ITHEA

International Journal INFORMATION MODELS & ANALYSES

2019 Volume 8 Number 1

International Journal
INFORMATION MODELS AND ANALYSES
Volume 8 / 2019, Number 1

IJ IMA is official publisher of the scientific papers of the members of the ITHEA® International Scientific Society

IJ IMA rules for preparing the manuscripts are compulsory.
The **rules for the papers** for ITHEA International Journals are given on www.ithea.org .
The camera-ready copy of the paper should be received by ITHEA® Submission system http://ij.ithea.org
.
Responsibility for papers published in IJ IMA belongs to authors.

International Journal "INFORMATION MODELS AND ANALYSES" Volume 8, Number 1, 2019

Edited by the **Institute of Information Theories and Applications FOI ITHEA**, Bulgaria, in collaboration with:
University of Telecommunications and Posts, Bulgaria,
V.M.Glushkov Institute of Cybernetics of NAS, Ukraine,
Universidad Politécnica de Madrid, Spain,
Hasselt University, Belgium,
University of Perugia, Italy,
Institute for Informatics and Automation Problems, NAS of the Republic of Armenia
St. Petersburg Institute of Informatics, RAS, Russia,

Publisher: **ITHEA®** Sofia, 1000, P.O.B. 775, Bulgaria. www.ithea.org , e-mail: office@ithea.org
Technical editor: **Ina Markova**

Printed in Bulgaria

ISSN 1314-6416 (printed) **ISSN 1314-6432 (Online)**

ANALYSIS OF HAND VEINS AND MULTIMODAL PATTERN RECOGNITION ALGORITHMS AND SYSTEMS

Mona A. Ahmed, Abdel-Badeeh M. Salem

***Abstract**: Recently, many researchers investigated the palm, hand, and finger vein recognition for automated personal authentication. The aim of this paper is to review the ideas published earlier and presents an analysis of palm, hand, and finger vein pattern recognition and the fusion of modalities. This model is used to improve the accuracy and response time of vein pattern authentication. The steps of pattern vein system are extensively analyzed. The main result of this paper is the comparison of the results of various researches preprocessing, feature extraction and matching algorithms.*

***Keywords:** Biometric, vein pattern, multimodal, feature extraction, matching, fusion level, intelligent algorithms, soft computing.*

Introduction

Identity verification (authentication) in computer systems has been traditionally based on something that one has (key, magnetic or chip card) or one knows (PIN, password). Things like keys or cards, however, tend to get stolen or lost and passwords are often forgotten or disclosed. To achieve more reliable verification or identification we should use something that really characterizes the given person [Marcos, 2006].Biometrics is automated methods of recognizing a person based on a physiological or behavioral characteristic. An example of or behavioral characteristic are face recognition, fingerprints, hand geometry, signature verification, iris, retinal, finger/hand/palm vein recognition, ear recognition, and voice recognition. In biometric applications, a relatively new technology is emerging, namely the optical scanning of superficial vein patterns. In order to be viable, a biometric parameter has to be easily identifiable but hidden from view so that it cannot be reproduced or simulated. It can be observed that the veins of the human body do not leave external marks like fingerprints, are not easily falsifiable like the voice, cannot be disguised like face traits, and are extremely hard to covertly extract during and after the lifetime of an individual in order to be reused by an impostor. In the same time, the technology used to acquire the vein pattern has reduced costs and is not invasive, requires minimal cooperation from a person, and is largely a noncontact procedure that allows it to be used where hygienic concerns are an issue [Crisan, 2008].

A biometric recognition system consists of the following steps: Image acquisition from the database and pre-processing, finding of region of interest, extraction of pattern features and recognition as showed in figure 1. When a user's hand is held over a scanner, a near-infrared light maps the location of the veins. The red blood cells present in the veins absorb the rays and show up on the map as black lines, whereas the remaining hand structure shows up as white[Sarkar,2010]. After image capture, a small area of a hand image is located as the region of interest (ROI) to extract the features and to compare different hands. Using the features within ROI for recognition can improve the computation efficiency significantly [Soliman,2012].In the image-based biometric systems there is a number of processing tasks used to produce a better quality of image that will be used on the later stage as an input image and assuring that relevant information can be detected. Normally, the captured vein pattern is grayscale and subject to noise. Noise reduction and contrast enhancement are crucial to ensure the quality of the subsequent steps of feature extraction [Soliman, 2012].Also, the vein pattern extracted from infrared-ray images is represented as dark lines. To extract these lines many researcher used edge detection and morphological operators [luo, 2010].Feature extraction plays an important role in pattern vein recognition because the performance of feature matching is greatly influenced by its output. Matching is the key procedure of deciding whether a particular vein pattern image is genuine or imposter using a set of appropriate matching algorithms. In vein pattern authentication it is a one to one matching process where as in vein pattern identification it is one too many. In effect feature matching is nothing but a task of similarity computation [Ezhilmaran, 2015].

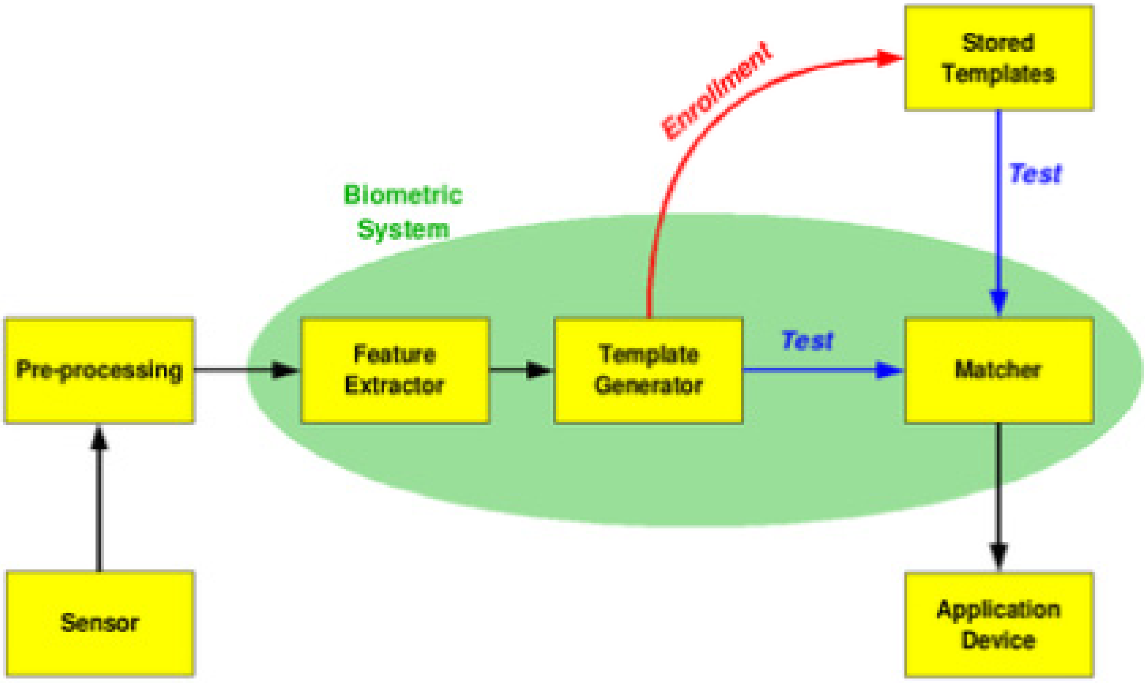


Figure 1.The basic block diagram of a biometric system

For some security systems, one method of authentication is not enough. Any single modal biometric has limitations. For example, iris recognition suffers from some problems like camera distance, eyelids and eyelashes occlusion, lenses, and reflections [Gad, 2011]. Face changes overages and unstable, and twins may have similar face features. Also, fake faces from mobiles as example, and masks used to attack the system. Fake fingers made from gelatin and/or silicon has ability to attack the fingerprint-based recognition system. Cold leads to voice problems and tape recordings may be used to hack the system [Gad, 2015]. The DNA needs several hours to be obtained. Besides, DNA includes sensitive information related to genetic of individuals and the test is expensive to perform. Hand geometry is not distinctive enough to be applied to a large population. Thus, it is not suitable for purpose of identification. Gait is sensitive to body weight and not stable; it is not used for large population and not reliable enough. Signature is not universal and changes with time. Offline ones are forgery while, Online signature cannot applied for documents verification. None of above traits alone can ensure perfect recognition performance.

Nevertheless, the biometric system can also be attacked by the outsider or unauthorized person at various points [Bhatia, 2013]. Combining multiple modalities is a good idea to decrease these conditions.

Palm Veins Pattern Techniques

Palm vein recognition is a part of physiological biometric uses blood vessel structure for the identification. The blood transfusion from body parts to heart is made by the blood vein. The blood vein carries de-oxygenated hemoglobin that can be sensed under near infrared light. The pattern classification proposes a design cycle to build a system for recognition [Shriram, 2015].Table 1 provides a detailed analysis of the existing feature extraction techniques and image enhancement algorithms for palm vein recognition.

Table 1 analysis of palm veins recognition techniques

Author(s)	Method of preprocessing	Method of feature extraction	Method of matching	Database size	Accuracy
Sasikala et al. [2016]	DWT	Sobal filter	-	-	-
Smorawa et al. [2015]	Histogram equalization	Proposed filters	Hamming distance Hidden Markov	800 images	99.62% & 99.76%
Raut et al. [2015]	Gabor filter	canny edge detector	Euclidean distance	100 images	-
bikoye et al [2016]	Histogram equalization & Zhang Suen	k-means	Euclidean distance	800 images	-
Gayathri et al.[2015]	WEF	canny edge detector & ACO	SVM	6000 images	98%
Pooja et al.[2016]	Gabor filter	LBP & PCA	ANN	500 images	99.5%
Dere et al.[2017]	Histogram equalization	canny edge detector	-	-	-
Sayed [2015]	2-D Gabor filter, Median filter & morphological	LLBP	Coset decomposition algorithm	250 images	99.8%

Author(s)	Method of preprocessing	Method of feature extraction	Method of matching	Database size	Accuracy
	operations				
Villariña et al.[2015]	Gamma correction & LRE	Sobel Directional Coding	BPNN	400 images	98%
Singh et al.[2015]	Gabor filter	Laplacian	k- nearest neighbor classifier	10 images	95.6%
Bayoumi et al.[2015]	Morphological operations	PCA	Euclidean distance	25 images	70%
Ali et al[2017]	Median filter	DoG	degree of similarity	6000 images	99.67%
Wang et al.[2014]	Retinex method &	-	-	-	-
Raut et al.[2018]	Gabor filter	Harris-Stephens Corner Point Detector	BPNN	250 images	95.8%

From the analysis of the above research, it can report the following important results;

a) [Sasikala, 2016] preprocessed images used sobal filter to detect edges which is shape features and historically, structural and statistical approaches have been adopted for texture feature extraction.

b) [Smorawa, 2015] studies included two ways of verification. The first one is based on comparing feature vectors using a Hamming distance. The second method takes into account the verification of identity, based on Hidden Markov Models.

c) [Shriram, 2015] used the Gabor filter to process the image matrices and extract the blood vessel structure. The computed distance is stored in the form of feature vector set. The binary vector from the set is to be extracted and further will lead to classify the sample set of one subject with another.

d) [Bikoye, 2016] proposed a technique for palm vein biometric verification enhancement and accuracy using statistical and data mining tools. The work brings to light a faster system with high efficiency for good experience when addressing security.

e) [Gayathri, 2015] proposed system to accommodate the rotational, potential deformations and translational changes by encoding the orientation conserving features. Compared to the previous methodologies.

f) [Pooja, 2016] proposed an approach for person’s authentication, which can be more reliable and achieve higher identification rate accuracy.

g) [Dere, 2017] proposed biometric system implemented for improving the security and authentication biometric system. The experimental result shows that it takes minimal time that is only 0.5 seconds to verify one input palm vein sample image. This system consumes low power and has less computational complexity.

h) [Sayed, 2015] developed an algorithm which merges the biometrics and cryptography. The model which is fit for matching user palm veins is, to the best of my knowledge, entirely new.

i) [Villariña, 2015] proposed an effective palm vein recognition system used directional coding scheme to extract the feature of the palm veins. They propose three verification neural network mechanisms to verify the palm vein images.

j) [Singh, 2015] proposed new method for human authentication, this approach performs very well even with the minimum number of enrollment images.

k) [Bayoumi, 2015] proposed a system for attendance using palm vein as a biomedical features based on Principal Component Analysis (PCA) to discriminate the variances between the image features instead of between the whole training set.

l) [Ali, 2017] introduced an approach of personal recognition system based on centerline of palm vein. The features were extracted from vein’s centerline by finding keypoints locations depending on scales of difference-of Gaussian (DoG) function. Comparing with other results of palm vein recognition algorithms, this algorithm gets the highest recognition performance.

m) [Wang, 2014] proposed a Retinex based method to reduce the bad influence caused by shadow. The experiment results show that the proposed method is robust to shadow effect and can better enhance the palm vein image.

n) [Raut, 2018] introduced an abstract way to do image processing operations to result edge extraction and detection, classification and assessment of the results by computing recognition rate. This paper put fourth scheme for classifier validation using neural pattern recognition model.

Dorsal Hand Veins Pattern Techniques

The dorsal hand vein pattern is unique biometric identity of the human beings. The dorsal hand vein recognition is a recent biometric technique which is used for authentication purposes in various applications. Different techniques used for designing the system has discussed here. Table 2 provides a detailed analysis of the existing feature extraction techniques and image enhancement algorithms for hand vein recognition.

Table 2 analysis of dorsal hand vein recognition techniques

Author(s)	Method of preprocessing	Method of feature extraction	Method of matching	Database size	Accuracy
Naidile et al.[2015]	Morphological operations	Gaussian shaped filter	Simple correlation method	-	75%
Belean et al. [2017]	Hough transform	mathematic morphology	BPNN	360 images	99.17%
Sree et al. [2016]	Histogram equalization	BPNN	SVM	-	99.76%
Premavathi et al.[2018]	Morphological operations	CB IN TP	K-nearest neighbor	2040 images	95.10%
Rajalakshmi et al.[2015]	Median filter& Morphological operations	CNN	Logistic Regression	403 images	96.77%
Raghavendra et al. [2018]	Histogram equalization	HoG-SRC, LBP-SRC, LG-SRC and LPQ-SRC	SRC	100 images	99.3%

From the analysis of the above research, it can report the following important results;

(a) [Naidile, 2015] proposed a system using the infrared camera to acquire the hand image. Experimental results show that this method can be successfully used for identification.

(b) [Belean, 2017] proposed a novel user authentication approach based on dorsal hand vein pattern analysis and multi-layer perceptron neural network. For image processing two different techniques are employed: rotation invariant Hough transform and clustering based segmentation and mathematic morphology. Both approaches lead to binary images containing the vein patterns.

(c) [Sree, 2016] proposed a new approach for personal verification, explained the function effectively locates the hand vein speedily by using edge detection. SVM approach avoids the challenge of dealing with the change of intensities for different NIR image slices.

(d) [Premavathi, 2018] proposed feature descriptor and classification method for an efficient recognition system. A new minimum distance classification is proposed with dataset and the results are checked for accuracy and reliability.

(e) [Rajalakshmi, 2015] proposed work combined multiple models together to form an ensemble, with minimal use of resources and computational power while ensuring appreciable scalability.

(f) [Raghavendra, 2018] presented a new dorsal hand vein sensor that can capture a good quality of hand vein images based on a NIR illumination that can emit light in a spectrum of 940nm.The sensor proposed use pegs allow users to directly capture ROI rather than whole dorsal hand vein image.

Finger Veins Pattern Techniques

Recently finger vein biometric has attracted increasing interest from many researchers and thus considerable development is seen in the past decade. Hitachi Ltd. of Japan has been exploring the finger vein technology since 1997 and was the first to commercialize the finger vein authentication into product form which was released in 2002. Hitachi developed ATM applications in 2004 and commercialized them in 2005 [Charaya, 2018].Table 3 provides a detailed analysis of the existing feature extraction techniques and image enhancement algorithms for finger vein recognition.

Table 3 analysis of finger vein recognition techniques

Author(s)	Method of preprocessing	Method of feature extraction	Method of matching	Database size	Accuracy
Mulyono et al. [2008]	Median filter	CCL	Rm threshold	1000 images	100%
Pham et al.[2015]	Gabor filter	LBP	HD	1200 images 1980 images 3816 images	96.35%
Meng [2018]	Morphological operations	SIFT	proposed method	1872 images 3816 images	99.68% 99.70%
Ayappan et al. [2017]	median filter, Gaussian low pass filter& Morphological operations	CN	MHD	150 images	100%
Ali et al. [2017]	DWT	GLCM	Euclidean distance	3816 images	92.4%
Bader et al. [2018]	Median filter & Zhang-thinning	FAST &Harris	Manhattan distance	636 images	99.71%
Janney et al. [2017]	Median filter	DWT	Manhattan distance	-	-
Qin et al. [2013]	Morphological operations	SIFT	SVM	4260 images 7120 images	95.04%
Das et al. [2019]	histogram equalization & Morphological operations	Gabor filters	CNN	3132 images 5904 images 3816 images 1440 images	95%
Kaur et al. [2015]	Repeated Line tracking	Gabor filters	SVM	-	-

From the analysis of the above research, it can report the following important results;

a) [Mulyono, 2008] introduced preliminary process to enhance the image quality worsened by light effect and noise produced by the web camera.

b) [Pham, 2015] proposed a finger-vein recognition system using a near infrared (NIR) image sensor. The experimental results obtained with three databases showed that their system can be operated in real applications with high accuracy.

c) [Meng, 2018] proposed technique based on that regular deformation, which corresponds to posture change, can only exist in genuine vein patterns.

d) [Ayappan, 2017] proposed new algorithm with high performance and optimum accuracy. The proposed image preprocessing method comprises of 8 sub-blocks. Seen from the experimental result the method is very effective as a biometric personal identification system.

e) [Ali, 2017] presented a robust method for finger vein recognition based on the discrete wavelet transform. The simulation results show that this method is robust and fast for feature extraction and classification.

f) [Bader, 2018] used computer vision algorithms (FAST and Harris) proved that use of two algorithms together produce a reliable system of finger vein identification.

g) [Janney, 2017] proposed method aims to implement the secured patient database in hospitals using a finger vein pattern.

h) [Qin, 2013] proposed a new approach which can extract two different types of finger-vein features and achieves a most promising performance. The experimental results suggest the superiority of the proposed scheme.

i) [Das, 2019] propose a convolution-neural-network-based finger vein identification system and investigate the capabilities of the designed network over four publicly-available databases. The main purpose of this work is to propose a deep-learning method for finger-vein identification, able to achieve stable and highly accurate performance when dealing with finger-vein images of different quality.

j) [Kaur, 2015] proposed a newly developed method. An enhanced Human Identification algorithm is developed using finger vein which based on Automatic Trimap Generation, Repeated Line Tracking, Gabor and SVM. This algorithm is fast and more accurate with respect to other finger vein identification technique and also takes less time as comparison to other technique.

Multimodal Patterns Techniques

Biometric is an automated authentication technique for identifying or verifying an individual based on one's physiological or behavioral characteristics [Down, 2012]. Biometric systems can be classified into two types namely, unimodal and multi-modal systems. A unimodal biometric system is one in which, only a single type of the constituent components is present, whereas in multi-modal system more than one type of the component is present.

[Ross, 2007] establishes six advantages of a multi-modal system. Multiple modalities address the issues of non-universality encountered by unimodal systems. For example, a person who has lost his hands cannot be authenticated by a fingerprint authentication system. Unlike the process of detecting any objects, detecting human face poses many challenges due to the dynamics of skin color and facial expression. The illumination conditions, occlusion, background structure and camera positions add complexities on to the existing challenges. So the system needs multiple sensors to acquire multimodal information to authenticate a person. Multi-biometric systems helps in reducing false match and false non-match errors compared to a single biometric device.

In multimodal biometric recognition systems, the fusion of various traits can be carried out at different levels: Sensor Level Fusion, Feature Level Fusion, Score Level Fusion and Decision Level Fusion. The first two approaches are known as pre-matching fusion whereas the last two approaches are known as post-matching fusion. In sensor level fusion, the raw data acquired from either the samples of the same modality with compatible sensors or multiple instances of the sample modality and the same sensor are fused together. In feature level fusion, the extracted features are fused together. When the feature sets are homogeneous, a single resultant feature set can be calculated as a weighted average of the individual feature sets. When the feature sets are non-homogeneous, we can concatenate them to form a single feature set. In score level fusion, the output matching scores of different biometric matchers are fused together to produce a final fused score. The decision is made using the fused score. Examples of score level fusion are weighted sum, weighted product or post-classifier approaches. In decision level fusion, the matching score of each biometric system is converted into a hard decision by comparing it with the threshold tuned for that matcher. The output decisions are then fused together to make the final decision. Examples of decision level fusion are majority vote, Borda count, Behavioral Knowledge Space, Bayes fusion or the AND and OR rule [Merati, 2011].

Table 4 Analysis of multimodal recognition techniques

Author(s)	**Method of preprocessing**	**Method of feature extraction**	**Method of matching**	**Fusion levels**	**Database size**	**Accuracy**
Bharathi et al. [2016]	palm vein + dorsal hand vein	Proposed filter	Euclidean distance	Feature level	2400 images 250 images	high accuracy
Stefani et al. [2016]	Face + iris	LBP Daughman’s	LBPH Hamming Distance	decision level	400 images 200 images	77%
Shruthi et al. [2013]	finger vein + fingerprint	Gabor filters	Hamming distance	decision level	6264 images	98.78%
Christo et al. [2017]	Hand geometry + palm veins	proposed algorithm HOG	SVM	Feature level	7.200 images	98.7%
Raghavendra et al. [2014]	Finger Vein + Fingerprint	SMR	weighted sum rule	decision level	1500 images	99.22%
Trabelsi et al. [2013]	Finger Vein + Hand Vein	MLBP	IGMF	decision level	3916 images 4846 images	98%
Dhameliy et al. [2013]	Palmprint and Fingerprint	Gabor filters	Euclidean-distance	Feature level	250 images	87%
Siddharth et al. [2017]	Palm Print + Palm Vein	Gray scale Gabor filter	Euclidean-distance	Feature level	28 images	100%
Park et al. [2013]	Hand geometry + Hand Vein	median filter	Euclidean-distance	decision level	300 images	99.94%
Author(s)	Method of preprocessing	Method of feature extraction	Method of matching	Fusion levels	Database size	Accuracy
Usharani et al. [2014]	Palm Print + Palm Vein	Wavelet packet tree	K NN Naive Bayes	Feature level	2400 images	95.95%
Faris et al. [2014]	Finger Vein + Iris	Current tracking point	Hamming Distance Matcher	decision level	120 images 140 images	92.40%

From the analysis of the above research, it can report the following important results;

a) [Bharathi, 2016] developed hand vein-based multi-modal biometric technique using palm and dorsal hand vein images as two modalities for biometric recognition. Based on the threshold value, decided whether the image of the corresponding person is presented in the database or not.
b) [Stefani, 2016] designed, implemented, and deployment a multi-modal biometric system to grant access to a company structure and to internal zones in the company itself.
c) [Shruthi,2013] employed a new approach the system simultaneously acquires the finger vein and low resolution fingerprint images and combines these two evidences using a two new score level combination strategy.
d) [Christo,2017] developed a multibiometric system using hand geometry and palm veins .For the hand geometry data, an algorithm for determining finger tips and hand valleys was proposed and from there was possible to extract a handful of other features related to the geometry of the hand. The palm veins features were extracted from a rectangle generated based on the hand's center of mass. The fusion was done on feature level.
e) [Raghavendra, 2014] presented a robust imaging device that can capture both fingerprint and finger vein simultaneously. The presented low-cost sensor employs a single camera followed by both near infrared and visible light sources organized along with the physical structures to capture good quality finger vein and fingerprint samples.
f) [Trabelsi, 2013] proposed a new multimodal biometric system based on fusion of both hand vein and finger vein modalities. Experimental results confirm that the proposed multimodal biometric process achieves excellent recognition performance compared to unimodal biometric system.
g) [Dhameliya, 2013] developed Multimodal biometric identification system based on palm print and fingerprint trait. Typically in a multimodal biometric system each biometric trait processes its information independently. The processed information is combined using an appropriate fusion scheme. The experimental results demonstrated that the proposed multimodal improving system accuracy and reliability.

h) [Siddharth, 2017] proposed biometric system uses two modalities, palm print and palm vein. The encrypted data of both modalities are fused using advanced biohashing algorithm.

i) [Park, 2013] proposed a hand biometric authentication method based on measurements of the user’s hand geometry and vascular pattern. The proposed multimodal biometric system uses only one image to extract the feature points. This system can be configured for low-cost devices.

j) [Usharani, 2014] proposed a multimodal biometric system using palm vein and palmprint. The proposed fused feature technique improves the recognition rate in the range of 5.05% to 7.65%.

k) [Faris, 2014] proposed system fuse personal finger vein and iris which utilizes a vein feature matcher for finger vein and Hamming Distance Matcher for iris. It has been more secure than a framework used a single identification of personal feature.

Conclusion

This paper presents a detailed review on palm veins, dorsal hand vein, finger vein and multibiometric recognition algorithms. Such tools include image acquisition, preprocessing, feature extraction and matching methods to extract and analyze object patterns. Multi-biometrics topic has attracted more interest in recent research. It is used to identify individuals based on their physiological and behavioral characteristics for security purposes. Overview of biometrics showed that it is impossible to find the best single biometric suitable for all applications, populations, technologies and administration policies. Also, integration of biometric modalities can solve unimodal system limitations to achieve higher performance. Benefits and limitations of multi-biometrics discussed as we introduced it as a solution. In this paper, a state of the art survey of integration strategies, and fusion levels prior to matching and after matching are discussed design and finally, evaluate the multi-biometric systems raises many issues and trends.

Bibliography

[Marcos, 2006] Marcos Faundez-Zanuy, “Biometric Security Technologies”, IEEE Aerospace and Electronic Systems Magazine, 21(6), pp.15 - 26, 2006.

[Crisan, 2008] S. Crisan, Researches concerning the development of biometric applications using infrared radiation. Ph.D. Thesis, Technical University of Cluj-Napoca, 2008.

[Sarkar, 2010] IshaniSarkar, FarkhodAlisherov, Tai-hoon Kim, and Debnath Bhattacharyya, “Palm Vein Authentication System: A Review”, International Journal of Control and Automation, Vol. 3, No. 1, pp.27-34, March, 2010.

[Soliman, 2012] Hassan Soliman, Abdelnasser Saber Mohamed and Ahmed Atwan, “Feature Level Fusion of Palm Veins and Signature Biometrics”, International Journal of Video & Image Processing and Network Security IJVIPNS-IJENS Vol: 12 No: 01, pp.28-39, 2012.

[Haoluo, 2010] Haoluo,Fa-XinYu,Jeng-ShyangPan,Shu-Chuan Chu and Pei-Wei Tsai, “A Survey of Vein Recognition Techniques”,information technology Journal,vol.9,no 6,pp.1142-1149,2010.

[Ezhilmaran, 2015] D.Ezhilmaran and P. Rose Bindu Joseph, “A Study of Feature Extraction Techniques and Image Enhancement Algorithms for Finger Vein Recognition”, International Journal of PharmTech Research, vol.8, no 8, pp.222-229, 2015.

[Gad, 2011] R. Gad, M. Mohamed, and N. El-Fishawy, “Iris Recognition Based on Log-Gabor and Discrete Cosine Transform Coding” Journal of Computer Science and Engineering, vol. 5, no. 2, pp. 19-26, 2011.

[Gad, 2015] Ramadan Gad, Nawal El-Fishawy, AYMAN EL-SAYED and M. Zorkany,“ Multi-Biometric Systems: A State of the Art Survey and Research Directions”, International Journal of Advanced Computer Science and Applications(IJACSA), Vol. 6, No. 6, 2015.

[Bhatia, 2013] R. Bhatia, "Biometrics and Face Recognition Techniques” International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 5, pp. 93-99, 2013.

[Raut, 2015] Shriram D. Raut and Dr. Vikas T. Humbe, “A Distance Metric Computation Model for the Design and Development of Palm Vein Recognition

System"International Journal of Bio-Science and Bio-Technology, Vol.7, No.6, pp.201-206, 2015.

[Sasikala, 2016] R. Sasikala , S. Sandhya , Dr.K. Ravichandran and Dr.D. Balsubramaniam"A Survey on Human Identification using Palm-Vein Images using Laplacian Filter" International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 4, April 2016.

[Smorawa, 2015] DorotaSmorawa and MariuszKubanek"Biometric Systems Based on Palm Vein Patterns", Journal of telecommunication and information technology, Vol. 10, No24, pp.18-22, Feb. 2015.

[Bikoye, 2016] A.O. bikoye, C.Chukwu, Michael and B.AkinbowaleNathanie,l"An Improved Palm Vein Based Recognition System", International Journal of Computing and ICT Research, Vol. 10, Issue 1,pp.9-18, June 2016.

[Gayathri, 2015] R. Gayathri, L. Jayalakshmi, K. Mahalakshmi and K. S. Nevethitha, "multispectral palmprint recognition using ant Colony optimization", ARPN Journal of Engineering and Applied Sciences, VOL. 10, NO. 14, pp.5934 -5940, AUGUST 2015.

[Pooja, 2016] Pooja,Vinay Bhatia and TanujaDogra,"Palm Vein Recognition: An Advanced Biometric Technique for Authentication", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 8, pp.14882-14888,August 2016.

[Dere, 2017] Swapnali N. Dere and A. A. Gurjar, "Identification of Human using Palm-Vein Images: A new trend in biometrics", International Journal of Engineering and Computer Science, Vol. 6, Issue 1, pp.19867-19871, Jan.2017.

[Sayed, 2015] M. Sayed, "Palm Vein Authentication Based on the Coset Decomposition Method", Journal of Information Security, Vol.6, pp. 197-205, July 2015.

[Villariña, 2015] M.C. Villariña and Noel B. Linsangan, "Palm vein recognition system using directional coding and backpropogation neural network", Proceedings of the World Congress on Engineering and Computer Science 2015 Vol 2, 2015.

[Singh, 2015] Sandeep Singh and V.K Banga, " Human identification using palm vein images", International Research Journal of Emerging Trends in Multidisciplinary, Volume 1, Issue 7,pp.7-11, September 2015.

[Bayoumi, 2015] S. Bayoumi, A. Aldayel, M. Alotaibi, M. Aldraihem,S. Alrashed and S. Alzahrahi, "Class Attendance System Based-on Palm Veinas Biometric Information", Journal of Theoretical and Applied Information Technology, Vol.77. No.2, pp.266-272, July 2015.

[Ali, 2017] Yossra H. Ali and ZahraaNajiRazuqi, "Palm Vein Recognition Based on Centerline", Iraqi Journal of Science, Vol. 58, No.2A, pp: 726-734, 2017.

[Wang, 2014] C. Wang, M. Peng, Lingfeng X and T.Chen, "A single scale retinex based method for palm vein extraction", US National Library of Medicine National Institutes of Health, May 2014.

[Raut,2018] S. D Raut, V.T Humbe and A. R Shinde, " A validation of biometric palm vein recognition system using neural network approach palm vein biometrics", International Journal of Advanced Science and Research ,Vol 3,Special Issue 1,pp. 1-3, Mar. 2018.

[Naidile, 2015] Naidile.S and Shrividya.G, "Personal Recognition Based on Dorsal Hand Vein Pattern", International Journal of Innovative Research in Science,Engineering and Technology, Vol. 4, Issue 5, pp.3189-3196,May 2015.

[Belean, 2017] B. Belean, M. Streza, S. Crisan and S. Emerich, "Dorsal Hand Vein Pattern Analysis and Neural Networks for Biometric Authentication", Studies in Informatics and Control 26 (3),pp. 305-314, September 2017.

[Kanakamahalakshmi, 2016] K. S. kanakamahalakshmi and V. G. Siva Kumar, "Biometric recognition system vein pattern verification using palm dorasum", ARPN Journal of Engineering and Applied Sciences, VOL. 11, NO. 11, pp.6854-6858, JUNE 2016.

[Premavathi, 2018] C. Premavathi and P. Thangaraj, " Efficient Hand-dorsa Vein Pattern Recognition using KNN classification with Completed histogram CB in TP Feature Descriptor", International Journal of Recent Technology and Engineering (IJRTE), Vol. 7, Issue 4S, pp. 50-55,November 2018.

[Raghavendra, 2015] R. Raghavendra, J. Surbiryala and C. Busch, "Hand Dorsal Vein Recognition: Sensor, Algorithms and Evaluation", IEEE International Conference on Imaging Systems and Techniques (IST), Sept. 2015.

[Rajalakshmi, 2018] M. Rajalakshmi, R. Rengaraj, M. Bharadwaj, A. Kumar, N. N. Raju and M.Haris, " An Ensemble Based Hand Vein Pattern Authentication System", CMES, vol.114, no.2, pp.209-220, 2018.

[NishaCharaya, 2018] NishaCharaya and Priti Singh," Human Authentication Based On Dorsal Hand Veins: A Review", International Journal of Pure and Applied Mathematics, Vol. 119, No. 16, pp. 2175-2185,2018.

[Mulyono, 2008] David Mulyono and Horng Shi Jinn," A Study of Finger Vein Biometric for Personal Identification", International Symposium on Biometrics and Security Technologies conferences, April 2008.

[Pham, 2015] T. D. Pham, Y. H. Park, D. T. Nguyen, S. Y. Kwon and K. R. Park," Nonintrusive Finger-Vein Recognition System Using NIR Image Sensor and Accuracy Analyses According to Various Factors", US National Library of MedicineNational Institutes of Health, Vol.13,No.7,pp.16866-16894, Jul 2015 .

[MENG, 2018] X. MENG, Xiaoming XI, G. YANG and Y. YIN, "Finger vein recognition based on deformation Information", SCIENCE CHINA Information Sciences ,Vol. 61, May 2018.

[Ayappan, 2017] G Ayappan and A Shankar, "Finger Vein biometric Authentication System", International Journal of Trend in Research and Development, Vol. 4, No.2,pp. 51-56, Apr 2017.

[Ali, 2017] M. M. Ali, .K. Nusrat, A. A. I. Mohamed, M. A. Hagal,H. H. A. Almezogi, J. Rahebi and A. Hançerlioğullari, "Finger Vein Recognition with Gray Level Co-Occurrence Matrix based on Discreet Wavelet Transform", International Journal of Computer Science And Technology ,Vol. 8, No2, pp.108-112,June 2017.

[BADER, 2018]A. S. BADER and A.M.SAGHEER, "Finger vein identification based on corner Detection", Journal of Theoretical and Applied Information Technology, Vol.96, No 9,pp.2697-2705, May 2018.

[Janney, 2017]B. Janney.J, S. Divakaran and G.Umashankar, "Finger vein recognition system for authentication of patient data in hospital",International Journal of Pharma and Bio Sciences ,Vol.8,No.2,pp.(B)-5 -10, Apr 2017.

[Qin, 2013]H.Qin, L.Qin,L. Xue, X.He, C.Yu and X. Liang, US National Library of MedicineNational Institutes of Health, Vol.13,No.11,pp.15048-15067, Nov 2013.

[Das, 2019]R. Das, E. Piciucco, E. Maiorana and P. Campisi, “Convolutional Neural Network for Finger-Vein-based Biometric Identification”, IEEE Transaction for Information Forensics and Security,vol. 14, no. 2, pp. 360-373, Feb. 2019

[Kaur, 2012]K. Kaur and S. Bhushan, “Finger Vein Enhancement Using Segmentation and SVM”, International Journal of Computer Science and Information Technologies, Vol. 6, No5, pp.4549-4554, 2015.

[Down, 2012]Michael P. Down and Richard J. Sands. “Biometrics: An Overview of the Technology, Challenges and Control Considerations”,International Journal of Computer Applications , Vol. 39, No.10, pp.35-52,February 2012.

[Ross, 2007]Arun Ross, “An Introduction to Multibiometrics”, Proceedings of the 15th European Signal Processing Conference (EUSIPCO), 2007.

[Merati, 2011]A. Merati, “Multi-Modal Biometric Authentication with Cohort-Based Normalization”,Phd thesis, Faculty of Engineering and Physical Sciences University of Surrey, September 2011.

[Bharathi, 2016]S.Bharathi and R.Sudhakar, “Biometric recognition using dorsal and palm vein images”, International Journal of Advanced Engineering Technology, Vol. 7, No.2, pp.415-419, June, 2016.

[Stefani, 2016]E. Stefani and C. Ferrari, “Design and Implementation of a Multi-Modal Biometric System for Company Access Control”, 2nd International Conference on Data Compression, Communication, Processing and Security (CCPS), pp.1-10, 2016.

[Shruthi, 2013]Shruthi.B.M, P. Mohnani, Mallinath and Ashwin.R.G, “ Multimodal Biometric Authentication Combining Finger Vein and Finger Print”, International Journal of Engineering Research and Development, Vol. 7, PP. 43-54, No 10 July 2013.

[Christo, 2017]L. E.Christo and A. Zimmer, “Multimodal biometric system for identity verification based on hand geometry and hand palm's veins”, Communication papers of the Federated Conference on Computer Science and Information Systems, Vol. 13,pp. 207–212,2017.

[Raghavendra, 2014]R. Raghavendra ,Kiran B Raja, J. Surbiryala and C. Busch, “A low-cost multimodal Biometric Sensor to capture Finger Vein and Fingerprint”, IEEE International Joint Conference on Biometrics, December 2014.

[Trabelsi, 2013]R. B. Trabelsi, A. D. Masmoudi and D. S. Masmoudi, “A New Multimodal Biometric System Based on Finger Vein and Hand Vein Recognition”, International Journal of Engineering and Technology (IJET), Vol 5, No 4 ,pp.3175-3183,Sep 2013.

[Dhameliya, 2013]M. D Dhameliya and J. P Chaudhari, “A Multimodal Biometric Recognition System based on Fusion of Palmprint and Fingerprint”, International Journal of Engineering Trends and Technology (IJETT) – Vol.4, No.5, pp.1908-1911, May 2013.

[Siddharth, 2017]J. A. Siddharth, A.P. H. Prabha, T.J. Srinivasan and N. Lalithamani, “Palm Print and Palm Vein Biometric Authentication System”, Artificial Intelligence and Evolutionary Computations in Engineering Systems, Advances in Intelligent Systems and Computing book,2017.

[Park, 2013]GiTae Park and Soowon Kim, “Hand Biometric Recognition Based on Fused Hand Geometry and Vascular Patterns”, US National Library of MedicineNational Institutes of Health,pp.2895-2910, February 2013.

[Usharani, 2014]V.Usharani,S.V.Saravanan, “Multi modal biometrics using palmprint and palmvein”, Journal of Theoretical and Applied Information Technology, Vol. 67 ,No.1,pp.177-185, Sep.2014.

[Mohammed, 2014]F. E. Mohammed, E. M. ALdaidamony and A. M. Raid, “Multi Model Biometric Identification System: Finger Vein and Iris”, Vol.4, No4, pp.50-55, Sep. 2014.

Authors' Information

Mona A. Ahmed *– PhD. Student, Computer Science Department, Faculty of Computer and Information Sciences Ain Shams University, Cairo, Egypt,* ***e-mail****:* eng.monaa_cs@hotmail.com

Abdel-Badeeh M. Salem *– Professor of Computer Science, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt, Head of Artificial Intelligence and Knowledge Engineering Research Labs,* *http://aiasulab.000webhostapp.com/ ; http://staff.asu.edu.eg/Badeeh-Salem*

e-mail*:* abmsalem@yahoo.com

EVENT-DRIVEN MODELS[1]

Dimiter Dobrev

Abstract: *In Reinforcement Learning we look for meaning in the flow of input/output information. If we do not find meaning, the information flow is not more than noise to us. Before we are able to find meaning, we should first learn how to discover and identify objects. What is an object? In this article we will demonstrate that an object is an event-driven model. These models are a generalization of action-driven models. In Markov Decision Process we have an action-driven model which changes its state at each step. The advantage of event-driven models is their greater sustainability as they change their states only upon the occurrence of particular events. These events may occur very rarely, therefore the state of the event-driven model is much more predictable.*

Keywords: *Artificial Intelligence, Reinforcement Learning, Partial Observability, Event-Driven Model, Action-Driven Model, Definition of Object.*

ITHEA Keywords: *I.2.6 Learning*

Introduction

What is an object? An object can be your favorite song, a single word or a semantic category (such verb or noun). An object can also be a person, an inanimate item or an animal; it can be your house or the whole neighborhood you live in.

[1] The author is grateful to the Bulgarian National Scientific Fund for the provided financial support, under project No. DN 19/1 (2017).

We are all able to identify objects. You will recognize your favorite song even if you hear only a short fragment of it. You may hear it performed by another singer or even from an audio stream of poor quality, but you will still know it is your favorite song.

In your mind the objects are structured according to some hierarchy, such as animal–dog–poodle –your poodle. Some objects possess the property *uniqueness*. E.g. your dog is unique and if you groom it will then be a groomed dog. If you groomed a random dog this will not imply that all dogs have been groomed.

Uniqueness will not be an invariable feature. You may believe that a certain person is unique, but all of a sudden you find out that he or she has a twin. For years on end you thought you were talking to the same person and now it turns out that you have talked to two different persons. Another example: There is a chair in your house and in your eyes this chair is unique. You certainly know that thousands of this type of chair have been produced, but your chair is the only one of these chairs in your house. If you repaint the chair you will expect it to be repainted the next time you see it. It may turn out that there are two such chairs in your house and you have repainted one of them.

How do we recognize (identify) objects? You may hear the beginning, the middle or the end of your favorite song. You may see a person you know in profile or in full face. You can explore a neighborhood by starting from many different junctions. Thus, in order to recognize an object, we need not see it each time in the way we have seen it already.

An oriented graph is also an object and we recognize it by following a certain path in that graph. Of course the path must be sufficiently specific to that oriented graph, otherwise we cannot be sure it is exactly the graph we mean. If we are not sure, we can make an experiment by turning in a direction which would provide more information. Let us say you are at some street crossing, but do not know if the crossing is in your neighborhood. So you take to the main street in order to find your bearings. Or you see someone you probably know, but are not sure. Then you can take a look at that person from all sides or shout out his name to see if he will look your way.

But we are not always in control of the situation or able to make an experiment. Sometimes we cannot hear well a spoken word. We can ask the speaker to repeat it, but in most cases it is a done deal and the party is over.

Objects in this article will be presented as oriented graphs (event-driven models). The vast majority of objects are not in front of eyes all the time, i.e. they appear and vanish from sight. Are there objects which we observe all the time? Are there oriented graphs in which we stay all the time and never leave? Yes, there are such objects (models). One example are the days of the week. They give us a model with seven states (Sunday to Saturday) and at any point of time we are in one of these states. Another example is our neighborhood. If we spend all our life in that neighborhood without ever leaving it, the neighborhood will be permanently within our sight.

Most event-driven models will have a special state which we will term *outside*. An object will be off our sight when we are in an *outside* state.

The key question with event-driven models is “Where am I now?”, i.e. in which state am I? If the event-driven model is an object, the key question is “Do I see the object now?”. In other words, am I in the *outside* state or in some of the other states?

We will begin this article by presenting an intuitive idea of event-driven models. Then we will make a comparison between event-driven models and action-driven models. Next, we will provide an informal description of the problem and will explain why we will not try to find the initial state. The term *history* will be defined and the objective of the agent will be announced. We will demonstrate that we can have a variety of criteria and will challenge Sutton’s assertion in [2] that there is only a single criterion. We will prove that the term *discount factor* must not be part of the definition of Reinforcement Learning (RL), because the *discount factor* is part of the strategy rather than of the meaning of life. We will continue with a discourse on whether the model of the world should be deterministic or nondeterministic. As regards incorrect moves, we will demonstrate that we can do without them, but should preferably assume that incorrect moves do exist. A definition of RL will be provided. Then we will define the *perfect model*, which is an action-driven model. We will then complicate the model by adding randomness. We will discard the assumption that random values have any distribution, and will demonstrate that such an assumption is wrong. We will point out that there must be a *trace*, i.e. some particular occurrence which distinguishes one state from another. Next, we will introduce the concept of *exhaustive models* and will demonstrate that they are as unattainable as the perfect ones. We will explain what is an event and will proceed with yet another complication of the model which will transform it from action-driven to event-driven. In fact we will replace actions with random events. We will add variables to the model. Finally we will create the Cartesian product of all adequate models discovered. That Cartesian product will be the model which we are looking for and which provides the best possible description of the world.

The intuitive idea

Before we entangle the reader in a swarm of words, we will try to offer an intuitive idea of what the event-driven model is. This is an oriented graph similar to the ones shown in Figures 1 and 2.

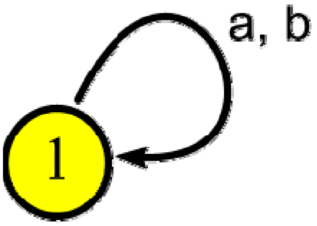


Figure 1

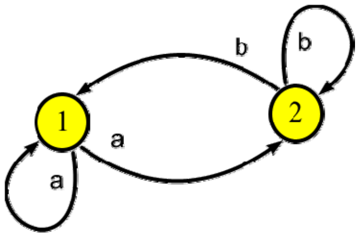


Figure 2

Here *a* and *b* are events. If *a* and *b* were actions, then Figures 1 and 2 would depict action-driven models. An action is of course an event, too. So action-driven models are a special case of event-driven models.

The model in Figure 1 is fairly simple and if we gathered statistics from it all we would get to know is how many times event *a* and respectively event *b* has occurred. In turn, that would indicate which event is more likely to occur.

The model in Figure 2 is more interesting. Here, if we are in state 1, event *b* cannot happen. The same applies to state 2 and event *a*. This model therefore *predicts* the next event (*a* or *b*). If we know the state we are in, we will also know which *the next event will be*.

If we flip the arrows (Figure 3), we will get a model which *remembers* the last event (*a* or *b*). If we know the state we are in, we will also know which event *was the last to occur*.

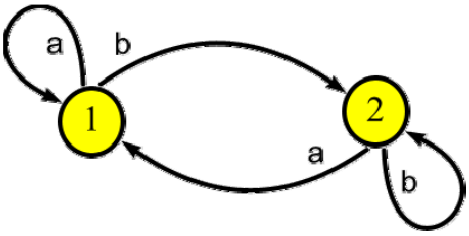


Figure 3

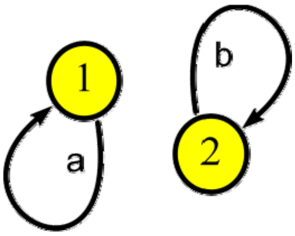


Figure 4

The commonality between Figure 2 and Figure 3 is that in either case there must be some difference between states 1 and 2. Something specific must occur in these states. We will name this specific occurrence *trace*. In the absence of a trace, i.e. of something specific, these two modes are useless.

A trace is indispensable. Each and every oriented graph is a model of the world (events correspond to arrows). If there were no trace (i.e. in the absence of a specific occurrence), the model would be inadequate and useless.

In Figure 2, the difference between states 1 and 2 is that state 1 has only outbound *a* arrows (and state 2 has only outbound *b* arrows). If this is all the difference they have (i.e. if there is no trace) we will never know whether we are in state 1 or in state 2, because when viewed from the perspective of the past the two states are indistinguishable.

In Figure 3 we know exactly the state we are in, but this would be useless if these states are indistinguishable from the viewpoint of the future (i.e. if there is no trace). In other words, it would be absolutely meaningless to remember the last event (*a* or *b*) if it does not bear any consequence for the future.

Let us illustrate this with a criminal scenario (Figure 4). We are looking for the perpetrator of a murder. Who is the perpetrator, the postman or the tramp? If the postman is the murderer, we are in state 1. If the tramp is the murderer, we are in state 2. Event *a* is “We proved that the postman perpetrated the murder”. Accordingly, event *b* is “We proved that the tramp perpetrated the murder”. The question is, which state are we in. In other words, who is the murderer? We will find out only when we come across hard evidence (the event *a* or *b*). Unfortunately, we may never come across hard evidence. The point is how to predict who the murderer is before we get hold of hard evidence. We may hypothesize that if the murder was perpetrated by the postman, we are more likely to come across clues that confirm he is the perpetrator than to clues that acquit the postman. In other words, the clues do not expose the perpetrator, but set the crosshairs on the more likely perpetrator. The clues will be provided by the *trace* in this model.

The model in Figure 4 is not very much to our liking, because we need recurrence. We want to traverse each arrow multiple times. Figure 4 however is an either-or scenario because the occurrence of *a* forecloses the occurrence of *b*.

Recurrence is very important. Heraclitus said that no man steps in the same river twice, for it is a different river each time. Likewise, no one can enter the same room twice (or meet the same person twice). Each time the room is different. Somebody has painted the walls, somebody else has rearranged the furniture. Somebody may come in, somebody else may go out. If all things are unchanged, a butterfly will come by and spoil all recurrence.

Nonetheless, we wish to step in the same river or room many times and meet repeatedly one and the same person. They may not be exactly the same, but we are willing to disregard minor or even major differences for the sake of making the world more simple and comprehensible.

This is exactly the underlying idea of event-driven models. These are models with a few states, which we visit multiple times and each visit lasts long (consists of many steps). The *modus operandi* of action-driven models and in particular of perfect models is exactly the opposite. A perfect model always exists. The easiest way to construct such a model is to arrange the states in a row and traverse all these states one time only. The state in an action-driven model describes everything, which essentially prevents one state from occurring twice (same as we cannot step in the same river twice).

Event-Driven vs Action-Driven

What is the difference between event-driven and action-driven models? In Markov Decision Process (MDP) the model is an oriented graph which changes its state after each action (i.e. at each step). This model resembles a machine which ticks too fast. It is fairly difficult to answer the question “Where am I now?” with such a model. Therefore, it is a huge challenge to tell the current state. Let us imagine that human being is a step device which makes 24 steps per second. (Cinematographic movies run at a rate of 24 frames per second, but we cannot see the intermittency and perceive the imagery as seamless. So 24 steps per second is a good proposition.) If your state was changing 24 times per second you would hardly be able to tell your current state. Now let us have a model which does not change at every action, but at the occurrence of a few more interesting events, e.g. sunrise and sunset. These events occur once in 24 hours. At sunrise we transition from night to day and vice versa at sunset. When I ask you if it is day or night now you will most probably be able to give the right answer. In other words, the current state of the event-driven model is far more predictable (because it is a much more stable and changes less often).

Another difference is that in action-driven models we can try describe the behavior of the world without considering the behavior of the agent, but the event-driven model requires us to describe the world and the agent living in it as a composite system.

The MDP for example calculates the probability of a certain transition for a certain state and action. It does not however calculate how probable is it for the agent to perform a certain action because it tries to describe only the world without the agent. In event-driven models we have to describe the world and the agent as a composite system since when an event occurs we cannot tell if it occurred only because the agent wanted it or because that’s the way the world goes. In MDP we do not predict the past, because the choice of the arrow which takes us to a certain state depends not only on the world, but on the agent as well. For example (Figure 5), if we are in state 2 and the last action is a red arrow, then we may have come to state 2 from states 1 and 5. If we knew that in state 1 the agent will never choose a red arrow, then the previous state must have been 5.

In event-driven models we make no distinction between past and future. We collect statistics for both and try predict both.

Describing the world and the agent as a composite system is much easier than describing the world without the agent. By way of example, you will never dye your hair in green colour, which is a simple description of you and of the world you live in. The reason is irrelevant – it may be that you are reasonable enough not to do so or that green dye is not available. If someone asked “Will your hair be green tomorrow?” you may answer confidently that this cannot happen. How you know it – from the analysis of your own behaviour or because of some limitations from the world – does not matter. What matters is that your hair cannot be green tomorrow.

Informal description

Initially we will provide an informal description of the problem. We have a world (environment) and an agent who lives in that world. The world is an oriented graph similar to the one depicted in Figure 5. The agent moves from one state to another by following the arrows. The states and the arrows are tagged with some labels. However, in Figure 5 we have used color codes instead of labels. As the agent moves, he enters (observes) information (the label of his current state) and performs actions (chooses the label of the arrow of his next move). Note that the agent chooses only the label (color) of the arrow, but not the particular arrow. E.g. if from state 6 the agent chooses a blue arrow, he cannot know whether the blue arrow will take him to state 5 or to state 7. Some choices may not be available, for example there are not outbound red arrows from state 2. So in state 2 the agent cannot choose to continue his path on a red arrow and must choose a blue one.

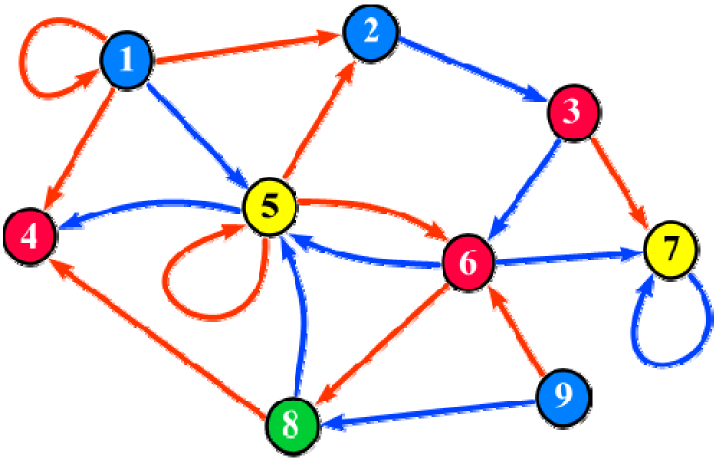


Figure 5

We will name the agent’s path in the oriented graph *the life* of the agent. Any state can be the begging or the end of that life. However, there are states which – if they are part of the path – can only be the beginning of that path. We will term these states *absolute beginning*. These states do not have any inbound arrows, so there cannot be any past before them. An example in Figure 5 is state 9. State 1 is not an absolute beginning because of the red loop. Accordingly, we will term the states without outbound arrows *sudden death*. If such a state is part of the path, it must always be the last state. There is no future after such a state, cf. state 4 in Figure 5. State 7 is not sudden death because of the blue loop.

By naming the states without outbound arrow *sudden death* we mean that the agent’s life may end in two ways – either we shut him down or he reaches a *sudden death* state and stops. We will say that shutting down the agent is natural death and when the agent stops on his own it is sudden death. (In everyday speech, when someone shuts us down we do not say it is natural death, but a murder. But in this article we assume that we will shut down the agent when much time has elapsed. So we will associate shutting down with dying of old age, therefore natural death.)

Note that life need not necessarily start from some absolute beginning. The past may be infinite in the same way as the future may be. Therefore, life can be an endless path which has no beginning and no end.

In MDP and RL we examine two scenarios. In one scenario the agent sees everything (Full Observability) and in the other scenario he sees only partially (Partial Observability). In Figure 5 we would have Full Observability if all the states had different colors or if the agent observed not the colour, but the number of states (i.e. if the agent sees which is the state he is in now).

Most articles suggest that in MDP we have Full Observability. This is the special case. They use the term Partially Observable Markov Decision Process (POMDP) to describe the generalization of Partial Observability. This article puts things the way round. We will use the term MDP to denote generalization, while for Full Observability will use the term Fully Observable Markov Decision Process (FO MDP).

Similarly, here we will assume that in Reinforcement Learning we have Partial Observability, and will flag explicitly the cases when we examine Full Observability.

The initial state

In Reinforcement Learning (RL) we make references to an initial state and to a current state. In order to describe one path (life), we need a state to start from. Most authors tend to go for the initial state guided by the assumption that it is easier to predict forward in the direction of the future than backward in the direction of the past. Therefore, going in the direction of the arrows is easier than going against the arrows. In Markov Decision Process (MDP) for example we have probabilities which indicate what we expect the state to be if we move ahead with certain action, but there are no probabilities for the opposite direction. In this article we will not refer to initial states but only to current states. This comes from the assumption that the past and the future are on a level playing field and are equally difficult to predict.

There are many reasons why we will not be interested in the initial state. There may not be any initial state at all, because the past can be infinite. Which is the initial state of a human being? Is it the moment of birth or the moment of conception? In our model of the world there are moments which have occurred before our birth. Therefore, birth in our model is not an absolute beginning. There is one model of the world with an absolute beginning, and this is the Big Bang theory. Is that a sound theory? Rather than asking whether a theory is true or false, we need to know if the theory is fit for our purpose (does it describe the world). There is a raft of theories which describe the word. We will deem that a theory is true insofar as it is impossible to design an experiment which disproves that theory.

The questions we will ask in RL therefore are “What is the world?” and “Where are we now?”. More often than not, we ask “Where am I now?” rather than “Where was I at the time I was born?”. Consequently, we will look for the current state and not for the initial state.

What is History?

What is life? Life is the whole history, i.e. a history which cannot continue either because it is infinite or because there is no next step, perhaps because the history has come to sudden death or because we have shut down the agent (natural death).

What is history? **Definition:** A *truncated history* will be the series of actions and observations from the initial moment to the current moment, including the last correct action.

$$a_1, v_1, a_2, v_2, \ldots, a_t, v_t, a_{t+1}$$

Definition: We will have *history* when we add all the incorrect (bad) moves which we have tried and which we know to be incorrect.

$$bad_0, a_1, v_1, bad_1, a_2, v_2, \ldots, bad_{t-1}, a_t, v_t, bad_t, a_{t+1}$$

In the above sequence *a* is action and *bad* is a set of actions. Note that each next step begins with the next correct move. The *bad* set can be regarded as part of the observations made at the previous step.

Definition: A *full history* is the result which we obtain when we replace the *subset* of incorrect (bad) moves which we tried with the *full set* of incorrect moves.

$$full_0, a_1, v_1, full_1, a_2, v_2, \ldots, full_{t-1}, a_t, v_t, full_t, a_{t+1}$$

What is the difference between *history* and *full history*? The first is what we have actually seen along the path and the latter is what we would have seen if were observing more attentively. So $bad \subseteq full$. That is, the moves which we have tried and know to be incorrect are part of all incorrect moves. We may keep trying more moves until *bad* = *full*, however, in doing so we may inadvertently play a correct move (different from a_i) which will make the history completely different.

Definition: A *local history* of length *k* will be the last *k* steps of a history. That is, the end of some history.

Definition: An *approximate history* will be some form of incomplete description of a history. Presumably, the history is too long and we will hardly be able to remember the whole of it. Thus, we will remember only the end of it, or a few more important events (the time of their occurrence), or some statistics about that history. Usually, we do not need the full history and an approximate history is sufficient to build a model of the world and plan our future actions.

Note that what matters for the agent are not the states he has traversed, but the observations he has made. While the life corresponds to a path in the oriented graph (the model of the world), two different paths may correspond to one life.

The agent's objective

What is the agent's objective? His objective is to achieve a better life.

But which life is better? We need a relation which compares lives. This relation must be a quasiorder (preorder). That is, it must be both reflexive and transitive. If we add antisymmetry, we will get partial order. We do not, however, need antisymmetry, because it will not be a problem where two lives are equally successful.

We would like this relation to possess some monotonicity. Let us divide life in two halves and let us have two lives. If the first and the second halves of Life 1 are better respectively than the first and the second halves of Life 2, we would expect Life 1 to be better than Life 2 overall. The opposite would be bizarre.

The relation in question can be obtained by adding up rewards and regrets. But in this article instead of rewards and regrets we say *scores*. Rewards are positive scores, while regrets are negative scores. For some scores, though, we do not know whether they are positive or negative. Relativity is pervasive. For example, if you earn a *D* grade at school, is that positive or negative? For some students or parents it might be positive, but for others it may be disappointing.

Thus, summing up all the scores we will give a number which represents the overall score of the whole life. In [3] the number is equal to the sum of the scores, but life there has a fixed length and the sum equals the arithmetic mean. In [4] the number is calculated as an arithmetic mean of the scores. Later articles use a *discount factor*, which is a mistake and later we will discuss why this is a mistake.

Most articles imply that the agent gets a score at each step, e.g. [3, 4] and many others. This proposition can be seen as an attempt to simplify the discourse, but in principle the idea of having a score at each step is illogical and unnatural. At schools for example they do not examine each kid every day.

Another possible approach is to assume that if we do not have any score, then the score is zero, however all these zeros will alter significantly the arithmetic mean. That would equally alter our strategy. If for example we play chess and a draw is zero, then ending in a draw or continuing the game would be all the same. The length of the game will also be very important. Thus, if our mean score is negative, we will try to extend the game as much as possible. If the mean score is positive we will try to cut the game as short as possible. While it seems quite reasonable to try end the game as soon as possible, we would be prone to risk the victory for the sole sake of ending the game faster. At chess tournaments they calculate the arithmetic mean of all wins, losses and draws without counting the average number of moves per game.

Hence, while most articles imply that the score is a real number, we will suggest that the score can be either a real number or the constant *Undef*. Therefore, we will suppose that at each step we may or may not have a score.

Different criteria

Most articles imply that we have only one criterion against which life can be measured. In economics this criterion is money and the better strategy is the one which reaps more profit. We will assume that there are two criteria: of course we want the highest profit, but without ending up in jail.

We will look at objectives which are defined by two or more criteria. Let us have two persons, one has more wealth, and the other has more children. Who has been more successful? If the two criteria are equally important, Life 1 must outperform Life 2 on both criteria in order to be better than Life 2. This makes the “better life” relation nonlinear, but we already said that we wish this relation to be quasiorder, therefore it need not necessarily be linear.

We may well have two criteria wherein one criterion has priority over the other. For example, if we write a computer program for a self-driving vehicle, the objective will be less delays and less casualties. We may approach this by expressing delays and human life in monetary value and seeking the least-cost policy. That is, reduce the two criteria to one. This is exactly what we do when we drive a car. We estimate the value of the delay and rush, thus heightening the risk level. So, when we run behind schedule we are prone to take a higher risk of killing somebody. While human drivers can behave like this, your program cannot. As soon as you put a price tag on human life, you will be condemned of inhumanness. That is why you should prioritise the second criterion and compare the two policies on the basis of casualties. When two policies yield the same number of casualties, the one which yields less delays wins.

Therefore, we will evaluate human life as infinitely more expensive than delays. This will give us a linear order, but with two criteria.

In this train of thought, when we apply *n* criteria, the score will be an *n*-tuple. The value of each coordinate will be either a real number or the *Undef* constant. The better life will be determined by calculating the arithmetic mean of each coordinate, and the so obtained vector will describe the ultimate result of the whole life. Whether the life is better or not will be determined by comparing the vectors so obtained.

In order to simplify the discussion, in this article we will assume that the criterion is only one. We will note though that Sutton in [2] shoots wide of target by tendering his *"reward hypothesis: all goals and purposes can be well thought of as the maximization of the expected value of the cumulative sum of a single externally received number (reward)".*

The discount factor

As mentioned already, the use of a discount factor in the MDP and RL definitions is a mistake. This mistake occurs often when we confuse "what" and "how". What do we want to do and how shall we do it? Well, where would we like to go for a vacation? Our first thought is Honolulu. Then we reckon that Honolulu is too far and on second thought we want a nearer place. This is a mistake, because we messed "what" and "how". What do we want and how shall we achieve it. We want Honolulu and the bare fact that we go to a nearer place does not change the place to which we want to go.

The concept of a discount factor derives from a perfectly natural strategy which tells us that the rewards nearby are more important than those far away. Popular wisdom puts this strategy in a nutshell by the saying "A bird in the hand is worth two in the bush".

The discount factor relates to the strategy rather than to the meaning of life. As from which moment shall we apply the discount factor? That should be the current moment, but it is a moving target, not a fixed time point. If we applied the discount factor as from the initial moment, that would make the beginning of life much more important than its end, which makes little sense.

What was Napoleon's strategy shortly before the battle at Waterloo – "First things first, let's win Waterloo and bother about chores later." So on the eve of the battle it was Napoleon's do-or-die, but if we look at his life as a whole Waterloo does not bear that much weight.

This means that using a discount factor to determine the best of two or more lives is a mistake, at least because the time from which the factor is to be applied and the value of the factor are unknown. If we are less certain we would choose a lower discount factor and hunt for the nearest rewards. The more confident we become, the farther we look in the future and the more prone we are to let the bird in the hand go in order to chase those in the bush. In other words, the more confident we are the more proximal to 1.00 our discount factor will be.

The good thing about the discount factor is that we can use it to evaluate infinite life. This will give us a number which is the sum of a geometric progression. How can we compare two infinite lives and say which one is better? By the following expression:

$$Life1 \geq Life2 \Leftrightarrow \exists n\, \forall (k \geq n)\ begin(Life1, k) \geq begin(Life2, k)$$

Here *begin(Life, k)* is the beginning of *Life* the length of which is *k*. The [$\geq$] symbol is our quasiorder relation which tells us which life is better. We assume that this relation is defined for finite lives and continues it for infinite lives as well. The above formula can also be used to compare a finite and an infinite life, assuming that if *k* is greater than the life's length then *begin(Life, k)= Life*.

What is the model as such?

As we said the model of the world will be an oriented graph. Several questions arise. First, is the model of the world deterministic or nondeterministic? Second, what type will be the randomness if we go for nondeterministic graphs? In [5] we discussed two types of randomness. With the first type of randomness we know the exact probability at which each arrow can be chosen (MDP). In the second type of randomness we know the possible arrows, but do not know the probability of choosing each arrow. Figure 5 depicts the second case because the probabilities of same-colour arrows are not indicated. In [5] we also examined a combination of the two randomness types wherein we do not know the exact probability but have an interval and know that the probability is in that interval.

In [5] we demonstrated that with RL the deterministic model and the models with different types of randomness produce equivalent definitions. Therefore, we cannot know whether the world is deterministic or nondeterministic. Of course this implies that we live only one life in the world. If we lived two lives in the world and did the same things each time, the two lives would be identical in the deterministic world and almost surely different in the nondeterministic world.

An important assumption is that with RL we live a single life and have a single history to draw conclusion from. Assuming that we live several times in the same world and gain experience from several histories will produce a substantially different problem where the deterministic and the nondeterministic models are not equivalent.

In this article we will start with a deterministic model, which we will name “perfect model”. Then we will define a nondeterministic model or “model with randomness”. Then we will replace actions with events and thus obtain an event-driven model.

Where is the score? In this article the score will be part of the observation. In most articles this is not the case. Typically we have two different observations. The first one is named *observation* and indicates the state we are in (Full Observability). The second one is named *score*. The first observation is a label of the state while the second observation is a label of the arrow. In this article we will have a single observation and the score will be part of that observation.

Incorrect moves

Should we allow for incorrect (bad) moves in the model or assume that each move is correct? For each model with incorrect moves we can construct an equivalent model in which all moves are correct. In [6] we did something similar by building a total model. That is, we added an observation *bad*, which is returned whenever an incorrect move is made.

Let us have two programs for our agent, a Total AI and a Partial AI. The first program will require all moves to be correct, while the second one will permit incorrect moves.

If we let the Total AI deal with the world presented as a total model, we will make the agent's task very demanding. In this case the agent will search for patterns in the series of the tried incorrect moves. The agent will not find any pattern, because it does not exist, but will never stop his hunt for patterns. Also, he will try to play an incorrect move twice. Finally the agent will realise that nothing happens by these experiments and will stop making these attempts. This agent will not know that trying an incorrect move does not change anything (other than realizing the move's incorrectness, but if he knew that already he does not change anything). These things can also be learnt, but whenever this agent is in a gridlock and ponders what to do he will again and again try things which are not worth trying. Life of the Partial AI in the partial model will be much easier because the Partial AI will be born with some knowledge about incorrect moves and will not sweat to discover and learn this knowledge on its own.

Accordingly, we had better assume that there are incorrect moves. This makes the world greatly simpler and comprehensible. The agent is spared the search of patterns which do not exist and the search for which would be a mere waste of time. Moreover, incorrect moves are a nice example of semi-visible events (these will be discussed in the next sections). Incorrect moves are also a nice example of test states (we discussed them in [5] and will expand in the next article).

Reinforcement Learning

Now let us provide the formal Reinforcement Learning definition which we will use. We will list what is given and what is to be found. Given are the following elements:

A – the set of the agent's possible actions;

V – the set of possible observations;

$Reward : V \rightarrow \square \cup \{Undef\}$ (a function which for each observation returns a score or *Undef*);

H – history or approximate history of what happened until moment t.

Given is also that a perfect model of the world exists and that model is in some current state s_t.

What is to be found? We try to answer three questions:

1. What is the world? (establish the model of the world);

2. Where am I now? (determine the current state of the world);

3. What should I do next? (decide our next action as well as our further actions, aiming to maximize the arithmetic mean of scores).

Note: This article will deal mainly with questions 1 and 2, while scores are only relevant to the answer of the third question. That is why we will not discuss scores in the next sections.

The perfect model

Let us now present the perfect model of the world:

S is the set of the internal states of the world,

s_t is the current state of the world, and

$G = \langle S, R \rangle$ is a total and deterministic oriented graph.

$R \subseteq S \times A \times S$

$View: S \rightarrow V$

$Incorrect: S \rightarrow P(A)$

Each edge (arrow) has a label which indicates the action we must make in order to traverse that arrow. Arrows and their labels are determined by the relation *R*. Each vertex (state) has two labels which indicate (i) what we see and (ii) the moves that are incorrect in that state. The labels therefore are (i) the observation in that state and (ii) the set of impossible actions. These labels of the state are determined by the functions *View* and *Incorrect*.

We will assume that the oriented graph *G* is total and deterministic. Thus, from each state per each action there is an outbound arrow and that arrow is unique. In this case the arrows that represent incorrect moves are somewhat redundant, because we will never use them, but we suppose that they exist as well, because further down we will discuss another model of the world wherein the set of incorrect moves is variable rather than constant. This means that from the same state we may have different incorrect moves at different moments.

Note that if we have a perfect model of the world and know the current state of the world, then the future is completely determinate. So, if we knew the next actions of the agent we would be able to tell exactly what will happen. Unlike the future, the past is not completely determinate because there may be multiple initial states which lead to that current state through history *H*. And even the full history is not determinate, because different initial states may have different full histories. Of course the various full histories must be consistent with history *H* (they must satisfy the condition $bad \subseteq full$). We will assume that there is at least one possible initial state which leads to the current state through history *H*, because we suppose that history *H* has occurred in that model anyway.

If, in addition to the current state we know the initial state as well, then from the perfect model we can obtain the so-called *abridged model*. In this model, all states and arrows that we have not been traversed through history *H* will be discarded. In the abridged model we can collect statistics by counting how many times we have traversed each arrow. These statistics will enable us predict the past, the future and the agent's behavior. By knowing how many times the agent has chosen action A or B in state *s*, we will be able to predict what the agent would do if he finds himself in that state. Indeed, we will try to predict our own behavior because as we said we will regard the world and the agent as a composite system.

The abridged model may not be perfect. We may come across an arrow which we have never traversed (i.e. an arrow which is absent from the abridged model). Nevertheless, the abridged model presents all we may know because if we never traversed an arrow we have no way to know where it leads to.

Random variables

Our effort to establish a perfect model of the world is a rather ambitious task which would be feasible only if the world is extremely simple. We are certainly interested in more complex worlds, therefore we can reasonably suggest that the complexity of the world is and will be far beyond full understanding.

Let us assume that there is a limit to our knowledge and that there are things which we will never be able to predict. We will describe them with the term *random variables*. We will regard them as dependencies the complexity of which is beyond our predictive capabilities. For example, if I threw a dice I would expect that it will fall on any number from 1 to 6 with a probability of 1/6. If I had a perfect model I would able to tell the exact number. There is such a perfect model and once I throw the dice I would know which number falls up, but the point is that I need to know it yet before I threw the dice. I cannot know which side will fall up before I threw the dice.

Let me note that if I were extremely smart I would be able to tell which side will be up. If I cannot tell it means that either I am not smart enough or do not have enough information on the basis of which to foretell which number will fall up.

I assume that I am not smart enough and my best guess is a number from 1 to 6 with a probability of 1/6. By saying “my best guess” I do not mean that I will not pursue an even better guess. I might suggest that the dice is skewed and yields 6 with a probability greater than 1/6 or if I blow a breath onto the dice the probability of getting 6 will raise. The last supposition is regarded as superstition (i.e. false), but here we will try to find a statistical dependence between events and if the statistical data indicates that breathing onto the dice works, this will be taken as proof even though it turns out that it has happened by chance.

What is a random variable? Let us suppose I am about to tell you either *black* or *white*. What do you expect to hear – *black* or *white*? I bet you have no idea what exactly I will say. Hence, you expect to hear *white* with a probability in the interval [0, 1].

Imagine I have a dice with one white side and five black sides. You know I will throw that dice and this is how I will decide whether to say *black* or *white*. So you expect *white* with a probability of 1/6. Now imagine I have another dice with two white sides and four black sides. I will throw one of my dices to decide whether I say *black* or *white*. Now you expect to hear *white* with a probability in the interval [1/6, 1/3]. Suppose you know I am more likely to throw the first dice than the second dice. Now you expect *white* with a probability in the interval [1/6, 1/4).

Let me now switch to alternating mode and say *white*, then *black*, then *white*, and so on. Now you expect to hear the opposite of what I said last time. This however is a memory-dependent pattern. I would like to describe non-parameterized random variables. A parameterized random variable is a function which at certain parameters returns a non-parameterized random variable. Later we will describe an oracle α, who depends on the past, on the future and on the current state. Therefore the past is one of the possible parameters.

Suppose when I am in good mood I say *white* or at least I am more likely to say *white*. This adds another parameter to the model, i.e. my mood. So you may try to predict my mood or if it is fully unpredictable then you have a random variable which does not use my mood as a parameter. The variable will nevertheless depend on my mood, but will not be a parameter. It will be a hidden parameter or a parameter which we will not try to predict.

You might suppose that what I am going to tell you is the worst of all worsts. This is a frequently used supposition. In chess playing computer programs, the Min-Max algorithm assumes that the opponent will play the move which frustrates us most. We therefore imagine an opponent whose aim is to hurt us. We do not always anticipate the worst. Sometimes we look forward to the best. E.g. when you come back home from a long journey, what do you expect for dinner? You expect the best because there is someone who loves you and has cooked your most favorite meal.

Let us assume you expect the best. If for some reason *white* is better than *black*, you would expect to hear from me *white*. It may happen that you do not know what is better and will find that out only after the event. Then your expectation will depend on the future. Therefore it will depend on yet another parameter.

Non-parameterized random variable

So what is a non-parameterized random variable (NPRV)? Most authors suggest that a NPRV has a strictly determined probability (or distribution if there are more than two possible values). Equally this is the supposition in MDP. The MDP chooses randomly the next state, but this randomness is not purely random due to the assumption that the probability of a certain state being selected is precisely defined.

Here we will assume that there are two levels of uncertainty. At the first level we do not know what will happen, but know the exact probability *p* for it to happen. At the second level of uncertainty we even do not know the probability *p*. It may be that *p* does not exist at all or it may exist, but we have no way to know it.

When we try to predict a certain event let a certain probability *p* for this event to occur. If we are smart enough and have the necessary data, we can predict that probability. Fine, but the event will also have a certain value when it occurs and if we are that smart we may even foretell that particular value. At the first level of uncertainty we admit that we cannot foretell the result, while at the second level of uncertainty we go further and admit that we are unable to even foretell the probability.

Can it be that an event has not any exact probability whatever? If it has some probability *p* and if we observe the event for infinitely long time, the statistically obtained probability should tend to *p* (the Law of large numbers). If we observe the event endlessly, the statistics may not tend to a certain value (we will have limit inferior and limit superior, but they may not coincide).

We can easily imagine a situation where we do not know the exact probability, and all we know is that it is somewhere in the range of *[0, 1]* or *[a, b]*. It is highly difficult to design a physical experiment, which (i) gives us such probability and (ii) that probability is our best guess. Let us for example take a sequence of zeros and ones where limit inferior and limit superior of the arithmetic means are 0 and respectively 1. This sequence will probably move in a zig-zag pattern and when the probability for the last 100 is high we would also expect that it is highly probable for the next number to be 1. Therefore, the probability of that sequence will be in the range *[0, 1]*, but this is not the best description of the random variable, because the last 100 results will give us a better description.

If we wanted to design a physical experiment which produces a precise probability *p*, we can do that easily. All we need is to construct a dice which yields a probability of *p*. We will have the random variable which describes the experiment and, on top of it, it will be the best possible description of the experiment.

How can we then design a physical experiment (i) which yields a probability in the interval *[0, 1]* and (ii) there is no better description of that experiment? Let us assume that the agent plays against a creature which says 1 or 0, but not randomly. The creature seeks to induce confusion in the agent to an extent where the agent is unable to tell whether the next number will be 0 or 1 and, to the worse, is unable to figure out the probability of the next number being 1. We will assume that the creature outsmarts the agent by a wide margin and is very successful in confusing the agent.

The above construct is not an accurate mathematical definition because no matter how smart you are there is always someone smarter. Thus, the creature depends on the agent. If the creature is able to read the agent's thoughts and figure out that the agent expects 1 with a probability in the interval $[a, b] \subset [0, 1]$, then the creature can elicit confusion by playing so as to create an appearance that the probability is outside that interval. Reading somebody else's thoughts may sound outlandish, but if the agent is an algorithm (computer program), then the creature can execute that program and find out what the agent's expectations are.

Thus, we have modeled an event the probability of which is in the interval $[0, 1]$. If we aim to design an event the probability of which is in the interval $[a, b]$, then we will construct two dices with probabilities a and b, and will give these dices to the creature. The creature will chose the dice to throw and still its aim will be to put the agent in confusion.

While the other authors suggest that a non-parameterized random variable has a precise probability (distribution), we will hold that the NPRV has a precise interval of probabilities (or distribution of intervals). Certainly, these probability intervals must satisfy the inequalities described in [5].

We would like to assume that the NPRV does not depend on the past nor of the future. Such NPRV is the dice game; the next number does not depend on the previous throws nor on the future throws. When this assumption comes into play, the order in which the experiments are made will not matter. Consequently, the sequence of results will be immune to permutations. That would be possible only in case we have a fixed probability. If the probability is not fixed, we cannot assume that dependency on the past and on the future does not exist. Accordingly, we will assume that NPRV may depend on the past and on the future, but the dependency is so complex that we are unable to comprehend it. Undetectable dependency is tantamount to non-existent dependency.

Model with randomness

We suggested that there is unpredictable randomness. We will now construct a second model of the world, which incorporates unpredictable randomness. First, the oriented graph will no longer be deterministic (in the MDP definition the graph is not deterministic, either). Second, the result of our observation in state (*s*) will not be a constant function, but a random variable (observations in POMDP are also random variables). Third, the set of incorrect moves in state *(s)* will also be a random variable. These three random elements will be determined by three oracles (α, β and χ). The oracles are parameterized random variables and when their parameters acquire a concrete value they become non-parameterized random variables (NPRV).

For each given state s_t and action a_{t+1} oracle α tells which the next state will be. His choices are to a large extent predetermined by graph *G*. Oracle α must choose one of the arrows available. In fact the oracle will have a choice only when the transition is nondeterministic.

For a given state s_{t+1} oracle β tells what will be the observation in the moment *t+1*. Here we wrote *t+1* instead of t, because all the three oracles use the same *Past* (a_{t+1} is the end of that *Past*). Oracle α speaks first and says which the next state will be, then oracles β and χ join to say what will be observed in that state (s_{t+1}) and which will be the incorrect moves.

Oracle χ tells whether event *e* occurred at moment *t*. We will use here χ for events like “a certain move is incorrect”. Later we will use χ to determine other events as well.

Here is the definition of the model with randomness:

S is the set of internal states of the world,

s_t is the current state of the world, and

G = <*S, R*> is a total oriented graph (nondeterministic).

$R \subseteq S \times A \times S$

$\alpha(Past, s_t, a_{t+1}, Future) \rightarrow s_{t+1}$

$\beta(Past, s_{t+1}) \rightarrow v_{t+1}$

$\chi(Past, s_{t+1}, e) \rightarrow \{true, false\}$

The idea behind our oracles is that the agent does not know which state the world is now in, what will be observed in that state, which the correct moves will be and which will be the next state. The agent does not, but the world knows everything. That is, the world has access to these oracles and is able to tell what exactly will happen. When we have a perfect model of the world (with an initial state), then we know how to define the oracles.

Let us take an arbitrary model *M* (with an initial state). We want to define the oracles for that arbitrary model. The question is whether the *M* model is a model of the world. Any model is a model of the world. Even a model which is not adequate (i.e. does not tell us anything) is still some model of the world.

Note: The definition will be based on a single life only. A different life may produce different oracles, but we have only one life.

From *Past* we will derive the step at which we are now (i.e. all we need to know is the length of *Past*). From here we can obtain the values of oracles β and χ for that step. Okay, but to which current state s_{t+1} do these values apply? We may say that they apply to any current state and this definition will be correct, but we prefer them to apply to a single current state (the state which gives us oracle α). So, once we define oracle α, we can obtain from it oracles β and χ.

When model *M* involves nondeterminism, we can define oracle α in various ways. Any such definition will produce some model of the world (although such model may not be adequate). Therefore, any model is determined by (i) an arbitrary oriented graph and (ii) an arbitrary oracle α. When we select a different oracle α, we will obtain a different trace and in practice a different model. Instead of obtaining the trace from oracle α, we can go the other way round. We can select the trace and look for an oracle α which aims to produce this trace. Let us only note that an oracle which produces that trace may not exist, so the oracle obtained will produce a trace that approximates the original trace as closely as possible.

Note: Here for each event *e* oracle χ returns a separate random variable. The definition gives the impression that these are independent random variables, but this may not always be the case. Two events may be related. For example, event *a* may occur only if event *b* has occurred. Another example is where *a* and *b* never occur at the same time. The observation can also be somehow related with the events. It might be appropriate to complicate the definition and merge oracles

β and χ in a more complex oracle, but we will not do so.

The oracles depend on the past. We will assume that *Past* is the full history. (We propose to use full history rather than plain history, because oracles depend on what has happened and not on what the agent has observed. The agent knows only the plain history, while the oracle knows the full history. We expect that the agent will learn how to figure out which moves are correct and in practice will also come to know the full history.)

We will assume that oracle α depends on the future as well. This may appear outlandish, but if we consider what is the cause and what is the sequel, the notion that the oracle sees the future is not that bizarre. Let us return to the postman and tramp story. It would be far-fetched to say that the oracle looked into the future, saw some proof of the postman's perpetration appearing in the future and then decided the murder to be committed by the postman. It would make more sense to say the murder was perpetrated by the postman and that is why evidence of the postman's evildoing has appeared in the future.

The oracle's dependence on the future will have some influence on the trace in the model. In Figure 2 we saw that the oracle's choice of arrows leads to a situation where at the next step there is no outbound *b*-event arrow from state 1. So the event b never occurred while we were in state 1. If the oracle's behavior was different, that would have been a *b*-event arrow from state 1.

We will suppose that the entire future (until the end of life or to infinity) is available to oracle α. Nevertheless, what we have in hand is not the entire future, but only the future until the current moment *t*. (The condition of the problem gives the history until that moment *t*.) If we need to know what oracle α will say at moment *t-k*, we have to split history in past (0 to *t-k*) and future (*t-k* to *t*). We must add that the oracle's decision usually depends only on the near future. In the murder example, if we fail to prove who the murderer is shortly after the murder, probably we will never resolve the murder at all. Once something is proven, there is no way to prove the opposite afterwards.

If we rely on the concept that a life can be lived only once, we can define oracle β as a function. Certainly, this function will be defined only for the histories which have played out in that life. As regards the other histories, we will have to think up the values of that function (define them howsoever). If we know the full history of that life, we would be also able to define oracle χ. If we have only the plain history, we will be able to define oracle χ partially. Oracle α can be defined in any way. We can take a random graph G and define oracle α by that graph. That would define oracle α in all cases with the exception of nondeterministic branches (where we will be at liberty to define the oracle as we wish). An example of a random graph is when we have only one internal state. Even that a graph is one possible model of the world (Figure 1 is such a model).

Note that if we have the same oriented graph, but different oracle α, the model will probably be different, because the trace in the model may be different. We will look for different traces in the model rather than for different oracles α. If two oracles α produce identical traces in the model, we will deem that these oracles are equivalent. Conversely, if we have the trace, we can reverse-engineer from it an oracle who induces that trace (if such an oracle ever exists, of course). Here is our reverse engineering process: from all possible choices of the oracle we will remove those which are not possible for the particular trace and for the given future. Then, if there are any choices left, we will select one of them (with the probability corresponding to the probability of this choice being possible with the given trace and future).

The trace

Our idea is that the oriented graph *G* describes the world somehow. Here it turns out that a random graph *G* can be a model of the world! The truth is that any graph can be a model, but very few graphs are adequate models of the world. When we are in some of the graph states, we expect something to happen. We expect that certain events will never occur in that state or vice versa – certain events must occur. An event may occur with a probability much higher or much lower than the expected probability for that event. If nothing happens in any state, then the model will be completely inadequate.

In our terminology, the occurrences during our journey along the arrows form the *trace* in the model. The perfect model gives us a very clear trace. In the perfect model, at each state there is a precisely defined observation and a precisely defined set of incorrect moves. In the abridged version of the perfect model some arrows will not be there. A missing arrow will mean that a certain event never occurred in a certain state (a certain move was never played). This is also a trace. The question is how to interpret it? Should we assume that this event will never happen in this state, or that it may happen albeit with a very limited probability.

An event may not necessarily depend only on your immediate observations or actions. It may also depend on the more distant history, for example on your observation at the previous step.

From where can we glean traces? From statistics. We will take the abridged model (that is, we will remove all arrows and states that have not been used until the present moment). We will count how many times each arrow has been used. For each state and each event we will count how many times the event occurred in that state. Of course we cannot count that for each and every event, because the events are infinitely many. We will only take count of certain more important events. From these statistics we can identify excursions from the expected value and these excursions will be the traces we are looking for.

Note that the statistical exercise is doable by the world, but a very challenging task for the agent. The world knows in which state it is and can count the arrows, while the agent can only make guesses. Even with in perfect model the agent may not know the state in which he has been (may not know the current state, but even if he knew that knowledge this does not determine the pervious states uniquely).

Sometimes the agent does not know his current state and can find that out only afterwards. That would not be a problem, because statistics will be compiled with time lag. But, it will be a problem if the agent never figures out the exact state in which he has been. Then the gathering of statistics becomes a very demanding task.

The exhaustive model

Let us assume that we have a model in which oracles α, β and χ do not depend on the past. This will be referred to as an *exhaustive model*, i.e. a model which cannot be improved any further. Everything that needs to be remembered about the past is already memorized in the current state. The model can be improved if we split one state in two and in a certain history we come to one of these states with a different probability. That is, the two states are distinguishable from the viewpoint of the past. In the exhaustive model, the two states will not be distinguishable from the viewpoint of the future, because the oracles are independent from the past. This renders such splitting of states meaningless.

There may be an exhaustive model with only one state. The world in this case is terrible and past occurrences are of no consequence for the future. It does not matter at all which action we choose in that world. Certainly, such a world is overly simplistic. We suggest that the worlds we are interested in are much more complicated and it is virtually impossible to establish perfect or exhaustive models of these worlds.

We will not attempt to achieve a perfect understanding of the world (establish a perfect model) or an understanding to the level of unsolvable randomness (establish an exhaustive model). We will try to establish a variety of multiple models which describe various features of the world and objects in the world. The Cartesian product of all these models will give us a model which probably will not be exhaustive either, but will describe the world reasonably well.

The comprehensiveness of the world is known as *Markov property*. In RL it is typically assumed that there is an exhaustive model of the world. Here we suggested something more. We suggested that there is a perfect model of the world.

However, here we will search for adequate models rather than for exhaustive models. Thus, we will look for models that have some trace. Trying to establish an exhaustive model is a superfluously ambitious problem and even an absolutely unsolvable problem in the case of more complex worlds.

Visible events

An event will be a Boolean function which at any point of time is true or false. First we will say what is a visible event: an event, which can be seen from the history (withal not from the whole history, but from its end).

Definition: A visible event will be the set of local histories. This event will be true when the history ends in some local history from the set of local histories.

An example of a visible event is our last move. An example of an event which is not visible is whether a move is correct. The correctness of the move can be seen from the full history, but cannot be seen from the plain history. This type of event will be termed *semi-visible*, because it contains in it (as a subset) one visible event. The visible event contained in it is: “This move is incorrect and we have already tried it”.

Another example of a semi-visible event is sunrise. You may well see sunrise but can also miss it if you overslept. The fact that you overslept sunrise does not mean there was no sunrise at all. There surely was, but you simply did not see it.

An example of an invisible event is: “I caught a cold”. You cannot eyewitness the occurrence of that event, but various signs afterwards will tell you that it has occurred (high body temperature, etc.).

So far the set of events which we used to find the model of the world was the set of our actions. Now we will generalize and use a random set of events.

The Event-Driven model

Now we will replace actions with events. We will take a set of events, and assume that the set is not big because if the events are far too many the model will become excessively complicated.

In the new model the arrows will no longer be labeled with actions, but with events. In the case of actions, two actions could not take place at the same time. When we have events however, it is perfectly possible that two events occur at the same time. That is why oracle α will not be defined by an action, but by events, and indeed not by a single event, but by the *events* (which is a set of events). When *events* is the empty set, then we do not go anywhere else and α returns the same state s_t. If in *events* there is only one event, the oracle will choose one of the possible arrows of that event. Where the *events* set contains two events, the oracle will have to decide whether to choose one of them (as if one event has obscured the other) or assume that the two events occurred one after the other. That is, traverse an arrow labeled with the first event and then another an arrow labeled with the second event. Moreover, the oracle should decide which event occurs first and which one occurs second.

S is the set of internal states of the world,

s_t is the current state of the world,

E is a set of events, and

G = <*S, R*> is a total oriented graph (nondeterministic).

$R \subseteq S \times E \times S$

$\alpha(Past, s_t, events, Future) \rightarrow s_{t+1}$

$\beta(Past, s_{t+1}) \rightarrow v_{t+1}$

$\chi(Past, s_{t+1}, e) \rightarrow \{true, false\}$

Besides the incorrect moves, here oracle χ determines all invisible events in E, and the invisible part of the semi-visible events. As regards visible events, they depend only on *Past* so the workings of the oracle are sufficiently clear. That is why it is superfluous for the oracle to recognize visible events. We do not want the oracle to recognize visible events, because we have an exhaustive model when oracles do not depend on *Past*. To avoid changing the exhaustive model definition, we will assume that χ recognizes only invisible events, while the visible events are known.

The event-driven model also lets us extract an abridged version with the arrows that have been used and count how times they have been used. Again, only the world can obtain an exact count, while the agent can only make a rough estimation of that count.

From those statistics we can derive the trace in the model and thereby judge the adequacy of the model.

The model with variables

It is perfectly reasonable to include variables in the model. In a certain state, one event may keep occurring for some time and then stop occurring. A variable is a convenient presentation of whether the event is occurring or not. That will be a local variable because it is pertinent to a certain state. Nothing prevents us from having global variables which are pertinent to multiple states.

An example of a model with variables was provided in [5]. There we had many doors (states) and each door was either locked or unlocked. Therefore, there was one variable attached to each door.

S is the set of internal states of the world,

Var is the set of variables,

s_t is the current state of the world,

$eval_t$ is the current evaluation of the variables, and

$G = <S, R>$ is a total oriented graph (nondeterministic).

$R \subseteq S \times E \times S$

$\alpha(Past, s_t, eval_t, events, Future) \rightarrow <s_{t+1}, eval_{t+1}>$

$\beta(Past, s_{t+1}, eval_{t+1}) \rightarrow v_{t+1}$

$\chi(Past, s_{t+1}, eval_{t+1}, e) \rightarrow \{true, false\}$

From a theoretical perspective, we have not altered anything here, because by including variables we only changed the number of states. This can also be seen above. Wherever we had a state before, we have a state and an evaluation of the variables now.

Although in theory the inclusion of variables does not change anything, in practice things change a lot, because if we have 10 Boolean variables, the number of states expands by a factor of 1024, which is a major increase. If we try to chart an oriented graph with that many states we will end up with something immensely complicated. In the end of the day, variables describe the trace (what happens in that state). The values of most variables will be completely unknown to us.

The Cartesian model

As mentioned already, we are looking for various models which describe various features of the world. Our model of the world, i.e. the model we have discovered, will consist of all these models. We can reckon that it is a Cartesian product of all these models.

In which state shall we be? It will be the current states of all these models and the current evaluations of the variables in these models. We can perceive the current state as a tuple of all these current states and current evaluations of variables.

You are now in your home city. The day is Monday. The time is 10 am. You are full because you had a steady breakfast. Each of these four sentences describes an event-driven model and the current state of that model.

At this moment you see the computer screen. This is an object (an event-driven model) and you can see it, so you are not in the *outside* state.

At this moment the door of your room is unlocked and the front door of the building is locked. So you have in your mind a model of the building and can tell whether two of the doors are locked or unlocked. Therefore, you know the values of two variables in this model.

We will assume that none of the models discovered is an exhaustive one, because if there is one exhaustive model, all other models are redundant. While a Cartesian product of multiple non-exhaustive models may return an exhaustive model, this is very unlikely if the world is sufficiently complicated.

Agents and objects

We should make a distinction between agents and objects because these are two different entities.

In previous articles [7] we referred to agents. He who changes something is an agent and we must predict his future actions to find out if he is friend or foe and try to come to terms with him.

What is man, an agent or an object? On the one hand, a man is an agent, because he changes the world. A man can move things around or even steal things. On the other hand, a man is an object, because we recognize (identify) him when we see him, hear him speaking on the phone or hear his name pronounced by somebody else.

Conclusion

In this article we discussed the relation between events and their sequels. (The sequel is what we call a *trace* in the event-driven model.) The question is, when does an event occur? For example, when does sunrise occur – when the sun peeks a little bit, when it is halfway up or when it is all up? Here we selected the moment in which we deem that the event has occurred and this is the moment when we understood that the event has occurred. It is by no means obligatory for the sequels of that event to commence exactly from that moment. When the sun rises, and even before it shows up, it sheds light all over and right away. The sun also sheds warmth, but much later, not right away. When we search for a trace we should be mindful that there may be sequels of the event or sequels of the fact that we come to a certain state of the model, and should not expect the sequel to emerge outright. We must allow for the sequel to play out slightly before or after. Moreover, the sequel (trace) will not be related to the states of the model only. We may as well see some sequels of the event proper. At sunrise for example the sky becomes red at about the time when the sun shows up. This is not related to the state before or after sunrise (i.e. to the night or to the day).

Therefore, when we gather statistics we should consider how close we are to the event and search for a trace (traits) both in the periods between events and around the events as such. The trace is a trait which occurs in a certain state of the model, but can also occur when we traverse some arrows in the model.

The trace will not consist exclusively of events. Besides events, we will have tests, and objects. Tests are special types of event, which we will highlight below. The appearance of an object is also an event and that event can help us define the trace in an event-driven model. That is, for defining an object we can use other objects. For example, in one room we see a cat and this is the room with the cat. This is how the cat object helps us define and remember a whole room. Of course the cat is an object which possesses the property *mobility*. We may have to search for the cat throughout the house. If the cat had the property *uniqueness*, once we find it in a room we will no longer search other rooms. The cat may not necessarily stay in one room, but in different rooms we stand different probabilities of seeing the cat. A different probability can also be a trace in the model (in this case the model is the house).

We saw so far that whatever events and whatever oriented graph we grab, it is always a model of the world. That bad news is that if the model is picked randomly, it will almost certainly turn out to be inadequate. In other words, it will have no trace whatever and nothing special (nothing interesting) will occur in its states.

In some cases (as in Figure 2) the model will have a trace, but that trace will only come from oracle α and we will be unable to use it (in this case to predict the next event) because we cannot know the state in which are.

It emerges therefore that the problem of discovering an adequate model of the world is very challenging. For this purpose we must search for specificities. We should keep observing various events and be alert enough to note when a specific combination of events occurs. For example, a black hat and untidy hair – that must be our colleague John.

Very few are the events which we can observe on a permanent basis. We will observe most events only from time to time. We call these events *tests* [5]. Tests are crucial for discovering adequate event-driven models.

The next article will be dedicated exactly to tests and how they can help us efficiently discover event-driven models.

Bibliography

[1] Richard Sutton, Andrew Barto (1998). Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA (1998).

[2] Richard Sutton (2008). Fourteen Declarative Principles of Experience-Oriented Intelligence. www.incompleteideas.net/RLAIcourse2009/principles2.pdf

[3] Johan Åström. (1965). Optimal control of Markov processes with incomplete state information. Journal of Mathematical Analysis and Applications. 10: 174–205.

[4] Apostolos Burnetas, Michael Katehakis. (1997). Optimal Adaptive Policies for Markov Decision Processes. Mathematics of Operations Research. 22 (1): 222.

[5] Dimiter Dobrev (2017). How does the AI understand what's going on. International Journal "Information Theories and Applications", Vol. 24, Number 4, 2017, pp.345-369.

[6] Dimiter Dobrev (2018). Minimal and Maximal models in Reinforcement Learning. August 2018. viXra:1808.0589.

[7] Dimiter Dobrev (2008). The Definition of AI in Terms of Multi Agent Systems. December 2008. arXiv:1210.0887.

Authors' Information

Dimiter Dobrev – *Institute of Mathematics and Informatics, Bulgarian Academy of Sciences. Acad. Georgi Bonchev Str., Block 8, 1113 Sofia, Bulgaria;*

e-mail: *d@dobrev.com*

Major Fields of Scientific Research: *Artificial Intelligence, Logic Programing*

CURRENT TRENDS AND FUTURE RESEARCH DIRECTION IN AUTOMATIC CRACKS DETECTION AND INTERPRETATION

Sara Ashraf, Islam Hegazy, Taha Elarif

Abstract. *The periodically checks and maintenance of structures is a critical step. Older than twenty years ago those checks was done manually, where specialist observe surface material and make analysis. However, the output of manual method always differs from one specialist to another. Thus, the manual method become unpractical, which motivates researchers to automate the check-up process. Digital image processing techniques for crack extraction are already widely implemented on large highway maintenance projects. In this paper, an overview is conducted to explain the automatic crack detection systems life cycle, concrete surfaces distress types, best-acting techniques is listed, and the performance measures. Followed by the challenges and open research points. Finally, present a proposed future work on this field. The challenges still opened on this field, beginning by selecting the right portable device for automatic inspection to give a high quality image for processing, can inspect on hard weather and can access danger places. Then, at pre-processing step, there exist many image noises have to be neglected to output accurate cracks; also, adapting lights is crucial, because on dark images, the cracks become unclear to be distinguished from background surface. Then, the detection process is depending directly on the previous pre-processing step, some algorithms still needs many trails for parameters adaptation. However, the Error-nous rates is decremented now a day with the improvement of crack detection algorithms. Later, the crack characterization with severity identification still needs more research attention to full automate systems and make decisions without any human intervention.*

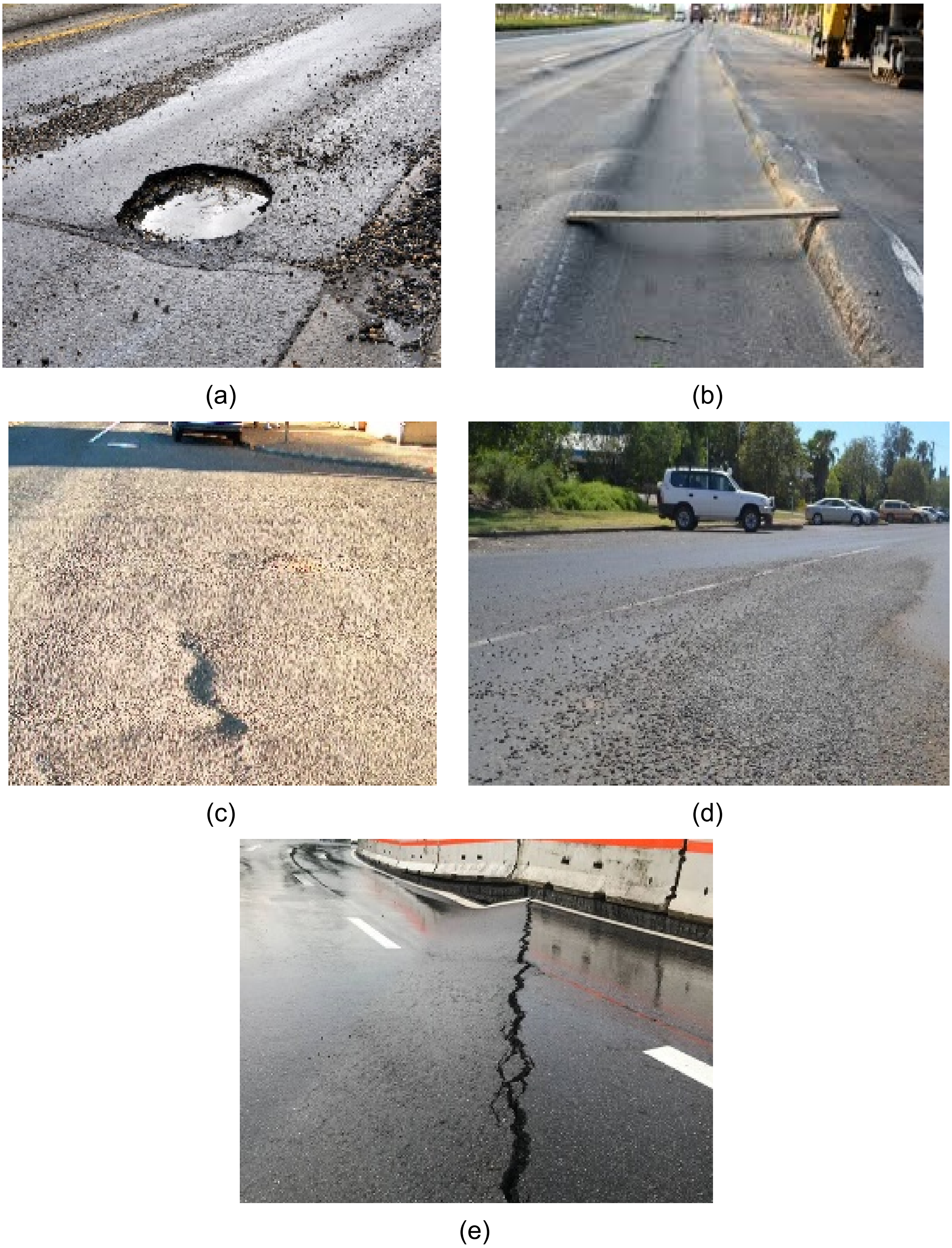
Keywords: *Crack Detection, Image processing, Crack Interpretation, Concrete Surface Distresses.*

ITHEA Keywords: *J. Computer Applications, J.6 COMPUTER-AIDED ENGINEERING.*

Introduction

The Concrete structures such as roads and bridges, affected daily by heavy loads and other environmental conditions. These conditions leave different types of defects over the surface. Figure1 show some of these defects. Figure1 (a) Potholes are shown as a form of holes on surface. While, (b) rutting is form as longitudinal deformation of a surface; also, (c) shoving is the opposite form of rutting, moving the material upper than the normal level; and (d) the bleeding is a loss of surface material, finally, (e) Cracks which has many forms, as if longitudinal cracks, fatigue cracks, block cracks, thermal and edge cracks. These cracks to be discussed on the next section in detail with its causes, and the applied fixes by engineers.

The history of crack detection is old, where the process was done manually by specialist engineer, starting the process by scanning the surface using eye observation, then write analysis notes and then decide if the surface contain cracks or not, also, provides severity status. This procedure has many drawbacks, as it needs a lot of time to scan, analyse and then output decision result. In addition, the decision is inaccurate as it depends on specialist experience. So, the manual way become unrealistic and cannot dependable on it to finish the check-up process over the huge number of constructed concrete surfaces. The drawbacks of manual method become a motivation to build automatic methods using digital image-processing algorithms. The first generation systems called semi-automatic crack detection systems. These systems can scan surface automatically and output results using computer programs, but it needs a specialist intervention during the detection process to improve the precision. Thus, although the process of detection became faster than manual way, but still needs a specialist help which slow down the process. Therefore, the researches was grownup on last ten years to build fully automatic systems. The goal of these systems to be dependable systems and give accurate results, and be faster than older traditional methods.



(a)

(b)

(c)

(d)

(e)

Figure 1. Different types of surface defects, (a) potholes, (b) rutting, (c) shoving, (d) bleeding, (e) cracking

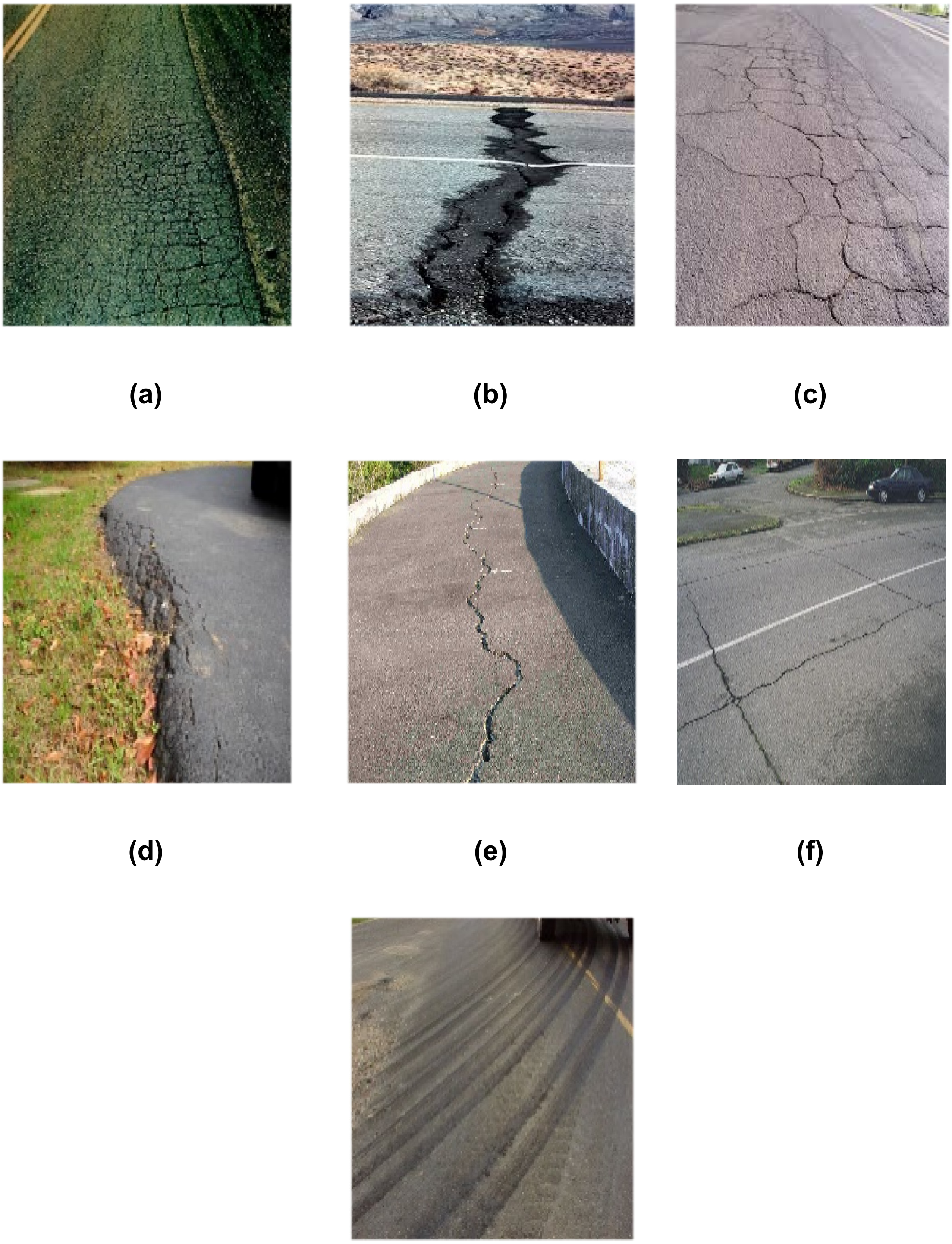
The safety on roads and bridges is highly necessarily, the [US Department of Transportation, 2014] conducted a detailed survey on all road defects with its severity, this study shows a crack as one of the major issues that can affect concrete surfaces. A crack is a defect on concrete surfaces generated due to tensions, heavy load, Physical and environmental conditions. The early detection and fixes of those cracks will save people lives, and decrease accidents.

This study presents an overview of newly automatic crack detection systems. The sub-sections on paper is beginning by discussing different types of concrete surface distresses, then present a lifecycle in order to build any automatic crack detection system, and then moving through each step, Discussing the input, output and best acting techniques per step. Also, presents a suggested research points and challenges. Finally, present a proposed future work.

1. Concrete surfaces distress types

The cracks was studied before by civil engineers, and classified using standard characteristics, like morphological attributes and the severity levels, as mentioned by [Lein Some, 2016] and [US Department of Transportation, 2014] .

Figure 2 show different types of cracks, Figure2 (a) called Fatigue Cracking Caused by heavy loads, end of pavement life or insufficient strengths, fixed by remove and replace procedure, Mill the surface, or overlay. (b) Called Thermal Cracking (Transverse) usually seen these kind of cracking on roads and bridges, caused by shrinking and expansion of concrete material due to changing temperatures, fixed by thick overlay, Crack filling or full-depth reclamation. (c) A Block cracking Thermal cracking can cause Block cracking. In addition, high void contents loaded on the surface. Fixed by Filling then Sealing, Seal Coat, Full-Depth Reclamation, and Thick Overlay. (d) Called Edge Cracking Caused by Soil Movement under pavement surface. Fixed by Crack Filling then Sealing or Full-Depth Reclamation. (e) Called Longitudinal cracks mostly seen on low temperature countries. Can fix it by crack sealing or thick overlay. (f) Reflective Cracking generated by cracks on underlying layers of surface, reflected on upper surface layers. Fixed by Crack Filling, Seal and Saw, and Seal Interlayer. Slippage cracking also called U-shaped cracks. Always exist on U-turns, parking areas. Caused by wheel movement in a form of U shape, Repaired by crack filling then sealing, and full-depth reclamation.



(a)

(b)

(c)

(d)

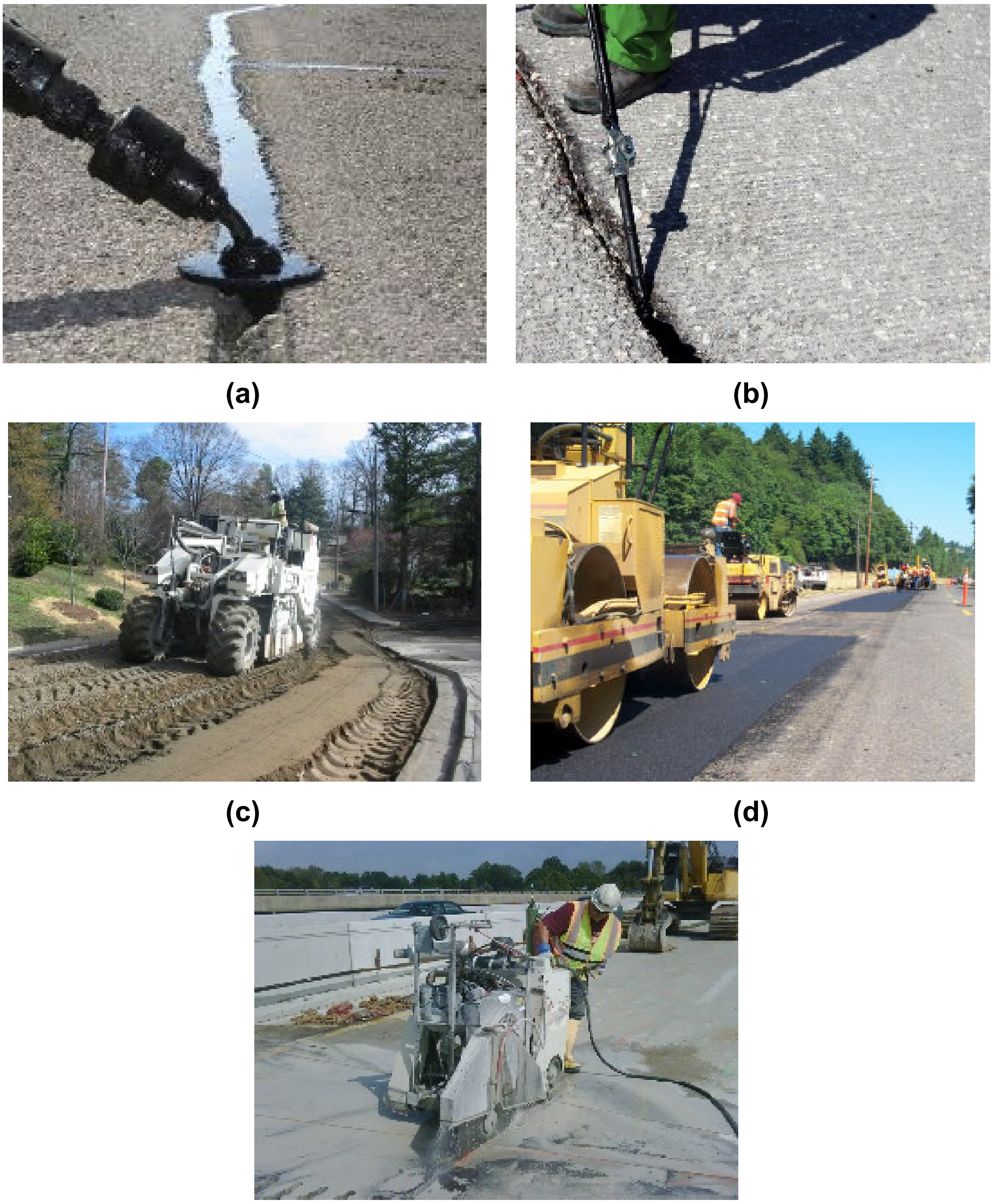
(e)

(f)

(g)

Figure 2. Different crack types

Figure3 show some of crack fixes procedures used by road engineers. Like crack filling, sealing, full-reclamation, thick overlay, saw and seal.



(a)

(b)

(c)

(d)

(e)

Figure 3. Engineering Crack fixing procedures, (a) Crack sealing. (b) Crack filling. (c) Full-depth reclamation. (d) Thick overlay. (e) Saw and sew.

2. Automatic Cracks detection systems life cycle

Automatic crack detection systems have a standard development steps. The life cycle as mentioned by [A. Mohan & S. Poobal, 2017] and shown on Figure 4 , The first step is Image Acquisition; this step is aimed to collect sample photos of concrete surface, using photo capture devices like cameras, robots, etc. Then the next step is image pre-processing step, it takes the captured photo and apply one or more refining image algorithms, these algorithms can remove noises, adapt lights, adapt the contrast, or sharpen the edges, etc. the output of pre-processing will be ready for processing. The processing step is a set of subsequent steps that takes input pre-processed image and output the highlighted crack if exist on image. After image has been processed, the interpretation step comes to define the characteristics of outputted cracks, like width, height and severity levels. Finally, as any system should evaluate it using performance measures, to insure the reliability, usability and calculate error rates. In next sections from section 4 to section 7, the development life cycle steps is discussed with best-implemented techniques.

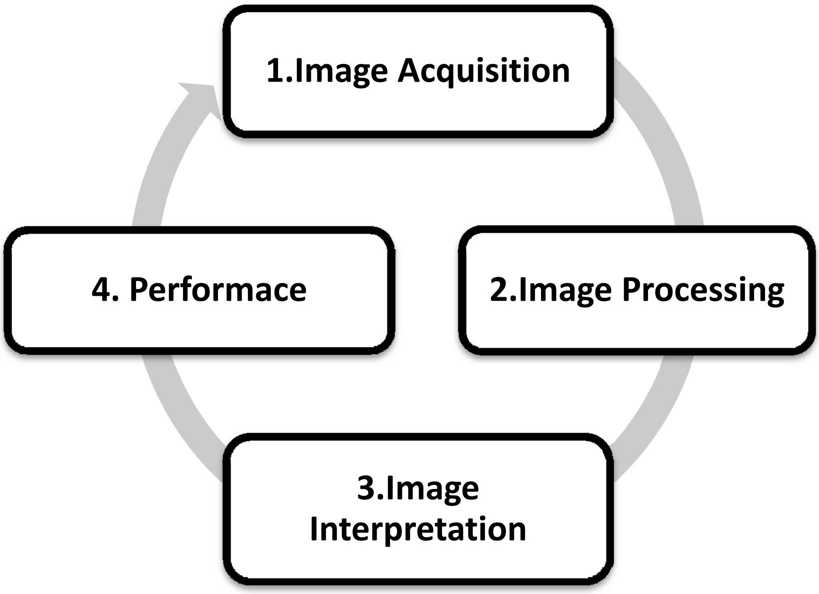


Figure 4. Crack detection life cycle

3. Image Acquisition

The first step, collect Figures of the target concrete structure. Using photo capture devices, like mobile cameras, High definition cameras, robots, UAV's etc. This step is very critical step. Because the final decision depending totally on camera's observations. If camera output unclear or noisy image, this will affect directly the output of detection algorithm and may give wrong results. Thus, selecting the right device is highly recommended to output the desired results. Some systems observe by equipped vans that contain sensitive sensors, easy to use but output noisy images. Some systems uses robot cars supported by machine vision system, it is fully automatic but expensive and complex to use. In addition, UAV (unmanned aerial vehicle) it is a tinny airplane supported by 3-D sensors, it is fast and low cost compared to robots, but its maintenance is costly, and hardly work on bad weather. In addition, quadcopters un-manned aerial vehicle airplane without pilot, can access critical ways like under bridges, and work on bad weathers, but it is heavy and not portable. Now a days with the improvement of cameras and mobile cameras, can depend on it as easiest and cheaper devices.

4. Image Processing

The second step after collecting sample images is Image processing. A series of image processing algorithms starting by pre-processing algorithms that enhance the image, then segmentation step that distinguish foreground from background then comes the feature extraction step, then selecting the features that define cracks, then detecting cracks and highlighting them and ignore other objects. Finally, classification step that minimizes the number of cracks set after analysis to get the truly cracks only not other objects. Figure 5 shows the image processing inner steps.

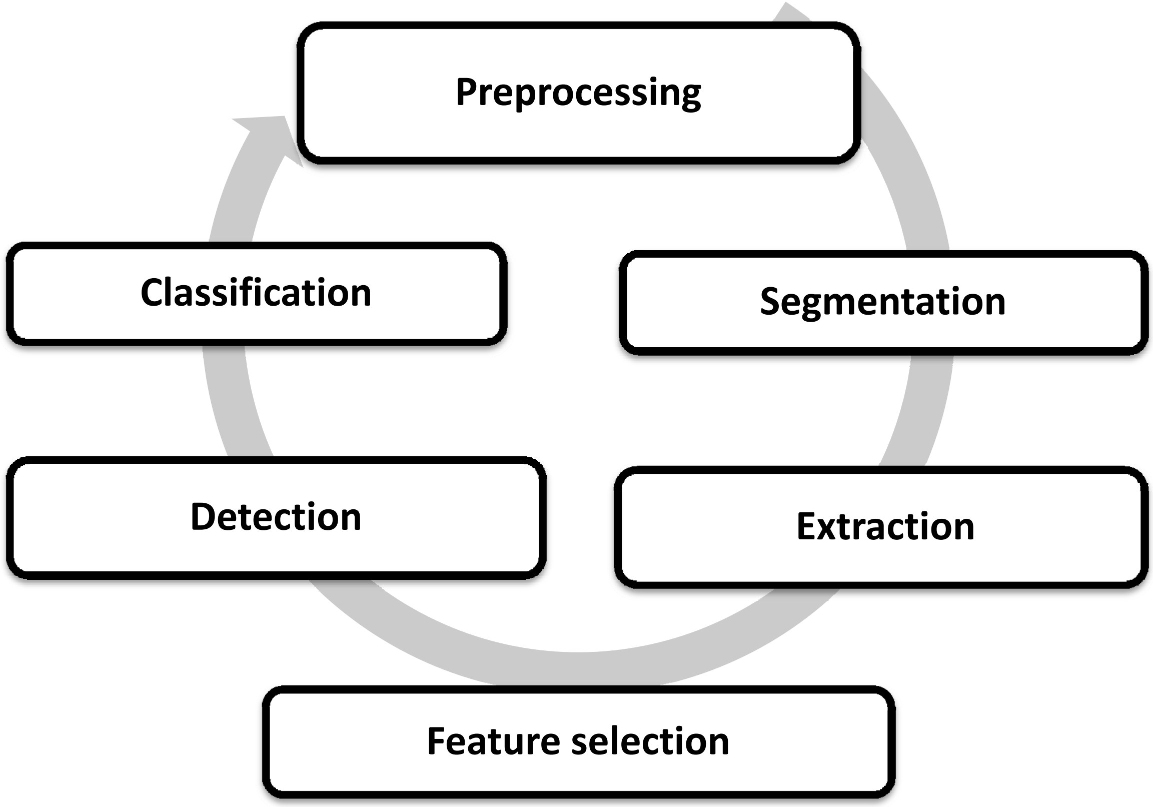


Figure 5. Architecture of Image processing phase in crack detection systems

4.1 Pre-processing Step

The preparation step of the sample images, Applying different filtration techniques to remove noise and adapt lighting also remove non-crack shapes. Common images enhancement methodologies discussed by [H. Zakeri, 2017]. First, Point operations methods are looping on each pixel, and apply specific conditions like thresholding method, point operation methods is easy to apply, but it needs many trails to adapt algorithms parameters to get optimal results. Second, Spatial operation methods looping on group of pixels using masking technique, like low pass filter, median filter, spatial operation methods also needs a lot and try and fail trails in order to reach the optimality. Third, Transformation Operation that transform images then apply some image enhancement operations then apply inverse transformation, like generalized linear filtering method. Finally, the Hybrid methods merge two or multiple techniques; Approved that it gives better results rather than other methods. Some of existing Hybrid algorithms are shown on Table.1, to prepare the image to segmentation step.

Table.1 Prepressing algorithms

Author	Algorithm	Experimental results
[Yao M, 2015]	Create a new imaging system to refine image before processing.	Not need to any artificial lighting.
[Zhou H, 2010]	Create an enhancement technique, called an Illumination Invariant algorithm.	Good results are shown on irregularities of the concrete surfaces.
[Jiang J, 2015]	Divide pixels into weak and strong pixels using Electro-magnetism.	Give extra good results on next segmentation step.

4.2 Segmentation Step

Segmentation step distinguishes the foreground from the background. [Tsai YC, 2010] conducted a comparative study of these methodologies and compared the six segmentation methodologies as shown on Figure 6, the threshold based segmentation (TBS) is very famous ,known and easy to implement, segment the background from foreground. While, The other segmentation methods, like Clustering based segmentation (CBS) that group the elements by common features, Fuzzy Based Image Segmentation (FBS) is probability based and Matching based segmentation (MBS) which compare input features with given model features then classify.

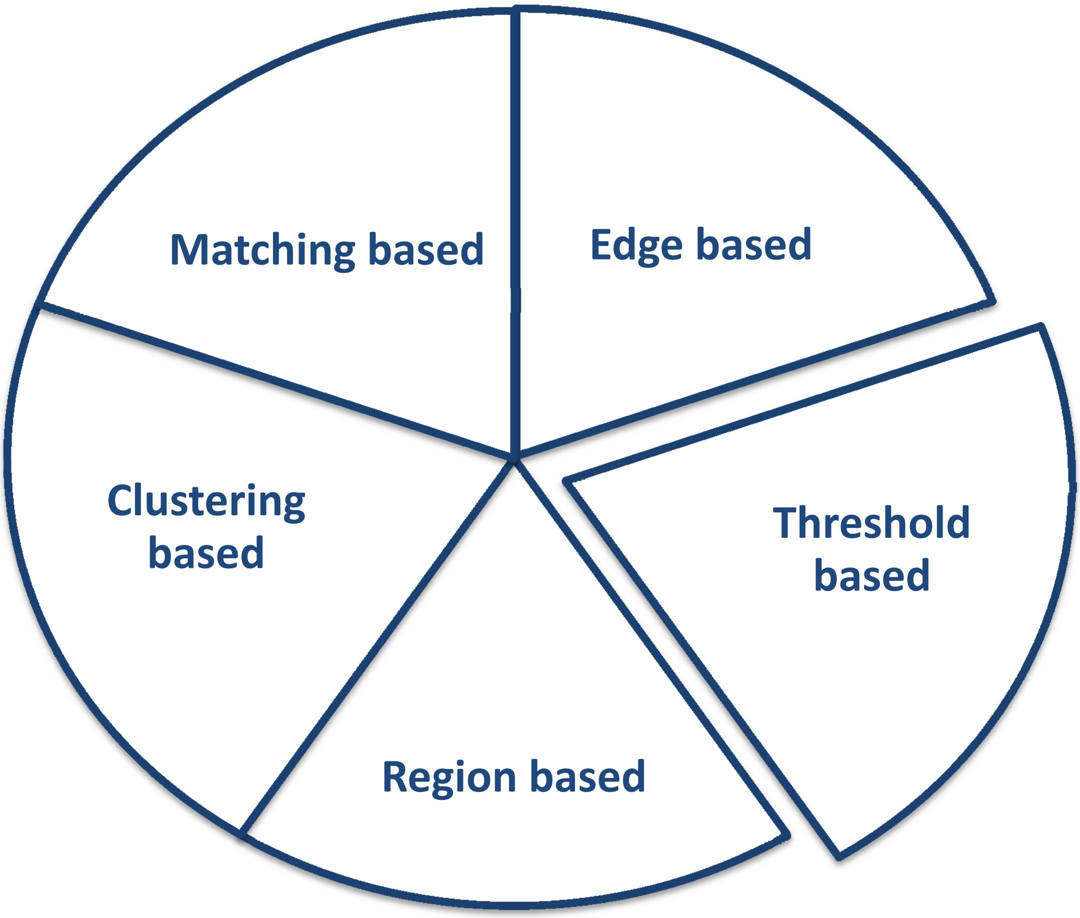


Figure 6. Segmentation methodologies

The techniques based on last segmentation methodologies, are shown in Table.2, The best of these techniques, is the threshold-based segmentation.

Table 2. Some of best segmentation techniques

Author	Method	Description	Results
[Ayenu-Prah A & Attoh-Okine N, 2008]	EBS	Used BI dimensional empirical mode decomposition method for crack evaluation. Then applied the Sobel edge	Better performance than canny edge detector.

		detector.	
[Mertz C, 2014]	RBS	Used Texture based features to segment cracks from the surface.	Good Accuracy and time.
[Lokeshwor H & Das LK, Goel S, 2013]	TBS	Used natural lighting then apply adaptive thresholding technique.	The accuracy is up to 96 %. Also, fast.
[Miss Hetal, 2013]	TBS	Otsu algorithms used in thresholding. It is a kind of global thresholding techniques.	Assumed as better threshold selection method.
[T. Romen Singh, 2011]	TBS	Create local adaptive thresholding technique.	Excellent quality and speed.

4.3 Feature Extraction

Feature extraction step is the third step after segmentation, that looping on each group from segmentation step and try to get common features. Figure 7 show some of feature extraction methods.

Histogram feature extraction, is a graph representation of features like moments, variance and mean value, it is easy to implement and maintain. While, Spatial Feature extraction method, uses spatial distribution and joint probability distribution. The crack represented by amplitude of Radon transform, this method is easy to implement and can detect crack severity level. Also, Feature transformation method; extract hidden information from image, like High Frequency Energy Percentage and High amplitude wavelet coefficient, this method is also simple to implement. Moreover, an Edge detection method, like zero crossing, gradient operators, Laplace operators, compass operators and stochastic gradient, the most practical systems uses an edge detection method but take on consideration its sensitivity to noise.

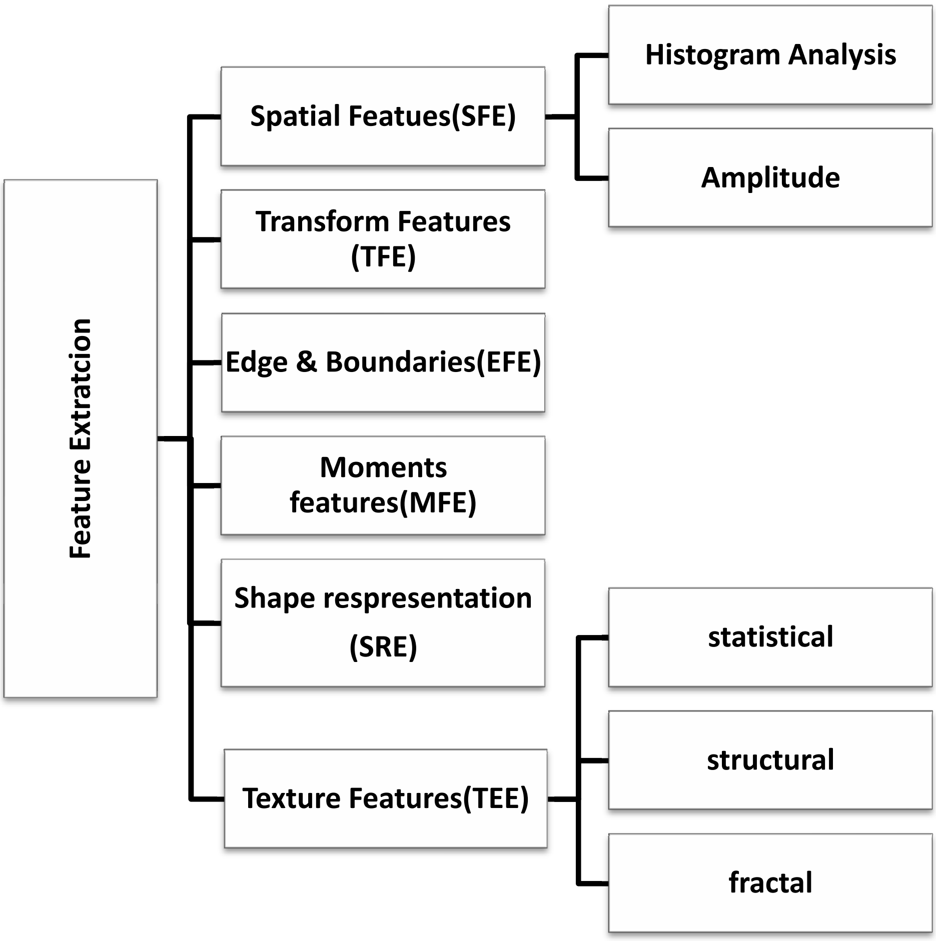


Figure 7. Feature Extraction Methods

[H. Zakeri, 2017] lists the feature extraction techniques. Some of better techniques are shown in Table.3. From analysis found that The Edge feature extraction method, was the most widely used.[Abdel-Qader, 2003] made a comparative study between multiple edge detection algorithms, Fast Haar Transform, Sobel, Fast Fourier Transform and Canny, and found that Fast Haar Transform is the most reliable one.

Table 3. Feature extraction techniques

Technique/Author	Description	Experimental Results
[Huili Z, 2010]	Used Mallat wavelet transformation, to improve canny edge detection algorithm.	Automatically adapting parameters, to detect noisy edges.
[Changxia M, 2009]	Create fractional differential and wavelet transformation method.	Compared with Soble, Prewitt and Laplacian of Gaussian then show the best performance, work on noisy images.
[Lokeshwor et al, 2013]	Using video inspection, and depending on canny algorithm.	A method accuracy is up to 96 %.
Hough Transformation	Work on curves not just straight lines.	Accurate detection results

4.4 Feature Selection

The fourth step after extracting features is selecting the crack features Figure 8. The feature selection methods classified to supervised, non-supervised, and semi-supervised. Due to the lack of labeling, unsupervised methods are mostly used than supervised methods.

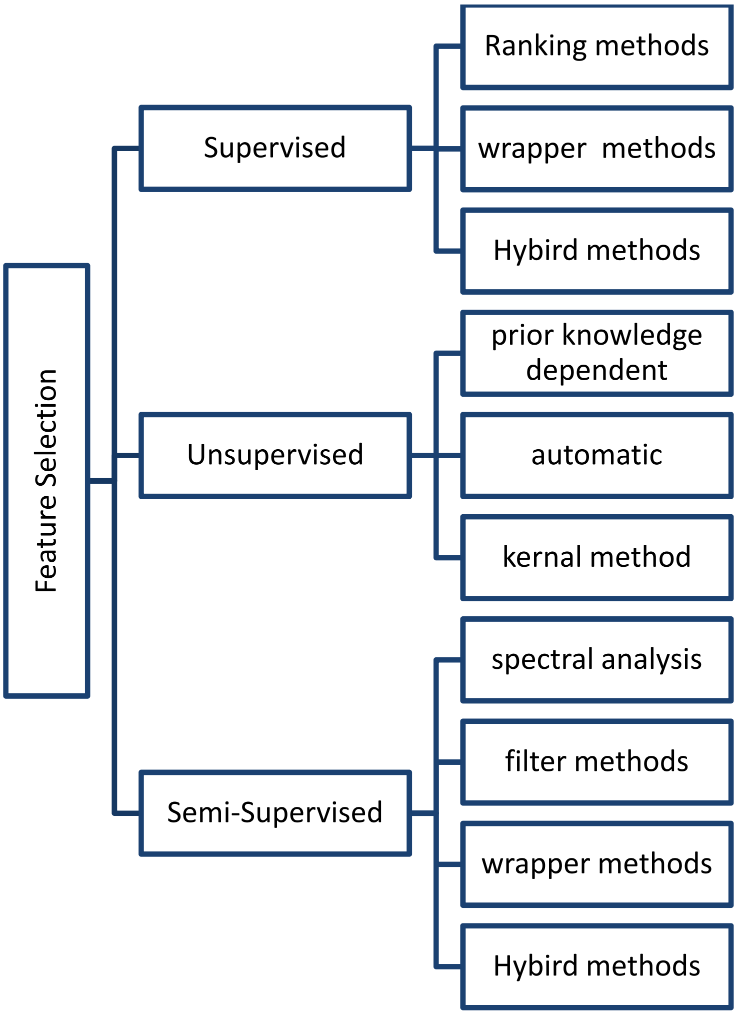


Figure 8. Feature selection approaches and Methods

Tables 4,5,6,7 show the methods used on supervised, Hybrid, non-supervised, semi-supervised respectively.

Table.4 Supervised feature selection Methods

Author	**Description**	**Results**
[Jin et al, 2015]	Used non-linear attributes and factoring.	The performance decreased with large image sizes.

[Lee et al & Kim et al, 2005]	Developed the Mutual Information algorithm.	Good performance.

Table.5 Hybrid ***approaches*** feature selection Methods

Author	Description	Results
[Zhang et al, Hu et al, 2005]	Use an ant-colony optimization technique.	Need huge training for learning.
[Foithong et al, 2012]	Use a wrapper approach and the mutual feature criterion.	Good performance and give multiple features.

Table.6. Unsupervised feature selection Methods

Author	Description	Results
[Maldonado et al, 2015]	Developed Kernel K-means, selecting the most relevant features, and then scaling factors.	All Un supervised approaches has low performance and accuracy.

Table.7 Semi Supervised feature selection Methods

Authors	Description	Results
[Gavila´n et al, 2011]	Using Ada Boost algorithm for selecting Gabor features.	Should be trained on large dataset for enhancement.

[Cong et al, 2013]	Used Backward Selection, Forward Selection (FS), Principal Component Analysis (PCA) and Genetic Algorithm, for road distress, feature selection.	(FS) is the best when the number of features is between 2 and 6. (PCA) is the best method when the number of features is greater than 5.

4.5 Detection

Detection step decides if this sample image containing crack or not Figure 9. Crack detection algorithms Show the classifications of crack detection methods at supervised, unsupervised and semi-supervised systems. Crack detection methods like Statistical Methods, liked adaptive thresholding, histogram analysis, Gaussian modelling. These methods are easy but not good for crack geometrical analyzation.

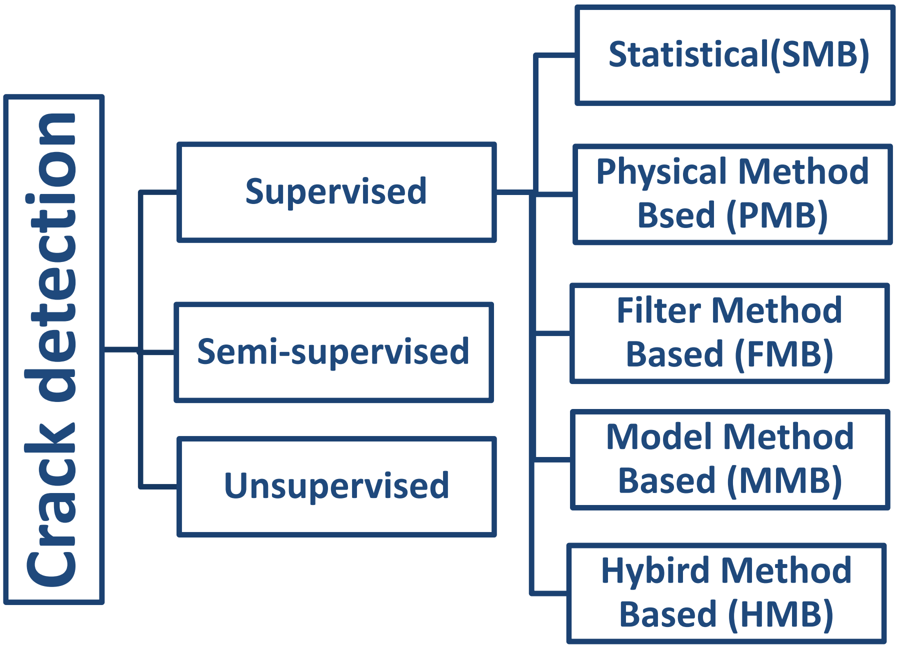


Figure 9. Crack detection algorithms

Another Method is The Physical Method Based (PMB) Used morphological tools or edge detection. Crack-Tree and multi-direction morphological structuring features are examples these methods. However, it assumes fixed width and depth for cracking and it is not reliable. In addition, the Filtering Method Based (FMB) Provide a variety scale modelling. Like, contour let transform, wavelet based method, shear let transform. In addition, the Model- Method Based (MMB) conducted some assumptions about the geometrics of crack. Like texture decomposition, Markova modelling, pattern based. Finally, The Hybrid Method Based (HMB) using two or more methods to make a better paradigm for detection and classification. Table.8 show some of the best-implemented detection techniques.

Table.8 Crack detection techniques

Author	Description	Results
[Changxia et al, 2009]	Depends on Wavelet transform to neglect noise, Then fractional differential to refine frequencies.	Can detect cracks even on noisy images.
[Gavila´n et al, 2011]	Depends on seed based algorithm, Detection done by computing the minimum cost paths, and adapt parameters.	Work without any human intervention, after setting optimal parameters.
[B.Santhi et al, 2012]	First, Extracting morphological features, then filter by Butterworth, then use canny edge detection algorithm.	Can detect hairline cracks.
[Miraliakbari et al, 2014]	Used Line filtering algorithm.	Detect cracks on multiple directions.

According to [Alvand et al, 2016] after evaluating techniques they concluded that, supervised and unsupervised techniques were the mostly used approaches for detection, whereas the semi supervised techniques not commonly used. Finally and after detection can measure Completeness by apply ratio on (1).

$$\text{Completeness} = \frac{Correctly\ detected\ cracks}{Correctly\ detected\ cracks + Non\ Correctly\ detected\ cracks} \qquad (1)$$

4.6 Classification

Classification step distinguishes the differences between crack and non-crack regions. In addition, determine types of distress, and severities. Table.9 shows many techniques used to achieve that.

Table.9 Classification Techniques

Technique/Author	Description	Result
[Lee et al & Kim et al, 2005]	Apply three neural network algorithms, Histogram-based, image based, proximity based neural network.	The proximity-based algorithm can classify multiple crack types. The accuracy of about 96%.
two-step unsupervised pattern recognition method	Test multiple unsupervised clustering algorithms, like k-means, hybrid and hierarchical. Then output crack types like longitudinal, Transversal and miscellaneous cracks.	The performance concluded as recall = 95.5 %, F-measure = 93.5 %, and the error rate = 0.6 %.

[Avila et al, 2014]	Calculating the shortest minimal path per pixel.	Can isolate cracks of thin width 2mm.
[Ouyang et al, 2013]	Developed the Beam let algorithm that detect crack images, then based on crack direction identify the crack type.	Detect transversal cracks, longitudinal cracks with accuracy 100 %, and block crack and alligator cracks by above 85 %.

5. Image Interpretation

Used to detect the severity of crack and the level of visual cracking, rarely working on it on research study. Table.10 Interpretation TechniquesTable.10 focuses on existing Interpretation techniques.

Table.10 Interpretation Techniques

Author	Description	Results
[Zhou H, 2010]	Developed three statistical methods for crack detection, Include HAWCP, HFEP, STD based on wavelet analysis.	The methods is effective detecting and counting the number of cracks.

6. Performance Measures

The final step before launching cracking detection system, the performance measures is calculated to conclude if the system is reliable or not. [Tom et al, 2004] was published the performance measures manual, measures accuracy , F-measure , Recall , true positives rate, true negatives rate which is very important to decide if the proposed system is able to behave correctly with good performance or not.

7. Current work analysis

The analysis of any system used to detect cracks will based on these standard parameters. The analysis can be Data set based, Objective based analysis, Error level based, Accuracy level based, Image processing techniques based or IP-based crack detection for safety monitoring. Each kind of these analysis phases has been discussed by [A.Mohan & S.Poobal, 2017] show statistical analysis of some techniques until 2017.

8. Proposed research direction

After discussing a full life cycle for all automatic crack detection systems on Figure 10, and list, the techniques for each phase on this life cycle, we find as a result that the area of image interpretation need more investigations and improvement to get more accurate results. The proposed system will accept input dataset of concrete structures like roads and bridges and the output will be crack positions with its characteristics like width and height. That helps expertise to take fastest decisions.

9. Conclusion

In this paper, we conducting comparisons between a couple of methods on the field of Automatic crack detection. After comparisons and evaluations of crack detection techniques, we conclude that, at pre-processing the hybrid approaches is work better than individual methods for noise removal and irregularities of illumination. While on segmentation step, the Threshold based systems (TBS) show best results but needs many trails to set the thresholding parameter. For feature extraction the Edge detection Methods give better results, the canny edge detection is the most used algorithm during extraction because of high accuracy. At selection step, the most researches works on supervised methods and it gives good results, second comes the hybrid methods which gives also good results with high performance and speed, while, unsupervised methods is limited in feature selection researches, which make it a good point for further researches. At detection step, the Filtering Model Based technique can be considered as the best technique. Either working alone or hybrid with other methods. The Filtering technique can detect cracks with multiple directions, and can detect hairline cracks. Finally yet importantly, the classification is best done by k-means algorithm, that successfully categorize all kinds of cracks, longitudinal, miscellaneous and Transversal cracks respectively, which make it better on decision making. After Detection process was done, need to interpret the crack characteristics, the best results come from wavelet analysis techniques that detect and count cracks, the crack interpretation is a suggested point of research in order to get more details about the detected crack, the width, height and the deep of crack. This information will lead to calculate severity levels and assist experts.

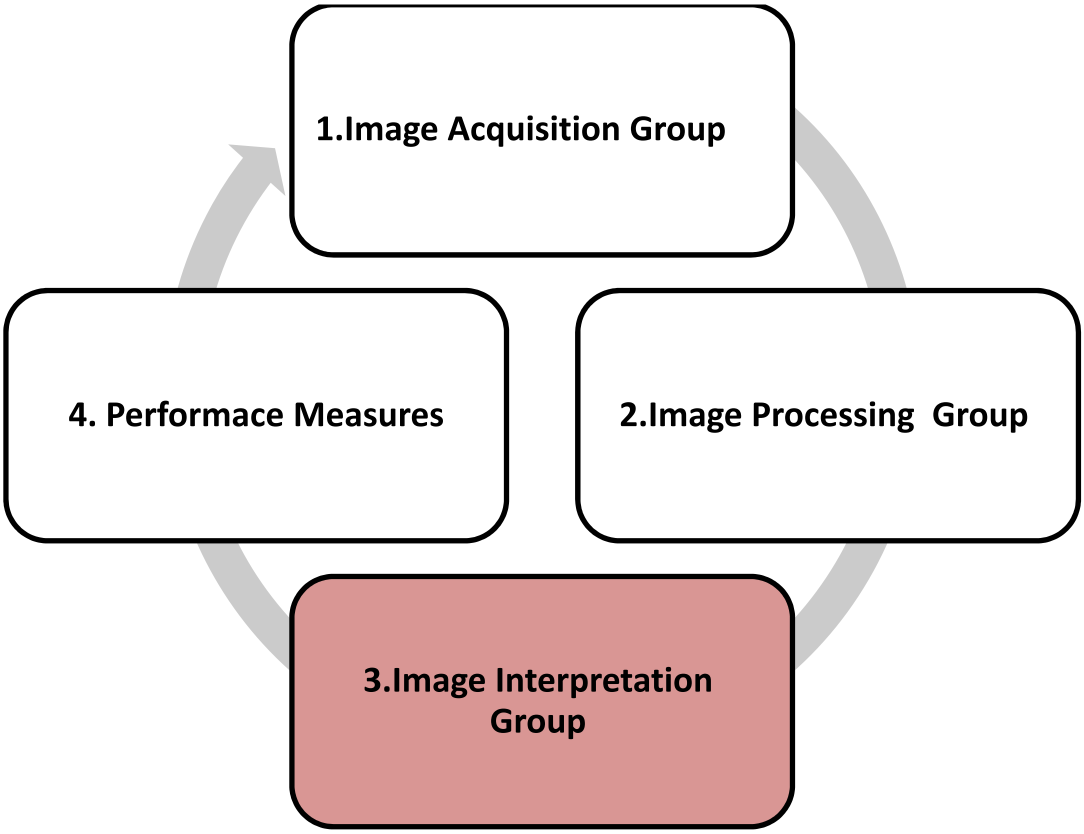


Figure 10. Cracks detection systems Life cycle, step 3. Image Interpretation the area of proposed system

Acknowledgment

This paper is published with partial support by the ITHEA ISS (www.ithea.org) and the ADUIS (www.aduis.com.ua).

Bibliography

[A.Mohan & S. Poobal, 2017] A. Mohan, S. Poobal, Crack detection using image processing: A critical review and analysis, In Proceedings of Alexandria Engineering Journal, 2017.

[Abdel-Qader, 2003] Abdel-Qader I,Kelly ME, , Abudayyeh O,Analysis of edge-detection techniques for crack identification in bridges, submitted by J Comput Civ Eng,2003, pp. 17(4):255–263.

[Alvand et al, 2016] Alvand Miraliakbari, Sokunthet Sok, Michael Hahn, Comparative Evaluation of Pavement Crack Detection Using Kernel-Based Techniques, In The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2016 Volume XLI-B1.

[Avila et al, 2014] Avila M, Begot S, Duculty F, Nguyen TS. 2D image based road pavement crack detection by calculating minimal paths and dynamic programming. In: 2014 IEEE international conference on image processing, ICIP 2014.

[Ayenu-Prah A & Attoh-Okine N, 2008] Ayenu-Prah A, Attoh-Okine N, Evaluating pavement cracks with bidimensional empirical mode decomposition, In EURASIP J Adv Signal Process Journal, 2008, pp.(1):861701.

[B.Santhi et al, 2012] B.Santhi, G.Krishnamurthy, S.Siddharth and P.K.Ramakrishnan, Automatic Detection of Cracks in Pavements Using Edge Detection Operator", In Journal of Theoretical and Applied Information Technology, February 2012, pp. Vol. 36 No.2.

[Changxia et al, 2009] Changxia M, Wenming W, Chunxia Z, Feng D, Zhengli Z. Pavement cracks detection based on FDWT. In: International conference on computational intelligence and software engineering, 2009. CiSE 2009.

[Changxia M, 2009] Changxia M, Wenming W, Chunxia Z, Feng D, Zhengli Z, Pavement cracks detection based on FDWT. In: International conference on computational intelligence and software engineering, 2009. CiSE 2009.

[Cong et al, 2013] Cong F, Hautakangas H, Nieminen J, Mazhelis O, Perttunen M, Riekki J, Ristaniemi T. Applying wavelet packet decomposition and one-class support vector machine on vehicle acceleration traces for road anomaly detection. Advances in neural networks–ISNN 2013. Springer, New York, pp 291–299.

[Foithong et al, 2012] Foithong S, Pinngern O, Attachoo B. Feature subset selection wrapper based on mutual information and rough sets. In Exp Syst Appl journal, 2012, pp. 39(1):574–584.

[Gavila´n et al, 2011] Gavila´n M, Balcones D, Marcos O, Llorca DF, Sotelo MA, Parral, Ocan˜a M, Aliseda P, Yarza P, Amı´rola A. Adaptive road crack detection system by pavement classification. In Sensors, 2011, pp.11 (10):9628–9657.

[H. Zakeri, 2017] H. Zakeri, Fereidoon Moghadas Nejad, Ahmad Fahimifar, Image Based Techniques for Crack Detection, Classification and Quantification in Asphalt Pavement: A Review, In Proceedings of Arch Computat Methods Eng., 2017, pp.935–977.

[Huili Z, 2010] Huili Z, Guofeng Q, Xingjian W , Improvement of canny algorithm based on pavement edge detection. In: 3rd International congress on image and signal processing (CISP), 2010.

[Jiang J, 2015] Liu H, Ye H, Feng F, Crack enhancement algorithm based on improved EM. In J Inf Comput Sci Journal, 2015, 12(3):1037–1043.

[Jin et al, 2015] Jin C, Liu J, Guo J .A hybrid model based on mutual information and support vector machine for automatic image annotation. Submitted on Artificial intelligence perspectives and applications. Springer, New York, pp 29–38.

[Lee et al & Kim et al, 2005] Lee J, Kim D-W. Mutual information-based multi-label feature selection using interaction information. Submitted on Exp Syst Appl journal, 2015, pp.42(4):2013–2025.

[Lein Some, 2016] Lein Some, Automatic image-based road crack detection methods. In Degree, Project in the Built Environment, Second Cycle, 30 Credits Stockholm, Sweden, 2016, pp. 1653-5227.

[Lokeshwor et al, 2013] Lokeshwor H, Das LK, Goel S, Robust method for automated segmentation of frames with/without distress from road surface video clips. In J Transp Eng Journal,pp. 140(1):31–41.

[Maldonado et al, 2015] Maldonado S, Carrizosa E.Weber R.Kernel Penalized K-means: a feature selection method based on Kernel K-means, In Inf Sci journal, 2015, pp.322:150–160.

[Mertz C, 2014] Mertz C, Varadharajan S, Jose S, Sharma K, Wander L, Wang J, City-wide road distress monitoring with smartphones. In: 21st World congress on intelligent transport systems: reinventing transportation in our connected world, ITSWC 2014, intelligent transport systems (ITS).

[Miraliakbari et al, 2014] Miraliakbari, A., Hahn, M., Maas, H.-G., 2014. Development of a Multi-Sensor System for Road Condition Mapping. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, 2014, pp. XL-1, 265-272, doi:10.5194/isprsarchives-XL-1-265-2014.

[Miss Hetal, 2013] Miss Hetal J. Vala and Prof. Astha Baxi, A review on Otsu segmentation algorithm, In International Journal of Advanced Research in Computer Engineering and Technology, February 2013, Volume 2, Issue 2.

[Ouyang et al, 2013] Ouyang A, Dong Q, Wang Y, Liu Y .The classification of pavement crack image based on beam let algorithm. In: 7th IFIP WG 5.14 international conference on computer and computing technologies in agriculture, CCTA 2013.

[Soji et al, Radhakrishnan et al, 2016] Soji Koshy, Radhakrishnan. Detection of Cracks Using Different Technique: A Survey, Dept. of Computer Science and Engineering, Baselios Mathews II College of Engineering Sasthamcotta,Kerala, India, IJCSN International Journal of Computer Science and Network, February (2016),pp. 2277-5420.

[T. Romen Singh, 2011] T. Romen Singh, Sudipa Roy, O. Imocha Singh and K.Manglem Singh, ”A new local adaptive thresholding technique in binarization ”International Journal of Computer Science Issues, November 2011, Vol. 8, Issue 6, No 2.

[Tom et al, 2004] Tom Fawcett. ROC Graphs: Notes and Practical Considerations for Researcheres, HP Laboratories, March 16, (2004).

[Tsai YC, 2010] Tsai YC, Kaul V, Mersereau RM, Critical assessment of pavement distress segmentation methods, In J Transp Eng Journal, pp.136 (1):11–19.

[US Department of Transportation, 2014] US Department of Transportation. Distress Identification manual for long-term pavement performance program. In Research, Development, and Technology Turner-Fairbank Highway Research Centre, Revised May 2014, pp. FHWA-HRT-13-092.

[Yao M, 2015] Yao M, Zhao Z, Yao X, Xu B, Fusing complementary images for pavement cracking measurements. In Proceedings of Meas Sci Technol, 2015, pp. 26(2):025005.

[Zhang et al, Hu et al, 2005] Zhang C-K, Hu H (2005) Feature selection using the hybrid of ant colony optimization and mutual information for the forecaster. In: Proceedings of 2005 international conference on machine learning and cybernetics, 2005. IEEE.

[Zhou H, 2010] Zhou H, Yang S, Zhu J, Illumination invariant enhancement and threshold segmentation algorithm for asphalt pavement crack image. In Proceedings of 2010 sixth International conference on wireless communications, networking and mobile computing, WiCOM 2010. Chengdu.

Authors' Information

Sara Ashraf *– Ain Shams University, Faculty of Computer and Information Sciences, Computer Science Department;* ***e-mail****:* sara.ashraf@cis.asu.edu.eg

Dr. Islam Hegazy - *Assistant Professor, Computer Science Department*

e-mail: islheg@cis.asu.edu.eg

Fields of research*: security, networks, wireless sensor networks, cloud computing, image processing;*

Prof. Dr. Taha I. Elarif *- Ain Shams University, Faculty of Computer and Information Sciences, Computer Science Department;*

e-mail: taha_elarif@yahoo.com, taha_elarif@cis.asu.edu.eg.

Major Fields of Scientific Research*: Computer Graphics, Image Processing and Artificial Intelligent.*

IMAGE CLASSIFICATION WITH CONVOLUTIONAL NEURAL NETWORKS

Karlen Mkrtchyan

Abstract*: The growing increase of demand in computer vision, arises the problem of handling specific algorithmic to this field. Image classification is one of these problems widely used in many applications. Starting from late 2000's Neural Networks based algorithms perform best in this task. The most accurate results are obtained with help of Convolutional Neural Networks, which currently outperform all other algorithms, at least by accuracy. Nevertheless, they have problems connected with the memory use and training time, which in some cases are not acceptable. Thus there is a need for techniques for tackling with problem and fining tradeoff between complexity, time and accuracy.*

Keywords*: Machine Learning, Distributed Learning, Supervised Learning, Classification, Image Classification, Deep Neural Networks, Convolutional Neural Networks, Tensorflow.*

ITHEA Keywords*: I.2.11 Distributed Artificial Intelligence, I.4 Image Processing and Computer Vision, I.5.1 Models*

1.Introduction

Nowadays Computer Vision [1] challenges are one of the most important problems in computer science and in machine learning. There is a huge variety of solutions to these classes of problems, and there is no one ultimately universal algorithm to handle all the cases. Image classification, is one of these problems, which gained popularity due to the need of analyzing vast amounts of media information generated every day, generated from various sources (WEB, Mobile etc.). Here are some of the real world application examples. Orbital Insights for instance, analyzes images to count cars and oil tank levels automatically for predicting mall sales, and oil production respectively. There are insurance companies using computer vision to analyze the damage on assets under policy for deciding which one must be the compensated. The automotive industry has embraced computer vision (and deep learning) aggressively in the past five years with applications such as scene analysis, automated lane detection, and automated road sign reading to set speed limits. The media world is leveraging computer vision to recognize images on social media to identify brands so that companies can better position their brands around the relevant content. eBay uses computer vision to search items with images. In health care, there is classic application of detecting disease in MRI scans. In retail, many companies are interested in analyzing the shopping carts in-store shoppers for detecting items, and making recommendations in store about what else they might want to buy. We can think of this as a recommendation engine. Also, retailers use even more complex cameras taking more complex pictures (hyper-spectral imagery). Optical character recognition (it's a technology to convert scanned images to documents), person identification with Iris code, self-driving cars, authentication with face, 3D modeling of real world scenes widely used in earth maps, obstacle tracking and vision based robotic learning. These are a few examples of computer vision ideas that are under the development now.

Computer vision and deep learning have their challenges when it comes to real world problem and production use. These challenges include

- Getting enough data of good quality
- Managing executives' expectations about model performances
- Being pragmatic about how bleeding-edge we really need our network to be
- Planning data ingest, storage, security, and overall infrastructure
- Understanding how machine learning differs from software engineering, to avoid misaligned expectations

Formally, the problem does not differ from a conventional classification problem; however, it has its specific hardness and solution methods. Conventional machine learning algorithms have their implementations on this problem, some of them are KNN based image classification [2] [3], SVM based image classification [3] [4] and Decision Tree image classification [5]. Each of these algorithms has their benefits and drawbacks comparing to each other. But in recent years by evolving of Neural Networks and Deep Neural Networks most conventional algorithms are outperformed by them at least by accuracy. Here specifically we will concentrate on Convolutional Neural Networks (CNN) due to their huge success in recent years on this type of problems.

2. Convolutional Neural Networks

Convolutional Neural Networks are designed to work with data that has grid like topology. From this perspective image can be viewed as 2D grid. The name “convolutional” is justified with the use of mathematical operation called convolution, in place of matrix multiplication. To get deeper into CNNs, firstly, let’s fill the background around operation of convolution.

In general, convolution is an operation on two functions. Suppose we have two real valued functions x and w the indefinite integral

$$s(t) = \int x(a)w(t-a)da$$

This operation is called convolution and commonly it is defined as follows

$$s(t) = (x * w)(t)$$

From the perspective of convolution neural networks the function x is referred to input information and w is referred to be a $kernel$ function. From the assumption, that x and w are defined only on integer values we will have the discrete from of convolution as follows

$$s(t) = (x * w)(t) = \sum_{-\infty}^{\infty} x(a)w(t - a)$$

In machine learning applications the input, as we mentioned above, is a set of multidimensional arrays, and, consequently the kernel is multidimensional and adapted to that specific learning case. For instance, if we use as an input two dimensional image data, we will probably use two-dimensional kernel function and the operation of convolution will be as follows

$$S(i,j) = (I * K)(i,j) = \sum_{m}\sum_{n} I(m,n)K(i - m,j - n)$$

The key points that defer CNN from ordinary neural networks are the followings sparse interactions, parameter sharing and invariant representations.

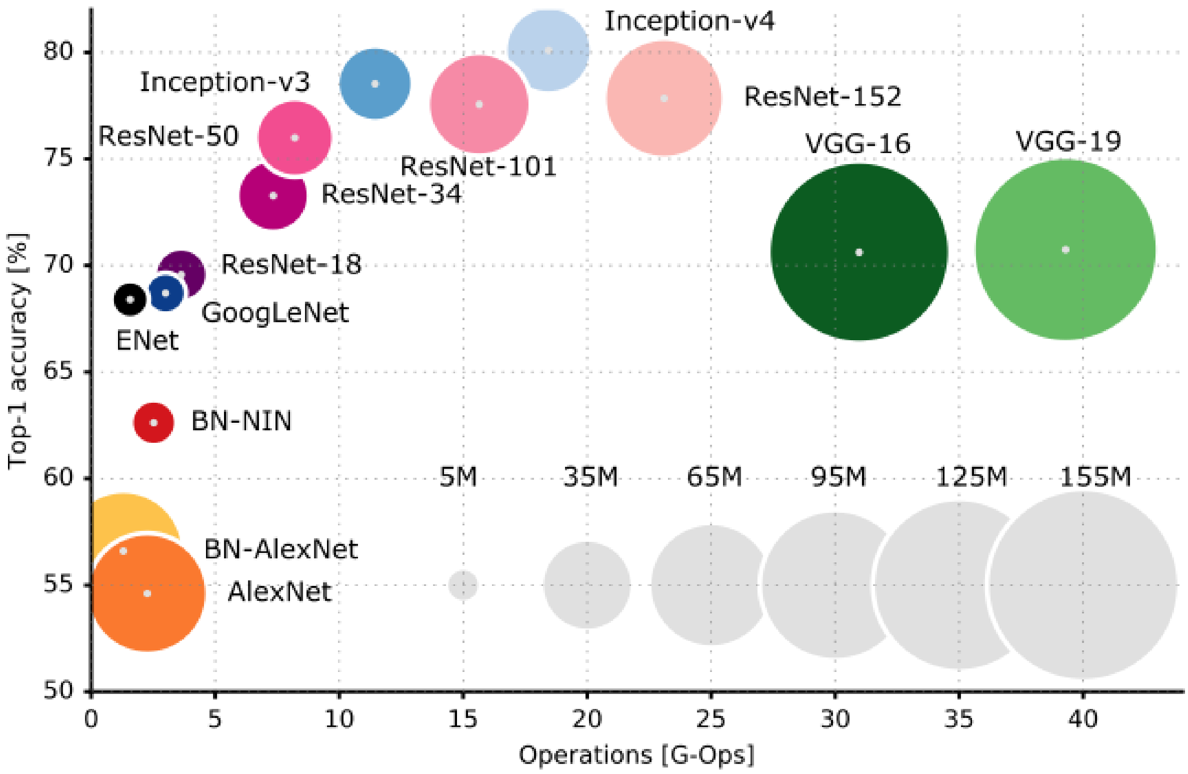
Traditional neural network layers use matrix multiplication by a matrix of parameters with a separate parameter describing the interaction between each input unit and each output unit. This means, that every input is connected with every output. Instead, CNNs have sparse interactions. Input image for example, may have millions of pixels, but after reducing with kernel we can detect meaningful feature with only hundred or thousand pixels, which is statistically efficient and reduces computational complexity. For instance, if there are m inputs and n outputs, then matrix multiplication requires $m \times n$ parameters and the algorithms used in practice have complexity $O(m \times n)$ (per example). If the number of connections each output may have, is limited to k, then the sparsely connected approach needs only $k \times n$ parameters and complexity becomes $O(k \times n)$.

Traditional neural networks, when computing output never get back to calculations used in that stage. For instance, weight matrix that has been calculated is never revised and in each step separate calculation is done. In the case of CNNs parameter sharing means that rather than learning a separate set of parameters. Only one set is learned. The complexity is still $(k \times n)$, but the memory requirements are very low here.

Function $f(x)$ is called invariant to a function g if $f(g(x)) = g(f\ (x))$. In the case of CNNs, if g is a function that translates the input, i.e., shifts it, then the convolution function is invariant to g. The benefit of this feature is that CNNs are invariant to scaling, rotating and some other standard transformations.

3. Purposed Method

As mentioned above the image analysis problem itself is very challenging and has many solutions. Nevertheless, no solution is universal. Traditional machine learning algorithms have problems dealing with huge data. In the case when dataset is small conventional algorithms have problems with accuracy. Overall, if we don’t narrow to very specific case, it is most likely that CNNs will outperform conventional algorithms, at least by accuracy. The famous ImageNet [8] 1000 class problem is one of the most common challenges in Computer Vision. ImageNet is a hierarchical dataset with approximately 1.2 Million labeled images. The problem is to classify the images as accurate as possible. There are many trained models that solve this problem with some accuracy, and the list is led by CNN-S. Here are the top models



The image circle *radius* shows the number of parameters comparing to others

As can be seen from the chart, the ResNet and Inception [6] are the best, having under consideration computational time and accuracy.

Yet, building Inception-V3 on ImageNet requires weeks of training with powerful GPU-s, which is too long and expensive for many use cases. Let’s say we need a model for small applications, online learners, mobile applications, research groups and so on. In these cases, we need small computational time and relatively high accuracy, which is unlikely to be achieved with only using bare use of CNNs. To achieve this, we purpose an algorithm to solve this problem by using CNNs. The idea of the application is to divide real dataset into small pieces, pick some of these pieces randomly and build multiple models separately. Then, integrate these models to predict a final output for unlabeled samples. The method has restriction on data distribution, it requires the distribution to be the same, i.e. data set must be drawn with exact uniform distribution. It comes from the fact that we pick random pieces of data and we silently assume that picked data in some degree will represent the specification of the whole data. The size of picked datasets depends on the dataset we are operating, but we assume that every time we pick enough data, so that it can represent the whole data in some way, and the exact formula of how much we take is out our topic and we will skip it here. It's necessary to note, that we don't need the whole data set to be in one place, and data can be distributed on different machines, which will be faster, for data sampling. Below are given the details of the implementation of algorithm.

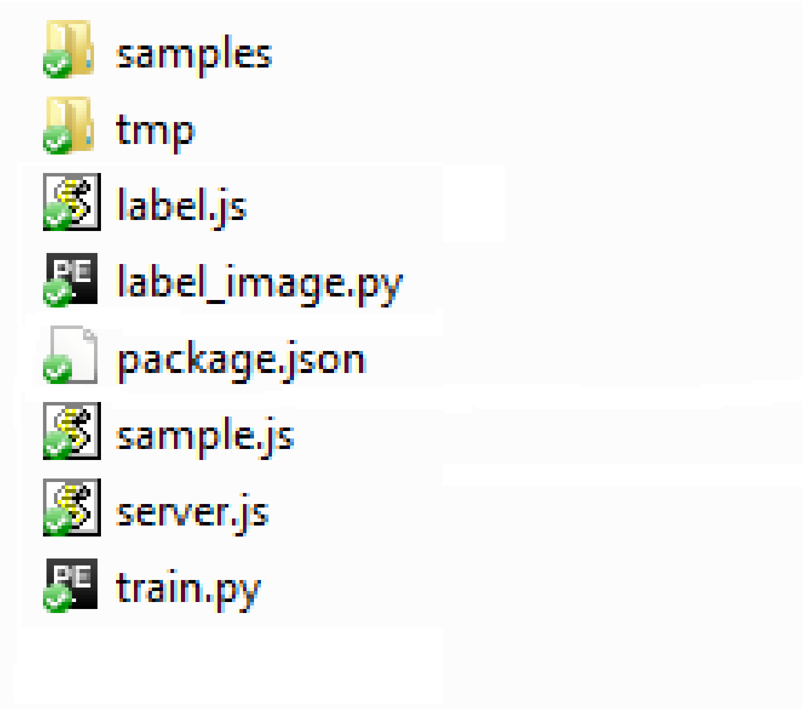
The algorithm is the neural network based image classifier. More specifically it uses convolutional neural networks. The algorithm works as explained in section 2. Training data is given as a set of labeled image files. Image files are processed through convolution layer that is, image file is analyzed and the operation of convolution is carried out for each convolution matrix. Then the output from these operations is collected and Max Pooling [5] operation is done, after which we receive matrix from our initial matrix (image pixels). Received matrix is given as input to Feed Forward Neural Network, and the process is done until the cost function is optimized or the preferred accuracy achieved. As an optimizer we have chosen gradient decent and the cost function is taken to be Cross Entropy [5].

4. Implementation Details

The core architecture of CNN used here, is called "Inception", mentioned above, which is developed by Google and practically proved itself to be one of the top accuracy classifiers in image classification problems. The learning process is carried out with the use of transfer learning for using previous trained Inception-V3 as a training basis.

Transfer learning or Inductive Transfer [9] is problem in machine learning, directed to knowledge storing while solving one problem, so that it can be used in another problem in the future. For example, once we learned to distinguish cars (doors, windows, tires and etc.), we can try to apply this model to distinguish objects with related features, for instance trucks. In the case of Convolutional Neural Networks, we just add one more layer to CNN. The justification of using this method with Inception-V3 is that, the algorithm is designed to distinguish any specific details and differences between images and, consequently, chosen dataset plays small role in this case. Thus, Inception-V3 can distinguish very small details in images, although initially it was deigned to solve ImageNet 1000 class problem.

Here is given the short description of the experimental program meant to solve the problem addressed in section 3. The experimental program is written in $Javascript$ and $Python$. The requirements to the environment are the followings **- Python 3.5+, Node JS 8.9 +, Tensorflow 1.7.0+** and any **64-bit** architecture operating system. The project folder structure is as shown below



samples
tmp
label.js
label_image.py
package.json
sample.js
server.js
train.py

Folder $\boldsymbol{samples}$ contains training dataset, that will be used to train a classifier, and it should contain labeled data in specific order. Let's consider the case we are given the set $\{c_1, c_2, c_3, \dots c_n\}$ of classes, then the file structure must be in the following order of $samples/c_i/\{actual\ files\ from\ c_i\}$ for $= 1,2, \dots n$. In Figure 1 we give the high level diagram of the implemented program, showing the connections among all modules.

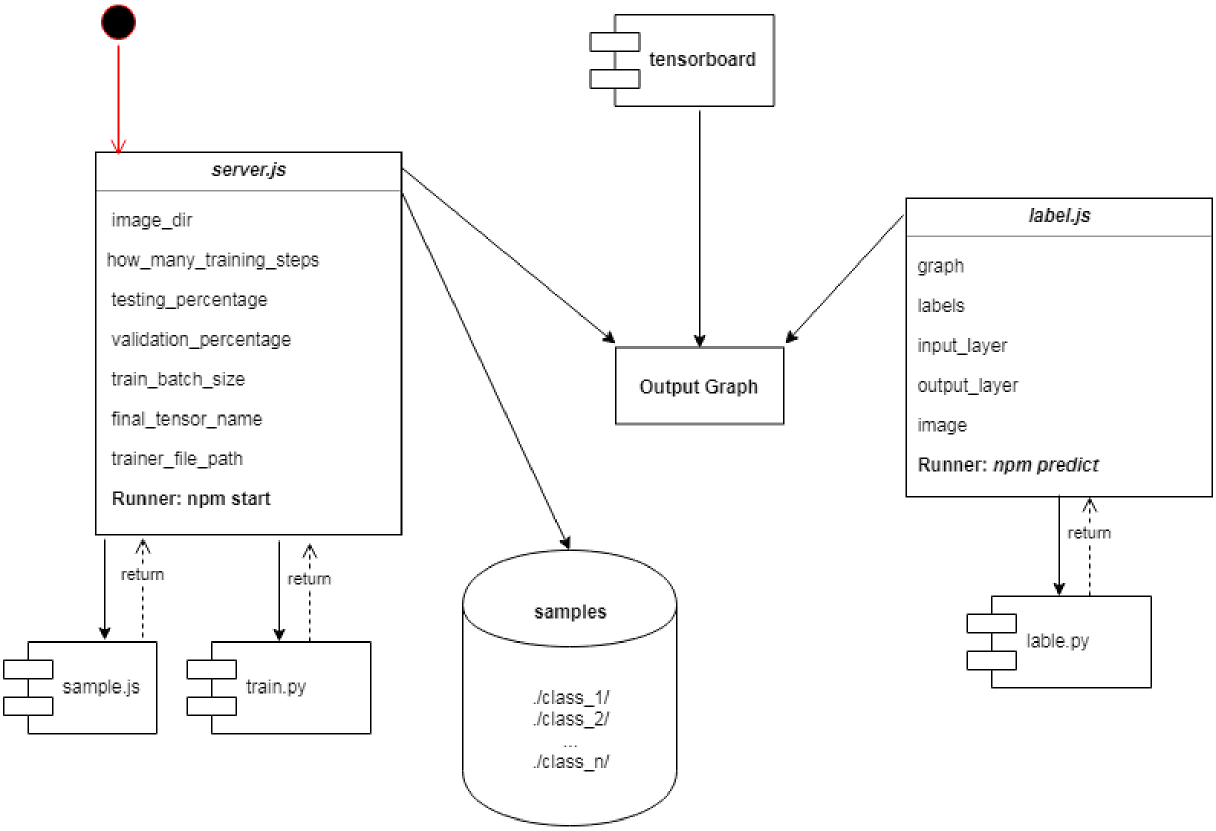


Figure 1. *High level diagram of image classifier implementation*

5. Training Process

$./train.py$ is $Python$ file, which uses Google's $Tensorflow$ for training classifier. It uses $Tensorhub$ to download Inception-V3 architecture by default, but can be configured from outside, and it is possible to set any known architecture that works with $Tesnorflow$. To set other module than Inception-V3, one should be use the **--tfhub_module** command line property. File also contains cost function definition and its optimizer which can be configured as well, by default we have used $\boldsymbol{GradientDescentOptimizer}$ as an optimizer and $\boldsymbol{sparse\_softmax\_cross\_entropy}$ as a loss function to be optimized.

$./server.js$ is a wrapper around the $./train.py$ and gives almost full control over it, that is, it provides many programmable parameters for using ./train.py. The parameter list is as follows:

$image\_dir$	data
$how\_many\_training\_steps$	500
$testing\_percentage$	10
$validation\_percentage$	10
$train\_batch\_size$	100

Additionally $server.js$ file uses $./sample.js$ to sample files for the cases when two or more classifiers needed for training. Sampler works as follows

$$dataSampler(DATA\_FOLDER, SAMPLED\_DATA\_FOLDER, SAMPLE\_COUNT)$$

After the function call, by default, in $./tmp$ folder will be created folder named $./data$ which contains sub folders as follows $./tmp/data/sample\_i/\{files\}$ where $i = 1, 2, 3 \dots, SAMPLE\_COUNT$, $SAMPLE\_COUNT$ meant to be equal to the number of classes c_i. $server.js$ uses the following function to perform the training

$$trainCnn(numberOfCNNS = 1, config, callback)$$

Function trains CNN classifiers in parallel. The count of CNNs is controlled by parameter $numberOfCNNS$. Each CNN fed with data from appropriate folder $./tmp/data/sample\_i/$. At each step program logs its current state in the streaming system log, which can be observed. After the training process is finished results are stored in the following way: $./tmp/\{date\ of\ execution\}/cnn\_i/$ is the root folder for each CNN $i = 0,1,3 \dots n$. Each folder contains its training process logs, output graph and output labels as follows: $./retrain\_logs$ as the log directory, $output\_graph.pb$ as the final graph of CNN, and $output\_labels.txt$ contains text labels of initially provided classes.

6. Training Parameters

As mentioned above, the training process is implemented by the *server.js* file. It provides command line interface with parameters and configuration file for operating with training parameters. Figure 1 displays the parameters of the trainer module.

image_dir is the dataset folder, which must contain labeled data, divided in folders. Each folder name represents a class value, and is obtained automatically by scanning the folders. The number of training examples is not restricted in any way, and exact count of files is very specific depending on data type used, practically, it is preferable to have at least some 100 examples from each class. Accepted file formats are the followings 'jpg', 'jpeg', 'JPG', 'JPEG'. If the ***image_dir*** attribute is not set, then the program looks to folder named $\boldsymbol{samples}$, in the root directory of the project.

how_many_training_steps parameter is the indicator for the program, whether how many tainting step needed for algorithm to optimize the cost function. In our case as an optimizer we pick gradient decent algorithm and this parameter shows how many times gradient will be calculated. It is suggested that 4000 is optimal count, but in testing purposes it can be minimized to build testing models faster for research purposes. By default the value is set to 4000.

testing_percentage is the percent of provided dataset, which will be used for testing during the training process. The data is selected randomly from dataset. As in many cases here it is recommended to set 10 percent, and by default it is 10.

train_batch_size is the count of files fed to training algorithm on each training step. The parameter is very important because its value may affect training efficiency and training time. Ideally, if we didn’t have problems with memory and computational time we would set the parameter equal to number of training examples, it means fed network with all data, which is very good for gradient, because it would be maximally accurate, for given dataset. But in real world applications there are time and memory constraints, that’s why we need this parameter. Below figure shows how batch size and gradient are connected.

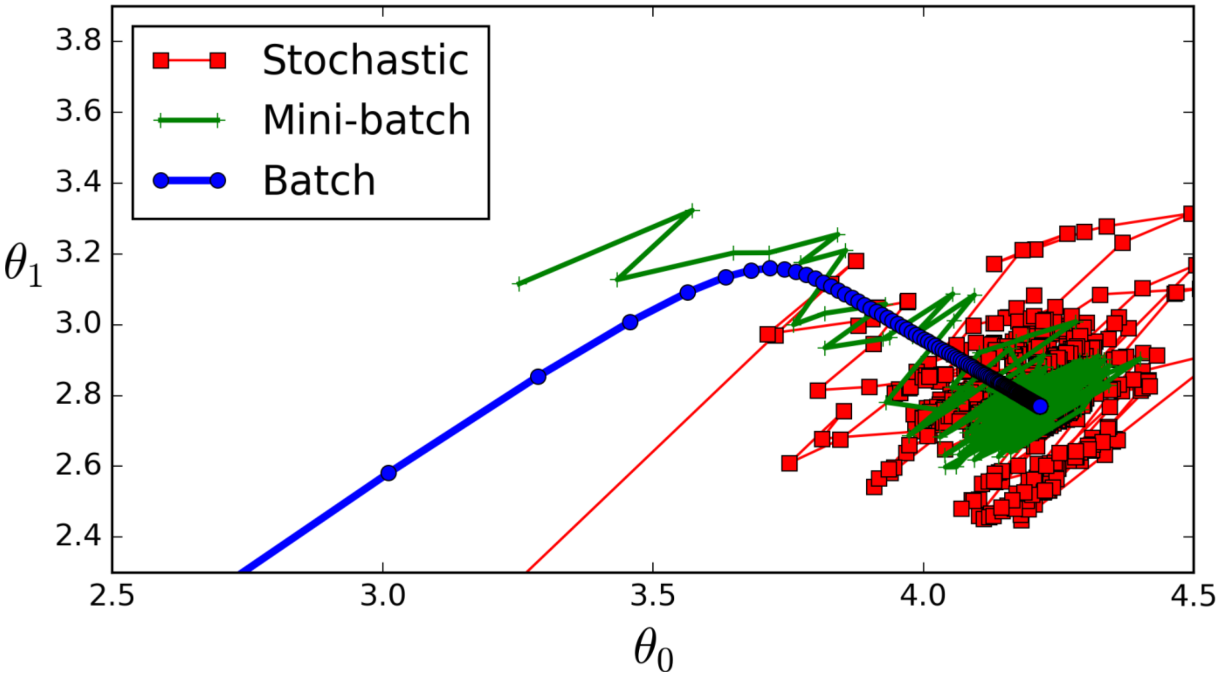


Figure 2. *Connection of gradient and file batch size*

Stochastic is a mini-batch with ***train_batch_size*** equal to 1. Batch is the full batch size case; Mini-Batch is some value different from 1. It is seen that mini-batch fluctuates compared to the full batch, but as explained above it is trade-off between time-memory and accuracy. By default, it is set to 100 and optimal count for specific data set can be calculated with experiments. The source files and the detailed description of the project are provided in our public repository *github.com/DM-lab5/transfer_learnig*.

6 Training Reports

To see the process of training, log files can be used, to visualize the whole training process reports. To do that $tensorboard\ --\boldsymbol{logdir}$ ./ need to be run in the directory where tmp folder is located, by default it is in the root folder. After execution of command web server will be running on the local machine and the URL will be given as an output of command. All reports can be observed by any web browser by using given URL. Here is the example of reports from $tensorboard$

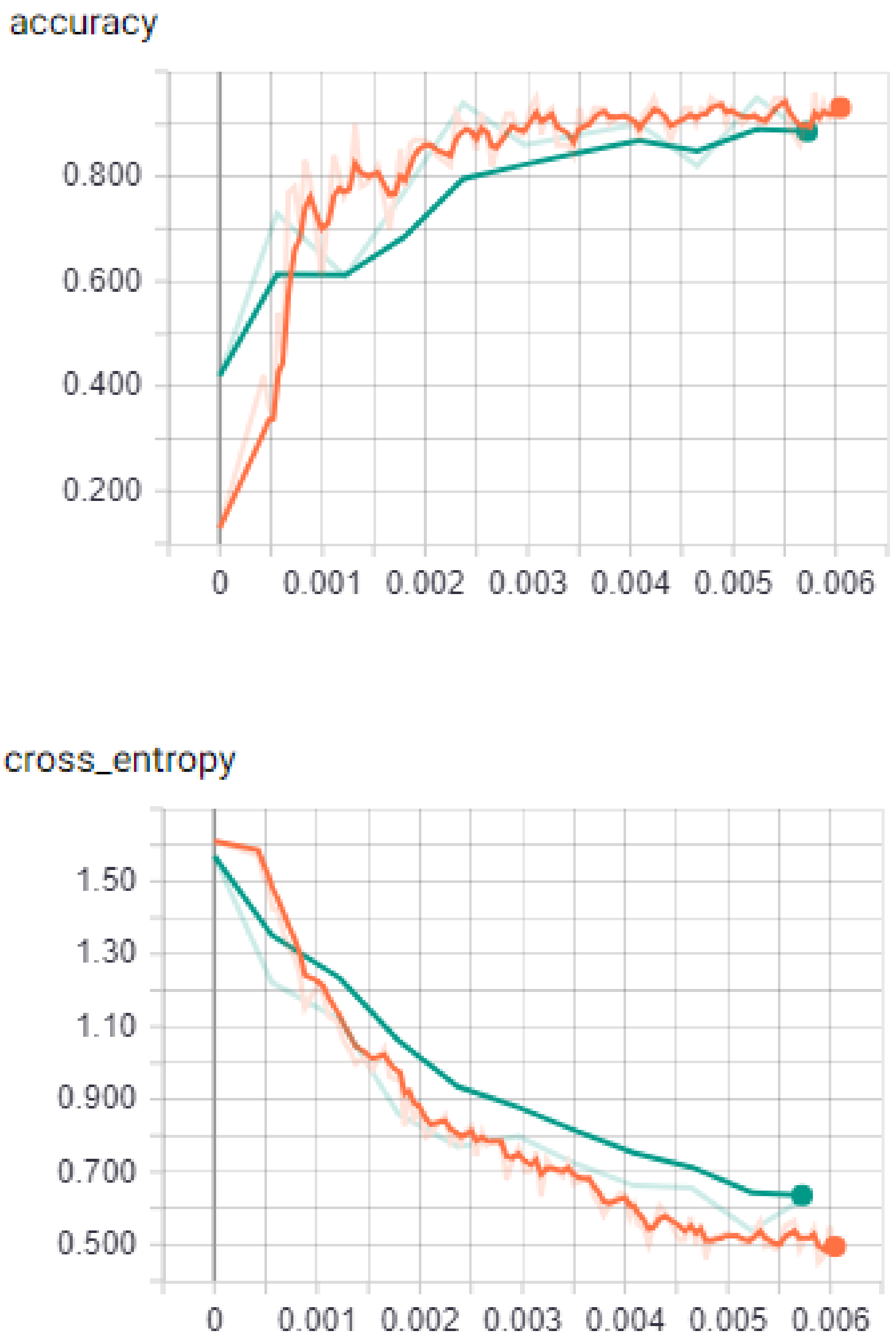


Figure 3. *The retrain log reports for already trained CNN, where shown accuracy and cross entropy values during training process.*

Figure 3 shows how the distribution of some Tensor in the $tesnsorflow$ graph has changed over time. It does this by showing many histogram visualizations of your tensor at different points in time.

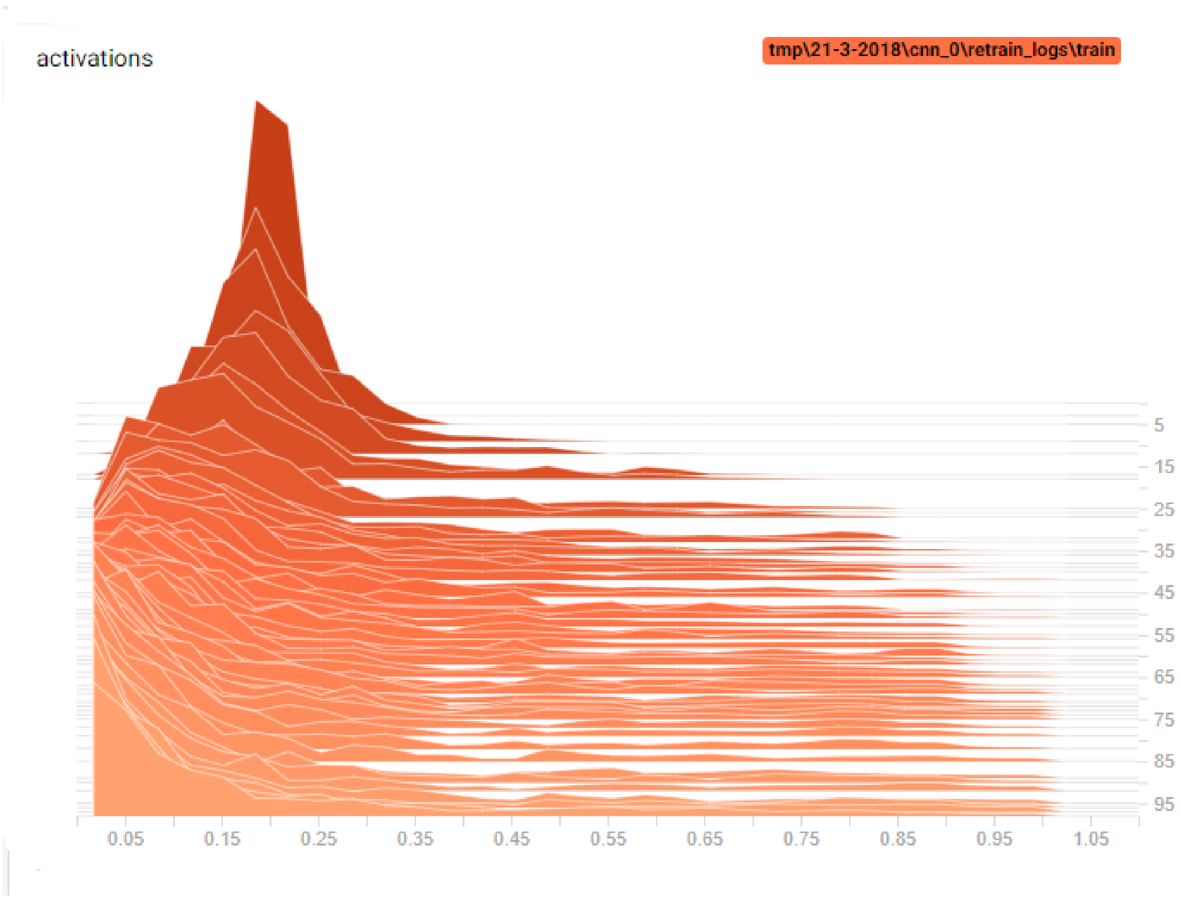


Figure 4. *The train histogram that shows how the distribution of initial tensor has been changed*.

Conclusion

Convolutional Neural Networks are showing themselves to be very useful and applicable, and for many real world problems it is enough, at least from perspective of accuracy. Nevertheless, it requires much computational time and in many applications it's not acceptable and expensive. The method we tried to apply here, is aimed to reduce this computational time, it practically shows itself to be applicable.

Training multiple weak classifiers and combing them into one is method applied in other conventional algorithms. The ways we can combine classifiers are many including majority voting, weighted voting, learning from the outputs of the classifiers and etc. Combining weak classifiers are useful not only from the perspective of time and memory restrictions. Firstly, when data is distributed training can be done in parallel and so we can train classifiers parallel by preserving data privacy issues. Secondly, when data distribution is not fixed, and samples are drawn from random distribution, in most cases, combinations of weak classifiers are more accurate than with the ones trained with all examples. It is necessary to note that transfer learning used here, can be replaced by and learning method, because the method don’t make assumptions on learning algorithm and it gives flexibility to replace transfer learning with any learning method that can work with provided data format.

References

1. L. G. Shapiro, G. C. Stockman Computer Vision, 2001.
2. O. Boiman, E. Shechtman, M. Irani, In Defense of Nearest-Neighbor Based Image Classification, Computer Vision and Pattern Recognition (IEEE), 2008.
3. J. Kim, B. Kim, S. Savarese, Comparing Image Classification Methods: K-Nearest-Neighbor and Support-Vector-Machines, 2012.
4. Z. Wang, J. Feng, S. Yan, Linear Distance Coding for Image Classification, IEEE Transactions on Image Processing, Volume: 22, Issue: 2, 2013.
5. K. Lu, D. Yang, Image Processing and Image Mining using Decision Trees, Journal of Information Science and engineering 25: 989-1003 (2009).
6. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the Inception Architecture for Computer Vision, Computer Vision and Pattern Recognition, IEEE, 27-30 June, 2016.
7. N. Buduma, Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms, 1^{st} Edition, 2017.

8. J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, Computer Vision and Pattern Recognition, 2009.

9. W. Jeremy, V. Dan, W. Sean, A Theoretical Foundation for Inductive Transfer, 2007.

Authors' Information

Karlen Mkrtchyan *– Institute for Informatics and Automation Problems of the National Academy of Sciences of Armenia, PhD candidate; 1 P.Sevak str., Yerevan 0014, Armenia;*

e-mail*: mkrthcyan.karlen@icloud.com*

Major Fields of Scientific Research*: Distributed Computation, Distributed Machine Learning.*

TABLE OF CONTENTS