M.Tech Dissertation Report

on

Discriminative Parts Discovery For Fine-grained Object Categorization

Submitted in the fulfillment of the requirements

for the award of degree

Master of Technology

in

Computer Science and Engineering

Submitted By:

DEEPIKA KADAM

Enrollment Number: 16535018

Under the supervision of

DR. BIPLAB BANERJEE

Assistant Professor, Dept. of Computer Science and Engineering

Department of Computer Science and Engineering

Indian Institute of Technology Roorkee

May, 2018

Author's Declaration

I declare that the work presented in this dissertation with title **"Discriminative Parts Discovery for Fine-grained Object Categorization"** towards fulfillment of the requirement for the award of the degree of **Master of Technology** in **Computer Science & Engineering** submitted in the **Department of Computer Science & Engineering, Indian Institute of Technology Roorkee, India** is an authentic record of my own work carried out during the period of **May 2017 to May 2018** under the supervision of **Dr. Biplab Banerjee**, Assistant Professor, Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, India. The content of this dissertation has not been submitted by me for the award of any other degree of this or any other institute.

Date: .....
Place: Roorkee

Deepika Kadam
M.Tech CSE (16535018)
Indian Institute of Technology Roorkee

Certificate

This is to certify that Thesis Report entitled **"Discriminative Parts Discovery for Fine-grained Object Categorization"** which is submitted by Deepika Kadam (16535018), towards the fulfillment of the requirements for the award of the degree of **Master of Technology** in **Computer Science & Engineering** submitted in the **Department of Computer Science & Engineering, Indian Institute of Technology Roorkee, India** is carried out by her under my esteemed supervision and the statement made by the candidate in declaration is correct to the best of my knowledge and belief.

Date: .....
Place: Roorkee

Dr. Biplab Banerjee
Assistant Professor
Department of Computer Science & Engineering
Indian Institute of Technology Roorkee

Acknowledgements

I would first like to thank my thesis advisor Dr. Biplab Banerjee for guiding me throughout my thesis work, helping me whenever needed and being a constant source of motivation. I am really grateful for having such a wonderful and understanding mentor. I am also thankful to the Department of Computer Science Engineering and Tinkering Lab, IIT Roorkee for providing the valuable resources to aid my research.

I would like to thank my fellow classmates and friends for being there and helping me overcome the obstacles in my thesis work.

Last but not the least, I would like to thank my parents and siblings for their blessings and support without which I would not have reached this stage of my life.

DEEPIKA KADAM

Abstract

In the past few years, many object classification techniques have been developed. However, these techniques do not perform very well when categorizing objects belonging to fine-grained categories. It is easier to differentiate between objects belonging to different broader categories as they have different higher-level features or parts, like differentiating a dog from a car is an easy task. But when it comes to distinguishing between objects belonging to same category, it can be a challenging task as there is very little variance among their parts, like distinguishing between dogs belonging to different breeds. The task is not only to find parts of an object, but to find discriminative parts that help us categorize objects correctly. Recently part-based techniques have shown promising results in fine-grained categorization.

Keywords: Fine-grained classification, Part-based object classification, Region proposals, Pattern mining

Contents

List of Figures

Chapter 1

Introduction

Fine-grained object classification refers to the task of classifying objects into their sub-categories within a category. For example, consider the task of classifying birds according to their species or classifying dogs according to their breeds. These tasks are often difficult for human brains to perform themselves. This is due to the problem that there exists only subtle differences between objects of different sub-categories. As shown in figure 1.1, the overall shape of the birds of two different categories is same. There is a large intra-class variance due to viewpoint and the background. However, there are only subtle differences between birds of different species ,e.g., the color of the belly. In order to distinguish between them, we need to find which parts are discriminative and also where are they located. We as humans can locate the part by pointing out and checking the difference. The challenge for the machine is to decide which are the discriminative parts and where are they located.

Fine-grained classification is required to accurately categorize objects. It can be useful in the areas of Artificial Intelligence that deal with the navigation of robotic systems through the environment. The system needs to have a detailed understanding about the environment. With fine-grained classification techniques, the robotic system will be better able to understand the objects in it's surrounding through it's visual sensors. Also, many image recognition apps have been developed in the recent years such as Google Goggles, Blippar, Microsoft's Bing Vision that scan images and provide information relating to it. Fine-grained classification can be useful in such apps to provide more detailed and accurate information about the objects. The machine can be trained at fine-grained level to categorize similar looking objects which can be helpful in filter results on e-commerce websites. Fine-grained classification can be used to identify different species of the animal kingdom, track them, maintain detailed information about them so as to preserve them.

Most of the classification techniques focus on detecting the objects accurately in an image by extracting high-level features. These high level features only

a) Acadian Flycatcher

b) Great Crested Flycatcher

FIGURE 1.1: Two different species of Flycatcher from the Caltech-UCSD Birds 200 dataset [1].

track the superficial differences between the objects that can only categorize the objects into their broader categories. However, they fail to recognize the features that can actually help in distinguishing objects from different fine-grained classes. Distinguishing features can be related to different parts of an object, it's peculiar shape or color or the pattern formed over it. For example, parts of bird like the head, beak, or eyes can be the distinguishing factor. Finding these mid-level parts can help us find these distinguishing factors for fine-grained classification. It can help by explicitly showing the subtle appearance differences associated with a particular object part. Also, locating object parts is important for establishing similarities between objects of same fine-grained class despite the pose variation and camera view position.

Mid-level discriminative parts are basically a set of image patches derived from a dataset where we know the image label but the parts are not annotated. According to Li, Liu, Shen, *et al.* [2], these parts need to satisfy two requirements - 1.Representativeness i.e the part should frequently occur in images of the target class. 2.Discriminativeness i.e the part should rarely occur in images of other classes. Association rule mining is a pattern mining technique which helps find frequent patterns within a database. Li, Liu, Shen, *et al.* [3] shows how association rule mining can be used to find frequently occurring parts in the target category of a image dataset. Representation of these parts is one of

the most important task at hand. Li, Liu, Shen, *et al.* [3] shows how CNN activations for these object parts can be appropriate to use with association rule mining algorithm.

In this work, we aim to show how finding different significant parts of an image can play a crucial role while performing the task of fine-grained classification. We will incorporate the association rule mining techniques into our classification task by finding frequent patterns that represent a class within the dataset. We will try to overcome the challenges faced with using association rule mining techniques but at the same time try to derive meaningful patterns or parts from the dataset. Lastly, we will thoroughly train our model to detect the significant frequent patterns in the images and correctly classify the image into one of the fine-grained classes.

Chapter 2

Related Work

Traditional methods used for image classification involved extracting features from the images and then applying learning algorithms like the Support Vector Machines(SVMs) over these features to generate classifiers. Various feature descriptors were popular such as Scale-Invariant Feature Transformation(SIFT) by Lowe [4], Histogram of Oriented Gradients(HoG) by Dalal and Triggs [5] and Speeded Up Robust Features(SURF) by Bay, Tuytelaars, Van Gool, *et al.* [6]. Bag-of-Visual-Words(BoVW) by Yang [7] which was based on the Bag-of-Words technique used for document classification was a famous choice for feature representation. Other feature representation techniques that gained popularity after BoVW were Fisher kernel by Perronnin and Dance [8] and Vector of Locally Aggregated Descriptors(VLAD) by Jégou, Douze, Schmid, *et al.* [9] The performance of these classification techniques depended on the features used. So to better these techniques, better set of features were required. Such techniques involved two steps: 1) Feature Extraction algorithm 2) Learning algorithm. Both of these steps were independent of each other.

The deep learning based techniques for image classification were introduced in ILSVRC Russakovsky, Deng, Su, *et al.* [10] competition in 2012. In this competition, AlexNet [11] was introduced which is a deep convolutional neural network(CNN) with the top-5 test error rate of 15.3% as compared to 26.2% rate of the next best entry.AlexNet is a simple network made up of 5 convolutional layers, dropout layers, max pooling layers, 3 fully-connected layers and uses 11x11 sized filters in the first layer. The fully-connected layers produce a 4096 dimension activations that can be used to represent the image. In subsequent years, many variations of the CNN were designed out of which ZFNet by Zeiler and Fergus [12], VGG Net in ILSVRC 2014 by Simonyan and Zisserman [13], GoogLeNet by Szegedy, Liu, Jia, *et al.* [14] gained popularity. VGG Net was introduced in ILSVRC 2014 by Simonyan and Zisserman [13] and had a top-5 test error rate of 7.3%. VGG-16 has 13 convolutional layers along with three fully connected layers whereas VGG-19 has three additional convolutional layers.It

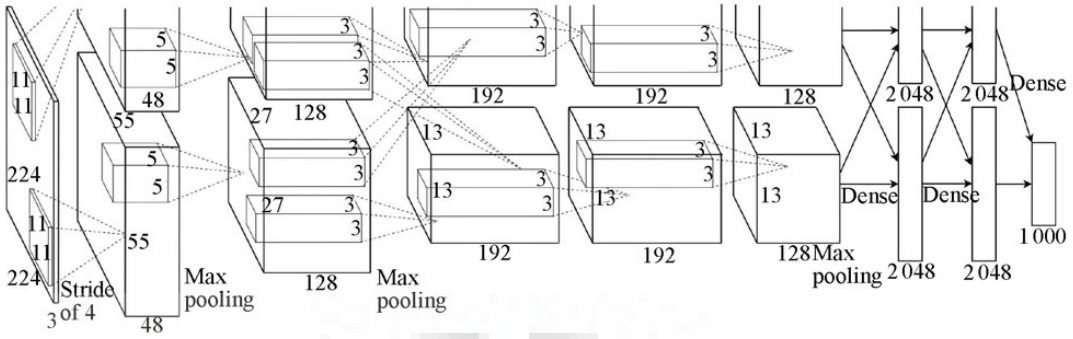


FIGURE 2.1: Architecture of AlexNet [11]

has a filter size of 3x3. The reason for using such small filter size is that two 3x3 layers have an effective receptive field of one 5x5 layer while three back-to-back 3x3 layers will have effective receptive field of 7x7. The model is able to simulate benefits of large filters while keeping the original filter size small and increasing the depth.

CNNs use the traditional sliding window approach where the stride parameter is specified which incurs an overhead. To overcome this, region proposal or object proposal method was presented which is nothing but detection of regions that are most likely to contain objects. The region proposal algorithms output large number of image patches at multiple scales that may probably contain objects. Selective Search by Uijlings, Van De Sande, Gevers, *et al.* [15] is one such region proposal algorithm. Object detection process is split in two steps : the region proposal step and classification step. Region-based CNN(R-CNN) by Girshick, Donahue, Darrell, *et al.* [16] makes use of Selective Search for region proposals. Selective search generates about 2000 region proposals per image. These proposals are then warped to a specific image size and fed to a trained CNN for generating fixed-size feature vector for each region. Then the vector is given as input to set of class-specific linear SVMs that output a class label for that region.The vector is also given as input to a bounding box regressor to output most accurate co-ordinates. Fast-RCNN - developed by Girshick [17] increase the speed by sharing computations of the layers between different proposals. The image is first fed through a CNN to obtain a feature map. The features of the regions are derived from this feature map. Faster R-CNN was introduced by Ren, He, Girshick, *et al.* [18] overcomes the complex pipeline in the previous two techniques. A Region Proposal Network(RPN) is inserted after the last conv layer. The RPN produces region proposals with object bounds and objectness score from the feature map of last conv layer. In our approach, we will be using this RPN to generate the region proposals.

The part-based approaches for fine-grained image classification follow a general approach - first the parts of objects are localized, then alignment of parts is done to overcome pose variations and then classification is done based on features extracted over the parts. In Part-based RCNN proposed by Zhang, Donahue, Girshick, *et al.* [19] whole-object and part detectors are learned. Test images have both objects and parts annotated. First, several region proposals are made using selective search and both object and part detectors are trained using deep convolutional features. At test time, all proposals are scored by all detectors and geometric constraints are applied to re-score the proposals in order to choose the best object and part detections. Drawback of this technique is having images in the dataset to be part annotated which involves a lot of work. There are other part-based techniques proposed that do not require part-annotations. One such technique is proposed by Krause, Gebru, Deng, *et al.* [20] which focuses on learning expressive appearance descriptors and localizing discriminative parts. It provides us with a representation of an image called Ensemble of Localized Learned Features(ELLF) which is nothing but collection of part appearances learned through CNN. Part discovery is fully unsupervised in this method. Another method by Krause, Jin, Yang, *et al.* [21] makes use of the concept alignment by segmentation for generating parts and R-CNN for feature extraction and classification. Juneja, Vedaldi, Jawahar, *et al.* [22] addresses the problem of discovering discriminative parts by an incremental part learning process starting from single part occurrence using Exemplar SVM and introduces entropy-rank concept to determine usefulness of a part. Expanded Parts Model(EPM) proposed by Sharma, Jurie, and Schmid [23] automatically mines parts and learns corresponding discriminative template along with their respective locations. The part-based method proposed by Mettes, Gemert, and Snoek [24] says that parts are naturally shared among image categories and should be modeled as such. Part selection is not done separately for each class but instead shared and optimized over all categories.

Pattern mining was primarily used to find frequent patterns in normal database. In recent years, pattern mining techniques have been used for computer vision tasks such as image classification and action recognition. Li, Liu, Shen, *et al.* [2] proposes the Mid-level Deep Pattern Mining Algorithm which focuses on mid-level visual element discovery. It makes use of the CNN to extract features from the image patches and then apply association rule mining to get category-specific patterns. The patterns are then mapped onto patches to get the mid-level visual elements. Ensemble merging is performed to merge similar visual elements and finally a set of visual elements along with their detectors

is obtained.Li, Liu, Shen, *et al.* [3] also proposes two feature encoding methods. The first method suggests Bag-of-Patterns representation based on the Bag-of-Visual-Words representation. The second method merges the visual elements and trains detectors, followed by construction of Bag-of-Elements representation.

Chapter 3

Problem Definition

Consider a training set T containing N images $T = \{X_1, X_2, \ldots, X_N\}$ with each image containing an object belonging to one of the categories in set C. Given an image X with a label Y such that $Y \in C$, we decompose the image into several part proposals. Let a function $F(X)$ map the image X into k part proposals $F(X) = \{p_1, p_2, p_3 \ldots, p_k\}$. For each of the part p_i, we have a corresponding feature representation x_i such that $x_i \in \mathbb{R}^d$. We perform the mapping for all training images in set T and collect all parts in a set $P = \cup_{X \in T} F(X)$. In our approach, we use RPN [18] to get part proposals from an image and and 4096-dimensional CNN activation represents the feature x_i for a part p_i.

Now, our problem is to find a suitable encoding technique to convert the feature representation x_i into itemset and apply association rule mining using these itemsets and setting proper parameters to generate frequent patterns. These frequent patterns would then be used as an representation of a particular class. We can then train a classifier such as SVM over these frequent proposals which can then be used to get a prediction function $h : X \rightarrow Y$ to predict label Y for image X.

Chapter 4

Proposed Solution

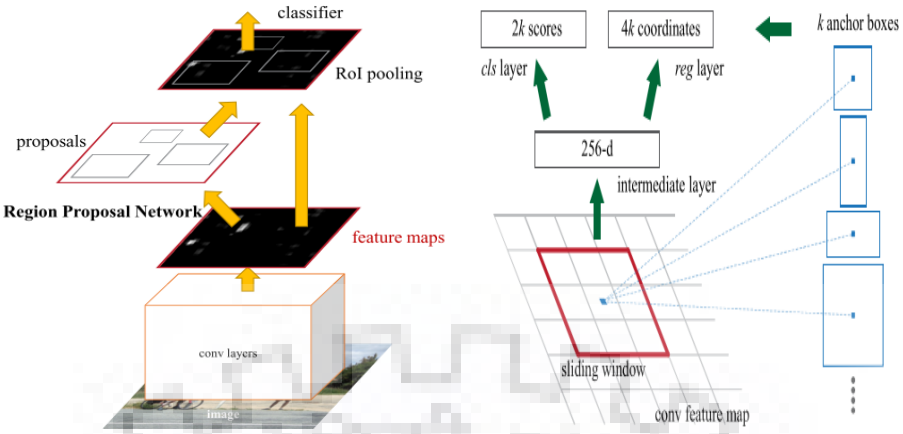
Following from the idea in [3], we will first extract part proposals from the images in the training set using the Region Proposal Network of Faster-RCNN [18]. Each of these proposals will have CNN feature of 4096 dimension associated with it. We then convert these feature dimensions into a transaction dataset to be used by the pattern mining algorithm - Apriori [25] to generate frequent patterns. These frequent patterns are nothing but the representative of the different parts of an object. To reduce the number of frequent patterns and to combine similar patterns into one, we apply KModes Clustering algorithm [26] and obtain a refined set of frequent patterns. We then use these frequent patterns to represent an image using the Bag of Patterns(BoP) representation [3]. After encoding all the images in the training set, we then train SVM classifier for multi-class classification. To classify a test image, we first transform it to the BoP representation and then apply the classifier to predict it's class. The detailed explanation of the steps is as follows:

4.1 Proposals and Features Extraction

4.1.1 Proposals from RPN

Our proposed approach makes use of Region Proposal Network(RPN) from the Faster-RCNN [18] to produce region proposals for an image. The reason for choosing this technique is that RPN helps us extract only those image patches that are more likely to contain the object within them rather than just the trivial patches containing the background. Also, the marginal cost of computing proposals is merely 10ms per image as compared to Selective Search which take 2 seconds per image.

RPN takes an image as input and generates a set of object proposals and their corresponding objectness score. As shown in Figure 4.1(a), the image is first passed through convolutional layers to generate a convolutional feature



a) Faster R-CNN b) Region Proposal Network

FIGURE 4.1: Faster R-CNN architecture with the RPN module and the RPN operation [18]

map. The region proposals are generated by sliding a small network over the feature map(shown in Figure 4.1(b). The network takes a $n \times n$ spatial window from the feature map. Each such window is mapped to a lower dimensional feature and this feature is given as input to two fully connected layers - box-regression layer(*reg*) and box-classification layer(*cls*). Multiple region proposals are made simultaneously at each sliding window location. The maximum number of proposals at each location can be k. The *reg* layer produces $4k$ outputs corresponding to the coordinates of k boxes and the *cls* layer produces $2k$ outputs corresponding to the scores that estimate the probability of object or not object for each proposal.The k proposals are generated relative to the k reference boxes called *anchors*. Each anchor is centered at the sliding window and has a specific scale and aspect ratio. If the size of the feature map is $W \times H$ then there will be a total of WHk anchors. The anchors are assign binary class label of being an object or not by considering their overlapping with the ground-truth box. The RPN is trained to minimize the loss function and get region proposals. Finally, RoI pooling is done to get feature map corresponding to each region proposal.

[3] takes 128×128 patches with a stride of 32 pixels from every image in the dataset. Using RPN will help generate useful and lesser number of proposals as compared to this sliding window approach. Also, RPN gives proposals of different scales and dimensions. However, we need to look at the time taken by RPN to generate proposals as compared to the sliding window approach. The number of proposals t generated matter a lot as they will correspond to the

number of transactions formed that will be fed to the association rule mining algorithm. And such algorithms run slowly as the number of transactions increases. So we need to get only really useful proposals so as to keep the number of transactions optimal.

4.1.2 CNN features

For each of the proposal extracted, we need to get it's feature representation. Every layer of the CNN produces some output which is given as an input to the next layer. The final layers in the CNN model are the fully connected layers that have full connections to all the activations in the previous layer. The output of the fully connected layer $fc6$ in faster RCNN is a $[1 \times 1 \times 4096]$ vector. This output can be used to represent the proposal. To reduce the number of dimensions, we only retain dimensions with K largest non-negative values. As shown in [3], these K largest dimensions are also equally representative of the image proposal as the original 4096 dimension without any significant loss of information.

4.2 Transactions creation

Creating a set of transactions is a very crucial task while working with pattern mining algorithms. It is mandatory for the transactions to fulfill certain requirements as noted in [3]:

1. Each transaction can only have a small number of items, as the potential search space grows exponentially with the number of items in each transaction.

2. What is recorded in a transaction must be a set of integers(which are typically the indices of items).

CNN features can help to fulfill these conditions based on it's properties discussed in the previous section. We need to design some encoding scheme so that the CNN feature representation of the proposed parts can be transformed into itemsets to create a transaction. We obtain a set of transaction with each transaction corresponding to a part proposal. To indicate whether this part belongs to the required object, we assign an additional item - *pos* or *neg* to the itemset. So now the itemset for each transaction will contain the feature encoding along with a *pos* or *neg*.

The proposals are represented by 4096 dimension CNN features with each dimension having a certain magnitude. So we take each 4096 dimensions of the CNN feature as items. We choose the top n features with largest magnitude to represent the proposal. We create the itemset taking the dimension indices (0-4095) of these top n features. We train the dataset using the one vs all technique i.e. for a certain class, we take all proposals from the images of that class as positive and the rest of the proposals as negative. To denote positive proposals, we append 4096 to the itemset of that proposals and for negative proposals, we append 4097. So the length of each complete transaction will be $n + 1$. For example, if $n = 3$ and the dimension indices of the three largest magnitude CNN feature of a part proposal belonging to target category is $[4, 253, 1190]$, then the final transaction set would be $[4, 253, 1190, 4096]$. In this way, we create the transaction database D for a particular class.

4.3 Frequent patterns generation

We perform association rule mining [25] , to find frequent patterns. Let us consider a set of K items $A = \{a_1, a_2, \ldots, a_K\}$, a transaction database D containing N transactions $D = \{T_1, T_2, \ldots, T_N\}$. Each transaction T is a collection of items in A such that $T \subseteq A$. Given a set P which is subset of A, we need to find fraction of transactions in D which contain P. This is known as the $support$ value of P.

$$supp(P) = \frac{|\{T|T \in D, P \subseteq T\}|}{N} \tag{1}$$

If this value is greater than a certain threshold min_supp, we call P as frequent itemset.
The confidence of an association rule $P \rightarrow i$ tells us how likely it is that item i is present in transactions which contain P within D.

$$\begin{aligned} conf(P \rightarrow i) &= \frac{supp(P \cup \{i\})}{supp(P)} \\ &= \frac{|\{T|T \in D, (P \cup \{i\}) \subseteq T\}|}{|\{T|T \in D, P \subseteq T\}|} \end{aligned} \tag{2}$$

For a pattern P to be called a frequent pattern, it must satisfy these two criteria:

$$supp(P) > min\_supp \tag{3}$$

$$conf(P \rightarrow pos) > min\_conf \tag{4}$$

[3] demonstrates how association rule mining implicitly satisfies the two requirements of essential parts discovered, i.e., representativeness and discriminativeness. Using the, Eq. (3) and Eq. (4), we can rewrite Eq. (2) as follows :

$$\begin{aligned} supp(P \cup \{i\}) &= supp(P) \times conf(P \rightarrow i) \\ &> min\_supp \times \min\_conf \end{aligned} \tag{5}$$

where $supp(P \cup \{i\})$ measures the fraction of pattern P found in transactions of the target category among all the transactions where i represents positive dimension. Therefore, having values of $supp(P)$ and $conf(P \rightarrow i)$ larger than their thresholds ensure that pattern P is found frequently in the target category,which shows representativeness. A high value of min_conf also ensures that pattern P is more likely to be found in the target category rather than any other class thus fulfilling the discriminativeness requirement. The output of this step is a collection of patterns P obtained for every class in the dataset.

4.4 Patterns Aggregation

The number of frequent patterns discovered by pattern mining algorithm can be very huge, with some patterns being highly correlated. This problem is known as pattern explosion. To overcome this, we need to combine the highly correlated patterns and only keep the major discriminative patterns that can help us with the classification task. For aggregating the patterns into a smaller set of discriminative patterns we use K-Modes clustering Huang [26]. K-Modes clustering algorithm is a variation of the well-known K-Means clustering algorithm that is adapted to suit categorical variable data. The function used to find dissimilarity between two objects calculates the total mismatches of the corresponding attribute categories of the two objects. The smaller the number of mismatches is, the more similar the two objects. To represent each cluster, it uses a frequency based method, where each attribute of the cluster center is calculated as the maximum occurring attribute among the cluster members that is nothing but the mode of that attribute.

Let X, Y be two categorical objects described by m categorical attributes. The dissimilarity measure between X and Y can be defined by the total mismatches of the corresponding attribute categories of the two objects. The smaller the number of mismatches is, the more similar the two objects. This measure is

often referred to as simple matching.

$$dist(X, Y) = \sum_{i=1}^{m} \delta(x_i, y_i) \tag{6}$$

where

$$\delta(x_i, y_i) = \begin{cases} 0 & x_i = y_i \\ 1 & x_i \neq y_i \end{cases}$$

The *mode* of a set is defined as follows:
Let X be set of categorical objects described by categorical attributes $A_1, A_2, \ldots A_m$.
A*mode* of $X = X_1, X_2 \ldots X_n$ is a vector $Q = [q_1, q_2 \ldots q_m]$ that minimizes

$$D(X, Q) = \sum_{i=1}^{n} dist(X_i, Q) \tag{7}$$

The K-Modes algorithm works as follows : Initially, random k modes are selected, one for each cluster. Allocate an object to cluster whose mode is nearest to it according to equation (6). Update the mode of the cluster after each allocation. After all the objects have been allocated to the clusters, retest the dissimilarity of the objects against the current modes. If an object is found such that its nearest mode belongs to another cluster rather than its current one, reallocate the object to that cluster and update the modes of both clusters.This is repeated until no object has changed its cluster in the cycle.

In our proposed approach, we have the set of frequent patterns P generated from the Apriori algorithm. We now represent each of the pattern in this set as a binary vector of 4096 dimension with only those bits set whose indices are present in the pattern. We run the K-Modes algorithm over the binarized patterns and obtain a set of cluster centroids as output. These cluster centroids can then be our final set of refined patterns P'. Choosing the parameter of number of cluster centroids is an important task. If the number of cluster centroids is too small, then the patterns won't be discriminative and if the number of cluster centroids is too large, it would increase the computational time in the next training step.

4.5 Image representation

To represent an image using a set of refined patterns P', we run RPN over the image to generate proposals at multiple scales and locations, and get their CNN activations. For each 4096-dimensional CNN activation vector of an image proposal, after finding C_i, the set of indices of dimensions that have non-zero values, we check for each selected pattern $P_k \in P'$ whether $P_k \subseteq C_i$. Thus, our Bag-of-Patterns representation $f_{BoP} \in R^{X \times Y}$ is a histogram encoding of the set of local CNN activations, satisfying $[f_{BoP}]_k = |i|P_k \in C_i|$. Our Bag-of-Patterns representation is similar to the well-known Bag-of-Visual-Words (BoW) representation [7] if one thinks of a pattern $P \in P'$ as one visual word. The difference is that in the BoW model one feature descriptor is assigned to one visual word, whereas in our BoP representation, one CNN feature can belong to multiple patterns.

4.6 Training and Classification

For training of the obtained data, an optimal choice of learning method would be Support Vector Machine (SVM) because SVM is effective in high dimensional spaces and it uses a subset of training points in the decision function (called support vectors), so it is also memory efficient. SVM works by finding a separating hyperplane to distinguish between objects belonging to two classes. For multi- class classification, "one-against-one" approach can be used. If n_class is the number of classes, then $n\_class \times (n\_class - 1)/2$ classifiers are constructed and each one trains data from two classes. The results from these classifiers are then aggregated to give the final output label.

Using the image representation discussed above, we encode all the images in the training set to create a mega histogram. In the mega histogram, each row represents an train image and each column represents the refined patterns obtained in the previous step. We feed this mega histogram along with the training labels for each sample to SVM. We hypertune the parameters C and $gamma$ and try different kernel functions such as - linear, polynomial, radial basis function so as to maximize the accuracy of the model. The parameter C, common to all SVM kernels, trades off misclassification of training examples against simplicity of the decision surface. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly. $gamma$ defines how much influence a single training example has. The larger $gamma$ is, the closer

other examples must be to be affected. Proper choice of *C* and gamma is critical to the SVM's performance.

To test the model, we first extract the part proposals using either the sliding window approach or the RPN. We get the CNN features for the parts. We then encode image as an histogram mapping the features to the found centroids and give it as an input to the model generated. The model then outputs a specific class label for that test image.

Chapter 5

Implementation

5.1 The Dataset

For experiments, we use Leeds Butterfly dataset [27]. It has 832 images of 10 different species(categories) of butterflies with 55-100 images per category. From the figure, we can see how the butterflies of different categories are visually similar looking. So, to distinguish between them, the machine needs to look at the various parts of the butterflies such as the shape of the wing, the pattern over the wings and so on. Our approach is likely to derive these unique part features from the images and train the classifier to distinguish among those.

FIGURE 5.1: Different butterfly species from Leeds Butterfly dataset

5.2 Extracting proposals and their CNN features

In this step, we first resize the image such that it's smaller dimension is 256 and the aspect ratio of the original image is maintained. We extract part proposals from the images in two ways. First way is the sliding window approach used in the [3]. We extract 128×128 size with a stride of 32 from every image. On an average, 37 proposals are extracted from each image. The second way is to run the RPN over these images to get region proposals. We use the Python implementation of Faster RCNN [18]. We keep the number of proposals generated from each image t as 37. The figure shows the proposals generated with both ways.

a) Without RPN

b) With RPN

FIGURE 5.2: Two different techniques for extracting part proposals from images

To extract CNN features from the image proposals, we use the state-of-the-art CNN models pre-trained on the *ImageNet* dataset [28]. For the sliding window approach, we use the BVLC Reference $CaffeNet$ [29] with five convolution layers followed by two 4096-dimensional and one 1000-dimensional fully-connected layers. This architecture is similar to *AlexNet* [11]. For the RPN, we use the pre-trained $VGG - VD$ model which has 19 layers [13]. We use the features from the output of layer $fc6$ of 4096 dimension in both the cases. The tradeoff here is that sliding window approach is faster than the RPN, but the proposals extracted by RPN focus more on the object rather than the background.

5.3 Applying Apriori Algorithm

For the CNN features extracted for each part proposal, we select the top-10 feature indices with largest values. To create the transaction database D for a particular class, we append 4096(positive) to proposals belonging to images of that class and 4097(negative) to all the others. We then run the Apriori Algorithm implementation from [30] over the transactions in database D while setting minimum support(min_supp) to 0.01% and minimum confidence(min_conf) to 80%. We derive the rules from the algorithm such item 4096 is in the consequent as 4096 represents positive proposals for that class. In the figure 5.3, the values in the bracket denote the $support$ and $confidence$ for that rule respectively. We then retain only the antecedent part of the association rules as our set P of frequent patterns for the class.

```
3063 269 3357 1267 2780 1075 2075 3265 1286 479 4096
2473 1943 3153 2161 1785 67 4047 3744 870 3859 4096
1376 989 1943 1391 34 726 2473 1785 3489 3514 4096
```

Transactions

```
4096 <- 1726 315 680 (0.0422805, 92.3077)
```

Apriori Rules

FIGURE 5.3: Input and output format of the association rule mining Apriori Algorithm

5.4 Applying K-Modes Clustering Algorithm

To aggregate the frequent patterns set P created by Apriori algorithm, we use the K-Modes clustering implementation from [26]. We take the frequent patterns generated for a class and give it as input to the K-Modes algorithm. We set the number of clusters to be generated as 50 and set the *n_init* variable in K-Modes to 5. The *n_init* variable specifies that the algorithm with initialize with random cluster centroids this many times and output the best clustering among them. So, for every class we get 50 refined patterns and the combined patterns from all 10 classes will $50 \times 10 = 500$. We try to map these refined patterns to the part proposals. As we can see in figure 5.4 and 5.5, the proposals from different images which contain the same pattern are visually similar. These set of proposals containing a certain pattern are called mid-level visual elements [3].

FIGURE 5.4: Mapping the obtained refined patterns to the part proposals (without RPN)

FIGURE 5.5: Mapping the obtained refined patterns to the part proposals (with RPN)

5.5 Bag of Patterns representation

As discussed in the previous section, we create BoP representation for every image in the training dataset and create a mega histogram for training purpose. The figure 5.6 shows BoP representation for image from every class.

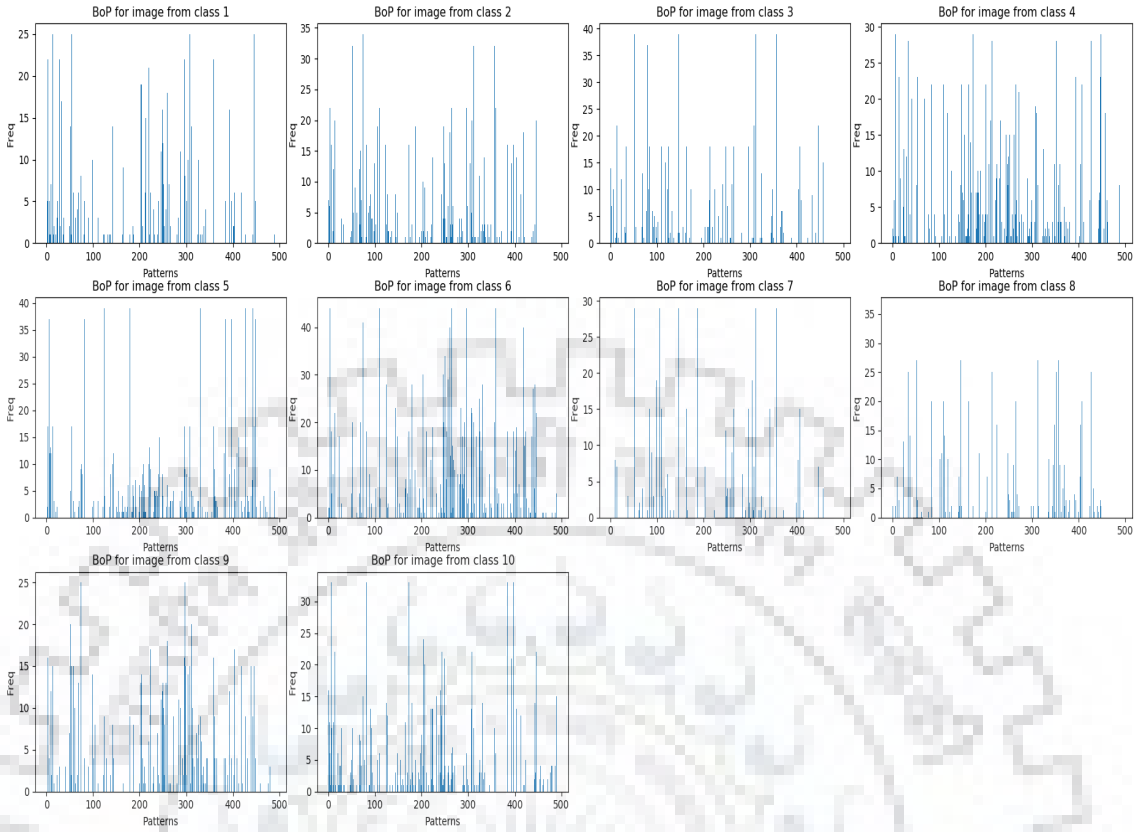


FIGURE 5.6: Bag of Patterns representation

5.6 SVM training and classification

For training purposes, we make use of the python $scikit-learn$ module's SVM - $C-SupportVectorClassification(SVC)$. The parameters C and $gamma$ are very crucial for optimal training. Deciding on these parameters can be done by using the $GridSearchCV$ function of the $scikit-learn$ module. $GridSearchCV$ performs an exhaustive search over the specified parameters range for the estimator using the estimator's scoring function to decide the best estimator parameters.

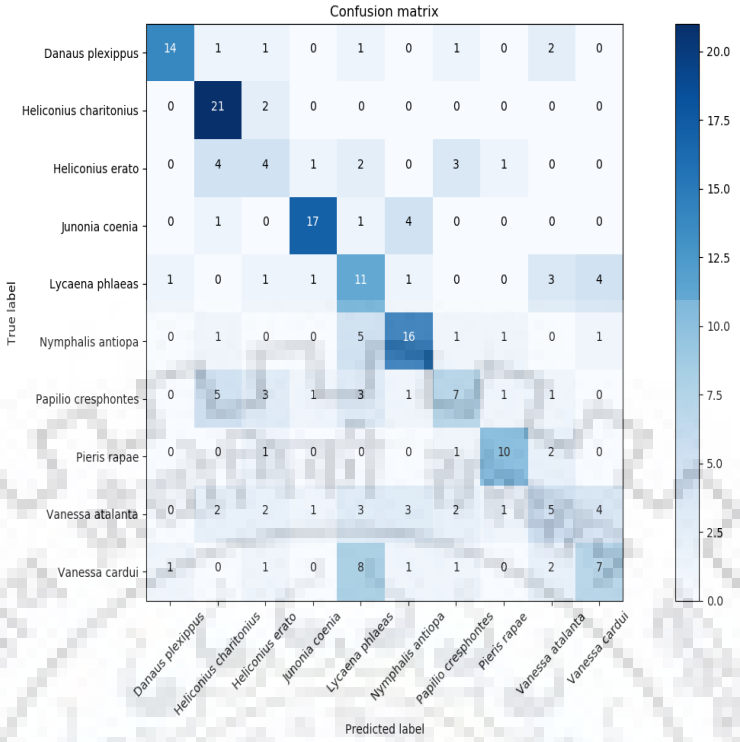
We first get the mega histogram and the corresponding training labels for every row in the mega histogram. We then split the data into train and test images. We standardize the training data by removing the mean and scaling to unit variance. We apply the same transformation on the test data before testing it as well. We specify a range of values in logspace 10 for both C and $gamma$. We also supply a cross-validation function with 5 folds to the $GridSearchCV$ object. The output of the $GridSearchCV$ is the best estimator which we then run on the test data to find the accuracy of the model.

The accuracy of the model is calculated as the number of images correctly classified according to their classes out of the total number of test images. For, multi-class classification in SVM, a better measure of the model is the per class accuracy. The per class accuracy is calculated as follows -
$TP = Truepositive$ i.e. the number of images whose predicted label and true label is same for the given class
$FP = Falsepositive$ i.e. the number of images whose predicted label is the given class but the true label is some other class.
$P = TP + FP$ Total number of images classified as belonging to this class i.e. their predicted label is the given class.

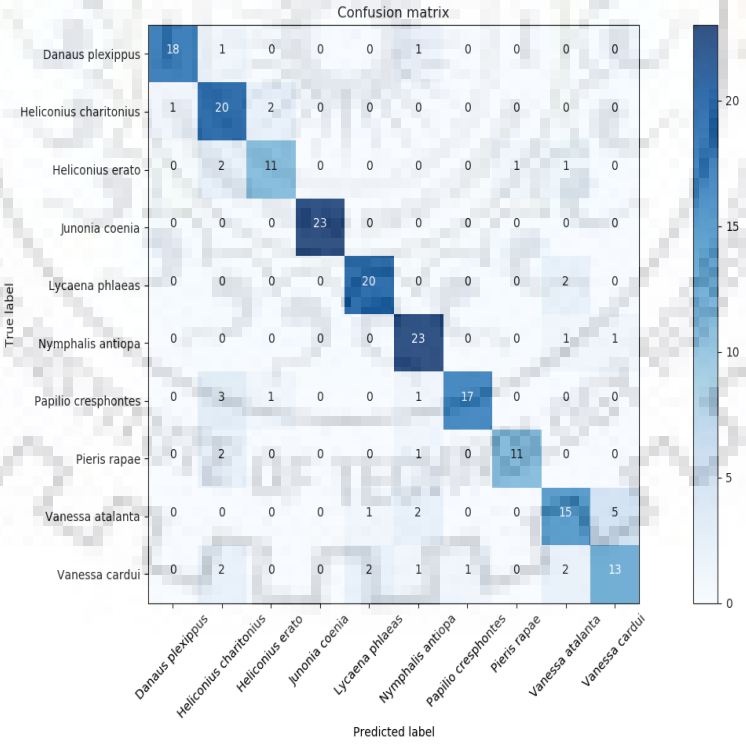
$$Accuracyperclass = \frac{TP}{TP + FP}$$

The model obtained with the sliding window approach has *C*: 13.11, *gamma* : 0.00025 and the model accuracy of 0.54. The model obtained with the sliding window approach has *C*: 100.0, *gamma* : 0.001 and the model accuracy of 0.82. The table shows the per class accuracy for both the models.

Accuracy	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9	Class 10
Without RPN	0.88	0.60	0.27	0.81	0.32	0.62	0.44	0.71	0.33	0.44
With RPN	0.95	0.67	0.79	1.00	0.87	0.79	0.94	0.92	0.71	0.68



a) Without RPN



b) With RPN

FIGURE 5.7: Confusion matrix for the SVM models

Chapter 6

Conclusion

Part-based techniques for fine-grained classification seem to be a promising approach. Moreover, parts are what distinguish objects belonging to same broader category but different fine-grained categories. Using RPN for part proposals which is the fastest technique available as of now for region proposals, we can expect our approach to be time-efficient. However, deciding on the optimal number of proposals to be generated from each image is crucial so as to maintain an optimal number of transactions for the association rule mining algorithm as the time taken by the algorithm increases proportionally to the size of the transaction database. As can be seen from the accuracy of the models generated from the sliding window approach and the RPN approach, clearly the RPN approach has an edge over the other technique thus promising us better results.

Using association rule mining also seems promising. One of the challenges faced while using the pattern mining approach in the classification task is the designing an optimal feature encoding for encoding part representations as itemset because as the number of items in a transaction increase, the complexity of the algorithm increases exponentially. CNN features seem to be an apt choice for this purpose as they clearly define the part proposals using just the 4096 features. Other tasks at hand while working with association rule mining techniques is deciding upon the parameters such as the minimum support for the itemsets to be called frequent and the minimum confidence to get strong association rules.

Application of K-Modes algorithm for aggregating the association rules generated by the association rule mining algorithm seems a very crucial task as it helps to combine the highly correlated rules and give us a refined set of rules thereby reducing the computational time in the training step.

Chapter 7

Future Work

Application of pattern mining approach to the image classification task seems give promising results. This can be further extended to be used on videos to detect the fine-grained categories of objects within them. Weighted Association Rule Mining by Tao, Murtagh, and Farid [31] is a modification over the traditional algorithms in association rule mining that allows a certain weight to be assigned to every item in the transaction database. We can use this weighted algorithm and assign higher weights to items corresponding to features present in positive proposals and then derive more significant rules. Currently, we are using the K-Modes clustering technique to aggregate the rules and create a smaller set of rules. We can experiment to find better algorithms to reduce the size of the patterns set. For training purposes, we have used SVM for multi-class classification. We can find if a better alternative exists to train our dataset and create a stronger classifier.

Bibliography

[1] P Welinder, S Branson, T Mita, C Wah, F Schroff, S Belongie, and P Perona, "Caltech-UCSD Birds 200", California Institute of Technology, Tech. Rep. CNS-TR-2010-001, 2010.

[2] Y. Li, L. Liu, C. Shen, and A. v. d. Hengel, "Mid-level Deep Pattern Mining", 2014. DOI: 10.1109/CVPR.2015.7298699. [Online]. Available: http://arxiv.org/abs/1411.6382%0Ahttp://dx.doi.org/10.1109/CVPR.2015.7298699.

[3] Y. Li, L. Liu, C. Shen, and A. V. D. Hengel, "Mining Mid-level Visual Patterns with Deep CNN Activations", *International Journal of Computer Vision*, vol. 121, no. 3, pp. 1–21, 2016, ISSN: 15731405. DOI: 10.1007/s11263-016-0945-y.

[4] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004, ISSN: 1573-1405. DOI: 10.1023/B:VISI.0000029664.99615.94. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. I, no. 3, pp. 886–893, 2005, ISSN: 1063-6919. DOI: 10.1109/CVPR.2005.177. [Online]. Available: http://eprints.pascal-network.org/archive/00000802/.

[6] H. Bay, T. Tuytelaars, L. Van Gool, A. Leonardis, H. Bischof, and A. Pinz, "SURF: Speeded Up Robust Features", *Computer Vision ECCV 2006*, vol. 3951, pp. 404–417, 2006, ISSN: 03029743. DOI: 10.1007/11744023{_}32. [Online]. Available: citeulike-article-id:2708013%0Ahttp://dx.doi.org/10.1007/11744023_32.

[7] J. Yang, "Evaluating Bag-of-Visual-Words Representations in Scene Classification", pp. 197–206, 2007.

[8] F Perronnin and C Dance, "Fisher Kernels on Visual Vocabularies for Image Categorizaton", *Proc. {CVPR}*, 2006.

[9] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3304–3311, 2010, ISSN: 10636919. DOI: 10.1109/CVPR.2010.5540039.

[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge", *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015, ISSN: 15731405. DOI: 10.1007/s11263-015-0816-y. [Online]. Available: http://dx.doi.org/10.1007/s11263-015-0816-y.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *Advances In Neural Information Processing Systems*, pp. 1–9, 2012, ISSN: 10495258. DOI: http://dx.doi.org/10.1016/j.protcy.2014.09.007.

[12] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks arXiv:1311.2901v3 [cs.CV] 28 Nov 2013", *Computer Vision–ECCV 2014*, vol. 8689, pp. 818–833, 2014, ISSN: 978-3-319-10589-5. DOI: 10.1007/978-3-319-10590-1{_}53. [Online]. Available: http://link.springer.com/10.1007/978-3-319-10590-1_53%5Cnhttp://arxiv.org/abs/1311.2901%5Cnpapers3://publication/uuid/44feb4b1-873a-4443-8baa-1730ecd16291.

[13] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *International Conference on Learning Representations (ICRL)*, pp. 1–14, 2015, ISSN: 09505849. DOI: 10.1016/j.infsof.2008.09.005. [Online]. Available: http://arxiv.org/abs/1409.1556.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 1–9, 2015, ISSN: 10636919. DOI: 10.1109/CVPR.2015.7298594.

[15] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition", *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013, ISSN: 09205691. DOI: 10.1007/s11263-013-0620-5.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014, ISSN: 10636919. DOI: 10.1109/CVPR.2014.81.

[17] R. Girshick, "Fast R-CNN", 2015, ISSN: 978-1-4673-8391-2. DOI: 10.1109/ICCV.2015.169. [Online]. Available: http://arxiv.org/abs/1504.08083.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", *Nips*, pp. 1–10, 2015, ISSN: 01689002. DOI: 10.1016/j.nima.2015.05.028.

[19] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for Fine-grained Category Detection", 2014, ISSN: 16113349. DOI: 10.1007/978-3-319-10590-1{_}54. [Online]. Available: http://arxiv.org/abs/1407.3867.

[20] J. Krause, T. Gebru, J. Deng, L. J. Li, and F. F. Li, "Learning features and parts for fine-grained recognition", *Proceedings - International Conference on Pattern Recognition*, pp. 26–33, 2014, ISSN: 10514651. DOI: 10.1109/ICPR.2014.15.

[21] J. Krause, H. Jin, J. Yang, and F. F. Li, "Fine-grained recognition without part annotations", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 5546–5555, 2015, ISSN: 10636919. DOI: 10.1109/CVPR.2015.7299194.

[22] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 923–930, 2013, ISSN: 10636919. DOI: 10.1109/CVPR.2013.124.

[23] G. Sharma, F. Jurie, and C. Schmid, "Expanded Parts Model for Semantic Description of Humans in Still Images", *Pami*, pp. 1–14, 2015, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2537325. [Online]. Available: http://arxiv.org/abs/1509.04186.

[24] P. Mettes, J. C. van Gemert, and C. G. M. Snoek, "No spare parts: Sharing part detectors for image categorization", *Computer Vision and Image Understanding*, vol. 152, pp. 131–141, 2016, ISSN: 1090235X. DOI: 10.1016/j.cviu.2016.07.008. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2016.07.008.

[25] R Agrawal and R Srikant, "Fast Algorithms for Mining Association Rules", *Organization*, vol. 1215, no. 1558601538, 487–499, 1994, ISSN: 15426270. DOI: 10.1.1.40.6757. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.40.7506&rep=rep1&type=pdf.

[26] Z. Huang, "Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values", *Data Mining and Knowledge Discovery*, vol. 2, no. 3, pp. 283–304, 1998, ISSN: 1573-756X. DOI: 10.1023/A:1009769707641. [Online]. Available: http://link.springer.com/article/10.1023/A:1009769707641.

[27] J. Wang, K. Markert, and M. Everingham, "Learning Models for Object Recognition from Natural Language Descriptions", *Procedings of the British Machine Vision Conference 2009*, pp. 1–2, 2009. DOI: 10.5244/C.23.2. [Online]. Available: http://www.bmva.org/bmvc/2009/Papers/Paper106/Paper106.html.

[28] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database", in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255, ISBN: 978-1-4244-3992-8. DOI: 10.1109/CVPRW.2009.5206848. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5206848.

[29] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding", 2014, ISSN: 10636919. DOI: 10.1145/2647868.2654889. [Online]. Available: http://arxiv.org/abs/1408.5093.

[30] C. Borgelt, "Frequent item set mining", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 6, pp. 437–456, 2012, ISSN: 19424787. DOI: 10.1002/widm.1074.

[31] F. Tao, F. Murtagh, and M. Farid, "Weighted Association Rule Mining using weighted support and significance framework", in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '03*, 2003, p. 661, ISBN: 1581137370. DOI: 10.1145/956750.956836. [Online]. Available: http://portal.acm.org/citation.cfm?doid=956750.956836.

Discriminative Parts Discovery for fine-grained object categorization

ORIGINALITY REPORT

8%	5%	7%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Yao Li, Lingqiao Liu, Chunhua Shen, Anton van den Hengel. "Mining Mid-level Visual Patterns with Deep CNN Activations", International Journal of Computer Vision, 2016 Publication	4%
2	scikit-learn.org Internet Source	1%
3	Zengyou He. "Improving K-Modes Algorithm Considering Frequencies of Attribute Values in Mode", Lecture Notes in Computer Science, 2005 Publication	1%
4	Submitted to Loughborough University Student Paper	1%
5	www.cv-foundation.org Internet Source	1%
6	www.cs.princeton.edu Internet Source	1%