Thesis no: MSCS-2016-11

Multi-Label Classification Methods for Image Annotation

Bekalu Mullu Brhanie

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author(s):
Bekalu Mullu Brhanie
E-mail: bebr14@student.bth.se

University advisor:
Dr. Huseyin Kusetogullari
Computer Science and Engineering

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

ACKNOWLEDGEMENT

Firstly, I thank God for his blessings to let me complete my thesis work. Then I want thank everyone who helped and advised me for my thesis work.

I would like to express my special thanks to my supervisor Dr. Huseyin Kusetogullari for his continuous support, inspiration and friendly behavior throughout my thesis work. His guidance, detailed comments and feedback helped me a lot to improve the quality of this thesis work.

I am also thankful for all staffs of BTH who directly and indirectly helped me for the accomplishment of my thesis work.

I am very grateful to Swedish institute for providing financial support during my work.

Finally, I thank my wonderful family members and friends for their never ending support and motivation.

ABSTRACT

Context: Multi-label classification is a task of assigning a given example to more than one class labels. Multi-label image annotation is mainly concerned with assigning semantic concepts or labels for a given image. Due to large increase of digital images all over the world, efficient ways to analyze, annotate and manipulate image data has become highly important. The task of multi-label classification of image can be conducted by using machine learning algorithms.

Objectives: In this study, comparison and analysis is done on classification performance of five commonly applied multi-label methods for image data classification. The main objective is to identify best performing algorithm for categorization of image data with respect to classification accuracy and execution time evaluation metrics and then to provide working principle of the identified best algorithm with different technical aspects.

Methods: The methodology of this research consists of literature survey, analysis and experiment. Firstly, a comprehensive literature review is done to identify commonly applied methods that are implemented to handle classification task of image data. Afterwards, experimental set up is done for five selected algorithms including RAKEL, BR, CLR, ML-KNN and LP. The experimental set up includes writing of a java code for the experiment at open source java library called Mulan. Each algorithm parameter is also set to their default value. Finally, each algorithm classification performance is evaluated using publicly available image datasets including Scene, Flags, Corel5K and NUS-WIDE5K. For all image datasets, default percentage decomposition provided by Mulan repository is used. In the experiment a variety of evaluation metrics are used. These different metrics include hamming loss, precision, F-measure, accuracy, recall and execution time. After measurement of experiment is finished, best performing algorithm in terms of accuracy and execution time measurements are identified. This provides answer for RQ1 and RQ2. Finally, comprehensive analysis of the best algorithm is done in order to understand technical detail and working principle of the best algorithm. This provides answer for RQ3.

Results: From the result of literature review, RAKEL, BR, CLR, ML-KNN and LP are identified as commonly applied algorithms for multi-label classification of image data. Although there is no clear winner between methods, the experimental result shows that ML-KNN has better classification accuracy and both ML-KNN and LP have good performance on execution time evaluation metric. In addition to this, working principle of the best performing algorithm is presented by providing its general description, pseudocode, strengths, weaknesses and computational complexity.

Conclusions: Although ML-KNN has its own weakness, it still has a good performance for image data classification on accuracy and execution time measurements. By handling ML-KNN weaknesses, its performance can be improved further and good classification accuracy can be achieved for image data annotation.

Keywords: Image annotation, Empirical study, Multi-label learning, classification, Machine Learning, Image Analysis.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF GRAPHS

LIST OF ABBREVIATIONS

BP-MLL	Back-Propagation Multi-Label
BR	Binary Relevance
CLR	Calibrated Label Ranking
ECC	Ensembles of Classifier Chains
EPS	Ensembles of Pruned Sets
HOMER	Hierarchy Of Multi-labEl leaRners
KNN	K Nearest Neighbor
LP	Label Powerset
ML-kNN	Multi-Label k Nearest
MAP	Maximum Posteriori Principle
PS	Pruned Set
RAKEL	RAndom K-labELsets
TREMC	Triple-Random Ensemble Multi-Label Classification
SVM	Support Vector Machine

1. INTRODUCTION

In binary or traditional classification task, a given observation of data is associated with a single value of label from a set of two class labels [1]. These set of two class labels are mutually exclusive and both of them cannot be assigned at the same time for categorization of unobserved data. But there are many real world situations where an instance can be associated with more than two class labels. For example a given instance of scene image can be associated with class label values of mountain, beach and sunset. Unlike binary or traditional classification, multi-label classification is concerned with assigning several target values for a given instance [2, 3].

Tradition classification problem has been studied widely. Recently, multi-label classification is becoming a hot area of study due to the increasing number different real world application domains, such as text documents categorization [4], medical diagnosis [5], functional genomics [6], semantic annotation of image and video [7,8], directed marketing [9], music categorization into emotions [10], protein function classification and web page categorization etc [11,12,13]. From these application domains, this study will provide useful insight on multi-label classification of image. Multi-label image annotation is mainly concerned with assigning multiple semantic concepts or labels for a given image [3, 14].

In order to deal with problem of multi-label classification task, different methods have been developed by research community. These multi-label methods can be grouped into three categories: problem transformation, algorithm adaption and ensemble based methods [15, 16]. Problem transformation methods map the original multi-label learning tasks into a number of binary classification tasks. Then each binary classification task will be handled by a traditional classification algorithm. An example of problem transformation methods includes Binary Relevance (BR) [1, 17], Label Powerset (LP) [17], Calibrated Label Ranking (CLR) [18] and Pruned Sets (PS) [19]. Whereas algorithm adaption methods concerned on extension and adaption of single label classifier for handling of multi-label classification task. Some of the methods that are under this category include Multi-Label k Nearest Neighbors (ML-kNN) [20], Hierarchy Of Multi-labEl leaRners (HOMER) [21] and Back-Propagation Multi-Label Learning (BP-MLL) [22]. On the other hand ensemble based methods ensemble a number of classifiers and handle multi-label classification task. An example of ensemble based methods includes RAndom K-labELsets (RAKEL) [23], Triple-Random Ensemble Multi-Label Classification (TREMLC) [24], Ensembles of Classifier Chains (ECC) [25] and Ensembles of Pruned Sets (EPS) [19].

The specific focus of this paper is to identify best performing algorithm for multi-label image annotation. Since the problem to be addressed is multi-label classification, a best performing algorithm will be selected from domain of multi-label methods. In order to identify best performing algorithm, extensive experiment has been conducted. A number of evaluation measures are used for performance comparison of methods. In the following sections, the previous works in this area, related works, problem focused, aim and objectives, research questions, expected outcome, methodology of the research, result of research, analysis and suggestion of future works will be provided.

2. RELATED WORKS

Related works regarding with image annotation can be categorized into multi-instance learning, multi-label learning and multi-label multi-instance learning.

Multi-instance learning is extension of supervised learning where labels are assigned to bags of instances [26]. The bag is labeled positive if there is at least one instance in it is positive. But a bag is labeled negative when all instances in it are negative. Multi-instance learning of image classification considers labels correlation and regions into account [6, 27]. Different researchers proposed different methods to handle multi-instance classification task [27, 28, 29].

Multi-instance multi-label learning is a classification task where a training example described by multiple instances and multiple class labels [30]. Such classification task occurs in real world scenarios. For example, a given image will contain multiple patches described by feature vector and the image can be classified into multiple categories as semantic can be recognized in different ways [31]. Different Researchers proposed different learning methods to handle Multi-instance multi-label learning task [30, 31, 32].

Other common way of handling image annotation can be done by using multi-label learning approach. One way to handle multi-label learning problem is transferring multi-label classification task into single label classification task. For example, Boutell et.al. [33] handle multi-label classification of scene by using a single classifier for each label. Instead of treating each label independently, Qi et al. [8] proposed a new multi-label method that handles multi-label classification and correlation of labels.

Nowadays, a number of recent studies are providing a high concern for multi-label classification of different application domains including image categorization [1, 3, 7, 11, 15, 16, 34, 35]. The main focus area of these multi-label classification researches are on performance comparison, technical analysis and classification performance improvement of existing methods. There are also studies concerned on development of a new algorithm.

For scene image categorization, Santos et al. [1] compared classification performance of BR, LP, RAKEL, PS, ECC and EPS methods by using five traditional learning algorithms as a base classifier. Their experimental results show that LP method with SVM algorithm as a base classifier achieved better classification accuracy than its counterparts. Authors in [36] also ensemble ML-KNN, IBLR, RAKEL, CLR and ECC by using different approach and compare classification performance of each method and the ensemble one for different application domains including scene image. Their classification performance comparison results show that the ensemble method has generally a good performance on scene image than each combined method. In addition to comparison of different methods effectiveness for image annotation, Nasierding & Kouzani [37] conducted an empirical experiment on EPS, BPMLL, TREMLC, ML-KNN, BR, LP, RAKEL, CLR and HOMER methods to explore length of execution time of each method. Their experimental results show that BR and ML-KNN consume less execution time.

In addition to performance comparison of methods with regard to classification accuracy and execution time, there are also previous works that provide general overview for technical detail of some algorithms. In [38], a general overview of working principle analysis and common notations of some problem transformation and algorithm adaptation methods are described. Moreover, Nasierding et al. [24] presented merit and demerit of some problem transformation methods. Zhang at el. [39] also provided overview for algorithmic details of eight representative multi-label methods.

3. PROBLEM DEFINITION

For the past decades, digital images have been growing enormously in numbers and size. This is due to advancement in technologies, such as high storage capacity of multimedia database, rising popularity of photography devices and internet sharing sites. As the volumes of images are increasing all over the world, efficient ways to analyze, annotate and manipulate the images are becoming highly important. Although annotation of image is highly important, there are difficulties to handle it effectively. The difficulty of image classification arises due to objects large variability in appearance, pose and illumination [7]. Usually an image will also consist of multiple objects which are associated with multi-labels and characterized by different regions in the image [40]. Another challenge of multi-label classification of image is for small number of available training data, there will be exponential number of possible class configurations. The class labels are also highly sparse and only few of them are active for most samples [3]. Even with these constraints, recognitions and segmentation of image can be done by using multi-label algorithms.

3.1. Problem Focused

According to our knowledge there is no extensive study that compares different multi-label methods by using considerable number of different image datasets and provides a detailed technical analysis of the best image classifier algorithms in terms of different aspects. Hence, it is highly important to conduct extensive and unbiased experimental comparison of different multi-label methods for image annotation. The experiment will help to identify best algorithm that will handle image annotation in a better. After identification of best algorithm, technical analysis will be done to understand its working principle. This will provide a better insight for future performance improvement of the best algorithm. The general process and problem formulation of this thesis work is described in the following diagram.

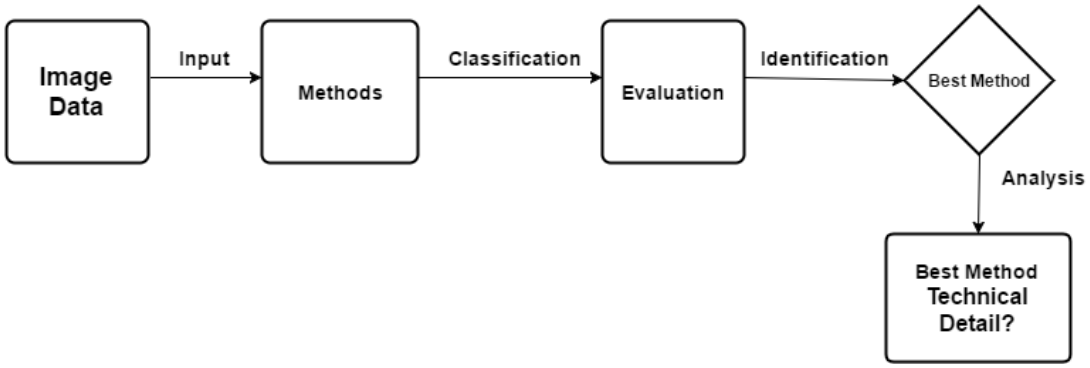


Figure1-Block diagram for problem formulation.

The methods will get image data and then classify it. After classification is done by each method, performance evaluation will be conducted. Then identification of best algorithm and its technical detail will be investigated. The detailed process of this thesis work will be described in the methodology section.

3.2. Aim and Objectives

3.2.1. Aim

The main aim of this thesis is to identify best performing method for categorization of multi-label image data with respect to classification accuracy and execution time evaluation metrics. Since the problem to be addressed is multi-label classification task, identification of best method will be done from domain of multi-label methods. Then working principle and technical detail will be provided for the identified algorithm that has most accurate classification performance among its counterparts.

3.2.2. Objectives

In order to address the main aim, the following objectives are listed below.

- Identification and selection of state-of-art algorithms which are commonly applied for image data categorization.
- Conducting extensive experimental evaluation and measuring a variety of performance metrics.
- Detailed comparative analysis for each of commonly applied algorithms.
- Analyze the technical detail of algorithmic property and working principle of the best algorithm.

3.3. Research Questions

The following research questions are prepared in order to address the problem focused and then to fulfill the aim and objectives set above.

RQ1. Which algorithm provides the most accurate result for multi-label classification of image data?

Motivation: It is required to conduct an experiment in order to evaluate classification performance of state-of-art algorithms. This is helpful to identify up-to-date multi-label algorithm which has better classification accuracy for image data categorization.

RQ2. Which algorithm has the best execution time for multi-label classification of image data?

Motivation: It is important to evaluate performance of multi-label algorithms in terms of execution time. When algorithm has less execution time, it will classify the image data fastly.

RQ3. What is the detailed working principle of the most accurate algorithm?

Motivation: It is to understand in detail how the best algorithm works. It is important to perform detail analysis of the best algorithm which has the best classification accuracy

among counterparts. It is also good to perform technical detail and working principle of fastest algorithm. But if an algorithm takes short time and classification accuracy performance is very low, the chance of selecting the fastest method for multi-label image annotation is less. Analyzing the most accurate algorithm will be highly helpful for better understanding, manipulation and further classification accuracy improvement of the algorithm.

3.4. Expected Outcome

The following outcomes are expected on completion of this master thesis.

- Identification and selection of state-of-art algorithms which are commonly applied for image data categorization.
- Conducting extensive experimental evaluation and measuring a variety of performance metrics.
- List of commonly applied algorithms for image categorization.
- Evaluating the performance of the most commonly applied algorithms for image categorization.
- A list of tables and diagrams that consist of data for different performance measurements.
- Understanding and analyzing the performance of the best performing algorithm which is the most effective for image categorization.
- Understanding of technical detail and working principle of the best algorithm.
- Pseudo-code of the most commonly applied algorithms will be provided.
- Suggestion for area of performance improvement for the best algorithm

4. RESEARCH METHODOLOGY

The methodology of this research consists of literature survey, analysis and experiment. Firstly, a comprehensive literature review will be done. This will help to identify five commonly applied methods that are implemented to handle classification task of image data. Afterwards, experiment will be set up for these identified algorithms. Next, experiment will be conducted to evaluate classification performance of selected algorithms on publicly available image datasets. Each algorithm performance will be evaluated using a variety of performance metrics including hamming loss, precision, F-measure, accuracy, recall and execution time. After measurement of experiment is finished, the best performing algorithm in terms of accuracy and execution will be identified. This will help to answer RQ1 and RQ2. Finally, comprehensive analysis of the best algorithm will be done in order to understand technical detail and working principle of the best algorithm. This will provide answer for RQ3.

The detailed process of research methodology is provided in the following figure below. As shown in the figure, collected image data will be provided to multi-label methods. Each datasets has its own train and test data. After experiment evaluation is done identification and analysis will be done to answer all research questions.

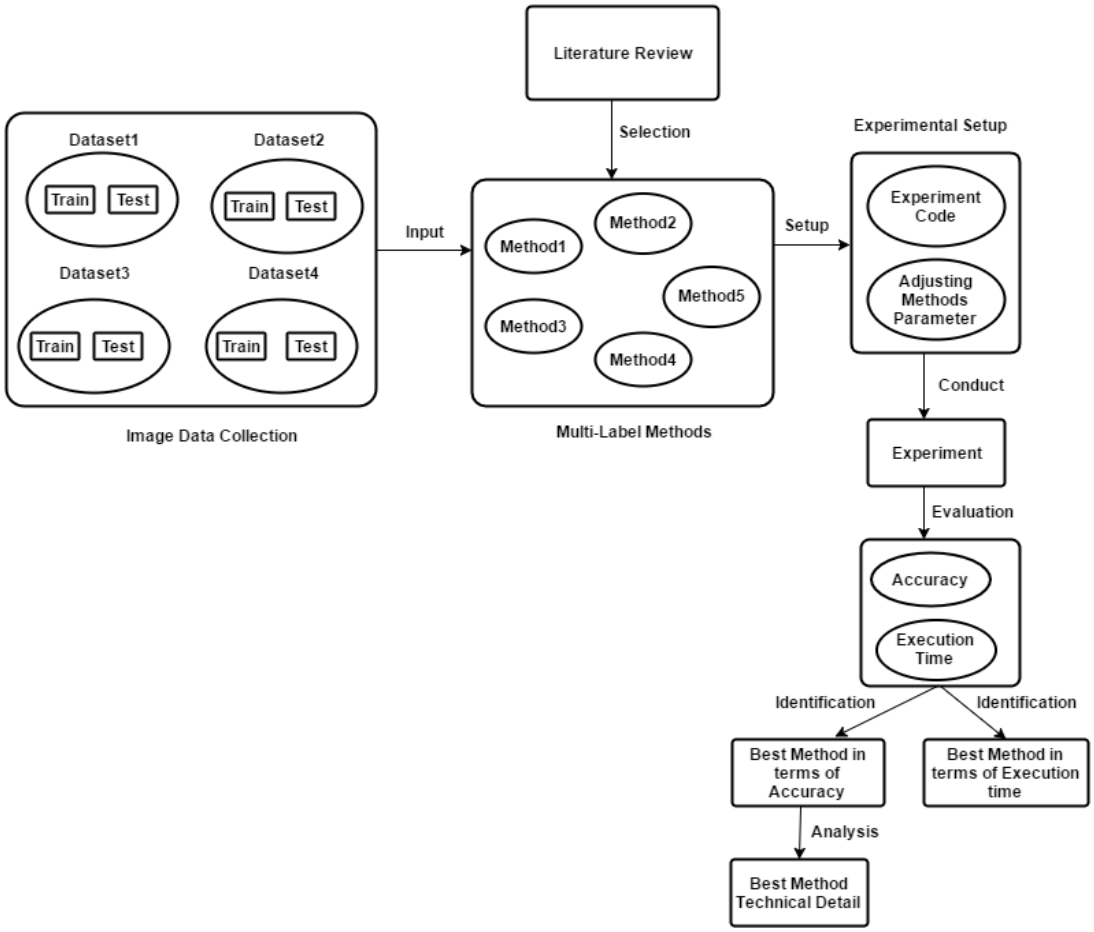


Figure2-Detailed process of Research Methodology.

The detail process of the research methodology is discussed the next sections.

4.1. Literature Review and Analysis

4.1.1. Algorithms Selection

There are a number of machine learning methods developed to handle multi-label classification of a particular dataset. Multi-label methods differ from other machine learning methods as they are the only type of machine learning methods that can handle multi-label classification task. Some of multi-label methods include LP, BR, BP-MLL, CLR, ECC, EPS, HOMER, ML-KNN, PS, RAKEL, TREMC etc.

Before conducting the experiment, firstly selection of commonly applied multi-label method for image classification is done by using literature review. Five multi-label algorithms are selected based on the frequency of method usage by other researcher for image annotation. The analysis is done on 20 related papers [1-3, 7, 11, 14, 15, 23, 24, 34-44] and result shows that RAKEL, ML-KNN, BR, CLR and LP are the most widely used algorithms by research community for image annotation. The following table illustrates the algorithms usage ranking and description.

Rank	Multi-Label Methods	Description
1	RAKEL	Considers label correlation but it has high time complexity
2	ML-KNN	Work well for text classification
3	BR	Simple, relatively fast but it does not consider label correlation
4	LP	Consider label correlation but its computational complexity increases with high number of labels and training data
5	CLR	Efficient pairwise label comparison approach but it is conceptually expensive

Table1-Selected Multi-label Methods

The detailed description of each of the identified multi-label method will be described in the following section.

4.1.2. Multi-Label Methods

For better understanding of each of the identified multi-label method, the general description and pseudocode of each method is provided below.

i. RAKEL (RAndom K-labELset)

It is one of ensemble based method. The training set labels will be decomposed randomly into a number of labelsets. For N number of training instances and L number of labels, the maximum bound for number of labelset is given by $\min(N, 2^L)$ [23]. Each labelset will have a size of k. Then for a given labelset, one LP model will handle assignment of instances. Each LP in turn uses one binary algorithm for each single label classification problem. After training a number of LP models for each label subset, prediction of unseen dataset is done by using average votes. Each LP model prediction for a given label is summed up and then divided by the total number votes. The final divided result will be the average vote of corresponding label. Then the average vote of each label will be used for the final prediction [23, 45, 46]. If the value of the vote result is greater than threshold value, the corresponding

label will be predicted as relevant. Otherwise it will be predicted as irrelevant. The pseudocode for training and classification process of RAKEL is provided below.

Definitions: L: Number of labels, k: Labelset size, T: Threshold, m_i: the i^{th} LP model (m), n: Number of models. T_i: Training set which consists of i^{th} labelset, t: test instance.

Input: T, k, t, m, n.

Output: A set of labels which consist of all j^{th} label (L_j) of t.

```
1.  //Training process.
2.  for i:1 to L  do
3.      Train m_i with T_i;
4.  endfordo
5.  //Classification process.
6.  for j:1 to L do
7.      sum_j=0;                    .
8.      numvotes_j=0;
9.      for i:1 to n do
10.         sum_j+=m_i (t, L_j);
11.         numvotes_j+=1;
12.     endfordo
13.     avgvote_j=sum_j/numvotes_j;
14.     if avgvote_j >=T then
15.        L_j=1;
16.     else
17.        L_j=0;
18.     endif
19. endfordo
20. return {L_1,L_2, … , L_L};
```

Figure3-Pseudo-code of random k-labelset.

ii. BR (Binary Relevance)

It is one of the most commonly applied problem transformation method [47]. A multi-label training dataset with L number of labels will be decomposed into L number of single label training dataset. Then for each of the label, a separate binary classifier will be trained. In order to classify a given instance, union operation will be done on relevant label prediction of each model [41]. The drawback of binary relevance is that it does not handle label correlation [38].

The pseudocode for training and classification process of BR is provided below.

Definitions: L: Number of labels, T_i: Binary training set with i^{th} label, m_i: The i^{th} binary classifier model (m), t: test instance.

Input: T, t, m.

Output A set of labels which consist of all j^{th} label (L_j) of t.

1. //Training Process.
2. for i:1 to L do
3. Train m_i with T_i;
4. endfordo
5. //Classification Process.
6. for j=1 to L do
7. $L_j = m_j(t, L_j)$;
8. endfordo
9. return $\{L_1, L_2, \ldots, L_L\}$;

Figure4-Pseudo-code of binary relevance.

iii. Calibrated Label Ranking (CLR)

It is an efficient pairwise label comparison approach. In addition to classifying the labels of an example into relevant and irrelevant, it provides ranks of each label set by conducting pairwise comparison of labels [18]. To conduct pairwise comparison of a given multi-label classification task with L number of labels, a total of L*(L-1)/2 binary classifiers will be trained. For each pair of labels, one binary model will be trained. In addition to this, another L number of binary classifier is required for comparison of each label with a calibration label. Calibration label is a label which is artificial inserted for separation of relevant and irrelevant classes. After training process is finished, each model will predict the preferred label from pair of labels. Finally the ranking of each label is provided by using voting scheme [18]. The voting of calibrated label will be used as a threshold. When the vote of a label is greater than the threshold value, the label will be classified as relevant. Otherwise the label will be classified as irrelevant.

The pseudocode for training and classification process of CLR is provided below.

Definitions: L**:** Number of labels, $L_{ij:}$ Pairwise label for i^{th} and j^{th} label, t: test instance, T: Training set, T_{ij}: Binary training set which consists of instances with distinct relevance value of i^{th} and j^{th} label. It will have class value of 1 when i^{th} class exists in original multi-label training set. Otherwise it will have a value of 0, T_{ic}: Binary training set which consists of the i^{th} and a calibration label. It will have a class value of 1 if i^{th} class is present in original multi-label training set. Otherwise it will have a value of 0, m_{ij}: A binary classifier (m) for i^{th} and j^{th} label preference comparison, m_{ic}: A binary classifier (m) for i^{th} and calibration label preference comparison.

Input: T, m, t.

Output: A set of labels which consist of all j^{th} label (L_j) of t.

1. //Training Process
2. for i=1 to L-1 do
3. for j=i+1 to L do
4. Train m_{ij} with T_{ij};
5. endfordo
6. endfordo
7. for i=1 to L do

```
8.      Train m_ic with T_ic;
9.  endfordo

10. //Classification Process
11. for i=1 to L do
12.     sumvote_i=0;
13.     sumvote_c=0;
14. endfordo;
15. for i=1 to L-1 do
16.     for j=i+1 to L do
17.         if(m_ij(t, L_ij) =1)
18.             sumvote_i=+1
19.         else
20.             sumvote_j=+1
21.     endfordo
22. endfordo
23. for i=1 to L do
24.     if(m_ic(t,L_ic) =1)
25.         sumvote_i+=1;
26.     else
27.         sumvote_c+=1;
28.     endif
29. endfordo
30. for i=1 to L do
31.     if(sumvote_i> sumvote_c)
32.         L_i=1;
33.     else
34.         L_i=0;
35.     endif
36. endfordo
37. return {L_1,L_2, … , L_L};
```

Figure5-Pseudo-code of calibrated label ranking.

iv. ML-KNN (Multi-Label K Nearest Neighbors)

ML-KNN is a lazy learning approach and it is one of algorithm adaptation based method. It extends the traditional K Nearest Neighbor (KNN) method for multi-label classification. In addition to KNN, ML-KNN uses Bayesian reasoning approach [44, 48]. Firstly, k- nearest neighbors of a given test instance will be selected from the training set. Then each label occurrence in the neighbor training set will be counted. Finally statistical analysis mechanism called Maximum Posteriori Principle (MAP) will be applied [20]. Prior and posterior probability of a label will be estimated from the training set. This probability estimation will be used for the final prediction of the label set of a test instance [44, 49].

The pseudocode for training and classification process of ML-KNN is provided below.

Definitions: L: Number of labels, L_j: j^{th} label of the test instance, $y_j i$: j^{th} label of the i^{th} training example, $r_j i$: Number of neighbors of the i^{th} training example with label j, T: training set, T_i: The i^{th} training example, n= Total number of training example, t=test instance, s=smoothing parameter. Generally it has a value of 1, $P(H_j)$: Prior probability of t has j^{th} label, $P(\neg H_j)$: Prior probability of t has no j^{th} label, n_j: Number of training examples

associated with the j^{th} label, $c_j[i]$:Counts number of training instances with j^{th} label and have exactly i neighbors with label j, $c'_j[i]$: Counts number of training instances without j^{th} label and have exactly i neighbors with label j, E_ji: donate event that t has i number of neighborhood training examples with label j, k: Size of neighborhood, $P(E_ji| H_j)$: Posterior probability that t has i neighbors with label j when t has the j^{th} label, $P(E_ji|\neg H_j)$: Posterior probability that t has i neighbors with label j when t is without j^{th} label.

Input: T, k, t, s.

Output: A set of labels which consist of all j^{th} label (L_j) of t.

```
1.  for i to n do
2.      Identify k nearest neighbors for T_i;
3.  endfordo

4.  //Compute prior probability
5.  for j=1 to L do
6.     P(H_j) =s+n_j/(s*2+n);
7.     P( ¬H_j) =1- P(H_j);
8.  endfordo

9.  //Compute posterior probabilities.
10. Identify  k nearest neighbors for t;
11. for j=1 to L do
12.     for i=0 to k do;
13.         c_j[i]=0; c_j'[i]=0;
14.     endfordo
15. endfordo
14. for  j=1 to L do
15.     for i=0 to k do
16.         if (y_ji==1) then
17.             c_j[r_ji]+=1;
18.         else
19.             c'_j[r_ji]+=1;
20.         endif
21.     endfordo
22. endfordo
23.  for j:0 to L do
24.     for i:0 to k  do
                                                  n=k
25.         P(E_ji| H_j)= (s+c_j[i])/( s*(k+1) + ∑ (c_j[n] ));
                                                 n=0

                                                  n=k
26.         P(E_ji| ¬H_j)= (s+c'_j[i])/( s*(k+1) + ∑ (c'_j[n] ));
                                                 n=0

27.     endfordo
28.  endfordo

29. //Compute labels of test instance, t.
30. for j=0 to L do
31.     if( (P(E_jn_j|H_j)*P(H_j)) > (P(E_jn_j| ¬H_j)*P( ¬H_j)) )
32.         L_j=1;
30.     else
```

```
31.          Lj=0;
         endif
32.  endfordo
33.  return {L1,L2, … , LL};
```

Figure6-Pseudo-code of multi-label k nearest neighbors.

v. LP (Label Powerset)

It is a problem transformation multi-label method. The main advantage of LP is that it considers correlation among labels. But its predictive performance suffers and computational complexity increases when there is high number of labels and training data [23]. In LP, each unique combination of labels in the original multi-label training data is considered as a single label classification task. For each single label, one binary classifier will be trained. For unseen data, a binary classifier of LP outputs the label which has the maximum probability compared to others [17]. The label predicted by a binary classifier is actually a set of labels and it will be used as the final prediction for the test instance.

The pseudocode for training and classification process of LP is provided below.

Definitions: T: Training set, t: test instance, L_{ui}: The i^{th} unique combination of labels in T, $L_{u:}$ Number of unique combination of labels in T, m_i: The i^{th} binary classifier model (m), T_i: A training instance which consists of the i^{th} unique labels combination, L_t: Multi-Label prediction of a test instance, $P(L_{ui}|t)$ or P_i: The probability of i^{th} unique label combination for a given test instance t.

Inputs: T, t, m.

Outputs: L_t .

```
1.  //Training process
2.  for i=1 to Lu do
3.      Train mi with Ti;
4.  endfordo

5.  //Classification process.
6.  for i=1 to Lu do
7.      P0=0;
8.      Pi=P(Lui|t);
9.      if(Pi>Pi-1)
10.       Lt=Lui;
11.     endif
12. endfordo
13. return Lt;
```

Figure7-Pseudo-code of label powerset.

4.2. Experiment

4.2.1. Datasets

To evaluate the performance of algorithms, experiments are conducted on publicly available real-world image datasets. The datasets are taken from Mulan repository [50] and they are from different image data domains including natural scene, flag, corel and social media. The brief description of each dataset is provided below:

- **Scene [33]:** It consists of a natural scene image dataset of different categories including sunset, beach, field, fall foliage, urban and mountain. It is represented by numeric features.
- **Flags [51]:** It consists of image of 194 country flags. It is represented by numeric and nominal attributes.
- **Corel5k [52]:** It consists of 5000 Corel image collection taken from various categories. It has only a nominal attribute. The numbers of labels are relatively large than other datasets.
- **NUS-WIDE5K:** It is image dataset taken from social media Flickr. NUS-WIDE dataset [53] consists of 269,648 images with a total class of 81. Since it is large and computationally expensive dataset, images only from 60 classes are selected. For images belonging to each class, 50 samples are taken. The selected class values include: bicycle, Buddha, calf, cat, bridge, airplane, clothes, cow, dog, actor, camels, donkeys, eagle, elephant, butterfly, fish, car, cathedral, fountain, building, coast, desert, fox, cliff, clouds, fruit, computers, furniture, garden, goat, flowers, forest, hills, grass, hand, food, horse, glacier, lion, lizard, lake, penguin, leaf, moon, motorcycle, people, monks, rabbit, sheep, sunglasses, mushrooms, umbrella, ocean, pyramid, birds, tables, police, tiger, train and woman.

Tabel2 presents the general statistics of image datasets used in the experiment. The described characteristics includes number of train and test instances, number of numeric and nominal attributes and number of labels. For all of the dataset, default percentage decomposition provided by Mulan repository is used.

Dataset Name	Train Instances	Test Instances	Nominal Attributes	Numeric Attributes	Labels
Scene	1211	1196	0	294	6
Flags	129	55	9	10	7
Corel5K	4500	500	499	0	374
NUS-WIDE5K	3060	2040	0	128	81

Table2-Statistics of each image dataset.

4.2.2. Evaluation Measures

In traditional classification, the label assignment of a given instance will be correct or incorrect. But in multi-label classification problem, assignment of labels for a given instance may consist of partially correct and partially incorrect. Due to this, performance of multi-label methods can be measured with different metrics. In our experiment different evaluation measurements are conducted on the identified methods. These evaluation measurements include example based accuracy, hamming loss, precision, F-measure and recall [1,37,54].
Before presenting the definition of measurements, it is important to provide the following definitions. Let T be multi-label dataset donated by (X_i , Y_i) and has a total of N number training examples, Y_i be a set of true labels of example X_i and Y_i* represents a set of labels predicted by multi-label method for the same example X_i. Let L be the number of labels.

i. Accuracy

Accuracy measures the average percentage of correctly predicted labels among predicted and true labels. For each instance, intersection and union between true labels of original training example and the predicted labels are calculated. The ratio between intersection and union will be computed and sum operation will be done for all dataset examples. Then the sum result is averaged with overall number of dataset examples and this provides the accuracy of the method.

The accuracy measurement is provided by the following equation:

$$\text{Accuracy}=1/N \left(\sum_{i=1}^{N} (Y_i \cap Y_i^*)/ (Y_i \cup Y_i^*)\right) \text{----------Equation1}$$

ii. Hamming Loss

It measures the average prediction error of multi-label classification. When a method has a relatively low hamming loss value, it means the method will have a relatively better performance.

$$\text{Hamming Loss} = 1/N \left(\sum_{i=1}^{N} (Y_i \Delta Y_i^*)/ (L)\right) \text{----------Equation2}$$

iii. Precision

It measures the rate of true positive instances from positively classified instances, then averaged by overall number of training example.

$$\text{Precision}=1/N \left(\sum_{i=1}^{N} (Y_i \cap Y_i^*)/ (Y_i^*)\right)\text{----------Equation3}$$

iv. Recall

It measures the rate of true positive instances with total number of labels.

$$\text{Recall}=1/N \left(\sum_{i=1}^{N} (Y_i \cap Y_i^*)/ (Y_i)\right)\text{----------Equation4}$$

v. F-measure

It is harmonic mean of precision and recall measurement

$$\text{F-measure}=1/N \left(\sum_{i=1}^{N} 2\times(Y_i \cap Y_i^*)/ (Y_i^* - Y_i)\right)\text{----------Equation5}$$

vi. Execution Time

In addition to different classification performance measurement of methods, the computational time spent is also measured. In the experiment of this thesis paper, execution time includes the time required to train the classifier and the time for prediction of test dataset.

Execution Time= Train time+ Prediction time. ----------Equation6

4.2.2.1. Motivation for different metrics measurement

Different performance metrics including example based accuracy, hamming loss, precision, F-measure and recall are provided in the experiment result. Even if accuracy is used for analysis of result, other metrics measurement is provided to give general overview of quality of classification performance from different perspective. The definition of each of the measurement metrics is provided in evaluation measures section

4.2.2.2. Motivation of selecting accuracy for result analysis

Multi-label classification will consist of partially correct and partially incorrect result. Due to this, evaluation of multi-label classification of image is difficult compared to binary classification. In this thesis paper, example based accuracy is used for performance comparison of methods. Compared to other evaluation metrics, accuracy gives balanced measurement and it is a better indicator of classification performance for most multi-label classification problems [45].

4.2.2.3. Motivation of execution time measurement

In addition to performance measurement of classification accuracy, it is important to investigate the time complexity. Even if the algorithm has better classification accuracy, if it has high execution time it may not be preferred as a best algorithm. So including execution time will help to get overview of total time spent for training and classification. After the result of each algorithm execution time is known, tuning of algorithm parameter can be done in order to reduce its execution time.

4.2.3. Experimental Setting

The experiment is conducted on the Mulan library [50]. Mulan is an open source Java library which extends Weka [27] for problem of multi-label classification task. The five commonly applied algorithms that are identified from literature review already exist at Mulan. For consistency, decision tree C4.5 (j48) is used as a base learner for all transformation based methods. The threshold (T) is set to default value of 0.5. Other parameters of all methods are also set with their default value. For example, for ML-KNN, there are 10 nearest neighbors and the smoothing parameter(s) has a value of 1 and for RAKEL, the size of labelset (k) is set to 3.

After experimental setting is finished, train-test experiment is conducted for measurement of accuracy and execution time performance of each method on each image dataset. For all of image datasets, default percentage decomposition provided by Mulan repository is used. The experiment is conducted on a Lenovo laptop with specification of processor: Intel core i5, CPU: 2.5 GHz and RAM: 6 GB. The experiment is repeated 20 times and the average is taken for the final result. The result of train-test experiment will be presented in the following section.

5. RESULT

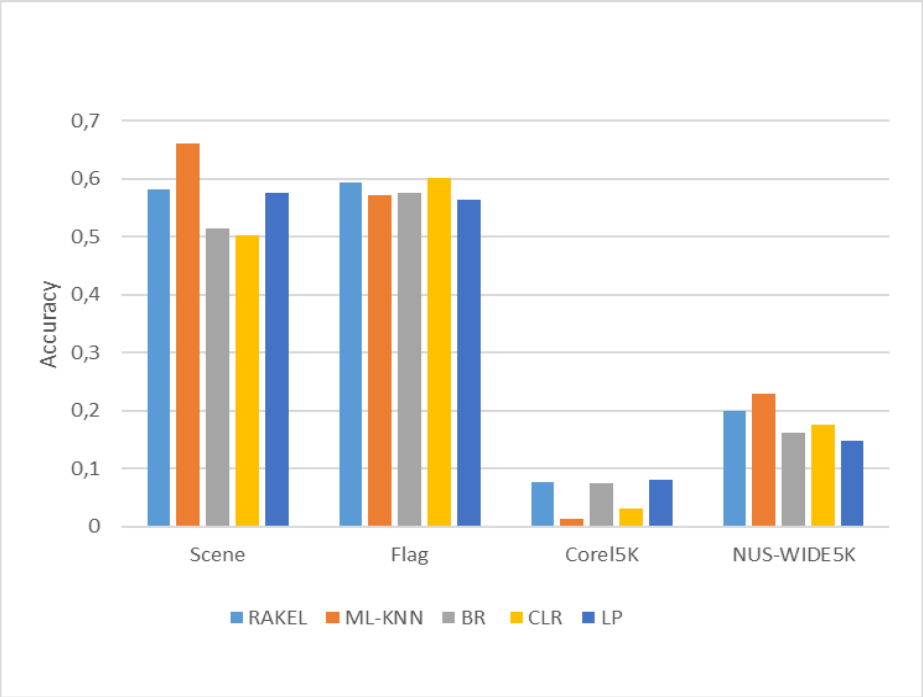
5.1. Experimental result

The train-test experiment is conducted on RAKEL, ML-KNN, BR, CLR and LP algorithms and their performance is evaluated in Mulan repository. The following table lists each algorithm performance on scene, flag, Corel5K and NUS-WIDE5K image datasets. Different evaluation measurement metrics including accuracy, hamming loss, precision, F-measure and recall is provided. These measurements will provide general overview each algorithm performance from different perspective. But in this thesis work, accuracy will be used for performance analysis and comparison.

Dataset	Algorithm	Accuracy	Hamming Loss	Precision	F-measure	Recall
	RAKEL	0.5818	0.1139	0.6087	0.6054	0.625
	ML-KNN	0.6614	0.0953	0.6923	0.6816	0.6906
	BR	0.5134	0.1389	0.534	0.5524	0.6112
	CLR	0.5033	0.1409	0.5216	0.5492	0.6321
Scene	LP	0.5761	0.1479	0.6019	0.5896	0.5907
	RAKEL	0.5934	0.2747	0.6767	0.7133	0.7726
	ML-KNN	0.5715	0.2681	0.7385	0.6961	0.7005
	BR	0.5763	0.2747	0.6956	0.711	0.7741
	CLR	0.6008	0.2615	0.6907	0.736	0.8451
Flag	LP	0.5637	0.2923	0.6754	0.6799	0.6869
	RAKEL	0.0762	0.0096	0.1979	0.1101	0.0825
	ML-KNN	0.014	0.0093	0.035	0.0195	0.0145
	BR	0.0753	0.0098	0.1959	0.1103	0.0838
	CLR	0.0313	0.0095	0.0997	0.0475	0.0323
Corel5k	LP	0.0798	0.0166	0.1226	0.116	0.1136
	RAKEL	0.199	0.0315	0.3153	0.2534	0.2434
	ML-KNN	0.2295	0.0248	0.3236	0.2568	0.2338
	BR	0.1624	0.0358	0.25	0.212	0.2128
NUS-WIDE5K	CLR	0.1757	0.0267	0.2039	0.1837	0.1757
	LP	0.1472	0.0415	0.2109	0.1904	0.2051

Table3-Performance of methods using different image dataset.

The graph below provides graphical comparison of methods classification performance with respect to accuracy is provided for each image dataset.



Graph1-Performance comparison of methods in terms of accuracy.

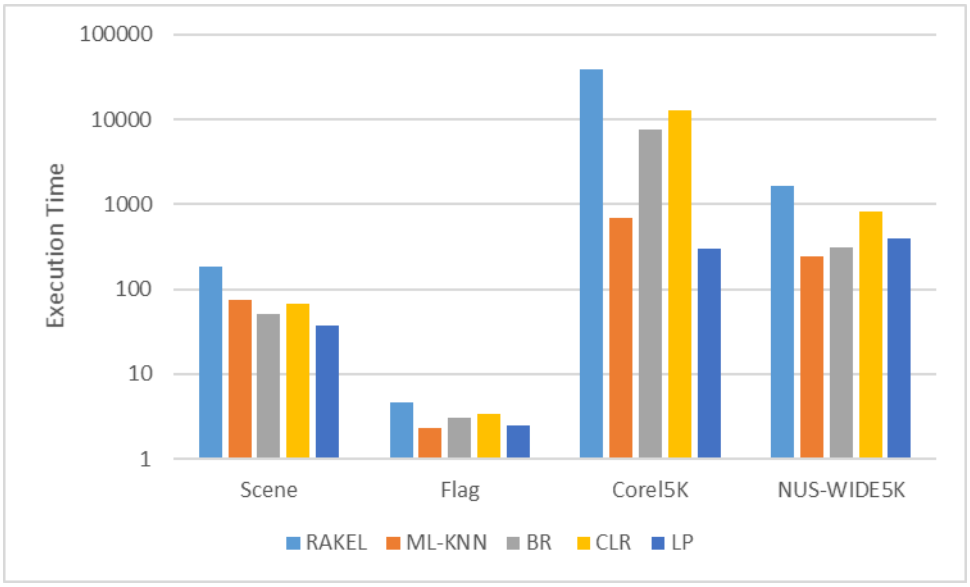
The graphical performance illustration of accuracy result shows that the classification accuracy of all methods is less than 70% for all datasets including scene, flag, corel5k, NUS-WIDE-5K. Most methods are good for classifying scene and flag data. With regarding to corel5k and NUS-WIDE-5K datasets, most algorithms have not good classification performance.

In addition to classification performance, the result of time execution is also provided below. This execution time includes the total time spent for training of a classifier and the time spent for prediction of the given image dataset.

Dataset	Algorithm	Execution time(s)
Scene	RAKEL	18.472
	ML-KNN	7.422
	BR	5.024
	CLR	6.74
	LP	3.76
Flag	RAKEL	0.462
	ML-KNN	0.23
	BR	0.3
	CLR	0.34
	LP	0.25
Corel5k	RAKEL	3847.715
	ML-KNN	69.37
	BR	750.978
	CLR	1284.552
	LP	30.106
NUS-WIDE5K	RAKEL	162.771
	ML-KNN	24.53
	BR	30.56
	CLR	82.018
	LP	39.798

Table4-Performance of methods with respect to execution time.

The graph below also provides graphical comparison of methods classification performance with respect to execution time is provided for image datasets. Since the execution time of scene and flag dataset is very low, firstly the measurement values are multiplied by 10 times and then base 10 logarithmic scale is used for execution time.



Graph2-Performance comparison of methods in terms of execution time.

As shown in the above diagram, compared to other datasets Corel5k takes longer execution time. Especially for RAKEL method, the time of execution is longer than one hour. For other dataset the time of execution is less than three minute.

5.2. Working principle and Technical detail

Before providing working principle and technical detail, it is important to identify the best multi-label method which has a better classification accuracy performance.

5.2.1. Selection of best algorithm

As can be seen from the experiment result, there is no single method which has better classification accuracy on all image datasets. Although there is no clear winner among methods, ML-KNN has a better performance on half of the datasets. Due to this ML-KNN is selected as a best algorithm for categorization of the image data. The general description and pseudocode of ML-KNN is provided in methodology section above. That will provide general overview of its working principle. In this section, its working principle will be analyzed in detail in terms of different technical aspects. Its strengths, weaknesses and computational complexity are described below. Suggestion of its performance improvement will be described in the analysis and discussion section.

5.2.2. Strengths and Weaknesses

Identifying the weakness and strength of ML-KNN is important for efficient and effective usage. This will also help for further performance improvement. Different approaches can be used to overcome its limitation and weakness. The main strength and weakness of ML-KNN compared to other algorithms is provided below.

5.2.2.1. Strengths

ML-KNN has its own strength when compared to other multi-label methods. The main strengths of ML-KNN include the following:

i. Simplicity

The concept of working principle, interpretation and implementation of ML-KNN is easy compared to other complicated algorithms such as SVM based methods. The idea of ML-KNN is to find category of test instance by first identifying k-nearest neighbors in the training data and then computing prior and posterior probabilities of labels.

ii. Easy to tune parameters

It is easy to tune parameters of ML-KNN. By configuring optimal value of size of neighbors (k), performance improvement can be gained. When size of neighbors is small, the noise will have a great effect on final result of classification. A large value of k will increase the computational complexity of ML-KNN. Generally the selection of k depends on the provided training data. Setting high value of k will be effective for training set that has relatively more similar training instances and a lower value of k can be set for relatively heterogeneous examples [56]. Hence by tuning parameters easily, effective usage of the method can be achieved.

iii. Good classification performance

Generally, ML-KNN has better performance for categorization of image data. As shown on result of extensive experiment, ML-KNN has achieved relatively better performance with respect to classification accuracy and execution time.

5.2.2.2. Weaknesses

In addition to its strengths, ML-KNN has also its own weakness. The effect of these weaknesses can be minimized by using different approaches. This will help to get better performance. The main weaknesses of ML-KNN include the following:

i. High computational complexity

In order to compute prior and posterior probability of each member of a labelset, identification and consideration of k-nearest neighbors of the whole training data is required. Due to this, there will be relatively high computational complexity.

ii. Lack of consideration for label correlation

One of the main features of effective multi-label algorithm is its consideration and exploitation of interdependency between labels. Multi-label data has a property of relationship between label elements. Being member of a given category provides information about other category membership. When a given instance belongs to a given class, there will

be a probability that it will also be a member of other class value. For example economic book will have a higher probability to have a class value of export. But the probability of the book to be comedy is less. Hence the class value of economy has a higher dependency with export than its dependency with comedy class value. A multi-label method should consider relationship between labels. Regarding to this property of multi-label classification task, ML-KNN lacks consideration of label correlation. ML-KNN works on principle that the labels of test instance are related to the number of neighbors with a similar labels value.

5.2.3. Computational complexity

The computational complexity of ML-KNN can be analyzed from its pseudocode. The pseudocode is already provided in methodology section above. For a given training data that has n number of training example with d dimension of the feature vector and l number of labels, the computational complexity for training and classification process can be provided.

For training processes, the computational complexity is provided by a function $O(n^2d + lnk)$. In the function provided: k is the size of neighborhood, l and d are dimension of feature vector and number of labels respectively.

Training computational complexity = $O(n^2d + lnk)$ ----------Equation3.

Whereas for the testing (classification) process, it has complexity function of $O(nd + lk)$.

Testing computational complexity= $O(nd + lk)$ ----------Equation4.

From these equation3 and 4, it can be seen that the computational complexity of ML-KNN depends on total number of training example (n), dimension of feature vector (d), number of labels (l) and size of neighborhood(k) parameter.

6. DISCUSSION AND ANALYSIS

6.1. Experimental result discussion and analysis

As can be seen from table3 in result section, there is no single method that perform well for all image datasets. ML-KNN achieved better classification accuracy for scene and NUS-WIDE5K dataset. But it has worst performance for corel5k data. For flag dataset, CLR has better classification accuracy. But it has worst performance for scene dataset. On other hand, LP has better performance for corel5k image data. But still it has the worst performance for classification of NUS-WIDE5K. Even though there is no clear winner among methods, ML-KNN has good classification accuracy on half of image datasets. Because of this ML-KNN is the best performing algorithm in terms of classification accuracy of image dataset categorization.

With regard to execution time, both ML-KNN and LP achieved a good performance. LP has a less execution time for scene and corel5k dataset. Whereas ML-KNN achieved a better classification time for flag and NUS-WIDE5K dataset. On the other hand, RAKEL has the worst classification time performance on all image datasets including scene, flag, corel5k and NUS-WIDE5K. Similar to classification accuracy performance, there is no single classifier that has a best execution time in all image data. But both ML-KNN and LP has less execution time on half of image datasets. Because of this both ML-KNN and LP are the best performing algorithms in terms of execution time of image dataset categorization.

Working principle of ML-KNN is also described by providing its general description, pseudocode, strengths, weaknesses and computational complexity. Even it has the provided weaknesses; generally, it has a better performance on both classification accuracy and execution time evaluation metrics. Because of this, a considerable concern should be given to improve its performance further. Some of the weaknesses can be handled for better performance.

The performance of ML-KNN can be improved by using different techniques. One way of performance increment can be done by building a model that learn different weight for different neighbors of a given test instance. Highest weight will be given for neighbor that is closest than others. Finally, the weighted vote will be used for determination of label set of the test instance. The other way of performance improvement can be done by extending ML-KNN to handle label correlation. One way of exploitation for label correlation can be done by combining nearest neighbor method with other approaches. Cheng and Eyke [44] showed that by combining k-nearest neighbor method with logistic regression, interdependence between labels can be handled and at the same time a better accuracy can be achieved. Label interdependency can also be handled and performance can be improved by adjusting rule of label estimation. Instead of considering only one label of each neighbor for prediction of a given label of unseen instance, performance achievement can be gained by applying a global maximum a posteriori probability (MAP) rule. In global MAP rule, posterior probability estimation considers all neighbors labels of a given unseen instance [49]. Changing Euclidean distance [20] measurement for selection of k nearest neighbors with other distance learning methods like neighborhood components analysis [57] and large margin nearest neighbor [58] can also be used to improve performance of ML-KNN. In this regard, extensive experiment can be conducted to assess the possible performance improvement. In addition to this, ML-KNN can be combined with other multi-label methods by using ensemble technique. This will help to overcome ML-KNN weakness and its performance can be improved. Different researchers applied this approach to improve performance of multi-label algorithms for classification of different application domains. For

example, Sanden and John [10] improves performance of ML-KNN for music genre classification by combining it with other four methods. Extensive experiment can be conducted to test this approach for image categorization.

6.2. Validity of Threats

In this section internal validity and external validity of the thesis work will be presented. The counter measurements taken to handle these threats will also be provided.

Internal validity: The experiment is conducted in one machine. In order to reduce this effect, experiment is conducted while other parallel computer tasks were closed. This was especially important for execution time measurement. In addition to this, experiment is repeated 20 times and average of the total measurement metrics are taken and presented in the result section.

External validity: In order to make the experiment more dependable and externally valid, a lot of effort is done on algorithms and datasets selection. Common multi-label algorithms are selected from recent studies based on their popularity. The selected datasets are representative of real world image datasets. The datasets are also taken from different application domains including satellite, Corel, countries flag and social media images.

6.3. Limitations

The limitations of this thesis work includes the following.

Limited number of dataset: In the experiment four datasets from different image domain is included. In order to see a large scale effect, large number of datasets can be used. Due to limited time available to adjust the image for multi-label algorithm, four sample image datasets are selected.

Limited number of algorithms: Experimental comparison is done on five algorithms. Large number of algorithms can be selected and then experiment can be conducted to asses each algorithm performance. Due to limited time, experiment is conducted on five most popular multi-label methods.

7. CONCLUSION AND FUTURE WORK

In this thesis paper, a comparison between different multi-label methods is conducted on image categorization by using scene, flag, corel5k and Nus-wide5k datasets. The experimental result shows that ML-KNN is the best performing algorithm with respect to classification accuracy and both ML-KNN and LP have a good performance on execution time evaluation metrics. In addition to this, working principle of the best performing algorithm is described by providing its general description, pseudocode, strengths, weaknesses and computational complexity.

In the future, different researches can be done in order to improve performance of ML-KNN further. This will provide better classification accuracy. Some of improvement areas are suggested at analysis and discussion section above. These performance improvement areas include handling label correlation, using other metrics learning than Euclidean distance measurement, ensemble with other classifier and applying a weighted ML-KNN model. A new algorithm that has a better performance than existing state-of-art methods can also be developed for categorization of image data. In addition to multi-label classification other standard and non-standard classification experiment including multi-tasking learning, deep learning, online learning and active learning can be conducted on image data.

8. REFERENCES

[1] Santos, A., A. Canuto, and Antonino Feitosa Neto. "A comparative analysis of classification methods to multi-label tasks in different application domains."*Int. J. Comput. Inform. Syst. Indust. Manag. Appl* 3 (2011): 218-227.

[2] Li, Tao, Chengliang Zhang, and Shenghuo Zhu. "Empirical Studies on Multi-label Classification." In *IcTAI*, vol. 6, pp. 86-92. 2006.

[3] Kim, Minyoung. "Multiple-concept feature generative models for multi-label image classification." *Computer Vision and Image Understanding* 136 (2015): 69-78.

[4] McCallum, Andrew. "Multi-label text classification with a mixture model trained by EM." In AAAI'99 workshop on text learning, pp. 1-7. 1999.

[5] Li, Guo-Zheng, Zehui He, Feng-Feng Shao, Ai-Hua Ou, and Xiao-Zhong Lin. "Patient classification of hypertension in Traditional Chinese Medicine using multi-label learning techniques." *BMC medical genomics* 8, no. 3 (2015): 1.

[6] Barutcuoglu, Zafer, Robert E. Schapire, and Olga G. Troyanskaya. "Hierarchical multi-label prediction of gene function." *Bioinformatics* 22, no. 7 (2006): 830-836.

[7] Cabral, Ricardo Silveira, Fernando De la Torre, João Paulo Costeira, and Alexandre Bernardino. "Matrix Completion for Multi-label Image Classification." In *NIPS*, vol. 201, no. 1, p. 2. 2011.

[8] Qi, Guo-Jun, Xian-Sheng Hua, Yong Rui, Jinhui Tang, Tao Mei, and Hong-Jiang Zhang. "Correlative multi-label video annotation." In *Proceedings of the 15th ACM international conference on Multimedia*, pp. 17-26. ACM, 2007.

[9] Zhang, Yi, Samuel Burer, and W. Nick Street. "Ensemble pruning via semi-definite programming." *Journal of Machine Learning Research* 7, no. Jul (2006): 1315-1338.

[10] Sanden, Chris, and John Z. Zhang. "Enhancing multi-label music genre classification through ensemble techniques." In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, pp. 705-714. ACM, 2011.

[11] Santos, Araken M., Anne MP Canuto, and Antonino Feitosa Neto. "Evaluating classification methods applied to multi-label tasks in different domains." In *Hybrid Intelligent Systems (HIS), 2010 10th International Conference on*, pp. 61-66. IEEE, 2010.

[12] Yu, Guoxian, Huzefa Rangwala, Carlotta Domeniconi, Guoji Zhang, and Zhiwen Yu. "Protein function prediction using multilabel ensemble classification." *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 10, no. 4 (2013): 1045-1057.

[13] Ciarelli, Patrick Marques, Elias Oliveira, and Evandro OT Salles. "Multi-label incremental learning applied to web page categorization." *Neural Computing and Applications* 24, no. 6 (2014): 1403-1419.

[14] Nasierding, Gulisong, Grigorios Tsoumakas, and Abbas Z. Kouzani. "Clustering based multilabel classification for image annotation and retrieval." In *Systems, Man and*

Cybernetics, 2009. SMC 2009. IEEE International Conference on, pp. 4514-4519. IEEE, 2009.

[15] El Kafrawy, Passent, Amr Mausad, and Heba Esmail. "Experimental Comparison of Methods for Multi-Label Classification in Different Application Domains." International Journal of Computer Applications 114, no. 19 (2015).

[16] Madjarov, Gjorgji, Dragi Kocev, Dejan Gjorgjevikj, and Sašo Džeroski. "An extensive experimental comparison of methods for multi-label learning."Pattern Recognition 45, no. 9 (2012): 3084-3104.

[17] Tsoumakas, Grigorios, Ioannis Katakis, and Ioannis Vlahavas. "Mining multi-label data." In Data mining and knowledge discovery handbook, pp. 667-685. Springer US, 2009.

[18] Fürnkranz, Johannes, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. "Multilabel classification via calibrated label ranking." Machine learning 73, no. 2 (2008): 133-153.

[19] Read, Jesse, Bernhard Pfahringer, and Geoff Holmes. "Multi-label classification using ensembles of pruned sets." In *2008 Eighth IEEE International Conference on Data Mining*, pp. 995-1000. IEEE, 2008.

[20] Zhang, Min-Ling, and Zhi-Hua Zhou. "ML-KNN: A lazy learning approach to multi-label learning." Pattern recognition 40, no. 7 (2007): 2038-2048.

[21] Tsoumakas, Grigorios, Ioannis Katakis, and Ioannis Vlahavas. "Effective and efficient multilabel classification in domains with large number of labels." In Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08), pp. 30-44. 2008.

[22] Zhang, Min-Ling, and Zhi-Hua Zhou. "Multilabel neural networks with applications to functional genomics and text categorization." Knowledge and Data Engineering, IEEE Transactions on 18, no. 10 (2006): 1338-1351.

[23] Tsoumakas, Grigorios, Ioannis Katakis, and Ioannis Vlahavas. "Random k-labelsets for multilabel classification." Knowledge and Data Engineering, IEEE Transactions on 23, no. 7 (2011): 1079-1089.

[24] Nasierding, Gulisong, Abbas Z. Kouzani, and Grigorios Tsoumakas. "A triple-random ensemble classification method for mining multi-label data." In Data Mining Workshops (ICDMW), 2010 IEEE International Conference on, pp. 49-56. IEEE, 2010.

[25] Read, Jesse, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. "Classifier chains for multi-label classification." Machine learning 85, no. 3 (2011): 333-359.

[26] Maron, Oded, and Tomás Lozano-Pérez. "A framework for multiple-instance learning." Advances in neural information processing systems (1998): 570-576.

[27] Chen, Yixin, and James Z. Wang. "Image categorization by learning and reasoning with regions." Journal of Machine Learning Research 5, no. Aug (2004): 913-939.

[28] Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." Journal of the royal statistical society. Series B (methodological) (1977): 1-38.

[29] Chen, Yixin, Jinbo Bi, and James Ze Wang. "MILES: Multiple-instance learning via embedded instance selection." IEEE Transactions on Pattern Analysis and Machine Intelligence 28, no. 12 (2006): 1931-1947.

[30] Zhou, Zhi-Hua, Min-Ling Zhang, Sheng-Jun Huang, and Yu-Feng Li. "Multi-instance multi-label learning." Artificial Intelligence 176, no. 1 (2012): 2291-2320.

[31] Zhou, Zhi-Hua, and Min-Ling Zhang. "Multi-instance multi-label learning with application to scene classification." In Advances in neural information processing systems, pp. 1609-1616. 2006.

[32] Xu, Xin, and Eibe Frank. "Logistic regression and boosting for labeled bags of instances." In Pacific-Asia conference on knowledge discovery and data mining, pp. 272-281. Springer Berlin Heidelberg, 2004.

[33] Boutell, Matthew R., Jiebo Luo, Xipeng Shen, and Christopher M. Brown. "Learning multi-label scene classification." Pattern recognition 37, no. 9 (2004): 1757-1771.

[34] Modi, Hiteshri, and Mahesh Panchal. "Experimental comparison of different problem transformation methods for multi-label classification using MEKA."International Journal of Computer Applications 59, no. 15 (2012).

[35] Nasierding, Gulisong, and Abbas Z. Kouzani. "Comparative evaluation of multi-label classification methods." In Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on, pp. 679-683. IEEE, 2012.

[36] Tahir, Muhammad Atif, Josef Kittler, and Ahmed Bouridane. "Multilabel classification using heterogeneous ensemble of multi-label classifiers."Pattern Recognition Letters 33, no. 5 (2012): 513-523.

[37] Nasierding, Gulisong, and Abbas Z. Kouzani. "Empirical study of multi-label classification methods for image annotation and retrieval." In *Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on*, pp. 617-622. IEEE, 2010.

[38] Mohammad, S. Sorower. "A literature survey on algorithms for multi-label learning." *Oregon State University, Corvallis* (2010).

[39] Zhang, Min-Ling, and Zhi-Hua Zhou. "A review on multi-label learning algorithms." *Knowledge and Data Engineering, IEEE Transactions on* 26, no. 8 (2014): 1819-1837.

[40] Zha, Zheng-Jun, Xian-Sheng Hua, Tao Mei, Jingdong Wang, Guo-Jun Qi, and Zengfu Wang. "Joint multi-label multi-instance learning for image classification." In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1-8. IEEE, 2008.

[41] Prajapati, Purvi, Amit Thakkar, and Amit Ganatra. "A survey and current research challenges in multi-label classification methods." Int. J. Soft Comput 2 (2012).

[42] Lu, Hong, Yingbin Zheng, Xiangyang Xue, and Yuejie Zhang. "Content and context-based multi-label image annotation." In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pp. 61-68. IEEE, 2009.

[43] Tawiah, Clifford, and Victor Sheng. "Empirical comparison of multi-label classification algorithms." In Twenty-Seventh AAAI Conference on Artificial Intelligence. 2013.

[44] Cheng, Weiwei, and Eyke Hüllermeier. "Combining instance-based learning and logistic regression for multilabel classification." Machine Learning 76, no. 2-3 (2009): 211-225.

[45] Rokach, Lior, Alon Schclar, and Ehud Itach. "Ensemble methods for multi-label classification." Expert Systems with Applications 41, no. 16 (2014): 7507-7523.

[46] Tsoumakas, Grigorios, and Ioannis Vlahavas. "Random k-labelsets: An ensemble method for multilabel classification." In Machine learning: ECML 2007, pp. 406-417. Springer Berlin Heidelberg, 2007.

[47] Brinker, Klaus, Johannes Fürnkranz, and Eyke Hüllermeier. "A unified model for multilabel classification and ranking." In Proceedings of the 2006 conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29--September 1, 2006, Riva del Garda, Italy, pp. 489-493. IOS Press, 2006.

[48] Lukasik, Michal, Tomasz Kusmierczyk, Lukasz Bolikowski, and Hung Son Nguyen. "Hierarchical, multi-label classification of scholarly publications: modifications of ML-KNN algorithm." In Intelligent Tools for Building a Scientific Information Platform, pp. 343-363. Springer Berlin Heidelberg, 2013.

[49] Younes, Zoulficar, Fahed Abdallah, Thierry Denoeux, and Hichem Snoussi. "A dependent multilabel classification method derived from the k-nearest neighbor rule." EURASIP Journal on Advances in Signal Processing 2011, no. 1 (2011): 1-14.

[50] Tsoumakas, Grigorios, Eleftherios Spyromitros-Xioufis, Jozef Vilcek, and Ioannis Vlahavas. "Mulan: A java library for multi-label learning." The Journal of Machine Learning Research 12 (2011): 2411-2414.

[51] Correa Goncalves, Eduardo, Alexandre Plastino, and Alex A. Freitas. "A genetic algorithm for optimizing the label ordering in multi-label classifier chains." In Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on, pp. 469-476. IEEE, 2013.

[52] Barnard, Kobus, Pinar Duygulu, David Forsyth, Nando De Freitas, David M. Blei, and Michael I. Jordan. "Matching words and pictures." The Journal of Machine Learning Research 3 (2003): 1107-1135.

[53] Spyromitros-Xioufis, Eleftherios, Symeon Papadopoulos, Ioannis Yiannis Kompatsiaris, Grigorios Tsoumakas, and Ioannis Vlahavas. "A comprehensive study over vlad and product quantization in large-scale image retrieval." Multimedia, IEEE Transactions on 16, no. 6 (2014): 1713-1728.

[54] Tsoumakas, Grigorios, and Ioannis Katakis. "Multi-label classification: An overview." Dept. of Informatics, Aristotle University of Thessaloniki, Greece (2006).

[55] Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. "The WEKA data mining software: an update." ACM SIGKDD explorations newsletter 11, no. 1 (2009): 10-18.

[56] Huang, Ke-Wei, and Zhuolun Li. "A multilabel text classification algorithm for labeling risk factors in SEC form 10-K." ACM Transactions on Management Information Systems (TMIS) 2, no. 3 (2011): 18.

[57] Goldberger, Jacob, Geoffrey E. Hinton, Sam T. Roweis, and Ruslan Salakhutdinov. "Neighbourhood components analysis." In Advances in neural information processing systems, pp. 513-520. 2004.

[58] Weinberger, Kilian Q., and Lawrence K. Saul. "Distance metric learning for large margin nearest neighbor classification." The Journal of Machine Learning Research 10 (2009): 207-244.