POTHOLE DETECTION IN ASPHALT IMAGES USING CONVOLUTIONAL NEURAL NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HIMMET ATEŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONIC ENGINEERING

MARCH 2019

Approval of the thesis:

POTHOLE DETECTION IN ASPHALT IMAGES USING CONVOLUTIONAL NEURAL NETWORKS

submitted by **HIMMET ATEŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronic Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Tolga Çiloğlu
Head of Department, **Electrical and Electronic Eng.** _____

Prof. Dr. İlkay Ulusoy
Supervisor, **Electrical and Electronic Eng., METU** _____

Examining Committee Members:

Prof. Dr. Uğur Halıcı
Electrical and Electronics Eng.,METU _____

Prof. Dr. İlkay Ulusoy
Electrical and Electronic Eng., METU _____

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Eng.,METU _____

Assist. Prof. Dr. Elif Vural
Electrical and Electronics Eng.,METU _____

Assist. Prof. Dr. Erdem Akagündüz
Electrical and Electronics Eng.,Çankaya University _____

Date: 20.03.2019

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Himmet Ateş

Signature:

ABSTRACT

POTHOLE DETECTION IN ASPHALT IMAGES USING CONVOLUTIONAL NEURAL NETWORKS

Ateş, Himmet
Master of Science, Electrical and Electronic Engineering
Supervisor: Prof. Dr. İlkay Ulusoy

March 2019, 89 pages

When asphalt defects are detected and not corrected, they can cause accidents and loss of property and lives. Potholes formed in such asphalt surfaces are one of the biggest causes of accidents. In order to minimize the loss of life and property, the potholes formed on the asphalt should be detected and corrected by the authorities as early as possible. The potholes formed on asphalt surfaces can be detected either manually or automatically. Automated methods can be more time and cost effective.

Vibration-based data processing, 3D reconstruction and processing, and image processing in 2D images are the basic methods used in automatic detection systems.

In this thesis, the aim is to develop a system which is easy to apply and has low error rate by using ”Convolutional Neural Networks" methods that will be applied on 2D images. In classical machine learning methods, fixed (unchanging) features are extracted and classification methods (either static or dynamic) are applied through these features. The success of these methods depends on the accuracy, structure and quality of the extracted features as well as the applied algorithms

A Convolutional Neural Network is constructed and compared with classical machine learning methods, which are already applied for pothole detection problem

in the literature, in terms of success rate and failure rate using the asphalt image sets. The different parameters of the convolutional neural network method are tested on the existing image sets and the effect of these parameters is also analyzed

Keywords: Convolutional Neural Networks, Classification, Image Processing, Pothole, Anomaly

ÖZ

ANOMALİ İÇEREN ASFALT RESİMLERİNDE DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK ÇUKUR TESPİT ETME

Ateş, Himmet
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği
Tez Danışmanı: Prof. Dr. İlkay Ulusoy

Mart 2019, 89 sayfa

Asfalt bozuklukları tespit edilip düzeltilmediği durumlarda kazalara, dolayısıyla mal ve can kayıplarına sebep olabilmektedir. Bu tür bozuklukların başında gelen asfalt çukuru, kazaların oluşmasındaki en büyük nedenlerden biridir. Can ve mal kayıplarının en aza indirilmesi için asfaltlarda oluşan çukurların otoriteler tarafından erken tespit edilip düzeltilmesi gerekmektedir. Asfaltlarda oluşan çukurlar göz ile veya otomatik olarak tespit edilebilmektedir. Göz ile tespit yöntemleri oldukça zaman alıcı ve maliyetli yöntemlerdir.

Otomatik tespit sistemlerinde, titreşim bazlı veri işleme, üç boyutlu geri çatım ve işleme ile iki boyutlu resimlerde görüntü işleme yöntemleri olarak temelde üç yaklaşım kullanılmaktadır. Titreşim bazlı metotlar uygulaması kolay ancak hata oranı yüksek sistemlerdir. Üç boyutlu yöntemler ise, başarı oranı yüksek ancak uygulaması zor sistemlerdir. İki boyutlu görüntü işleme yöntemleri bu ikisi arasında yer almaktadır.

Bu tezde amaç, iki boyutlu resimler üzerinde “Derin Öğrenme” yöntemleri kullanılarak uygulaması kolay ve hata oranı düşük bir çukur bulma yöntemi geliştirmektir. Klasik makina öğrenme yöntemlerinde sabit (değişmeyen) öznitelikler çıkartılarak bunlar üzerinden statik veya dinamik yapıda sınıflandırma uygulanmaktadır. Bu yöntemlerin başarısı algoritmalara da bağlı olmakla birlikte çıkartılan özniteliklerin doğruluğuna, yapısına ve niteliğine dayanmaktadır. Derin öğrenme yöntemlerinde, gerçekte dinamik olarak değişen öznitelikler arasından mevcut problem için en doğru öznitelik seti öğrenme yoluyla elde edilmektedir.

Derin öğrenme yöntemi ve mevcut alanyazında bulunan klasik yöntemler anomali içeren asfalt görüntü seti üzerinde denenmiş ve başarı oranı, hatalı tespit oranı gibi başarı kriterleri yönünden kıyaslanmıştır. Araştırmanın sonucunda derin öğrenmenin diğer yöntemlere göre daha başarılı olduğu gözlenmiştir.

Anahtar Kelimeler: Derin Öğrenme, Sınıflandırma, Görüntü İşleme, Çukur, Anomali

To my lovely wife Gülbin Ayşı and beloved daughter Beliz Lena

ACKNOWLEDGMENTS

I would like to express my deepest respect and thanks to my thesis supervisor Prof.Dr. İlkay Ulusoy. I would not complete this thesis works without her invaluable support, encouragement, supervision and helpful advice

I am deeply indebted to my wife for her endless patience and encouragement, support and love to me.

I would like to thank my friends whom always supported me to finish this study and motivated me when I felt unwilling sometimes.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1. PRESENTATION

Road bumps, potholes and cracks are some of the major important structural defects which occur on the asphalt roads due to various causes such as heavy trucks passing on them, overcast and construction errors. These defects, especially potholes, unless taken care of on time may cause serious damages some of which are flat tire and wheel damage, damage on the lower part of a vehicle, sudden brake and slipping, collisions, accidents and unfortunately death.

In 2016 [26] according to Turkish Statistical Institute (henceforth TUIK), the number of traffic accidents reached up to 1,182,491. 997, 363 of these accidents caused physical damage; 303.812 people got injured and 7300 people were killed in these accidents [26]. As reported by TUIK, nearly 1 % of the accidents occurred due to distorted road conditions and around 90% took place as a result of driver faults. However, it should also be noted that some of the driver faults may also result from poor road conditions which haven't been diagnosed.

As can be inferred from the statistics given above, detection and maintenance of the potholes on time is very crucial to prevent damages before they occur. These defects can be detected automatically and manually by Local Governments and General Directorate of Highways. Currently almost all detection processes are done manually which is time consuming and very expensive. Hence, developing automatic pothole

detection systems helps authorities to fix the defects on time, save people's lives and contribute to national economy by preventing the accidents.

A system that quickly detects and repairs potholes on the pathways will consist of a few sub-systems. These sub-systems are data collection system, data analysis and decision system, and maintenance planning system. Data collection system is the sensor system e.g. camera, ultrasonic sensor, laser scanner, GPS etc. which is mounted on vehicles to provide necessary information for data analysis and decision system. Data analysis and decision system is responsible for processing the sensor data and decide whether there is a defect on the road or not, and if any defect exists, the system stores it with the location information obtained from the GPS data. Final part is responsible for making a maintenance program using the stored data. There are plenty of sensors and GPS devices manufactured and used in the market. Also, lots of business planning software and algorithms are in use. Hence, the system that needs to be developed to work on this issue is *the data analysis and decision system* responsible for pothole detection.

There are many studies conducted across the world to develop systems that automatically detects road defects especially potholes. These studies can be categorized mainly into three parts: vibration based, 3D reconstruction based, and vision-based. None of the existing methods have yet turned into products. Engineers and scientists are still working on these methods to have satisfactory results that will turn into working products.

1.2. OBJECTIVE AND SCOPE

Detection of potholes from the sensor data is the main problem for automatic detection and maintenance system. As stated above there are three main approaches to this

problem, methods that use vibration sensor data like 3D accelerometer, methods that use 2D vision sensor data like camera and methods that use 3D sensor data like laser scanner or built 3D data from 2D sensor like stereo vision.

The sensors may vary in terms of price and usability i.e. some sensors are cheap whereas some are very expensive and some sensors are easy to mount and obtain data while some need robust mounting and calibration all the time. As well as the sensors some methods are very hard to implement in real life conditions and some methods are error prone like producing lots of false alarms. In contrast, some provide very accurate results.

In this study the main aim is to develop and propose a method that is applicable in real world, which needs data from a sensor that is not too expensive and is easy to use and finally which needs reasonable computing power which provides satisfying accuracy. In general, vibration-based methods are cheap and easy to apply, but they are error prone. 3D based methods provide good accuracy, but they require high computing power and expensive sensors. Also, they are hard to implement in real life. 2D vision-based systems are not so expensive e.g. there are many cheap cameras in the market which are easy to apply. They need reasonable computing power comparing to 3D methods however the accuracy of them should be improved for more practical use.

Anomaly detection on the road surface is a relatively solved problem. Lokeshower et.al [35] separates video frames into two categories namely frames with or without distress with more than 96% accuracy. The main object of the proposed method in this thesis study is to detect potholes among distress frames truly. While studies up to now have used basic features obtained from the frame to detect potholes, the method in this study obtains and uses high levels of complex features to classify the frames that

contains potholes or not. The complex features and the neural network that is used for classification improves the accuracy among other methods.

This study consists of 5 chapters. First chapter sets the definition of problem, the scope and objectives of the study. Second chapter summarizes the literature review and studies up to now. Performance and detailed explanation of present 2D methods are depicted in 3rd chapter. 4th chapter introduces the convolutional neural networks (CNN) explaining its application to pothole detection problem and comparing its performance against other 2D methods. Last chapter summarizes and discusses what can be done to improve this problem.

CHAPTER 2

LITERATURE SURVEY

2.1. VIBRATION BASED METHODS

Vibration based methods, which are used to detect potholes, generally use the output of 3 axis accelerometers to detect potholes and use GPS to locate the pothole. As wheels of a car pass over a pothole, the output of the accelerometer gives a different output from normal operation (a fluctuation) and the existence of a pothole is detected. However flat tires, sudden brakes, rail road crossings etc. also produce different outputs from the regular road conditions. These sensors are quite cheap and algorithms require comparatively low calculation costs however as potholes and other factors do not have characteristic properties the methods are prone to errors especially they produce false alarms. Moreover, drivers escape from passing over potholes and sensors are not able to produce the necessary outputs.

Based on three axis accelerometer sensor output, Eriksson et.al [13] uses cluster-based filters. Data collected from smooth road, crosswalks, railroad crossing, potholes, manholes, hard stop. The detector is trained according to these data set and 5 filtering stages are conducted on the collected data: Speed, High Pass, Z-peak, xz-Ratio, Speed vs. Z ratio. The PSD (power spectral density) of the road is first calculated in [8]. After calculating the PSD using Fourier transform, International Roughness Index (IRI) and Riding Quality Index (RQI) are calculated. Based on velocity and RQI, the road is classified as good quality and bad quality. Jang et.al [23] first corrects mean-shifting, caused by normal vehicle movements, such as making turns, changing lanes, and going up or down hills by taking out the exponential moving averages from the

accelerometer signals. Then, the root mean square (RMS) values of the accelerometer signals are calculated in a fixed time window length, which is equivalent to 0.8 seconds. After collecting the data, supervised machine learning technique is used to classify the collected data fragments into three different categories: back-end server impulse, rough, and smooth classes, respectively. Phone accelerometers are used as sensors in [38]. The proposed study investigates Z axis mainly in such a way that values on the z axis that are greater than a certain threshold indicates the existing of a pothole. Next, fast changes in vertical acceleration data are explored and this way, potholes are detected. Sound-level meters (phonometer of smart phones) are used in [2] with a 3-axis accelerometer, to evaluate the differences in sound level (dB) over time (s). The ultrasonic sensor *HC-SR04* mounted on a vehicle is used in [36] to detect potholes. It measures the threshold distance between smooth ground surface and the car and if new measurement is higher than this threshold, it is considered as a pothole. The ultrasonic sensors are not robust while moving it does not give coherent outputs. Moreover, they produce irrelevant outputs due to current, voltage changes and vibrations due to vehicle movements.

2.2. 3D LASER AND RECONSTRUCTION METHODS

3D laser and reconstruction methods usually use either a 3D laser or stereo cameras to obtain 3D surface of the region of interest. After using depth and other features, they try to detect whether or not a pothole exists. 3D laser gives 2D points as output and the elevation of these points are used for pothole detection. The 3D laser sensors are quite expensive, and the calibration of the sensor and adjustment of the vehicle speed is vital. By using the elevation of the point Chang et.al [7] segments laser points according to randomly chosen N points near high frequency changes. Then, other points are clustered according to nearest N cluster points. After all, pixels have been

assigned; a new mean location is computed, and the steps are repeated until no significant change occurs.

3D reconstruction methods either use stereo vision cameras [19] [51] [58] or successive video frames [25] to obtain the 3D surface map of the road. Stereo vision methods generally use optical centers of the cameras (C1, C2), 3D point P, and the image points (P1, P2) of P on both camera images. If P1 and P2 are known, the 3D coordinates of P can be calculated from the geometrical relation. Giving a point in one image and finding the corresponding point is called *correspondence or matching problem*. Computing the 3-D coordinates of the corresponding point in space with a given correspondence is called *3-D reconstruction*. The alignment and the calibration of the cameras are so important that more than a half pixel misalignment between the two stereo imagers would start to degrade the system performance. Jog et.al [25] detects pothole candidates using 2D methods. When a frame is accounted as a pothole, the system starts to obtain 3D map of the road using successive frames. The main problem of this method is the speed of the car. To obtain accurate 3D map, the velocity of the car should not change over time. Besides 3D laser and stereo cameras, Microsoft Kinect sensor is used in [39]. The kinect sensor has two cameras on it. The IR sensor helps understand the depth of the image. The local minimum of each column in the image is evaluated and subtracted from the column itself and eventually, pothole is obtained from the rest of the data.

3D reconstruction methods are not applied on moving vehicles; rather they are used on stable vehicles. Calibration and alignment of the cameras are quite important. 3D methods give robust and accurate results however they require either high cost sensors or high operation costs (calibration, alignment etc.).

2.3. VISION BASED METHODS

A camera is used as a sensor in vision-based methods. Images obtained from the camera are examined and features are obtained using image processing techniques. After features are obtained, classification methods are applied on these features to decide whether a pothole exists or not. The features that are classified can be surface related (like texture, standard deviation of pixels etc.) or edge and orientation based (e.g. Hog, Sift, Haar etc.) Morphologic operations, Bayesian Classifiers, Support Vector Machines (SVM) are examples of classification methods.

Images are first converted to grayscale. Either edge detector algorithms (canny etc.) or threshold algorithms (adaptive, histogram-based Otsu etc.) are applied to obtain binary images in [46] [27] [34] [49] and [6]. After binary images are gathered, contours are obtained using connected components. The features of the contours like texture, standard deviation, area and intensity are examined according to the rest of the image and potholes are detected according to some rules. Defects are segmented using Partially Differential Equations (PDE) in [32] and nonlinear SVM is used for classification. Hog features and Bayesian Classifiers are used in [4] to detect potholes. Assuming high frequency occurs at big changes on road surface, Jang et.al [24] uses a spatio temporal saliency method to detect potholes. Details and performance of the main vision-based methods are explained in chapter 3.

Cameras are slightly more expensive than accelerometers and they need neither calibration nor adjustment. Vision based methods are good at detecting saliencies on the road surface; however, as potholes do not have distinctive characteristics when compared to most of manholes, dirt and patching, they are not good at classifying potholes among latter ones. Lacking the depth information in 2D frames, performance of the vision-based methods needs to be improved.

The major differences among main methods are summarized in Table 2.1. The major features (Accuracy, computational cost etc.) of these methods separated into as low, medium and high based on the information provided by the given references. As mentioned in the first chapter, the main goal of this study is to propose a low-cost sensor and low-cost operation system so that data can be collected by many vehicles and obtained data (i.e. in a center) is classified at a satisfying accuracy. As can be seen from the table that, 2D vision-based systems are at an affordable cost (both computational and operational), but the accuracy should be improved to be used by authorities conveniently. Considering this information, a new method is proposed to improve the performance of the existing vision based methods.

Table 2-1. *Comparison of Pothole Detection Methods*

Method	**Sensor Cost**	**Computational Cost**	**Operational Cost (calibration, adjustment etc.)**	**Accuracy**
Vibration Based	low	low	low	low
3D Based	high	high	high	high
2D Based	medium	medium	low	medium

CHAPTER 3

VISION BASED METHODS

3.1. INTRODUCTION

In this chapter the image-based pothole finding methods mentioned in chapter 2 will be elaborated in more detail. As mentioned above, determining whether there is an abnormality on the road surface is a partially solved problem; therefore, the main goal is to detect accurately if there is a pothole in the frames that include an anomaly. The vision based methods can basically be grouped into 2 categories. The first group locates the anomaly in the image and later on, analyzes some properties of that region and decide on it according to the rules developed by the authors of the methods. This group will be called *morphological examination methods*. The second group obtains a feature vector from the image such as histogram of gradients (HOG) and uses a proper classifier to detect whether that image contains a pothole or not. The second group will be called *feature-based classification methods*. The performances of the proposed algorithms are analyzed under chapter 5.

3.2. MORPHOLOGICAL EXAMINATION METHODS

3.2.1. A Method for Automated Assessment of Potholes

In [34] an automated method is developed to detect potholes, cracks and patches using three sets of visual properties of those defects. These visual properties are:

Image texture: The image texture inside a pothole, crack and patch is more contrast and varied than the distress free road surface area. However, contrast variation inside a pothole region is much more than a patch region in the same image.
Shape factor: The potholes and patches have shapes more likely to be circular than cracks. The cracks have more elongated shapes. Circularity is measured in terms of area and perimeter of the shapes
Dimension: Potholes and patches have bigger dimension (width) when compared to cracks.

The algorithm first enhances the image, then the image is segmented according to anomaly regions. After that, the three sets of visual properties of segmented regions are extracted and finally a decision algorithm is applied for classification. The algorithm is used to classify potholes, patches and cracks; however, our data set consists of potholes, manholes, dirt and patches. Therefore, the decision part is changed so that it can help figure out whether the region of interest is a pothole or not. The algorithm steps and how they are implemented is listed below:

- Frame is inputted;
- Its blue channel is selected and converted into 8-bit depth format;
- To remove outside noise, median filtering is applied. For this, 1024*768 frames filter of size 45 is applied;
- A weighted mean based on adaptive thresholding is applied to convert the enhanced image into binary image with black pixels representing objects of interest;
- To fill the gaps in the binary image, the frame is eroded with size 20 twice to add black pixels;
- To remove isolated black pixels or their small cluster, the morphologic dilation of size 4 is applied 5-6 times
- Morphological erosion of size 20 is applied twice again to add black pixels to the binary image;

- Connected component labeling and chain coding techniques is applied to count the number of objects or regions of interest and estimate the area (A) and perimeter (P)of each of the object;
- STD (standard deviation), CIRC (circularity) and W (average width) of each of the remaining objects are determined. CIRC and W are calculated according to (3.1) and (3.2) respectively
- Each region of interest is classified as pothole or not according to the decision rationale/criteria

  Type (object) =

  (a) Potholes, if A>177 & STD >= 10 & CIRC >= 0.10 & W >= 60mm;

  (d) Non-pothole, if otherwise;
- Store the result into a file
- Repeat the steps for all other frames.[34]

$$CIRC = 4\pi \times \frac{Area}{Perimeter2} \tag{3.1}$$

$$W = 4 \times \frac{Area}{Perimeter} \tag{3.2}$$

Dirt, manhole and patching all may have shapes similar to potholes in terms of CIRC and W. On the other hand, STD parameter is able to differentiate potholes from other types of anomalies. However, there are potholes whose STD is small (<10) like in Figure 3.1 and while there are dirt and manholes whose STD is bigger than 10 shown in Figure 3.2.

Figure 3.1. A pothole with STD < 10

Figure 3.2. A dirt with STD > 10

3.2.2. Potholes Detection in Asphalt Pavements

Another morphological examination method is presented in [27]. Like the method above, it uses visual properties of pothole areas to detect pothole within a frame. Three assumptions are used to detect potholes;

(a) A pothole area is darker than the surrounding road surface

(b) The shape of a pothole is approximately elliptical, due to a perspective view.

(c) The surface texture in a pothole is much rougher and grainy than the surrounding surface texture [27].

Based on these assumptions, the algorithm has three components namely image segmentation, shape extraction and texture extraction and comparison. In the image segmentation part, first the image is converted into a grayscale image. After that, a 5*5 median filter is applied to remove noise from the image. Histogram based thresholding algorithm is used to segment the image as defect and non-defect regions. To determine the threshold value T, a line is drawn from the maximum intensity value (Pmax) to the origin (P0). The point which has the maximum distance to this line is the threshold value (Figure 3.3) [27]. The points below the threshold value are the defects region, the rest is the surface region.

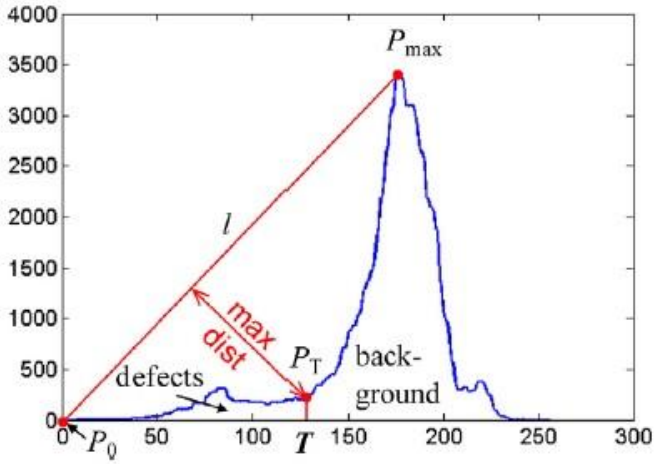


Figure 3.3. Histogram based thresholding [27]

After the image is segmented, black regions (R) are examined whether they are pothole shades or not. If eccentricity of the shape is bigger than 0.99 and the position of the centroid is inside the region of interest R, and the major axis over the size of R is bigger than a threshold. The region is assumed to be a pothole candidate area. After detecting the pothole candidates, three Leung-Malik (LM) filters and 1 Schmid (S)

filter is applied both to inside and outside of the pothole region. Two-feature vectors containing standard deviations are obtained calculated using these filters. F_o and F_i are the feature vectors obtained for outside and inside of the region of interest respectively. If $F_i > F_o$,the region is assumed to be pothole.

The proposed algorithm explained above is not implemented and tested because manholes, dirts and patches all may have similar potholes. Thus, the algorithm does not differentiate among these anomalies. STD is the only device to use to distinguish among them. This is already is implemented and tested in the above method.

3.3. FEATURE BASED CLASSIFICATION METHODS

3.3.1. HOG Features Descriptors and Bayesian Classifiers

Pothole detection in the asphalt roads is a kind of object detection technique. The main idea behind object detection is to find such a feature set that the object can be differentiated from the background image and/or other objects using these feature set. Potholes usually have elliptical shapes and contain coarser texture comparing to asphalt surface. Considering these observations [4] using HOG features are thought to be a good method to distinguish potholes from its environment. HOG feature descriptors [10] are used in many fields in computer vision to detect and recognize objects.

A feature descriptor [18] represents an image patch that simplifies the image by removing useful information and releasing redundant information. In the HOG feature descriptor, the distribution properties of the gradients directions are used. Image gradients (x and y variants) are useful in sense that the magnitudes of the gradients are quite high at the corners and edges. Corners and edges usually contain the most

knowledge about the shape of the object. In obtaining HOG feature descriptors first horizontal and vertical gradients are calculated using the masks in (3.3).

$$F_x = [-1 \quad 0 \quad 1] \qquad \text{and} \qquad F_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \tag{3.3}$$

Obtaining the vertical and horizontal gradients magnitude and phase (orientation) of gradients are calculated using equations (3.4) and (3.5)

$$|G|_F = \sqrt{F_x^2} + F_y^2 \tag{3.4}$$

$$\emptyset = \tan^{-1}(F_y / F_x) \tag{3.5}$$

First, images are resized to 200x200 pixels (if their size is different) afterwards the image is segmented to 8x8 cells which are non-overlapping into each other. Each cell is divided again into 4x4 pixel blocks and orientation of each block is calculated quantizing into 8 bins and finally a HOG feature descriptor vector is obtained. Figure 3.4 shows a pothole image and its HOG representation.

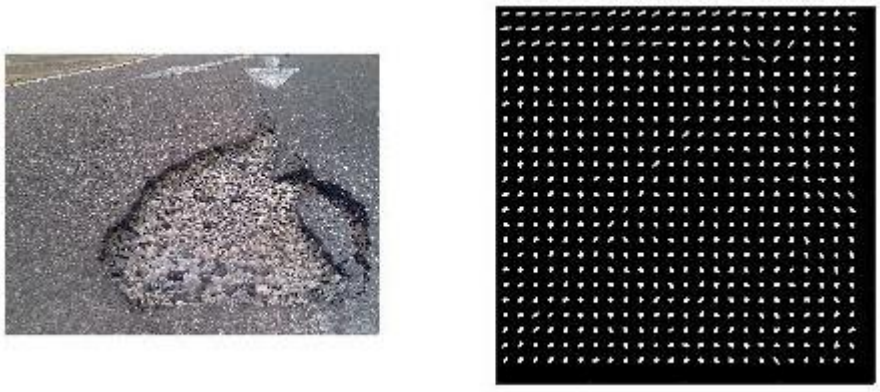


Figure 3.4. HOG feature representation of a pothole image

The next step after obtaining the feature vector is to feed this vector into a proper classifier. In [4] a Naïve Bayes Classifier is used for discriminating pothole images from non-pothole containing images. The Naive Bayesian Classifier is based on Bayes' theory with assumptions of independent variables between predictors. The Naive Bayesian model can be easily generated without complex parameter estimates, especially useful for very large data sets. The feature vector obtained from the images has a very large scale and assuming the independence of these features Naïve Bayes Classifier is considered to be a suitable classifier. Bayes' theorem is a way to calculate the posterior probability which is shown in equation (3.6) where C_i is the class label (1,2 …n) which is pothole and non pothole for this case , V_f is the feature vector, P(C_i) is prior probability of class and P(V_f) is prior probability of predictor. V_f is assigned into a class C_i if the posterior probability $P\left(C_i/V_f\right)$ takes the highest value for that class among other classes.

$$P\left(C_i/V_f\right) = \frac{P\left(V_f/C_i\right)P(C_i)}{P(V_f)} \tag{3.6}$$

This method is tested using Matlab Image Processing Tools (IPT) which has already had a built in classifier for Naive Bayes Classifier. 60 images chosen randomly among 1000 images (24 pothole, 36 non pothole) and the rest 940 is used for training. The detailed results will be discussed in chapter 5.

3.3.2. Potholes Detection in Asphalt Pavements

As the pothole detection problem is clearly a two class problem instead of Bayes Classifier, a Support Vector Machine (SVM) classifier is also tested using Matlab built in IPT SVM classifier using the same HOG feature descriptive vector. “Support Vector Machine” (SVM) is a supervised machine learning algorithm which is generally used for two class separation problems which are not linearly differentiable. An SVM classifies data by finding the best hyper plane that separates all data points of one class from those of the other class [54]. Vi is the training vector and Cj is their categories. For some dimension d, the Vi ∈ Rd, and the Cj= ±1. The equation of a hyper plane is expressed in (3.7) where β ∈ Rd and b is a real number.

"f(v)=v'β+b =0" (3.7)

The decision boundary is computed by finding β and b that will minimize || β || such that all data points satisfy the condition expressed in (3.8)

C_j f(V_j)≥1 (3.8)

Vj that satisfies Cjf(Vj) =1 conditions are the support vectors which are on the boundary. SVM has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Figure 3.5 illustrates these definitions and SVM representation.

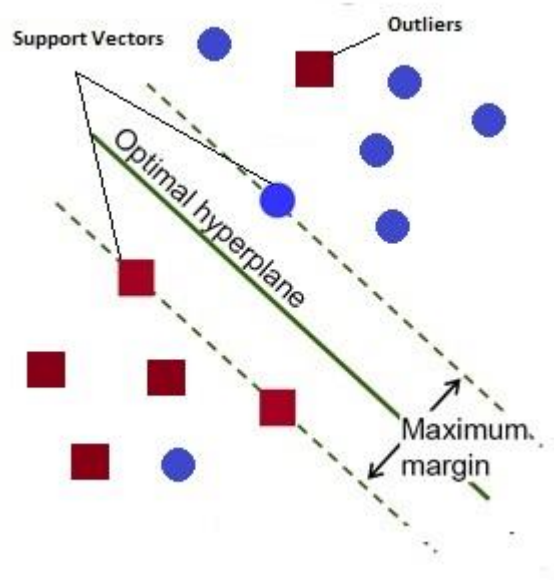


Figure 3.5. SVM definitions and representation

The same test and train image sets used in Bayesian Classifier are used and results are compared in chapter 5.

3.3.3. Haar-like Features and Linear Cascade Classifiers

Haar-like features like HOG features are another feature set used for object detection and classification. They are introduced in [55] for rapid object detection and a developed version is used in [56] for face detection. Haar-like features which are like convolutional kernels are applied to every 24x24 segment of an image in various size

and at each step pixels under the black regions are summed and subtracted from the sum of the pixels under the white regions. Hence at every step one feature is obtained resulting near 160.000 features for a 24x24 window. Haar-like features are shown in Figure 3.6 and their rotated versions are also used in obtaining the feature vector.

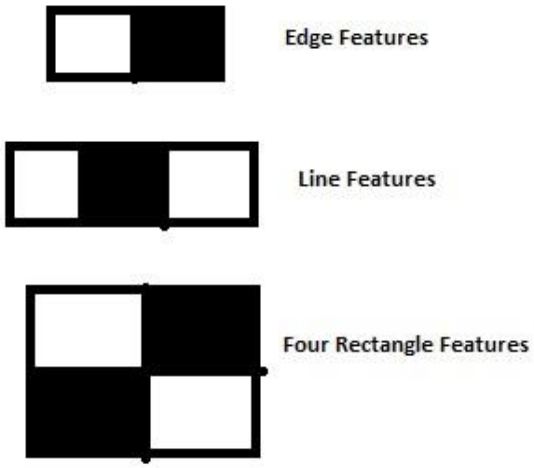


Figure 3.6. Haar-like features

Even for a 24 by 24 segment too many features are obtained, and it costs a lot of computation. Most of the features obtained during this process is unnecessary. Considering Figure 3.7[56] it is seen that most significant features are obtained in eye-bridge of nose-eye area and eyes-nose and cheeks area. Eye area is darker than nose-cheek area and eyes are darker than the bridge of the nose.

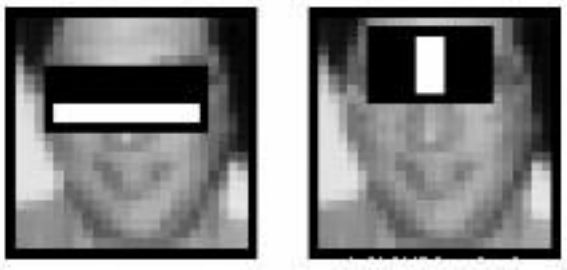


Figure 3.7. Haar-like features and a face [56]

To get rid of the irrelevant features and choose the best ones Adaboost is used. Adaboost [14] is a super-efficient feature selector using weak classifiers. Each feature on the training images are applied and for each feature best threshold is found which classifies the faces to positive and negative. Features with minimum error rates are chosen and the final classifier is obtained as a weighted sum of these weak classifiers. This process reduces 160.000 features to nearly 6.000 features. Again, for all 24x24 region of an image it needs too much computation because most regions are not face regions and thus are not candidates. To solve this problem, linear cascade classifiers are used. A small set of features in an image's sub window is applied to a linear classifier. If it fails, it is a non-face region. If it passes, a bigger feature set is applied to a second linear classifier. If it fails, it is a non-face region. If it passes a bigger feature set is selected and applied to a third classifier. This goes on until the chosen number of classifiers is reached. If the last feature set passes the last classifier, it is labeled as face region. The representation of cascade classifiers is shown in figure 3.8

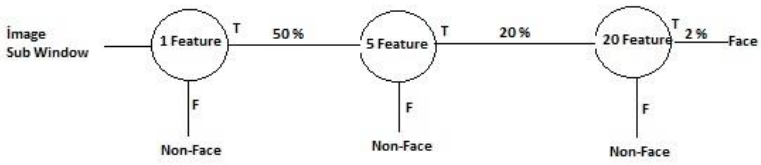


Figure 3.8. Representation of cascade classifiers

The method is developed basically for face detection but can be applied to any object for detection. Matlab has a built-in method for this purpose in its IPT. First, Matlab's Training Image Labeler is used to label pothole areas. Approximately, 256 pothole areas are selcted and labeled as potholes. As the second step, negative images (non pothole images) are introduced. Later the number of linear cascade classifiers and their false alarm rate are defined. A detector (feature selector) is trained acording to the training set and when a new image is introduced its feature vector is obtained using this detector and it is fed to the cascade classifiers. Haar-like features are very good at detecting objects that have well defined patterns such as eye-nose-eye but produces lots of false alarms if the object doesn't have such a pattern. The detailed results will be discussed in chapter 5.

3.3.4. Image Classification with Artificial Neural Networks

In the above feature-based methods, it is necessary to obtain a set of feature vector before training the system. However, in this method, the image itself is given to an artificial neural network (ANN) as a feature vector. Assuming that the anomaly detection is a solved problem, anomaly region is first found in the image and then extracted from the image and this anomaly region image is given as input to the system. Figure 3.9 and 3.10 shows the original and extracted images.

Figure 3.9 Original image

Figure 3.10 Cropped image

Figure 3.11 shows the constructed artificial neural network structure.

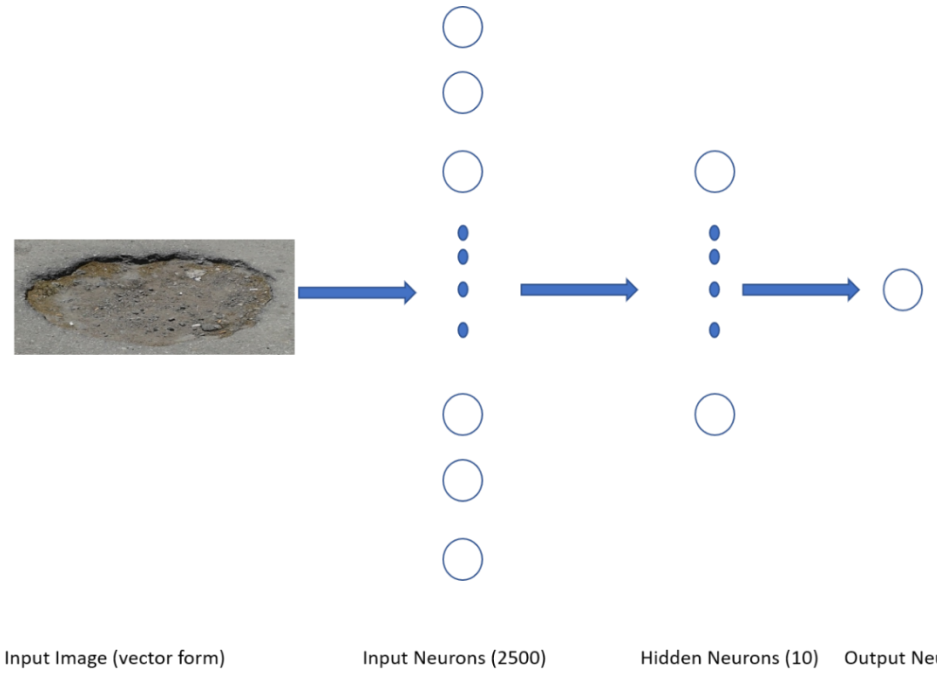


Figure 3.11 ANN structure

The cropped image is resized to 50 pixels by 50 pixels and then turned into a 2500x1 vector form and fed to the network. Detailed information about the neural networks is explained in the next chapter. Success result of this method will be discussed in chapter 5

CHAPTER 4

POTHOLE DETECTION USING CONVOLUTIONAL NEURAL NETWORKS

4.1. INTRODUCTION

The traditional pattern of recognition (or classification) methods is to first obtain a set of simple and fixed (not trainable and do not change over time) feature set and use trainable classifier to achieve the desired classification or clustering over those features. As stated earlier the distress asphalt surface is smooth and less varied when compared to the anomalies existed in the asphalt. However, anomalies on the road surface do not have a fixed pattern and a fixed decomposer feature set. Dirt, patching and a pothole may all have similar variance (in intensity), shape and gradients and so does the feature set.

To overcome this problem either we should obtain and use the depth information of the anomaly space whose pros and cons are mentioned in chapter 2 or we should develop a method to obtain a more sophisticated and adaptive (trainable) feature set to classify these anomalies. This chapter will explain the proposed method to solve pothole detection problem which is The Convolutional Neural Networks (CNN) or namely deep learning. In this chapter, first the perceptron after the neural networks will be introduced. After these concepts are introduced convolutional neural networks and the proposed version of CNN will be explained in detail.

4.2. NEURAL NETWORKS

4.2.1. The Perceptron

Human brain consists of billions of neurons. They are electrically excitable cells that receive, process, and transmit information through electrical and chemical signals [44]. These signals between neurons occur via specialized connections called synapses. Neurons has three main parts: body, dendrites and axon. Neurons are interconnected to each other through these synapses which are the junction of axon and dendrites. A typical neuron is shown in figure 4.1[45]

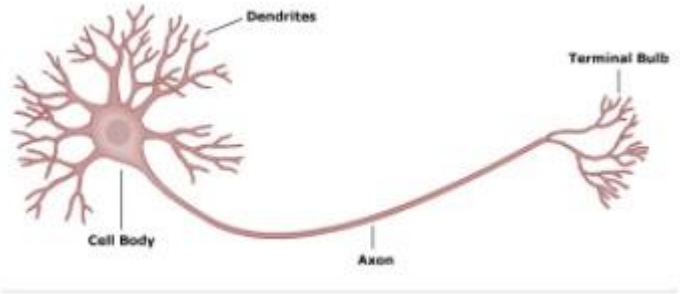


Figure 4.1. A neuron [45]

Dendrites are the input parts of the neuron. If the electrical signals reach a certain threshold level, the neuron fires provide a chained input and the other neurons are connected to this neuron through the synapses.

The perceptron shown in figure 4.2 is a mathematically modeled (computational model) neuron to simulate the learning mechanism of the human brain.

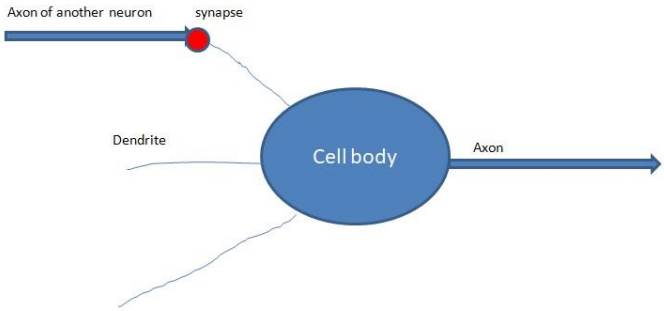


Figure 4.2. The perceptron

In the computational model of a neuron, the signals that travel along the axons (e.g. x_0) interact multiplicatively (e.g. w_0x_0) with the dendrites of the other neuron based on the synaptic strength at that synapse (e.g. w_0). The idea is that the synaptic strengths (the weights w) are learnable and they control the strength of influence or inhibitory (negative weight) of one neuron on another [40].

Firing of a biological neuron is modeled through activation function of the perceptron where impulses from other axons summed (4.1) and the sum is evaluated according to the chosen activation function f which is shown in figure 4.3

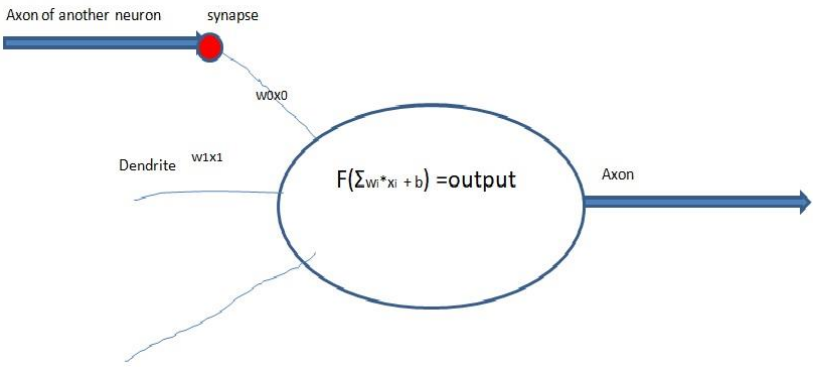


Figure 4.3. The perceptron and activation function

$$output = f(\Sigma wi * xi + b) \tag{4.1}$$

The activation function can be sigmoid, tanh, ReLU etc. In its basic form the output can be determined by (4.2)

$$output = \begin{cases} 0 \; if \; \Sigma wi * xi + b < 0 \\ 1 \; if \; \Sigma wi * xi + b > 0 \end{cases} \tag{4.2}$$

Sigmoid, tanh, ReLU and Leaky RLU is shown through (4.3) to (4.6) respectively

$$Sigmoid(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \; where \; x = \Sigma wi * xi + b \tag{4.3}$$

$$\tanh(x) = 2\sigma(2x) - 1 \tag{4.4}$$

$$ReLU(x) = f(x) = \max(0, x) \tag{4.5}$$

$$Leaky \; ReLU = f(x) = \max(\alpha x, x) \; ; \; usually \; \alpha \; is \; 0.1,0.001 \tag{4.5}$$

In summary, a perceptron takes many inputs and produce a binary output depending on inputs, weights and activation function.

4.2.2. The Neural Networks

With one perceptron, one can made very limited decisions. In many problems, the solution is complex and input output relation is not linear. To be able to solve more complex problems, collection of perceptron connection models, namely, neural networks are built. A typical Multi Layer Perceptron (MLP) network consists of input layer, output layer and a number of hidden layers which is shown in figure 4.4

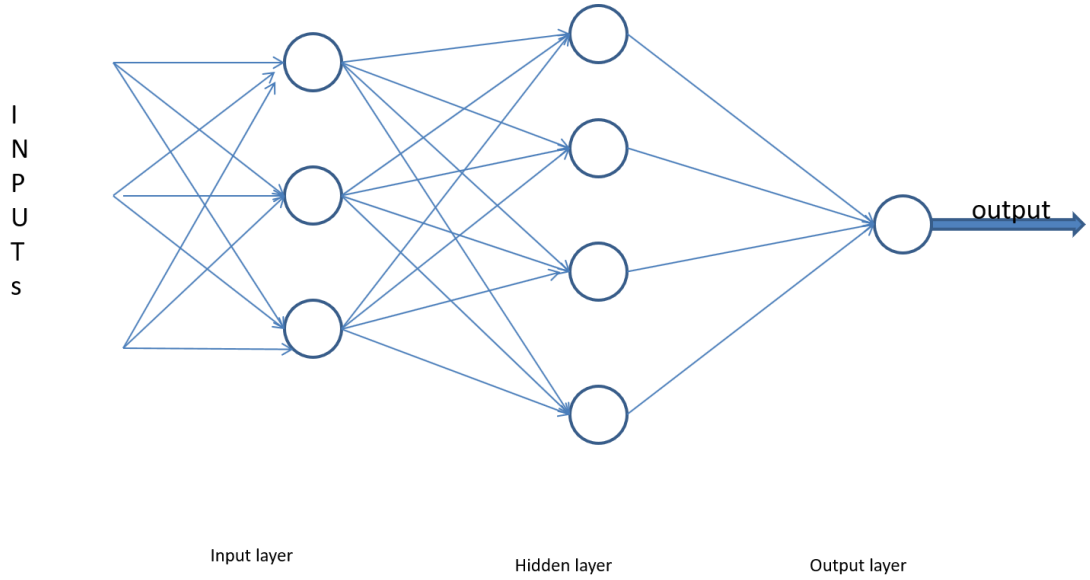


Figure 4.4. Multi layer perceptron network

For the network in figure 4.4, the first column of perceptron-input layer- makes three very simple decisions, by weighing the input evidence [43]. The second layer -hidden layer- is making a decision by weighing up the results from the input layer of decision-making. In this way a perceptron in the second layer can make a decision at a more complex and more abstract level than perceptrons in the first layer [43]. Also, even more complex decisions can be made by the perceptron in the third layer. In this way, a many-layer network of perceptrons can engage in sophisticated decision making [43]. Input layer shall be equal to the number of inputs i.e if we have a 20*20 pixel image, the input layer shall have 400 neurons. The above neural network is also an example of feed forward neural network.

Generally a network is trained with some known input-output pairs i.e. weights are updated (adapted) in order to minimize the error at the output. Training has two main parts. In the feed forward part input is fed to the network and the output obtained according to the activation functions of the neurons. In the second part (back propagation part) obtained, output is compared to the target or desired output and the error according to a cost function is calculated. Having obtained the error, the gradient

of it which is a multivariable generalization of the derivative is calculated and this gradient is back propagated through input layer. In [16] a gradient is defined as the gradient points in the direction of the greatest rate of increase of the function, and its magnitude is the slope of the graph in that direction. Backpropagation is used to calculate the gradient of the error [5] to update the weights in the network to minimize the error at the output i.e. after updating the weights, the actual output will be closer to the target output. An example network is seen in figure 4.5 and whole process is defined through (4.6) to (4.10)

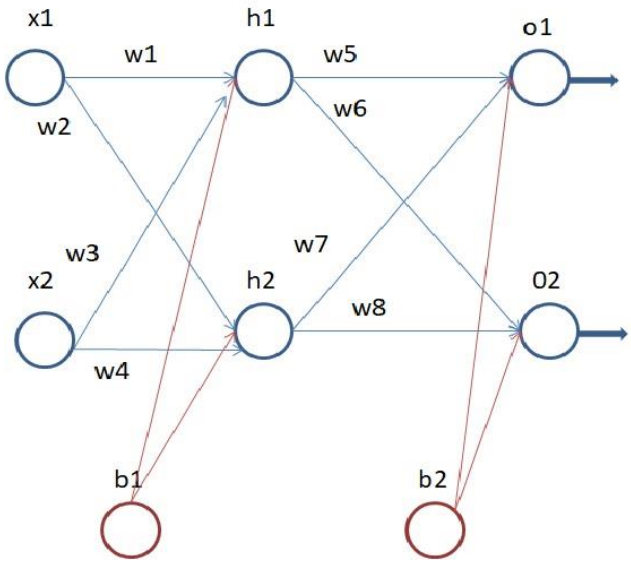


Figure 4.5. Back propagation example network

Forward Pass:

$$neth_1 = w_1 * x_1 + w_2 * x_2 + b_1 \tag{4.6}$$

neth₂ is calculated similarly.

$$outh_1 = \frac{1}{1 + e^{-neth1}} \tag{4.7}$$

outh₂ is calculated similarly

$$neto_1 = w_5 * outh_1 + w_6 * outh_2 + b_2 \tag{4.8}$$

$neto_2$ is calculated similarly.

$$outo_1 = \frac{1}{1 + e^{-neto1}} \tag{4.9}$$

Obtaining $outo_1$ and $outo_2$ error functions Eo_1, Eo_2 and total error E_{total} can be calculated. The error function depends on the designer choice. Most commonly used one is the mean squared error function. Assuming mean squared error function Eo_1, Eo_2 and E_{total} can be calculated as:

$$Eo_1 = \frac{1}{2} * (targeto_1 - outo_1)^2 \tag{4.10}$$

$$\mathrm{E_{total}} = \mathrm{Eo_1} + \mathrm{Eo_2} \tag{4.11}$$

Now is the backward phase. By calculating the gradient from error function to weights w_5 and w_6 we can compute how much a change in these weights affects the total error. By chain rule:

$$\frac{\sigma \mathrm{E_{total}}}{\sigma \mathrm{w_5}} = \frac{\sigma \mathrm{E_{total}}}{\sigma \mathrm{outo_1}} * \frac{\sigma \mathrm{outo_1}}{\sigma \mathrm{neto_1}} * \frac{\sigma \mathrm{neto_1}}{\sigma \mathrm{w_5}} \tag{4.12}$$

$$\frac{\sigma \mathrm{E_{total}}}{\sigma \mathrm{outo_1}} = -(targeto_1 - outo_1) \tag{4.13}$$

$$\frac{\sigma \mathrm{outo_1}}{\sigma \mathrm{neto_1}} = outo_1(1 - outo_1) \tag{4.14}$$

$$\frac{\sigma \mathrm{neto_1}}{\sigma \mathrm{w_5}} = outh_1 \tag{4.15}$$

Obtaining gradient of E_{total} according to the w_5 we can update w_5 as follows

$$w_5^{+} = w_5 - \eta * \frac{\sigma E_{total}}{\sigma w_5} \quad (4.16)$$

Where η is the learning rate, it is typically small i.e. 0.001. w_6 , w_7 and w_8 can be calculated in a similar way. Gradient of E_{total} with respect to w_1 can be calculated through (4.17) and w_1 is updated as like in (4.16)

$$\frac{\sigma E_{total}}{\sigma w_1} = \frac{\sigma E_{total}}{\sigma out_1} * \frac{\sigma out_1}{\sigma net_1} * \frac{\sigma net_1}{\sigma w_1} \quad (4.17)$$

4.2.3. Factors Affecting the Success of Neural Networks

The performance and the training efficiency of neural networks depend on activation function, learning rates and weight updates, back propagation and the cost (error) function

4.2.3.1. Activation Functions

In [40] it is stated that classical sigmoid function has three main disadvantages. Firstly, saturated neurons kill the gradient during backpropagation and weight update process. Secondly sigmoid outputs are not zero centered i.e. if always a positive input is fed to the network gradients are either always positive or negative. Finally, exponential functions are relatively computationally expensive.

Unlike sigmoid function tanh is zero centered but still it kills the gradients when saturated. On the other hand, ReLU does not saturate in (+) region. Converges much faster than sigmoid/tanh e.g. 6 times comparing to sigmoid [28]. However, it is still not zero centered and it kills the gradient when the input is negative. Leaky ReLU

solves the negative input and do not die in negative regions. Mostly ReLU and Leaky ReLU is used as an activation function.

4.2.3.2. Cost Function

In neural networks, cost function is a measure that shows how well the network is trained with respect to its input training data and target output. Gradient is calculated according to the cost function and with respect to the calculated gradients the weights are updated. To be able to compute gradients cost functions should be written as average and must depend on only the outputs of the network. There are many cost functions used by people such as Sum of Squared Error or mean squared error SSE, Cross Entropy CE and Exponential Cost EXP. SSE and CE are discussed in [37] and both of them are explained and compared in terms of their impact on the reconstruction performance of hidden layers in deep neural networks [1]. CE yields minimum errors on the other hand SSE provides best layer-wise reconstruction performance [1].

4.2.3.3. Learning Rates and Weight Updates

In neural networks the main purpose while training the network is to reduce the loss function or the cost function. The gradient is calculated and back propagated at each new input (or sample batch) and weights are updated similar to (4.16) with respect to the calculated gradient. Choosing the right learning rate in (4.16) is very important. With high learning rate improvements in the loss function will be exponential at first but loss may explode or get stuck around worst values of loss [41]. On the other hand, if it is too small, loss may not go down with the training data available. There are also many weight update procedures introduced in the literature. The basic form is steepest gradient descent SGD as introduced in (4.16). Choosing the right learning rate is

important, it is guaranteed to make non-negative progress in loss [41] with low learning rate but it converges to slowly. In physical terms gradient directly affects the position of convergence. Another update method is the Momentum update which is shown in (4.18) and (4.19). V in the equations represent velocity, in physics *mu* represents the friction. Here, gradient affects the velocity of convergence which in turn has an effect on position [41]. V is initially 0 and mu is chosen between 0.5-0.9. Overall momentum update is faster than SGD.

$$\mathrm{v} = mu * v - \eta * \mathrm{gradient} \tag{4.18}$$

$$\mathrm{w} = w + v \tag{4.19}$$

Another weight update procedure is the modified version of the momentum update namely Nesterov Momentum Update. It basically approximates the future position of convergence based on the gradient as a "lookahead" and update the parameters according to (4.20) through (4.22). Nesterov update shows slightly better performance than the momentum update.

$$\mathrm{w}_{\mathrm{ahead}} = w + mu * v \tag{4.20}$$

$$\mathrm{v} = mu * v - \eta * \mathrm{gradient\ of\ w\_ahead} \tag{4.21}$$

$$\mathrm{w} = w + v \tag{4.22}$$

There are also pre-parameter adaptive learning rate methods proposed by [12] called Adagrad shown in (4.23) and (4.24).

$$cache = cache + gradient^2 \tag{4.23}$$

$$\mathrm{w} = w - \eta * gradient/(\sqrt{cache} + eps \tag{4.24}$$

Eps is a smoothing term usually set between 1e-4 to 1e-8. Its convergence rate is well but it may stop learning too early in deep neural networks [41]. RMSprop and Adam

are other adaptive learning rates which are slightly modified versions of Adagrad but performs better. RMSprop is shown in (4.25) and (4.26). Decay rate is typically chosen as 0.9 or 0.99.

$$cache = decay_{rate} * cache + (1 - decay_{rate}) * gradient^2 \quad (4.23)$$

$$w = w - \eta * gradient/(\sqrt{cache} + eps \quad (4.24)$$

4.2.3.4. Weight Initialization and Regularization

Initialization process of weights in neural networks is one of the most important tasks to be achieved successfully. Assume we start with all zero initialization, if all the neurons in the network produce the same output, the gradient will remain the same for all the weights [42] and network will not converge. As an initial guess some of the weights will be negative and some will be positive. Starting with small numbers around zero mean with Gaussian distribution is one of the most used methods for weight initialization.

Another problem that can be faced while training neural networks is the over fitting. The weights are updated with the training set in a way that the actual output is very close to the target output, but the network fails to fit additional data or predict future observations reliably [47]. To overcome this problem a technique called "dropout" for effective regularization of the network is introduced in [51]. While training, dropout is implemented by only keeping a neuron active with some probability p (p=0.5 generally) or setting it to zero otherwise.

4.3. CONVOLUTIONAL NEURAL NETWORKS

As introduced earlier in the classical neural networks, the feature or the input to the network is constant. Only the weights between neurons are updated at each training sample batch. This may work well for some problems; however, sometimes the problem is not so easy to solve. We need more complex features for solving the problem i.e. truly classifying the objects. Also, decision of the feature is not an easy task to carry out. We may not be able to extract the right feature for proper classification. Moreover, as Hubel & Wiesel showed in their successive studies [21][22], cats' visual cortex system neurons are specialized in object feature extraction. Featural hierarchy between the neurons is visualized in figure 4.6 [20].

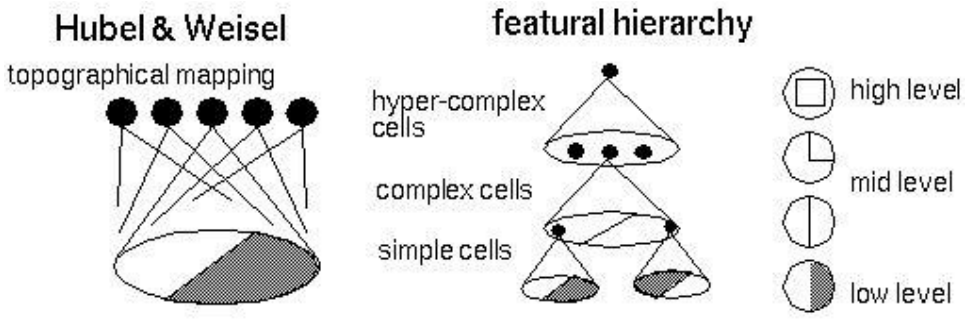


Figure 4.6. Featural hierarchy in cats' visual cortex neurons [20]

Inspired from the cats' visual cortex hierarchy LeNet is introduced in [29]. Convolutional Neural Networks (CNN) are a specialized version of ordinary neural networks. They are trained in a similar way and the weights are updated using back-propagation. What differs in Convolutional Neural Networks is the architecture of it. Visual patterns in pixel images are directly recognized by the CNN with robustness to distortions and geometric transformations [30]. A simple example of Convolutional Neural Networks is seen in figure 4.7 [29]. That is, "LeNet-5" is introduced in [29]. It

is designed to detect handwritten letters. As can be seen from the figure, Convolutional Neural Networks consists of Convolution layers, Subsampling (pooling) layers, filters applied in convolution layers and finally a fully connected neural network at the end to classify the objects (obtained complex features). These layers will be explained in detail in the following section.

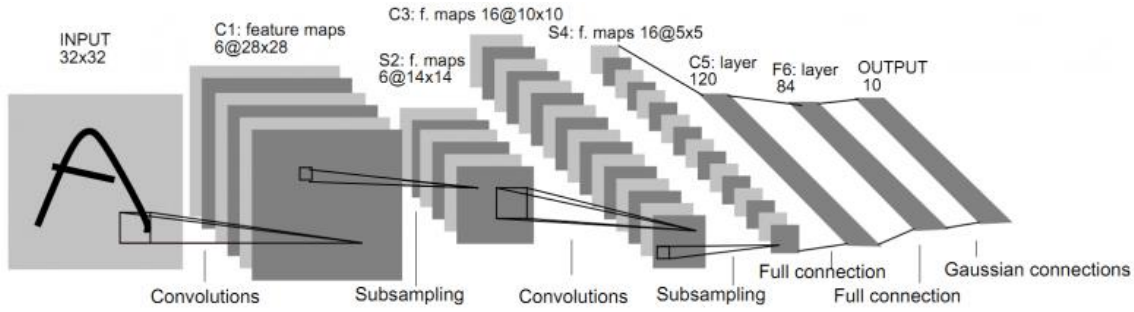
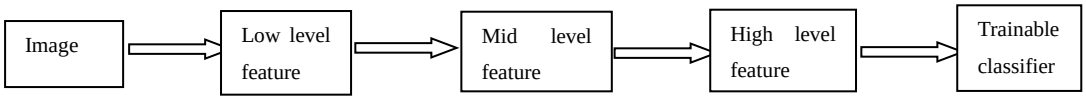


Figure 4.7. LeNet-5 [29]

Convolution and pooling layer make the feature extraction part of the network whereas the fully connected layer makes the classification. The CNN is also called deep learning due to more than one non-linear feature extraction. The CNN structure can be summarized as in the chart below.



While we obtain simple features from images like edges in classical classifiers, in Convolutional Neural Networks, we obtain complex layered features first layer includes edges second layer motif and final layer the object itself. To sum up, classical pattern recognition methods use fixed i.e. engineered features and simple trainable classifiers whereas CNN consists of trainable feature extractor and trainable classifiers.

4.3.1. The Convolutional Layer

Convolutional (henceforth Conv) layer is the core part of CNN structure and does the most computational part. Conv layer consists of small filters (in width and height) which are trainable and extend through the full depth of input images volume. In the forward phase Each filter slides across the width and height of the image, dot product is computed between the filter and the input part of the image (convolve the filter with the image) and two-dimensional activation map produced at the end. As stated earlier, these filters are trainable i.e. they are updated through back-propagation until they see some visual features like edge or motif or the object itself which depend on the layer that the filter belongs to. By sliding one filter over the entire image first hidden layer is obtained [11]. Applying the filter i.e. sliding it through the entire image, computing the dot product and obtaining the hidden layer are shown in successive figures 4.8 and 4.9.

As seen from the figures, while sliding the filter across the image, the width and height of the image get smaller at the output because we can slide the filter up and down less than the size of the width. For example, in figure 4.8 and 4.9 the input is 7x7 image, the filter size is 3x3 and we can slide the filter just 5 times in each direction unless a stride is applied during sliding. If we applied the filter with stride 2, we could slide the filter only 3 times in each direction. The spatial output of the general formula is shown in (4.25). N is the image size; F is the filter size

$$output\ size = \frac{(N-F)}{stride} + 1 \tag{4.25}$$

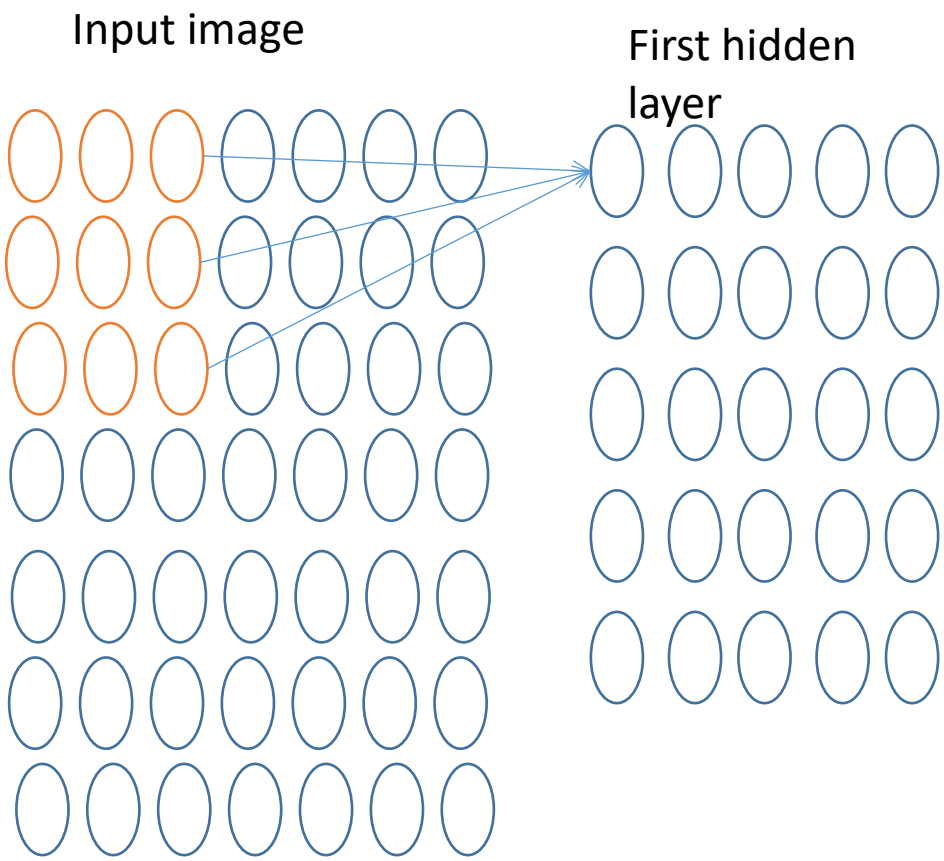


Figure 4.8. Sliding the filters and computing the dot product

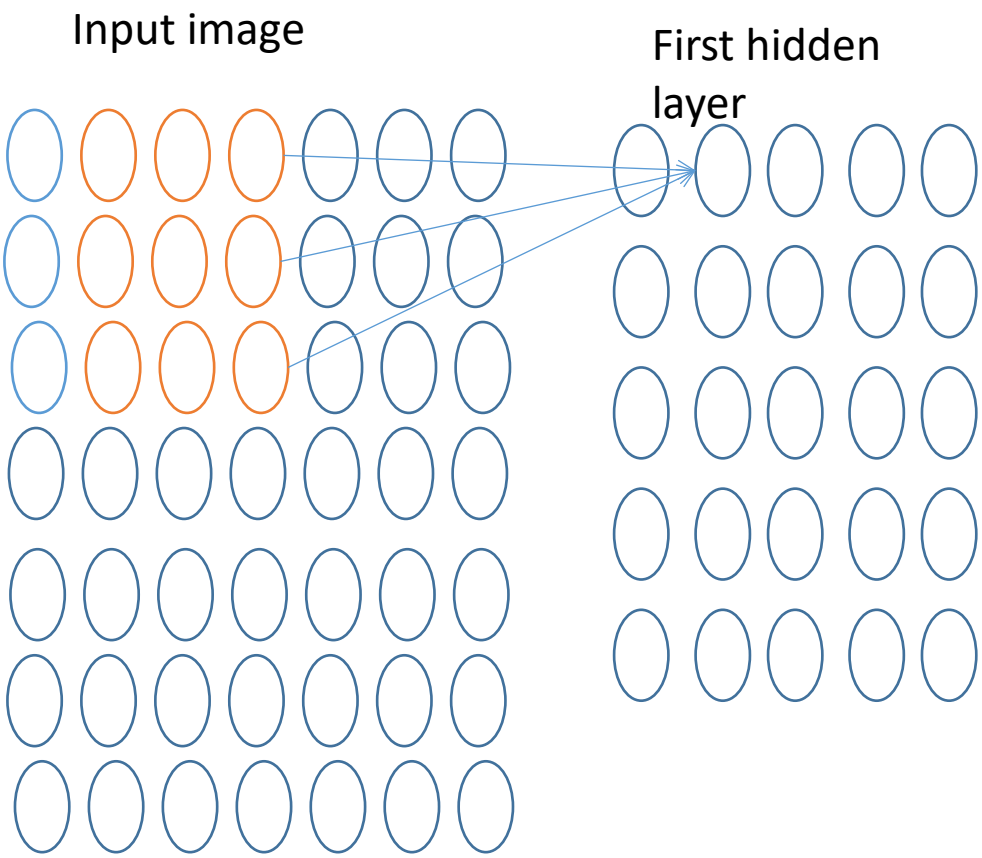


Figure 4.9. Sliding the filters and computing the dot product

If we have “n” filters at that layer, at the end we obtain “n” hidden layer and the input volume transforms to the number of filters e.g. if we have 28*28*3 image as input and have 3*3*3 10 filters at the first conv layer after the forward phase, we have a 26*26*10 volume at the output and this will be input to the next layer. The obtained output at the end of the layer is illustrated in figure 4.10. This is sometimes called the feature map.

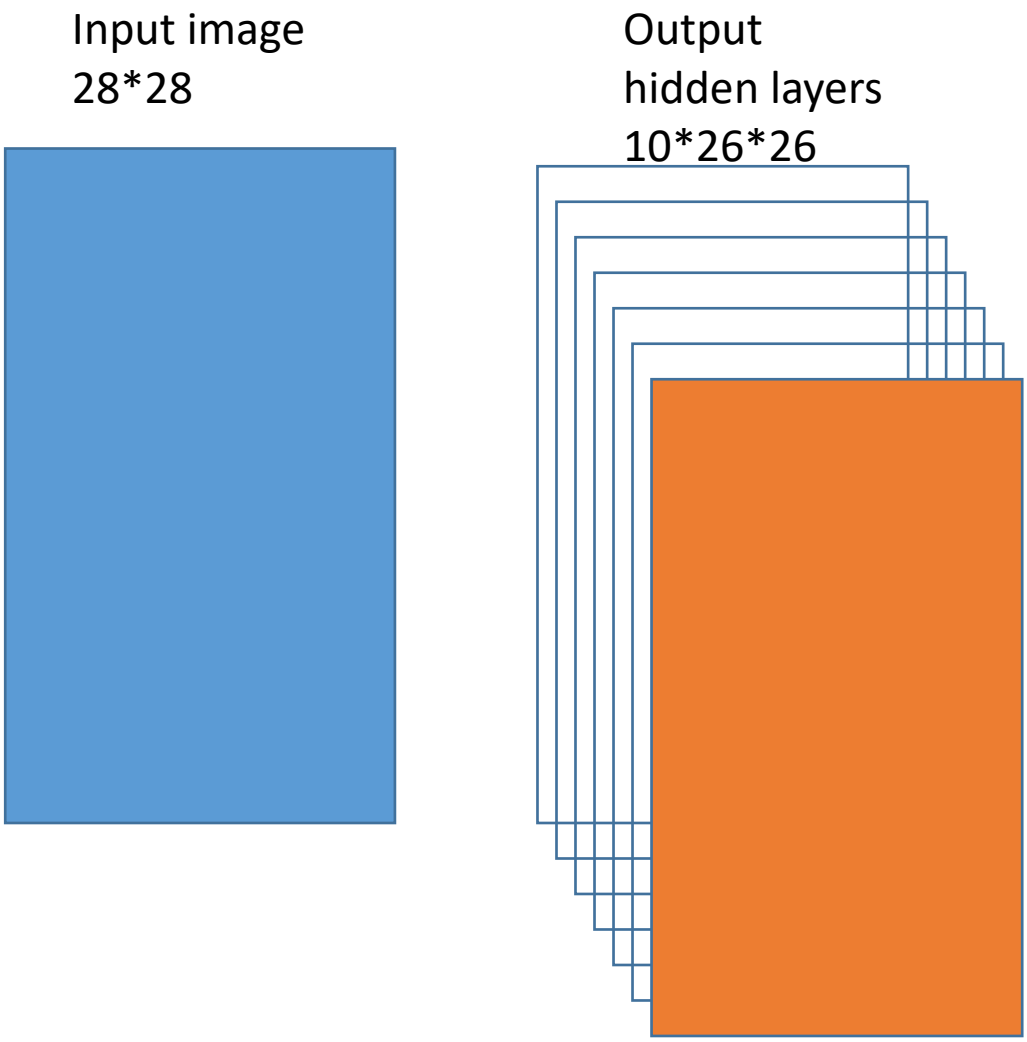


Figure 4.10. Conv layer output-feature map

3-D input volume is transformed to the 3-D output volume of neuron activations. Neurons are arranged in three dimensions. While sliding the filter at the end of each dot product computation, ReLU activation is applied to the output of the dot product. Because of that, sometimes conv layer is named Conv-ReLU layer.

It is seen that while applying the filters to the image the width and height of the input decays. As the number of the conv layer increases the size gets smaller and smaller.

To prevent this outside of the image is zero padded so that the out size does not change after applying the filters. Zero padding example is shown in figure 4.11.

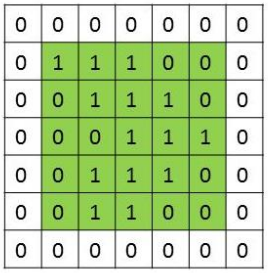


Figure 4.11. Zero padding example

Assuming the filter size F by F to preserve the input size spatially, the input image is zero-padded in each direction according to

$$size\ of\ zero - padding = (F - 1)/2 \tag{4.26}$$

During the back-propagation phase, the filter values are updated. This provides the learnable feature extraction part of Convolutional Neural Networks. The number of parameters to be updated is equal to the number of filters multiplies the total volume of the filter e.g. if we have 10 7*7*3 filters to be applied to the input image 10*7*7*3=1470 parameters are updated during back-propagation phase

4.3.2. Pooling Layer

In addition to the convolutional layers, CNN also contains pooling layers. Pooling layer is generally applied after the convolutional layer and down sample the output. This process simplifies the information contained at the output of convolutional layer. It reduces the size of representation number of parameters and computation in the

network [9]. Pooling layer processes each hidden layer independently and resizes them spatially using Max operation. In the most common usage, 2x2 max filter is applied in pooling layer and at each step max value is taken from 4 neighbor pixels. The max pooling process is shown in figure 4.12. During pooling processes, we lose some of the information stored but gain a lot of computing time instead. After pooling, we still have many meaningful parts of the features.

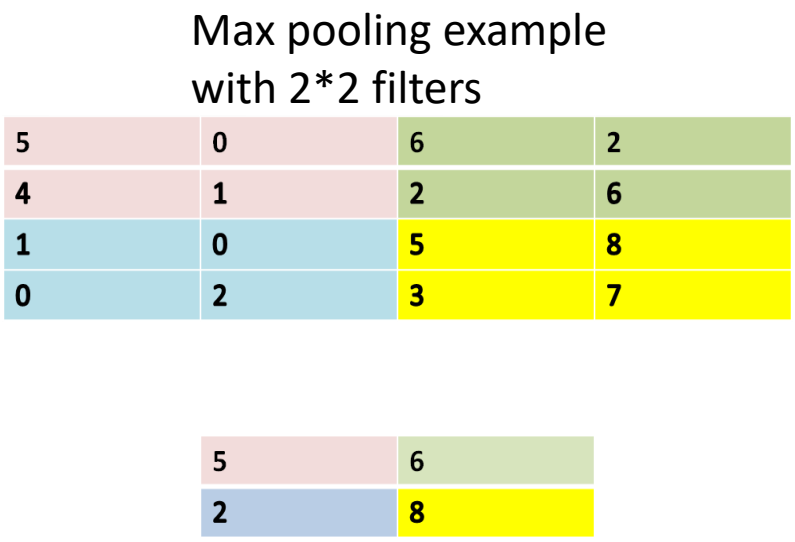


Figure 4.12. Max pooling with 2x2 filters

Instead of max pooling, other pooling techniques can be applied like average pooling; however, generally max pooling shows better performance [9].

Convolutional layer together with the pooling layer is one of the building blocks of the Convolutional Neural Networks. Bringing a number of conv layers together with some pooling layer between them, we constitute the "deep network". At each conv layer applied to the output of previous conv layer, complexity and volume of the features obtained increases.

4.3.3. Fully Connected Layer

The output of the last convolutional layer (or pooling layer) is introduced to a fully connected neural network (henceforth FC layer) . The input layer neurons of FC layer are all fully connected to last output obtained in convolutional part. The activations can be computed using matrix multiplications and weights are updated as like in the back-propagation stage of neural networks. FC layer serves as a classifier part of the CNN

4.3.4. Cost Function

There are many cost functions that are used at the end of neural networks to calculate the gradient that is used to be in backpropagation phase. Some of them already mentioned in section 4.2.3.2. Softmax is commonly used as a loss function in Convolutional neural networks. Soft max is a generalization of the logistic function that squashes a K-dimensional vector of arbitrat real values to a K-dimensional vector of real values in the range (0,1) that add up to 1[52]. The main motivation of using softmax is that it can an approximate taking maximum operation. Softmax emphasis the highest probability element of the prediction vector and suppress the lower elements using the exponential characteristic of the function. Moreover, the continuously differentiable property of the softmax helps users to produce a robust classifier model using related softwares. It is mostly used for multi class classification at the output of final neural network layer (fully connected layer) and it is very efficient in terms of calculation of each class probabilities comparing to other methods [33].

4.4. THE APPLIED CONVOLUTIONAL NEURAL NETWORK STRUCTURES

Convolutional Neural Networks, as discussed earlier, resemble the cat featural hierarchy. With the convolutional layers, we obtain and accomplish adaptive and complex feature extraction. With the traditional methods discussed in chapter 3, feature sets have to be static and because dirt, patching and a pothole may show similar low-level features it was impossible to obtain a successful classification with very low error. Either depth information must be included to the feature set or more complex features should be used for better classification.

There are many CNN structures introduced to literature like [57], [50], [15] and [48]. GoogleNET, LeNET, AlexNET, ZFNET, VGGNET, ResNET, YOLO, R-CNN are examples of most famous structures used in many areas such as digit recognition, object detection and activity detection. There are many convolutional neural networks introduced to detect different type of objects. In [59], recent CNN methods' classification success rate for different type of objects and datasets are compared. These networks compared in [59] are too deep and are designed to obtain very complex feature sets. On the other hand, although pothole, dirt, patch and manhole may have similar edges, gradients or textures and there need to be adaptive feature sets to dissociate them, these anomalies do not possess too complex features. In [3], different CNN's that are constructed for age and gender classification are compared in terms of their success rate. Inspired by these works, it is assumed that near-age people also possess similar features and hence methods that are successful in age classification can also be applied to pothole classification. It is seen in [3] that method introduced in [31] is very successful at gender and age classification and is composed of simple convolutional blocks to apply. A 3 Convolutional layer CNN structure (henceforth 3 conv CNN), followed by 3 fully connected layer, very similar to [31] is

constructed, tested and compared with vision-based methods. Seeing that this method is very successful against vision-based methods, depth of the network needed for pothole classification is also investigated. On account of this 4 & 1 convolutional layers (henceforth 4 conv CNN & 1 conv CNN), followed by 3 fully connected layers are also constructed and tested. 227*227*3 images are introduced as inputs for training 3 conv and 1 conv layer networks. 242*242*3 images used for 4 conv layer network

4.4.1. 3 Convolutional Layer Network

This structure is very similar to [31]. In the first conv layer, 96 units of 7*7*3 filters are used with stride 4 and no padding. A rectifier linear unit and a 3*3 pooling layer follow each conv layer. Pooling layer gets the max of each 3*3 segment and with a stride of 2. Thus, the output is down sampled to half of its size. 56*56*96 hidden layer is obtained at the output of the first conv layer. The second conv layer consists of 256 units. 5*5*96 filters with a stride 1 and 2 zero padding and final conv layer consists of 384 units 3*3*256 filters with stride 1 and 1 zero padding. The final output is 7*7*384 hidden layer.

Following the 3 convolutional layers 3 fully connected layer 512*512*2 is used. To connect the 7*7*384 feature set to the first FC layer which contains 512 perceptrons, a conv layer containing of 512 units of 7*7*384 filters with stride 1 and zero padding are used. The structure of the network is shown in figure 4.13.

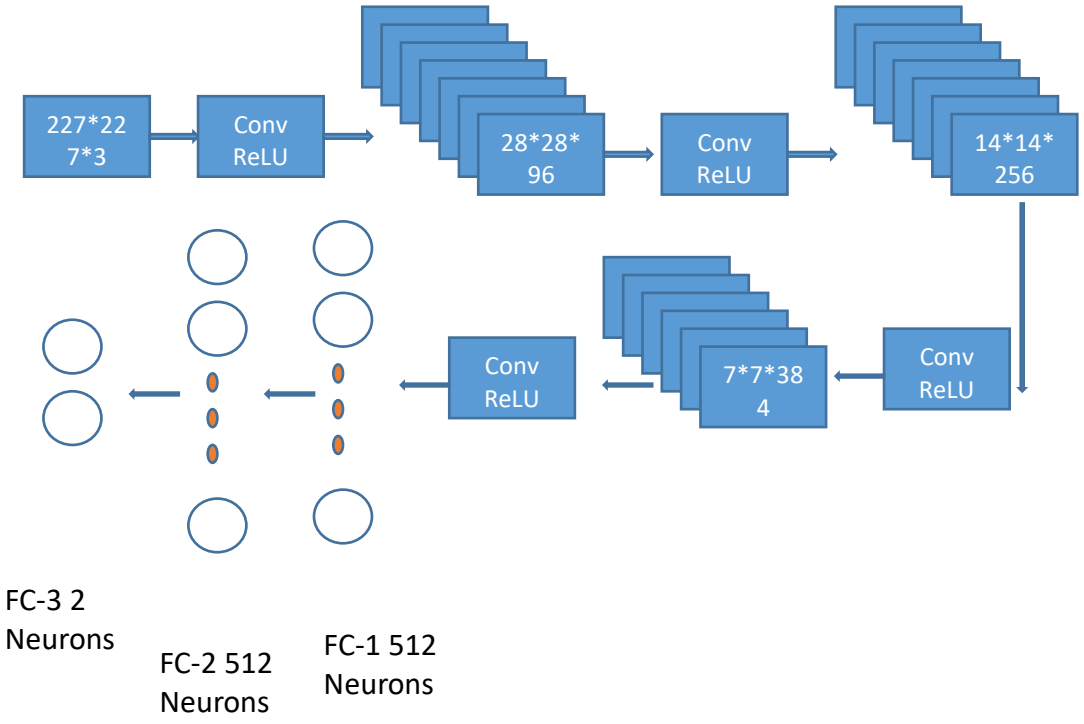


Figure 4.13. Structure of the 3 Conv CNN

For the activation function again ReLU is used in FC layers. For the weight update, method momentum update is used. First, a bigger learning rate is used; however, it exploded and get stuck around local minimas than a smaller a value is chosen. 3e-4 is used as learning rate and mu is taken as 0.9 for the momentum. At the output of final FC layer softmax is used as loss function.

4.4.2. 4 Convolutional Layer Network

In the first layer of the 4 Conv layer, 48 units of 9*9*3 filter is used with stride 1 and no padding. A rectifier linear unit and a 3*3 pooling layer follow each conv layer. Pooling layer again down sample the input to half of its size. The second conv layer consists of 96 units 7*7*48 filters with stride 2 and zero padding. Third layer consists

of 256 units 5*5*96 filters with a stride of 1 and 2 zero padding and final conv layer consists of 384 units of 3*3*256 filters with stride 1 and 1 zero padding. The structure of the network is shown in figure 4.14.

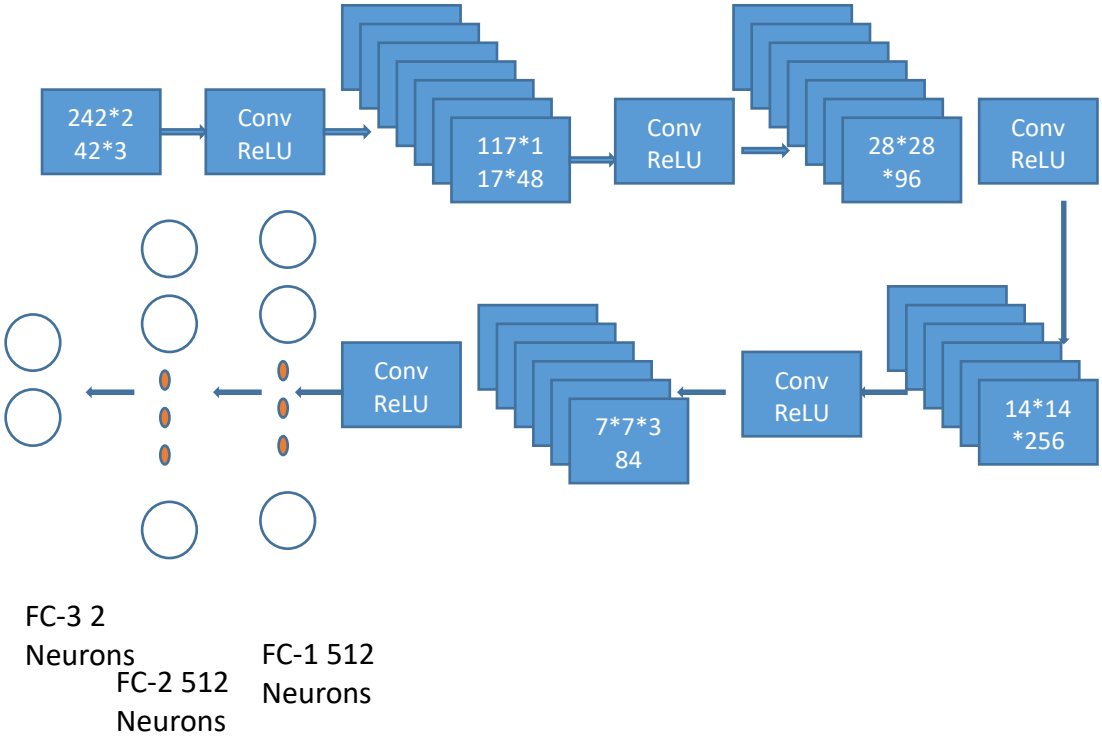


Figure 4.14. Structure of the 4 Conv CNN

4.4.3. 1 Convolutional Layer Network

In the first conv layer, 96 units of 7*7*3 filters are used with stride 4 and no padding. Following the convolutional layer 3 fully connected layer 512*512*2 is used. To connect the 56*56*96 feature set to the first FC layer which contains 512 perceptrons, a conv layer containing of 512 units of 56*56*96 filters with stride 1 and zero padding are used. The structure of the network is shown in figure 4.15

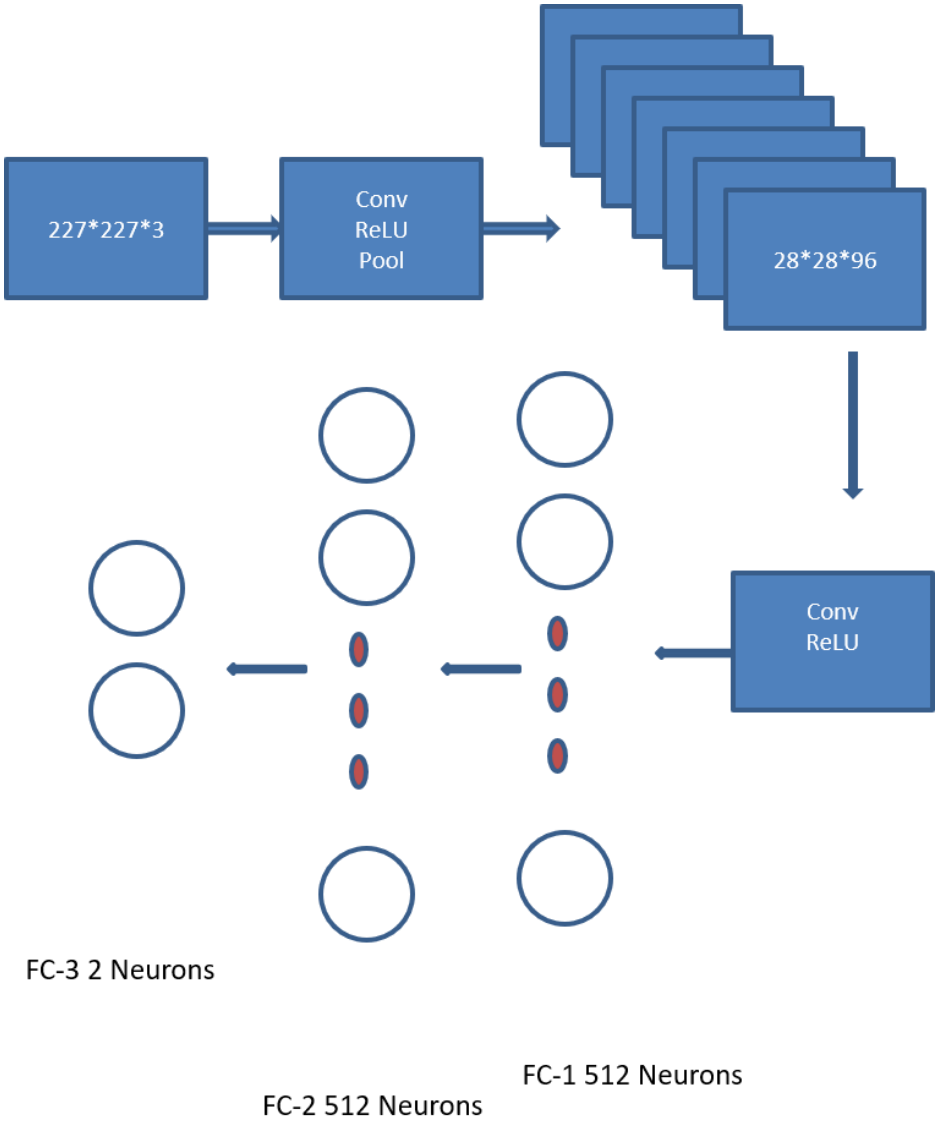


Figure 4.15 Structure of the 1 Conv CNN

For the implementation of the applied 3 Conv, 4 Conv and 1 Conv layer CNNs matlab (2015) and a toolbox called MatConvNet for implementing Convolutional Neural Networks for computer vision applications are used. The result of the CNN and other methods which have been mentioned in chapter 3 will be discussed at chapter 5.

CHAPTER 5

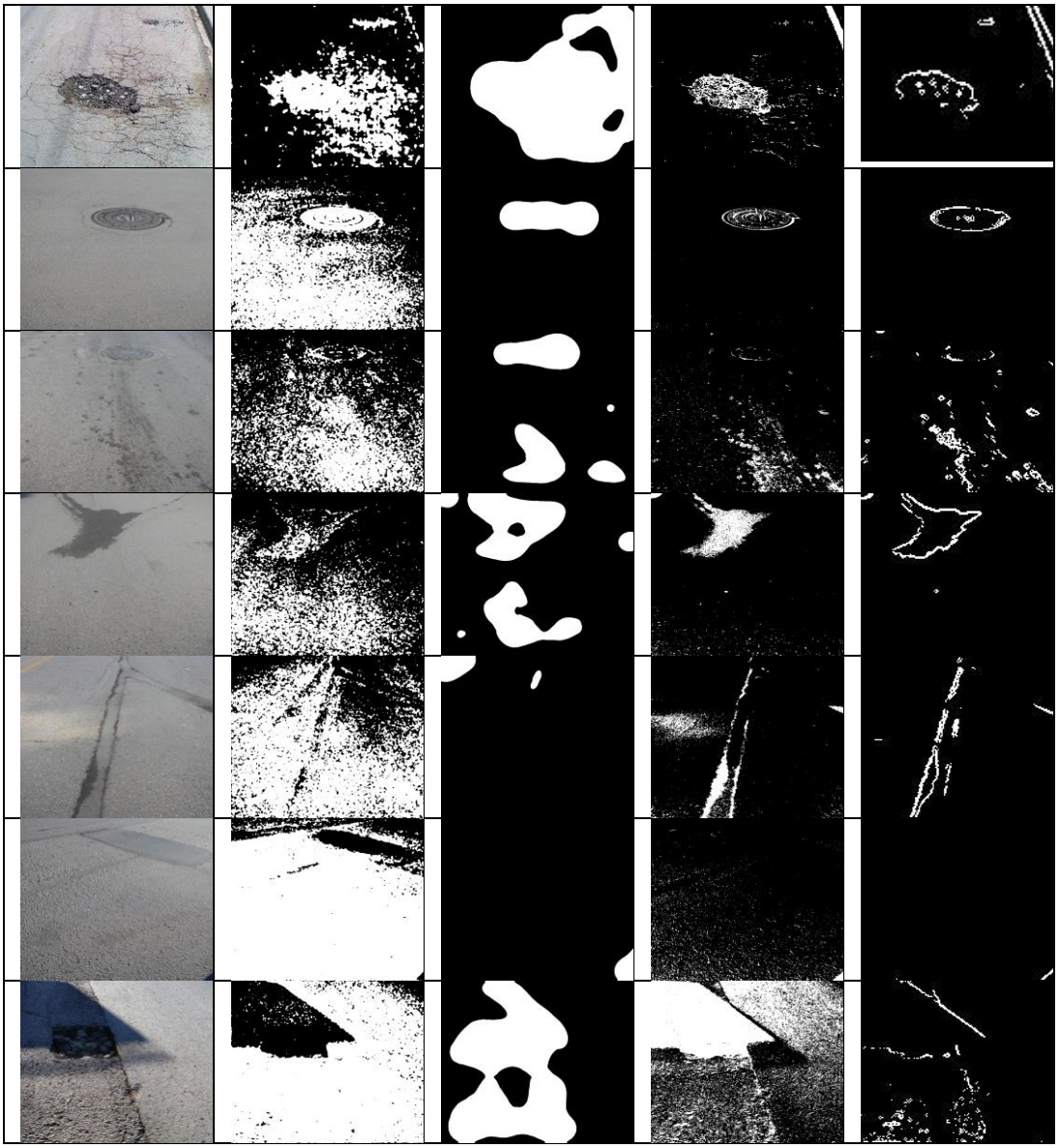
TESTS AND RESULTS

5.1. INTRODUCTION

The main goal of this study is to develop a method to detect potholes as soon as they are formed and inform the authorities to fix them in time before any damage happens to drivers or cars. The success rate must be satisfactory enough so that this system can be used in a healthy way i.e. if the system gives lots of false alarms, it will not be useful after a while. As stated earlier, anomaly detection on the road surfaces can be detected with more than %96 accuracy in [35]. Also, saliency methods can also be used to detect anomalies on the road surfaces. In Table 5.1 we can see the results of some saliency methods which are used in [17] and applied on randomly chosen 10 images.

Table 5-1. *Different anomaly detection method output images*

Original Frame	Spectral Residual	Image Signature	Frequency Tuned	Gradient Based
				



As can be seen from the saliency methods like Spectral Residual or Image Signature, anomalies can be detected with a very high precision. The main problem is to decide whether the detected anomaly is a pothole or not. The main anomalies that can be found on road surfaces are potholes, dirt, patching and manholes. Some example images for these anomalies are shown in Figure 5.1, 5.2, 5.3 and 5.4. My study begins

just after the anomalies detected in asphalt images. Anomaly detection method can be regarded as front step of this study.

Figure 5.1. Pothole examples

Figure 5.2. Dirt examples

Figure 5.3. Manhole examples

Figure 5.4. Patch examples

Examining the figures one can see that a pothole do not have a specific pattern and non pothole images can easily be classified as pothole images with basic feature extracting and classification methods. As stated earlier the method shall have a satisfactory and reliable success metrics. Assuming corrrectly detected potholes condition as True Positives (henceforth TP), correctly detected non potholes condition as True Negatives(henceforth TN), missing the detection of potholes as False Negatives (henceforth FN) and false detection condtion of potholes as False Positives (henceforth FP), the aim is to achieve better than 90% success rate which is formulated as

$$Success\ Rate = \frac{(TP+TN)}{Total\ Samples} * 100 \quad (5.1)$$

Moreover, achieving a False Positive percentage defined in (5.2) less than 4% to avoid false alarms as much as possible is the second main goal of the study.

$$False\ Alarm\ Rate = \frac{(FP)}{Total\ Samples} * 100 \quad (5.2)$$

5.2. DATA PREPARATION

Data is collected for training and testing purposes both from the Internet and Ankara roads. The images from Ankara roads are taken with Samsung S3 mini cell phone. 546 non-pothole images (190 Dirt, 177 manhole, 165 patching) were all taken from the

camera and 474 pothole images (179 taken with camera 295 downloaded from the Internet) were used in the tests.

For the comparison of visual based methods with Convolutional Neural Networks 10 sets are prepared for training and testing. For each set, 60 images are chosen randomly for the test and the rest is used for training when required. None of the sets contains an image in common. 60 images consist of 36 non-pothole (12 from each) and 24 pothole (12 from each) images. For comparison of the 3 Conv and 4 Conv CNN and also comparison of different filters used in 3 Conv networks, 5 different sets are prepared.

For the initial tests used in chapter 5.3.2, 10 different sets are prepared. Again, for each set, 60 images are chosen randomly as test data and 240 images are used for training. Test images consist of 24 pothole and 36 non-pothole images. For the ANN method all the images are labeled and given to the network. Matlab automatically chooses validation, training and test sets.

For the vision-based methods 510 non-pothole 450 pothole images are used for training. For the 3 Conv and 4 Conv CNN networks, images taken with the cell phone are rotated 30 degrees (each image is multiplied 12 times) and images obtained from the Internet are rotated 90 degrees (each image is multiplied 4 times). Test images are never used in training sets. Moreover, neither in the validation nor in the training are the rotated images used. 2820 pothole and 5280 non-pothole images are used for training CNN networks. 316 pothole and 600 non-pothole images are used for validation in the training of CNN networks. Total of 9016 images used in training.

5.3. TESTS

5.3.1. Introduction

In the first part of the tests, existing methods in the literature are applied and tested on the collected images and the success rate is investigated through Red, Green and Blue channels. In the second part of the tests, existing methods are compared with 3 Convolutional layer CNN method and in the latter part of the tests, improving the performance of the CNN methods are investigated.

5.3.2. Vision Based Methods in the Literature and Tests in R,G,B Channels

First vision-based methods introduced in chapter 3: Method for Automated Assessment of Potholes (will be called Test Morph), HOG Features Descriptors and Bayesian Classifiers (henceforth Hog + Bayesian), HOG Features Descriptors and SVM Classifiers (henceforth HOG + SVM) are implemented. Tests are done explicitly on red, blue and green channels along with the full RGB images. The main aim is to investigate whether a specific channel provides more information about potholes. Test and training images are chosen randomly, and this process is repeated 10 times. The results are shown in Table 5.2. All the values are given in percentage i.e.

$$Average\ True\ Positives = \frac{(TP)}{Condition\ Positive} * 100 \quad (5.3)$$

$$Average\ False\ Positives = \frac{(FP)}{Condition\ Negative} * 100 \quad (5.4)$$

$$Average\ True\ Negatives = \frac{(TN)}{Condition\ Negative} * 100 \quad (5.5)$$

$$Average\ False\ Negatives = \frac{(FN)}{Condition\ Positive} * 100 \quad (5.6)$$

$$Average\ STD = \frac{(STD)}{Condition} * 100 \quad (5.7)$$

Table 5-2. *Results of vision based methods tests*

		Test Hog+ Bayesian			Test Hog+SVM			Test Morph		
True Positives		R	G	B	R	G	B	R	G	B
	Avg	40.83	40.83	40	41.6	41.6	41.25	41.6	45	49.58
	STD	8.96	8.96	9.46	10.58	10.76	9.71	11.62	14.80	15.02
RGB		38.33			44.17			42.50		
		10.72			10.61			14.27		
True Negatives										
	Avg	75.56	75.83	75.56	78.89	81.39	78.06	79.44	78.33	82.22
	STD	4.68	5.56	5.20	5.59	3.94	5.15	5.59	5.83	5.43
RGB		74.44			78.33			81.11		
		4.86			5.68			6.11		
False Positives										
	Avg	24.44	24.17	24.44	21.11	18.61	21.94	20.56	21.67	17.78
	STD	4.68	5.56	5.20	5.59	3.94	5.15	5.59	5.83	5.43
RGB		25.56			21.67			18.89		
		4.86			5.68			6.11		
False Negatives										
	Avg	59.17	59.17	60.00	58.33	58.33	58.75	58.33	55.00	50.42
	STD	8.96	8.96	9.46	10.58	10.76	9.71	11.62	14.80	15.02
RGB		61.67			55.83			57.50		
		10.72			10.61			13.18		

It can be seen from the table that using a specific channel does not improve the success rate of the performance significantly and the results are far from reaching the performance metrics defined in (5.1) and (5.2).

5.3.3. Success of ANN Network

The artificial neural network that is constructed in section 3.3.4. is trained, validated and tested with all the images. The best success rate that is obtained with this structure is approximately 79 percent which is shown in figure 5. 5. Evaluating this method along with other vision-based method, it is seen that obtaining a satisfying successful result with static features is not feasible.

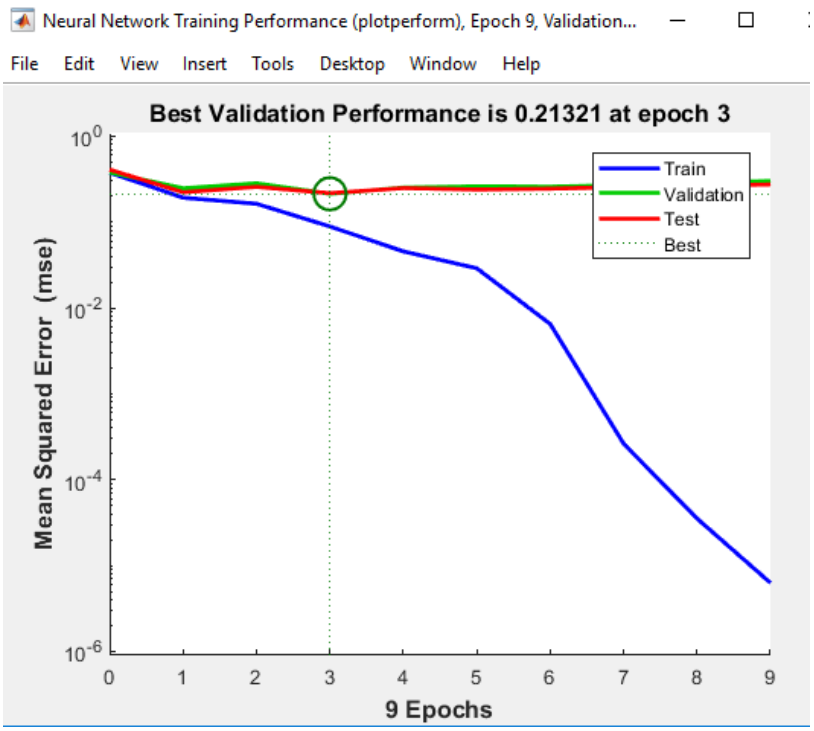


Figure 5.5 Success rate graph of ANN

5.3.4. Vision Based Methods in Literature and 3-Conv CNN

In test 3, Convolutional layer CNN is implemented, tested and compared with the above vision-based methods. The method is mainly used in face detection, namely, in Haar-like Features and Linear Cascade Classifiers (henceforth Haar + Linear or Linear Cascade) is also implemented and tested. 10 sets of data defined in section 5.2 are used for tests. Success rate of each method is shown in Table 5.3 and False Alarm Rate is shown in Table 5.4

Table 5-3. *Success rate of vision based methods and 3 Conv CNN*

	Test Hog+Bayesian	Test Hog+SVM	Test Linear Cascade	Test Morph	Test 3-Conv CNN
Success Rate	73.17	71.50	54.33	69.00	86.33

Table 5-4. *False alarm rate of vision based methods and 3 Conv CNN*

	Test Hog+Bayesian	Test Hog+SVM	Test Linear Cascade	Test Morph	Test 3-Conv CNN
False Alarm Rate	11.83	15.17	33.67	17.67	3.67

Average and standart deviation of methods and their percentages are shown in Table 5.5

Table 5-5. *Test result of 3 Conv CNN and vision based methods*

		Test Hog+Bayesian	Test Hog+SVM	Test Linear Cascade	Test Morph	Test 3-Conv CNN
%Average	True Positives	62.50	66.67	70.00	66.67	75.00
	True Negatives	80.28	74.72	43.89	70.56	93.89
	False Positives	19.72	25.28	56.11	29.44	6.11
	False Negatives	37.50	33.33	30.00	33.33	25.00
		Test Hog+Bayesian	Test Hog+SVM	Test Linear Cascade	Test Morph	Test 3-Conv CNN
Average	True Positives	15.00	16.00	16.80	16.00	18.00
	True Negatives	28.90	26.90	15.80	25.40	33.80
	False Positives	7.10	9.10	20.20	10.60	2.20
	False Negatives	9.00	8.00	7.20	8.00	6.00
		Test Hog+Bayesian	Test Hog+SVM	Test Linear Cascade	Test Morph	Test 3-Conv CNN
St. Dev	True Positives	2.16	2.40	4.13	2.11	3.13
	True Negatives	1.97	2.47	6.76	3.24	1.40
	False Positives	1.97	2.47	6.76	3.24	1.40
	False Negatives	2.16	2.40	4.13	2.11	3.13
		Test Hog+Bayesian	Test Hog+SVM	Test Linear Cascade	Test Morph	Test 3-Conv CNN
%St.Dev	True Positives	9.00	10.02	17.21	8.78	13.03
	True Negatives	5.47	6.86	18.79	9.00	3.88
	False Positives	5.47	6.86	18.79	9.00	3.88
	False Negatives	9.00	10.02	17.21	8.78	13.03

And finally, success rate of each method in each test is shown in Figure 5.6

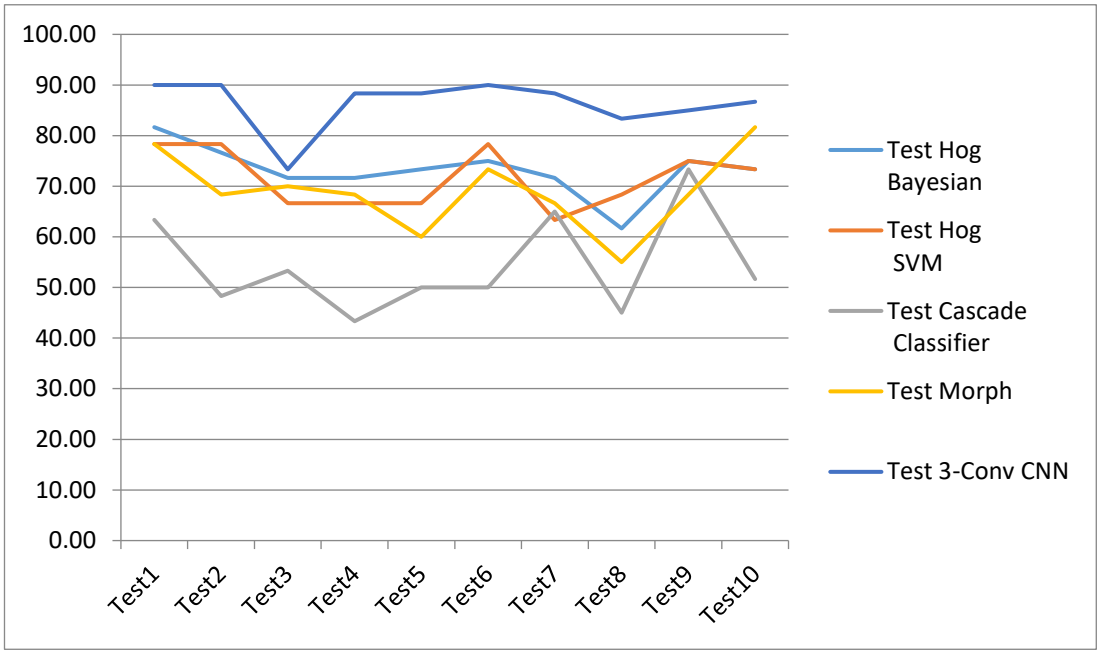


Figure 5.6. Success rate of each method

As can be seen from the tables Convolutional Neural Network increases the success rate of pothole detection significantly. In the average of the 10 tests, the success rate of CNN is at least 13 percent higher than other methods. Moreover, it is also seen in Table 5.4 that false alarm rate of 3-Conv CNN is much smaller than the other methods. It is a critical parameter in terms of having a stable and usable system. On the other hand, training time of Convolutional Neural Networks lasts much longer than the classical methods. Figure 5.7 shows an error rate versus number of epochs obtained in training of 3-Conv CNN. 1 epoch means 1 forward pass + 1 backward pass and each epoch lasts around 8 times as much as classical methods. It lasts around 160 epochs to have the CNN network to be trained.

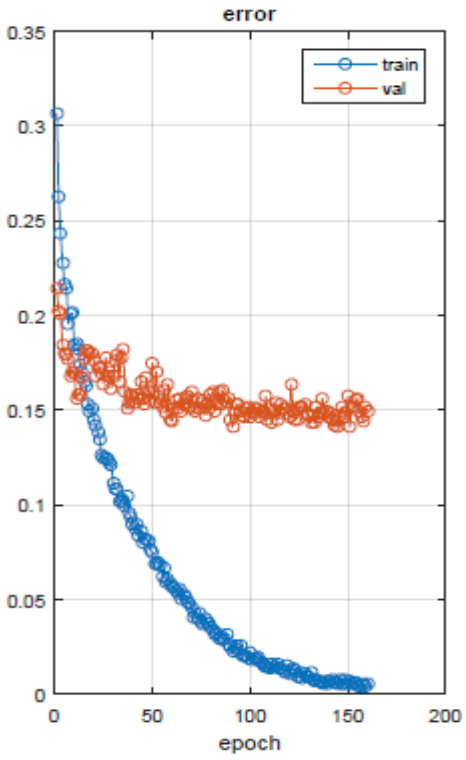


Figure 5.7. Error rate vs number of epochs in training 3 Conv CNN

5.3.5. Comparison of 3 Conv and 4 Conv CNNs

As seen in the previous tests, CNN improves the success rate significantly however it is still not at the desired level. To improve the success rate of the 3 Conv CNN the depth is increased and a 4 Conv CNN is constructed as described in chapter 4. The main goal is to see whether increasing the depth of the network increases the performance. 5 Test is done over 5 new sets of data. The average success rate is shown in Table 5.6 and false alarm rate is shown in Table 5.7

Table 5-6. *Success rate of 3 Conv and 4 Conv CNN*

	Test 4-Conv CNN	Test 3-Conv CNN
Success Rate	87.00	85.33

Table 5-7. *False alarm rate of 3 Conv and 4 Conv CNN*

	Test 4-Conv CNN	Test 3-Conv CNN
False Alarm Rate	4.00	5.33

Averages and standart deviation of methods and their percentages are shown in Table 5.8

Table 5-8. *Test results of 3 Conv and 4 Conv CNN*

		Test 4 Conv CNN	Test 3 Conv CNN
	True Positives	77.50	76.67
	True Negatives	93.33	91.11
	False Positives	6.67	8.89
%Average	False Negatives	22.50	23.33

		Test 4 Conv CNN	Test 3 Conv CNN
	True Positives	18.6	18.4
	True Negatives	33.6	32.8
	False Positives	2.4	3.2
Average	False Negatives	5.4	5.6

		Test 4 Conv CNN	Test 3 Conv CNN
	True Positives	1.14	1.52
	True Negatives	1.14	1.30
	False Positives	1.14	1.30
St. Dev	False Negatives	1.14	1.52

		Test 4 Conv CNN	Test 3 Conv CNN
	True Positives	4.75	6.32
	True Negatives	3.17	3.62
	False Positives	3.17	3.62
%St.Dev	False Negatives	4.75	6.32

It is seen from table 5.6 that increasing the depth of the networks slightly improves the success performance. Also, the average false alarm rate which is 5.33 % in 3 Conv CNN also improves up to 4% in 4-Conv CNN. However, training time of 4-Conv CNN is around 2.5 times longer than the 3-Conv CNN.

5.3.6. Tests with Different Size Filters in 3 Conv CNN

3 Conv CNN has 96 units 7*7 *3 filters in the first Convolutional layer, 256 units 5*5*96 filters in the second Convolutional layer and finally 384 units 3*3*256 filters in the third layer. It seen from the previous test that increasing the depth of the CNN slightly increases the success rate and in this test effect of increasing the filter sizes are investigated. Instead of 7*7,5*5 and 3*3 filters 9*9,7*7 and 5*5 filters are used and the result is examined. 5 Test is done over first 5 sets of 10 sets of data. The average success rate is shown in Table 5.9 and false alarm rate is shown in Table 5.10

Table 5-9 *Success rate of 3 Conv CNN with different filters*

	Test 3-Conv CNN	Test 3-Conv CNN with higher filters
Success Rate	86.00	84.33

Table 5-10. *False alarm rate of 3Conv CNN with different filters*

	Test 3-Conv CNN	Test 3-Conv CNN with higher filters
False Alarm Rate	3.00	4.00

Average and standart deviation of methods and their percentages are shown in Table 5.11

Table 5-11. *Test results of 3 Conv CNN with different filter size*

		Test 3-Conv CNN with higher filters	Test 3 Conv CNN
	True Positives	70.83	72.50
	True Negatives	93.33	95.00
	False Positives	6.67	5.00
%Average	False Negatives	29.17	27.50
		Test 3-Conv CNN with higher filters	Test 3 Conv CNN
	True Positives	17	17.4
	True Negatives	33.6	34.2
	False Positives	2.4	1.8
Average	False Negatives	7	6.6
		Test 3-Conv CNN with higher filters	Test 3 Conv CNN
	True Positives	3.54	4.28
	True Negatives	1.14	1.30
	False Positives	1.14	1.30
St. Dev	False Negatives	3.54	4.28

		Test 3-Conv CNN with higher filters	Test 3-Conv CNN with higher filters
	True Positives	14.73	17.82
	True Negatives	3.17	3.62
	False Positives	3.17	3.62
%St.Dev	False Negatives	14.73	17.82

As can be seen from the table, success rate does not change much with the filter size. Actually with bigger filters false alarm rate increases from 4% to 3% and average success rate drops from 86% to 84.33%.

5.3.7. Comparison of 3 Conv and 1 Conv CNNs

Analyzing the results obtained in 5.3.5 it is seen that increasing the depth of the network do not have a major effect on success rate. To address the minimum necessary depth of the network required to distinguish potholes from other anomalies, a one convolutional layer CNN (1 Conv CNN) is constructed as described in chapter 4. The main goal is to see whether 1 convolutional layer provides satisfying performance comparing to 3 Conv CNN. 5 Test is done over 5 new sets of data. The average success rate is shown in Table 5.12 and false alarm rate is shown in Table 5.13

Table 5-12. *Success rate of 3 Conv and 1 Conv CNN*

	Test 1-Conv CNN	Test 3-Conv CNN
Success Rate	84.33	86.00

Table 5-13. *False alarm rate of 3 Conv and 1 Conv CNN*

	Test 1-Conv CNN	Test 3-Conv CNN
False Alarm Rate	5.33	4.00

Average values of the methods are shown in Table 5.14

Table 5-14. *Test results of 3 Conv and 1 Conv CNN*

		Test 1 Conv CNN	Test 3 Conv CNN
Average	True Positives	17.6	18
	True Negatives	33	33.6
	False Positives	3	2.4
	False Negatives	6.4	6

It is seen from tables 5.12 through 5.14 that 1 convolutional layer network produces nearly as much successful results as 3 Conv CNN. It can be concluded from the above results that adaptive feature set is adequate for differentiating the asphalt anomalies. Major differentiable features lie in the low-level feature sets. Pothole, manhole, dirt and patches do not have differentiable high-level features

5.3.8. Test with Internet and Camera Images

As stated in data preparation section, pothole image sets used in tests consist of 12 pothole images downloaded from internet and 12 pothole images taken with cellular phone camera. The 10 test conducted in 5.3.4 with 3 Conv CNN is repeated with internet only and camera only pothole images. The average true positives (out of 12) for each set is shown in Table 5.15.

Table 5-15 *Comparison of internet and camera pothole image success*

	Test Internet Images	Test Camera Images
Avg. True Positives (out of 12)	10.7	7.3

As seen from the above table, the system produces much more successful results with the internet images. Comparing the images shown in Figure 5.8 and 5.9 it can be concluded that the major difference that can cause this result is the anomaly region size to whole image ratio.

Figure 5.8 Camera pothole image

Figure 5.9 Internet pothole image

5.3.9. Tests with Cropped Images

The tests conducted in 5.3.8 shows that as the anomaly region size ratio according to whole image increases so does the success rate. As the aspect ratio of the anomaly region increases the network do not have to learn every part of the image. As stated

earlier the anomaly detection is a solved method by either using the method in [35] or the saliency methods which are used in [5xx]. Assuming that anomaly region is already detected, we can crop this anomaly region and feed the network with this cropped image. Hence the anomaly part of the images (with a large bounding box) are cropped and 5 tests is done over 5 new sets of data. The average success rate of non-cropped and cropped image sets is shown in Table 5.16 and false alarm rate is shown in Table 5.17

Table 5-16. *Success rate of cropped and non-cropped image set*

	Test Cropped Images	Test Non-cropped Images
Success Rate	92.33	86.00

Table 5-17. *False alarm rate of cropped and non-cropped image set*

	Test Cropped Images	Test Non-cropped Images
False Alarm Rate	2.00	3.66

Analyzing the results listed in above tables, it is seen that in line with the test results in 5.3.2, the success rate of the cropped-image sets is quite high comparing to the non-cropped image sets. False alarm also drops significantly with the cropped images.

5.3.10. Tests with Different Drop-out Rates

As mentioned before, to overcome over fitting problem a technique called “dropout” is implemented in [53]. When dropout is implemented a neuron is updated at each

step with a probability of p, otherwise it is ignored. In 3 Conv CNN and 4 Conv CNN p is taken as 0.5 that is only half of the neurons are updated at each step.

In terms of robustness and success rates, the effect of dropout rate is explored and tested. Dropout rate of various rates are planned to be tested and compared. Hence the tests are done with dropout rate of 0.1, 0.25, 0.5 and 0.75. 3 Conv CNN is used, and the tests are done on the first 5 set of the 10-test set. The results are summarized in table 5.18 and 5.19. The number of epochs against error rate with different drop-out rates are shown in Figure 5.10 through 5.13

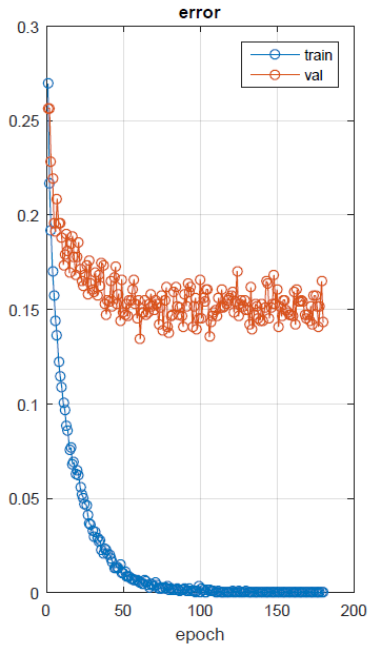


Figure 5.10. Number of epochs vs error rate with 0.10 dropout

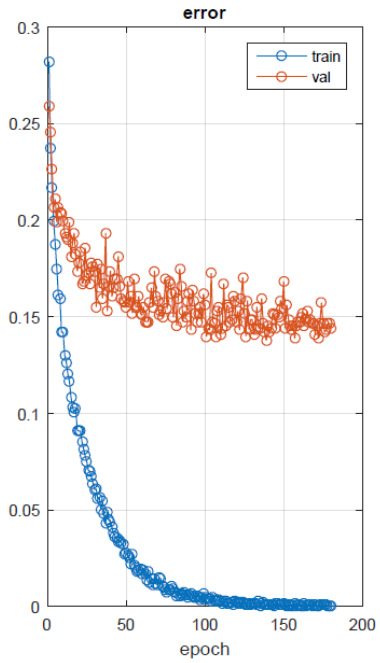


Figure 5.11 Number of epochs vs error rate with 0.25 dropout

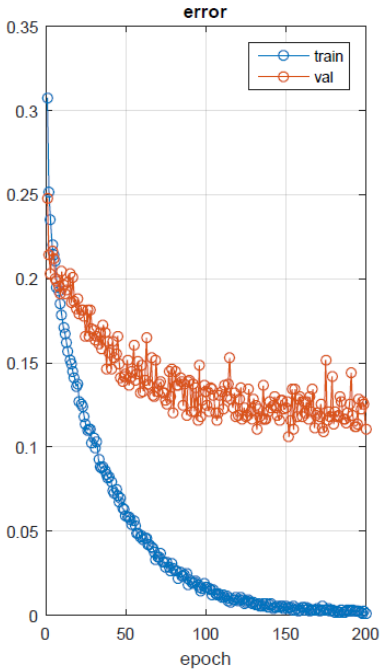


Figure 5.12 Number of epochs vs error rate with 0.50 dropout

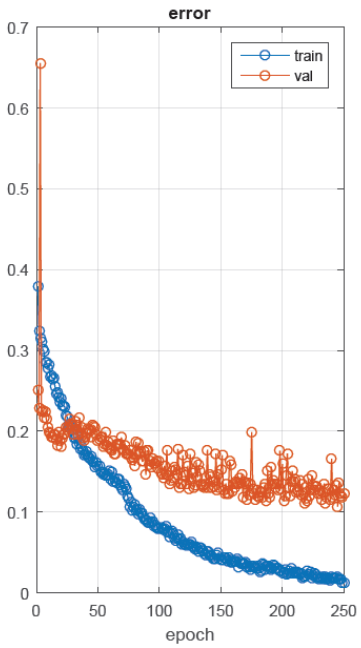


Figure 5.13 Number of epochs vs error rate with 0.75 dropout

Table 5-18. *Success rate of 3 Conv CNN with different dropouts*

	Dropout with 0.1	Dropout with 0.25	Dropout with 0.5	Dropout with 0.75
Success Rate	87.00	86.67	86.00	86.33

Table 5-19. *False alarm rate of 3 Conv CNN with different dropouts*

	Dropout with 0.1	Dropout with 0.25	Dropout with 0.5	Dropout with 0.75
False Alarm Rate	3.00	4.00	3.66	3.66

Exploring the figures 5.10 through 5.13, it is seen that with increasing dropout rate validation success is increased and it prevented overfitting. On the other hand, analyzing table 5.18 and 5.19 it is seen that changing the dropout rate does not have a huge effect on the system performance. Considering the test results obtained in section 5.3.7 and 5.3.8 together with the dropout test results, it can be concluded that

the network does not need to have 3 Convolutional layer and higher layers. 1 convolutional layer may be enough to differentiate pothole among other anomalies. Hence, a new dropout test is conducted with 1 Conv CNN but reducing the fully connected layer from 512 neurons to 256 neurons. The tests are repeated with dropout rate of 0.1, 0.5 and 0.75 over 5 datasets and the results are shown in table 5.20 and 5.21

Table 5-20. *Success rate of 1 Conv CNN with different dropouts*

	Dropout with 0.1	Dropout with 0.5	Dropout with 0.75
Success Rate	85.00	87.66	87.66

Table 5-21. *False alarm rate of 1 Conv CNN with different dropouts*

	Dropout with 0.1	Dropout with 0.5	Dropout with 0.75
False Alarm Rate	6.66	3.66	3.66

The test conducted with 1 Conv CNN shows that increasing the dropout rate improves the system performance up to a point and after that point system performance become stable. Increasing the dropout rate higher than 0.75 ratio may cause performance degradation.

CHAPTER 6

CONCLUSION

6.1. SUMMARY

In this study a 3 layer convolutional neural network is applied to classify images that contain potholes. As stated earlier, anomaly detection in asphalt road images is a nearly solved problem. The main problem is to determine whether this anomaly is a pothole or not. Many images that contain pothole, dirt, patching or manhole is collected through the roads of Ankara. Examining these images, it can be seen that it is not easy to differentiate these images through classical feature extraction and classification methods without the depth knowledge. The main reason is that images that contain potholes do not have a clear specific pattern over images that contain non pothole anomalies i.e. they all show similar low-level features.

Convolutional neural networks on the other hand provides us to obtain dynamic features extraction. It resembles the object detection structure in humans. It has a layered structure (Conv layers) to obtain adaptive and complex features. Having obtained these complex features the success rate of pothole classification increases significantly comparing to basic feature extraction and classification methods.

Seeing that convolutional neural networks have strong advantage comparing to classical methods, the possibility of improving the performance and the robustness of these structures are also explored. The necessary network depth that is needed to distinguish potholes among other anomalies is explored, tested and compared. For this

purpose, a 4 layer Conv CNN and a 1 Conv CNN is presented. Internet pothole and camera pothole performance is compared and with the obtained findings the system is tested with cropped images. Performance of different filter size is also tested on the same images and compared with each other. And finally, different dropout rates used in neuron updates to prevent overfitting problem is tested and compared both in 3 Conv CNN and 1 Conv CNN to see the robustness and performance of the system.

6.2. CONCLUSIONS AND FUTURE WORK

Conclusions that can be extracted from this study and the conducted various tests are summarized as follows;

- Obtaining dynamic, adaptive and complex features with respect to classical methods, Convolutional neural networks increases the success rate significantly. As can be seen from the test results, CNN structures are at least 14 percent more successful than other classical methods.
- False alarm rate in convolutional neural networks is much lower than the classical methods. It is one of the important factors for the usability of the systems. If a system produces too much false alarm people are starting not to use those systems anymore. One of the main goals for this study is to obtain a system that has a false alarm rate lower than 5 percent and it is achieved in this study.
- To differentiate pothole among other asphalt anomalies 1 Convolutional layer is adequate. Increasing the network depth do not much affect the system performance but improves the false alarm rate a little bit. However, training time also increases significantly.

- Training and testing the system with the cropped images increases the system performance significantly The main target is to have a success rate more than 90 percent. In this study it is achieved with the cropped image set.
- Increasing the filter size in 3 layer convolutional networks drops the success rate a little bit. It seems from the test that 7*7, 5*5 and 3*3 filters are very suitable for the 3 Conv CNN structure.
- Using dropout in neuron updates prevents the system from overfitting. Tests on different dropout rates over 3 Conv CNN shows that dropout rate does not affect the system performance due to over depth of the network. However, test with 1 Conv CNN shows that increasing the dropout rate increases the system performance up to a certain dropout rate. It seems that dropout rate of p=0.5 is suitable for training the system.

Although the above findings were obtained in this study, all the necessary studies could not be completed. The success of the system shall be increased to 95% or more success rates to be much more useful while keeping the false alarm rate the same or lower. To achieve these goals below studies can be conducted;

- 546 non-pothole images (190 Dirt, 177 manhole, 165 patching) all taken with Samsung s3 mini and 474 pothole images (179 taken with camera 295 downloaded from the Internet) are used in this study. The images are multiplied by rotating in their y axis and used in trainings. Number of pothole and non-pothole examples shall be increased to improve the success rate of the system.
- The effect of anomaly size ratio to whole image size can be explored and investigated. In this study, only the positive affect over success rate of cropping the anomaly region instead of entire image is shown. The anomaly regions are cropped with a broader boundary. Better boundaries hence better

cropped images can be obtained and the system performance can be measured with these cropped images.

- Different filters and number of filters can be constructed with 1 Conv CNN and the performance of each structure can be explored.
- Different type of convolutional neural network layers and structures can be conducted and tested to achieve the highest success rate. Training the convolutional neural networks requires a lot of time. With ten sets of test data, it is very time consuming and could not be completed in this study.

REFERENCES

1. Amaral T., Silva L.M., Alexandre L.A., Kandaswamy C., Santos J. M., Marques de Sa J., "Using Different Cost Functions to Train Stacked Auto-Encoders", Artificial Intelligence (MICAI) 2013 12th Mexican International Conference on, pp. 114-120, 2013.

2. Astarita V., Festa D.C. , Mongelli D.W.E. , Tassitani A., "New methodology for the identification of road surface anomalies" Service Operations and Logistics, and Informatics (SOLI), 2014 IEEE International Conference on

3. Aydoğdu M.F., Çelik V., Demirci M.F., "Comparison of Three Different CNN Architectures for Age Classification", Semantic Computing (ICSC2017), 2017

4. Azhar K., Mirtaza F., Yousaf M.H., and Habib H.A., "Computer Vision Based Detection and Localization of Potholes in Asphalt Pavement Images" , 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)15-18 May 2016

5. Backpropagation [online] Available at http://www.wiki-zero.com/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvQmFja3Byb3BhZ2F0aW9u [Accessed 16 April 2018]

6. Buza E., Omanovic S., Huseinovic A., "Pothole detection with image processing and spectral clustering", Recent Advances in Computer Science and Networking pp. 1-6

7. Chang K.T., Chang J. R., and Liu J. K., "Detection of pavement distress using 3D laser scanning technology", In Proceedings of the ASCE International Conference on Computing in Civil Engineering (2005), 1-11.

8. Chen K., Lu M., Fan X., Wei M., Wu J., "Road Condition Monitoring Using On-board Three axis Accelerometer and GPS Sensor," 6th International ICST

Conference on Communications and Networking in China (CHINACOM)(2011)

9. Convolutional neural networks [online] Available at http://cs231n.github.io/convolutional-networks/ [Accessed 20 April 2018]

10. Dalal N. and Triggs B., “Histograms of Oriented Gradients for Human Detection”. In CVPR, pages 886-893, 2005.

11. Deep learning [online] Available at http://neuralnetworksanddeeplearning.com/chap6.html [Accessed 19 April 2018]

12. Duchi J., Hazan E., Singer Y.. “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization”, The Journal of Machine Learning Research, pp.2121-2159,2011

13. Eriksson J., Girod L., Hull B., Newton R., Madden S., and Balakrishnan H., “Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring,” proceeding of the 6th international conference on mobile systems, applications, and services, 2008.

14. Freund Y., Schapire R.E., “A decision-theoretic generalization of on-line learning and an application to boosting.” Journal of Computer and System Sciences 55(1), 119–139 ,1997

15. Girshick R., Donahue J., Darrell T., and Malik J., “Rich feature hierarchies for accurate object detection and semantic segmentation,”in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

16. Gradient [online] Available at http://www.wiki-zero.com/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvR3JhZGllbnQ [Accessed 17 April 2018]

17. Hatipoglu P., Aytekin Ö., Ulusoy İ., Halici U., “Saliency Analysis for High Resolution Satellite Images with Challenging Contents”, ICT Innovations,

Ohrid, 2014 Web Proceedings ISSN 1857-7288 -Springer Advances in Intelligent Systems and Computing, Vol. 311, pp 97-106

18. Histogram of Oriented Gradients [online] Available at https://www.learnopencv.com/histogram-of-oriented-gradients/ [Accessed 18 Nov. 2017]

19. Hou Z., Wang K.C.P., and Gong W.," Experimentation of 3D Pavement Imaging Through Stereovision" ,International Conference on Transportation Engineering 2007 (ICTE 2007)

20. Hubel & Wiesel [online] Available at http://cns-alumni.bu.edu/~slehar/webstuff/pcave/hubel.html [Accessed 19 April 2018]

21. Hubel D.H., Wiesel T.N., "Receptive fields of single neurones in the cat's striate cortex." J Physiol (Lond) 1959;148:574–59

22. Hubel D.H., Wiesel T.N., "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex." J Physiol (Lond) 1962;160:106–154

23. Jang D., Park R., "Pothole detection using spatio-temporal saliency", IET Intell. Transp. Syst., 2016, Vol. 10, Iss. 9, pp. 605–612

24. Jang J., Smyth A.W., Yang Y., Cavalcanti D., " Road Surface Monitoring via Multiple Sensor –Equipped Vehicles," IEEE Infocom Poster Presentation (2015)

25. Jog G.M., Koch C., Golparvar-Fards M., Brilakis I., "Pothole Properties Measurement Through Visual 2D Recognition and 3D Reconstruction", Computing in Civil Engineering (2012) pp. 553-560

26. Karayolu Trafik Kaza İstatistikleri, 2016 [online] Available at http://www.tuik.gov.tr/PreHaberBultenleri.do?id=24606 [Accessed 12 Sept. 2017]

27. Koch C., and Brilakis I., “Pothole detection in asphalt pavement images”, Advanced Engineering Informatics, Vol. 25 (2011), 507-515.

28. Krizhevsky A., Sutskever I., Hinton G.E., “Imagenet classification with deep convolutional neural networks” 25th International Conference on Neural Information Processing Systems , 1097–1105 ,2012

29. LeCun Y., Bottou L., Bengio Y., and Haffner P., “Gradient-based learning applied to document recognition.”Proceedings of the IEEE, november 1998

30. LeNet-5, convolutional neural networks [online] Available at http://yann.lecun.com/exdb/lenet/ [Accessed 19 April 2018]

31. Levi G. and Hassner T., “Age and gender classification using convolutional neural networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2015, pp. 34–42.

32. Lin J., Liu Y., “Potholes Detection Based on SVM in the Pavement Distress Image”, IEEE 2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science, pp544-547

33. Linear Classifiers [online] Available at http://cs231n.github.io/linear-classify/#softmax [Accessed 24 April 2018]

34. Lokeshwor H., Das L. K., and Sud S. K., “Method for automated assessment of potholes, cracks and patches from road surface video clips”, Procedia – Social and Behavioral Sciences, Vol. 104 (2013), 312-321.

35. Lokeshwor H., Das L.K., Goel S., Robust method for Automated Segmentation of Frames with/without Distress from Road Surface Video Clips, Journal of Transportation Engineering, Vol. 140, No. 1, January 2014, pp. 31-41

36. Madli R., Hebbar S., Pattar P., and Golla V., “Automatic Detection and Notification of Potholes and Humps on Roads to Aid Drivers”, IEEE SENSORS JOURNAL, VOL. 15, NO. 8, AUGUST 2015

37. Marques de Sá J., Silva L., Santos J., and Alexandre L.. “Minimum Error Entropy Classification”, volume 420 of Studies in Computational Intelligence. Springer, 2013

38. Mednis A., Strazdins G., Zviedris R., Kanonirs G., Selavo L., “Real Time Pothole Detection using Android Smartphones with Accelerometers”, Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on pp 1-6

39. Moazzam I., Kamal K., Methavan S., Usman S., Rahman M., “Metrology and Visualization of Potholes using the Microsoft Kinect Sensor”, Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013) pp. 1284-1291

40. Neural Networks [online] Available at http://cs231n.github.io/neural-networks-1/#intro [Accessed 16 April 2018]

41. Neural Networks [online] Available at http://cs231n.github.io/neural-networks-3/#intro [Accessed 16 April 2018]

42. Neural Networks [online] Available at http://cs231n.github.io/neural-networks-2/#reg [Accessed 16 April 2018]

43. Neural Networks and Deep Learning [online] Available at http://neuralnetworksanddeeplearning.com/chap1.html [Accessed 16 April 2018]

44. Neuron [online] Available at http://www.wiki-zero.com/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvTmV1cm9u [Accessed 16 April 2018]

45. Neurons [online] Available at https://online.science.psu.edu/bisc004_activeup002/node/5397 [Accessed 16 April 2018]

46. Nienaber S., Booysen M., and Kroon R.,“Detecting Potholes Using Simple Image Processing Techniques and Real-World Footage”, 34th Annual Southern African Transport Conference SATC 2015

47. Over fitting [online] Available at http://www.wiki-zero.com/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvT3ZlcmZpdHRpbmc [Accessed 16 April 2018]

48. Ren S., He K., Girshick R., and Sun J., "Faster r-cnn: Towards real-time object detection with region proposal networks," in Advances in neural information processing systems, 2015, pp. 91–99.

49. Schiopu I., Saarinen J. P., Kettunen L., and Tabus I., "Pothole Detection and Tracking in Car Video Sequence", Telecommunications and Signal Processing (TSP), 2016 39th International Conference on

50. Sermanet P., Eigen D., Zhang X., Mathieu M., Fergus R., and Le-Cun Y., "Overfeat: Integrated recognition, localization and detection using convolutional networks," arXiv preprint arXiv:1312.6229, 2013.

51. Shen T., Schamp G., Haddad M., "*Stereo Vision Based Road Surface Preview",* 2014 IEEE 17th International Conference on Intelligent Transportation System, (Qingdao,China)

52. Softmax Function [online] Available at http://www.wiki-zero.com/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvU29mdG1heF9mdW5jdGlvbiNjaXRlX25vdGUtYmlzaG9wLTE [Accessed 24 April 2018]

53. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., "Dropout: a simple way to prevent neural networks from overfitting", The Journal of Machine Learning Research, v.15 n.1, p.1929-1958, January 2014

54. Support Vector Machines for Binary Classification [online] Available at https://www.learnopencv.com/histogram-of-oriented-gradients/ [Accessed 18 Nov. 2017]

55. Viola P., Jones M.J., " Rapid Object Detection using a Boosted Cascade of Simple Features", In CVPR 2001 ,8-14 Dec 2001

56. Viola P., Jones M.J., "Robust Real-Time Face Detection", International Journal of Computer Vision 57(2), 137–154, 2004

57. Zeiler M.D. and Fergus R., “Visualizing and understanding convolutional networks,” in European Conference on Computer Vision. Springer, 2014, pp. 818–833.

58. Zhang X., Ai X. Chan C.K and Dahnoun N., "An efficient algorithm for pothole detection using stereo vision" *IEEE International* Conference *on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2014. Florence. pp. 564-568

59. Zinal K, & Monali R,. “ A Review: Object Detection using Deep Learning”. International Journal of Computer Applications, 2018, 180. 46-48. 10.5120/ijca2018916708.