Image Source Identification using Machine Learning Techniques

Vaibhav Gupta

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Image Source Identification using Machine Learning Techniques

Thesis submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Vaibhav Gupta

(Roll: 111CS0069)

under the guidance of

Prof. Ruchira Naskar

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

May 2015

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India.

May 8, 2015

Certificate

This is to certify that the work in the thesis entitled *Image Source Identification using Machine Learning Techniques* by *Vaibhav Gupta*, bearing roll number *111CS0069*, is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Ruchira Naskar
Supervisor

Acknowledgment

The successful completion of the project would have been inconceivable, notwithstanding the kind support of numerous people and also this institute. I would like to express my deepest sense of gratitude to every one of them. Foremost, I would like thank my supervisor for the project, Prof. Ruchira Naskar, Department of Computer Science and Engineering, National Institute of Technology, Rourkela, for her incalculable contribution to the project. She stimulated me to work on the topic and provided valuable information which helped in the completion of the project. I would also like to acknowledge her exemplary guidance, monitoring and constant encouragement throughout the course of this thesis.

I am indebted to all the professors of the Department of Computer Science and Engineering, NIT Rourkela for instilling in me the basic knowledge about the fields that greatly benefited me while working on the project.

I also thank my friends and peers who extended their help and support whenever needed. Their contribution has always been significant. Finally, I would like to take this opportunity to thank my family, who have always been a source of inspiration and motivation for me, and also for the love they have provided in stressful periods, which has been a guiding force for the completion of the thesis.

Vaibhav Gupta

Abstract

In digital forensics, identifying the model of the camera that was used to obtain a given digital image, presents an interesting problem. In a scene of a crime, it may help the forensic investigator to identify the camera that has taken the digital image used as evidence. The properties of the image given in the Exchangeable Image File Format are easy to modify and thus, cannot be authenticated. The exponential increase in digital image capturing devices in the past few years has further added to the problem. In this work, we aspire to provide experimental results and compare the accuracies generated by a number of two-class classifiers when fed with multiple features of images captured by two camera models, which help in the identification of the source camera model of the given image. Hence, to improve the efficiency of source identification, we strive to improve upon the classifiers in use.

***Keywords*:** Digital Forensics, Image Source Identification, Camera Model, Feature, Two-Class Classifier

Contents

List of Figures

Chapter 1

Introduction

In the non-digital space, an image has by, and large been acknowledged as a confirmation of happening of the portrayed occasion. In the digital age of the day, generation and control of digital images are made straightforward by an ease of equipment and programming devices that are effortlessly and broadly accessible. Therefore, we are quickly arriving at a circumstance where nobody can hold the credibility and honesty of digital images for granted. This pattern undermines the validity of digital images introduced as confirmation in a court of law, as news items, as a component of a medicinal record or as monetary reports since it might never again be conceivable to recognize whether a given digital image is the first or a (maliciously) adjusted form or even a depiction of genuine events and articles.
To address these immediate problems, we try to look for authentic answers to the following questions:

- Is this an original image or was it generated by an amalgamation of different images?
- Does this image genuinely depict the actual scenario or has it been modified to defraud the observer?
- What is the history of processing of the image?
- What portions of the image have been transformed through processing and up to what extent?
- Has the image been acquired by a source manufactured by vendor X or Y?

- Has the image originated from source X as claimed?

These inquiries are only a small sample of problems confronted often by examination and administration agencies. Nonetheless, there is an absence of strategies that can offer assistance in discovering legitimate solutions. Albeit advanced watermarks as a device to authenticate the images have been proposed , it's a fact that the larger part of the images caught today don't incorporate a digital watermarking. Furthermore, this circumstance is prone to proceed for the predictable future. Subsequently without broad selection of digital watermarking, we trust it is basic to create strategies that can assist us in making explanations about the starting point, accuracy and description of digital images.

The issues confronted in Image Forensics are to a considerable degree troublesome and maybe even difficult to define in a simple and basic way. Through this work, we try to solve the problem of determination of the camera model that was used to capture the given image. This question is a recurring one during investigations [1]. Albeit description about the model of the camera, date, time and type of the image are all stored in the header of the JPEG image, it is impossible to authenticate them. The underlying assumption for the success of blind image authentication techniques is that all images produced by a digital camera will exhibit certain characteristics regardless of the captured scene, which are unique to that camera, due to its proprietary image formation pipeline [1].

1.1 Formation of an Image in Digital Cameras

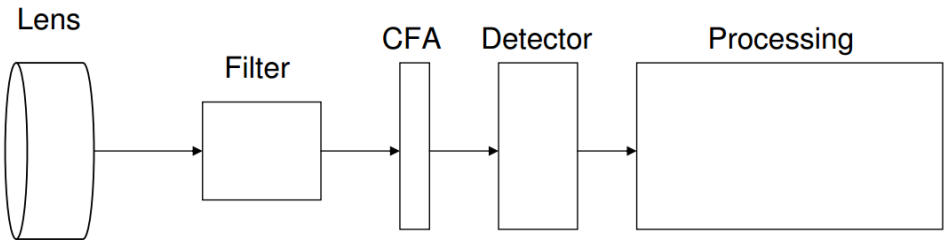

Figure 1.1: Camera pipeline showcasing the processing stages

G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R
B	G	B	G	B	G

Figure 1.2: RGB values through CFA pattern

M	G	M	G	M	G
C	Y	C	Y	C	Y
M	G	M	G	M	G
C	Y	C	Y	C	Y
M	G	M	G	M	G
C	Y	C	Y	C	Y

Figure 1.3: YMCA values through CFA pattern

The general sequence and structure of the processing phases of image formation pipeline in a digital camera remains to be much similar to every other digital camera (notwithstanding the proprietary nature of the underlying technology). A filter (anti-aliasing filter being the one of maximum importance) is applied to the light entering the camera through the lens, followed by focusing the incoming light onto an array of charge-coupled device (CCD) elements, i.e., pixels. The CCD array is the most expensive and central component of a digital camera. Each element of CCD array that senses light integrates the incident light over the entire spectrum and procures an electric signal representation of the scenario. Because of the mostly monochromatic nature of every CCD element, separate CCD arrays for each colour component are required for capturing colour images. Be that as it may, because of expense contemplations, in most digital cameras, only a single CCD array is utilized by orchestrating them as a part of an example where every element has a different spectral filter, commonly one of red, green or blue (RGB). This applied mask in front of the sensor is called the Colour Filter Array (CFA). Thus, only one band of

wavelengths is sensed by each CCD element, and the raw image thus formed from the array is a mosaic of red, green and blue pixels. CFA pattern using RGB and YMCG colour space for a 6x6 pixel block are displayed in Figures 1.2 and 1.3 respectively. The missing RGB values for each pixel need to be obtained through interpolation (demosaicing) as each sub-partition of pixels only provides information about a fixed number of green, red, and blue pixel values. By applying a kernel to the neighbouring pixels around a missing value, the interpolation is typically carried out. Most, each manufacturer uses a proprietary interpolation algorithm, i.e., kernels with different shapes, sizes and interpolation functions. This is then followed by the processing block, shown in the Figure 1.1, which involves a number of operations like colour compression and processing producing the final image.

1.2 Motivation

Digital forensics is an upcoming branch of computer science whose need increases with an increase of every counterfeit technology for digital images. This is coupled with an implementation of classifiers, a sub-topic of Machine Learning, which has untapped potential and is one of the most researched fields in Computer Science. In our work, we strive to improve the accuracy generated by the classifiers, which solidify the prediction of the source of the digital image in question. With a vast array of classifiers available to explore, we shall choose the ones that would cater to our requirements and try to optimize them. An improvement in feature extraction from images along with the evolution of more robust classifiers will leave ample space for developments of our work in future.

1.3 Contributions

The following are the contributions that have been made in this thesis:

- For a set of 156 images, the split, which determines the size of training and testing datasets, is optimized for every of the 8 classifiers.
- The accuracy, precision, recall and F1 Score generated by the 8 classifiers are

compared.

1.4 Thesis Organization

The thesis consists of five chapters following the present one:

Chapter 2: Literature Review

This chapter outlines the existing work on extraction of features necessary for Image Source Identification.

Chapter 3: Feature Extraction

This chapter discusses the various features important for distinguishing between two images of the same scenario, taken by two different camera models.

Chapter 4: Classifiers

This chapter discusses the eight different classifiers whose accuracies are compared.

Chapter 5: Experimental Results and Analysis

This chapter shows all the results of the performance measures of the aforementioned classifiers.

Chapter 6: Conclusion

This chapter presents analytical remarks to overall achievements.

Chapter 2

Literature Review

From the last decade, Digital Forensics has been a popular area of research work. Several researchers have worked on accurate and efficient extraction of features from images that help distinguish between them. This thesis is an extension of Blind Camera Source Identification [1], which proposes a number of features which the classifier uses for image source identification in a blind manner, along with providing experimental outcomes and credible accuracy in distinguishing images between the two different camera models with the help of the proposed features. However, with the advancements in the efficiency of the classifiers in use, the LibSVM package [2] that was used in [1] is rendered weak by today's standards.

There has been some earlier work on source identification based on camera characteristics such as noise levels, headers of images, pixel locations and format of the image [3]. However, there is a need for the presence of the original camera that had taken the image, which is not the case with our work.

Image Quality Metrics play a pivotal role in providing quantitative data on the quality of a rendered image [4]. Memon et al. [5] have also previously used IQM's. We use a subset of those IQM's for our studies in this work.

Chapter 3

Feature Extraction

A set of features that characterize the underlying interpolation algorithm implemented by a specific digital camera are determined in order to identify the source of the digital image. Multiple order of statistics of the digital image procured, are examined to apprehend the dissimilarities in the hidden colour characteristics for the varied camera models. We consider 22 characteristics as candidates that would help in differentiating the camera models:

- *RGB Pairs Correlation*: The fact that there could be a variation in the correlation between different colour bands, depending on the structure of the camera, is attempted to be measured through this feature. The three correlation pairs are RG, GB & BR.

- *Average Pixel Value*: According to the *gray world assumption*, the RGB bands of an image average to gray, considering that there is enough colour variation in the image. Thus, the mean of the 3 RGB channels constitute the 3 features.

- *Neighbour Distribution Center of Mass*: For each pixel value, its number of pixel neighbours is calculated separately for each band, where every pixel having a value difference of -1 or 1 from the pixel in consideration is defined as its neighbour. The camera pipeline sensitivity to different intensity levels is reflected in the obtained distribution. Since, a very similar but shifted distribution is observed for two same scenario images taken by two different camera models, to catch the shift as a feature, we calculate the center of mass of the neighbourhood plot.

- *RGB pairs energy ratio*: An integral part of the pipeline of a camera, white point correction is observed through RGB pairs energy ratio. The calculated features are:

$$E_1 = |R|^2/|G|^2, E_2 = |G|^2/|B|^2, E_3 = |B|^2/|R|^2$$

Apart from colour features, cameras also produce images of different 'qualities'. Visual differences such as sharpness, brightness, colour quality, etc. that can commonly be seen in images of different camera models motivate to apply a number of Image Quality Metrics (IQM) as features that help in distinguishing between camera models. Ten of the founded IQM's [5] are used in this work.

- *Mean Square Error*: Since the RGB colour space is not adequate to differentiate the intensity between two colours as is the colour difference estimation by the human eye, the image is defined in L*a*b* colour space. The Mean Square Error is the average of the square of the differences between each corresponding pixel of the two images of the same scenario, taken by the two camera models.

- *Mean Absolute Error*: It is a measure of the closeness of predictions or forecasts to the eventual outcomes.

- *Modified Infinity Norm*

- *Normalized Cross Correlation*: The normalization of the variation in the brightness of an image, which may arise because of lighting and exposure conditions, is realized through this feature.

Chapter 4

Classifiers

The identification of which of a set of categories, does a new observation belong to, based on the previously extracted features, is called classification. In Machine Learning, an instance of supervised learning is also called classification. The corresponding procedure for unsupervised learning is called clustering. Individual features help in the categorization of the final output. In our work, we have considered two-class classifiers, because of the usage of two camera models. The two camera models are identified by 0/1, which is the final prediction of the classifier in use.

4.1 Different Classifiers Used

We have made use of a total of 8 two-class classifiers, whose accuracies are compared. We strive to improve upon the efficiency of these classifiers, by optimizing their parameters.

4.1.1 Two-Class Averaged Perceptron

The Two-Class Averaged Perceptron module is implemented for the averaged perceptron algorithm, used in a machine learning model. A very simple and an early version of a neural network, the averaged perceptron method classifies the input into several possible outputs based on a linear function, and then combines with a set of weights that are derived from the feature vector.

4.1.2 Two-Class Bayes Point Machine

The Two-Class Bayes Point Machine module can be used in creating an untrained binary classification model. The Bayesian approach to linear classification is carried out through the Bayes Point Machine. One "average" classifier, the Bayes Point, is chosen through an approximation of the theoretically optimal Bayesian average of linear classifiers [6]. This implementation is an improvement on the original algorithm in the following ways:

- The expectation propagation message-passing algorithm is used.
- No parameter sweeping is required.
- Data does not need to be normalized.

4.1.3 Two-Class Boosted Decision Tree

This module can be used in creating a boosted decision trees algorithm based machine learning model. When configured properly, boosted decision trees are generally the easiest methods through which we can achieve top performance on a variety of problems. However, being a more memory-intensive learner, the current employment holds everything in memory, so the ease of handling larger datasets may not be as smooth as for a linear learner.

4.1.4 Two-Class Decision Forest

Decision forests are fast, supervised ensemble models, that can be used in the prediction of a two-valued target. They work by voting the most popular output class out of the earlier built multiple decision trees. Decision trees do not accept parameters, and support data with varied distributions.In each tree, for each class, a sequence of simple tests is run, traversing the levels until a leaf node is reached. The many advantages of decision tree are:

- Non-linear decision boundaries can be represented.
- High efficiency in memory usage and computation during training and prediction.

- Integrated feature selection and classification are performed.
- Resilient in the presence of noisy features.

4.1.5 Two-Class Decision Jungle

The Two-Class Decision Jungle module can be used in creating decision jungles, which is a supervised ensemble learning algorithm, based machine learning model [7]. They are an extension to the aforementioned Decision Trees. However, instead of the nodes of Decision Trees, Decision Jungles consist of Directed Acyclic Graphs (DAGs). The following are the advantages over decision trees:

- By merging the branches, a decision DAG has a lower memory footprint and hence, better performance.
- They do not accept parameters and can represent non-linear decision boundaries.
- Integrated feature selection and classification are performed.

4.1.6 Two-Class Logistic Regression

Logistic regression is a famous statistical implementation that is used in the prediction of the probability of an outcome. It creates a logistic regression model that predicts one of the two states of the target variable.

4.1.7 Two-Class Support Vector Machine

This classifier predicts between two possible outcomes dependent on categorical or continuous predictor variables. Support vector machines (SVMs) are supervised learning models that recognize patterns through analyzing data. They can be used for regression tasks and classification. The feature space that contains the training dataset is also called a hyperplane, and it may have a large number of dimensions.

4.1.8 Two-Class Locally Deep Support Vector Machine

The Two-Class Locally Deep Support Vector Machine is an extension of Support Vector Machines, which tries to optimize the efficient scaling of SVMs to larger

training datasets. In Two-Class Locally Deep Support Vector Machine, the kernel function that is used for mapping data points to feature space is specifically designed to reduce the time needed for training while maintaining most of the classification accuracy.

4.2 Performance Measures

The metrics by which we judge a binary classification model are: *Accuracy, Precision, Recall, F1 Score* and *AUC*. In addition, *true positive, false negatives, false positives, and true negatives*, as well as *ROC, Precision/Recall* and *Lift* curves also help us determine the performance of the model.

4.2.1 Accuracy

The proportion of correctly classified instances is called Accuracy. It generally is the first metric that we look at when evaluating a classifier. However, in case of an imbalanced test data (where the majority of instances belong to one class), or when we are mostly interested in the performance of either of the classes, accuracy does not serve a good parameter to depict the effectiveness of a classifier. For that reason, it is desirable to compute additional metrics that capture the more specific aspects of the evaluation.

4.2.2 Precision

Precision is the proportion of correctly classified positives: TP/(TP+FP)

4.2.3 Recall

Recall is the True Positive rate of the classifier: TP/(TP+FN)

4.2.4 F1 Score

F1 Score takes both Recall and Precision into consideration. It is the harmonic mean of the 2 metrics: 2*(Precision*Recall)/(Precision+Recall). The F1 Score is a good measure to summarize the evaluation in a single quantity.

4.2.5 True Positive

The positive instances correctly predicted by a classifier are called True Positives.

4.2.6 False Negative

The negative instances incorrectly predicted by a classifier are called False Negatives.

4.2.7 False Positives

The positive instances incorrectly predicted by a classifier are called False Positives.

4.2.8 True Negative

The negative instances correctly predicted by a classifier are called True Negatives.

4.2.9 ROC

The closer the Receiver Operating Characteristic (ROC) curve, which is a curve between the true positive rate and the false positive rate, is to the upper left corner, the better is the classifier's performance.

4.2.10 Precision/Recall

There is a visible trade-off between precision and recall. A classifier that predicts mostly positive instances, would have a high recall, but a rather low precision as many of the negative instances would be misclassified resulting in a large number of false positives.

Chapter 5

Experimental Results and Analysis

To check the validity of the aforementioned features in classifying images captured from a digital camera, a number of procedures are conducted. The two different camera models used to take images were, Nikon D5100 and Sony SLT-A58. These pictures were taken with maximum resolution, a size of 3696 X 2448 for Nikon, and a size of 3872 X 2576 for Sony, auto-focus, no flash, and the other settings set to default. The corresponding images depict the same scenario. This is necessary for comparing corresponding pixels of the images produced by the two camera models. In order to not disturb the interpolation algorithm used by the CFA of the camera model, we did not compress the pictures to the same size, rather cropping the upper-left of the larger image to the size of the smaller one.

Figure 5.1: The image on the left was captured using Nikon D5100, and the right using Sony SLT-A58.

An image dataset was made by taking 78 images with each of the cameras from throughout the institute campus of NIT Rourkela, as shown in Figure 5.1. As the resolution, and subsequent size, of the right image is more than the left, the upper left section of the larger image is cropped to the smaller image's size. After the collection of the dataset, the aforementioned features are calculated for every image. The extracted feature vectors are then stored in a comma separated value (.csv) file, as shown in Figure 5.3. 8 different classifier models were created in Microsoft Azure. Each of the classifiers was used in order to see the effectiveness of the features, by inputting the .csv file for every created model. The resultant accuracies of the 8 classifiers were analyzed and compared.

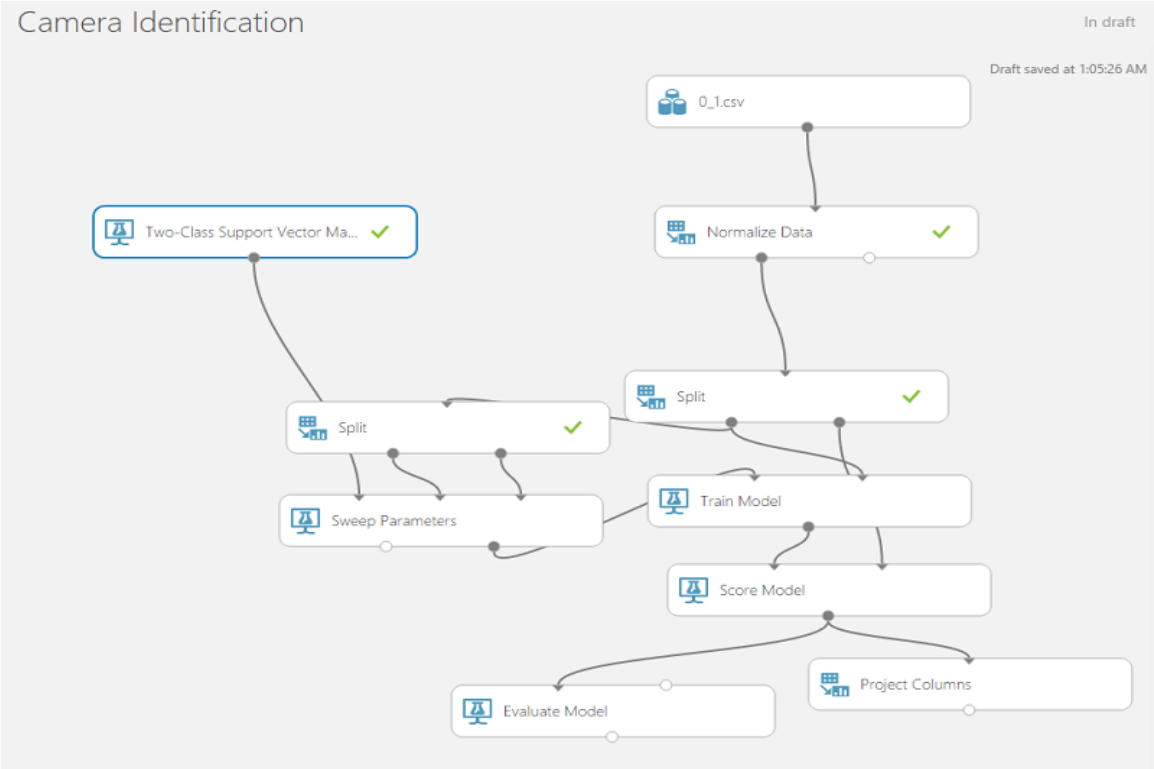

Figure 5.2: SVM classifier model generated in Microsoft Azure

The non-IQM features are exclusive to each camera model (12 features). The IQM features, which are a pixel-wise comparison of the two images, are common to both

34	73.30946	71.55526	56.51356	1.266161	0.976071	0.77089	2510	-519	-122587	0.952571	0.903901	0.85748	44.29121	20.35215	18.99587	9.090446	2.225278	2.016987	202.6601	193.9608	186.1789	0.761925	0
35	76.95835	76.80051	62.3425	1.231913	0.997949	0.810081	-690	-4340	-385771	0.954633	0.923659	0.860178	65.99611	29.18676	27.51838	17.06523	2.975274	2.819089	228.5049	220.4395	224.8243	0.332583	0
36	93.22959	99.77534	96.81271	1.030602	1.070211	1.038433	7	-53	-115755	0.921823	0.928799	0.771928	47.08392	18.14687	23.05815	10.35121	1.720346	2.241858	171.9571	167.7835	166.098	0.707174	0
37	110.7731	115.2248	110.1941	1.045653	1.040188	0.994773	-91008	-165743	-419077	0.966108	0.955743	0.908909	64.81963	21.90807	31.9125	14.82142	2.076249	3.274489	188.5633	183.7071	182.4212	0.659479	0
38	97.73697	115.1378	113.509	1.01435	1.178038	1.161372	1631	-50	-235021	0.953363	0.915422	0.809492	89.66816	21.05476	59.20583	23.0054	2.72186	6.058274	185.6585	187.5184	205.4608	0.682527	0
39	82.96382	96.98238	86.30582	1.123706	1.168972	1.040283	1026	-32	-364507	0.963671	0.909424	0.817191	81.66158	20.98541	47.16064	17.92279	2.784996	4.79218	191.0204	194.2978	204.3435	0.782921	0
40	124.1428	119.5811	111.4566	1.072893	0.963254	0.89781	-10275	-13752	-375556	0.91199	0.944224	0.763341	88.77607	34.83734	39.89024	22.68471	3.521382	4.499228	201.8145	198.625	213.404	0.700275	0
41	127.3443	122.8102	113.986	1.077415	0.964395	0.895101	-23258	-27789	-632649	0.914335	0.944998	0.769773	120.5372	32.86329	41.8802	26.91232	3.361036	4.727951	202.7079	199.2982	216.4624	0.705398	0
42	97.95241	105.8752	114.3894	0.925568	1.080884	1.167806	-1848	-1120	-1124	0.9208	0.957107	0.807173	53.6048	15.49206	34.18026	10.55524	2.26848	3.48373	145.1279	145.6413	172.7345	0.755997	0
43	118.33	112.8471	98.92387	1.140747	0.953664	0.836	-795	-63415	-374051	0.946811	0.958682	0.869293	64.85108	11.45674	29.06269	15.33773	1.311837	2.87662	204.3493	198.1491	203.0388	0.663031	0
44	123.0591	120.1988	116.9135	1.028101	0.976757	0.95006	-444938	-416213	-126894	0.989765	0.989089	0.967374	43.28011	4.867158	12.54684	8.45271	0.720209	1.365819	145.3051	137.9963	133.6009	0.851938	0
45	118.9798	115.2449	105.7905	1.089369	0.968609	0.889147	-70	-93	-62610	0.938843	0.959971	0.849531	51.20446	11.66816	23.22418	10.74028	1.389658	2.251441	185.6961	175.1054	177.9265	0.691199	0
46	82.2954	83.07447	69.759	1.190878	1.009467	0.847666	-2482	-26429	-231596	0.931235	0.911247	0.861625	67.60898	25.75347	34.16503	14.90244	2.68578	3.570413	206.0347	196.9432	199.6471	0.466322	0
47	103.2227	115.6267	119.0634	0.971135	1.120168	1.153462	-73	-48	-810351	0.963198	0.971685	0.903934	44.23203	8.057513	20.76915	9.711033	1.112075	2.189849	191.9065	189.1254	197.875	0.800275	0
48	141.2453	139.1462	124.7294	1.115584	0.985139	0.883069	-16026	-1044476	-1453349	0.955838	0.952605	0.880668	108.2426	18.81928	47.1084	25.31725	1.998572	4.982253	194.9408	190.0674	190.0805	0.748349	0
49	138.4411	136.8631	122.8123	1.114409	0.988602	0.887109	-4437	-957981	-1417430	0.956892	0.953794	0.882496	84.0236	18.55612	44.40612	18.09162	1.940183	4.682024	194.4461	190.2827	189.6842	0.75216	0
50	118.9799	121.8467	106.4173	1.14499	1.024095	0.894414	-3946	-5676	-190917	0.950823	0.929842	0.855893	50.40714	16.80682	36.14795	10.15023	1.879144	3.694644	201.4641	196.3791	201.6654	0.790103	0
51	90.71328	88.97067	70.65619	1.259206	0.98079	0.778896	-161963	-240150	-259828	0.915573	0.901417	0.883367	142.6239	49.66304	36.40387	31.6934	5.072788	3.865403	210.8166	202.1846	203.7586	0.507171	0
52	120.5825	119.2474	117.7363	1.012834	0.988928	0.976397	-637	-118	-159928	0.934686	0.955451	0.842558	69.42541	21.36245	39.99715	19.16279	2.174855	4.307782	179.5102	185.2533	199.7382	0.747888	0
53	112.2552	120.3451	113.7644	1.057845	1.072067	1.013445	-3302	-90	-155313	0.905208	0.891679	0.738088	98.1578	19.88028	51.91086	26.83184	2.324802	5.43097	193.4404	195.8337	214.0919	0.550622	0
54	108.6955	114.7491	103.9622	1.103757	1.055693	0.956454	-6746	-4030	-65490	0.893606	0.902544	0.720632	87.38386	24.58863	46.7033	19.78983	2.848109	5.121434	184.3423	173.2672	204.1156	0.648599	0
55	89.64268	92.94242	84.35374	1.101817	1.03681	0.941	-25871	-24486	-2075	0.994276	0.981628	0.988844	83.08275	8.313521	18.02133	16.31294	1.401285	2.028583	186.1238	179.989	181.1769	0.777578	0
56	107.1161	109.5075	102.6652	1.066646	1.022325	0.958448	121	-289	-195409	0.977354	0.953038	0.915234	60.28302	29.7018	43.77794	15.11967	3.261823	4.997254	195.6029	192.654	211.3382	0.732879	0
57	113.9719	116.9407	108.7533	1.075284	1.026049	0.954212	-370	-445	-192950	0.974211	0.967857	0.917448	54.19454	20.1387	42.64713	13.13593	2.473778	4.294793	184.3162	181.5143	195.3415	0.729374	0
58	96.23069	101.4704	65.24956	1.555113	1.05445	0.678054	-1381	-786	-850320	0.888204	0.771911	0.764477	62.657	33.26509	44.37095	15.62535	3.490797	5.442201	201.7406	200.694	232.7974	0.416923	0
59	113.5591	121.3223	94.29191	1.286667	1.068363	0.830334	-7328	-2396	-176367	0.972657	0.911845	0.895024	122.8338	20.82558	69.61793	25.17322	2.547663	7.779226	190.3207	190.9449	227.9024	0.833019	0
60	121.3367	119.5821	118.7552	1.006963	0.985539	0.978725	-224	-96	40189	0.962182	0.979256	0.90701	53.23628	11.3274	22.51589	10.09123	1.29545	2.284119	181.1761	173.7741	177.7516	0.873819	0
61	111.3963	108.5333	109.0544	0.995222	0.974299	0.978977	334	-49	-25434	0.87928	0.944559	0.738633	75.59026	17.55696	29.73927	16.69445	2.000504	3.023453	149.4003	146.2365	141.3427	0.649558	0
62	106.8331	88.07753	74.8297	1.17704	0.82444	0.700435	-4274	-2740	-6558	0.949541	0.952516	0.868555	59.28323	11.7512	29.93847	15.07743	1.308398	2.951721	206.3983	205.1373	207.3717	0.515212	0
63	77.82006	81.66628	82.15086	0.994101	1.049425	1.055651	10467	7981	-250383	0.962202	0.978637	0.906741	55.27823	18.53952	26.82986	17.86957	1.925204	2.744364	219.8868	215.8419	220.4579	0.445924	0
64	100.7845	111.3222	101.7735	1.093822	1.104556	1.009813	-72142	-14647	-1364661	0.962087	0.932762	0.88545	68.84292	17.37441	28.94665	13.40944	1.916318	3.171156	193.5711	190.1303	208.9171	0.751363	0
65	120.2371	141.585	137.1371	1.032434	1.177548	1.140556	-281772	-6770	-2764535	0.910321	0.923696	0.791057	202.3939	11.06174	62.28088	45.28192	1.240875	7.177883	198.172	208.0743	238.2958	0.711411	0
66	135.8237	137.9162	137.5297	1.002811	1.015406	1.01256	-3951	-88369	-596602	0.958355	0.980571	0.899517	80.41683	5.211006	5.670965	24.71059	0.631787	0.60263	173.4028	174.9596	180.8342	0.723197	0
67	86.07019	83.49095	80.20206	1.041007	0.970033	0.931822	3138	62	-28114	0.936759	0.942395	0.788998	56.42415	15.29459	26.59113	13.44045	1.855905	2.761621	196.6654	181.0805	186.4011	0.689225	0
68	100.014	104.1647	80.80501	1.289087	1.041501	0.807937	-214850	-252333	-429786	0.977537	0.885673	0.893993	143.8481	15.89575	46.65944	34.82942	1.942228	5.134835	217.7512	217.232	222.5392	0.503706	0
69	109.8139	117.5511	120.1752	0.978164	1.070457	1.094353	-128	-19	-62743	0.880504	0.907024	0.685922	45.00059	14.00128	28.63024	12.62558	1.883978	3.233962	143.1381	142.6409	153.8219	0.647267	0
70	95.1947	102.4208	105.9821	0.966397	1.075908	1.113319	116401	2843	-151205	0.93153	0.933889	0.804642	42.53933	17.00687	30.8327	12.36935	1.996556	3.224564	175.5748	176.6242	189.1393	0.719699	0
71	98.72243	89.83978	78.1612	1.149417	0.910024	0.791727	-1620	-10828	-337178	0.933484	0.946355	0.867354	72.6458	25.60781	40.61948	17.19982	2.675336	4.297703	210.7553	217.9163	229.8472	0.417195	0
72	109.8417	106.1041	100.0378	1.06064	0.965972	0.910745	-416	-4089	-112361	0.974914	0.977061	0.93484	61.44651	19.39296	38.89952	14.79085	2.249395	4.040626	193.3219	195.0433	202.7333	0.623324	0
73	115.5918	108.1424	104.1475	1.038358	0.935554	0.900994	1549	-23	543	0.923056	0.936503	0.754982	65.13944	28.42871	83.02014	18.24465	2.957731	9.427625	184.8991	177.7288	191.1752	0.4959	0
74	77.77391	75.05554	64.79665	1.158324	0.965048	0.833141	56	1	-135897	0.964717	0.919897	0.869673	45.15929	14.6751	20.21767	9.406474	1.68213	2.096577	226.9984	222.1818	224.6969	0.497196	0
75	99.99434	96.968	85.36902	1.135869	0.969735	0.853739	-60	-36	-116115	0.938488	0.902681	0.809386	32.23344	6.726133	11.19738	6.14984	0.8433	1.207796	191.5192	179.904	182.1066	0.612718	0
76	115.5679	113.4464	108.7435	1.043248	0.981643	0.940949	-8	3	-12480	0.868331	0.94353	0.716689	43.03686	12.39928	18.97473	9.656208	1.282303	1.906296	159.9011	141.2075	137.0421	0.725865	0
77	115.8239	120.8445	105.6411	1.143916	1.043347	0.912084	-151	-144	-320232	0.953904	0.938661	0.891072	80.64074	11.84573	36.60631	18.67088	1.340504	3.810391	177.1017	176.4261	193.5719	0.695872	0
78	125.1701	123.9092	115.3522	1.074181	0.989927	0.921564	-632	-6	-305060	0.823346	0.917908	0.684348	46.59729	17.4509	33.0362	11.7221	1.923693	3.17774	173.0617	165.8868	184.3109	0.622502	0
79	157.8537	157.7243	156.2658	1.009333	0.99918	0.989941	-139	-72	70	0.927742	0.966331	0.825969	23.79923	8.169595	19.55766	5.142733	1.029913	1.855114	116.4861	106.3415	110.4922	0.78205	1
80	164.8714	162.4094	154.9452	1.048173	0.985067	0.939794	-16	-1	251	0.91918	0.942308	0.765436	29.06836	17.16846	39.82456	7.025479	2.14629	4.049771	141.0012	136.3243	144.3709	0.670645	1
81	153.8867	151.639	136.4284	1.111492	0.985394	0.886551	-3678	-512	-1067	0.984009	0.963716	0.938167	53.87099	12.99549	49.74746	11.84427	1.533407	5.021345	195.4346	188.6119	218.748	0.756497	1
82	159.7131	153.4028	146.9722	1.043754	0.96049	0.920227	-69	-6	7301	0.957545	0.981401	0.900303	16.1615	6.83294	10.78248	2.594226	0.924915	1.280666	153.0768	154.3627	168.0486	0.911018	1
83	145.3516	143.7696	136.9149	1.050065	0.989116	0.941957	144	3	-980	0.976955	0.980953	0.940139	42.30226	8.83286	21.54431	8.700399	1.208864	2.174262	178.6904	179.5515	189.5637	0.767979	1
84	151.273	145.9607	140.2197	1.040943	0.964883	0.926931	-1430	-736	1636	0.939694	0.976424	0.905948	23.71718	8.580568	19.90597	3.887576	0.899751	1.920339	162.6197	156.2496	172.0511	0.840341	1
85	183.3374	162.3595	151.6463	1.070646	0.885577	0.827143	-850	-192	-490	0.776841	0.936286	0.538322	32.65119	15.41776	25.08658	7.116697	1.436812	2.335964	141.2455	127.9604	130.3337	0.563309	1
86	172.7326	164.0804	165.5305	0.99124	0.94991	0.958305	-176	-23	-9	0.969172	0.984811	0.932216	22.68257	16.67069	26.63647	3.661501	1.506709	2.957746	166.826	169.5078	174.1238	0.723081	1
87	164.5057	156.6154	155.0935	1.009812	0.952036	0.942785	-79	6	736	0.959295	0.985435	0.915234	38.19611	15.26769	17.97256	7.338603	1.359918	1.719711	124.8893	117.9179	119.0257	0.84155	1
88	147.9584	139.2959	137.7981	1.01087	0.941453	0.93133	82	714	-5219	0.935888	0.985648	0.887687	41.69977	10.21953	22.21137	7.826588	1.033081	2.150615	166.7496	156.9596	164.0278	0.770392	1
89	142.7805	133.8543	133.3709	1.003625	0.937484	0.934098	-341638	-126887	-787	0.986053	0.997609	0.981983	149.3467	3.774134	5.927311	20.95191	0.548475	0.877588	165.7529	173.9743	167.7618	0.887893	1
90	169.012	143.9965	120.6153	1.193849	0.85199	0.71365	-399	-67	255	0.868227	0.949096	0.707641	41.02449	42.44424	32.22094	7.033456	4.557939	3.524616	140.8897	139.9886	145.6998	0.724123	1
91	156.4881	149.0092	138.2838	1.077561	0.952208	0.88367	-676	-118	-1101	0.936602	0.961991	0.841913	46.83779	17.94458	30.59112	10.18572	2.279587	3.188018	159.5731	155.3975	162.3109	0.68007	1
92	168.9955	140.3753	117.5182	1.194498	0.830646	0.695393	-447	-31	449	0.867809	0.948074	0.702855	36.20482	24.71582	17.57588	5.644472	3.117847	2.22032	146.3828	147.3807	149.2467	0.797565	1
93	139.4009	101.7038	94.23849	1.079217	0.729578	0.676025	-132	60	102	0.943373	0.983456	0.89192	67.69679	4.208616	7.683014	15.22689	0.486381	0.995601	115.701	117.2377	109.2243	0.838872	1
94	152.889	136.5999	120.4214	1.134349	0.893458	0.787639	-8571	-27240	-47713	0.945951	0.975718	0.898542	56.54995	14.73114	28.8852	11.72278	1.612604	2.929992	188.9767	186.556	188.904	0.723806	1
95	120.2591	105.7164	86.8456	1.217291	0.879072	0.722154	-1310	-1380	-55436	0.942512	0.928792	0.82835	81.01866	18.95361	35.72891	19.95499	2.013007	4.008411	198.4236	196.0707	215.4865	0.359148	1
96	162.5937	145.6779	138.4146	1.052475	0.895963	0.851291	-63	-121	-2920	0.967609	0.989346	0.939279	43.14861	18.18233	29.07939	9.127322	1.982928	3.242271	175.9534	176.8762	185.4612	0.90127	1
97	105.1278	97.01814	79.26511	1.22397	0.922859	0.753988	-153165	-222054	-413272	0.972834	0.949559	0.932632	59.39276	9.830636	19.76134	10.82197	1.323247	2.129648	185.8321	171.3533	170.5106	0.789225	1
98	125.3	121.294	84.56186	1.434382	0.968028	0.674875	-15884	-10258	-395251	0.973377	0.908277	0.886124	91.77575	24.84902	57.95736	23.24433	2.697409	6.25378	216.8264	215.6965	207.3186	0.718879	1
99	143.4158	133.5808	115.2364	1.159189	0.931423	0.803512	-1454	-215	-79961	0.976469	0.955758	0.933324	42.17614	12.6816	13.66991	9.715971	1.449901	1.553301	197.5551	194.645	192.5392	0.713995	1
100	144.8635	133.2417	122.6808	1.086085	0.919774	0.846872	-1032893	-1037722	-612658	0.978298	0.988211	0.948892	75.08711	5.383423	4.819819	12.01018	0.716765	0.591407	147.4416	143.1181	141.1021	0.87009	1
101	115.1685	113.4551	85.41548	1.328274	0.985123	0.741657	-375095	-437865	-436517	0.972682	0.908406	0.901243	99.57512	30.2347	25.87758	21.55987	3.238759	2.631454	202.1295	200.5829	210.2962	0.490305	1
102	102.3351	98.19377	85.82795	1.144077	0.959532	0.838695	1814	54290	-106825	0.987733	0.955455	0.9383	95.96352	7.970178	20.17881	22.52381	1.334651	2.141432	205.5944	196.7202	194.6573	0.421286	1
103	89.33442	89.73371	82.61033	1.086229	1.00447	0.924731	-4619	-37710	-350002	0.984406	0.963483	0.950749	42.1295	3.778281	9.869521	6.486874	0.868231	1.323591	161.6642	156.0102	156.6985	0.934553	1
104	132.3692	120.9338	112.8429	1.0717	0.913609	0.852486	-634400	-628468	-658913	0.945949	0.95181	0.878739	48.77628	15.58333	25.71588	9.453607	1.753096	2.868993	201.1058	194.9261	192.636	0.540128	1
105	102.5806	95.88271	69.50485	1.379511	0.934706	0.677563	-2178	1223	-111166	0.950123	0.923773	0.875273	89.68793	25.64045	42.89714	21.82403	2.827378	4.225554	194.5282	183.0625	186.4036	0.492694	1
106	107.8013	99.02795	86.16952	1.149223	0.918615	0.799336	-111400	-57871	-53879	0.963537	0.909743	0.867767	41.65473	22.02876	48.34148	8.045236	2.424941	5.124898	210.3072	209.4489	204.5429	0.516566	1
107	102.282	97.18251	86.48371	1.123709	0.950143	0.845542	-571520	-653299	-752301	0.98686	0.962683	0.950274	60.38692	7.303655	19.62571	12.15072	1.065151	2.089006	197.0176	187.0278	190.7659	0.715058	1
108	126.2241	118.0245	96.00258	1.229389	0.935039	0.760573	-45268	-15928	-72961	0.969104	0.937565	0.93237	52.72139	12.69308	20.01457	10.45183	1.525231	2.105606	196.6221	182.201	180.991	0.692743	1
109	106.4183	105.7829	90.23961	1.172245	0.994029	0.847971	-186307	-179542	-273742	0.975994	0.944283	0.946727	51.66883	18.83329	22.88192	9.955676	2.21312	2.458947	200.7933	194.4105	193.5016	0.695132	1
110	111.8499	101.5841	74.33663	1.366541	0.908218	0.664611	-1421	748	9607	0.962518	0.921744	0.861452	57.93534	15.79453	24.53742	12.84836	1.79473	2.517232	198.0437	188.4093	179.9914	0.715773	1
111	144.7107	121.963	109.9183	1.109578	0.842805	0.759573	-189552	-278604	-297681	0.946545	0.97982	0.893645	63.36933	7.906183	10.69765	13.15626	0.851158	1.061411	170.0139	167.9959	168.6985	0.741221	1
112	107.0907	100.7866	75.32086	1.338097	0.941134	0.703337	-36695	-18292	-59317	0.975863	0.924336	0.905715	44.29121	20.35215	18.99587	9.090446	2.225278	2.016987	202.6601	193.9608	186.1789	0.761925	1
113	117.1092	107.8939	79.09815	1.364051	0.92131	0.675422	-50858	-108468	-303251	0.967124	0.917409	0.899365	65.99611	29.18676	27.51838	17.06523	2.975274	2.819089	228.5049	220.4395	224.8243	0.332583	1
114	122.6339	116.9095	106.5879	1.096836	0.953322	0.869156	-14232	1512	-42672	0.931668	0.916321	0.78815	47.08392	18.14687	23.05815	10.35121	1.720346	2.241858	171.9571	167.7835	166.098	0.707174	1
115	144.7487	133.5553	121.7209	1.097225	0.92267	0.840912	-332849	-618274	-1207436	0.955829	0.951188	0.918941	64.81963	21.90807	31.9125	14.82142	2.076249	3.274489	188.5633	183.7071	182.4212	0.659479	1
116	101.5579	115.1096	106.8327	1.077475	1.133438	1.051939	27713	7006	-74909	0.964496	0.928108	0.844157	89.66816	21.05476	59.20583	23.0054	2.72186	6.058274	185.6585	187.5184	205.4608	0.682527	1
117	92.50875	105.9169	89.63701	1.181621	1.14494	0.968957	26847	5456	-23836	0.968633	0.924431	0.850414	81.66158	20.98541	47.16064	17.92279	2.784996	4.79218	191.0204	194.2978	204.3435	0.782921	1
118	126.9668	119.5104	106.6764	1.120308	0.941273	0.840191	-1756	191	-2107	0.919703	0.939483	0.770589	88.77607	34.83734	39.89024	22.68471	3.521382	4.499228	201.8145	198.625	213.404	0.700275	1
119	122.2465	114.9915	102.9744	1.1167	0.940652	0.84235	-1154	363	2765	0.919293	0.939048	0.76598	120.5372	32.86329	41.8802	26.91232	3.361036	4.727951	202.7079	199.2982	216.4624	0.705398	1
120	113.8244	120.6385	125.8053	0.95893	1.059865	1.105258	-2106	2033	566	0.92162	0.95904	0.8121	53.6048	15.49206	34.18026	10.55524	2.26848	3.48373	145.1279	145.6413	172.7345	0.755997	1
121	141.6377	126.4479	107.8936	1.171969	0.892756	0.761757	-37943	-287904	-582417	0.959605	0.962351	0.890916	64.85108	11.45674	29.06269	15.33773	1.311837	2.87662	204.3493	198.1491	203.0388	0.663031	1
122	140.8563	132.4483	126.8278	1.044316	0.940308	0.900405	-741942	-763656	-478849	0.990993	0.992322	0.978703	43.28011	4.867158	12.54684	8.45271	0.720209	1.365819	145.3051	137.9963	133.6009	0.851938	1
123	156.2278	141.0871	121.1575	1.164494	0.903086	0.775518	-5254	-12477	-183455	0.94633	0.953927	0.856431	51.20446	11.66816	23.22418	10.74028	1.389658	2.251441	185.6961	175.1054	177.9265	0.691199	1

Figure 5.3: Features extracted in a .csv file

the cameras (10 features). The .csv file thus made, consists of 156 rows (78 for each camera model) and 22 columns. A 23rd column is added which is binary in nature

(0/1) with 0 denoting Nikon and 1 denoting Sony. The split module, which is variable in the model, splits the input dataset into Training dataset and Testing dataset.

5.1 Results

5.1.1 Two-Class Averaged Perceptron

For a training dataset of 144 images and a test dataset of 12 images (a split of 0.92), the accuracy is 91.7%.

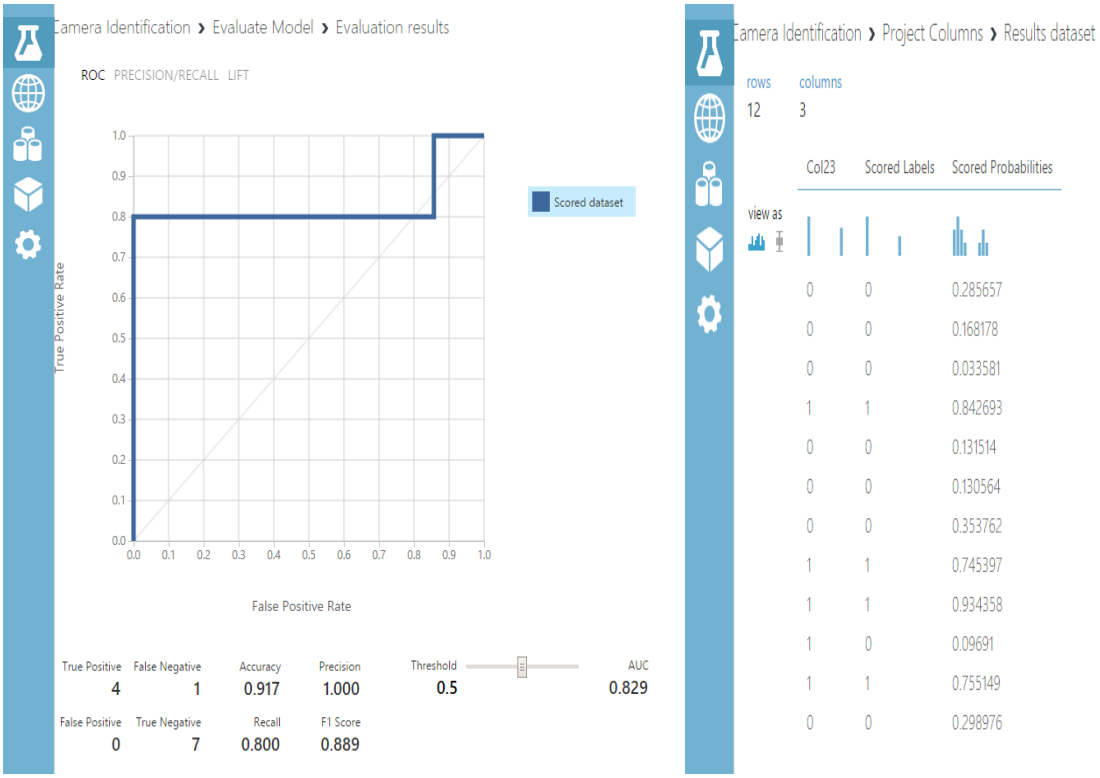

Figure 5.4: Evaluation and Score Results of Two-Class Averaged Perceptron

5.1.2 Two-Class Bayes Point Machine

For a training dataset of 140 images and a test dataset of 16 images (a split of 0.81), the accuracy is 90.0%.

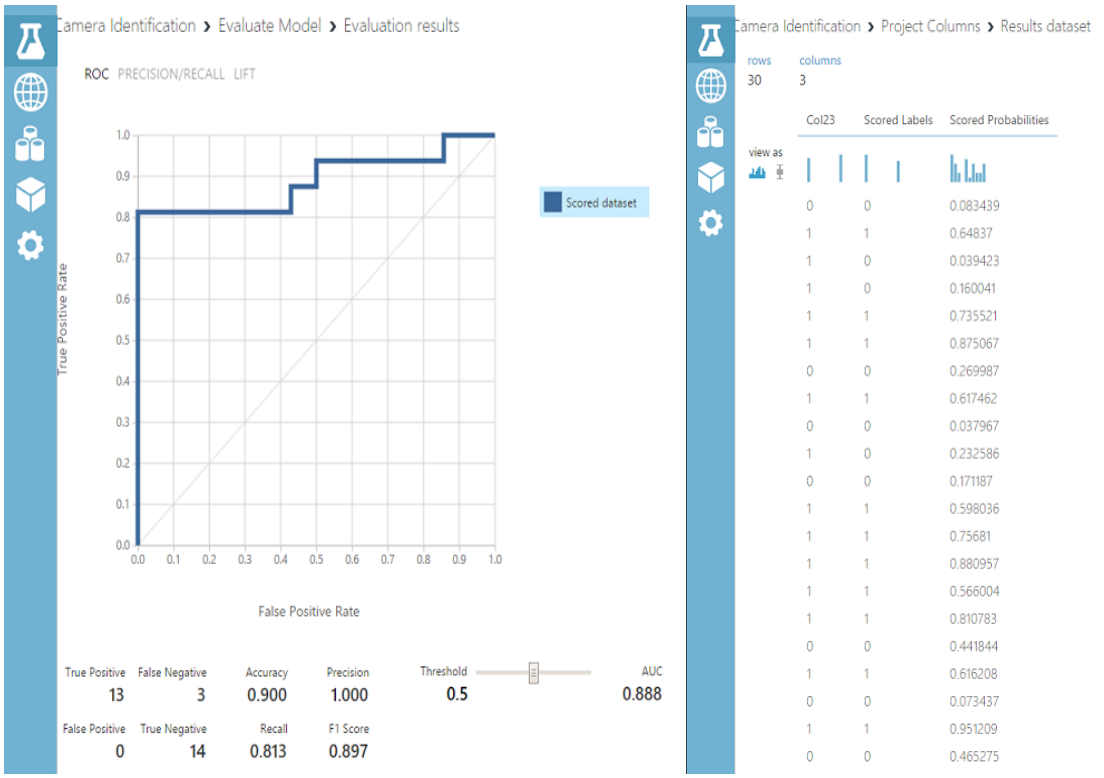

Figure 5.5: Evaluation and Score Results of Two-Class Bayes Point Machine

5.1.3 Two-Class Boosted Decision Tree

For a training dataset of 134 images and a test dataset of 22 images (a split of 0.86), the accuracy is 95.5%.

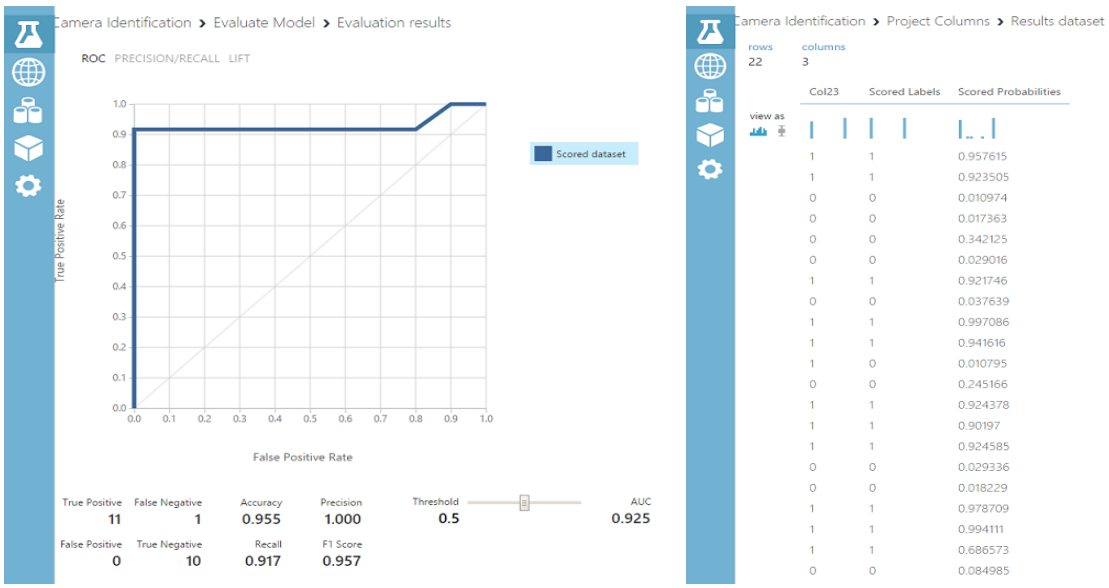

Figure 5.6: Evaluation and Score Results of Two-Class Boosted Decision Tree

5.1.4 Two-Class Decision Forest

For a training dataset of 142 images and a test dataset of 14 images (a split of 0.91), the accuracy is 92.9%.

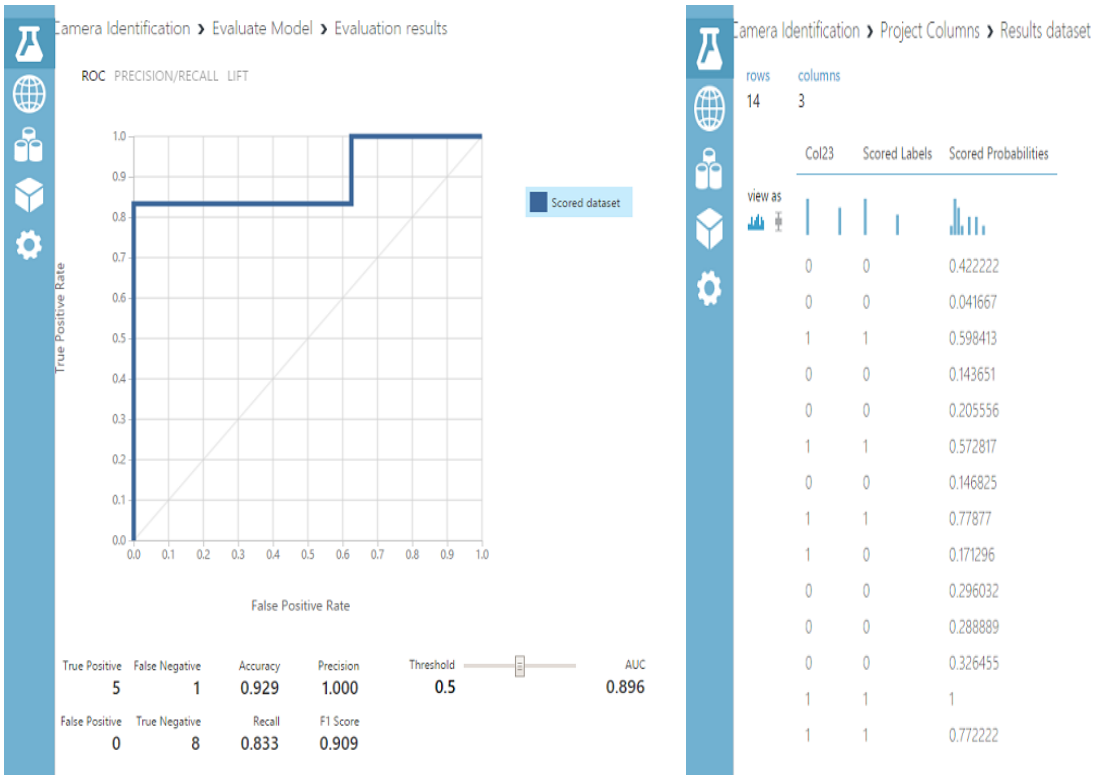

Figure 5.7: Evaluation and Score Results of Two-Class Decision Forest

5.1.5 Two-Class Decision Jungle

For a training dataset of 128 images and a test dataset of 28 images (a split of 0.82), the accuracy is 89.3%.

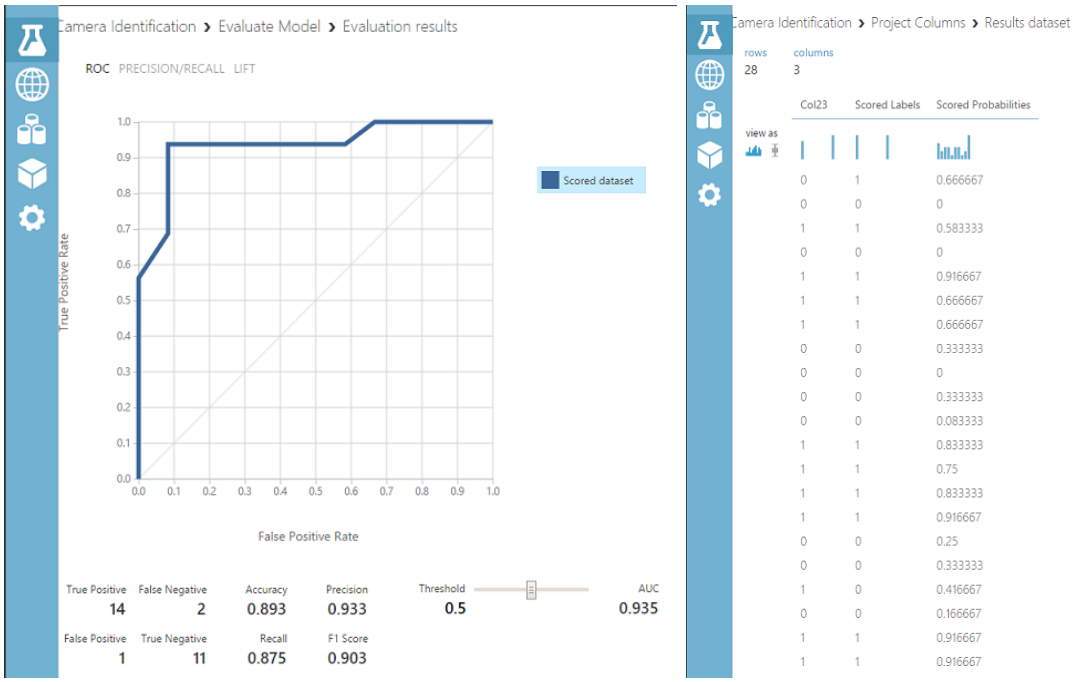

Figure 5.8: Evaluation and Score Results of Two-Class Decision Jungle

5.1.6 Two-Class Logistic Regression

For a training dataset of 137 images and a test dataset of 19 images (a split of 0.88), the accuracy is 89.5%.

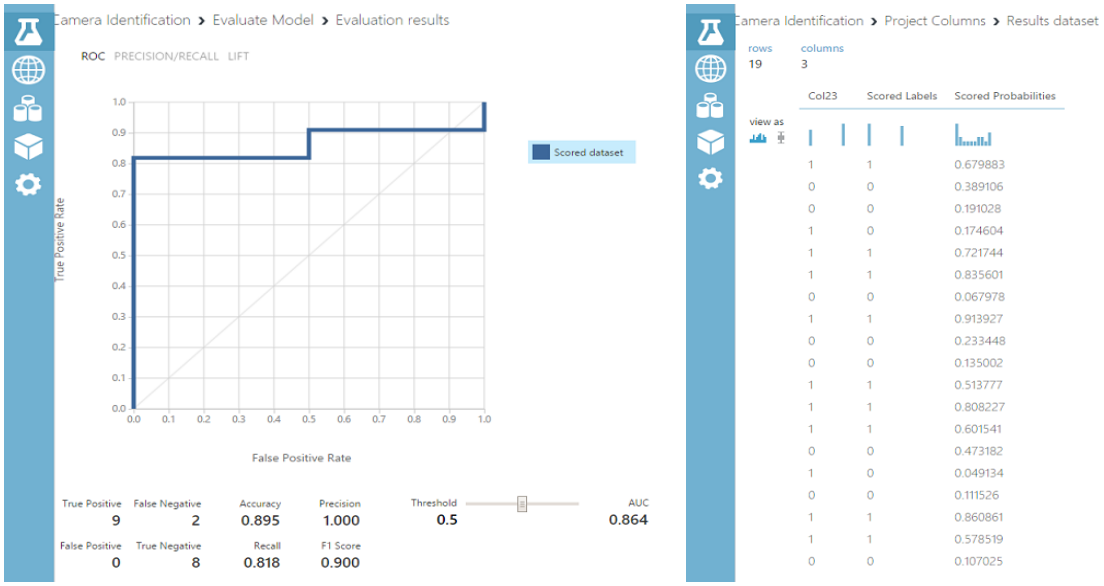

Figure 5.9: Evaluation and Score Results of Two-Class Logistic Regression

5.1.7 Two-Class Support Vector Machines

For a training dataset of 144 images and a test dataset of 12 images (a split of 0.92), the accuracy is 91.7%.

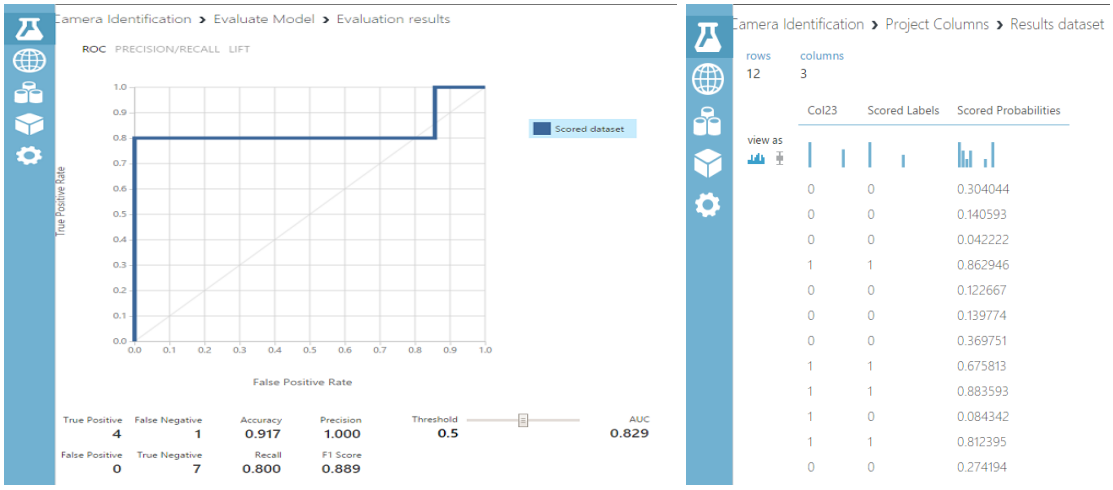

Figure 5.10: Evaluation and Score Results of Two-Class Support Vector Machines

5.1.8 Two-Class Locally-Deep Support Vector Machines

For a training dataset of 142 images and a test dataset of 14 images (a split of 0.91), the accuracy is 85.7%.

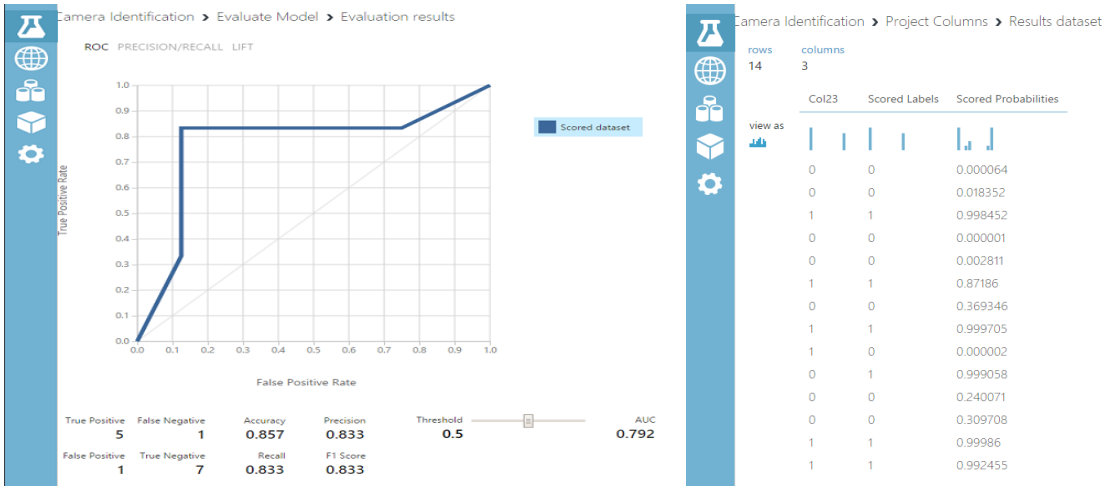

Figure 5.11: Evaluation and Score Results of Two-Class Locally-Deep Support Vector Machines

Chapter 6

Conclusion and Future Work

We have considered the image characteristics that are important for the image source identification of a digital image. We then optimize the various classifiers in use, to the dataset provided by us. Finally, we compare the efficiency of these classifiers. With improvements in image feature extraction, the boundary of which characteristics are important becomes more distinct, thus, discarding the redundant ones. However, there is ample room for improvement in both extraction as well as classification. Using more than the extracted 22 features can further increase the accuracy of the classifiers:

- Wavelet Domain Statistics
- Pratt Measure
- f Divergences
- HVS Modified Spectral Distortion

The dataset of 156 images is not large enough to satiate the training and testing needs of the classifiers. Thus, as we increase the number of images that are used for training and testing, we will also increase the accuracy of the classifiers.

Bibliography

[1] Mehdi Kharrazi, Husrev T Sencar, and Nasir Memon. Blind source camera identification. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 1, pages 709–712. IEEE, 2004.

[2] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[3] Zeno J Geradts, Jurrien Bijhold, Martijn Kieft, Kenji Kurosawa, Kenro Kuroki, and Naoki Saitoh. Methods for identification of images acquired with digital cameras. In *Enabling Technologies for Law Enforcement*, pages 505–512. International Society for Optics and Photonics, 2001.

[4] Khalid Sayood et al. Statistical evaluation of image quality measures. *Journal of electronic imaging*, 11(2):206–223, 2002.

[5] Ismail Avcibas, Nasir Memon, and Bülent Sankur. Steganalysis using image quality metrics. *Image Processing, IEEE Transactions on*, 12(2):221–229, 2003.

[6] Ralf Herbrich, Thore Graepel, and Colin Campbell. Bayes point machines. *The Journal of Machine Learning Research*, 1:245–279, 2001.

[7] Jamie Shotton, Toby Sharp, Pushmeet Kohli, Sebastian Nowozin, John Winn, and Antonio Criminisi. Decision jungles: Compact and rich models for classification. In *Advances in Neural Information Processing Systems*, pages 234–242, 2013.