MO444 - Pattern Recognition and Machine Learning
Practical Assignment #2

Osvaldo Xavier Dias Junior[*]
Prof. Anderson Rocha[†]

Abstract

This report is regarding the second assigment of the module MO444 (Pattern Recognition and Machine Learning), and it presents results of logistic regression, k-nearest neighbors and neural network models applied to the Kaggle challange "IEEE's Signal Processing Society - Camera Model Identification". This challenge is about identifying the camera that an image was taken from. This kind of problem envolved feature extraction, image processing and finding the most suitable classifier. The best accuracy obtained was not very good mainly because of the feature extraction (feature engineering), althought the process of feature engineering improved the results for different techniques.

1. Introduction

The logistic regression is a mathematical model that correspond to a linear relationship of parameters and inputs, which interacts with an exponential function as follows:

$$h_\theta(x) = \frac{1}{1 + e^{-\theta x}} \tag{1}$$

$$z = \theta x \tag{2}$$

Where x is the vector of the inputs, θ is the vector of the parameters and h_θ is the vector of predicted targets. This model can be transcripted to an algorithm to do continuos value prediction, however due to the characteristic of the sigmoid function (1) it provides a pratically binary value as output, once a small variation in z, positive or negative, regarding the origin results in saturation of the function into one of its peaks, therefore this function is used to classification.
K-neareast neighbors is another classifier, which has a very simple algorithm that given a new example executes a search for a defined number of nearest neighbors (classes) in the feature space. Among all these neighbors the class with more aperances is the class defined to this example. This algorithm can use to find the nearest neighbors euclidean distance, manhattan distance, minkowski distance or any other suitable distance measure.
The MLP classifier is a neural network algorithm which is composed of perceptrons organized in a multilayer composition. This multilayer compostion is divided in three parts: the first layers are the inputs, the last layer are the outputs and every layer between the first and the last one are the hidden layers. As explained, each layer has a number of perceptrons, which is a unit of processing that receives weighted inputs and apply a non linear function to this data. The function can be chosen arbitrarily according to the type of the machine/deep learning problem, one example of this function is the rectifier linear unit (ReLU) as follows:

$$f(z) = max(0, z) \tag{3}$$

The problem under discussion in this report is about predict the camera of a given picture taken. The dataset of this problem envolves 275 pictures taken from 10 different cameras: Motorola Nexus 6, Motorola DROID MAXX, LG Nexus 5x, Apple iPhone 6, Apple iPhone 4s, HTC One M7, Samsung Galaxy S4 and Samsung Galaxy Note 3.
Regarding the testing dataset, a subpart os the dataset was modified using resize, gamma modifications and compression. To perform such classification was used the classifier algorithms described before and these algorithms executed on top of other feature extractor algorithms such as local binary patterns and wavelets features.

2. Proposed Solutions

The solutions proposed were based of classification models: logistic regression, k-nearest neighbors and neural network, each of the classifiers executes on top of feature extractors. It follows a list of techniques applied, in chrological sequence, to the dataset to achieve classification:

[*]Is with the Institute of Computing, University of Campinas (Unicamp). **Contact**: `osvaldoxdjr@gmail.com`

[†]Is with the Institute of Computing, University of Campinas (Unicamp). **Contact**: `anderson.rocha@ic.unicamp.br`

1. Feature Extraction - Local Binary Patters
2. Feature Extraction - Wavelets Features
3. Feature Scaling
4. Trainning and Validation - Logistic Regression
5. Trainning and Validation - K-nearest neighbors
6. Trainning and Validation - Neural Networks

3. Experiments and Discussion

To build the algorithms and image processing the programming language chosen was Python, inside an Ubuntu enviroment. The first goal to achieve is to perform a feature engineering, which is obtain features using the examples given in the dataset. The first idea was to extract the noise of the each picture of both trainning, validation testing dataset, because dealing with just the noise is more appropriate to the given problem, because digital cameras leave a footprint of each picture taken, the reason for that is because the internal circuits and image processing of the camera tends to be unique or partially unique from each camera model.
Hence, it was loaded sequentially the pictures from each camera, but dealing with the original dimension of the pictures from dataset would be overwhelming to process, thus each picture was cropped from the trainning dataset to be 512x512 pixels of dimension, which is by the way the dimensionality of the testing dataset pictures. The size of cropping was chosen euristically respecting a trade-off between performance and processing time. In the sequence is shown an example of an dataset's input:

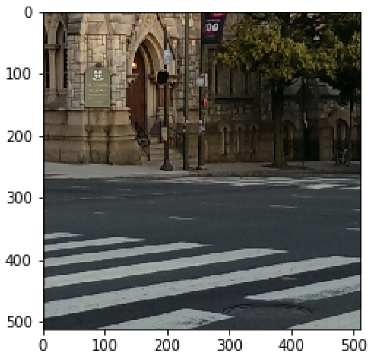

Figure 1. Example of a picture taken with the camera HTC-1-M7

Once the picture was loaded and cropped it was needed to remove the noise of the picture. To do this noise removal was applyed a waveltes denoise of the image, after that subtract the raw cropped image to the denoised image, therefore the result of this operation was the noise of the picture. The figure (2) shows the noise of the picture shown in figure (1).

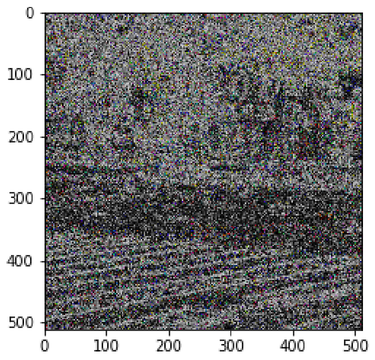

Figure 2. Noise picture taken with the camera HTC-1-M7

With all the images pre processed it was mandatory to obtain the features from these noise images. For this purpose, it was implemented a method called local binary patterns (LBP), this method is related to feature extraction that computes local representation of texture.
The resulting image from a LBP feature is an image with the same dimensionality and each pixel of this resulting image is constructed by comparing the pixels of the original image with its surrounding neighborhood of pixels. An histogram should be created to obtained the features themselves (values of bins), so it is possible to create an histogram from 0 to 255 that are the possible range of the resulting pixels in the LBP transformed image, and set a number of bins, a radius for the surroundings and a method. In this experiments the parameters were: 26 bins, radius of 10 and an uniform method.
It is important to highlight that this method should be applied to grayscale images. But the noise images obtained in the previous image processing are composed of RGB channels, then the algorithm was passed trought each channel at a time, this kind of extraction produced 78 features, being 26 features per channel.
Another feature extractor used in this problem was the wavelets features extractor, this feature extractor uses wavelet decomposition, to each RGB channel, to obtain as an output an aproximate image, vertical details of image, horizontal details of image and diagonal details of an image. All these outputs are resulting images that can be a good image description, to obtain these images the algorithm basically apply low pass filters and high pass filters cascated on rows and columns to obtain a frequency description of useful information.
The resulting images from the wavelet decomposition are not the features, to reach the features it was proposed in

this experiments to calculate the 9 central moments (mean, variance, kurtosis, skew and etc.) and also another type of calculation of kurtosis and skew in the vertical details of image, horizontal details of image and diagonal details. Joining all the features together it results in 99 extra-features, besides that ones obtained throught LBP feature extractor.
Then, possessing 125 features (LBP + Wavelets) it was necessary to perform a feature scaling to adjust the emphasis of the features, because there was a considerable variance among each feature. It was chosen the z-score feature scaling:

$$X_n = \frac{X_n - \bar{X}}{\sigma} \tag{4}$$

Where X_n is a single feature, $\bar{X}$ is the mean of the set of features and σ is the standart deviation. At this point, the features extraction was concluded for the trainning, validation and testing dataset and in the sequence it was applied classifiers on top of theses features for trainning, validation and testing. The following table presents the results of trainning and validations for different configurations of classifiers and features group:

Table 1. Scores of trainning and validation

	SCORE					
	LBP		WAV		LBP + WAV	
	T	V	T	V	T	V
LR	0.33	0.32	0.44	0.41	0.54	0.54
KNN	0.54	0.31	0.62	0.42	0.69	0.55
MLP	0.57	0.39	0.56	0.47	0.83	0.62

Where the features group are LBP (local binary patterns), WAV (wavelets features) and LBP+WAV (local binary patterns joint with wavelets features). There are two types of evaluating the score: T (trainning) and V (validation), and finnally the classifiers: LR (logistic regression), KNN (k-nearest neighbors) and MLP (multilayer perceptron). To came up with the validation dataset it was spplited the trainning dataset into two: 80% for trainning and 20% for validation.
Once the logistic regression is a binary classifier, therefore it only classify two classes each time, for our problem it was needed to perform an one vs all algorithm, which is to classify each class to the other rest and choose the classification that maximizes the propability of been that given class. While the KNN classifier was set up to compare the new example with 5 nearest neighbors to perform the classification, the euclidean distance was the metric to get this 5 nearest neighbors. The MLP was configured with 200 hidden layers and relu as the activation function.
Analyzing the results of table 1 is possible to see that there were no overffiting during trainning nor during validation to the three classifiers and for the three different set of features. LBP features gave the worst results, while the combination of LBP and WAV features gave the best results, in the middle remains just the wavelets features with intermediate results. This happened because increasing the number of features the model trainned better, or in other words, increasing the complexity of the model the model became more accurate in prediction.
Another interesting aspect that table 1 shows is that the classifier that presents the best results is the MLP, this is expected because this classifier is the most complex among the three, it has several hidden layers and many activation function that can learn the non linearity of the problem. The logistic regression presented the worst result because of its simplicity, the logistic regression can be viewed as an unit of a multilayer perceptron. The KNN provided an intermediate due to the fact that is not also a very complex classifier.
Well, after perform all trainning and validation for the different configurations it was time for perform a testing and submit it to the Kaggle's platform. In the sequence, it is shown the results provided by the submission on kaggle's competition "IEEE's Signal Processing Society - Camera Model Identification":

Table 2. Kaggle's score of testing dataset

	SCORE	
	LBP + WAV	
	Private	Public
LR	0.250714	0.265833
KNN	0.234404	0.244166
MLP	0.310119	0.285000

For comparison purposes it was tested each classifier for the best set of features (LBP + WAV). Confirming what was seen in the trainning and validation the MLP classifier gave the best test result, however surprinsingly the logistic regression performed better than the k-nearest neighbors.
Other advantege of MLP comparing to the other classifiers is that the private score is grater than the public score, this is very good for kaggle competitions because the competitors tends to overfit the public score.
The results of testing were not good enough mainly because of the features enginnering performed, the results shown in table 1 were indicating that the testing results might not be good. Another aspect that was neglected during trainning was the modifications of the test dataset pictures (resize, gamma modifications and compression) were not considered while the models were trainned, this obviously helped to achieve a not so good results.

4. Conclusions and Future Work

This experiments performed in order to suceed in the kaggle's competition were very interesting. The feature engineering without using convolution neural networks (neu-

ral networks that does feature extraction and trainning at the same time) is a very difficult task, mainly because it demands some previous knowlegde about the problem, in the specific case it meant to know that digital cameras leave a footprint of each picture taken, the reason for that is because the internal circuits and image processing of the camera tends to be unique or partially unique from each camera model. This is not trivial at all, it demands seek for specialist information and a deep study of the problem, reading related papers and so on.
Therefore, the feature enginnering performed in this experiments were not good enough to reach a good score at Kaggle's competition.
Running several classifiers shown that the results related to each classifier will depend of its topology. The parameters of the classifier should be chosen carefully to proper classification, avoiding overfitting and underfitting.
Possible future work is to try to consider modifications of the test dataset pictures in the trainning model and to apply correlation to gain more features.

5. References

1. http://www.ic.unicamp.br/ rocha/teaching/2018s1/mo444/index.html

2. Pattern Recognition and Machine Learning. Christopher M. Bishop. Springer. (2006) Acrononym: PRML