Blind Source Camera Identification using Machine Learning Techniques

Sudhanshu Patel & **Om Prakash Acharya**

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Blind Source Camera Identification using Machine Learning Techniques

Thesis submitted in partial fulfillment

of the requirements of the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Sudhanshu Patel

(Roll Number: 112CS0174)

and

Om Prakash Acharya

(Roll Number: 112CS0131)

based on research carried out

under the supervision of

Prof. Ruchira Naskar

May, 2016

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Department of Computer Science and Engineering
National Institute of Technology Rourkela

May 11, 2016

Certificate of Examination

Roll Number: *112CS0174*
Name: *Sudhanshu Patel*
Roll Number: *112CS0131*
Name: *Om Prakash Acharya*
Title of Thesis: *Blind Source Camera Identification using Machine Learning Techniques*

We the below signed, after checking the dissertation mentioned above and the official record book (s) of the student, hereby state our approval of the dissertation submitted in partial fulfillment of the requirements of the degree of *Bachelor of Technology* in *Computer Science and Engineering* at *National Institute of Technology Rourkela*. We are satisfied with the volume, quality, correctness, and originality of the work.

Ruchira Naskar
Principal Supervisor

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Prof. Ruchira Naskar
Professor

May 11, 2016

Supervisor's Certificate

This is to certify that the work presented in the thesis entitled *Blind Source Camera Identification using Machine Learning Techniques* submitted by *Sudhanshu Patel*, Roll Number 112CS0174and *Om Prakash Acharya*, Roll Number 112CS0131, is a record of original research carried out by him under my supervision and guidance in partial fulfillment of the requirements of the degree of *Bachelor of Technology* in *Computer Science and Engineering*. Neither this thesis nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

Ruchira Naskar

Declaration of Originality

We, *Sudhanshu Patel*, Roll Number *112CS0174* and *Om Prakash Acharya*, Roll Number *112CS0131* hereby declare that this dissertation entitled *Blind Source Camera Identification using Machine Learning Techniques* presents my original work carried out as a undergraduate student of NIT Rourkela and, to the best of my knowledge, contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom we have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the sections "Reference" or "Bibliography". We have also submitted our original research records to the scrutiny committee for evaluation of our dissertation.

We are fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to us on the basis of the present dissertation.

May 11, 2016
NIT Rourkela

Acknowledgment

This thesis has been possible due to the help and endeavor of many people.

Foremost, we would like to express our gratitude towards our project advisor, Prof. Ruchira Naskar, whose mentor-ship has been paramount, not only in carrying out the research for this thesis, but also in developing long-term goals for our career. Her guidance has been unique and delightful. She provided her able guidance whenever we needed it. Yet she always inspired us to be independent thinkers, and to choose and work with independence.

We would also like to extend special thanks to our project review panel for their time and attention to detail. The constructive feedback received has been keenly instrumental in improvising our work further.

We would like to thank other researchers in our lab and our friends for their encouragement and understanding.

Sudhanshu Patel
Roll Number: 112CS0174
Om Prakash Acharya
Roll Number: 112CS0131

May 11, 2016
NIT Rourkela

Abstract

In this digital era, use of digital images in the field of investigation and authentication has become essential. The authenticity of these images hence is a very crucial problem which needs robust solutions. The identification of the source camera of a digital image can authenticate the evidence provided by it. Kharrazi et al. [1] proposed a number of features that can be successfully used to identify the source camera of an image. In this thesis, application of various machine learning techniques, based on previously analyzed features for 3 classes, are experimented and analyzed. For cameras with lower volumes of images available, a modification to feature extraction is proposed which creates k times volume of extracted data. This modification is a block-based feature extraction inspired by feature extraction used in convolutional neural network followed by pooling which creates a matrix of features for every image instead of single-valued entries.

Keywords: ***Image Source***; ***Machine Learning***; ***Convolution***; ***classification.***

Contents

List of Figures

List of Tables

Acronyms

ANN Artificial Neural Network

SVM Support Vector Machine

CNN Convolution Neural Network

CFA Color filter array

ReLU Rectified linear unit

SPN Sensor pattern noise

Chapter 1

Introduction

In the real world, images are generally accepted as a proof of occurrence of an event. In this era because of enormous availability of image capturing devices and image sharing platform, digital images have become a part of our life. But since copying, downloading, forging or redistribution of images has become easier and easier because of the availability of powerful automated tools to create and manipulate a digital image. So there is need of tools to verify the authenticity of an image in order to reduce forgery and backtrack the origin of controversial images.

Different digital cameras use different pipeline architecture or hardware so there is a series of different artifacts left on the image during the image acquisition phase. These artifacts are the basis of our technique to identify source camera in blind fashion (without using watermarks).

During image capturing phase a digital camera performs a series of complex operation including focus of lenses, interpolation of the different color channel, Color Filter Array (CFA) configuration, brightness adjustment etc. Since these operations are non-invertible, so they leave traces of artifacts in the final image, and we can use these traces as a footprint in order to trace back the source camera.

There are different proposed approaches to exploit different traces of footprint like traces of CFA interpolation [2, 3], effects of lens distortion [4], traces due to auto white balance algorithm[5] and traces of dust particles on acquisition sensors.

Kharrazi et al.[1] identified a set of 34 features i.e. average pixel value (3 features), RGB pair correlation (3 features), neighbor distribution center of mass (3 features), RGB pair energy ratio (3 features), wavelet domain statistics (9 features), image quality metrics (13 features) that can be used for source identification. They tested the performance of these features for classification of images based on their origin. They achieved an accuracy of 93.42% for two cameras and accuracy of 88.02% for three cameras using multiclass SVM classifier.

In this thesis, we try to improve the performance by enhancing feature extraction method and classification techniques.

Chapter 2

Related Work

For any classification problem, there is a need to identify the properties or features that can be used to categorize the images. Kharrazi et al.[1] proposed 34 features for extraction from the pixel intensities of each image in order to identify the image processing method used by the specific camera model. These features are based on color filter array (CFA) configuration and the demosaicing algorithm, and the color processing or the transformation involved. Taking motivation from [1], Bayram et al. [6] were convinced that CFA configuration is enough to distinguish among the camera models. They established an interpolation function that detects the white pixels in the CFA. This creates a frequency spectrum of the probability maps, which is exploited to classify images. Choi et al. [7] used features from the previous works and reduced the feature array using stepwise discriminant analysis upon which iterative experiments are done with full, selective and random feature sets to test the classifier's performance.

A number of papers reviewing the work in [1] was published until 2010 when Li [3] and Kang et al. [8] polished the approach of classification based on sensor pattern noise. They attenuated the influence of details from scenes on the SPN, so as to improve the classifier's performance. In 2012 following [8], Liu et al. [9] made an empirical study of the effects of the enhancement of SPN on the performance statistics of different classifiers. Classification based on other features like footprints from lens aberration [10], chromatic aberration [11], wavelet statistics [12] and intrinsic radial distortion [13] was also introduced.

Chapter 3

Classification Techniques

3.1 Support Vector Machine

Support vector machine (SVM) is a supervised learning algorithm. It can be used for both classification and regression problem. Support vector machine is a mathematical model of drawing the best possible hyperplane to classify two classes of data in n-dimensional space (n = number of features).

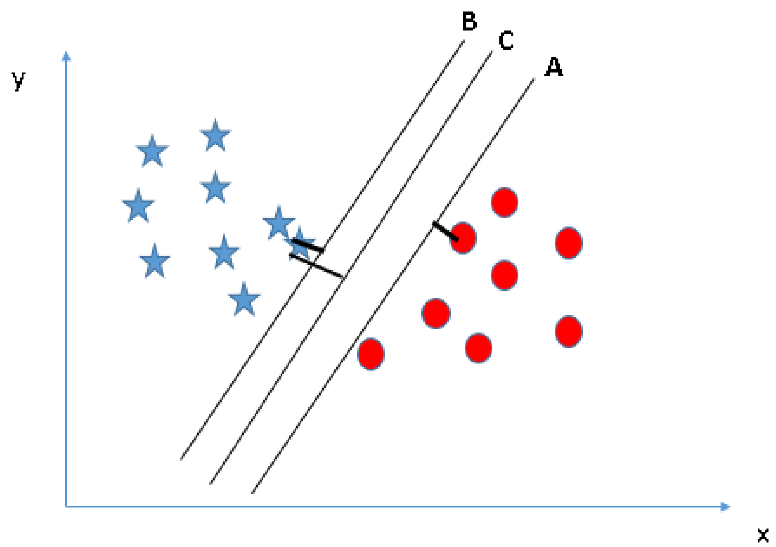

Figure 3.1: Finding most optimise hyperplane in SVM

Each feature represents value of a particular coordinate in a n-dimensional space, So we plot the feature sets as points in space and SVM try to find the best hyperplane to classify the feature sets into two classes. SVM finds the best hyperplane by maximizing distance between nearest data point (of both class) and the hyperplane.

SVM has the characteristic to ignore outliers and find the most optimal hyper-plane that classifies data into two classes.

3.1.1 Tuning of parameters of SVM

- **Kernel:** Kernel decides the type of hyperplane to be used. There are a variety of kernels available like *rbf* and *poly*; used for nonlinear hyperplane and *linear*; kernel used for linear hyperplane.

- **Gamma:** *gamma* is also known as kernel coefficient for *rbf* ,*poly* and *sigmoid* kernels. Higher value of *gamma* tries to fit the training data exactly, and so it leads to overfitting problem.

- **C:** This is the penalty parameter of the error term. It controls the trade-off between smooth decision boundary and classify the training point correctly. So for the better generalization of result we should look for effective combination of these parameters and avoid overfitting .

3.2 Decision Tree

Decision tree, a supervised learning algorithm, is mostly used in classification problems. In this technique we split the sample (population) into two or more homogeneous sets (subpopulation) based on a splitter in the input variable. Creation of subpopulation increases the homogeneity of the resultant subpopulation.

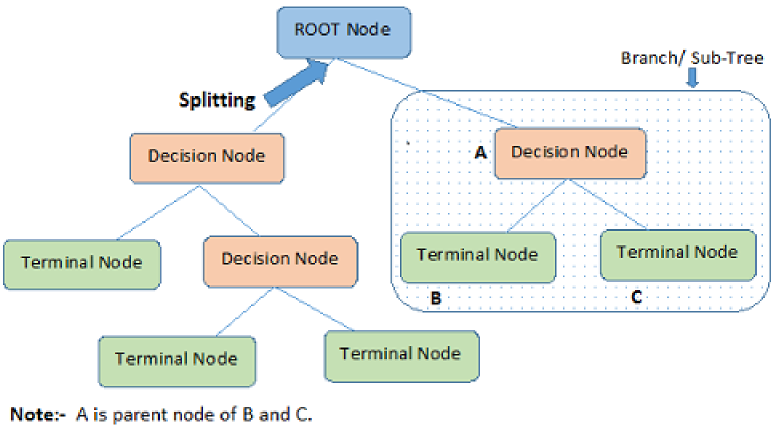

Figure 3.2: Decision Tree Classifier

In the above diagram, the root node represents the entire population, and splitting dictates the process of breaking a node into subnodes. Decision tree does not require lots of data cleaning because outliers and null values do not influence it. Data type constraints are absent because it can handle both numerical and categorical data. But Decision trees are very prone to overfitting.

3.3 Artificial Neural Network

Artificial neural network is a computational model of biological neurons. Information that passes through the network affect the weights of neurons because neural network changes (or learns) according to the input-output data.

Neural network has the remarkable ability to detect pattern in very complex data which goes unnoticed by humans or machine. So ANNs are used as random function approximation tools.

$$a = f(\sum_{i=0}^{N} w_i x_i) \tag{3.1}$$

Where, a is the output of neurons (which becomes input to the next layer neuron), f represents activation function, (there are many transfer function like linear, Sigmoid, Elliot, Symmetric Elliot, ReLU etc), w and x represents weight and input and b represents bias.

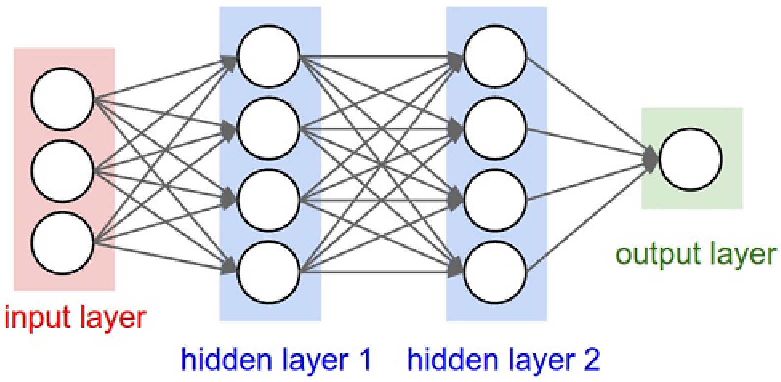

Figure 3.3: Architecture of Neural network

Network consists of a number of layers and each layer consists of few neurons. Output of a layer becomes input to the next layer except for the last layer (known as output layer). Output of the output layer is the result. All the layers between input and output layer are called hidden layers.

ANN uses supervised learning algorithm to train itself. Process of learning undertakes the following steps:

1. Initialize random weights to all neurons
2. Feed forward the input data.
3. Find error in result (i.e difference of output of feed-forward from the expected result).

4. Backpropagate error and modify weights in order to minimize the error.

5. Repeat the process until error is minimized to desired extent.

3.4 Convolutional Neural Network

CNNs, being hierarchical neural networks, are biologically-inspired variants of MLPs. CNNs vary in how convolution and subsampling layers are implemented and how the nets are trained. This framework of neural network is quite complex and requires detailed exploration.

3.4.1 Sparse connectivity

Neurons in CNN are sparsely connected to its preceding layer neurons. That is, each neuron need not be connected to all the neurons in the previous layer. This is possible by implementing convolution as a means to find the output from a layer where a value in the output only takes a fraction of the input values.

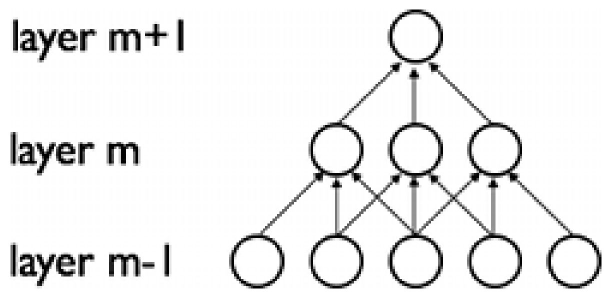

Figure 3.4: Sparse Connectivity

3.4.2 Convolution Layer

A convolution layer is parameterized by the input size, number of maps, kernel size, stride, and connection table. Each convolution layer has a depth d, where each sublayer along the depth has the same size rxc. Thus Each kernel has a filter of size kxxky. There can be a number of kernels acting upon the input layer which determines the depth of the output convolution layer.

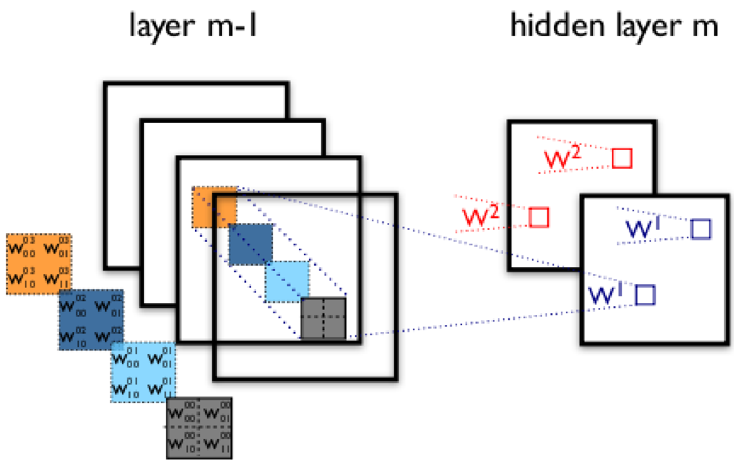

Figure 3.5: Convolution Layer

3.4.3 Activation Function

Like all artificial neural network, each convolution layer has an activation function that evaluates the result from the summer or in this case from the convolution. There are usually two types of activation functions.

1. Rectified Linear Unit: ReLU restricts all the values that are less than 0 and keeps the values unchanged if greater than zero.

$$f = max(0, x) \tag{3.2}$$

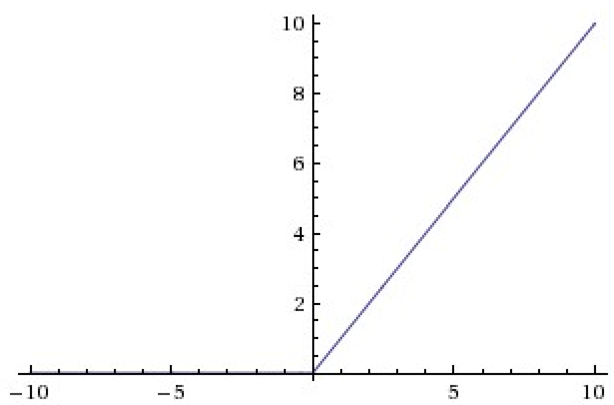

Figure 3.6: ReLU Activation Function

2. Continuous non-linear triggers: Though ReLU is quite popular there are other continuous activation functions that are continuous like logistic, tanH, Exponential Linear Unit and Gaussian.

3.4.4 Subsampling Layer

In the earliest days of the CNNs, for subsampling function merely nearby pixels were skipped prior to convolution. Then came the concept of Subsampling layer which negated the idea that subsampling has to be done at every convolution. Then came forward the concept of pooling which subsamples the image by using filters. These filters can be of three types

1. Average filter
2. Max filter
3. Min filter

Each of the above filters takes a patch of values from the input matrix and finds the required operated output for that patch while reducing the output resolution.

3.4.5 Architecture

Different forms of architecture has been derived for different classification problems.

- **LeNet Architecture:**As one of the most sought after architecture, this model consists of pairs of alternative convolution and subsampling layers followed by fully connected networks.

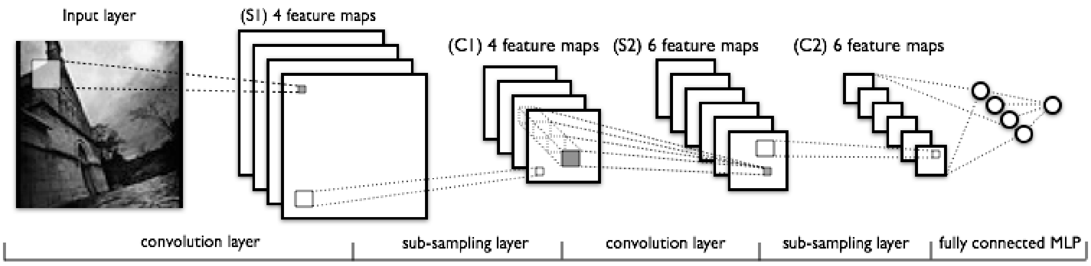

Figure 3.7: LeNet Architecture

3.5 Classification result optimization

For any classification problem there lies certain parameters that determines the correctness of the classification even if the accuracy remains high.

3.5.1 Overfitting

Overfitting occurs when machine learning algorithm tries to exactly fit the training data, that is to fit the data too well.

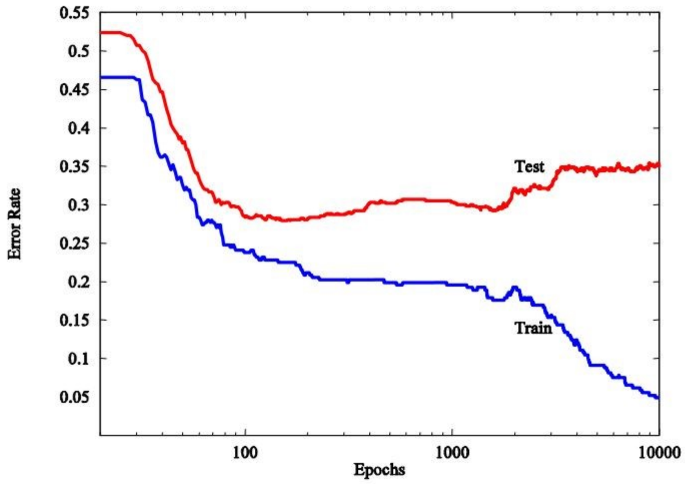

Figure 3.8: Graph showing behaviour of error due to overfitting

The above figure clearly shows that with increase in training epoch, error rate of training and testing data decreases but after some iterations learning algorithm starts to overfit on training data, so resultant accuracy on training data starts to increase. But that on testing data starts to decrease. It clearly suggests that overfitting is not good for generalisation,hence generalised classifier should not be overfitted.

3.5.2 Underfitting

Underfitting occurs when machine learning algorithm does not capture underlying trends in the training data. Intuitively, it occurs when it does not fit the training data well enough. So the prediction result shows high error on both training and testing data.

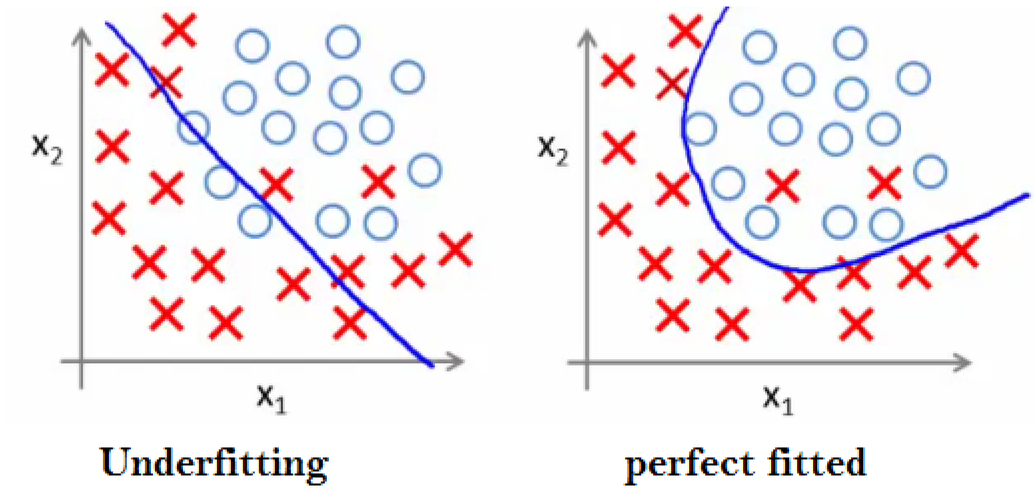

Figure 3.9: Underfitted and perfectly fitted hyperplane

3.5.3 Decision Criteria

For the best classification results we must have to find optimal situation lie in between underfitting and overfitting as shown in figure below.

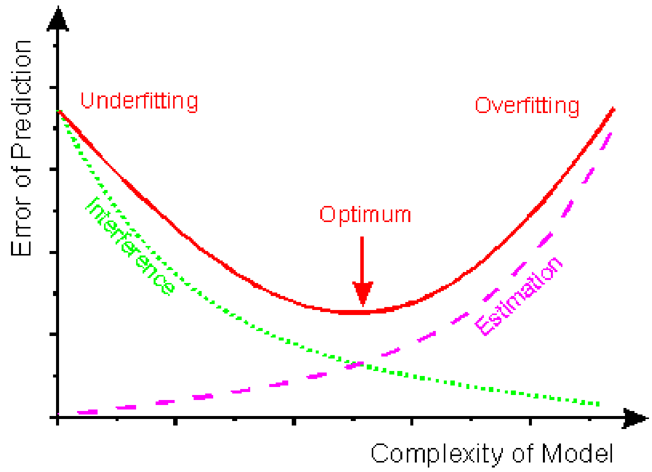

Figure 3.10: Decision criteria to reduce over fitting as well as underfitting

Chapter 4

Proposed Method

Different machine learning algorithms are applied over previously proposed classical features (12 among 34 features) and over its modified version. The results are then systematically obtained and compared.

4.1 Candidate Features

For our paper we take the features proposed in [1] as our basis for experimentation. The variation of features proposed by Kharrazi makes each tuple in the data set unique. Application of different classification techniques on a detailed dataset can provide an appropriate comparison of the techniques. Features used as base for experiments are listed as follows:

4.1.1 Average pixel value

This feature is the average values of the grayscale intensities of the image. This is attributed to each of the color band of the image and hence a total of three features.

$$\begin{aligned} A_r &= \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} Image_r(i,j)}{M \times N} \\ A_g &= \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} Image_g(i,j)}{M \times N} \\ A_b &= \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} Image_b(i,j)}{M \times N} \end{aligned} \tag{4.1}$$

4.1.2 Energy Ratio

RGB pair energy ratio is used in the process of white point correction which is an integral part of camera pipeline. This energy ratio is done for three pairs GR, GB, BR.

$$E_1 = \frac{|G|^2}{|B|^2} \quad E_2 = \frac{|G|^2}{|R|^2} \quad E_3 = \frac{|B|^2}{|R|^2} \tag{4.2}$$

4.1.3 Neighbour distribution center of mass

This feature is extracted by first calculating neighbor values for each gray scale value. This value is the sum of the number of pixels which differ from the pixel in question by 1 or -1.

$$neigh_i = freq_{i-1} + freq_{i+1} \tag{4.3}$$

The distribution provides an insight to the sensitivity of the camera to different intensity levels. Then the center of mass of the neighbor distribution is then calculated using the following function:

$$Neigh_{COM} = \frac{\sum_0^{255} i * neigh_i}{\sum_0^{256} neigh_i} \tag{4.4}$$

Then 3 features are attributed by finding the center of mass of the neighbor distribution of 3 color bands.

4.1.4 Correlation

The camera structure impacts some variation to the correlation among the color bands. Each camera model can have unique camera structure and hence different correlation values. RGB pair correlation consists of 3 values for 3 pairs of correlation namely RG, RB, BG.

$$r_{RG} = \frac{\sum^M \sum^N (R_{mn} - \bar{R})(G_{mn} - \bar{G})}{\sqrt{(\sum^M \sum^N (R_{mn} - \bar{R})^2)(\sum^M \sum^N (G_{mn} - \bar{G})^2)}} \tag{4.5}$$

4.2 Classical Approach

12 features obtained through previously proposed methods are extracted over 143 images of 512 x 512 resolution for each class.

Different classification techniques for the given problem has been applied for 2 classes and 3 classes. Initial expectation remains that accuracy for 2 class classification will be considerably greater than that of the latter.

4.3 Modified approach: Block based feature Extraction

The mode of feature extraction in the above approach is single valued or only one value for the whole image. According to [15] natural images have a *stationary* property that dictates that features trained from a part of the image will show similar statistics if trained on other parts of the image. Following this concept we assumed that its similar case is true, that is feature extracted from one part of the image will show similar results with that extracted from other parts of the image. Using working of the convolution layers as a motivation we used the concept of convolution to extract feature values from every valid continuous patch of the image to form a matrix of values for each feature called feature map.

The convolution is followed by pooling in which we use average function to find aggregate values in the feature map. These two operations ensures that each value in the feature map contributes to multiple patches in the image.

Chapter 5

Experimental Results

The dataset used for the input for feature extraction was created using 3 cameras namely Nikon, Sony and Canon. Each scene in was recorded by all the cameras to maintain statistical order. This dataset contains 429 images (143 images for each camera).

5.1 Classical Approach

5.1.1 2-way Classification

We applied different classification technique on 2 class of images.

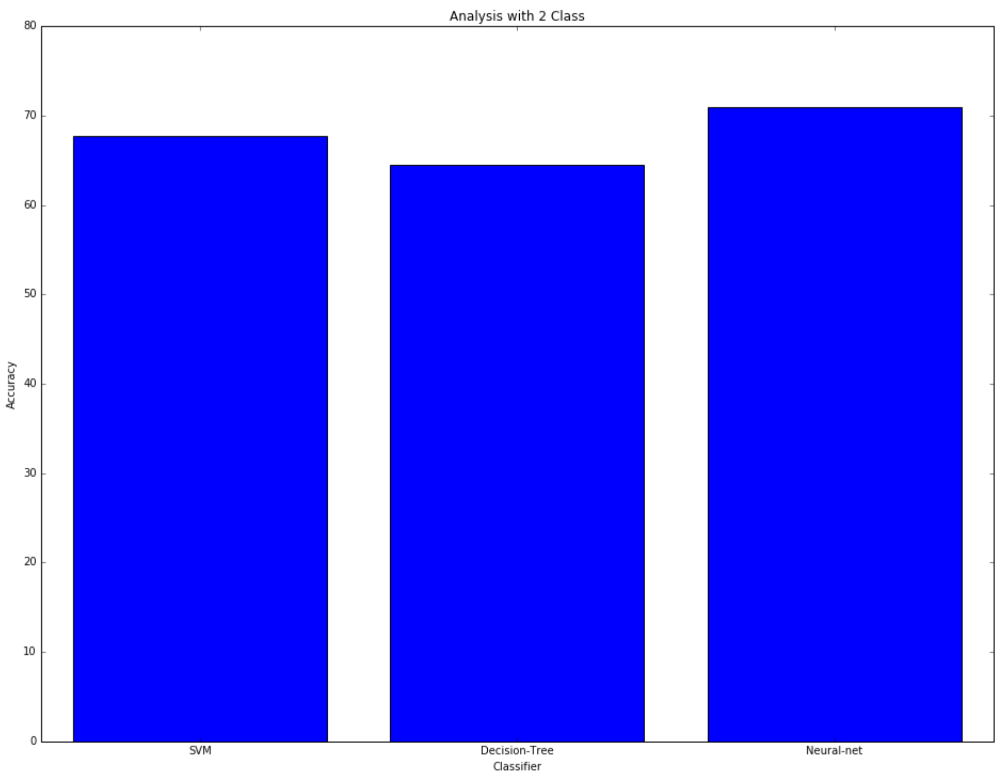

Figure 5.1: Analysis with 2 classes

- SVM classifier : 67.74 %
- Decision tree Classifier : 64.51 %

- Neural Network : 70.96 %

5.1.2 3-way Classification

We applied different classification technique on 3 class of images.

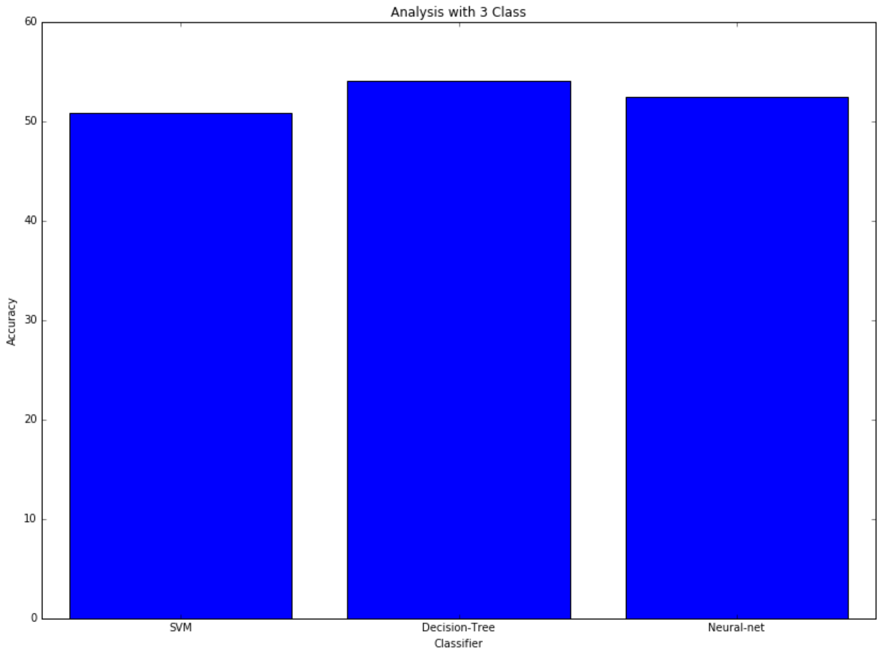

Figure 5.2: Analysis with 3 classes

- SVM classifier : : 50.81 %
- Decision tree Classifier : 54.09 %
- Neural Network : 52.46 %

5.2 Block Based Approach

5.2.1 2-way Classification

We applied different classification technique on 2 class of images.

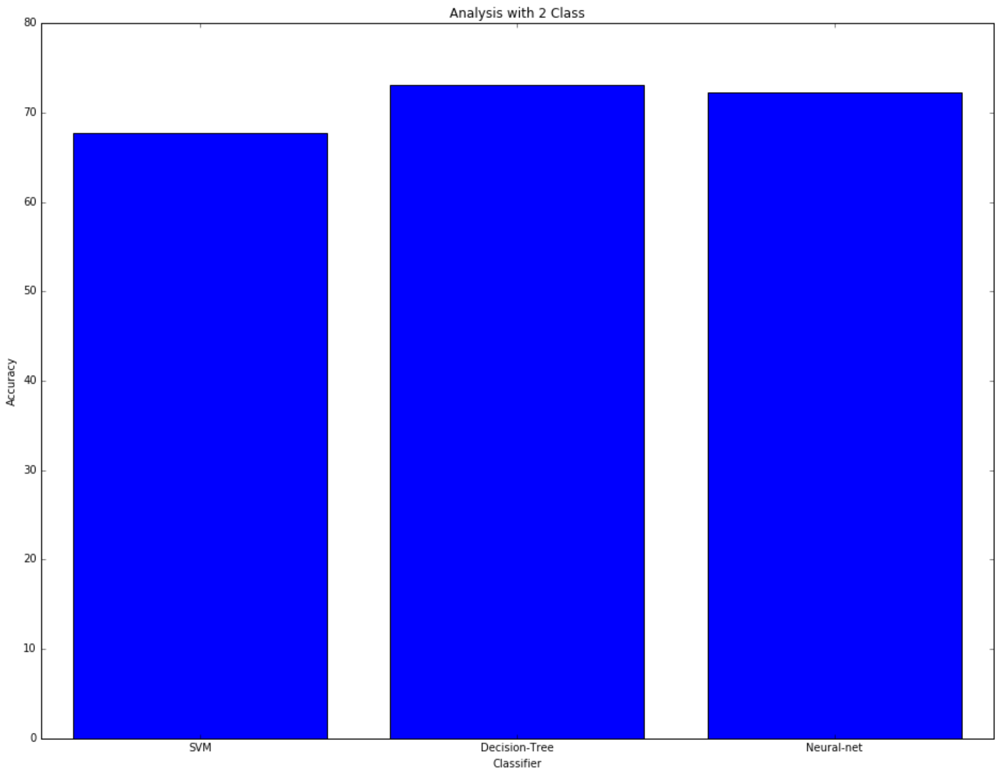

Figure 5.3: Analysis with 2 classes

- SVM classifier : 67.69 %
- Decision tree Classifier : 73.07 %
- Neural Network : 72.30 %

5.2.2 3-way Classification

We applied different classification technique on 3 class of images.

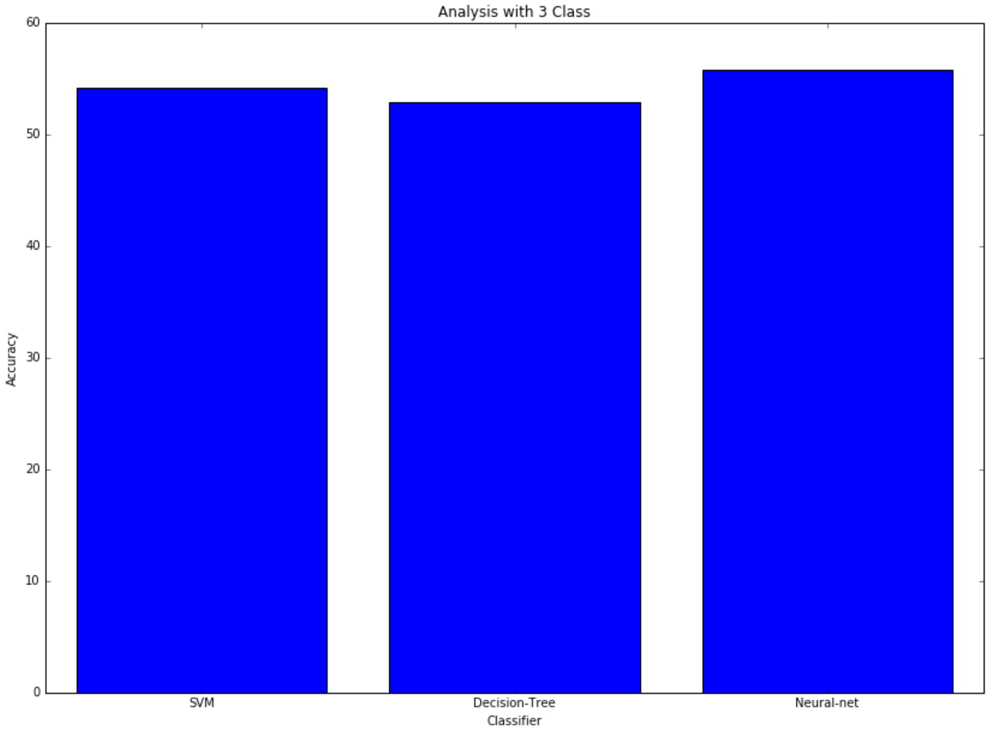

Figure 5.4: Analysis with 3 classes

- SVM classifier : : 54.17 %
- Decision tree Classifier : 52.91 %
- Neural Network : 55.83 %

Feature extraction from an image is a computationally expensive job. For computational purposes we used High Performance Computing (HPC) machine with 12 CPU cores and Matlab parallel programming tool. We found that average time to extract all 12 features from a 512x512 image took 8 minutes/image, and with the usage of parallel programming on Matlab with 12 CPU we could reduce it to 43 seconds/image.

From the simulations shown in fig[], it can be clearly identified that Artificial Neural Network is the most suitable classifying technique for 2 class classification. With 30 testing tuples out of 286, ANN gives a maximum accuracy of 70.96%.

Comparison of classification on 2 class image			
Feature Type	SVM	Decision tree	ANN
Classical approach	67.74%	64.51%	70.96%
Block Based approach	67.69%	73.07%	72.30%

Table 5.1: Comparison of 2 class classification

For 12 features taken as base, we get a maximum accuracy of 70.96% for 2 class and 54.09% for three class using a dataset of 429 images (143 images of each class) and classical feature extraction approach. Using block based feature extraction motivated from CNNs, the accuracy over the same dataset could be increased up to 73.07% for 2 class and 55.83% for 3 class. The magnitude of increase depends upon the filter size, stride value during pooling and so on. For smaller filter size number of patches increases linearly and hence the size of the feature map.

Comparison of classification on 3 class image			
Feature Type	SVM	Decision tree	ANN
Classical approach	50.81%	54.09%	52.46%
Block Based approach	54.17%	52.91%	55.83%

Table 5.2: Comparison of 3 class classification

Chapter 6

Conclusion and Future Work

We successfully extracted features from 429 images using classical as well as modified approach. The classifiers were trained with the extracted datasets and comparisons were made.

Our proposed modification to the feature extraction techniques either provides equal or greater accuracy results. Future work in this direction would include experimentation of the implication of various stride and filter size on classification results to achieve concrete inference from this block based approach.

Block based approach, which can be only applicable for natural images, is an alternative to imbalanced dataset, when certain classes have very fewer images for training. Underfitting problems can solved by the extraction of multiple dataset from an image instead of single values. This method should give better results than that of oversampling by duplication because it creates new but valid data by capturing different variations of the features in the given image. Future work in this direction would include comparison of experimental results upon training of imbalanced and insufficient data sets.

Classification with 12 features shows drastic decrease in accuracy with increase in number of classes. With evolutionary success of Convolutional Neural Nets, CNNs may be able to improve accuracy upon multiple class at which other techniques are not much successful.

Bibliography

[1] M. Kharrazi , H. T. Sencar and N. Memon, ”Blind source camera identification”, Proc. ICIP, 2004

[2] J. Adams, K. Parulski and K. Sapulding, “Color Processing in Digital Cameras,” IEEE Micro, Vol. 18, No.6, 1998

[3] C.-T. Li, ”Source camera identification using enhanced sensor pattern noise”, IEEE Trans. Inform. Forensics Security, vol. 5, no. 2, pp. 280-287, 2010

[4]] Liu, Bei-bei, Xingjie Wei, and Jeff Yan. ”Enhancing sensor pattern noise for source camera identification: an empirical evaluation.” In Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security, pp. 85-90. ACM, 2015.

[5] Z. Deng, A. Gijsenij, and J. Zhang, “Source camera identification using auto-white balance approximation,” in IEEE International Conference on Computer Vision (ICCV), 2011.

[6] S. Bayram , H. T. Sencar and Î. Memon, ”Source camera identification based on CFA interpolation”, IEEE Int. Conf. Image Processing, pp. 69-72, 2005

[7] Choi, Kai San, Edmund Y. Lam, and Kenneth KY Wong. ”Feature selection in source camera identification.” In Systems, Man and Cybernetics, 2006. SMC’06. IEEE International Conference on, vol. 4, pp. 3176-3180. IEEE, 2006.

[8] X. Kang , Y. Li , Z. Qu and J. Huang, ”Enhancing source camera identification performance with a camera reference phase sensor pattern noise”, IEEE Trans. Inf. Forensics Sec., vol. 7, no. 2, pp. 393-402, 2012

[9] Liu, Bei-bei, Xingjie Wei, and Jeff Yan. ”Enhancing sensor pattern noise for source camera identification: an empirical evaluation.” In Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security, pp. 85-90. ACM, 2015.

[10] San Choi, Kai, Edmund Y. Lam, and Kenneth KY Wong. ”Automatic source camera identification using the intrinsic lens radial distortion.” Optics express 14, no. 24 (2006): 11551-11565.

[11] Van, Lanh Tran, Sabu Emmanuel, and Mohan S. Kankanhalli. "Identifying source cell phone using chromatic aberration." In Multimedia and Expo, 2007 IEEE International Conference on, pp. 883-886. IEEE, 2007.

[12] Wang, Bo, Yiping Guo, Xiangwei Kong, and Fanjie Meng. "Source camera identification forensics based on wavelet features." In Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP'09. Fifth International Conference on, pp. 702-705. IEEE, 2009.

[13] San Choi, Kai, Edmund Y. Lam, and Kenneth KY Wong. "Automatic source camera identification using the intrinsic lens radial distortion." Optics express 14, no. 24 (2006): 11551-11565.

[14] Baroffio, Luca, Luca Bondi, Paolo Bestagini, and Stefano Tubaro. "Camera identification with deep convolutional networks." arXiv preprint arXiv:1603.01068 (2016).

[15] D. Ruderman. The Statistics of Natural Images. Network: Computation in Neural Systems, 5(4):517-548, 1994. 2