International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

Image Validation and Forgery Detection

Anmol Bhat[1], Vinay Hegde[2]

[1]*B.E, Dept. of Computer Science and Engineering, R V College of Engineering, Bangalore, India*
[2]*Professor, Dept. of Computer Science and Engineering, R V College of Engineering, Bangalore, India*

Abstract - *This paper reports the use of image processing for image validation and forgery detection. In the current scenario, the expense of a business trip is calculated by all the bills and receipts the person submits to their company's finance team. It becomes very tedious and nearly impossible to manage all the bills and receipts manually both by the finance team as well as the person on the trip. The purpose of the paper is to automate the process by moving everything online by using the concepts of Image processing and Machine learning. All the expenses will be managed by the system. The user only needs to take an image of the receipt or bill and upload it. The system will validate whether the image is genuine or forged and then create an expense item. At the end of the trip, the user just needs to file the expense and it will generate an Expense Report with all the expenses and send it for further processing to the finance team.*

Key Words: **Image Processing, Machine Learning, Expense Item, Expense Report, Optical Character Recognition, Object Detection, Text Detection**

1. INTRODUCTION

Image processing is a domain of Artificial Intelligence where relevant information is extracted from the image and processed further according to the requirement [1]. Image forgery is a way of manipulating the digital image to hide some meaningful or useful information about the image. The detection of a forged image is driven by the need for authenticity and to maintain the integrity of the image [2]. The purpose of image validation and forgery detection is to completely remove the use of paper and manual effort of filing an expense report for reimbursement of money.

The user needs to take a picture of the receipt and create an expense item and upload it. Then the image will be processed and validated if it is not forged and is in accordance with our requirements. A timestamp will be returned to tell the user that it has been validated. The system requires a Model that will be trained to detect a forgery in the image and then validate the image. OCR will be used to extract the text out of the validated image to automate the process of filling the form. Also, text detection will be used to optimize the whole process.

The principle behind this system is, firstly denoising the image then text detection, image forgery and validation and OCR to extract the text. A combination of various algorithm is used firstly for denoising of image Autoencoder concepts [5][6] have been used, then for text detection [4] a faster-RCNN [3] model is used and is trained on self-made dataset. Then for the image forgery detection [8] a combination of 2 algorithms is used, Pixel based, and Format based. The OCR is build using DRAM (Deep Recurrent Attention Model) [7] the extracted text then used to populate the fields and create an expense item.

At the end of the trip the person can just file the expense report to their finance team and they just need to check the validity of the expense report and process it further.

1.1 Motivation

This project has been chosen to simplify the process of reimbursement as lakhs of our users have to go through a lot of pain to get their money reimbursed and many times it takes more than a month as it takes time to manually do everything. A better user experience and simplification of the process of reimbursement and filing an expense is required. Also, this improves the persons productivity as it simplifies a major task by automating it so that they can focus on other important matters as, reimbursement of money is not a priority for them.

2. METHODOLOGY

The whole system is divided into different modules and each module needs to be working for the whole system to be working. Fig 1 gives the general architecture of the whole system.

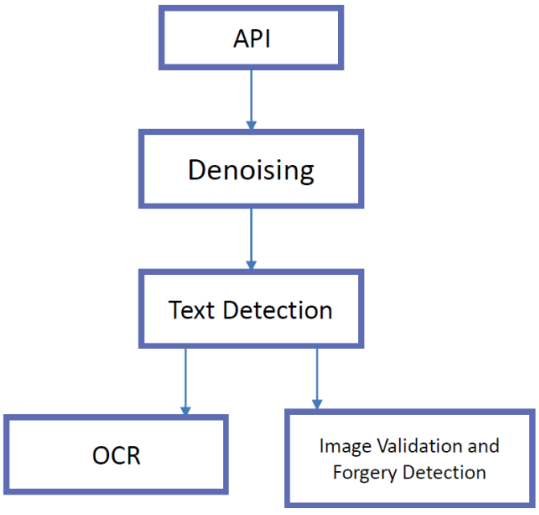

Fig-1 General Architecture Diagram

From Fig 1 it can be seen that our system is divided into 5 subsystems where each performs a specific task and the result are sent to the next subsystem. A better understanding will be achieved by going through each subsystem one after another.

2.1 Creation of API

The first step in the system is to create an API which will take care of all the processing and validation and will return the timestamp of the validation. This API receives the user UUID and the image as multipart/form-data in JSON. The API then validates the user UUID and its access role and after validation it sends the image for further processing. The API end point has been written using JAVA Springboot [9]. A FullyRESTFul API has been implemented for the same.

2.2 Denoising of Image

The next step is to denoise the image to increase the quality of the image and to remove noise from the image so that the further steps can perform more accurately. This step is needed as the user might not take the image properly or additional noise might be added to the image due to bad network. Image Denoising is done by using a model called Autoencoders. Autoencoder is a technology that use encoding and decoding to increase the performance. It is done by extracting important features from the image, encoding them and then training the model on the encoded data by generating the same image from the encoded data by decoding it according to the output as shown in Fig 2. The model has been trained using the online MNIST dataset as it satisfied our use case. Firstly, to train the Autoencoders an actual image is taken and small gaussian noise is added to it then the important features from the image are extracted and then encoded to reduce the time taken and to remove unnecessary details. Then this encoded data is passed through a decoder which tries to decode it to generate the actual image. This is how the Autoencoder model is trained and then used this trained model to generate denoised image. The denoised image is sent for further processing.

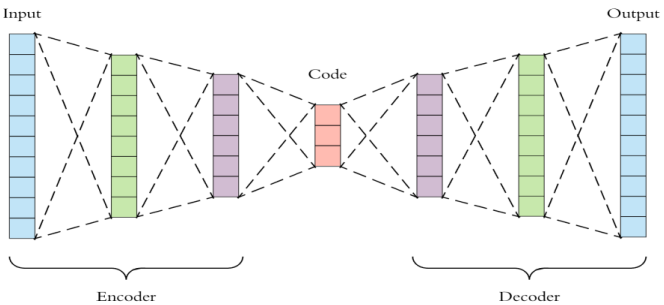

Fig-2: Architecture of Autoencoders

2.3 Detecting Text in Image

The denoised image is passed through the object detection model where the text is detected from the image. For this faster-RCNN model has been used as they are fast and very accurate. Fig 3 shows the architecture of it. The faster RCNN [11] is based on RCNN model where instead of having a lot of regions there are lesser regions which makes it faster but to cope up with less regions more features are extracted from the image. Classifier for each region to detect whether text is there or not [10] is applied. If text is there, then that region is taken and then proceed to next region. This happens till all the regions have been verified once and then in the end image with only text in it is created. A custom dataset has been used for text detection as for this project there was a particular use case of bills and receipts, so around 1000 images with 700 training and 300 test images were taken.

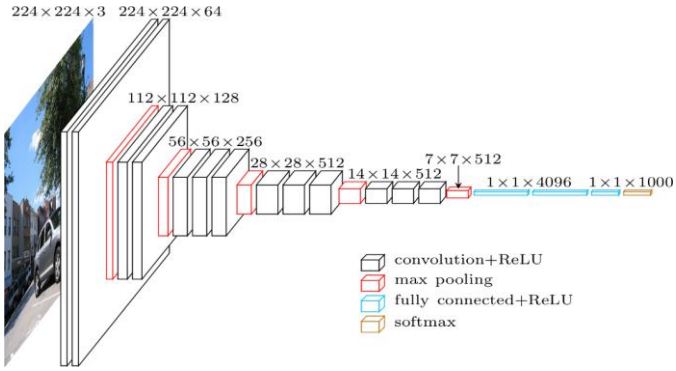

Fig-3 Architecture of Faster-RCNN

2.4 Text Extraction Using OCR

The new image with only text in it then passed through OCR. This OCR is used to extract text from the image. DRAM model has been used to extract text from the image. The dataset use is NIST data set as it has around 8,00,000-character images. The DRAM model is based on recurrent model with attention mechanism. Instead of using a single RNN, DRAM uses two RNNs, a location RNN to predict the next glimpse location and another Classification RNN dedicated to predicting the class labels or guess which character are being looked at in the text as shown in Fig 4. A context network is used to down sample image inputs for more generalizable RNN states. It also chooses to refer to the location network in RAM as Emission Network. The training is done using an accumulated reward and optimizing the sequence log-likelihood loss function using the REINFORCE policy gradient.

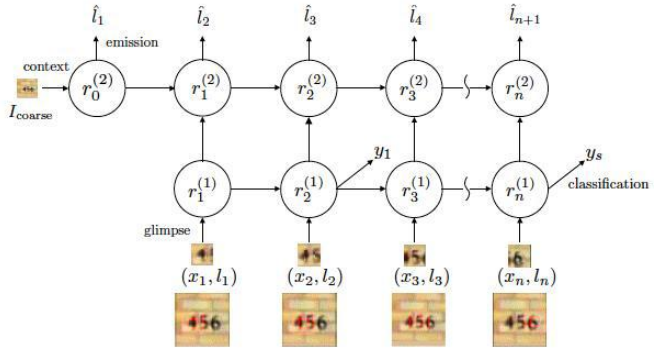

Fig-4 DRAM Architecture

2.5 Detecting Forgery and Validating the Image

The new image generated from step 3 is sent to the image forgery detection model. Here two different models have been used. First is the Pixel Based model, this model checks the validity of each pixel if it has been broken or there are multiple colors on a single pixel and various other pixel related calculation. The other model that has been used is Format-Based model that checks for the image forgery due to compression algorithms. The image could be overlapped with another image while compressing. That is validated by reversing the compression and then checking the validity of the image. Here three compression algorithms i.e. JPEG Quantization, Double JPEG and JPEG Blocking are being checked to verify if the image is forged or not. CNN is utilized to execute the calculation as these procedures misuse the substance-based highlights of a picture for example the visual data present in the picture as appeared in Fig 5. CNN's are propelled by visual cortex. These systems are intended to extricate highlights significant for order for example the ones which limit the misfortune work. The system parameters part loads are found out by Gradient Descent to produce the most separating highlights from pictures took care of to the system. These features are then fed to a fully connected layer that performs the final task of classification.

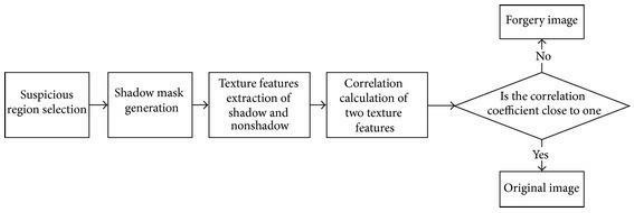

Fig-5 Image Forgery Detection Model

2.6 Complete Structure of the System

After all the elements of the whole system have been completed, the user needs to be notified that the expense item has been created as the process is asynchronous. This is done by publishing a Pub Sub which is subscribed by the application. When the Pub Sub is published the application receives it as it has been subscribed to it. On listening to this Pub Sub the user is notified of the expense item created and then he can review the same and if there are any changes required can do so. Once these elements have come together, the whole flow of the system is achieved. Fig 6 shows the complete architecture of the system. It describes the end to end flow of the system where the user inputs the image of the receipt and his details from the UI. Which then goes for validation to the API and then is validated with the data in the Couchbase DB. If the user is validated it goes to denoising process. Where the image is denoised using autoencoders. After denoising it goes for text detection phase. In this phase the image is scanned for text and eventually the imaged is cropped to the part where the text is present. Once the cropped image from text detection phase is received then it is passed to OCR and Forgery detection parallelly. This is done to optimize the time taken by the whole system as they both independent processes. If the image is forged, then the validator doesn't allow further processing and publishes a failure Pub Sub and the data from OCR is dropped. But if the image is valid then the data from the OCR is taken to fill the required fields and a success Pub Sub is published. The image is then stored in the DB with the validation time stamp and an expense item is created. The Pub Sub published then notifies the user that the expense item has been created and he can then check if the expense item created is correct.

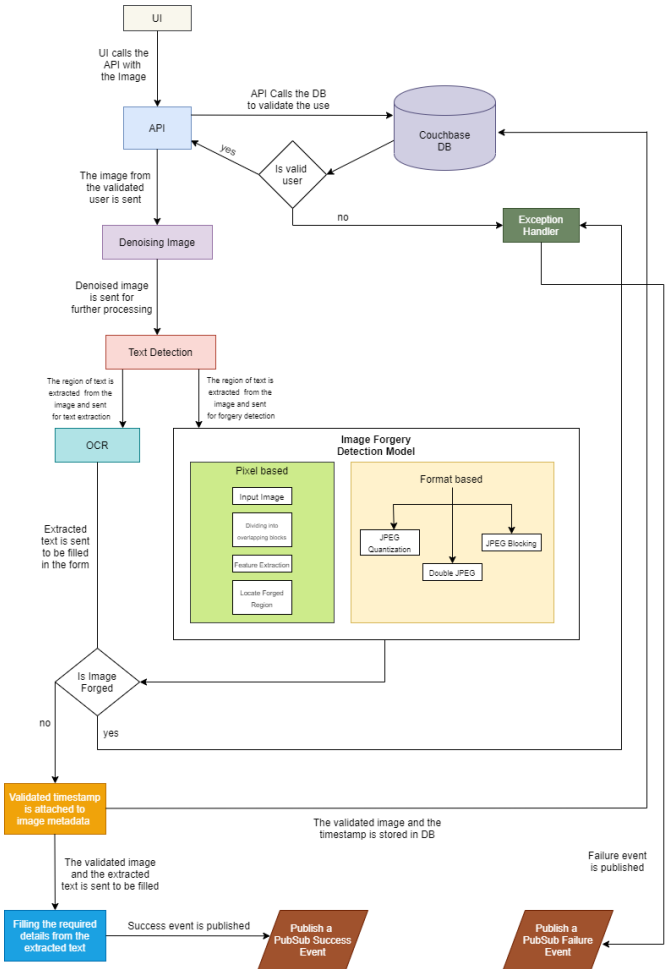

Fig-6 Complete Architecture of the System

3. RESULTS

The experiment was conducted successfully with all the required steps and end to end testing was also carried out to verify the implementation of the system. The following results were achieved after implementation of the project with the mentioned datasets:

1. To implement the denoising model TensorFlow GPU 1.12 and OpenCV were used and have got a good accuracy of about 94.45%. Fig 7 shows the result of the Denoising. Fig 8 shows the Loss curve.

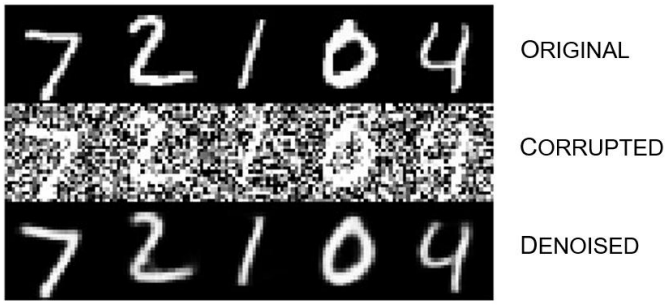

Fig-7 Output of Autoencoder

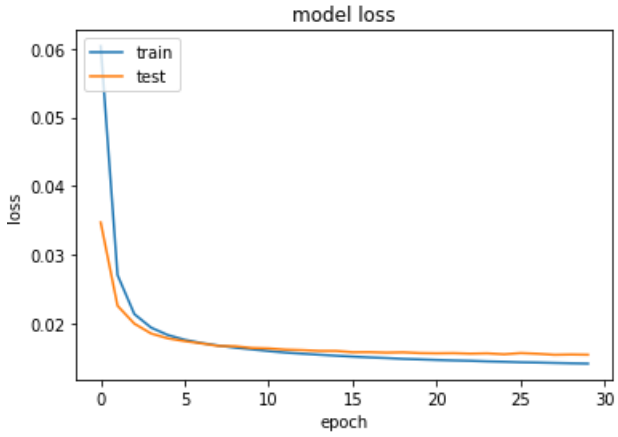

Fig-8 Loss Curve for Autoencoders

2. To implement Text detection, TensorFlow GPU 1.10 OpenCV has been used. To label the images and create a bounding box labeling software called label-Img was used. An accuracy of around 96.57% was achieved this is mainly coz of less data set and can be increased by using more images. Fig 9 shows the test image that has been used. Fig 10 to Fig 12 show the loss curve during the training.

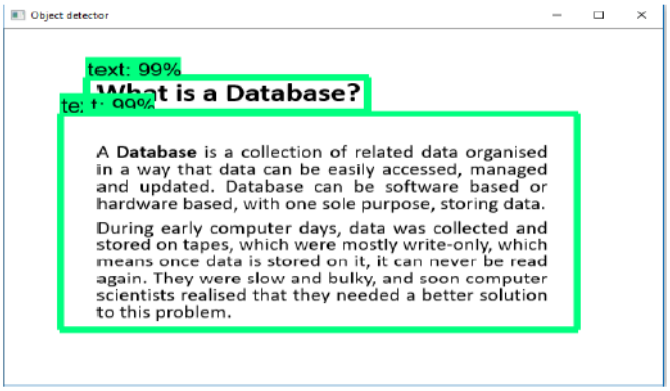

Fig-9 Output of Text Detector

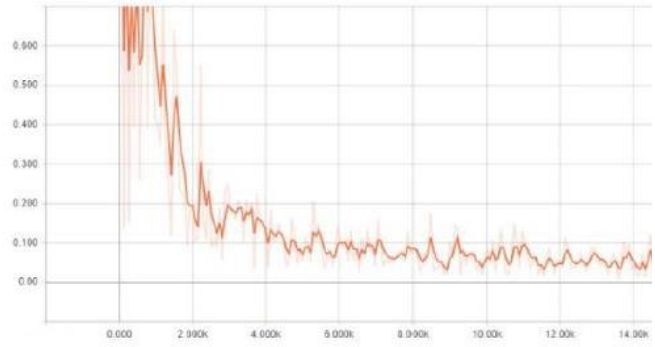

Fig-10 Total Loss

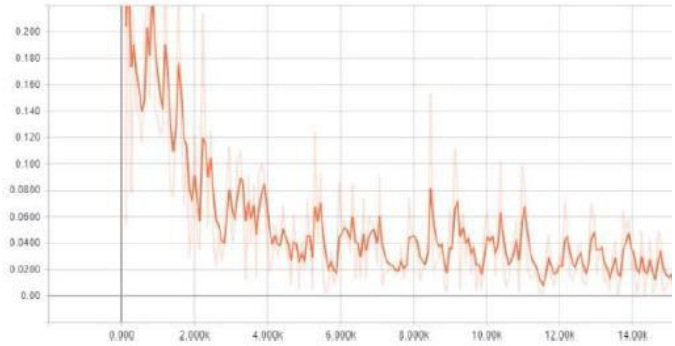

Fig-11 Localization Loss

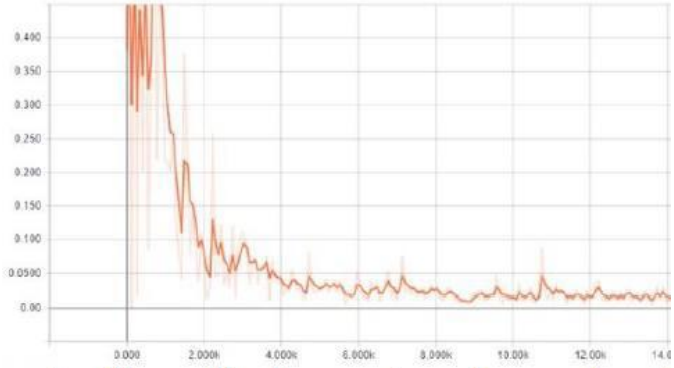

Fig-12 Classification Loss

3. The OCR is implemented using OpenCV and Fig 13 shows the output for Fig 9. Also, Fig 14 shows the loss curve for OCR. The accuracy was around 95.24%.

What is a database? A Database is a collection of related data organized in a way that data can be easily accessed, managed and updated. Database can be software based or hardware based, with one sole purpose, storing data.

During early computer days, data was collected and stored on tapes, which were mostly write-only, which means once data is stored on it, it can never be read again. They were slow and bulky, and soon computer scientists realized that they needed a better solution to this problem.

Fig-13 Output of OCR

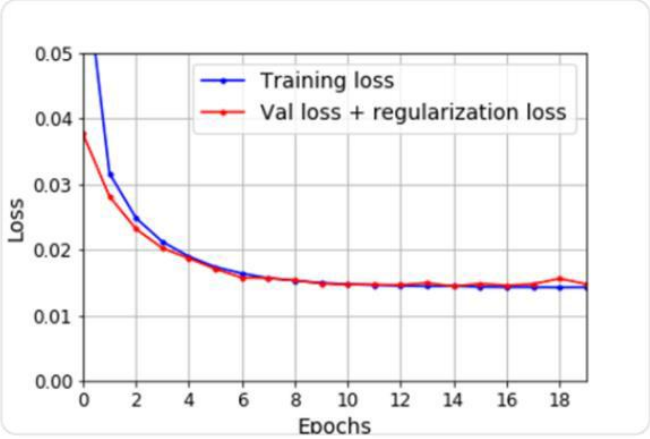

Fig-14 Loss Curve of OCR

4. The Image Forgery detection returns the timestamp of the image and this is shown in Fig 14. This is stored in the database to keep track of duplicates. The accuracy achieved for image forgery model was around 98.55%

Fig-14 Final Output after Image Forgery Detector

4. CONCLUSION AND FUTURE WORK

The project is implemented in parts and each part was a challenge and the biggest challenge of all was to achieve the 60s mark with so many models to be run on image. But after a lot of thought a solution to optimize the overall process was thought upon so an extra phase of text detection was added into the main flow as it will reduce the focus of image to only required part and will remove any unwanted part. Then OCR and Forgery detection models were run parallelly as one was not dependent on the other. The only condition was to only create expense item after OCR data if the Forgery model returned timestamp. These additions and thinking made us achieve our SLA goal. All the models were tested thoroughly and the whole system was tested repeatedly, and overall time taken was maximum of 45s on the production environment and overall accuracy of about 97.38%.

The working system is efficient enough as per requirements. However, there are a few advancements that can be incorporated into the project are:

1. Processing requirements increase the processing time of the system which in turn introduces some delay in the system. It can be encountered by increasing the processing capacity of the system which increases the cost.
2. Although the SLA of 60s was achieved, it is only for a single request when multiple request come, the same SLA needs to be provided. This created a need of scalability which in this case can be very costly.

REFERENCES

[1] Jensen, J.R., "Introduction to Digital Image Processing: A Remote Sensing Perspective", Prentice Hall, New Jersey, 1996.

[2] Barnali Sarma, Gypsy Nandi, "A Study on Digital Image Forgery Detection", International Journal of Advanced Research in Computer Science and Software Engineering, 2014.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in Advances in Neural Information Processing Systems, 2015.

[4]R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14), pp. 580– 587, Columbus, Ohio, USA, June 2014.

[5] Zhenyu Zhao, "Image Denoising by AutoEncoder: Learning Core Representations", The Australian National University, 2012.

[6] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio and Pierre-Antoine Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep

Network with a Local Denoising Criterion", Journal of Machine Learning Research 11, 2010.
[7] Karez Hamad, Mehmet Kaya, "A Detailed Analysis of Optical Character Recognition Technology", International Journal of Applied Mathematics, Electronics and Computers, 2016.
[8] Jyoti Dadwal, Bhubneshwar Sharma, "Design of Image Processing Technique in Digital Enhancement Application", Asian Journal of Applied Science and Technology (AJAST), 2017.
[9] Singh, T, "JAVA Web Design Frameworks: Review of Java Frameworks for Web Applications", International Journal of Advance Research in Science and Engineering, 2015.
[10] Anmol Bhat, Aneesh C Rao, Anirudh Bhaskar, Adithya V, Prof. Pratibha D, "A Cost-Effective Audio-Visual Summarizer for Summarization of Presentations and Seminars", Third International Conference on Computational Systems and Information Technology for Sustainable Solutions, 2018.
[11] Yun Ren, Changren Zhu, Shunping Xiao, "Object Detection Based on Fast/Faster RCNN Employing Fully Convolutional Architectures", in Hindawi. Mathematical Problems in Engineering, Vol. 2018.
[12] Constantine Papageorgiou & Tomaso Poggio, "A Trainable System for Object Detection", International Journal of Computer Vision volume, 2000.
[13] Muhammad AliQureshi, MohamedDeriche, "A bibliography of pixel-based blind image forgery detection techniques", Signal Processing: Image Communication, 2015.
[14] Gajanan K.Birajdara, Vijay H.Mankar, "Digital image forgery detection using passive techniques: A survey", Digital Investigation, 2013.
[15] G. Li, Q. Wu, D. Tu, and S. Sun, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD". In IEEE International Conference on Multimedia and Expo, pages 1750–1753, Beijing, China, 2008.
[16] H. Gou, A. Swaminathan, and M. Wu, "Noise features for image tampering detection and steganalysis". In IEEE International Conference on Image Processing, San Antonio, TX, 2007.
[17] Hany Farid, "A Survey of Image Forgery Detection", IEEE Signal Processing Magazine, 2009.
[18] Susan Hyde, Thad Dunning, "The Analysis of Experimental Data: Comparing Techniques", Proceeding of the Annual Meeting of American Political Science Association, Boston, 2008, pp 233-242.
[19] C.P. Papageorgiou, M. Oren, T. Poggio, 'A general framework for object detection', Sixth International Conference on Computer Vision, 1998.