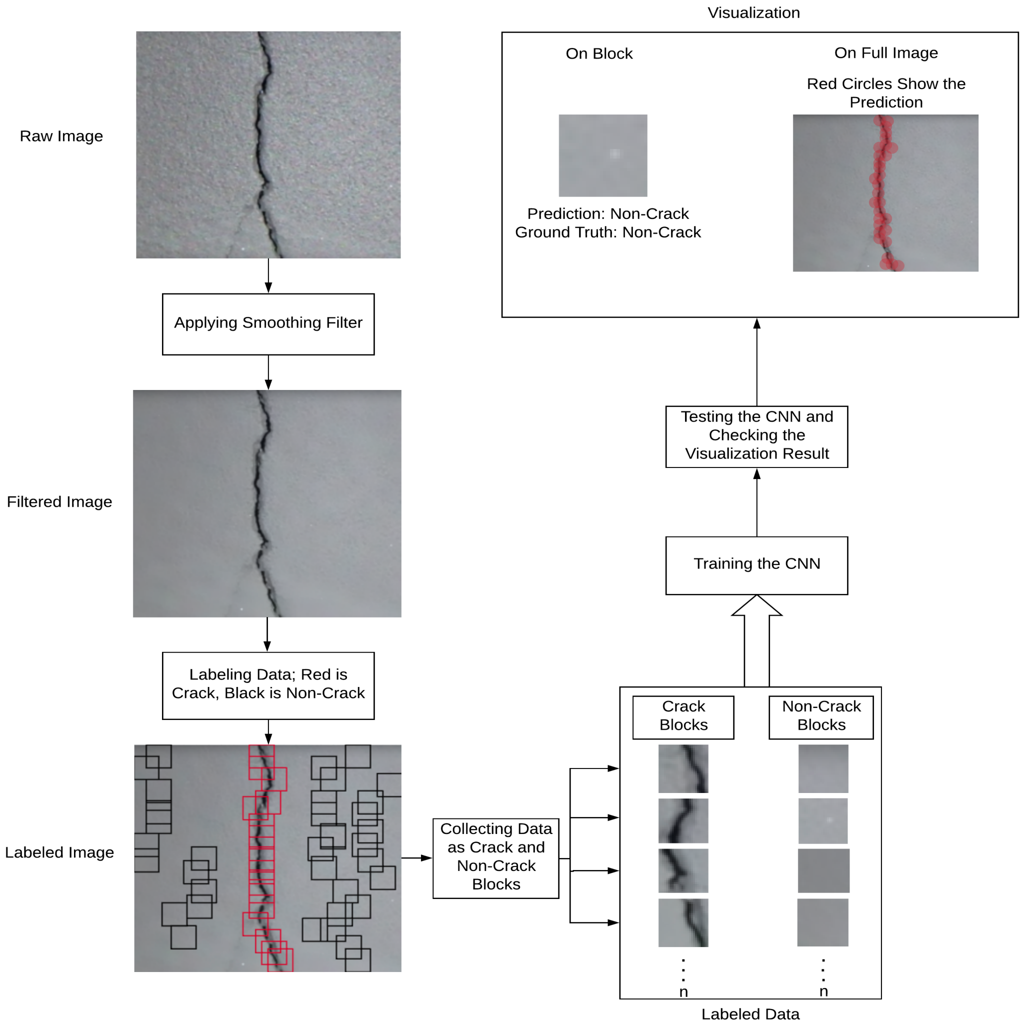

Figure 2.1 : Work Progr ess Visual Chart; the process is visually shown step by step starting from raw image and ending with visualization part.

Table 2.1 : Augmentation parameters used in model training.

Transformation Type	Range
Rotation	20 degrees
Width Shift	0.2
Height Shift	0.2
Horizontal Flip	True

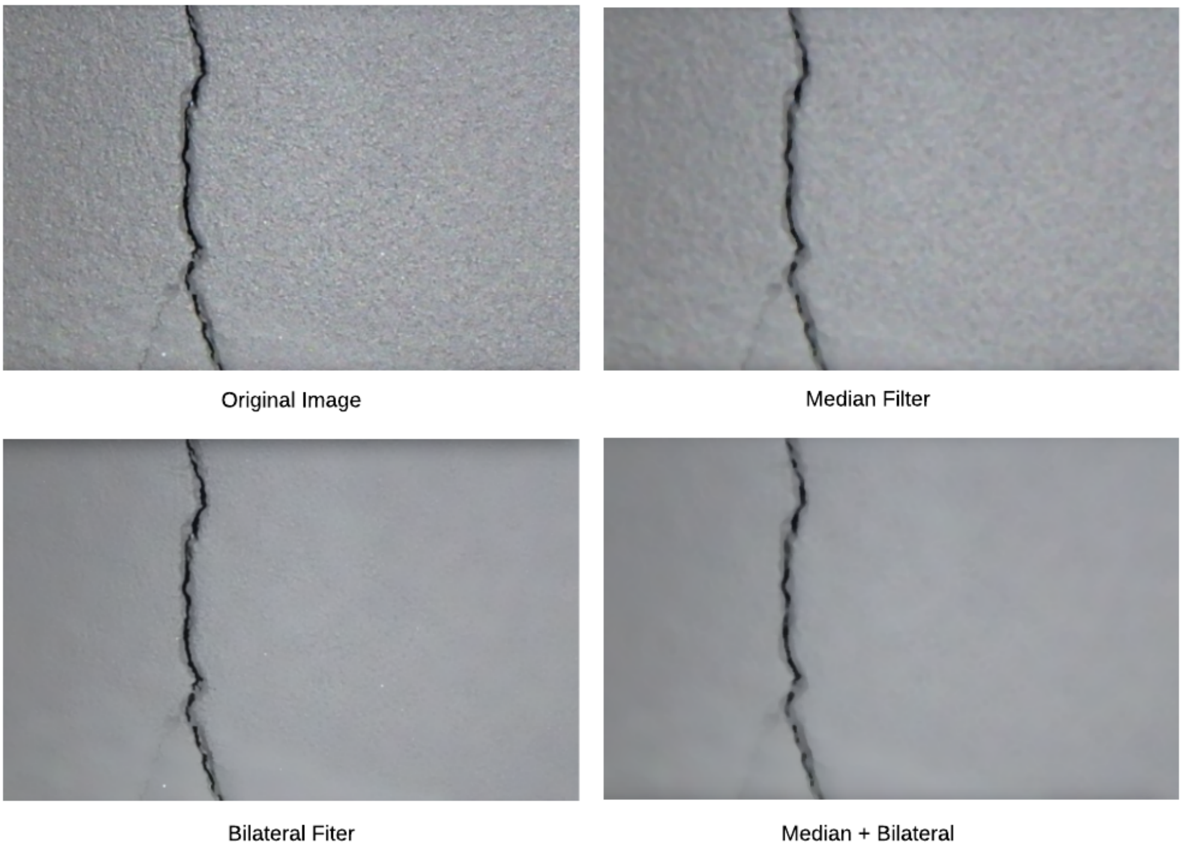

Figure 2.2 : Application of Median and Bilateral filters.

surfaces, median and bilateral filters are a good fit for this operation. These filters and their usages are as follows.

Median Filter: Median filter is a smoothing method, mostly used for salt-and-pepper noise removal [1] [18]. In images, it's expected to have similar pixel values in neighbouring pixels, but the values of noise do not follow any order and are not similar to others in a particular area of pixels. Therefore, using average value of neighboring pixels in the corrupted or noisy pixel tends to be a good smoothing method. Median filter does this process by replacing each pixel with the median value of the neighboring pixels. Thus, it is potentially a good method to smooth out the small bright and dark points on an image, without having to blur the image any further. Effect of the Median filter on turbine component images is demonstrated in Figure 2.2.

Bilateral Filter: Bilateral filtering is introduced by Tomasi and Manduchi [19] to replace anisotropic diffusion [20] in which they average particular regions by solving partial differential equations. Bilateral filter is a non-iterative, non-linear and edge-preserving filter. This filter, can reduce image's contrast, while preserving the details. This filter performs smoothing on the images with low pass filters and applies noise removal via high-pass filters (HPF). HPFs are helpful in edge detection, which help at preserving the edges. A smoothed image by bilateral filter is demonstrated in Figure 2.2. Output of the bilateral filter applied for the input image $f(x)$, is calculated as in equation (2.1), where closeness function $c(\varepsilon,x)$ measures the geometric closeness between the neighborhood center x and a nearby point ε. The similarity function $s(f(\varepsilon),f(x))$ measures the photometric similarity between the pixel at the neighborhood center x and a nearby point ε. Additionally, normalization value can be obtained using equation (2.2).

$$h(x) = k^{-1}(x) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\varepsilon) c(\varepsilon,x) s(f(\varepsilon), f(x)) d\varepsilon \tag{2.1}$$

$$k(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\varepsilon,x) s(f(\varepsilon), f(x)) d\varepsilon \tag{2.2}$$

Utilization of both filters benefits the overall image quality as illustrated in Figure 2.2, but Due to the characteristics of this study, Bilateral filter outperforms other smoothing methods in both edge preserving and blurring grounds.

2.2.3 Application of smoothing methods

The dataset comprises images with various qualities and resolutions. As a result, filters have to adapt to image size. In the implementation of median filtering, smoothing was applied after generating block dataset. Thus, it is not necessary to provide a kernel size estimation. The median filter kernel size is a fixed value for all blocks. However, bilateral filtering should be applied prior to the block generation step. This is due to the edge preserving property of this filter. Similar to block size estimation, kernel size of the smoothing filter is also variable with respect to the resolution of input image. To overcome this issue, a function with empirical upper and lower bounds was defined to estimate a kernel size for bilateral filter. Generally, it is expected to have the filter size as a fraction of image size. However, this method failed to provide proper smoothing solution. Therefore, another method is being used for defining kernel size, which has been shown to work better. The equations (2.3) and (2.4) provide a suitable filter size for bilateral smoothing. Other values were provided empirically in this study.

$$Ratio = (resMax - resMin)/(ImageWidth - resMin) \tag{2.3}$$

$$KernelSize = ImageWidth/[((sizeMax - sizeMin)/Ratio) + (sizeMin)] \tag{2.4}$$

Where $r$esMin is 400 pixels and $r$esMax is 1920 pixels. $s$izeMax and $s$izeMin are 75 and 30 respectively.

3. CONVOLUTIONAL NEURAL NETWORKS

3.1 Classification Using CNN

Several different CNN architectures were examined for this work, differing in their hyper-parameters such as number and parameters of convolutional layers, max pooling layers, number and dimensions of dense layers, etc.

Table 3.1 : Grid search Parameters; grid search algorithm iterates over the provided range of the hyper parameters in this table.

Hyper Parameters	Range
Number of Convolution Layers	2 - 5
Convolution Layer kernel Range	2 - 12
Range of First Layer Kernel Number	20 , 25 , 30 , 32
Number of Max Pooling Layers	1 - 3
Number of Dense Layers	1 - 4
Range of Dense Layer Neurons (except last layer)	100, 1000, 3125, 3136, 3200
Optimizers	SDG , Adamax

Finding optimal hyper-parameters to achieve higher model accuracy, requires CNN architecture tuning. However, manual tuning is only possible for small and moderate size networks, with limited configurable hyper-parameters. For this study, an automated hyper-parameter tuning algorithm was implemented to obtain the architecture with highest accuracy. Hyper-parameter search was conducted via a grid search in order to minimize further changes and reduce the time required to find an optimal CNN architecture. The grid search simply iterates over all combinations of provided architecture hyper-parameters and returns the architecture with highest test accuracy.

Although the network parameters are optimized using grid search, an initial architecture and parameter range should be specified based on the problem definition. As discussed earlier, the problem being tackled is detecting cracks in gas turbine components. In contrary to similar studies, finding cracks in gas turbine components bears different type of challenges. These components may carry many structural shapes and have features very similar to cracks that are undesired.

Detecting cracks from digital images, requires the classifier to classify image segments as cracks and non-cracks. The input size of the network is the first parameter we consider. There are more than one limiting factor for selecting a proper input size. Typically, input size might be constrained for performance of a network and also training/inference speed factor. In addition to the effect of input size on the CNN, this study also has a maximum constraint for the sliding window size, which is use for ensuring acceptable visualization results and precision in detection of cracks. However, very small inputs may also fail to fulfill the performance metrics by dropping too much valuable data and introducing noise. 40×40 input size was selected for acceptable classification performance and visualization results. In order to determine the number and dimension of convolutional and max pooling layers, a range of possible parameters was generated with regards to the input size, which later were fed into the grid search algorithm for selecting the best configuration.

Table 3.1 provides range of the hyper-parameters used in the grid search algorithm. It is worth mentioning that the convolutional layers are all implemented in pyramid shape and the grid search algorithm was designed to only follow this design. In this pyramid design, number of filters for following layers of the first layer were multiplications of 20, 25, 30 or 32. Additionally, dense layers were given between 100 to 3200 neurons.

The final architecture selected by grid search is illustrated in Fig. 3.1. In the final network architecture, all convolutional layers have stride of 1×1 and all max pooling layer have stride of 2×2. The numbers under the blocks indicate number of filters / neurons of each layer. The network has 2 convolutional layers, 2 max pooling layers and 2 dense layers. First dense layers are followed by ReLU and the last one with a softmax layers.

3.2 Grid Search

An automated parameter tuning algorithm was implemented to obtain the architecture with highest accuracy. Parameter search was conducted via a grid search in order to minimize further changes and reduce the time required to find an optimal CNN

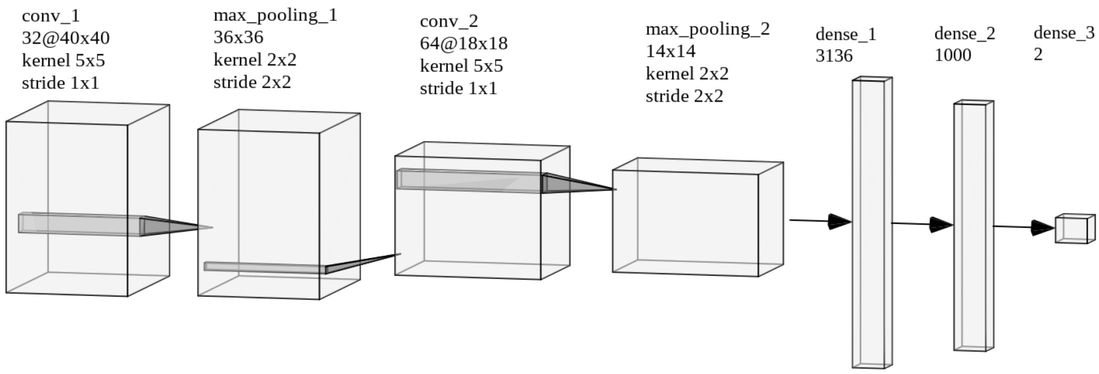

Figure 3.1 : Final CNN architecture; containing 2 convolution, 2 maxpooling and 3 dense layers.

architecture. The grid search simply iterates over all possible combinations of architecture parameters and returns the architecture with highest test accuracy.

3.3 Visualization

Detection of cracks is applied via a sliding window, which moves over images and classifies each window separately. In particular, this method has also been implemented in other works like [21], [22], where each component image is broken into blocks and classified as a crack/non-crack using a sliding window technique.

Hence, higher detection and visualization accuracy will require a window with smaller size, which results in generation of more windows to process and more classifications. Consequently, this will have an unpleasant effect on detection speed of images with higher resolution.

The final detector is expected to examine images with varying resolutions and qualities. These images may include any type of cracks, including large or small cracks, which makes it difficult to have a single window size for any given image. Thus, the window size was designed to adapt to the resolution. However, this will only solve the resolution problem, yet the crack size (camera distance from the crack) problem might still persist. In order to improve the detection performance, a pre-processing step has been embedded into the detector. Figure 3.2 provides few detection samples, with and without pre-processing (smoothing).

A good visualization, heavily relies on low amount of false positives and false negatives. Aside from the CNN configuration, classification confidence can also have

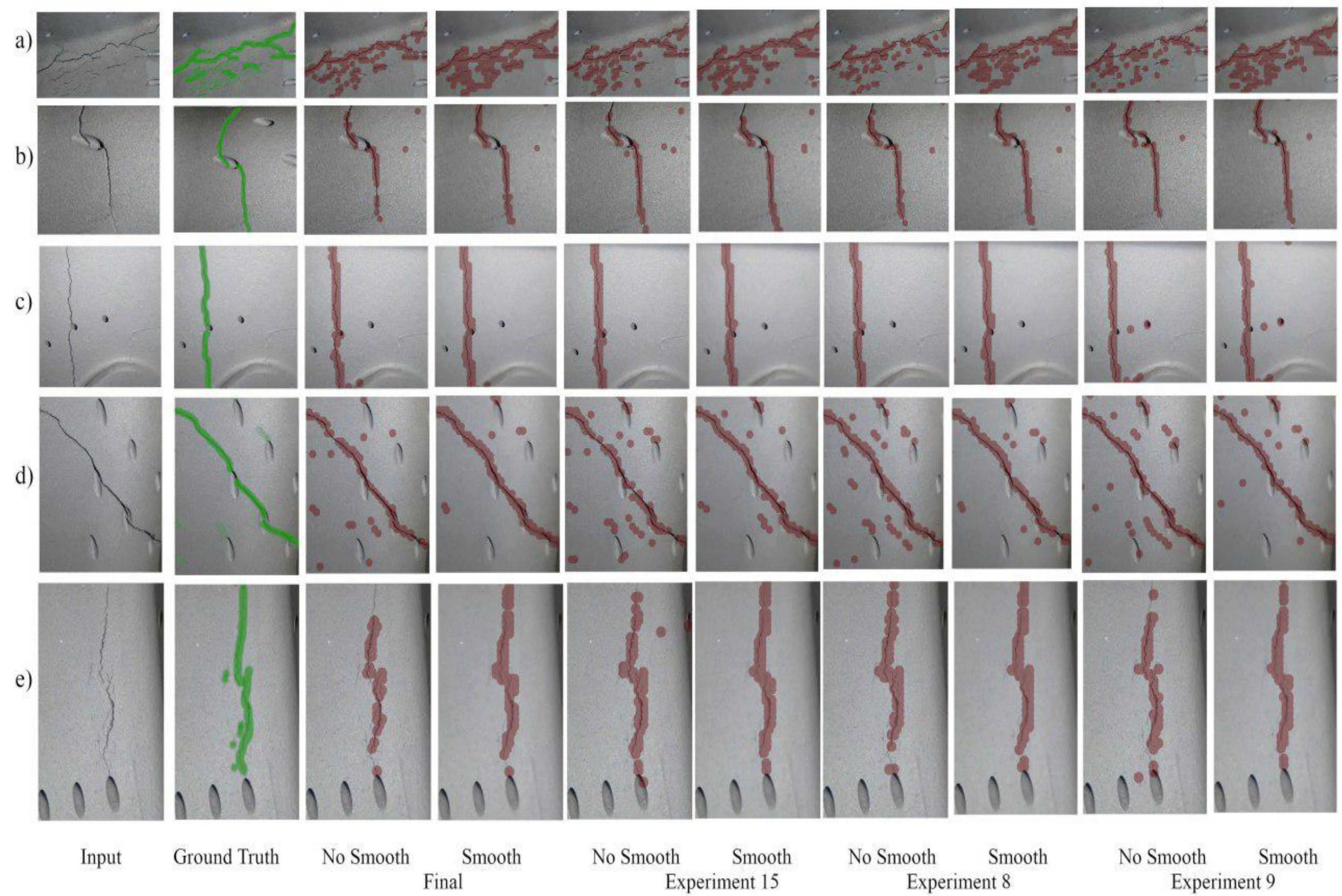

Figure 3.2 : Sample results: Predictions have been marked with a red shade on both smoothed and original input images. Ground truth cracks have been marked with two different green shades. The brighter green shade emphasizes the obvious cracks, which the model should not miss.

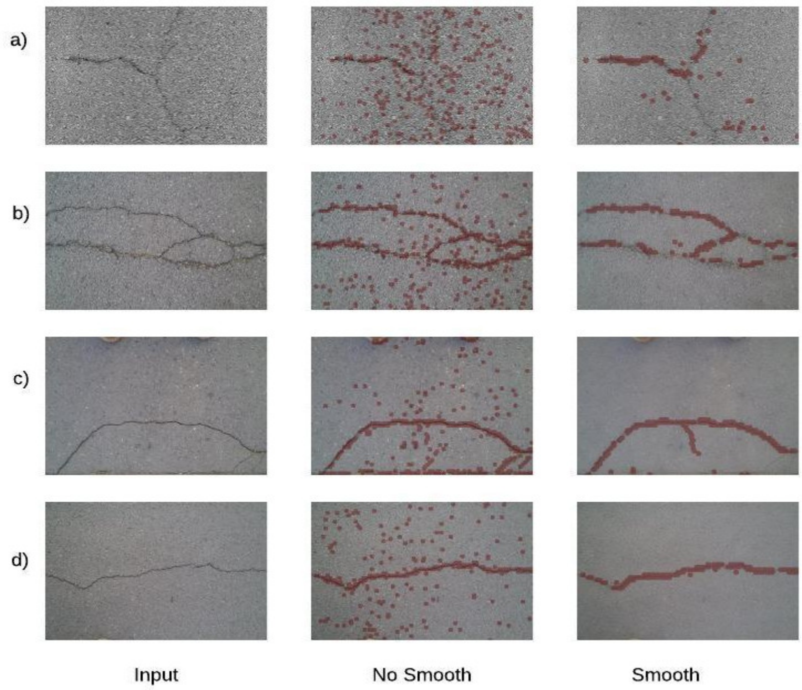

Figure 3.3 : Results on different dataset: Predictions have been marked with a red shade on both smoothed and non smoothed input images.

an important impact on visualization. Higher confidence threshold value for the model output, might result in higher number of false negatives. On the other hand, a high threshold value might reduce the number of false positives. Based on the experiments, 0.9 has been selected as the best confidence threshold value for detection among our models and visualization techniques.

4. RESULT

4.1 Results

Results of the grid search experiments are indexed in Table 1. The columns of this table are variables of the experiments, which are pre-processing, augmentation and the architecture of the network. The column "Filter", refers to the pre-processing method used in the experiment. Some experiments have benefited from Bilateral or Median smoothing, which have been recorded in this column. "Augmentation" column tells whether the model has been trained with augmentation applied or not. "CNN Architecture / Dense Architecture" briefly illustrates the architecture of the networks. All convolution layers in the networks have stride of 1×1 and pooling layers are maxpoolings with stride of 2×2.

In order to make the comparison valid, all experiments were conducted in similar environments, using the same dataset, and having the same network input size of 40×40 pixels. The results can be analyzed from different perspectives: Quality of dataset, parameters of layers, data augmentation, smoothing and finally, effect of data inversion.

- It could be observed that the balance of data had tremendous impact on the model accuracy. The test accuracy results improved to over 90% after changing the number of samples in different classes. Different dataset scales were tested and finally it was concluded that a dataset size with bias toward non-crack image dataset size provides much better results and lower number of false positives. There are multiple reasons behind this result. Initially, having more non-crack samples generates higher probability of non-crack detection, which represents the real world conditions. Additionally, non-crack objects are more complex and have higher variations in contrast to crack objects. Thus, collecting more non-crack samples yield better classification performance.

- Regarding the number of layers and other hyper-parameters, models with larger kernel size were observed to be less sensitive to size of the sliding window, which deteriorates the classification performance. For instance, comparing rows 1, 2 and 3, to the models represented in rows 4, 5 and 6 at Table 1 are less sensitive to the size of the sliding window and they provide even better results after smoothing. The reason behind this behaviour is having cracks that are wider than the kernel size, which results in learning shades instead of cracks. This is believed to result in extracting wrong features in smaller kernel sizes, especially at the initial CNN layer.

- The original dataset does not cover all types of cracks including different shapes and angels. Data augmentation, in addition to increasing the amount of samples in the dataset, increases the variance in dataset by randomly generating some of the lacking crack shapes and angles. This regularization technique boosts accuracy as demonstrated in Table 1. Data augmentation can result in over 1.1% of test accuracy improvement, based on our results. Therefore, data augmentation can also be considered as an important factor to consider in similar tasks.

- The results revealed that effective smoothing on visualization has positive effect on detecting standard cracks as shown in Figure 3.2. In addition, performing smoothing on the dataset prior to training increases the validation accuracy significantly. Such smoothing operations can improve test accuracy as much as 5.97%, like experiments 1 and 2 in Table 1. However, smoothing might also reduce the chances of detecting minor cracks.

- Because the cracks appear as dark parts within bright regions, it might seem helpful to invert the color of samples in the data, so the maxpooling layers can pick the cracks features. However, applying inversion to the samples resulted in less than $\pm 1\%$ of variation in accuracy. In some cases, the results were affected negatively. Thus, it can be concluded that the inversion of data has no clear advantage for our experiments.

Figure 3.2 illustrates different visualizations outputs in the experiments. Details regarding the referred experiments at the bottom of the Figure 3.2, can be found in Table 1. As demonstrated in the Figure 3.2, experiment 4 is the experiment with the best result. In this experiment, not only false positive parts are reduced in comparison to the previous models, detecting cracks can be performed with higher success rate. For instance, in row ’c’ of the experiment 9, some false positive errors can be detected on the holes, which is eliminated in the final result. Our proposed method successfully achieved an average test accuracy of 96.26%. The performance of the final model is represented by accuracy, precision, recall and F1 score metrics in Table 4.2.

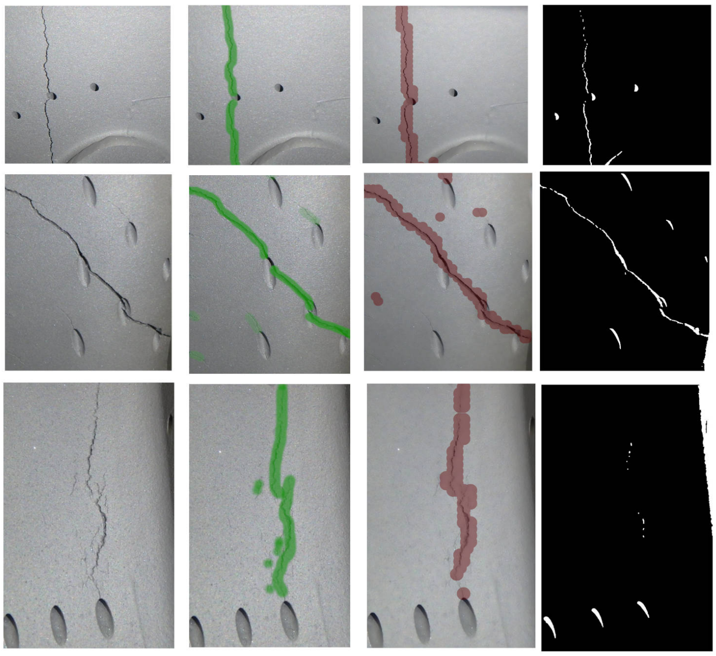

Figure 4.1 : Sample Results: Comparison of double thresholding method [1] (Thresholding column) and the proposed method (CNN column) in this work.

For validation, method was tested on another dataset (CrackForest dataset) consisting of 155 images of pavement cracks which is available in [23]. It is shown that the proposed method can generalize to this datasets as well, and can detect cracks with a similar accuracy of 96.01%. The visualization result is shown in Figure 3.3.

Method developed by Cha et al. in [6] reports an overall accuracy of 97.4%. However, the CNN model in the aforementioned paper yield an accuracy of 82.15% on our dataset. We suspect that this difference is due to having a large network input size and a dataset consisting only of high resolution images in the original paper. In order to remedy the resolution problem, we modified the CNN in Cha’s work in a way to get input sizes of 128, instead of 256 pixels. As a result, an overall accuracy of 88.26%

Table 4.1 : Comparison of the proposed method and state of the art method; in the second and third rows comparison is made on methods, and the forth row is about comparing datasets.

Method	Dataset	Input Size	Network Architecture	Accuracy
Proposed method	GE	40	$conv(32,5,1) + pooling(2,2)$ $+conv(64,5,1) + pooling(2,2)$ | dense:3136|1000|2	96.26%
[6]	GE	256	$conv(24,20,2) + pooling(7,2)$ $+conv(48,15,2) + pooling(4,2)+$ $conv(96,10,2)$ | dense: 96|2	82.15%
Modified [6]	GE	128	$conv(24,10,2) + pooling(3,2)$ $+conv(48,7,2) + pooling(2,2)$ $+conv(96,5,2)$ | dense:96|2	88.26%
Proposed method	[23]	40	$conv(32,5,1) + pooling(2,2)$ $+conv(64,5,1) + pooling(2,2)$ | dense: 3136|1000|2	96.01%

was achieved. Table 4.1 illustrates the differences in architectures and input sizes of two networks.

Comparing the proposed method to classic IPTs, also shows the advantage of implementing deep learning methods for detecting cracks. In order to validate this claim, the method suggested in [1] was implemented on gas turbine components images. Implementing a method based on thresholding can generate the exact shape and mask of the crack. However, the examined surfaces in this work are relatively more complex and they carry features very similar to cracks in terms of shape and color, like holes, bents and dents of the components. Figure 4.1 is an example of inefficiency of thresholding method in this case. Figure shows that it is not possible to detect a feature, if shade of the feature is the same as crack shade. Therefore, it can be concluded that simple image processing methods does not suffice and a deep learning usually yields better results. Although the proposed method by Ni and Zhang in [12] is a segmentation method, it utilizes a GoogLeNet CNN feature extractor and detector prior to segmentation. To make a fair comparison, this feature extractor was implemented in a very similar manner on our previously annotated dataset. The GoogLeNet was trained in a transfer learning setup as described in the paper, with ImageNet weights. As expected, the network input size made the model very resource intensive and the training was comparably slow in contrast to our proposed method.

Table 4.2 : Performance evaluation of the proposed framework.

Evaluation Metrics	
Accuracy	96.26%
Precision	92.18%
Recall	87.13%
F1 score	0.8958

Similar to the work by Cha et al., the large CNN input size (224×224), slows the converging down dramatically to the point of stopping. Therefore, the final result is not shared and compared in this section.

4.2 Conclusions

In this paper, a framework for detecting cracks in different structures is proposed. The proposed framework can be summarized in a 3 stage pipeline, which takes a raw image as input and marks the cracks on the output image. The pipeline stages are pre-processing, sliding window and detection of crack in window patches. Detection stage utilizes a CNN which has been trained with 700 annotated and pre-processed gas turbine images. The trained model achieves an accuracy of 96.26% on test set.

In this study, the impact of pre-processing on a detection pipeline has been investigated. The results show that bilateral filtering improves generalization of the detector on surfaces with complex textures. The proposed detection framework in this study can be implemented in different surfaces, with minimal changes.

These result warrant further investigation into IPTs as pre-processing methods, in order to achieve robust and computationally cheaper techniques. In the future work, it is planned to implement advanced hyper-parameter optimization methods such as randomized search methods to enable architecture search over larger parameters spaces (which is currently not feasible by grid search). Additionally, replacing the sliding window with a more advanced detector will provide faster and more accurate detection in the visualization stage.

REFERENCES

[1] **Li, P.**, **Wang, C.**, **Li, S. and Feng, B.** (2016). Research on crack detection method of airport runway based on twice-threshold segmentation, *Proceedings - 5th International Conference on Instrumentation and Measurement, Computer, Communication, and Control, IMCCC 2015*, 1716–1720.

[2] **Zou, Q.**, **Cao, Y.**, **Li, Q.**, **Mao, Q. and Wang, S.** CrackTree: Automatic crack detection from pavement images, *Pattern Recognition Letters*, (3), 227–238.

[3] **Oliveira, H. and Correia, P.L.** (2009). Automatic road crack segmentation using entropy and image dynamic thresholding, *European Signal Processing Conference*, (Eusipco), 622–626.

[4] **Fan, Z.**, **Wu, Y.**, **Lu, J. and Li, W.** (2018). Automatic Pavement Crack Detection Based on Structured Prediction with the Convolutional Neural Network, *arXiv preprint arXiv:1802.02208*.

[5] **Stentoumis, C.**, **Protopapadakis, E.**, **Doulamis, A. and Doulamis, N.** (2016). A Holistic Approach for Inspecton of Civil Infrastructures Based On Computer Vision Techniques., *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, *41*.

[6] **Cha, Y.J.**, **Choi, W. and Büyüköztürk, O.** Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks, *Computer-Aided Civil and Infrastructure Engineering*, (5), 361–378.

[7] **Krizhevsky, A.**, **Sutskever, I. and Hinton, G.E.** (2012). Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, pp.1097–1105.

[8] **Zhang, L.**, **Yang, F.**, **Zhang, Y.D. and Zhu, Y.J.** (2016). Road crack detection using deep convolutional neural network, *Image Processing (ICIP), 2016 IEEE International Conference on*, IEEE, pp.3708–3712.

[9] **Chollet, F.** (2017). *Deep learning with python*, Manning Publications Co.

[10] **Yeum, C.M. and Dyke, S.J.** (2015). Vision-Based Automated Crack Detection for Bridge Inspection, *Computer-Aided Civil and Infrastructure Engineering*, *30*(10), 759–770.

[11] **Yang, X.**, **Li, H.**, **Yu, Y.**, **Luo, X.**, **Huang, T. and Yang, X.**, (2018), Automatic Pixel-Level Crack Detection and Measurement Using Fully Convolutional Network.

[12] **Ni, F. and Zhang, J.** (2018). Pixel - level crack delineation in images with convolutional feature fusion, (September), 1–18.

[13] **Ronneberger, O.**, **Fischer, P. and Brox, T.** (2015). U-net: Convolutional networks for biomedical image segmentation, *International Conference on Medical image computing and computer-assisted intervention*, Springer, pp.234–241.

[14] **Zhang, L.**, **Yang, F.**, **Daniel Zhang, Y. and Zhu, Y.J.** Road crack detection using deep convolutional neural network, *2016 IEEE International Conference on Image Processing (ICIP)*, (September), 3708–3712.

[15] **Maeda, H.**, **Sekimoto, Y.**, **Seto, T.**, **Kashiyama, T. and Omata, H.** (2018). Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone, *arXiv preprint arXiv:1801.09454*.

[16] **Makantasis, K.**, **Protopapadakis, E.**, **Doulamis, A.**, **Doulamis, N. and Loupos, C.** (2015). Deep Convolutional Neural Networks for efficient vision based tunnel inspection, *Proceedings - 2015 IEEE 11th International Conference on Intelligent Computer Communication and Processing, ICCP 2015*, 335–342.

[17] **Stentoumis, C.**, **Protopapadakis, E.**, **Doulamis, A. and Doulamis, N.** (2016). A holistic approach for inspection of civil infrastructures based on computer vision techniques, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *41*(July), 131–138.

[18] **Chan, R.H.**, **Ho, C.W. and Nikolova, M.** (2005). Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization, *IEEE Transactions on Image Processing*, *14*(10), 1479–1485.

[19] **Tomasi, C. and Manduchi, R.** (1998). Bilateral filtering for gray and color images, *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 839–846.

[20] **Perona, P. and Malik, J.** Scale-space and edge detection using anisotropic diffusion, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (7), 629–639, `1102.0183`.

[21] **Eisenbach, M.**, **Stricker, R.**, **Seichter, D.**, **Amende, K.**, **Debes, K.**, **Sesselmann, M.**, **Ebersbach, D.**, **Stoeckert, U. and Gross, H.M.** (2017). How to get pavement distress detection ready for deep learning A systematic approach, *Proceedings of the IEEE International Joint Conference on Neural Networks, Anchorage, AK, USA*.

[22] **Pauly, L.**, **Hogg, D.**, **Fuentes, R. and Peel, H.** (2017). Deeper networks for pavement crack detection, *Proceedings of the 34th ISARC*, IAARC, pp.479–485.

[23] **Shi, Y.**, **Cui, L.**, **Qi, Z.**, **Meng, F. and Chen, Z.** (2016). Automatic road crack detection using random structured forests, *IEEE Transactions on Intelligent Transportation Systems*, *17*(12), 3434–3445.

APPENDICES

APPENDIX A.1 : Experiences Table

Table 1 : Test accuracy results for different network hyper-parameters and method parameters. Filter column expresses the applied pre-processing step filter type. Augm. column defines the augmentation step within training. Inverse column expresses whether there there have been color inversion in pre-processing step. Optimizer column expresses the network optimizer.

Filter	Augm	Inverse	CNN Architecture / Dense Architecture	Optimizer	Accur.
None	False	False	$conv(30,3,1)+pooling(2,2)+conv(60,2,1)+pooling(2,2)$ / $1000\vert 2$	SGD	84.5%
Median	True	False	$conv(30,3,1)+pooling(2,2)+conv(60,2,1)+pooling(2,2)$ / $1000\vert 2$	SGD	91.47%
Bilateral	True	False	$conv(30,3,1)+pooling(2,2)+conv(60,2,1)+pooling(2,2)$ / $1000\vert 2$	SGD	95.32%
Bilateral	True	False	$conv(32,5,1)+pooling(2,2)+conv(64,5,1)+pooling(2,2)$ / $3136\vert 1000\vert 2$	Adamax	96.26%
Bilateral	True	True	$conv(32,5,1)+pooling(2,2)+conv(64,5,1)+pooling(2,2)$ / $3136\vert 1000\vert 2$	SGD	96.22%
Bilateral	True	False	$conv(32,5,1)+pooling(2,2)+conv(63,5,1)+pooling(2,2)$ / $3136\vert 1000\vert 2$	SGD	96.11%
Bilateral	True	True	$conv(30,5,1)+pooling(2,2)+conv(60,3,1)$ $+pooling(2,2)+conv(90,2,1)+pooling(2,2)$ / $1000\vert 2$	SGD	95.67%
Bilateral	True	False	$conv(30,5,1)+pooling(2,2)+conv(60,3,1)$ $+pooling(2,2)+conv(90,2,1)+pooling(2,2)$ / $1000\vert 2$	SGD	95.79%
Bilateral	False	False	$conv(30,5,1)+pooling(2,2)+conv(60,3,1)$ $+pooling(2,2)+conv(90,2,1)+pooling(2,2)$ / $1000\vert 2$	SGD	94,24%
Bilateral	False	False	$conv(25,10,1)+conv(50,5,1)+conv(100,3,1)$ $+pooling(2,2)+conv(150,2,1)+pooling(2,2)+conv(200,2,1)$ / $1000\vert 2$	SGD	93.62%
Bilateral	True	False	$conv(25,10,1)+conv(50,5,1)+conv(100,3,1)$ $+pooling(2,2)+conv(150,2,1)+pooling(2,2)+conv(200,2,1)$ / $1000\vert 2$	SGD	95.02%
Bilateral	True	True	$conv(25,10,1)+conv(50,5,1)+conv(100,3,1)$ $+pooling(2,2)+conv(150,2,1)+pooling(2,2)+conv(200,2,1)$ / $1000\vert 2$	SGD	94.85%
Bilateral	False	True	$conv(25,10,1)+conv(50,5,1)+conv(100,3,1)$ $+pooling(2,2)+conv(150,2,1)+pooling(2,2)+conv(200,2,1)$ / $1000\vert 2$	SGD	93.22%
Bilateral	True	True	$conv(25,10,1)+pooling(2,2)+conv(50,5,1)$ $+conv(100,3,1)+pooling(2,2)+conv(150,2,1)+conv(200,2,1)$ / $1000\vert 2$	SGD	94.33%
Bilateral	True	False	$conv(25,10,1)+conv(50,5,1)+conv(100,3,1)$ $+pooling+conv(150,2,1)+pooling(2,2)+conv(200,2,1)$ / $3125\vert 1000\vert 2$	SGD	95.83%

CURRICULUM VITAE

Name Surname :Mahtab Mohtasham Khani

Place and Date of Birth :Iran 1991/07/31

E-Mail :khani17@itu.edu.tr

EDUCATION:

- **B.Sc.:** 2015, Shahid Madani University of Azarbaijan, Computer Science, Information Technology

PROFESSIONAL EXPERIENCE AND REWARDS:

- Working as researcher at ITU ARC Laboratory , Supervisor: A.Prof. N. Kemal Ure , 2017 - present.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Mahtab Mohtashamkhani**, Sahand Vahidnia, leila Ghasemzadeh, Deep-Learning-Based Crack Detection with Applications for the Structural Health Monitoring of Gas Turbines,Structural Health Monitoring Journal of SAGE Journals (Submıtted).