

# Camera Model Identification

Kinjal N. Patel

G. H. Patel College of Engineering & Technology, Anand, Gujarat, India

## I. INTRODUCTION

Everyone is interested in proving that a particular picture is taken by his/her camera, in order to claim the property. Moreover, In general a picture was taken by particular camera, but it is very important element for decisions in court and It cannot rely on Meta data (EXIF tags) which can be easily manipulated. It is noted that in 2015 more than 1.8 billion images published on the internet each day<sup>[12]</sup>, and this trend is going more and more each day. Source Identification majorly relying on the Photo Response Non-Uniformity (PRNU) pattern, it is stable in time, it is originated by the inerasible imperfections occuring during the sensor manufacturing process. Since each picture is taken by particular camera has traces of PRNU Pattern, It can be credible and plausible identification, Image Falsification detection and also improving recognition algorithms.

According to the survey PRNU approach is not more convenient because a large number of images taken by that camera is necessary and also it is impossible without cooperation of the camera owner. Additionally, PRNU based techniques are greatly time-consuming and it can't be easily applied to a large dataset of pictures. The output image is obtained by applying a several number of sophisticated algorithms; each one is characterized by parameters. For example, Demosaicing and JPEG compression, in this quantization matrix can be defined by the user.

Reviewed that Kharrazi et al.[2] in 2004 considered the use of generic features ( Average Pixel Value, RGB Pairs Correlation, Image Quality Matrix,etc)for camera model identification. Actually it is the first paper to present an approach that did not focus on a specific camera artifacts.

There are mainly 3 approaches of camera model identification: Image Metadata based , watermark based and feature based. Image metadata based approach relies on Image source related information such as camera model, brand , date and time. However image metadata is easy to be manipulated. The Watermark based approach that has watermark carries source related information. It is inserted during the creation of an image. This increases the production cost of the digital cameras. In recent era of research, important efforts have been devoted in the Feature based approach. In this approach, it extracts features on intrinsic hardware artifacts and software related fingerprints left during the image acquisition process.

Figure 1 gives us an overview of a common image processing pipeline in digital cameras. Each of the these stages differently implemented by manufacturer of different camera models. Previous researchers focuses on the certain stages of the pipeline such as lens defects, CFA (Color Filter Array), Demosaicing , Sharpening , White balance and gamma correction , etc. Some consider more than one stages or whole pipeline.

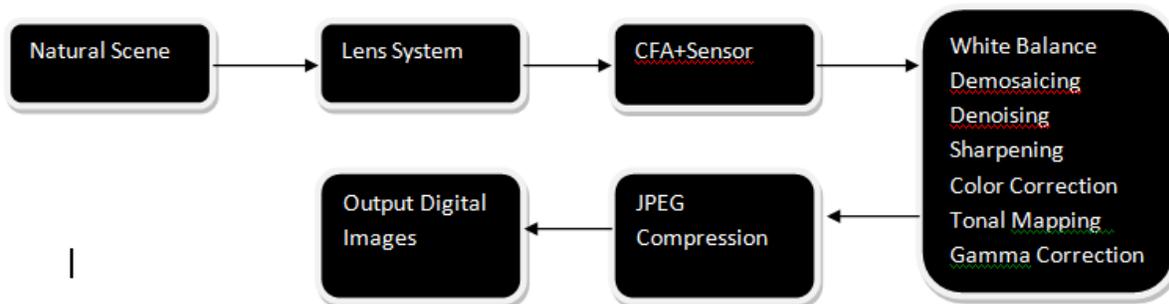


Figure 1. Image Processing Pipeline in Digital Camera

Figure 1. gives an overview of a common image processing pipeline in digital cameras. Each of these stages differently implemented by manufactured of different camera models. Previous researchers focus on certain stages in this pipeline such as lens defects, Color Filter Array(CFA),

Demosaicing, JPEG Compression, Denoising, Sharpening, white balance and gamma correction , etc. Some consider more than one stages or whole pipeline.

Pragmatic trial settings for camera demonstrate recognizable proof require in excess of one camera from each show with

the end goal to evacuate the vagueness of whether the highlights, on which the classifiers are manufactured, catch camera show attributes or individual camera qualities [11-13]. Inside each model, testing pictures ought not come from a similar individual cameras that are associated with preparing. In any case, the greater part of the past looks into just utilize one camera to speak to a camera display due to the constraints of camera sources

Binary similarity measures (BSM) calculated from three least significant bit-planes was used in [10] for camera model identification. Together with another two types of feature sets (HOWS and IQM).

In this paper, we propose to local binary patterns (LBP) as statistical features. Considering 8-neighbor graylevel difference for each image pixel around a circle, 59 local binary pattern are extracted, respectively, from spatial domain of red and green color channels, their prediction-error 2D arrays, and the 1st-level diagonal wavelet subband of each image. Logistic Regression model are built for classification of 18 camera models from ‘Dresden Image Database’. Compared with the results in literatures, the detection accuracy reported in this paper is much better.

The rest of the paper is structured as follows. In Section 2, the LBP features that we use, how to extract features. In Section3, experimental works are presented and some discussions are made. Conclusions are drawn in Section 4.

## II. PROPOSED METHOD

In this section, we first give a brief description of uniform local binary patterns proposed in [15]. Our proposed feature extraction framework will then be introduced.

### 2.1 Local Binary Patterns

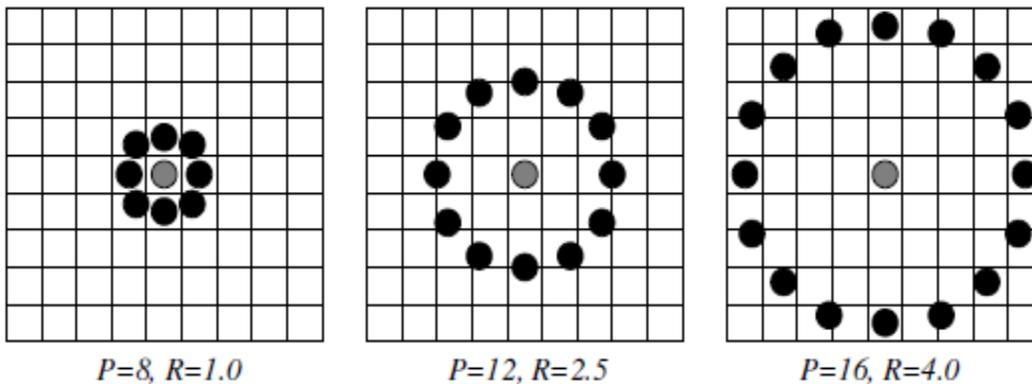


Figure 3. Circularly symmetric neighbor sets. Samples that do not exactly match the pixel grid are obtained via interpolation.



Figure 2. A color texture and its gray-scale version.

Texture analysis methods have been developed with gray-scale images, intuitively for good reasons. Humans can easily capture the textures on a surface, even with no color information. Figure.2 shows a photograph of tricolor pasta and its gray-scale version. The only thing that cannot be told, based on the gray-scale information, is the color of the pasta — the texture itself is the same. The human visual system is able to interpret practically achromatic scenes for example in low illumination levels. Color acts just as a cue for richer interpretations. Even when color information is distorted, for example due to color blindness, the visual system still works. Intuitively, this suggests that at least for our visual system, color and texture are separate phenomena. Nevertheless, the use of joint colortexture features has been a popular approach to color texture analysis.

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \dots(2.1)$$

where  $R$  is the radius of a circularly symmetric neighborhood used for local binary patterns calculation

$P$  is the number of samples around the circle.

In this paper, we set  $R = 1, P = 8$ .  $g_c$  and  $g_p$  represent gray levels of the center pixel and its neighbor pixels, respectively.

In practice, Eq. 2.1 means that the signs of the differences in a neighborhood are interpreted as a  $P$ -bit binary number, resulting in  $2^P$  distinct values for the LBP code. The local gray-scale distribution, i.e. texture, can thus be approximately described with a  $2^P$ -bin discrete distribution of LBP codes:

$$T = t(LBPP,R(x_c, y_c)) \dots(2.2)$$

Let us assume we are given an  $N \times M$  image sample  $(x_c \in \{0, \dots, N-1\}, y_c \in \{0, \dots, M-1\})$ .

In calculating the LBPP,R distribution (feature vector) for this image, the central part is only considered because a sufficiently large neighborhood cannot be used on the borders. The LBP code is calculated for each pixel in the cropped portion of the image, and the distribution of the codes is used as a feature vector, denoted by  $S$ :

$$S = t(LBPP,R(x, y)), x \in \{dRe, \dots, N-1-dRe\}, y \in \{dRe, \dots, M-1-dRe\} \dots(2.3)$$

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0. \end{cases} \dots(2.4)$$

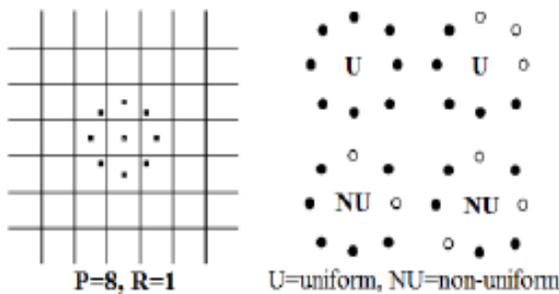


Figure 4. (Left) Constellation of neighborhood. (Right) Examples of 'uniform' and 'non-uniform' local binary patterns.

According to Equations (2.1) and (2.4), graylevel difference is first calculated between center pixel and its eight neighbors. The difference will then be binary quantized and coded, producing local binary patterns, which, in essence, form an 8-dimensional histogram with a total of 28 of 256 bins.

## 2.2. Feature Extraction Framework

Enlivened by the way that a very some of picture preparing calculations, for example, demosaicing, sifting, JPEG pressure, are square shrewd executed inside cameras, it is sensible to think about that some confined qualities or then again curios have been produced. These attributes or curios could be successfully caught by the uniform grayscale invariant neighborhood twofold examples, presented in Section 2.1. Grayscale invariance is accomplished by figuring contrast among focus and neighbor pixels' gray levels. This procedure to some degree stifles the impact of different picture substance. The presentation of 'uniform' nearby double examples empowers a characteristic component dimensionality decrease which is wanted by example order calculations. Along these lines, we propose to utilize the uniform grayscale invariant neighborhood double examples as highlights to catch camera display attributes.

As most of the camera image processing algorithms work in spatial domain, a good choice would be extracting features directly from graylevels of each color channel in spatial domain. From each color channel, a 59-dimensional LBP feature set is calculated by Equation (2.1) under the assumption of  $R = 1, P = 8$  (Each 59D LBP feature set are normalized to eliminate the influence of different image resolution). Besides, the same set of LBP features are extracted from prediction-error (PE) image. PE image is obtained by subtracting a predicted image from the original image. Considering a  $2 \times 2$  image pixel block, prediction of a pixel value is achieved

$$\hat{x} = \begin{cases} \max(a, b) & c \leq \min(a, b) \\ \min(a, b) & c \geq \max(a, b) \\ a + b - c & \text{otherwise} \end{cases} \dots(2.2.1)$$

where  $a, b$  are, respectively, the immediately horizontal and vertical neighbors of the pixel  $x$ .  $c$  is at the diagonal neighbor of  $x$ , and  $\hat{x}$  is the prediction value of  $x$ .

As some image processing algorithms differ largely at edges such as demosaicing and filtering, the prediction error image, which is, in essence, a spatial domain high pass filtered image, is another ideal choice to extract features from.

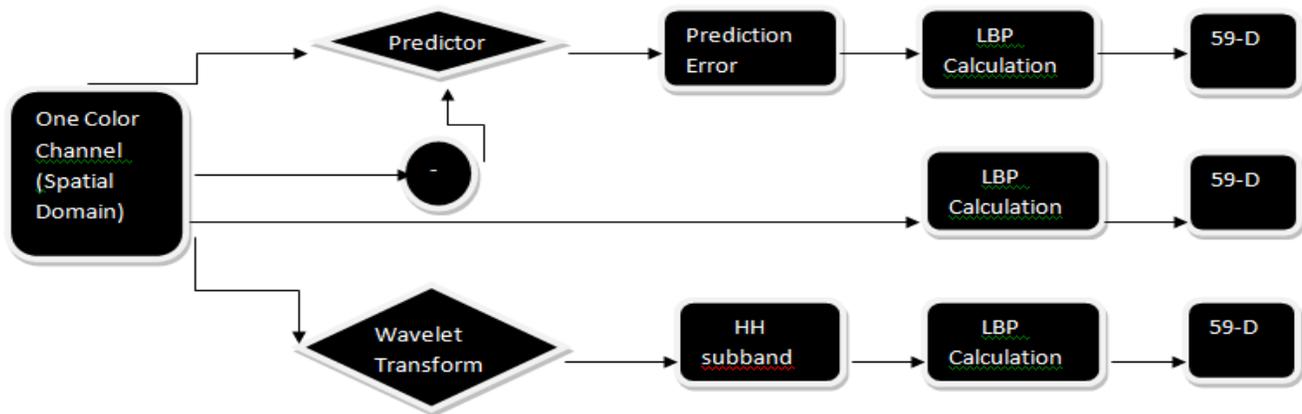


Figure 4. Feature extraction framework for one color channel.

One of the reactions of the element of LBP is its harshness to dim dimension change in spatial area. In spite of the fact that this could be a decent component for a few applications, it isn't wanted for camera display recognizable proof, as some picture preparing calculation, for example, gamma correction has spatial space monotonic nature and hence the distinction of these calculations couldn't be caught by our LBP highlights. To upgrade the separation capacity, notwithstanding the spatial space, wavelet area is considered and we propose to remove another 59-dimensional LBP highlight set from corner to corner subband (HH subband) of first dimension Haar wavelet change. parallel examples empowers characteristic element dimensionality decrease which is wanted by example order calculations. In this manner, we propose to utilize the uniform grayscale invariant nearby paired examples as highlights to catch camera show qualities.

To conclude, from each color channel, we extract LBP features from original image, its prediction-error 2D array, and its 1st-level diagonal wavelet subband, resulting in a total of  $59 \times 3 = 177$  features. The feature extraction framework of one color channel is shown in Fig. 4. Considering the fact that red and blue color channels usually share the same image processing algorithms, we only use red and green channels. Therefore, the final feature dimensions extracted from a color image is  $177 \times 2 = 354$ .

### III. EXPERIMENTS AND DISCUSSIONS

#### 3.1 Dataset for Experiments

Table 1. Experimental Dataset

List of Camera Models	# of cameras
Sony NEX-7	275
Motorola Moto X	275
Motorola Nexus 6	275

Motorola DROID MAXX	275
LG Nexus 5x	275
Apple iPhone 6	275
Apple iPhone 4s	275
HTC One M7	275
Samsung Galaxy S4	275
Samsung Galaxy Note 4	275

We picked the same 10 camera models from 'Dresden Image Dataset' as used in [10]. The number of camera devices for each model ranges from 2 to 5. The number of images per model is 275. All the images are direct camera JPEG outputs which are captured with various camera settings. Details are given in Table 1.

Images in the test set were captured with the same 10 camera models, *but using a second device*. For example, if the images in the train data for the iPhone 6 were taken with Ben Hamner's device (Camera 1), the images in the test data were taken with Ben Hamner's *second* device (Camera 2), since he lost the first device in the Bay while kite-surfing.

None of the images in the test data were taken with the same *device* as in the train data.

While the train data includes full images, the test data contains only single  $512 \times 512$  pixel blocks cropped from the center of a single image taken with the device. No two image blocks come from the same original image.

#### 3.2 Experimental Settings

In all of our experiments, Logistic Regression is trained and used as the classifiers for testing. From the whole dataset, we randomly select one camera for each model, and use all the images taken by the selected cameras for testing. Images from the rest of the cameras form the training data. This random selection procedure is iterated 20 times for each experiment.

Involving images from more than one camera of each model (except those have only 2 cameras) for training can greatly reduce the chance of overtraining[9]. Using the cameras that

are not involved in the training procedures for testing makes the experiments more practical [13].

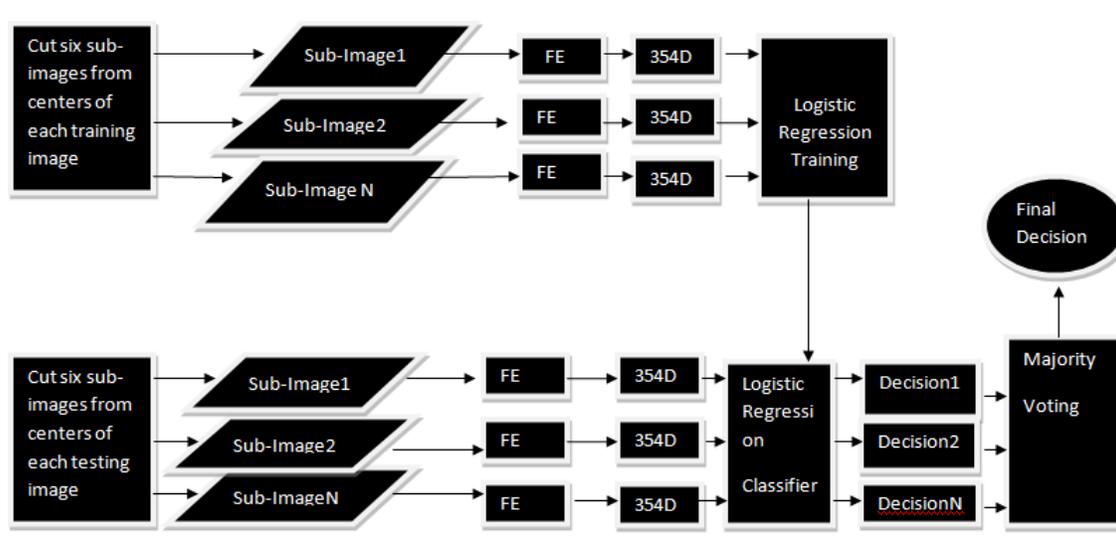


Figure 5. Block diagram of training and testing stages. FE=feature extraction.

In each iteration, images for both training and testing are cut into six sub-images from centers. The final decision in testing stage is made for each image by majority voting based on the six individual decisions. This cropping and voting procedure not only increases the number of samples for training, but also brings robustness against the regional anomalies in testing images. A block diagram is shown in Fig. 5 which includes both the training and testing stages (only one image is shown in the testing stage).

### 3.3. Results and Discussions

The proposed was tested on dataset of 10000 images with the 10 cross fold validation technique. This approach involves randomly dividing the set of observations into 10 groups, or

folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining  $k - 1$  folds.

The choice of  $k$  is usually 5 or 10, but there is no formal rule. As  $k$  gets larger, the difference in size between the training set and the re-sampling subsets gets smaller. As this difference decreases, the bias of the technique becomes smaller.

A confusion matrix is a table that is often used to **describe the performance of a classification model** (or "classifier") on a set of test data for which the true values are known.

As we can see in the below figure , from that we achieved 60 Percent accuracy through the confusion matrix via logistic regression classifier model.

```
# Fitting Logistic Regression to the Training set
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(x_train, y_train)
y_pred = classifier.predict(x_test)

# Use score method to get accuracy of model
score = classifier.score(x_test, y_test)
print(score)

import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import metrics

cm = metrics.confusion_matrix(y_test, y_pred)
print(cm)

0.6
[[2 0]
 [2 1]]
```

In the future, the system will be extended with the different classifier model to achieve better accuracy. We will use more features to get perfect accuracy with Support Vector Machines and Neural Networks.

#### REFERENCES

- [1] Mehdi Kharrazi, Husrev T Sencar, Nasir Memon, "BLIND SOURCE CAMERA IDENTIFICATION", IEEE, 0-7803-8554-3/04/\$20.00, 2004
- [2] J. Adams, K. Parulski, and K. Spaulding, "Color processing in digital cameras," *Micm*, IEEE, vol. 18, pp.20-30, Nov.-Dec 1998.
- [3] Z. Deng, A. Gijssenij, and J. Zhang, "Source camera identification using auto-white balance approximation," in *Proc. 13th IEEE Int. Conf. Comput. Vis., Barcelona, Spain, Nov. 2011*, pp. 57-64.
- [4] Amerini I, Becarelli R, Bertini B, Caldelli R (2015) Acquisition source identification through a blind image classification. *IET Image Process* 9(4):329-337
- [5] Avciabas, I, Memon N, Sankur B (2003) Steganalysis using image quality metrics. *IEEE Trans Image Process* 12(2):221-229.
- [6] Bayram S, Sencar H, Memon N (2006) Improvements on source camera-model identification based on CFA. In: *Advances in Digital Forensics II*, IFIP international conference on digital forensics, pp 289-299.
- [7] Z. Deng, A. Gijssenij, and J. Zhang, "Source camera identification using auto-white balance approximation," in *Proc. 13th IEEE Int. Conf. Comput. Vis., Barcelona, Spain, Nov. 2011*, pp. 57-64.
- [8] *Camera & Imaging Products Association, Exchangeable Image File Format for Digital Still Cameras: Exif Version 2.3*, CIPA DC-008-2010 & JEITA CP-3451B Standard, 2010.
- [9] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*. San Francisco, CA, USA: Morgan Kaufmann, 2002.
- [10] *Camera & Imaging Products Association, Exchangeable Image File Format for Digital Still Cameras: Exif Version 2.3*, CIPA DC-008-2010 & JEITA CP-3451B Standard, 2010
- [11] G. Xu and Y. Q. Shi, "Camera model identification using local binary patterns," in *International Conference on Multimedia and Expo (ICME), 2012 IEEE*. 2012, pp. 392-397
- [12] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Determining Image Origin and Integrity Using Sensor Noise," *Information Forensics and Security, IEEE Transactions on*, vol. 3, pp. 74-90, 2008
- [13] A. E. Dirik, H. T. Sencar, and N. Memon, "Digital Single Lens Reflex Camera Identification From Traces of Sensor Dust," *Information Forensics and Security, IEEE Transactions on*, vol. 3, pp. 539-552, 2008.
- [14] S. Bayram, H. Sencar, N. Memon, and I. Avciabas, "Source camera identification based on CFA interpolation," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on, 2005*, pp. III-69-72.
- [15] Y. Long and Y. Huang, "Image Based Source Camera Identification using Demosaicking," in *Multimedia Signal Processing, 2006 IEEE 8th Workshop on, 2006*, pp. 419-424.
- [16] A. Swaminathan, W. Min, and K. J. R. Liu, "Nonintrusive component forensics of visual sensors using output images," *Information Forensics and Security, IEEE Transactions on*, vol. 2, pp. 91-106, 2007.
- [17] F. Marra, D. Gragnaniello, L. Verdoliva, On the vulnerability of deep learning to adversarial attacks for camera model identification, *Signal Processing: Image Communication* (2018), <https://doi.org/10.1016/j.image.2018.04.007>.
- [18] Z. Deng, A. Gijssenij, and J. Zhang, "Source camera identification using auto-white balance approximation," in *Proc. 13th IEEE Int. Conf. Comput. Vis., Barcelona, Spain, Nov. 2011*, pp. 57-64.
- [19] *Camera & Imaging Products Association, Exchangeable Image File Format for Digital Still Cameras: Exif Version 2.3*, CIPA DC-008-2010 & JEITA CP-3451B Standard, 2010.
- [20] E. Kee, M. K. Johnson, and H. Farid, "Digital Image Authentication from JPEG Headers," *Information Forensics and Security, IEEE Transactions on*, vol. 6, pp. 1066-1075, 2011.
- [21] M. C. Stamm and K. J. R. Liu, "Forensic detection of image manipulation using statistical intrinsic fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 3, pp. 492-506, Sep. 2010.
- [22] G. Xu, S. Gao, Y. Q. Shi, R. Hu, and W. Su, "Camera-Model Identification Using Markovian Transition Probability Matrix," in *Digital Watermarking*. vol. 5703, ed: Springer Berlin / Heidelberg, 2009, pp. 294-307.
- [23] T. Gloe, K. Borowka, and A. Winkler, "Feature-Based Camera Model Identification Works in Practice," in *Information Hiding*. vol. 5806, ed: Springer Berlin / Heidelberg, 2009, pp. 262-276.
- [24] M. J. Weinberger, G. Seroussi, and G. Sapiro, "LOCO-I: a low complexity, context-based, lossless image compression algorithm," in *Data Compression Conference*.
- [25] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. *Proc. of ICIP*, 1:900:903, 2002.
- [26] A. Hadid, M. Pietikainen and T. Ahonen. A Discriminative Feature Space for Detecting and Recognizing Faces. *Proc of CVPR* 2004.