Final project

**Blind image Source Identification Using Machine Learning Technique**

horizontal line

# Placeholder image

|  |
| --- |
| [Name] | [Course Title] | [Date] |

|  |
| --- |
| [Report Subtitle] |

**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. proposed a total of 34 features[1] 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 extaction used in convolutional neural network followed by pooling which creates a matrix of features for every image instead of single-valued entries.

**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, Photos becomes a part of our life. But since copying, downloading, forging or redistribution of images becomes easier and easier because of the availability of automated powerful tools to create & manipulate a digital image. So there is need of tools to verify the authenticity of an image in order to reduce forgery and backtrack origin of controversial images.

Different digital cameras use different pipeline architecture or hardware so a series of different artifacts left on the image during 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 digital camera performs a series of complex operation including focusing using lenses to interpolation of the different color channel, Color Filter Array (CFA), brightness adjustment etc. Since these operations are noninvertible, So they left traces of artifacts in the final image, and we can use these traces as a footprint in order to trace back source camera.

There are different approaches based on different traces of footprint have been proposed. Example : Using traces of CFA interpolation ([3][4]),effect of lens distortion([6]), traces due to auto white balance algorithm([7]) and exploiting traces of dust particle on acquisition sensors.

Kharrazi et al. identified a set of 34 features ( average pixel value(3 features), RGB pair Correlation(3 features), Neighbour distribution center of mass(3 feature),RGB pair energy ratio(3 feature),Wavelet domain statistics (9 feature),Image Quality Metrics(13 features)) that can be used for source identification. They tested the performance of these features for classification of the image based on their origin. They found the 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 of result by enhancing feature extracting method & classification techniques.

**RELATED WORKS**

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. His 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. [2] 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. [3] 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 Chang–Tsun li [4] and Kang et al. [5] polished the approach of classification based on sensor pattern noise by attenuating the influence of details from scenes on the SPN, so as to improve the classifier’s performance. In 2012 following [5], Liu et al. [6] 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 [7], chromatic aberration [8], wavelet statistics [9] and intrinsic radial distortion [10] was also introduced.

**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:

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

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

3. ***Neighbour distribution center of mass***This feature is extracted by first calculating pixel 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. The distribution provides an insight to the sensitivity of the camera to different intensity levels. Then 3 features are attributed by finding the center of mass of the neighbor distribution of 3 color bands.

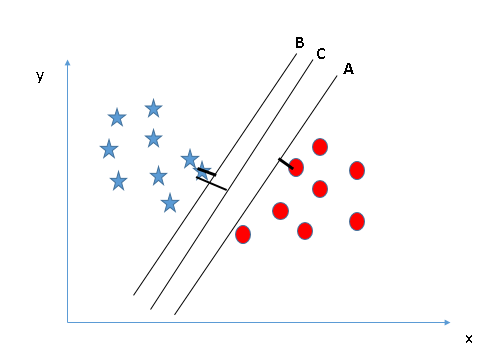
4. ***Correlation***The camera structure impacts some variation to the correlation among the color bands. Each camera model can have unique camera structure and hence different correlation values. RGB pair correlation consists of 3 values for 3 pairs of correlation namely RG, RB, BG.

**Classification Techniques:**

Support Vector Machine:

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

Each feature represent value of a particular coordinate in n-dimensional space, So we plot data items as a point in space and SVM try to find the best hyperplane to classify data into two classes.



SVM finds the best hyperplane by maximizing distance between nearest data point (of both class) and the hyperplane (as shown in above fig).

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

**Tuning of parameters of SVM**

**Kernel :** Kernel decides type of hyperplane. There are a variety of kernels available like ‘rbf’ and “poly” used for nonlinear hyperplane and ‘linear’ kernel used for linear hyperplane.

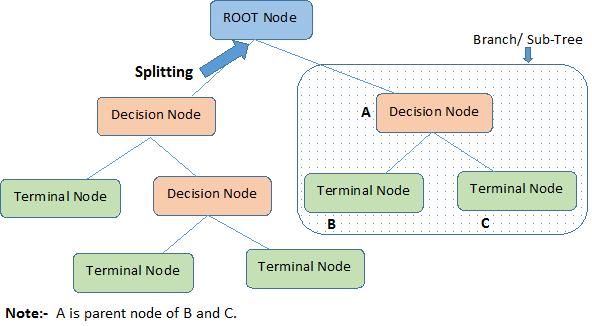
**Gamma** : ‘gamma’ also known as kernel coefficient for ‘rbf’ ,’poly’ and ‘sigmoid’ kernels. Higher value of gamma try to fit the training data exactly, and so it leads to overfitting problem.

**C** : 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 .

Decision Tree :

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



Root Node: Represent entire population.

Splitting: Process of breaking a node into subnodes.

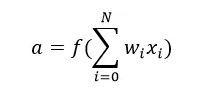
Decision tree does not require lots of data cleaning because it is not influenced by outliers and null values, and also Data type constraints is not there because it can handle both numerical and categorical data. But Decision tree are very prone to overfitting.

Artificial Neural Network:

Artificial neural network is a computational model of biological neurons. Information that passes through the network effect the weights of neurons because neural network changes (or learn) based on the input-output data.

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

A neuron is building block of ANN. For a single neuron



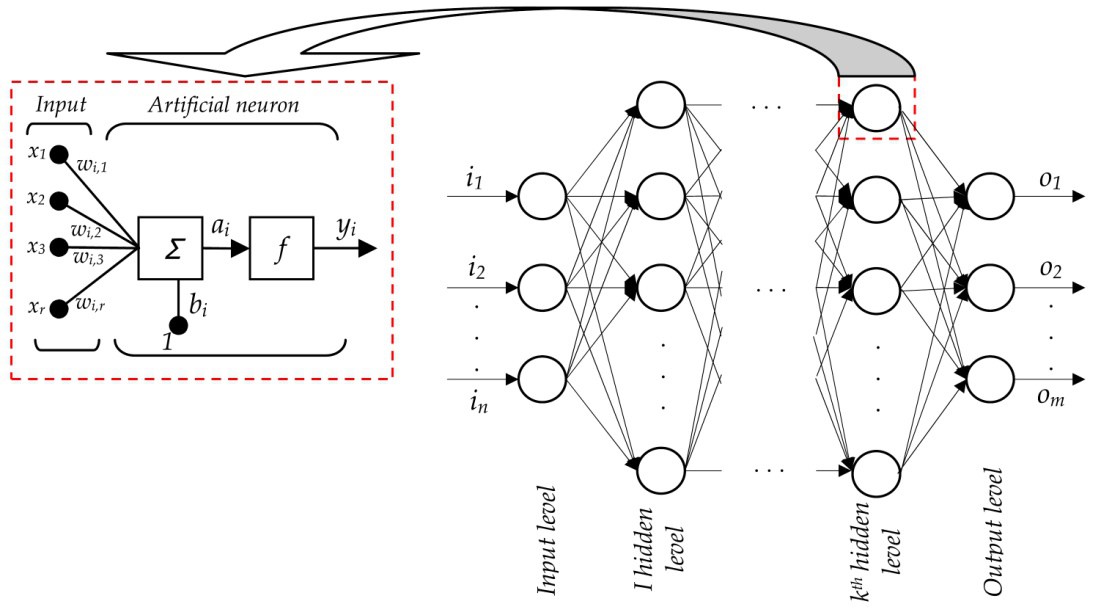
Where,

· **a** is output of neuron (become input of next layer neuron)

· f() represent activation function, (there are many transfer function like linear, Sigmoid, Elliot, Symmetric Elliot,ReLU etc)

· w and x represents weight and input

· b represent bias



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

ANN is uses supervised learning algorithm to train itself. Process of learning

· Initialize random weights to all neurons

· Feed forward the input data .

· Find error in result (i.e difference of expected result – output of feed-forward).

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

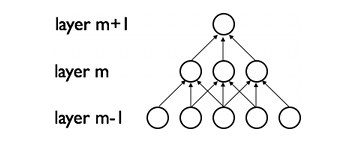
· Repeat the process until error minimized to desired extent.

Convolutional Neural Network:

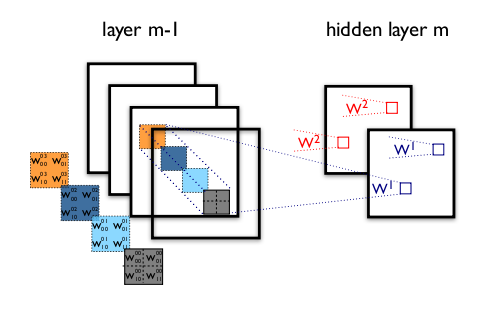
CNNs are hierarchical neural networks are biologically-inspired variants of MLPs. CNNs vary in how convolutional and subsampling layers are realized and how the nets are trained.

Understanding Convolutional Neural Net:

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.

2. Convolutional Layer

A convolutional layer is parameterized by the input size, number of maps, kernel size, stride, and connection table. Each convolutional layer has a depth d, where each sublayer along the depth has the same size (r,c). This Each kernel has a filter of size (kx,ky). There can be a number of kernels acting upon the input layer which determines the depth of the output convolutional layer.

3. Activation Function

Like all artificial neural network, each convolutional 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.

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

f(x)= max(0,x)



b) 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.

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:

a) Average filter b) Max filter c) 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

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 2 pairs of alternative convolutional and subsampling layers followed by fully connected networks.

**AlexNet Architecture:**

**Implementation**

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

The features are computed in the following fashion :-

1. Average Pixel Value:-

f = )/(m x n)

1. RGB pair Energy Ratio:-

1. Neighbor Distribution COM:-
2. Correlation:-

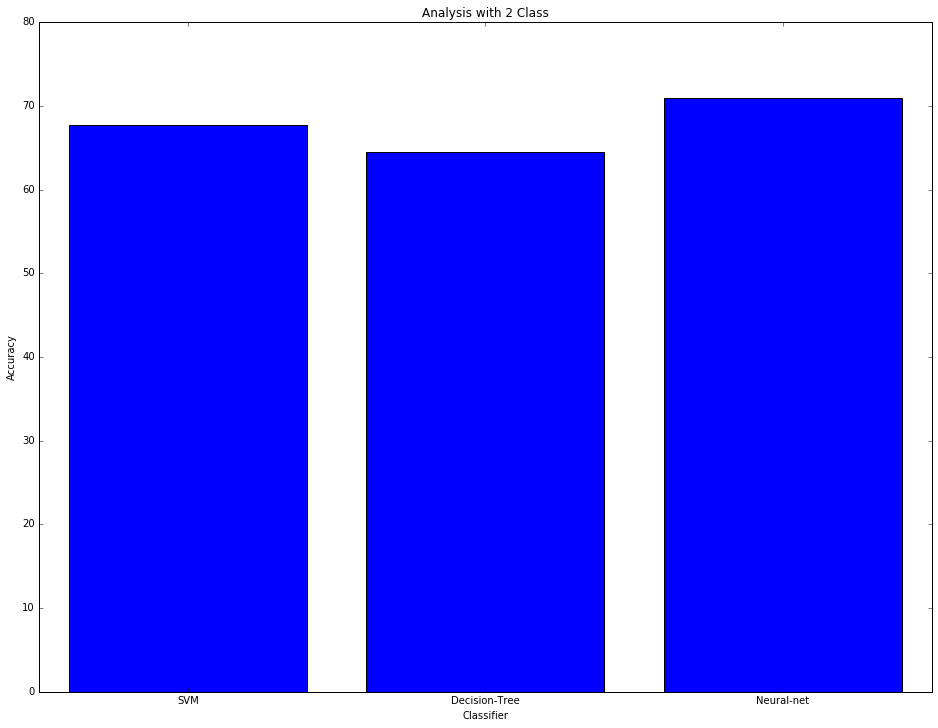
**Classical Approach**:

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 latter.

1. CLASSIFICATION OF 2 CLASSES:

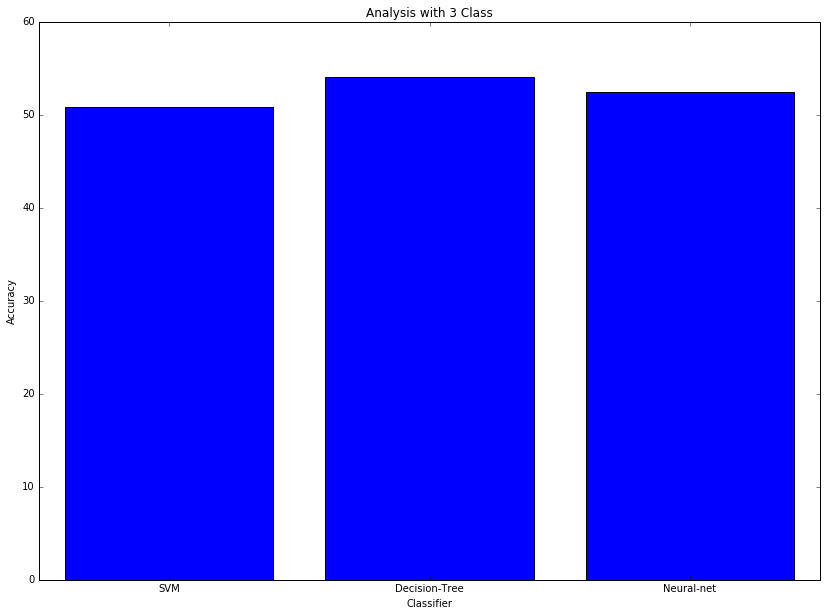
We applied different classification technique on 2 class of images.



* SVM classifier : 67.74 %
* Decision tree Classifier : 64.51 %
* Neural Network : 70.96 %

1. CLASSIFICATION OF 3 CLASSES:

We applied different classification technique on 3 class of images.



* SVM classifier : 50.81 %
* Decision tree Classifier : 54.09 %
* Neural Network : 52.459 %

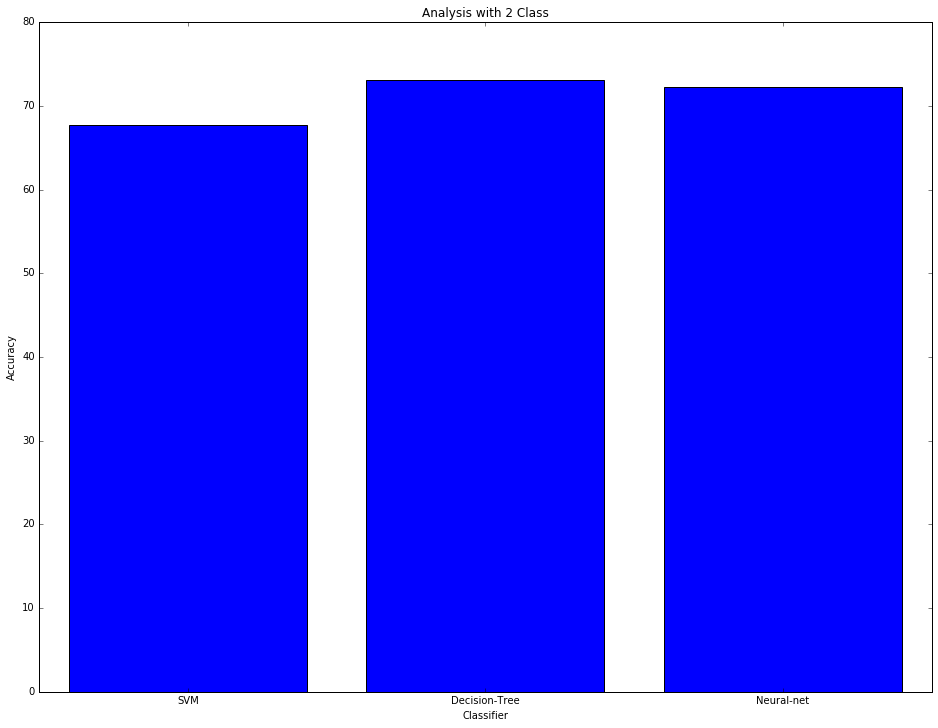
**Modified approach: Block based feature Extraction**

The mode of feature extraction in the above approaches till now was single valued or only one value for the whole image. According to [] 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 contrapositive 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 convolutional 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. This two operations ensures that each value in the feature map contributes to multiple patches in the image.

1. CLASSIFICATION OF 2 CLASSES:-

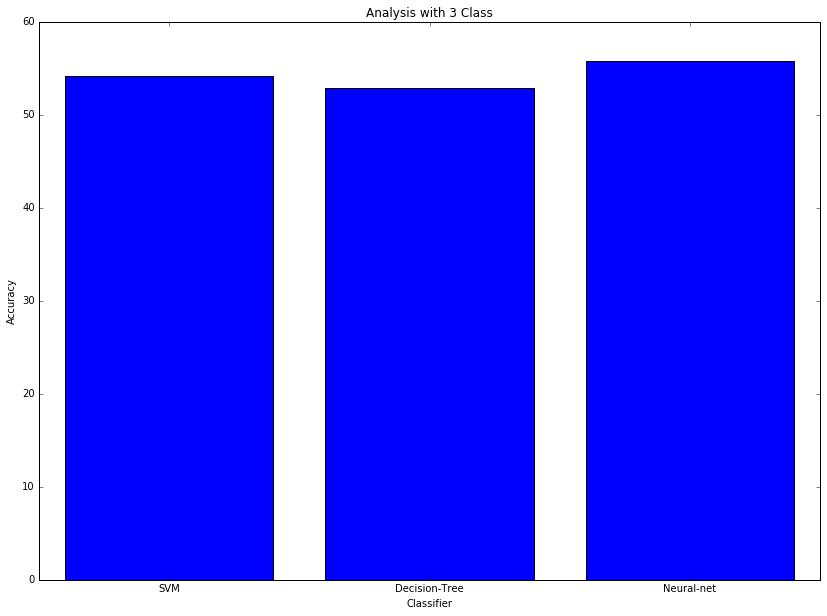
We applied different classification technique on feature extracted through block based technique on 2 class of images.



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

1. CLASSIFICATION OF 3 CLASSES:-

We applied different classification technique on feature extracted through block based technique on 2 class of images.



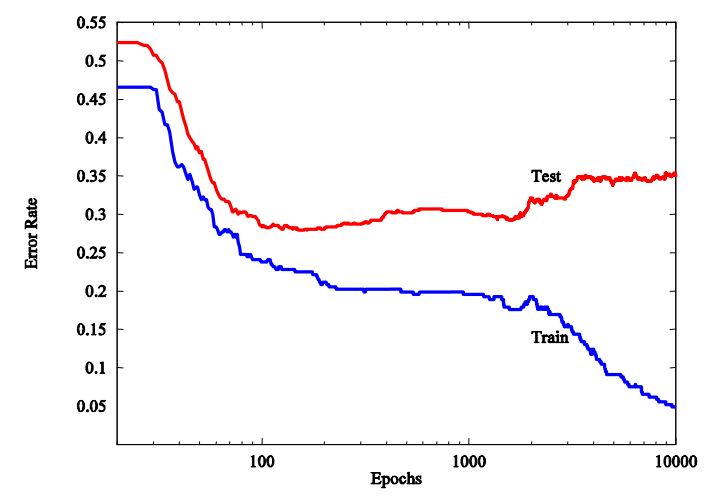
* SVM classifier : 54.17 %
* Decision tree Classifier : 52.91 %
* Neural Network : 55.83 %

**Experimentation**

The dataset used for the input for feature extraction was created using 3 cameras namely Nikon(model no), Sony(model no), and Canon(model no). Each scene in an image was recorded by all the cameras to maintain statistical order. This dataset contains 429 images (143 images for each camera).

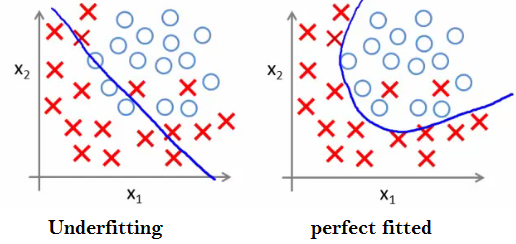
For any classification problem there lies certain parameters that determines the correctness of the classification even if the accuracy remains high.

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

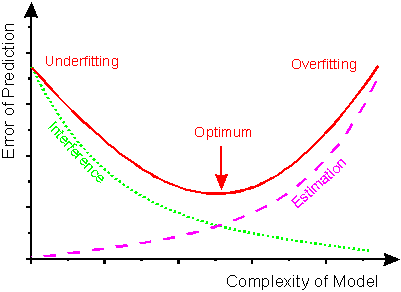


fig() clearly shows that with increasing training epoch error rate of training and testing data decreases but after some iterations learning algorithm starts to overfit on training data, so resultant accuracy on training data starts to increase. But that on testing data starts to decrease. It clearly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

1. 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.



So, for the best classification results we must have to find optimal situation lie in between underfitting and overfitting as shown in fig()



Feature extraction form an image is a computationally very 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 an 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.

**Results and Conclusions**

**Reference:**

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

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

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

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

[5] 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

[6] 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.

[7] San Choi, Kai, Edmund Y. Lam, and Kenneth KY Wong. "Source camera identification using footprints from lens aberration." In *Electronic Imaging 2006*, pp. 60690J-60690J. International Society for Optics and Photonics, 2006.

[8] 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.

[9] 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.

[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] 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.

[12] Huang, Yonggang, Jun Zhang, and Xinkai Lan. "Source Camera Identification with Imbalanced Training Dataset." *International Journal of Database Theory and Application* 9, no. 2 (2016): 205-214.

[13] Khurana, Mukul, Vivek Sharma, Pulkit Mehndiratta, and Shelly Sachdeva. "Source Camera Identification for Mobile Phones using EXIF Data and Lens Features." *International Journal of Computer Science Issues (IJCSI)* 12, no. 3 (2015): 225.

[14] Alles, Erwin J., Zeno JMH Geradts, and Cor J. Veenman. "Source camera identification for heavily jpeg compressed low resolution still images." *Journal of forensic sciences* 54, no. 3 (2009): 628-638.

[15] Huang, Y. Longand Y. "Image based source camera identification using demosaicing." In *Proc. of MMSP*. 2006.

[16] Lin, Xufeng, and Chang-Tsun Li. "Enhancing sensor pattern noise via filtering distortion removal." (2016).

[17] Li, Ruizhe, Constantine Kotropoulos, Chang-Tsun Li, and Yu Guan. "Random subspace method for source camera identification." In *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*, pp. 1-5. IEEE, 2015.

[18] Al-Ani, Mustafa, Fouad Khelifi, Ashref Lawgaly, and Ahmed Bouridane. "A novel image filtering approach for sensor fingerprint estimation in source camera identification." In *Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on*, pp. 1-5. IEEE, 2015.

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