A Reality Check of Images

Abhishek Jain
13027

Akhil Garg
13065

Kanhaya Namdhar
13337

Sumit Kumar
13720

Varun Gupta
13771

1 Introduction

The availability of low-cost, efficient and powerful content editing tools has enabled the computer generated images (CG) to a degree of unrivaled realism. Differentiating a photo-realistic computer generated (CG) image from a real photograph (PG) is almost impossible for a naked human eye to detect.

The aim of the project is to build a classifier that distinguishes real images from artificial images. In this problem, we consider any image originated from an acquisition device like a camera as a real image or a photograph (PG). In turn, any scene partially or totally rendered by a computer software is an artificial image or a computer-generated graphic (CG).

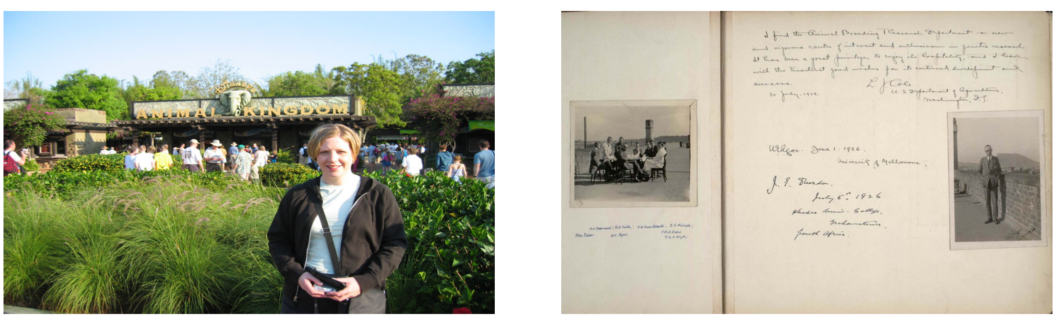

Figure 1: Photographic Images

Figure 2: Computerized Images

2 Literature Review

Tokuda et al.[1] has presented a complete study of various methods present in literature to distinguish between CGs and PGs. They have also compared the implemented methods using the same validation environment showing their pros and cons with a common benchmark protocol. Moreover, they have also performed fusion of various methods to classify the two classes and claimed that the ensembling approach achieves higher accuracy compared to the individual methods on the same test set. The problem of categorization of images as CG or PG has been approached in various forms.

The human visual system is quite complex and uses many visual features to classify a scene. Using it as inspiration, they tried to identify the visual features that distinguish between PG and CG and use them in their approach. Edges, colors and shapes are examples of visual characteristics that could be used. To improve the current results, a naive approach is to use the new and more relevant features. Another enriched approach could be a novel way of combining the existing methods, such as an ensemble of them. Most of the existing proposals for distinguishing CGs and PGs in the literature contemplate two steps:

- Identification and extraction of features that reveal the differences between the two classes (CG vs. PG) and
- Classification of images based on the set of extracted features.

The main difference between the various methods in the literature is the choice of the characteristics to describe an image (descriptor). The effectiveness of this process is fundamental to obtaining a good generalized accuracy of a method. In this work, we have implemented the method proposed by Ng et al.[2] They have proposed a new geometry-based image model, motivated by the physical image generation process, to tackle the problem. The proposed model exploits the following differences between CGs and PGs:

- *Object Model Difference*: Formation of fractal surfaces take place on real-world objects due to erosion, aggregation and fluid turbulence. However, the computer graphics 3D objects are often represented by polygonal models to reduce memory requirement and computational load.
- *Light Transport Difference* : The physical light field captured by a camera is a result of the physical light transport from the illumination source, reflected to the image acquisition device by an object. However for CGs, a simplified model based on isotropy, spectral independence and parametric representation is often used.
- *Acquisition Difference* : PG carry the characteristics of the imaging process while CG may undergo different types of post-processing.

Features for distribution of local patches include moments of inertia, center of mass and mean and variance of the distance of data-points from the center of mass. Features for distribution of fractal dimension, surface gradient, the second fundamental theorem and the Beltrami flow vector include the first 4 moments namely mean, variance, skewness and kurtosis.[3]

3 Methodology

For the dataset part, we used **Columbia Photorealistic Computer Graphics**[5] dataset comprising of 800 CGs and 800 PGs. From each image, we created a 192-D feature vector and so we accumulated data in the form of 1600x192 matrix. We used a ratio of 60:20:20 for training, validation and test dataset respectively. The implementation was done using MATLAB and Python. For the purpose of binary classification, we experimented with the following individual methods:

3.1 Naive Bayes

In machine learning, Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the feature

3.2 Decision Tree

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning.

3.3 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

3.4 Logistic Regression

In statistics, logistic regression or logit regression or logit model is a regression model where the dependent variable (DV) is categorical. Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution

3.5 Linear Discriminant Analysis (LDA)

Linear discriminant analysis is a generalization of Fisher's linear discriminant, a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.

3.6 Quadratic Discriminant Analysis (QDA)

Quadratic discriminant analysis is closely related to linear discriminant analysis, however, in QDA there is no assumption that the covariance of each of the classes is identical. Hence QDA is able to learn more flexible models, but has the disadvantage of having to learn more number of parameters.

3.7 Support Vector Machine (SVM)

Support Vector Machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

3.8 Feedforward Neural Network (FFNN)

Feed Forward Neural Networks are composed of several perceptron like units arranged in multiple layers. It consists of an input layer, one or more hidden layers (which compute a non-linear transform of the inputs) and a output layer. All nodes between layers are assumed connected to each other.

3.9 Ensembling

Ensemble methods are learning algorithms in machine learning that evolve a classifier from a set of basic classifiers. Dietterich [4] reviews the ensembling methods and explains why ensembling method can perform better than any single classifier. We experimented with four ways of performing ensembling: two variants of averaging of predictions of multiple pre-trained models and two variants of stacking methods. We considered a set of eight of classifiers namely: Naive Bayes, Decision Tree, Random Forest, LDA, QDA, SVM with polynomial kernel of degree 2, SVM with rbf kernel and Neural Network.

3.9.1 Ens1

The first ensembling method (Ens1) does simple voting without any weighting. It is the simplest implementation among the all the ensembling methods. Once all the individual methods have been implemented, we performed the classification of each method. An image is classified as PG or CG for each of the k implemented methods and each of these classifications is called a vote. The class with the highest number of votes is elected the class of our classifier. Let $vot_i(Im)$ be the vote of the classifier i in the image Im. We define

$$vot_i(Im) = \begin{cases} 1 & \text{if the vote is for class CG} \\ 0 & \text{if the vote is for class PG} \end{cases}$$

We define the rule of the classifier Ens1 as:

$$H = \begin{cases} \text{CG if } \Sigma vot_i(Im) > k/2 \\ \text{PG otherwise} \end{cases}$$

3.9.2 Ens2

The second ensembling method applies weighted voting and combines individual hypotheses or classifier to obtain a final hypothesis. The models are assigned weights based upon their performance on the test set. The model with higher test accuracy will be given higher weight in determining the overall mandate of the final hypothesis. Let $A(i)$ be the mean accuracy of the i^{th} hypothesis from among $|H|$ hypothesis. Then, the weight of each hypothesis can be formulated as

$$w(i) = \frac{A(i)}{\sum_{i=1}^{|H|} A(i)}$$

Let $vot_i(Im)$ be the vote of an image Im to be classified by the method i. Using the weighted voting, we perform the classification by each method i and obtain the weighted voting $w_i * vot_i(Im)$. The decision rule for this hypothesis can be given as:

$$H = \begin{cases} \text{CG if } \sum_{i=1}^{|H|} w_i \times vot_i(Im) \geq 0.5 \\ \text{PG if } \sum_{i=1}^{|H|} w_i \times vot_i(Im) < 0.5 \end{cases}$$

3.9.3 Ens3

The motivation for formulating another voting method is that each method can provide more than a simple binary vote. Instead of using a simple vote, we used a classification with probability of prediction named as *confidence*. A confidence value of greater than 0.5 means the classifier believes the example to be a CG and vice versa. All the individual methods except SVM and FFNN automatically provide this confidence score. For these two methods, we used sigmoid function to scale the score between 0 and 1 accordingly.

Let $vot_i(Im)$ be the vote of an image Im to be classified by the method i and $c_i(Im)$ be the confidence score given by classifier i on image Im. The decision rule for this hypothesis can be given as:

$$H = \begin{cases} \text{CG if } \sum_{i=1}^{|H|} w_i \times c_i(Im) \geq 0.5 \\ \text{PG if } \sum_{i=1}^{|H|} w_i \times c_i(Im) < 0.5 \end{cases}$$

3.9.4 Ens4

Ens4 is a stacking method in which the binary votes of all the classifiers are stacked together to form an 8-D vector. This will be the feature vector used by another classifier in a meta-level. Although any supervised learning classifier will serve the purpose, we used SVM classifier with rbf kernel to classify the examples in this meta-level.

3.9.5 Ens5

This stacking method is similar to the previous one in the structure and the classifier used. However, unlike the Ens4 method the features of the meta-classifier are not binary vectors. Instead of vote of individual classifiers on an image Im, we have used the confidence score of the classifier on that image.

4 Results

All the methods explained above were implemented and the corresponding hyperparameters were selected via cross-validation. The results obtained for the individual classifiers are summarized below:

Method	Accuracy	Precision	Recall	F-Score
Decision Tree	.71	.74	.70	.72
Naive Bayes	.74	.79	.68	.73
Random Forest	.76	.74	.84	.78
Logistic Regression	.78	.80	.76	.78
QDA	.84	.85	.83	.84
SVM (rbf)	.84	.86	.83	.84
SVM (poly-2)	.84	.85	.84	.84
Neural Network	.85	.83	.86	.84
LDA	.86	.87	.84	.85

Table 1: Statistics of individual methods

In a similar fashion, the five ensemble methods explained above were implemented and the following results were obtained:

Method	Accuracy	Precision	Recall	F-Score
Ens1	.84	.85	.83	.84
Ens2	.85	.86	.85	.85
Ens3	.86	.87	.85	.86
Ens4	.87	.87	.85	.86
Ens5	.88	.89	.88	.88

Table 2: Statistics of Ensemble methods

ROC curve is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.
The ROC curves for Ens2 and Ens4 methods are shown in Figure 3 and Figure 4 respectively. For each of these plots, we find that the ROC curve for the final ensemble lies above that of standard SVM with rbf kernel model. In Figure 3, the AUC for the ensemble hypothesis is 0.923 and that of the simple SVM with rbf kernel is 0.896.

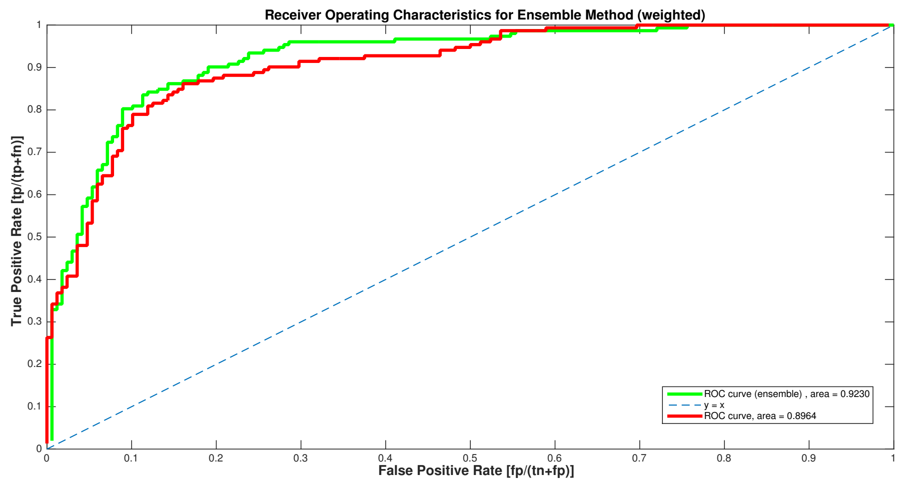

Figure 3: ROC curve of Averaging method for Ensembling

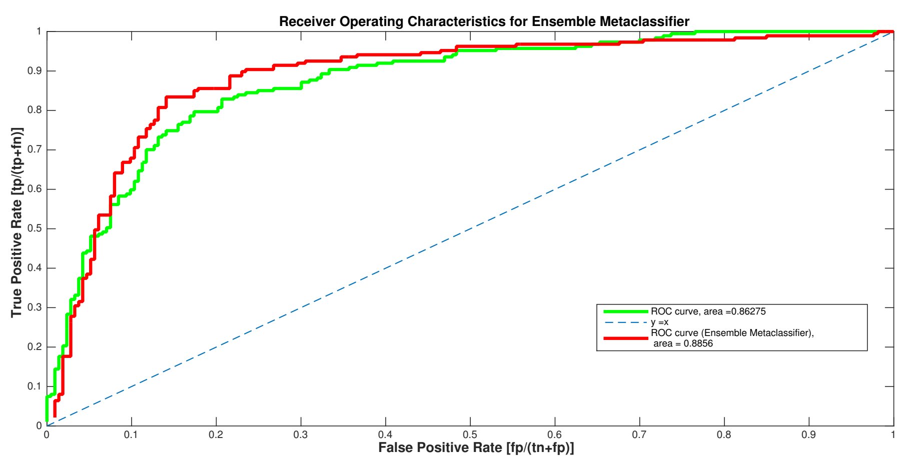

Figure 4: ROC curve for the meta-classifier Ensemble

5 Discussion

Several conclusions can be drawn from the results obtained. Firstly, since the dataset used had equal number of CGs and PGs, the Precision, Recall and F-Score values obtained are very close to accuracy, and hence test accuracy is a sufficient criterion for rating different methods. Moreover, all the ensemble methods perform considerably better than the individual classifiers. Among the ensemble methods, as we collect more data in the successive ensemble methods, the test accuracy improves, which is expected.

Things We Learnt

Throughout the project, we studied and understood various Machine Learning Algorithms in the literature and their implementation on a particular dataset. We analyzed the performance of all these methods and tuned their hyperparameters to achieve optimum performance. Furthermore, we understood the fundamental differences between a Computer Graphic and a Photograph. Literature review of the techniques used to classify the two categories taught us about the various methods used so far to use feature for the purpose.

Acknowledgment

We would like to thank our instructor, **Dr. Piyush Rai**, *Department of Computer Science and Engineering, IIT Kanpur*, for motivating all the students towards this evolving field of Machine Learning by providing them with meaningful and interesting course projects. This project could have only been possible because of the teaching methodology that he adopts which makes beginners like us to explore this field in depth.

We would like to thank Columbia University, New York for providing the dataset We would also like to thank our colleague Nishant Rai, *Department of Computer Science and Engineering, IIT Kanpur* and Harkirat Behl, *Department of Computer Science and Engineering, Oxford University* who helped us during the initial stages of the project. Their contribution towards our project is really worth recognizing.

References

[1] Tokuda, Eric, Helio Pedrini, and Anderson Rocha. "Computer generated images vs. digital photographs: A synergetic feature and classifier combination approach." Journal of Visual Communication and Image Representation 24.8 (2013): 1276-1292.

[2] Ng, Tian-Tsong, et al. "Physics-motivated features for distinguishing photographic images and computer graphics." Proceedings of the 13th annual ACM international conference on Multimedia. ACM, 2005.

[3] B. B. Mandelbrot. The fractal geometry of nature .San Francisco: W.H. Freeman, 1983

[4] Dietterich, Thomas G. "Ensemble methods in machine learning." International workshop on multiple classifier systems. Springer Berlin Heidelberg, 2000.

[5] Tian-Tsong Ng, Shih-Fu Chang, Jessie Hsu, Martin Pepeljugoski, Columbia Photographic Images and Photorealistic Computer Graphics Dataset. ADVENT Technical Report 205-2004-5, Columbia University, Feb 2005.