ECE 759 Project

Spring 2018

The goal of this project is to help a student acquire a deeper understanding of ML theoretical concepts and foundations, and appropriately apply in realistic scenarios with practical inference applications, more specifically image content classification. This will particularly focus on specific image data bases, each with its own specificities. A set of 3 papers are included, more as references and guidance to tools of specific interest in this project. The image datasets include MNIST (digit data base), Extended YaleB (human face database), and Caltech10 (various objects database). A randomly selected ***Half*** of each dataset is meant for training and the other random ***Half*** is meant for testing.
The project includes 2 parts, each due within 5 weeks. The project will be conducted in teams of 2 students, and each team is responsible for the implementation, experiments, documentation and a clean and clear typeset report. The students are urged to team up with a classmate. Two datasets per team will be randomly assigned, as well as two algorithms will be randomly assigned. A 10-15 min presentation will be required at the end of the term.

Project-Part1 Deadline: March 16th

1. **Feature Selection**

   There are a great many ways of selecting features and some are data dependent.

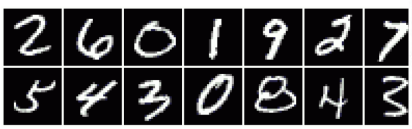
   a. Explain both theoretically as well as qualitatively the rationale for your choice. Detail your work to allow a technically capable individual to understand it.

2. **Algorithmic Implementation**

ECE 759 Pattern Recognition and Machine Learning

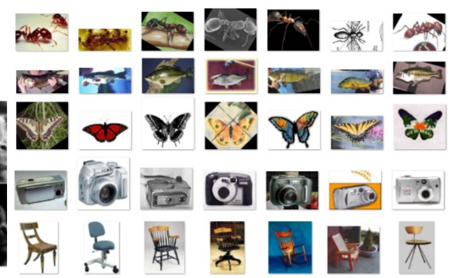
Program the randomly assigned algorithms using your favorite programming language and provide the pseudo codes in the report- - you are allowed to use algebraic functions routines/libraries, but the Algorithm code should be your own code.

a. And explain how you implemented the algorithm, what are the inputs and outputs,
b. Explain any iteration and updates of variables
c. Highlight all critical steps for smooth run of algorithms



Example of MNIST

Example of Extended YaleB



Example of Catech10

ECE 759 Pattern Recognition and Machine Learning

Project-Part2 **Deadline: April. 20th**

Building on Part 1 of the project, and your understanding of the classifiers you have implemented, you are to demonstrate your understanding by validating the choice of hyper-parameters of each algorithm and its associated classification performance.

1. **Cross-Validation**

   In order to find the best performance as well as most generalized model for your algorithm, a 5-fold cross validation is to be applied in your training phase. Explain how this Cross-Validation helps you find your hyper parameters.

2. **Demonstration of Performances**

   a. A "demonstration" implementation of the algorithm should be run as a command to train the classifier, test the classifier, print the performance on training and testing datasets
   b. Draw the following graphs, a) convergence, b) performance vs. time, c) performance vs. different hyperparameters.

3. **Analysis of Results**

   a. Based on the performance evaluation in **Q.2**, analyze the results by discussing the advantages as well as drawbacks of each algorithm, and how they may arise
   b. To avoid such drawbacks, how would you go about improving an overall smooth performance?

A Comparison of Generic Machine Learning Algorithms for Image Classification

Raphaël Marée, Pierre Geurts, Giorgio Visimberga †,
Justus Piater, Louis Wehenkel
Montefiore Institute, University of Liège
Liège, Belgium
http://www.montefiore.ulg.ac.be/~maree/

† On leave from Politecnico Di Bari
Bari, Italy

Abstract

In this paper, we evaluate 7 machine learning algorithms for image classification including our recent approach that combines building of ensembles of extremely randomized trees and extraction of sub-windows from the original images. For the approach to be generic, all these methods are applied directly on pixel values without any feature extraction. We compared them on four publicly available datasets corresponding to representative applications of image classification problems: handwritten digits (MNIST), faces (ORL), 3D objects (COIL-100), and textures (OUTEX). A comparison with studies from the computer vision literature shows that generic methods can come remarkably close to specialized methods. In particular, our sub-window algorithm is competitive with the state of the art, a remarkable result considering its generality and conceptual simplicity.

1 Introduction

Image classification is an important problem which appears in many application domains like quality control, biometry (face recognition), medicine, office automation (character recognition), geology (soil type recognition)...

This problem is particularly difficult for traditional machine learning algorithms mainly because of the high number of input variables that may describe images (i.e. pixels). Indeed, with a high number of variables, learning methods often suffer from a high variance (models are very unstable) which deteriorates their accuracy. Futhermore, computing times can also be detrimental in such extreme conditions. To handle this high dimensionality, image classication systems usually rely on a pre-processing step, specific to the particular problem and application domain, which aims at extracting a reduced set of interesting features from the initial huge number of pixels. The limitation of this approach is clear: When considering a new problem or application domain, it is necessary to manually adapt the pre-processing step by taking into account the specific characteristics of the new application.

At the same time, recent advances in automatic learning have produced new methods which are able to handle more and more complex problems without requiring any a priori information about the application. These methods are increasingly competitive with methods specifically tailored for these domains. In this context, our aim in this paper is to compare several of these recent algorithms for the specific problem of image classification. Our hypothesis is that it will be possible to obtain with some of these approaches competitive results with specialized algorithms without requiring any laborious, manual pre-processing step.

In this goal, our approach was to choose several problems which we think are representative of the image classification domain and then to apply several learning algorithms on each of these problems, without any specific pre-processing, i.e. by directly using the pixel values. Among recent learning algorithms potentially able to handle this complex problem, we have chosen a panel of 7 algorithms, including one generic approach that we have recently proposed for image classification [17]. These algorithms will be compared essentially on two criteria: accuracy of the models and computational efficiency (of the learning and test phases). Although several of these algorithms have been already applied to image classification, usually these studies either do not compare several algorithms or are focused on only one particular application problem.

The paper is structured as follows. In Section 2, we briefly describe the machine learning algorithms chosen for our comparison. Of course, we spend more time on the description of our own proposal. The essential characteristics of the four datasets used for the comparison are summarized in Section 3. The experimentation protocols and the results of the experiments are discussed in Section 4. We end the paper with our conclusions and discusssions about future work directions.

2 Algorithms for generic image classification

The input of a generic learning algorithm for image classification is a training set of pre-classified images,

$$LS = \{(A^i, c^i), i = 1, \dots, N\}$$

where A^i is a $W_x \times W_y$ matrix describing the image and $c^i \in \{1, \dots, M\}$ is its classification (among M classes). The elements $a^i_{k,l}$ of A^i ($k = 1, \dots, W_x$, $l = 1, \dots, W_y$) describe image pixels at location (k, l) by means of an integer value in the case of grey level images or by means of 3 integer RGB values in the case of color images.

Then, to handle information from pixels without any pre-processing, the learning algorithm should be able to deal efficiently with a large amount of data, first in terms of the number of images and classes of images in the learning set, but more importantly in terms of the number of values describing these images (i.e. the attributes). Assuming for example that $W_x = W_y = 128$, there are already $128 * 128 = 16384$ integer values describing images and this number is further multiplied by 3 if colors are taken into account.

We describe below seven classification algorithms that we think could work in such difficult conditions and that we have compared in our experiments. The common characteristics of these methods is that they are essentially non parametric and that they can efficiently handle very large input spaces.

2.1 Decision trees

Decision tree induction [3] is one of the most popular learning algorithms with nice characteristics of interpretability, efficiency, and flexibility. However, the accuracy of this algorithm is often not competitive with other learning methods due to its high variance [9]. In this study, we therefore do not expect decision trees to be satisfactory but we still evaluate it as it is the basis of other promising and recent algorithms.

2.2 Ensemble of decision trees

Ensemble methods are very popular in machine learning. These methods improve an existing learning algorithm by combining the predictions of several models obtained by perturbing either the learning set or the learning algorithm parameters. They are very effective in combination with decision trees that otherwise are often not competitive with other learning algorithms in terms of accuracy. Several ensemble methods have been applied to image classification problems, either in combination with traditional algorithms or with ad hoc computer vision system (e.g. in [7] and [12]). In this paper, we propose to compare four different ensemble methods based on decision trees. Two of these methods are the now famous bagging and boosting techniques. The two other methods, random forests and extra-trees, are two recent methods that essentially improve bagging in terms of accuracy but also in terms of computational efficiency. As the extra-trees method is our own proposal, we give below a more detailed description of it.

2.2.1 Bagging

Bagging [1] (for "boostrap aggregating") consists in drawing T boostrap learning samples from the original learning set (by random re-sampling without replacement) and then in producing from each of them a model using the classical decision tree algorithm. Then, a prediction is computed for a test instance by taking the majority class among the predictions given by the T trees for this instance.

2.2.2 Random forests

With Random Forests [2], each of the T trees is grown on a bootstrap sample of the original learning set like in bagging. But here, during tree construction, at each node of the tree, only a small number (k) of attributes randomly selected among the whole set of attributes is searched for the best test. In [2], it has been shown that random forests give better results than bagging and often yield results competitive with boosting (see below). It is also faster than these two algorithms since it requires to consider only a small subset of all attributes when developping one node of the tree.

2.2.3 Extremely randomized trees

Extremely randomized trees [9], [10] (extra-trees in short) are another ensemble method for decision trees that is extreme in terms of the randomization introduced when growing the trees of the ensemble. Indeed, an extremely randomized tree is grown by selecting at each node of the tree the parameters of the test fully at random. In the context of image classification, this yields the very simple recursive function shown in Table 1 to build an extra-tree. Several extra-trees are then built according to this algorithm and their predictions are aggregated just like in other ensemble methods. Experiments in [9] have shown that this method gives better results than bagging and is also competitive with boosting. Its main advantage with respect to other ensemble methods for decision trees is that it is also extremely fast. The complexity of the algorithm of Table 1 is independent of the number of attributes and, like other decision tree based algorithms, it is (empirically) linear with respect to the learning sample size.

2.2.4 Boosting

Boosting also builds an ensemble of decision trees but, contrary to previous ensemble methods, it produces the models sequentially and is a deterministic algorithm. It sequentially applies the learning algorithm to the original learning sample by increasing weights of misclassified instances. So, as the iteration proceeds, the models are forced to focus on the "difficult" instances. Several variants of this algorithm have been proposed in the literature. In our experiment, we will use the original algorithm, called AdaBoost.M1, described in [8] and we will apply it to decision trees.

2.3 Support Vector Machines

Support Vector Machine (SVM) is a machine learning algorithm originally motivated by advances in statistical learning theory [24]. It first applies a transformation of the initial input space into a new potentially very high dimensional transformed input space where classes are very likely to be linearly separable and then deriving a hyperplane to separate each pair of classes in this transformed input space. There exist several efficient implementations of this algo-

Build_extra_tree(input: a learning sample, LS):

- If LS contains images all of the same class, return a leaf with this class associated to it;
- Otherwise:
  1. Set $[a_{k,l} < a_{th}]$=Choose_a_random_split(LS);
  2. Split LS into LS_{left} and LS_{right} according to the test $[a_{k,l} < a_{th}]$ and build the subtrees $\mathcal{T}_{left}$ = build_extra_tree(LS_{left}) and $\mathcal{T}_{right}$ = build_extra_tree(LS_{left}) from these subsets;
  3. Create a node with the test $[a_{k,l} < a_{th}]$, attach $\mathcal{T}_{left}$ and $\mathcal{T}_{right}$ as successors of this node and return the resulting tree.

Choose_a_random_split(LS):

1. Select a pixel location (k, l) at random;
2. Select a threshold a_{th} at random according to a distribution $N(\mu_{k,l}, \sigma_{k,l})$, where $\mu_{k,l}$ and $\sigma_{k,l}$ are resp. the mean and standard deviation of the pixel values $a_{k,l}$ in LS;

Table 1: Extra-tree induction algorithm for image classification

rithm. In our experiment, we use the algorithm proposed in [5] with Gaussian and polynomial kernels. SVMs already gave very impressive results in terms of accuracy and computational efficiency in many complex domains including image classification problems (e.g. in [4], [11] or [23]).

2.4 Extra-trees with sub-window extraction

Even though ensemble methods with decision trees can handle a very large number of input variables efficiently (especially our extra-trees), the tree complexity (and, hence, the number of pixels that are combined along a path from the root node to a leaf in the tree) is limited by the size of the learning set. When the number of images is small compared to the total number of pixels, a tree cannot combine enough pixels to provide acceptable models. To solve this problem, we have adopted another generic approach that is popular in image classification (e.g. [13] and [6]). It artificially augments the number of images in the learning set by building models from sub-windows extracted from original images of the learning set.

Although it can be combined with any learning algorithm, we have combined this idea with extra-trees, essentially for computational efficiency reasons. In this variant, the construction of one extra-tree from the ensemble is carried out in two steps, given a window size $w_1 \times w_2$ and a number N_w:

- Extract N_w sub-windows at random from training set images (by first selecting an image at random from LS and then selecting a sub-window at a random

Table 2: Database summary

DBs	# images	# attributes	# classes
MNIST	70000	784 $(28*28*1)$	10
ORL	400	10304 $(92*112*1)$	40
COIL-100	7200	3072 $(32*32*3)$	100
OUTEX	864	49152 $(128*128*3)$	54

location in this image) and assign to each sub-window the classification of its parent image;

- Grow an extra-tree to classify these N_w sub-windows by using the $w_1.w_2$ pixel values that characterize them.

To make a prediction for an image with an ensemble of extra-trees grown from sub-windows, the following procedure is used:

- Extract all possible sub-windows of size $w_1 \times w_2$ from this image;
- Assign to the image the majority class among the classes assigned to the sub-windows by the ensemble of extra-trees.

3 Four image classification problems

To evaluate the machine learning algorithms for generic image classification and to allow replication of our experiments, we selected four publicly available datasets corresponding to common image classification problems: recognition of handwritten characters (here, digits), faces, objects, and textures. The main characteristics of the datasets are summarized in Table 2 and an overview of their images is given in Figure 1. We briefly describe each problem below.

3.1 MNIST, database of handwritten digits

The MNIST database[1] [16] consists of 70000 handwritten digits that have been size-normalized and centered in images of 28×28 pixels with 256 grey levels per pixel. The goal is to build a model that classifies digits. Different writing styles are characterized by thin or thick strokes, slanted characters, etc.

3.2 ORL, face database

The ORL database[2] from AT&T contains faces of 40 distinct persons with 10 images per person that differ in lighting, facial expressions (open/closed eyes, smiling/not smiling), facial details (glasses/no glasses) and contain minor variations in pose. The size of each image is 92×112 pixels, with 256 grey levels per pixel. The goal is to identify faces.

[1] http://yann.lecun.com/exdb/mnist/
[2] http://www.uk.research.att.com/facedatabase.html

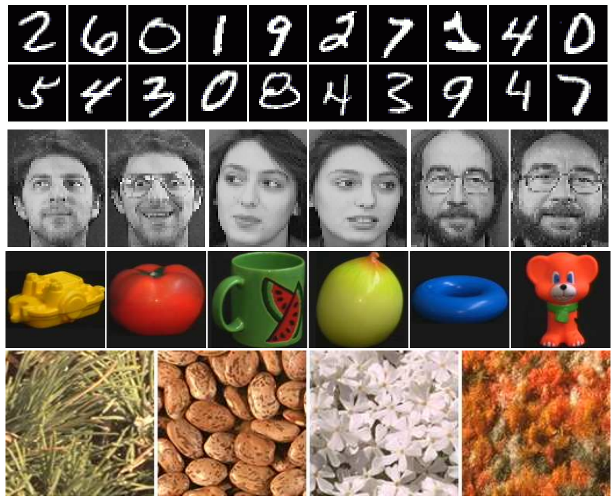


Figure 1: Overview of the four databases: MNIST, ORL, COIL-100, OUTEX

3.3 COIL-100, 3D object database

The Columbia University Object Image Library[3] COIL-100 is a dataset with colored images of 100 different objects (boxes, bottles, cups, miniature cars, etc.). Each object was placed on a motorized turntable and images were captured by a fixed camera at pose intervals of 5 degrees. This corresponds to 72 images per object. In COIL-100, each image has been normalized to 128×128 pixels and are in true color. For our experiments, we have resized the original images down to 32×32 pixels.

3.4 OUTEX, texture database

Outex[4] [21] provides a framework for the empirical evaluation of texture analysis algorithms. The Contrib_TC_0006 dataset we took from Outex has been derived from the VisTeX dataset. It contains 54 colored textures and 16 images of 128×128 pixels in true colors for each VisTex texture.

4 Experiments

In this section, the seven classification algorithms are compared on the four problems. Before describing the experimentation protocol and discussing the results, the next subsection discusses some implementation details and the way we have tuned the different parameters of the methods.

[3]http://www.cs.columbia.edu/CAVE/
[4]http://www.outex.oulu.fi/

4.1 Implementation and determination of the parameters

For all algorithms except for SVM, we have used our own software which is implemented in C. For SVM, we have used the LibSVM [5] package which is a C++ implementation of the algorithm presented in [5].

For each machine learning method, the values of several parameters need to be fixed. We discuss this tuning stage for each method below. For some algorithms, the values of the parameters are fixed on the basis of the resulting error on the test sample. We are aware that this would lead to slightly underestimated error rates, however, we believe that this will not be detrimental for comparison purposes, especially since the number of parameters in each method is quite small.

Classical decision trees are fully developed, i.e. without using any pruning method. The score measure used to evaluate tests during the induction is the score measure proposed in [25] which is a particuler normalization of the information gain. Otherwise our algorithm is similar to the CART method [3].

Ensemble methods all are influenced by the number of trees T which are aggregated. Usually, the more trees are aggregated, the better the accuracy. So, in our study, we have used for each problem and for each algorithm, a number of trees that appeared to be large enough to give stable error rates on the test samples. Usually, extra-trees that are more randomized requires more trees than the other variants (from 50 on MNIST to 500 on ORL and COIL-100). Extra-trees with sub-windows however are stabilized much sooner (10 trees are sufficient in all problems).

Random forests depends on an additional parameter k which is the number of attributes randomly selected at each test node. In our experiments, its value was fixed to the default value suggested by the author of the algorithm which is the square root of the total number of attributes. According to [2] this value usually gives error rates very close to the optimum.

Boosting does not depend on another parameter but it nevertheless requires that the learning algorithm does not give perfect models on the learning sample (so as to provide some misclassified instances). Hence, with this method, we used with decision trees the stop-splitting criterion described in [25]. It uses an hypothesis testing based on the G^2 statistic [14] to determine the significance of a test. In our experiments, we fixed the nondetection risk α to 0,005.

For SVM, we used LibSVM with default parameters. We tried their implementation of linear, polynomial (with degree 2 and 3) and radial basis kernel functions. Again, the best kernel was choosen on the test sample.

For extra-trees on sub-windows, additional parameters are the size of sub-windows $w1 \times w2$ and the number N_w of them extracted during the learning phase. Like for the number of trees in the ensemble, accuracy appears to be a monotonically increasing function of N_w. On the last three problems, N_w was fixed to 120000. On MNIST, as the initial learning sample size is already quite large, we further increase this number to 360000. Accuracy is on the

[5] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Table 3: Results on all problems

MNIST

Algorithm	Error rate
Classical Decision Tree	11.5%
Bagging ($T = 50$)	4.42%
Extra-Trees ($T = 100$)	3.17%
Random Forests ($T = 100$)	3.0%
Extra-Trees + Sub-Window	2.54%
Boosting ($T = 50$)	2.29%
SVMs (poly2)	1.95%
LeNet-4[16]	0.7%

ORL

Algorithm	Error rate
Classical Decision Tree	29.25% $\pm$ 6.89
Bagging ($T = 50$)	9.5% $\pm$ 5.7
Boosting ($T = 50$)	3.75% $\pm$ 2.79
Random Forests ($T = 200$)	1.25% $\pm$ 1.68
Extra-Trees ($T = 500$)	1.25% $\pm$ 1.68
SVMs (linear)	1.25% $\pm$ 1.25
Extra-Trees + Sub-Window	0.5% $\pm$ 1.0
-	-

COIL-100

Algorithm	Error rate
Classical Decision Tree	20.80%
Bagging ($T = 50$)	2.24%
Extra-Trees ($T = 500$)	1.96%
Random Forests ($T = 500$)	1.17%
Boosting ($T = 100$)	0.54%
SVMs (linear)	0.44%
Extra-Trees + Sub-Window	0.35%
Local Affine Frames [20]	0.1%

OUTEX

Algorithm	Error rate
Classical Decision Tree	89.35%
Bagging ($T = 50$)	73.15%
SVMs (linear)	71.99%
Boosting ($T = 50$)	69.44%
Random Forests ($T = 1000$)	66.90%
Extra-Trees ($T = 1000$)	65.05%
Extra-Trees + Sub-Window	2.78%
RGB Histograms [18]	0.2%

other hand very much influenced by the size of sub-windows. The optimal size is problem-dependent and has been tuned manually on each problem.

4.2 Protocols and results

The test protocols are discussed in this section on each problem. Results are summarized in Table 3. For each problem, the methods are sorted by decreasing error rates.

4.2.1 MNIST

In the literature, the first 60000 images are often used for learning and the remaining 10000 examples are used for validation. In [16], the results of many learning methods are reported, they range from 12% with a one-layer neural network to 0.7% with the authors' method "Boosted LeNet-4". Our results are obtained by strictly following this protocol. Error rates with generic methods vary from 11.5% with a classical decision tree to 1.95% with support vector machines. Using sub-windows, the error rate is 2.54% (with $T = 10$, $N_w = 360000$ and $w_1 = w_2 = 24$) which is less accurate than Boosting and SVM. In [4], an error rate of 1.1% was obtained with another implementation of SVM.

4.2.2 ORL

In the literature, various algorithms with pre-processing steps have been tested on this dataset, including hidden Markov models [19], convolutional neural networks [15], SVMs [11], and variants of nearest neighbors [22]. But the protocol for testing is different from one paper to another. Given the fact that this database is quite small and as there is no well-defined test protocol, our experiments use 10-fold cross-validation to provide a fair assessment of the generic methods. It means that the learning set was randomly partitioned into 10 learning samples with 360 images (9 views per subject) while the remaining images (1 view per subject) used for tests samples. Following this procedure, we get an average error rate (10 runs) of 29.25%, with a classical decision tree, down to 0.5% with our sub-windows (with $T = 10$ and $w_1 = w_2 = 32$). Random Forests, Extra-Trees, and SVM give a slightly inferior result with an average error rate of 1.25%.

4.2.3 COIL-100

This problem was approached in the literature with different methods, some of them specific to 3D object recognition, that use different matching techniques of local or global features (color histograms, eigenwindows, locale affine frames [20], etc.). For the learning sample, we took 18 views for each of the 100 objects, starting with the pose at 0 and then going on with intervals of 20. The remaining views were devoted to the test sample. Methods in the computer vision literature provide error rates from 12.5% to 0.1%. Using this protocol, classical decision tree yields an error rate of 20.80% that drops down to 0.35% with the combination of extra-trees and sub-windows (with $T = 10$ and $w1 = w2 = 16$). Boosting and SVMs are close to our approach with respectively an 0.54% and 0.44% error rate.

4.2.4 OUTEX

This dataset has a small number of objects but a very large number of attributes. The OUTEX framework precisely defines the images to use in the learning and test sample (8 images for each texture in both ensembles). The paper [18] evaluates several feature extraction techniques and image transformation methods in combination with a traditional nearest neighbors algorithm. Their resulting error rates on this dataset vary from 9.5% to 0.2%. Using the same protocol, most of the popular learning methods are especially bad with an error rate varying from 89.35% (classical decision tree) to 66.90% (random forests). Extra-trees are also not satisfactory with an error rate of 65.05% (even with $T = 1000$). On the other hand, sub-windows reduce error rates down to 2.78% (with $T = 10$ and $w1 = w2 = 4$). These results can be explained by the nature of the problem. As textures are generally based on the repetition of small patterns, one can extract small sub-windows from the original images that are quite well classified by models since they contain sufficient information to classify the whole image. This statement is further confirmed by the fact that small sub-windows of size 4×4 give the best results on this problem. On the other hand, extra-trees alone and other learning algorithms are not able to find relevant information among the high number of pixels describing the textures because the characteristics patterns are not associated with specific image locations.

4.3 Discussion

As expected, for each problem, classical decision trees are not satisfactory. This is explained by the high variance of this method. However, ensemble methods are well-known in the literature for improving accuracy and this is confirmed by our experiments. Indeed, Bagging and in a more impressive way Boosting give much more accurate results for each problem. Random forests and Extra-Trees are competitive with Boosting. SVM is very close to boosting but maybe slightly better in average. Extra-trees with sub-windows is the best generic method in terms of accuracy on three of the four problems (ORL, COIL-100, OUTEX) but it is nevertheless beaten by boosting and SVM on MNIST. On all problems, the best results, always obtained either by SVM or extra-trees with sub-windows, are competitive with state-of-the-art techniques in computer vision. This is remarkable considering that these algorithms are very easy to use.

Another criterion for comparing algorithms for computer vision is their computational efficiency. To give an idea of the difference between the algorithms, Table 4 reports the computing times [6] for learning each model on the COIL-100 and MNIST problems. Extra-trees is undoubtedly the fastest method. On COIL-100, growing 500 extra-trees is even faster than growing one single decision tree. Not surprisingly, bagging and boosting that build T trees with the classical decision tree induction algorithm are the slowest methods. Random forests are much faster but still slower than the extra-trees (especially on COIL-100). SVMs are very fast on COIL-100 but slower on MNIST which consists of much more images. Our sub-windows method increases significantly the computing times of extra-trees but the resulting algorithm is still much faster than bagging and boosting on COIL-100. On MNIST, high computing times are attributed to the augmented learning sample ($N_w = 360000$).

The prediction times for all tree-based algorithms are negligible. With sub-windows however, the test of a new image requires the propagation of all sub-windows in the trees of the ensemble and it depends mostly on the size of the sub-windows. For example, it takes about 54s to predict the classes of the 5400 test objects on COIL-100 and about 12s to test the 10000 images on MNIST. Although our implementation is not optimal, we believe that these times are nevertheless reasonable. With SVM, prediction times depends on the number of support vectors. On both problems, it was the slowest method for testing with a prediction time of 3m19s for COIL-100 and 8m09s on MNIST.

5 Conclusions

We compared seven generic algorithms for image classification including our recent approach that combines building of ensembles of extremely randomized trees and extraction of sub-windows from the original images. Classical decision tree, Bagging, Boosting, Random Forests and SVM are the other meth-

[6] On a Pentium IV 2.53Ghz.

Table 4: Learning time on MNIST and COIL-100

MNIST

Algorithm	Time
Boosting ($T = 50$)	6h22m29s
Extra-Trees + Sub-Window	5h06m24s
Bagging ($T = 50$)	5h01m11s
SVMs (poly2)	28m28s
Random Forests ($T = 100$)	20m16s
Extra-Trees ($T = 100$)	11m39s
Classical Decision Tree	7m17s

COIL-100

Algorithm	Time
Boosting ($T = 100$)	5h25m01s
Bagging ($T = 50$)	1h53m25s
Random Forests ($T = 500$)	51m34s
Extra-Trees + Sub-Window	45m45s
Classical Decision Tree	3m08s
SVMs (linear)	1m02s
Extra-Trees ($T = 500$)	9s

ods we evaluated on four different image classification problems for which test protocols were rigorously specified. The accuracy of our generic sub-window technique is the best on three of the four problems and is comparable to state-of-the-art techniques but slightly inferior to the best known results. In fact, our experiments demonstrate that generic methods, in particular our sub-window algorithm, can come remarkably close to specialized methods. In many practical application contexts, a slight performance drop in exchange for reduced task-specific pre-processing and manual intervention may constitute a very desirable trade-off.

The future directions are two-fold. First, as our wrapping method of extraction and classification of sub-windows from images is generic, it is attractive to combine it with other learning algorithms (in particular Boosting and SVMs). Second, experiments should be carried out to compare the robustness of these generic approaches to rotation, scaling, occlusion, and noise. Although some algorithms are close in terms of accuracy, it is not sure that they would be all affected in the same way by these perturbations. In [17], we provided a preliminary study of the behavior of our sub-window approach in the presence of rotation, scaling, and occlusion on the COIL-100 problem. We observed good robustness to small transformations introduced in test images and also suggested some improvements that preserve the generic nature of the algorithm.

6 Acknowledgments

Raphaël Marée is supported by the “Région Wallonne” (Belgium). Pierre Geurts is a research associate of the F.N.R.S., Belgium.

References

[1] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[2] L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[3] L. Breiman, J.H. Friedman, R.A. Olsen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth International (California), 1984.

[4] C. J. C. Burges and B. Schölkopf. Improving the accuracy and speed of support vector machines. In M. C. Mozer, M. I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, page 375. The MIT Press, 1997.

[5] Chih-Chung Chang and Chih-Jen Lin. Libsvm : a library for support vector machines. Technical report, Computer Science and Information Engineering, National Taiwan University, 2003.

[6] J. Dahmen, D. Keysers, and H. Ney. Combined classification of handwritten digits using the 'virtual test sample method'. In *Proc. Second International Workshop, MCS 2001 Cambridge, UK*, pages 99–108, July 2001.

[7] H. Drucker. Fast decision tree ensembles for optical character recognition. In *Proc. Fifth Annual Symposium on Document Analysis and Information Retrieval*, pages 137–147, 1996.

[8] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proc. Thirteenth International Conference on Machine Learning*, pages 148–156, 1996.

[9] P. Geurts. *Contributions to decision tree induction: bias/variance tradeoff and time series classification.* Phd. thesis, Department of Electrical Engineering and Computer Science, University of Liège, May 2002.

[10] P. Geurts. Extremely randomized trees. Technical report, Department of Electrical Engineering and Computer Science, University of Liège, 2003.

[11] G.-D. Guo, S. Li, and K. Chan. Face recognition by support vector machines. In *Proc. International Conference on Automatic Face and Gesture Recognition, 196-201.*, 2000.

[12] G.-D. Guo and H.-J. Zhang. Boosting for fast face recognition. In *Proc. IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 96–100, 2001.

[13] M.S. Hoque and M. C. Fairhurst. A moving window classifier for off-line character recognition. In *Proc. of the Seventh International Workshop on Frontiers in Handwriting Recognition, Amsterdam*, pages 595–600, September 2000.

[14] T.O. Kvålseth. Entropy and correlation : Some comments. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-17(3):517–519, 1987.

[15] S. Lawrence, C. Lee Giles, A. C. Tsoi, and A. D. Back. Face recognition: A convolutional neural network approach. *IEEE Transactions on Neural Networks*, 8(1):98–113, 1997.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, 1998.

[17] R. Marée, Geurts P., Piater J., and Wehenkel L. A generic approach for image classification based on decision tree ensembles. Submitted.

[18] T. Menp, M. Pietikinen, and J. Viertola. Separating color and pattern information for color texture discrimination. In *Proc. 16th International Conference on Pattern Recognition*, 2002.

[19] A. Nefian and M. Hayes. Face recognition using an embedded HMM. In *Proc. IEEE Conference on Audio and Video-based Biometric Person Authentication*, pages 19–24, March 1999.

[20] S. Obrzalek and J. Matas. Object recognition using local affine frames on distinguished regions. In *Electronic Proceedings of the 13th British Machine Vision Conference, University of Cardiff*, 2002.

[21] T. Ojala, T. Menp, M. Pietikinen, J. Viertola, J. Kyllnen, and S. Huovinen. Outex - new framework for empirical evaluation of texture analysis algorithms. *Proc. 16th International Conference on Pattern Recognition, Quebec, Canada, 1:701-706*, 2002.

[22] R. Paredes and A. Perez-Cortes. Local representations and a direct voting scheme for face recognition. In *Pattern Recognition in Information Systems, Proc. 1st International Workshop on Pattern Recognition in Information Systems*, pages 71–79, July 2001.

[23] M. Pontil and A. Verri. Support vector machines for 3d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):637–646, 1998.

[24] V.N. Vapnik. *The nature of statistical learning theory.* Springer Verlag, 1995.

[25] L. Wehenkel. *Automatic learning techniques in power systems.* Kluwer Academic, Boston, 1998.

Journal of Theoretical and Applied Information Technology
10th January 2015. Vol.71 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

COMPARISON OF IMAGE CLASSIFICATION TECHNIQUES USING CALTECH 101 DATASET

[1]NUR SHAZWANI KAMARUDIN, [2]MOKHAIRI MAKHTAR, , [3]SYED ABDULLAH FADZLI, [4]MUMTAZIMAH MOHAMAD, [5]FATMA SUSILAWATI MOHAMAD, [6]MOHD FADZIL ABDUL KADIR

[1]Research Scholar, Faculty of Informatics and Computing, University Sultan Zainal Abidin, Tembila Campus, Terengganu, Malaysia.

[2,3,4,5,6]Senior Lecturer, Faculty of Informatics and Computing, University Sultan Zainal Abidin, Tembila Campus, Terengganu, Malaysia.

Emel: [1]wanikamarudin@gmail.com, [2]mokhairi@unisza.edu.my, [3]fadzlihasan@unisza.edu.my, [4]mumtaz@unisza.edu.my, [5]fatma@unisza.edu.my, [6]fadzil@unisza.edu.my

ABSTRACT

This paper presents the technique for the classification of single object images. First, this paper aims to introduce the efficient technique in order to classify single object image. Second, each single methods uses in order to propose the techniques were elaborated in this paper. It start from image segmentation, object identification, feature extraction, feature selection and classification. Finally, the best classifier that can provide the best results were identified. The efficiency of the proposed method is define by comparing the result of classification using two different datasets from author's previous paper. The obligation for development of image classification has been improved due to remarkable growth in volume of images, as well as the widespread application in multiple fields. This paper explores the process of classifying images by the categories of object in the case of a large number of object categories. The use of a set of features to describe 2D shapes in low-level images has been proposed. The proposed technique aims a short and simple way to extract shape description before classifying the image. Using the Caltech 101 object recognition benchmark, classification was tested using four different classifiers; BayesNet, NaiveBayesUpdateable, Random Tree and IBk. Estimated accuracy was in the range from 58% to 99% (using 10-cross validation). By comparing with Amazon data, it is proved that the proposed model is more suitable for single object image. Amazon images give higher accuracy with the range from 80% to 99.48%.

Key words: *Image Classification, Feature Extraction, Feature Selection, Classifier.*

1. INTRODUCTION

One of the crucial tasks among computer vision field is an object classification. Image classification is the process of labelling the images into one of a number of predefined categories. The steps of classification include image sensors, image pre-processing, object detection, object segmentation, feature extraction, and object classification. There are numbers of classification techniques that have been developed for image classification [1].

Image classification is a crucial and challenging task in various application domains, including remote sensing, vehicle navigation, biomedical imaging, video-surveillance, biometry, industrial visual inspection, robot navigation, and vehicle navigation [2]. Classification process consists of the following steps [1]:

A. **Pre-processing:** Enhances the quality of the input image such as noise removal, image masking, main component analysis, and others, and locate the data of interest.
B. **Detection and extraction of an object:** Detection contains detection of position and other features of moving object image found from camera. In addition, extraction stage captures the unique characteristics from the detected object guessing the route of the object in the image plane.

Journal of Theoretical and Applied Information Technology
10th January 2015. Vol.71 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

C. **Training:** Selection of the specific attribute which best defines the pattern.
D. **Classification of the object:** Object classification step classifies detected objects into predefined classes by using proper method that matches the image patterns with the target patterns.

From author's previous article [3], they proposed a classification model. The model consists of five main stages, starting from image segmentation, object identification, feature extraction, feature selection and finally, image classification. The model was tested using Amazon images. To make a further comparison, Caltech101 dataset were chosen for this paper.

Shape is one of the objects representation in images with the most significant properties, which is famously used in CBIR (content based image retrieval) and in recognition tasks [4].

The Caltech101 object class dataset consists of 101 class of images. It can be downloaded from [5]. With a wide variety of images in each class, this dataset provides a significant intraclass variant [6]. [7] has mentioned that the name of 101 was accidentally set while the author was flipping through pages of Webster Collegiate Dictionary, and they came out with the idea of listing all 101 categories of images. All images in Caltech101 dataset were downloaded using Google Image search engine.

This paper presents the classification of single object in an image using 4 different classifiers. First, images were segmented for partitioning the meaningful part of the image. Second, object identification was applied to the segmented image to detect the connected line in the image. Then, feature extraction of the object was conducted for encoding the valuable features. Next, the experiments were continued with feature selection using Weka tools. Lastly, the classification accuracies of images using 4 types of classifiers selected earlier were presented.

The focus of this paper is to test the proposed techniques toward Caltech101 dataset and compare with Amazon dataset's results. The results of experiments are presented in the paper, and conclusions are drawn.

2. RESEARCH METHOD AND DESIGN

One of the popular research area throughout current years is content-based image classification using large image databases [8].

The proposed technique aims a short and simple way to extract shape description followed by the classification and annotation processes. The proposed method followed these steps:

Step 1: Image Segmentation

Step 2: Object Identification

Step 3: Image Feature Extraction

Step 4: Image Feature Selection

Step 5: Classification

The process of partitioning images into meaningful region is known as ad image segmentation in the computer vision field [9].

The feature is a measurement process which specifically defines a property of an object based on the characteristics of the object. Shape is one of the crucial visual features because of its primitive feature. Shape can be described in two ways; region-based and contour-based. The method that uses the whole area of an object for shape description is region-based. On the other hand, contour-based method uses the information given from the contour of an image [10]. Normally, shape descriptors are combined to create a more effective shape descriptor because individual simple shape descriptor is not robust [11].

The most important step after feature extraction is feature selection. It plays an important role, especially in classification problems. A well-extracted feature must have the value of robustness, discriminative, and easy to compute an efficient algorithm.

The main issue in object detection is to locate the object in an arbitrary image and pose after a non-rigid transformation.

A classifier is chosen after features of an image have been extracted. There are plenty of classifiers that can be used for the experiment.

As stated in Fig.1, Phase 1 is named as data repository, meaning that the process of collecting data takes place before other phases are

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

initiated. In this paper, Caltech101 was used for collecting dataset. The collection process showed that Caltech101 was the most suitable dataset that can be used to test the proposed techniques.

The second phase is the training images, which involved the rest of the processes (from segmentation to classification process) to build the training set.

Initially, in Phase 2, which was named as the training phase, the dataset (images) underwent the image pre-processing steps. The purpose of this measure is to substitute the high-dimensional images with lower-dimensional features that capture the main properties of the images and enable the model to forge on the data with limited storage and computational resources. It includes three main processes starting from image segmentation, followed by feature extraction, and end up with image classification. For the purpose of the next phase of the research, all extracted features were stored in the features database. A feature selection step was added before the classification step for comparison throughout the experiments.

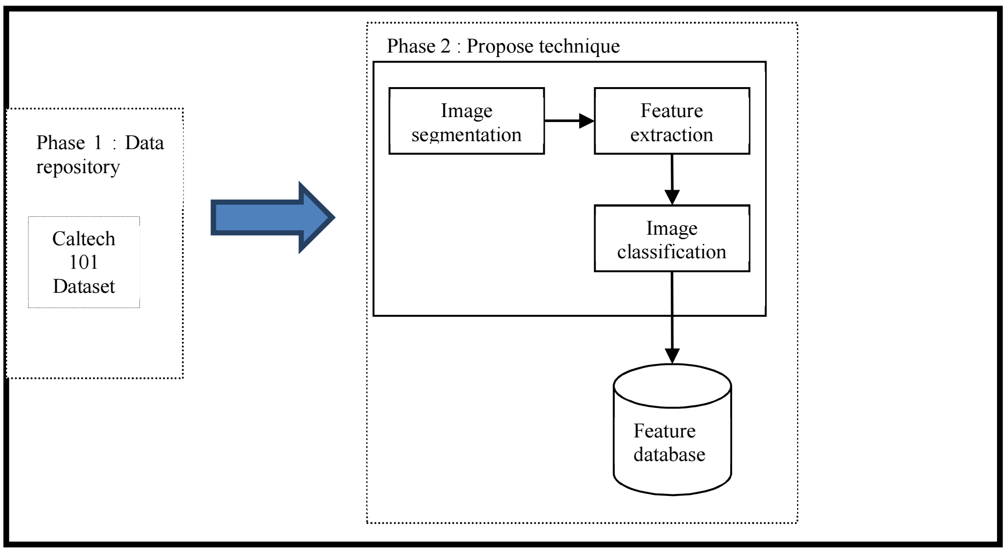


Figure 1: Research method

Journal of Theoretical and Applied Information Technology
10th January 2015. Vol.71 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. SEGMENT IMAGES

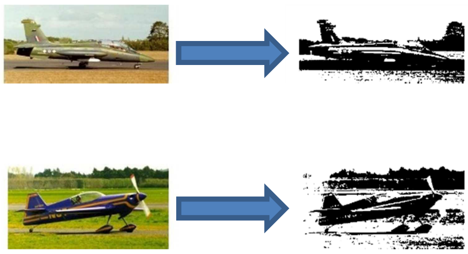


Figure 2: Sample of image classification results using Otsu method

By applying a large dataset from Caltech101, the image data were segmented using thresholding based method. This paper used one of the global thresholding methods, which is Otsu method. [12]. This technique thresholds the entire image with a single threshold value [12], and this system is dependent upon discriminant analysis.

As discussed by [13][14][15], threshold operation can divide the image into two classes; A_1 and A_2, at gray Q such that A_1 = [22 ... , Q] and A_2 = {Q + 1, Q + 2, ... , k-1}, where k is the total number of the gray levels of the image. Let the number of pixels at **i** gray level be n_i, and $N = \sum_{i=0}^{k-1} n_i$ be the total number of pixels in a given image. The probability of occurrence of gray level **i** is defined as $p_i = \frac{n_i}{N}$, $p_i \geq 0$, $\sum_{i=0}^{k-1} p_i = 1$. A_1 and A_2 are normally corresponding to the object of interest and the background. The probabilities of the two classes are:

$P_{A1} = \sum_{i=0}^{Q} p_i$ and $P_{A2} = \sum p_i = 1 - P_{A1}$.

The mean of A_1 and A_2 classes can be computed as:

$$\mu_{A1} = \sum_{i=0}^{Q} \frac{i^* p_i}{P_{A1}} \qquad (1)$$

$$\mu_{A1} = \sum_{i=Q+1}^{k-1} \frac{i^* p_i}{P_{A2}} \qquad (2)$$

Thus, we can get the equivalent formula:

$$\sigma^2(Q) = P_{A1} P_{A2} (\mu_{A1} - \mu_{A2})^2 \qquad (3)$$

The optimal threshold Q* can be obtained by maximizing the between-class variance.

$$Q^* = Arg \max_{0<Q<k-1} \sigma^2 (Q) \qquad (4)$$

4. FEATURE EXTRACTION

MATLAB was chosen as a tool to develop the extraction procedure using built-in Image Processing Toolbox function known as region props. The process of attaining image features from an input image was initiated with the image properties calculation such as area, eccentricity, extent, solidity, filled area, and others. The features were figured using built-in principles in MATLAB.

5. CLASSIFIER

The classifiers that were applied in our research include BayesNet, NaieveBayesUpdateable, Random Tree and IBk.

5.1 weka.classifiers.bayes.BayesNet

Bayesian networks (BNs), also known as belief net-works (or Bayes nets for short), belong to the family of probabilistic graphical models (GMs).

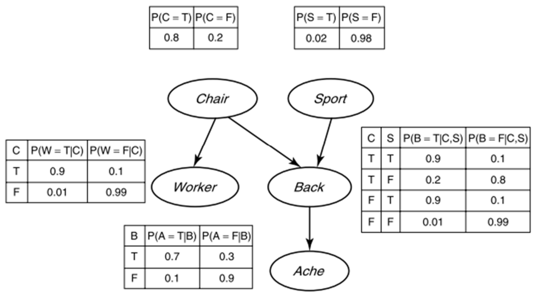


Figure 3: The backache BN example

Figure 3 shows the example of BayesNet process. The parents of the variable *Back* are the nodes *Chair* and *Sport*. The child of *Back* is *Ache*, and the parent of *Worker* is *Chair*. Following the BN independence hypothesis, some independence statements can be detected in this case. For example, the variables *Chair* and *Sport* are slightly independent, but when *Back* is given they are conditionally dependent. This relation is often called explaining away. When *Chair* is given, *Worker* and *Back* are conditionally independent. When *Back* is given, *Ache* is conditionally independent of its ancestors *Chair* and *Sport* [16].

5.2 Weka.Classifiers.Bayes.Naivebayesupdateable

A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive

Journal of Theoretical and Applied Information Technology
10th January 2015. Vol.71 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

term for the underlying probability model would be "independent feature model".

The Class is used for a Naive Bayes classifier using estimator classes. This is the updateable version of Naïve Bayes. This classifier will use a default precision of 0.1 for numeric attributes when build Classifier is called with zero training instances [17].

5.3 weka.classifiers.tree.RandomTree

The decision tree (DT) is a multi-stage decision making or classification tool. It is different to other classification model because it uses input-output relationship that can be expressed using human understandable rules. Meanwhile, other classification model is much difficult to describe [11].

A random tree is a tree drawn at random from a set of possible trees. In this context "at random" means that each tree in the set of trees has an equal chance of being sampled. Another way of saying this is that the distribution of trees is "uniform".

5.4 weka.classifiers.lazy.IBk

IBk is called instance-based learning that generates classification predictions using only specific instances. Instance-based learning algorithms do not maintain a set of abstractions derived from specific instances. This approach extends the nearest neighbor algorithm, which has large storage requirements [17].

6. IMAGE CLASSIFICATION (10-Fold Cross Validation)

The cross validation technique works as repeated holdout. It divides dataset into 10 parts (fold) by holding out each part in turn. Then, it will compute the average results where each data point was used once for testing and 9 times for training.

7. DATASET

The experiment used Caltech101 dataset as mentioned earlier. Originally, the dataset were collected from Google Image search engine to gather as many images as possible for each group [7]. Fig.3 shows examples of the 101 object categories used in this paper. In addition, the model parts were ordered by their x-coordinate, which was problematic for vertical structures. Therefore, the categories with mostly vertical structure were rotated to a random angle. It is for the sake of programming simplicity. Finally, the images were scaled roughly to around 300 pixels wide, producing Caltech101.

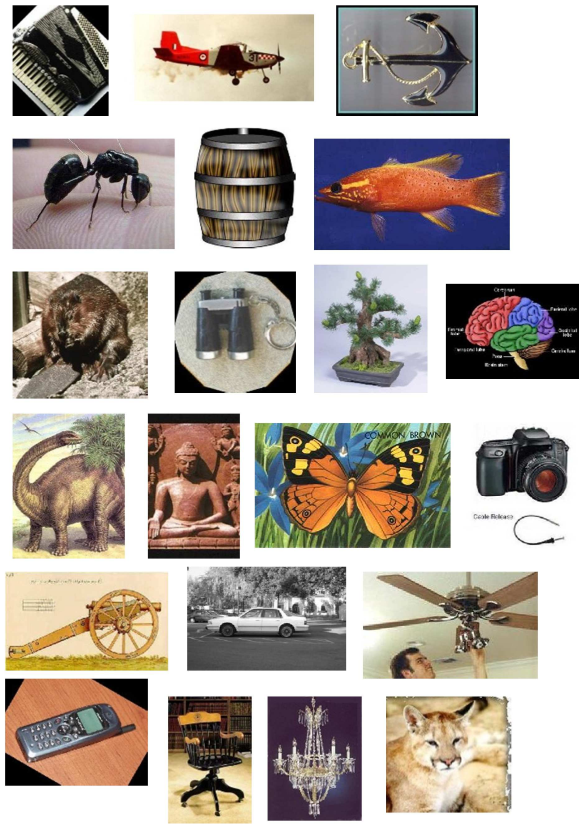


Figure 4: Caltech101 dataset

Figure 5 shows the sample images from Amazon dataset. The images also undergo same steps as Caltech101 dataset because finally the classification results will be compares.

Journal of Theoretical and Applied Information Technology
10th January 2015. Vol.71 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

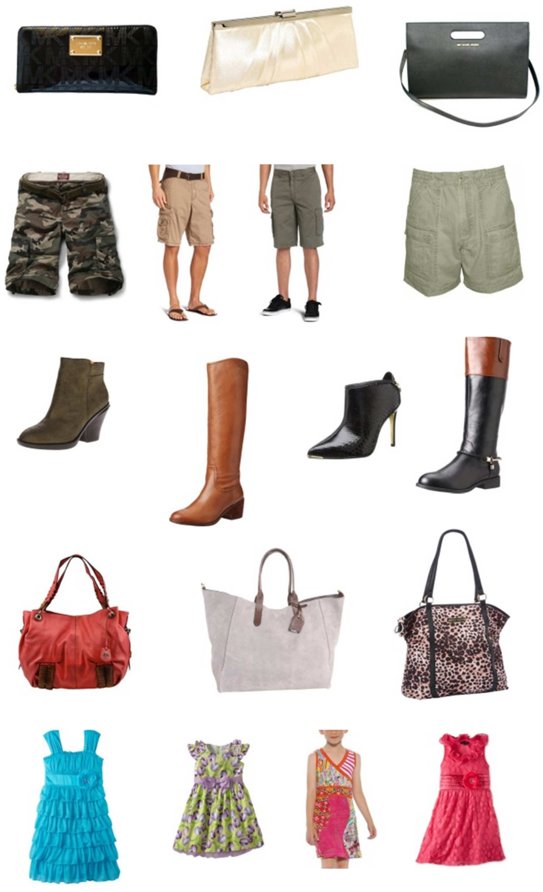


Figure 5: Amazon dataset

8. FEATURE SELECTION

The irrelevant input features may lead to overfitting. Feature selection focuses on the outstanding attributes over the dataset, which offers higher accuracy. There are lots of potential benefits of feature selection such as facilitating data visualization and data understanding, reducing utilization times and techniques, reducing storage requirements and measurement, and defying the curse of dimensionality to improve prediction performance.

CFS (Correlation-based Feature Selection) is an algorithm that couples the evaluation formula that is grounded on ideas from test theory, as well as provides an operational definition of this surmise with an appropriate correlation measure, and a heuristic search strategy.

The evaluator cannot work alone. There must be a search method in order to provide a good predictive power. There are three groups of variable subset selections; wrapper, filters, and embedded [18]. This paper used wrapper, which was provided in the Weka tool. Wrappers utilize the machine learning of interest as a black box to mark subsets of variable according to their predictive power. Its methodology offers a powerful and simple way to notify the problem of variable selection regardless of the chosen machine learning.

As already mentioned earlier, 11 features were combined in this experiment. After performing feature selection using CFSSubset evaluator, only 2 features were selected from the overall set of features. They were area and minor axis length. To make it fair, another evaluator was also used in this experiment. Principle components evaluator suggested seven features out of eleven. They were area, major axis length, minor axis length, eccentricity, orientation, convex area and filled area.

At this point, it was clear that 2 main features were suggested by both evaluators, which were area and minor axis length.

9. CLASSIFICATION RESULTS AND ANALYSIS

The main objective of image classification is to calculate the accuracy of classified images based on the categories stated. The tests were performed on the Caltech101 dataset that consist of 12 features for each single image. A series of experiments was conducted using all features, each with a different number of training images per category (only 30 categories were used). There are a total of 2,594 number of images used in this experiment from part of the 101 categories into training and test sets.

Table 1 lists the results of classification accuracy for images using four different classifiers. As stated in the table, the results show that random tree provides the highest accuracy with feature selection CFSSubsetEvaluator, with 99.00%.

Journal of Theoretical and Applied Information Technology
10th January 2015. Vol.71 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	BayesNet	NaiveBayesUpdateable	Random Tree	IBk
No Feature Selection	95.95	87.36	70.89	58.02
GreedyStepWise + CFSSubsetEvaluator	98.92	97.80	99.00	90.02
Ranker + PrincipleComponent	97.73	92.06	86.89	64.46

Table 1: Result Caltech101

Compare to result in Table 1, Amazon data which is single object image shows higher accuracy when applying CFSSUbsetEvaluator with 99.48%. It shows that the proposed model are more suitable for single object image because most of the accuracy shows in Table 2 is higher compare to Table 1.

	BayesNet	NaiveBayesUpdateable	Random Tree	IBk
No Feature Selection	80.33	90.32	93.81	92.23
GreedyStepWise + CFSSubsetEvaluator	98.12	94.72	99.48	95.20
Ranker + PrincipleComponent	91.53	93.02	98.18	93.99

Table 2: Result Amazon

After comparing both results, it clearly stated that the proposed model is more suitable for single object images. It is because, Caltech images is not a single image. As introduced in section 7, Caltech is an image with various colour of background. It cannot be said it is a single image. It is a multiple images. Compare to Amazon images, it is clearly a single object image with similar background.

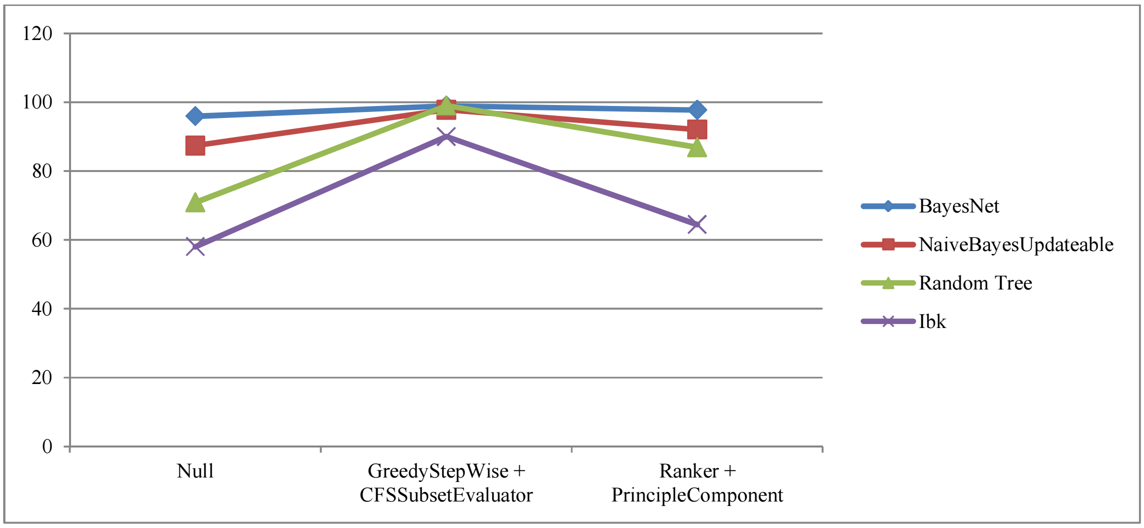


Figure 6:Graph for Table 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

10. CONCLUSION

We have presented in this paper new image classification technique based on shape for single object image. This paper presents a simple method based on a few set of image features to describe shapes. The contributions of the paper are the proposed technique which contain five steps starting from image segmentation, object identification, feature extraction, feature selection and image classification. Then, we proved that the technique can provide good results and it is can be said that compatible with the dataset. Experimental results using Caltech101 datasets show that the proposed technique achieve better image classification performance when using Random Tree classifier compare to other classifier.

The success of an image classification depends on several factors. In order to identify or define objects represented in images, shape is one of most valuable features.

11. ACKNOWLEDGEMENT

This work is partially supported by UniSZA (Grant No. UNISZA/13/GU (029).

REFERENCES:

[1] P. Kamavisdar, S. Saluja, and S. Agrawal, “A Survey on Image Classification Approaches and Techniques,” vol. 2, no. 1, pp. 1005–1009, 2013.

[2] D. L. & Q. Weng, “A survey of image classification methods and techniques for improving classification performance,” *Int. J. Remote Sens.*, 2007.

[3] Mokhairi Makhtar, Engku Fadzli, and Shazwani Kamarudin, “The Contribution of Feature Selection and Morphological Operation For On- Line Business System ’ s Image Classification,” *World Appl. Sci. J.*, 2014.

[4] J. F. Nunes, P. M. Moreira, and J. M. R. S. Tavares, “Shape Based Image Retrieval and Classification,” 2010, pp. 433–438.

[5] Fei-Fei, Fergus, and Perona, “Caltech 101 dataset,” 2004. [Online]. Available: http://www.vision.caltech.edu/feifeili/Datasets.htm.

[6] A. C. Berg, T. L. Berg, J. Malik, and U. C. Berkeley, “Shape Matching and Object Recognition using Low Distortion Correspondences,” 2005.

[7] L. Fei-Fei, R. Fergus, and P. Perona, “Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories,” *Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 178–178, 2004.

[8] S. Banerji, A. Sinha, and C. Liu, “New image descriptors based on color, texture, shape, and wavelets for object and scene image classification,” *Neurocomputing*, vol. 117, pp. 173–185, Oct. 2013.

[9] P. A. B. Miss Hetal J. Vala, “A Review on Otsu Image Segmentation Algorithm,” *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. Volume 2, no. Issue 2, 2013.

[10] R. S. Chora, “Image Feature Extraction Techniques and Their Applications for CBIR and Biometrics Systems,” *Int. J. Biol. Biomed. Eng.*, vol. 1, no. 1, 2007.

[11] M. M. Dengsheng Zhang n GuojunLu, “A review on automatic image annotation techniques,” *Pattern Recognit.*, pp. 346–362, 2012.

[12] P. . Sahoo, S. Soltani, and a. K. . Wong, “A survey of thresholding techniques,” *Comput. Vision, Graph. Image Process.*, vol. 41, no. 2, pp. 233–260, Feb. 1988.

[13] N. OTSU, “A Threshold Selection Method from Gray-Level Histograms,” *2EEE Trans. SYSTREMS, MAN, Cybern.*, vol. Vol 9, 1979.

[14] A. A. M. Al-kubati, J. A. M. Saif, and M. A. A. Taher, “Evaluation of Canny and Otsu Image Segmentation,” pp. 24–26, 2012.

[15] M. Huang, W. Yu, and D. Zhu, “An Improved Image Segmentation Algorithm Based on the Otsu Method,” *2012 13th ACIS Int. Conf. Softw. Eng. Artif. Intell. Netw. Parallel/Distributed Comput.*, pp. 135–139, Aug. 2012.

[16] Ruggeri F., Faltin F., and Kenett R., “Bayesian Networks,” *Encyclopedia of Statistics in Quality & Reliability*. Wiley & Sons, 2007.

[17] P.Santhi and V. Murali Bhaskaran, “Improving the Performance of Data Mining Algorithms in Health Care Data,” *IJCST*, vol. 2, no. 3, pp. 152–157, 2011.

[18] I. Guyon, “An Introduction to Variable and Feature Selection,” vol. 3, pp. 1157–1182, 2003.

Journal of Theoretical and Applied Information Technology
10th January 2015. Vol.71 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

COMPARISON OF IMAGE CLASSIFICATION TECHNIQUES USING CALTECH 101 DATASET

[1]NUR SHAZWANI KAMARUDIN, [2]MOKHAIRI MAKHTAR, , [3]SYED ABDULLAH FADZLI, [4]MUMTAZIMAH MOHAMAD, [5]FATMA SUSILAWATI MOHAMAD, [6]MOHD FADZIL ABDUL KADIR

[1]Research Scholar, Faculty of Informatics and Computing, University Sultan Zainal Abidin, Tembila Campus, Terengganu, Malaysia.

[2,3,4,5,6]Senior Lecturer, Faculty of Informatics and Computing, University Sultan Zainal Abidin, Tembila Campus, Terengganu, Malaysia.

Emel: [1]wanikamarudin@gmail.com, [2]mokhairi@unisza.edu.my, [3]fadzlihasan@unisza.edu.my, [4]mumtaz@unisza.edu.my, [5]fatma@unisza.edu.my, [6]fadzil@unisza.edu.my

ABSTRACT

This paper presents the technique for the classification of single object images. First, this paper aims to introduce the efficient technique in order to classify single object image. Second, each single methods uses in order to propose the techniques were elaborated in this paper. It start from image segmentation, object identification, feature extraction, feature selection and classification. Finally, the best classifier that can provide the best results were identified. The efficiency of the proposed method is define by comparing the result of classification using two different datasets from author's previous paper. The obligation for development of image classification has been improved due to remarkable growth in volume of images, as well as the widespread application in multiple fields. This paper explores the process of classifying images by the categories of object in the case of a large number of object categories. The use of a set of features to describe 2D shapes in low-level images has been proposed. The proposed technique aims a short and simple way to extract shape description before classifying the image. Using the Caltech 101 object recognition benchmark, classification was tested using four different classifiers; BayesNet, NaiveBayesUpdateable, Random Tree and IBk. Estimated accuracy was in the range from 58% to 99% (using 10-cross validation). By comparing with Amazon data, it is proved that the proposed model is more suitable for single object image. Amazon images give higher accuracy with the range from 80% to 99.48%.

Key words: *Image Classification, Feature Extraction, Feature Selection, Classifier.*

1. INTRODUCTION

One of the crucial tasks among computer vision field is an object classification. Image classification is the process of labelling the images into one of a number of predefined categories. The steps of classification include image sensors, image pre-processing, object detection, object segmentation, feature extraction, and object classification. There are numbers of classification techniques that have been developed for image classification [1].

Image classification is a crucial and challenging task in various application domains, including remote sensing, vehicle navigation, biomedical imaging, video-surveillance, biometry, industrial visual inspection, robot navigation, and vehicle navigation [2]. Classification process consists of the following steps [1]:

A. **Pre-processing:** Enhances the quality of the input image such as noise removal, image masking, main component analysis, and others, and locate the data of interest.
B. **Detection and extraction of an object:** Detection contains detection of position and other features of moving object image found from camera. In addition, extraction stage captures the unique characteristics from the detected object guessing the route of the object in the image plane.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

C. **Training:** Selection of the specific attribute which best defines the pattern.
D. **Classification of the object:** Object classification step classifies detected objects into predefined classes by using proper method that matches the image patterns with the target patterns.

From author's previous article [3], they proposed a classification model. The model consists of five main stages, starting from image segmentation, object identification, feature extraction, feature selection and finally, image classification. The model was tested using Amazon images. To make a further comparison, Caltech101 dataset were chosen for this paper.

Shape is one of the objects representation in images with the most significant properties, which is famously used in CBIR (content based image retrieval) and in recognition tasks [4].

The Caltech101 object class dataset consists of 101 class of images. It can be downloaded from [5]. With a wide variety of images in each class, this dataset provides a significant intraclass variant [6]. [7] has mentioned that the name of 101 was accidentally set while the author was flipping through pages of Webster Collegiate Dictionary, and they came out with the idea of listing all 101 categories of images. All images in Caltech101 dataset were downloaded using Google Image search engine.

This paper presents the classification of single object in an image using 4 different classifiers. First, images were segmented for partitioning the meaningful part of the image. Second, object identification was applied to the segmented image to detect the connected line in the image. Then, feature extraction of the object was conducted for encoding the valuable features. Next, the experiments were continued with feature selection using Weka tools. Lastly, the classification accuracies of images using 4 types of classifiers selected earlier were presented.

The focus of this paper is to test the proposed techniques toward Caltech101 dataset and compare with Amazon dataset's results. The results of experiments are presented in the paper, and conclusions are drawn.

2. RESEARCH METHOD AND DESIGN

One of the popular research area throughout current years is content-based image classification using large image databases [8].

The proposed technique aims a short and simple way to extract shape description followed by the classification and annotation processes. The proposed method followed these steps:

Step 1: Image Segmentation

Step 2: Object Identification

Step 3: Image Feature Extraction

Step 4: Image Feature Selection

Step 5: Classification

The process of partitioning images into meaningful region is known as ad image segmentation in the computer vision field [9].

The feature is a measurement process which specifically defines a property of an object based on the characteristics of the object. Shape is one of the crucial visual features because of its primitive feature. Shape can be described in two ways; region-based and contour-based. The method that uses the whole area of an object for shape description is region-based. On the other hand, contour-based method uses the information given from the contour of an image [10]. Normally, shape descriptors are combined to create a more effective shape descriptor because individual simple shape descriptor is not robust [11].

The most important step after feature extraction is feature selection. It plays an important role, especially in classification problems. A well-extracted feature must have the value of robustness, discriminative, and easy to compute an efficient algorithm.

The main issue in object detection is to locate the object in an arbitrary image and pose after a non-rigid transformation.

A classifier is chosen after features of an image have been extracted. There are plenty of classifiers that can be used for the experiment.

As stated in Fig.1, Phase 1 is named as data repository, meaning that the process of collecting data takes place before other phases are

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

initiated. In this paper, Caltech101 was used for collecting dataset. The collection process showed that Caltech101 was the most suitable dataset that can be used to test the proposed techniques.

The second phase is the training images, which involved the rest of the processes (from segmentation to classification process) to build the training set.

Initially, in Phase 2, which was named as the training phase, the dataset (images) underwent the image pre-processing steps. The purpose of this measure is to substitute the high-dimensional images with lower-dimensional features that capture the main properties of the images and enable the model to forge on the data with limited storage and computational resources. It includes three main processes starting from image segmentation, followed by feature extraction, and end up with image classification. For the purpose of the next phase of the research, all extracted features were stored in the features database. A feature selection step was added before the classification step for comparison throughout the experiments.

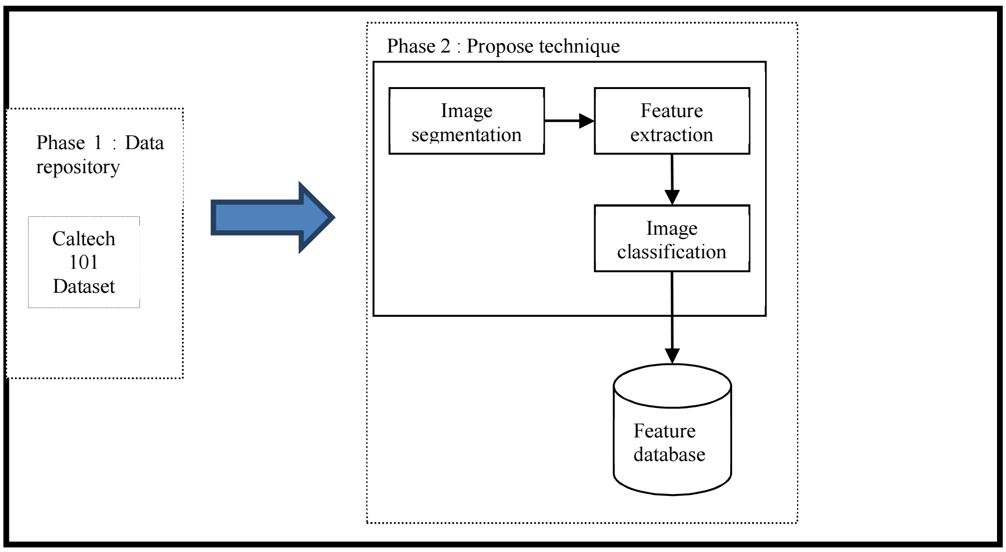


Figure 1: Research method

Journal of Theoretical and Applied Information Technology
10th January 2015. Vol.71 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. SEGMENT IMAGES

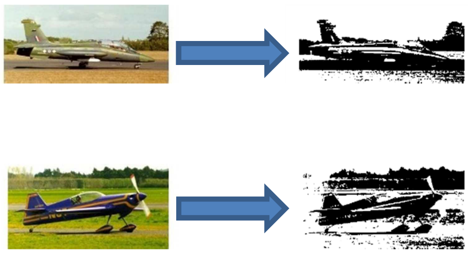


Figure 2: Sample of image classification results using Otsu method

By applying a large dataset from Caltech101, the image data were segmented using thresholding based method. This paper used one of the global thresholding methods, which is Otsu method. [12]. This technique thresholds the entire image with a single threshold value [12], and this system is dependent upon discriminant analysis.

As discussed by [13][14][15], threshold operation can divide the image into two classes; A_1 and A_2, at gray Q such that A_1 = [22 ... , Q] and A_2 = {Q + 1, Q + 2, ... , k-1}, where k is the total number of the gray levels of the image. Let the number of pixels at **i** gray level be n_i, and $N = \sum_{i=0}^{k-1} n_i$ be the total number of pixels in a given image. The probability of occurrence of gray level **i** is defined as $p_i = \frac{n_i}{N}$, $p_i \geq 0$, $\sum_{i=0}^{k-1} p_i = 1$. A_1 and A_2 are normally corresponding to the object of interest and the background. The probabilities of the two classes are:

$P_{A1} = \sum_{i=0}^{Q} p_i$ and $P_{A2} = \sum p_i = 1 - P_{A1}$.

The mean of A_1 and A_2 classes can be computed as:

$$\mu_{A1} = \sum_{i=0}^{Q} \frac{i^* p_i}{P_{A1}} \quad (1)$$

$$\mu_{A1} = \sum_{i=Q+1}^{k-1} \frac{i^* p_i}{P_{A2}} \quad (2)$$

Thus, we can get the equivalent formula:

$$\sigma^2(Q) = P_{A1} P_{A2} (\mu_{A1} - \mu_{A2})^2 \quad (3)$$

The optimal threshold Q* can be obtained by maximizing the between-class variance.

$$Q^* = Arg \max_{0<Q<k-1} \sigma^2(Q) \quad (4)$$

4. FEATURE EXTRACTION

MATLAB was chosen as a tool to develop the extraction procedure using built-in Image Processing Toolbox function known as region props. The process of attaining image features from an input image was initiated with the image properties calculation such as area, eccentricity, extent, solidity, filled area, and others. The features were figured using built-in principles in MATLAB.

5. CLASSIFIER

The classifiers that were applied in our research include BayesNet, NaieveBayesUpdateable, Random Tree and IBk.

5.1 weka.classifiers.bayes.BayesNet

Bayesian networks (BNs), also known as belief net-works (or Bayes nets for short), belong to the family of probabilistic graphical models (GMs).

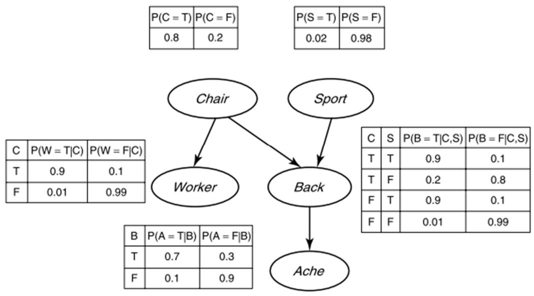


Figure 3: The backache BN example

Figure 3 shows the example of BayesNet process. The parents of the variable *Back* are the nodes *Chair* and *Sport*. The child of *Back* is *Ache*, and the parent of *Worker* is *Chair*. Following the BN independence hypothesis, some independence statements can be detected in this case. For example, the variables *Chair* and *Sport* are slightly independent, but when *Back* is given they are conditionally dependent. This relation is often called explaining away. When *Chair* is given, *Worker* and *Back* are conditionally independent. When *Back* is given, *Ache* is conditionally independent of its ancestors *Chair* and *Sport* [16].

5.2 Weka.Classifiers.Bayes.Naivebayesupdateable

A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive

Journal of Theoretical and Applied Information Technology
10th January 2015. Vol.71 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

term for the underlying probability model would be "independent feature model".

The Class is used for a Naive Bayes classifier using estimator classes. This is the updateable version of Naïve Bayes. This classifier will use a default precision of 0.1 for numeric attributes when build Classifier is called with zero training instances [17].

5.3 weka.classifiers.tree.RandomTree

The decision tree (DT) is a multi-stage decision making or classification tool. It is different to other classification model because it uses input-output relationship that can be expressed using human understandable rules. Meanwhile, other classification model is much difficult to describe [11].

A random tree is a tree drawn at random from a set of possible trees. In this context "at random" means that each tree in the set of trees has an equal chance of being sampled. Another way of saying this is that the distribution of trees is "uniform".

5.4 weka.classifiers.lazy.IBk

IBk is called instance-based learning that generates classification predictions using only specific instances. Instance-based learning algorithms do not maintain a set of abstractions derived from specific instances. This approach extends the nearest neighbor algorithm, which has large storage requirements [17].

6. IMAGE CLASSIFICATION (10-Fold Cross Validation)

The cross validation technique works as repeated holdout. It divides dataset into 10 parts (fold) by holding out each part in turn. Then, it will compute the average results where each data point was used once for testing and 9 times for training.

7. DATASET

The experiment used Caltech101 dataset as mentioned earlier. Originally, the dataset were collected from Google Image search engine to gather as many images as possible for each group [7]. Fig.3 shows examples of the 101 object categories used in this paper. In addition, the model parts were ordered by their x-coordinate, which was problematic for vertical structures. Therefore, the categories with mostly vertical structure were rotated to a random angle. It is for the sake of programming simplicity. Finally, the images were scaled roughly to around 300 pixels wide, producing Caltech101.

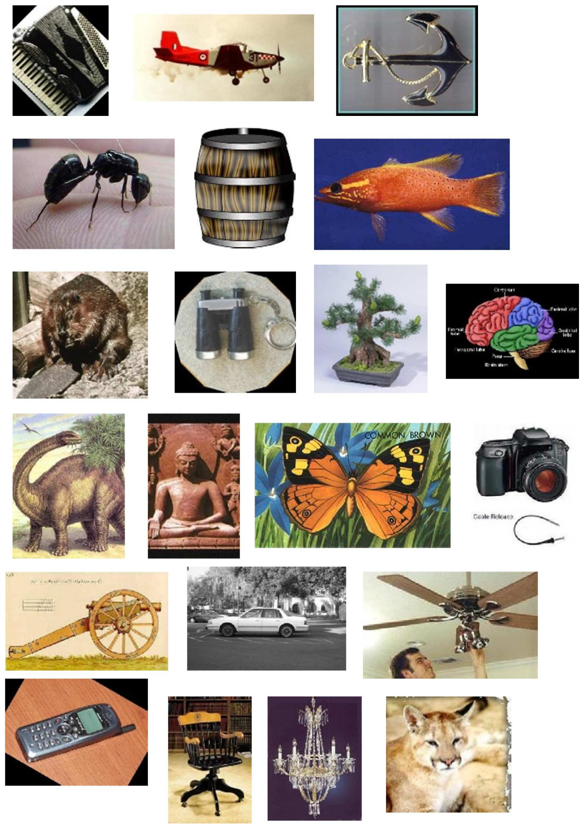


Figure 4: Caltech101 dataset

Figure 5 shows the sample images from Amazon dataset. The images also undergo same steps as Caltech101 dataset because finally the classification results will be compares.

Journal of Theoretical and Applied Information Technology
10th January 2015. Vol.71 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

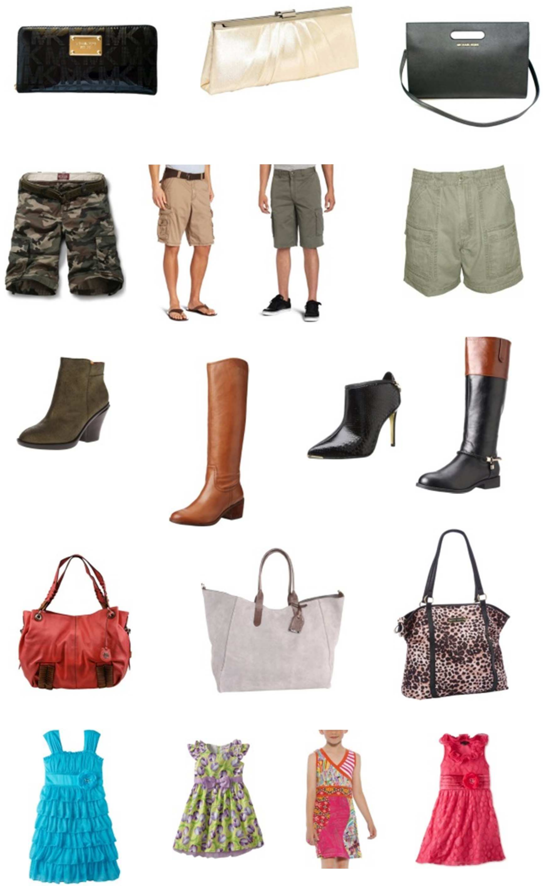


Figure 5: Amazon dataset

8. FEATURE SELECTION

The irrelevant input features may lead to overfitting. Feature selection focuses on the outstanding attributes over the dataset, which offers higher accuracy. There are lots of potential benefits of feature selection such as facilitating data visualization and data understanding, reducing utilization times and techniques, reducing storage requirements and measurement, and defying the curse of dimensionality to improve prediction performance.

CFS (Correlation-based Feature Selection) is an algorithm that couples the evaluation formula that is grounded on ideas from test theory, as well as provides an operational definition of this surmise with an appropriate correlation measure, and a heuristic search strategy.

The evaluator cannot work alone. There must be a search method in order to provide a good predictive power. There are three groups of variable subset selections; wrapper, filters, and embedded [18]. This paper used wrapper, which was provided in the Weka tool. Wrappers utilize the machine learning of interest as a black box to mark subsets of variable according to their predictive power. Its methodology offers a powerful and simple way to notify the problem of variable selection regardless of the chosen machine learning.

As already mentioned earlier, 11 features were combined in this experiment. After performing feature selection using CFSSubset evaluator, only 2 features were selected from the overall set of features. They were area and minor axis length. To make it fair, another evaluator was also used in this experiment. Principle components evaluator suggested seven features out of eleven. They were area, major axis length, minor axis length, eccentricity, orientation, convex area and filled area.

At this point, it was clear that 2 main features were suggested by both evaluators, which were area and minor axis length.

9. CLASSIFICATION RESULTS AND ANALYSIS

The main objective of image classification is to calculate the accuracy of classified images based on the categories stated. The tests were performed on the Caltech101 dataset that consist of 12 features for each single image. A series of experiments was conducted using all features, each with a different number of training images per category (only 30 categories were used). There are a total of 2,594 number of images used in this experiment from part of the 101 categories into training and test sets.

Table 1 lists the results of classification accuracy for images using four different classifiers. As stated in the table, the results show that random tree provides the highest accuracy with feature selection CFSSubsetEvaluator, with 99.00%.

Journal of Theoretical and Applied Information Technology
10th January 2015. Vol.71 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	BayesNet	NaiveBayesUpdateable	Random Tree	IBk
No Feature Selection	95.95	87.36	70.89	58.02
GreedyStepWise + CFSSubsetEvaluator	98.92	97.80	99.00	90.02
Ranker + PrincipleComponent	97.73	92.06	86.89	64.46

Table 1: Result Caltech101

Compare to result in Table 1, Amazon data which is single object image shows higher accuracy when applying CFSSUbsetEvaluator with 99.48%. It shows that the proposed model are more suitable for single object image because most of the accuracy shows in Table 2 is higher compare to Table 1.

	BayesNet	NaiveBayesUpdateable	Random Tree	IBk
No Feature Selection	80.33	90.32	93.81	92.23
GreedyStepWise + CFSSubsetEvaluator	98.12	94.72	99.48	95.20
Ranker + PrincipleComponent	91.53	93.02	98.18	93.99

Table 2: Result Amazon

After comparing both results, it clearly stated that the proposed model is more suitable for single object images. It is because, Caltech images is not a single image. As introduced in section 7, Caltech is an image with various colour of background. It cannot be said it is a single image. It is a multiple images. Compare to Amazon images, it is clearly a single object image with similar background.

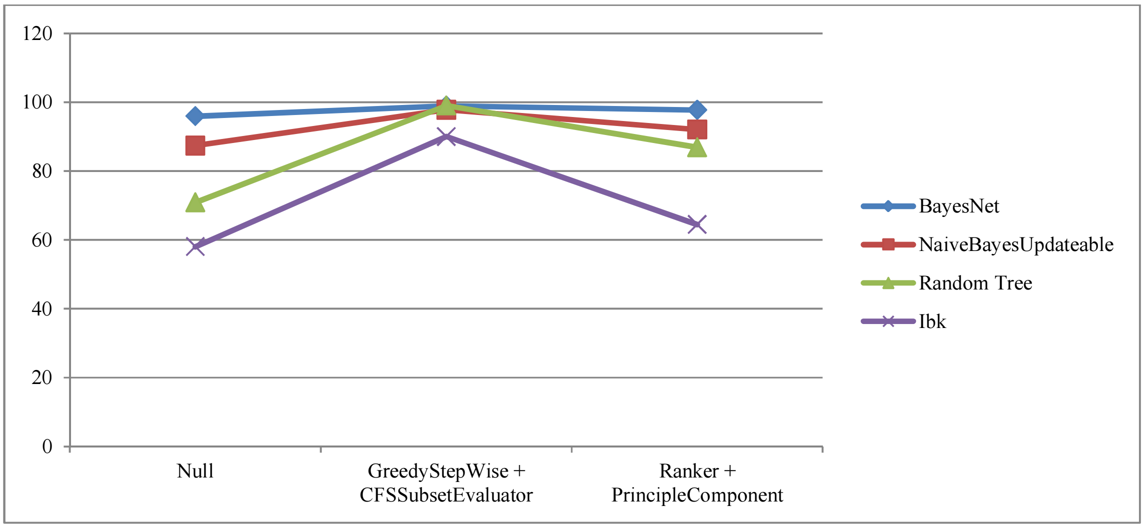


Figure 6:Graph for Table 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

10. CONCLUSION

We have presented in this paper new image classification technique based on shape for single object image. This paper presents a simple method based on a few set of image features to describe shapes. The contributions of the paper are the proposed technique which contain five steps starting from image segmentation, object identification, feature extraction, feature selection and image classification. Then, we proved that the technique can provide good results and it is can be said that compatible with the dataset. Experimental results using Caltech101 datasets show that the proposed technique achieve better image classification performance when using Random Tree classifier compare to other classifier.

The success of an image classification depends on several factors. In order to identify or define objects represented in images, shape is one of most valuable features.

11. ACKNOWLEDGEMENT

This work is partially supported by UniSZA (Grant No. UNISZA/13/GU (029).

REFERENCES:

[1] P. Kamavisdar, S. Saluja, and S. Agrawal, “A Survey on Image Classification Approaches and Techniques,” vol. 2, no. 1, pp. 1005–1009, 2013.
[2] D. L. & Q. Weng, “A survey of image classification methods and techniques for improving classification performance,” *Int. J. Remote Sens.*, 2007.
[3] Mokhairi Makhtar, Engku Fadzli, and Shazwani Kamarudin, “The Contribution of Feature Selection and Morphological Operation For On- Line Business System ' s Image Classification,” *World Appl. Sci. J.*, 2014.
[4] J. F. Nunes, P. M. Moreira, and J. M. R. S. Tavares, “Shape Based Image Retrieval and Classification,” 2010, pp. 433–438.
[5] Fei-Fei, Fergus, and Perona, “Caltech 101 dataset,” 2004. [Online]. Available: http://www.vision.caltech.edu/feifeili/Datasets.htm.
[6] A. C. Berg, T. L. Berg, J. Malik, and U. C. Berkeley, “Shape Matching and Object Recognition using Low Distortion Correspondences,” 2005.
[7] L. Fei-Fei, R. Fergus, and P. Perona, “Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories,” *Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 178–178, 2004.
[8] S. Banerji, A. Sinha, and C. Liu, “New image descriptors based on color, texture, shape, and wavelets for object and scene image classification,” *Neurocomputing*, vol. 117, pp. 173–185, Oct. 2013.
[9] P. A. B. Miss Hetal J. Vala, “A Review on Otsu Image Segmentation Algorithm,” *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. Volume 2, no. Issue 2, 2013.
[10] R. S. Chora, “Image Feature Extraction Techniques and Their Applications for CBIR and Biometrics Systems,” *Int. J. Biol. Biomed. Eng.*, vol. 1, no. 1, 2007.
[11] M. M. Dengsheng Zhang n GuojunLu, “A review on automatic image annotation techniques,” *Pattern Recognit.*, pp. 346–362, 2012.
[12] P. . Sahoo, S. Soltani, and a. K. . Wong, “A survey of thresholding techniques,” *Comput. Vision, Graph. Image Process.*, vol. 41, no. 2, pp. 233–260, Feb. 1988.
[13] N. OTSU, “A Threshold Selection Method from Gray-Level Histograms,” *2EEE Trans. SYSTREMS, MAN, Cybern.*, vol. Vol 9, 1979.
[14] A. A. M. Al-kubati, J. A. M. Saif, and M. A. A. Taher, “Evaluation of Canny and Otsu Image Segmentation,” pp. 24–26, 2012.
[15] M. Huang, W. Yu, and D. Zhu, “An Improved Image Segmentation Algorithm Based on the Otsu Method,” *2012 13th ACIS Int. Conf. Softw. Eng. Artif. Intell. Netw. Parallel/Distributed Comput.*, pp. 135–139, Aug. 2012.
[16] Ruggeri F., Faltin F., and Kenett R., “Bayesian Networks,” *Encyclopedia of Statistics in Quality & Reliability*. Wiley & Sons, 2007.
[17] P.Santhi and V. Murali Bhaskaran, “Improving the Performance of Data Mining Algorithms in Health Care Data,” *IJCST*, vol. 2, no. 3, pp. 152–157, 2011.
[18] I. Guyon, “An Introduction to Variable and Feature Selection,” vol. 3, pp. 1157–1182, 2003.