

**Automatic Sentence Generation for Images  
via Key-phrase Estimation  
using Large-Scale Captioned Images**  
(大規模な説明文つき画像を用いた  
キーフレーズ推定に基づく画像説明文の自動生成)

牛久 祥孝

This thesis is dedicated to my late maternal grandfather, without  
whose interesting talks about science and technology  
I would not have written a Ph.D. thesis.

## Acknowledgements

My heartfelt appreciation is extended to Prof. Tatsuya Harada, who has supervised my research for six years. He has been extraordinarily tolerant and supportive of my work. I have invariably been inspired and motivated by his vision and strategy for research. The opportunity to study under his distinguished direction has been a blessing.

I would particularly like to thank Prof. Yasuo Kuniyoshi, Director of the Laboratory for Intelligent Systems and Informatics (ISI). In April 2013, Prof. Harada set up the Machine Intelligence Laboratory (MIL). Before the establishment of MIL, he and I worked in the ISI Laboratory. Prof. Kuniyoshi has advised me not only about my work, but also philosophy I should have. His great insight and philosophy have been always helpful, suggesting future directions for my efforts. I am proud of working in the ISI Laboratory for five years.

I would also like to express my gratitude to Prof. Kiyoharu Aizawa, Prof. Masayuki Inaba, and Prof. Yoshihiko Nakamura for their many constructive comments and discussions, which have been an invaluable help in finishing my work.

The labmates in ISI and MIL gave me useful comments and persistent help. Especially, advice and comments provided by Prof. Hideki Nakayama and Dr. Asako Kanazaki have been of great help in my Ph.D. work. Although I am only a Ph.D. student at MIL, I have many friends who are Ph.D. students at other laboratories and at other universities. I received wide-ranging discussion and warm encouragement from them using social networking services.

I owe an extremely important debt to my family for their moral and financial support over these many years. Finally, I offer my special thanks and love to Maika. Without her encouragement, this thesis would not have materialized.

## Abstract

With advances in information technology, the explosively increasing multimedia environment has created demand for methods by which computers can interpret multimedia contents on behalf of humans. By virtue of such methods, we can use multimedia efficiently. For example, we can seek desired data from vast amounts of multimedia information and understand their contents without viewing them. Such a method would facilitate the use of robots in the real world and help visually impaired people in the future.

Generic object recognition is a problem to ascertain the contents of general images and videos. In the last decade, many researchers have attacked this problem. After early studies using a small-scale dataset to learn a few categories, generic object recognition has been divided into several problems: (a) classifying not images but videos, (b) classifying images using large-scale datasets, and (c) associating images with more than one label (annotation). For the past few years, moreover, studies to explain multimedia not with several labels but with a sentence including the relations among such labels have begun to attract much attention.

Generating sentences to explain images and videos is the ultimate goal of generic object recognition. Most methods examined in earlier studies require semantic knowledge such as ⟨object, action, scene⟩. Such labels with attributes should be labeled manually. Therefore, collecting a large-scale dataset is difficult.

In this paper, we develop a methodology for sentence generation using a dataset consisting only of pairs of an image and sentences because numerous images or videos are associated with captions in general web sites and sharing sites. To realize sentence generation using only images and sentences, we examine a hypothesis: “Almost all contents of an image are identifiable with a few descriptive phrases (keyphrases). A sentential caption can be generated by connecting these with an experimental grammar model”. Therefore, we present the Multi-keyphrase Problem to estimate multiple keyphrases and to generate a sentence.

Because keyphrases are combinations of various objects and events, larger datasets are preferred. Scalability of the data amount is necessary, as is accuracy of keyphrase estimation.

Consequently, we first specifically investigate large-scale visual classification. To learn labels using a large-scale dataset, online learning by which each datum is loaded and learned one-by-one is useful. Nevertheless, no investigation or comparison of online learning methods for visual recognition is reported in the literature. As described in this paper, we select and fix state-of-the-art features and evaluate online algorithms.

Next, we propose a method to generate a sentence by combining estimated keyphrases using an experimental grammar model. Our pilot study, conducted using only pairs of an image and sentences, demonstrates that sentences can be generated without semantic knowledge, which is difficult to collect manually. We also modify an existing online learning method to annotate images according to results obtained from our investigations of state-of-the-art methods. In existing works for image annotation, combinations of metric learning and non-parametric approaches are mainstream. However, scalability remains an open question for non-parametric approaches. The proposed method achieves state-of-the-art performance and superior scalability for image annotation problems.

As described above, the number of keyphrases is much larger than that of single words. To train many classifiers efficiently, we use subspace learning to reduce the number of parameters for classifiers. In the subspace, (a) all feature vectors associated with the same label should be mapped as mutually close. Moreover, (b) classifiers for each label should be learned in the subspace. Therefore, we propose a novel learning method: Common Subspace for Model and Similarity (CoSMoS).

Finally, we evaluated our methodology using three datasets, each consisting of pairs of an image and a sentence. Experimental results show that our system is more accurate than those presented in earlier works. The scalability of our system and experimentally obtained results show that the accuracy increases when the dataset increases.

# Contents

<b>Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Objective . . . . .	3
1.3 Structure of the Thesis . . . . .	4
<b>2 Methods to Describe Multimedia with Natural Language</b>	<b>6</b>
2.1 Generic Object Recognition . . . . .	6
2.1.1 Datasets . . . . .	8
2.1.2 Feature extraction . . . . .	9
2.1.3 Classification and Annotation . . . . .	10
2.2 Challenge to Associate Multimedia with a Caption . . . . .	11
2.3 Multi-Keyphrase Approach for Sentential Description . . . . .	13
<b>3 Investigation of Online Learning Methods for Multiclass Classification</b>	<b>16</b>
3.1 Necessity to Investigate Existing Algorithms . . . . .	16
3.2 Related Works for Large-scale Classification . . . . .	18
3.3 Online Learning Algorithms . . . . .	19
3.3.1 Perceptron . . . . .	20
3.3.2 Stochastic Gradient Descent SVM . . . . .	20
3.3.3 Passive–Aggressive . . . . .	22
3.3.4 Confidence-Weighted . . . . .	22
3.3.5 Adaptive Regularization of Weight . . . . .	23
3.3.6 Gaussian Herding . . . . .	23
3.3.7 Soft Confidence-Weighted . . . . .	24

3.4	Common Qualitative Issues . . . . .	26
3.4.1	OVR vs. MUL . . . . .	26
3.4.2	Averaging . . . . .	26
3.5	Experiments on ImageNet . . . . .	27
3.5.1	ILSVRC 2010 Dataset . . . . .	27
3.5.2	ILSVRC 2012 Dataset . . . . .	27
3.5.3	Result 1: Accuracies of not Averaged Classifiers . . . . .	28
3.5.4	Result 2: Averaging Does Boost All . . . . .	29
3.5.5	Result 3: MUL vs. OVRs . . . . .	34
3.5.6	Summarized Guidelines . . . . .	37
<b>4</b>	<b>Multi-Keyphrase Problem and Sentence Generation</b>	<b>38</b>
4.1	Multi-Keyphrase Estimation as an Annotation Problem . . . . .	38
4.2	Related Methods for Annotation . . . . .	40
4.3	Sentence Generation with Multi-keyphrase Estimation . . . . .	42
4.3.1	Multi-keyphrase Extraction . . . . .	42
4.3.2	Methods of Multi-keyphrase Learning . . . . .	42
4.3.3	From Multi-keyphrases to a Sentence . . . . .	45
4.4	Evaluation of Multi-keyphrase Estimation . . . . .	46
4.4.1	Experiment Setting . . . . .	46
4.4.2	Experiment Result on Benchmark Datasets . . . . .	47
4.5	Experiment and Discussion of Sentence Generation . . . . .	49
4.5.1	Examples of Generated Sentences . . . . .	52
4.5.2	Automatic Evaluation . . . . .	52
4.5.3	Discussion about Keyphrase Extraction . . . . .	54
4.6	Summary of Proposed Approach for Sentential Description . . . . .	58
<b>5</b>	<b>CoSMoS: Common Subspace for Model and Similarity</b>	<b>59</b>
5.1	Subspace for Image Recognition . . . . .	59
5.2	Related Subspace Learning Methods . . . . .	62
5.3	Methodology . . . . .	65
5.3.1	Classification Rule and Objective Function . . . . .	66
5.4	Experiments for Image Annotation . . . . .	67
5.4.1	Dataset . . . . .	67
5.4.2	Comparison to State-of-the-art Methods . . . . .	68
5.4.3	Comparison to Subspace Learning Methods . . . . .	69
5.5	Towards Sentential Description for Images . . . . .	72
<b>6</b>	<b>Evaluation of Sentential Description for Images</b>	<b>73</b>
6.1	PASCAL Sentence . . . . .	75
6.2	IAPR-TC12 . . . . .	79

## CONTENTS

---

6.3	SBU . . . . .	79
6.4	Discussion . . . . .	81
6.4.1	Evaluating human-generated sentences . . . . .	81
6.4.2	Oracle keyphrase estimation . . . . .	83
6.4.3	Describing PASCAL Sentence using SBU . . . . .	85
<b>7</b>	<b>Conclusion and Future Work</b>	<b>89</b>
7.1	Conclusion . . . . .	89
7.2	Unsolved Problems and Future Works . . . . .	91
	<b>References</b>	<b>94</b>
	<b>Publications</b>	<b>106</b>



# List of Figures

1.1	Two images might have the same labels, “white”, “blue”, “sky” and “airplane”. However, the relations among these labels are ignored when using only these labels. . . . .	2
2.1	Variations of image understanding. . . . .	8
2.2	Methodology overview of [1]. . . . .	13
2.3	Pipeline of the proposed framework. We first estimate some <i>keyphrases</i> from an input image. Then a sentential caption is generated by spinning them with an experimental grammar model. . . . .	14
3.1	Overview of learning binary classifiers. . . . .	20
3.2	Overview of learning multiclass classifiers. . . . .	21
3.3	Comparison using ILSVRC 2010 1.2M dataset with SIFT+FV. White bars show performance using a 100k subset of ILSVRC 2010. . . . .	29
3.4	Comparison using ILSVRC 2012 subset. Each bar represents the mean accuracy among mid-level features from four descriptors. For example, the accuracy using FVs is the mean of the accuracies using SIFT+FV, LBP+FV, GIST+FV and CSIFT+FV. . . . .	29
3.5	Comparison using ILSVRC 2010 1.2M dataset with SIFT+FV. The darker bar for each algorithm shows the accuracy with averaging. The brighter one shows the accuracy without averaging for easy reference. . . . .	30
3.6	Comparison using ILSVRC 2012 subset. Each bar represents the mean accuracy among mid-level features from four descriptors. . . . .	30
3.7	Comparison about averaging. . . . .	31
3.8	Comparison of MUL and OVRs. Dashed lines and solid lines respectively show u-OVR and MUL. . . . .	34
3.9	Convergence speeds of online learning methods. Solid lines represent accuracies of averaged weights. Dotted lines represent accuracies of non-averaged weights. w-OVR (x) signifies that negative samples are selected randomly x times as often as positive samples. . . . .	35

## LIST OF FIGURES

---

3.10	Three guidelines for online learning for large-scale visual recognition.	37
4.1	Comparison of hinge-loss and averaged pairwise loss. . . . .	45
4.2	Comparison between PAAPL and TagProp with lower dimensional FVs. . . . .	48
4.3	Convergence comparison conducted among online learning methods. . . . .	49
4.4	Examples of estimated keyphrases and generated sentences. The first row depicts successful examples. The second row have partly correct examples thanks to appropriate keyphrases. The last row includes humorous mistakes. . . . .	53
4.5	Frequencies of the word gaps between grammatically related words.	54
4.6	Accuracy of generated sentences with each filter for keyphrase extraction. . . . .	57
5.1	Simple overview of subspace learning. Now we would like to obtain one-dimensional subspace (black line) given training samples in two-dimensional feature space. The blue line orthogonal to each subspace is the decision plane between the green triangle class and purple rectangle class. Two crossed circles are the mean of each class. . . . .	61
5.2	Comparison among CoSMoS variations using 16-dimensional subspace. . . . .	71
5.3	Comparison among CoSMoS variations using 1024-dimensional subspace. . . . .	71
6.1	Qualitative comparison. A common input image is shown in the upper left. We compare our result with Corpus-Guided [2], Midge [3], and BabyTalk [4]. . . . .	75
6.2	Good examples of estimated keyphrases and generated sentences for the PASCAL Sentence dataset. The first column shows input images. The second column shows estimated keyphrases for each input image. The third column shows the generated sentence at the top and ground truth in the dataset at the bottom. Red-colored words in generated sentences derive from estimated keyphrases. . . . .	77

## LIST OF FIGURES

---

6.3	Partially incorrect examples of estimated keyphrases and generated sentences for the PASCAL Sentence dataset. The first column shows input images. The second column has estimated keyphrases for each input image. The third column shows the generated sentence at the top and ground truth in the dataset at the bottom. Red-colored words in generated sentences come from estimated keyphrases. . . . .	78
6.4	Examples of estimated keyphrases and generated sentences for the IAPR-TC12 dataset. The first column presents input images. The second column has estimated keyphrases for each input image. The third column shows the generated sentence at the top and ground truth in the dataset at the bottom. Red-colored words in generated sentences come from estimated keyphrases. Three images in the top are regarded as correct examples, whereas the other two images in the bottom are regarded as incorrect. . . . .	80
6.5	Examples of estimated keyphrases and generated sentences for the SBU dataset. The first column shows input images. The second column has estimated keyphrases for each input image. The third column shows the generated sentence at the top and ground truth in the dataset at the bottom. Red-colored words in generated sentences derive from estimated keyphrases. Three images at the top are regarded as correct examples, although the other two images at the bottom are regarded as incorrect. . . . .	82
6.6	Rough illustration of evaluation for sentence generation from oracle keyphrases. . . . .	84
6.7	Comparison of two sentences generated by learning different datasets. The one at the top is generated after learning SBU dataset consisting of 1M web images, the other at the bottom is generated after learning PASCAL Sentence consisting of 1K well-organized images. . . . .	85
6.8	Examples of sentences generated for PASCAL Sentence images using a varying number of SBU datasets. . . . .	86
6.9	Examples of sentences generated for SBU images using a varying number of SBU datasets. . . . .	87
6.10	Impact of the dataset size evaluated using the BLEU score. . . . .	88
6.11	Impact of the dataset size evaluated using the NIST score. . . . .	88

# List of Tables

3.1	Update rules. Variables in Figure 3.1 and Figure 3.2 are shown in the middle columns. The last column shows the parameters to be tuned. For binary classification, $\gamma_t = y_t(\boldsymbol{\mu}_t \cdot \mathbf{x}_t)$ , $v_t = \mathbf{x}_t^\top \Sigma_t \mathbf{x}_t$ , and $l(\mathbf{x}_t)^2 = \ \mathbf{x}_t\ ^2$ . For multiclass classification, $\gamma_t = \boldsymbol{\mu}_t^{y_t} \cdot \mathbf{x}_t - \boldsymbol{\mu}_t^{y'_t} \cdot \mathbf{x}_t$ , $v_t = \mathbf{x}_t^\top (\Sigma_t^{y_t} + \Sigma_t^{y'_t}) \mathbf{x}_t$ , and $l(\mathbf{x}_t)^2 = 2\ \mathbf{x}_t\ ^2$ . Whether the classification is binary or multiclass, $\phi = \Phi^{-1}(\eta)$ ( $\Phi$ is the cumulative function of the normal distribution), $\psi = 1 + \phi^2/2$ , and $\zeta = 1 + \phi^2$ . . . . .	25
3.2	Accuracy(%) comparison of ILSVRC2010 using SIFT+FV. . . . .	32
3.3	Accuracy (%) comparison on ILSVRC 2012 <i>without averaging</i> . The best accuracy is highlighted for each mid-level feature. . . . .	32
3.4	Accuracy (%) comparison on ILSVRC 2012 <i>with averaging</i> . The best accuracy for each mid-level feature is highlighted. . . . .	33
3.5	Accuracy(%) comparison of ILSVRC2012 using SIFT+FV <i>without averaging</i> . MUL and three OVR manners are described in Section 3.4.1. For w-OVR, here, we extracted 32 times as many negative samples as positive samples. . . . .	36
3.6	Accuracy(%) comparison of ILSVRC2012 using SIFT+FV <i>with averaging</i> . MUL and three OVR are described in Section 3.4.1. For w-OVR, here, we extracted 32 times as many negative samples as positive samples. . . . .	36
4.1	Comparison of performance in terms of <b>P</b> , <b>R</b> , and <b>F</b> on Corel 5K. . . . .	48
4.2	Comparison of performance in terms of <b>P</b> , <b>R</b> , and <b>F</b> on ESP Game and IAPR-TC12. . . . .	50
4.3	Statistics of all phrases and filtered phrases with local TF-IDF. The upper side of each cell presents the number of training samples including at least one keyphrase. The lower side presents the number of keyphrases. . . . .	50

## LIST OF TABLES

---

4.4	Statistics of filtered phrases with global DF. The upper side of each cell presents the number of training samples include at least one keyphrase. The lower side presents the number of keyphrases. . . . .	51
4.5	Accuracy of multi-keyphrase estimation and of generated sentence for each method. . . . .	51
4.6	Frequent grammatical relations in two continuous words. “Freq.” shows the frequency of each relation in two continuous words against “Total”, which is the number of all occurrences of each relation. The definition of “Relation” is described in [5]. . . . .	55
5.1	Comparison of annotation performances among state-of-the-art methods. . . . .	69
5.2	Comparison of annotation performances among subspace learning methods produced using Corel 5k. . . . .	70
6.1	Statistics of datasets. IAPR-TC has 20,000 images, but 37 images are not associated with sentences. For PASCAL Sentence, we extract not only keyphrases with SWF but also keyphrases as described in Chapter 4. . . . .	74
6.2	Automatic evaluation for output sentences using PASCAL Sentence dataset. Scores in parentheses are computed by matching synonyms. . . . .	76
6.3	Automatic evaluation for output sentences using IAPR-TC12 dataset. Scores in parentheses are computed by matching synonyms. . . . .	79
6.4	Automatic evaluation for output sentences using SBU dataset. Scores in parentheses are computed by matching synonyms. . . . .	81
6.5	Automatic evaluation for human-generated sentences using a PASCAL Sentence dataset. Scores in parentheses are computed by matching synonyms. . . . .	83
6.6	Automatic evaluation for output sentences from oracle keyphrases and estimated keyphrases. We use PASCAL Sentence and IAPR-TC12 because both have human-generated sentences. . . . .	84

# Chapter 1

## Introduction

### 1.1 Background

Recently, the development and spread of information technology has increased the amount of accessible multimedia such as images and videos. For efficient usage of those resources, multimedia should be retrieved and understood easily by users. Therefore, methods by which computers can interpret multimedia contents on behalf of humans are necessary. Future methods must not only use the contents of multimedia but also events and objects in the real world for interpretation by computers and robots working in the real world, by systems helping vision impaired people, and so on.

Consequently, object recognition and event recognition from multimedia have been widely investigated in the last decade. Objects and events are recognized by being labeled with the names of objects and events. However, the multimedia contents cannot be understood completely with such labels. Of course, information about locations is lost. More generally, relations among these objects and events cannot be understood merely using independent labels. A simple example is shown in Figure 1.1. Both images have the common labels of “white”, “blue”, “sky”, and “airplane”. However, these labels cannot reflect the relations between objects and colors. Additionally, spatial relations between “airplane” and “sky” cannot be described.

Therefore, methods to associate multimedia (especially images) with a natural sentence are starting to be addressed widely. For the example presented in Figure 1.1, the goal is to generate sentences such as “A white airplane is flying in the blue sky.” and “A blue airplane is under the white sky.”

In 2010, Farhadi et al. published a landmark work [6] for describing images with sentences. Images and sentences are manually labeled with a triplet of  $\langle \text{object, action, scene} \rangle$ . Therefore, relations between objects and events in images are determined with these triplets. Subsequently the mappings from images to

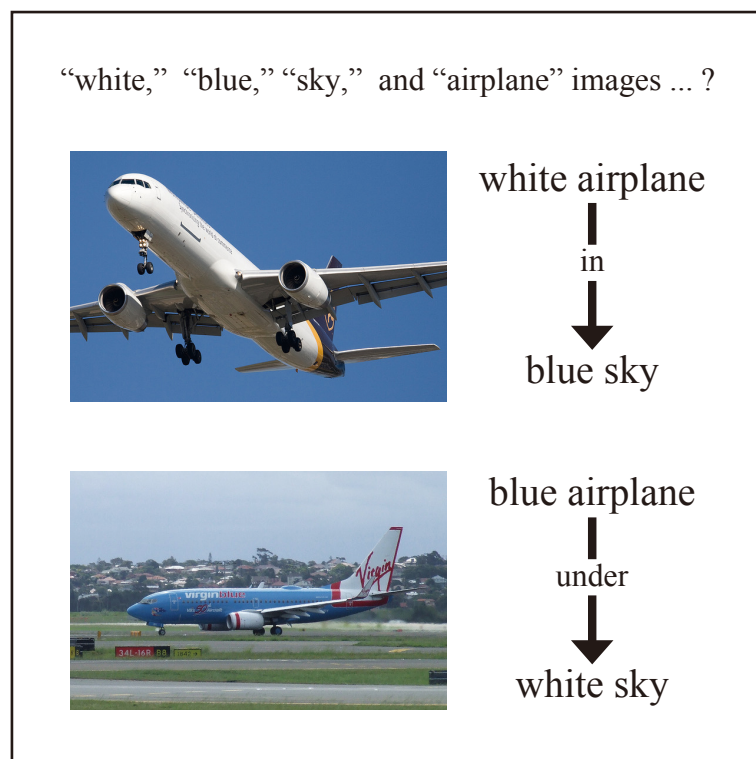


Figure 1.1: Two images might have the same labels, “white”, “blue”, “sky” and “airplane”. However, the relations among these labels are ignored when using only these labels.

triplets and from triplets to sentences are learned. Then images can be described by estimating the triplet and retrieving a sentence associated with a triplet that resembles the estimated triplet.

Most recent studies use multiplets. To define the spatial relation, the labels are estimated for each element: object, action, scene, and proposition. A few works attempt to describe videos with sentences using a multiplet. However, such an approach to describe multimedia with sentences presents two problems. First, the contents and their relations beyond the multiplet cannot be described. Secondly, collecting numerous images with such well-organized multiplets is difficult.

A sufficient number of high-quality training samples with desired outputs is generally needed. Therefore, consideration of preparing such a sufficient dataset presents an important issue. Fundamentally, visual recognition from images and videos has several problem settings depending on the form of recognition result. At the same time, the type of training dataset changes according to the problem. For example, Image Segmentation, which is a problem of assigning each pixel

---

to each object in the input image, requires a dataset consisting of images and pixel-wise segments about the objects in each image. Such data are difficult to collect in large amounts. Therefore, improving the accuracy of segmentation is also difficult. Associating an input image with a single label (Classification) or with multiple labels (Annotation) requires only pairs of an image and labels. Using a large amount of labeled images from general web pages and multimedia-sharing sites such as Flickr and YouTube, datasets for Image Classification and Annotation can be collected in large amounts. As a result, these problems can be assessed and resolved rapidly.

What can we collect in large numbers for describing multimedia with natural sentences? The answer is web data. General web sites and sharing sites have numerous images and videos associated with captions.

## 1.2 Objective

We develop a system to generate natural sentences for images using a dataset consisting only of pairs of images and sentences.

To realize sentence generation using only images and sentences, first we present a hypothesis: *Almost all contents of an image are identifiable with a few descriptive phrases (keyphrases). A sentential caption can be generated by connecting these with an experimental grammar model.*

Secondly, we tackle a problem to estimate multiple keyphrases as an Image Annotation problem, where images are associated with multiple labels. In other words, keyphrases are represented as labels consisting of a set of words. Recent studies of large scale Image Classification adopt online learning for linear classification. In an online learning scheme, each training sample is classified repeatedly with the classifiers at the time. Then some are updated if a mistake occurs. For reading one training sample at a time, online learning is suitable for a large dataset. Although various learning methods for linear classification have also been proposed in the machine learning literature, they have rarely been evaluated for visual recognition. As described herein, we present guidelines via investigations of state-of-the-art online learning methods of linear classifiers. After the investigation, we propose several online learning methods for keyphrase estimation. Our methods achieve state-of-the-art performance and superior scalability to that of existing methods for Image Annotation.

Finally, we develop a method to generate a natural sentence from estimated keyphrases and a grammar model. Natural Language Generation (NLG) is an open problem for Natural Language Processing. Existing works for generating sentences for images are grouped into two major categories: (i) reuse of existing sentences in the dataset and (ii) usage of a template consisting of subjects, verbs,



---

and prepositions. For sentence generation with a template, each part of speech is definable because those works use multiplet consisting of such as object, action, scene, and preposition. Therefore, both approaches lack flexibility of descriptions for each image. Instead, we propose a novel method by solving sentence generation as a graph search problem based on our intuition that a sentence can be generated by combining the estimated keyphrases using a grammar model.

The contributions of this thesis are summarized as follows:

- Proposal of the keyphrase approach to realize sentence generation for images using a dataset consisting only of images and sentences.
- Study investigating state-of-the-art algorithms for large-scale visual recognition and evaluating those algorithms in unified experimental settings.
- Development of a learning method for keyphrase estimation and annotation with scalability and accuracy.

### 1.3 Structure of the Thesis

In this thesis, we aim to develop a system to generate a sentence that explains the contents of images. First, we investigate state-of-the-art online learning methods for large-scale visual recognition. Secondly, we develop a basic pipeline to generate a sentence via keyphrase estimation. Thirdly, we propose a novel method to estimate keyphrases accurately. Finally, we conduct experiments to generate sentences using a dataset consisting of image and sentence pairs.

This thesis is organized as follows: we have already described the background and the objective of this thesis in Chapter 1. Chapter 2 describes related works, especially works to associate images with sentences. Works for visual recognition such as classification and annotation are also related to this thesis. In Chapter 3, we provide novel guidelines for large-scale visual recognition by application of state-of-the-art online learning methods. This knowledge affords a useful clue to development of a keyphrase estimation algorithm. Our particular methodology of sentence generation via keyphrase estimation is proposed in Chapter 4. First, we train keyphrases using pairs of an image and sentences. Secondly, for an input image, keyphrases are estimated. A sentence is generated by combining the keyphrases using a grammar model. Additionally, this chapter modifies an existing online learning method. Although the existing method is developed for classification, we devote attention to the fact that each image has several labels for annotation problems. Chapter 5 proposes a novel online learning method to learn numerous keyphrases. The proposed method learns a subspace in which (a) image features associated with the same keyphrase mutually approach and

---

(b) a linear weight vector to classify that keyphrase is obtainable. Experimental results obtained using several de-facto standard datasets for Image Annotation show that the proposed method is useful to associate images with several labels. In Chapter 6, we evaluated our methodology using three datasets consisting of pairs of an image and a sentence. Experimental results demonstrate that our system is more accurate than previous works. Finally, Chapter 7 concludes this thesis and describes future works.

## Chapter 2

# Methods to Describe Multimedia with Natural Language

Recent works for image recognition have advanced by virtue of machine translation. Whereas traditional works for machine translation are based on manually organized rules, statistical machine translation using a large bilingual corpus has become the mainstream from the close of the 20th century [7]. Because knowledge of statistical machine translation is brought to image recognition [8], where rule-based approaches were also mainstream, methods based on statistical learning continue to gain mainstream acceptance.

Recognizing contents in images and videos is divisible into several groups according to the form of output. Because enormously various and numerous approaches are conducted for those problem settings, it is difficult to describe all of them. Therefore, this thesis first presents efforts for Image Classification and Annotation, where the whole input image is associated with one or more labels. Secondly we describe recent challenges using output sentences to explain images.

### 2.1 Generic Object Recognition

As described in Chapter 1, studies using computers to recognize contents of images or videos have been widely undertaken. Moreover, wide variation prevails in the type of target object and in the form of output. Figure 2.1 presents a variation of the visual recognition problem.

First, targets to be recognized are divisible into two groups: objects and events. Although a few works attempt to detect events from still images, event recognition is performed mainly with videos. Video features such as optical flow and image features from each frame of the video are extracted first. Then these features are trained and recognized as usual object recognition.

---

If the targets are objects, then problems are divided depending on whether the objects are specific objects (e.g. Boeing 787 and iPhone) or generic objects (e.g. an airplane and a car). In mainstream studies for specific object recognition, interest points in the images are detected first. Secondly, descriptors representing pixels surrounding these interest points are extracted. The input image is recognized by matching its descriptors to all descriptors in the dataset. Most approaches for generic object recognition also extract descriptors. The difference from specific object recognition is that the descriptors are extracted from pixels surrounding not interesting points but fine grid-points because descriptors around all grid points include information related to the background of the object. Generic object recognition uses such background because relation between the object and its background is stronger than that for specific object recognition. However, some challenges for recent generic object recognition have been undertaken to recognize fine-grained objects such as dogs of 200 kinds, whereas usual generic object recognition consider all dogs as one class: “dog”. Consequently, the difference between specific and generic becomes continuous.

Moreover, when a certain object is recognized, the types of the output are divided depending on whether (a) the fact that the input image includes the object is reported (classification and annotation), (b) a bounding box is output for that object (detection), or (c) the region of the object is partitioned (segmentation). The difference between classification and detection is that the input image is associated with a single label or more than one label. As described in Chapter 1, the cost to collect datasets to train and evaluate systems influences the dataset size. Statistical approaches for visual recognition require sufficient data in proportion to the number of target objects and events. For labeling the whole image, collecting images associated with labels from the web is possible. As a result, whereas detection and segmentation are performed using hundred-thousands and thousands of images at most, respectively, classification and annotation are performed using millions or billions of images.

In this thesis, we discuss methods to associate images with labels or sentences. As described above, these methods cannot clarify the regions of recognized objects. However, object detection uses such labeling to reduce the object candidates because scanning detectors for objects of all kinds takes much time.

Video recognition uses a similar approach to that of image recognition. For example, most works in an international competition, TRECVID [9], where many tasks for videos such as event detection and indexing based on contents are addressed, are based on pipelines for image recognition.

Therefore, discussion of image recognition by labeling images contributes most settings for visual recognition.

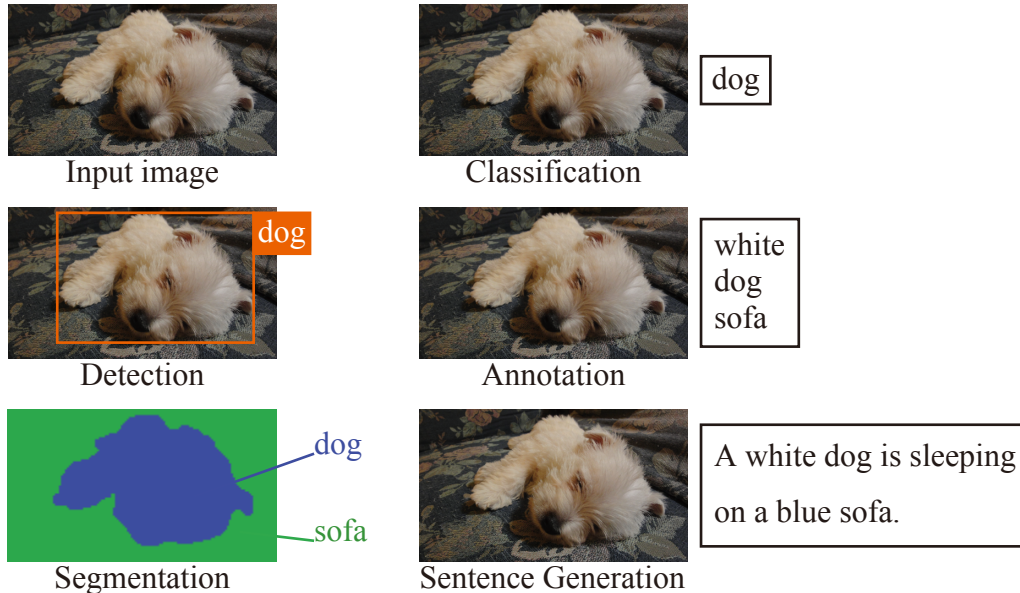


Figure 2.1: Variations of image understanding.

### 2.1.1 Datasets

With progress of visual recognition methods, the size of datasets used for visual recognition also increases. As pioneering datasets, Caltech 101 [10] for Image Classification and Corel 5k [8] for Image Annotation are widely used. Caltech 101, consisting of 101 object categories and one background category, includes about 9000 images downloaded using Google image search. Because all images are rotated and organized manually, Caltech 101 is easy to learn. Corel 5k, a subset of an image library provided by Corel, has about 5000 images for 260 categories. Each image has around 3.4 labels. Additionally, as a similar problem to that of Image Classification, there is another problem called Scene Classification. Because the pipelines used are mutually similar, we equate Scene Classification with Image Classification. For this problem, LSP15 [11], consisting of 15 scene categories, is a widely known dataset. Caltech 256 [12], consisting of 256 categories and around 30k varied images, is available for image classification. For image annotation, ESP Game [13] and IAPR-TC12 [14] are used as datasets consisting of around 20k images and around 250 categories.

Recently, several larger-scale datasets collected from the web are being used. NUS-WIDE [15] includes 5000 labels and 200k images collected from Flickr. Torralba et al. [16] collected 80 million images using image search engines. Using reduced-size images directly as image features, performance improvements over

---

several tasks such as Image Classification are reported. Wang et al. [17] organized a huge dataset called ARISTA, which consists of 2 billion images from the web. The authors devote their attention to images that closely resemble input images and insist that the so-called “near-duplicate” images are useful for Image Annotation.

Because labeled images from the web tend to have much label noise, the methodologies used for visual recognition using those images must be noise-tolerant. Therefore, Deng et al. [18] first collect images from the web with the queries based on a thesaurus called WordNet [19]. Then noisy images are removed using crowdsourcing, Amazon Mechanical Turk. As a result, their dataset, called ImageNet has 10 million images for 20k categories. Clowdsourcing is also used to organize the dataset called SUN [20] for scene classification. There are 100k images for 900 scenes in this dataset.

ImageNet seems to present no problem related to relevance between images and associated labels because this dataset is organized manually. In fact, however, not all labels are relevant to their images because of the difficulty in controlling workers having various backgrounds in crowdsourcing and because of the fact that usual images have more than one object and event. Indeed, the international competition called ImageNet Large Scale Visual Recognition Challenge (ILSVRC) uses a subset consisting of 1.2 million images for 1000 objects and evaluates performance using not the top-1 label but the top-5 labels estimated by participants.

### 2.1.2 Feature extraction

This subsection describes advances in feature extraction from images. The popular pipelines consist of descriptor extraction from thousands of local patches in images and aggregation of the local descriptors into one vector as a representation for each image. The traditional method is Bag of Visual Words (BoVW) [21], which generates a codebook with k-means clustering among descriptors extracted from images. By assigning each descriptor to a codeword in the obtained codebook, a histogram of assigned codewords can be extracted as an image feature vector. Although the means to define the number of codewords is uncertain, experimental reports show powerful performances for visual recognition.

However, because BoVW is a kind of vector quantization of local descriptors, a large amount of information is lost during quantization. To overcome the loss of information, nonlinear classification is adopted for visual recognition using small-scale dataset such as Caltech 101. The popular approach is a combination of Spatial Pyramid Match (SPM) Kernel [11] and Kernel SVM. With SPM, BoVW histograms are generated respectively over partitioned image. To use Kernel SVM, histogram intersection is used to calculate the kernel function

---

between BoVW+SPM vectors. Moreover, Multiple Kernel Learning [22, 23], which combines multiple kernel functions such as SPM Kernel, can achieve better performance than that with single kernel. However, because Kernel SVM requires  $O(N^3)$  time complexity and  $O(N^2)$  space complexity with respect to all  $N$  training samples, the problem of scalability persists.

Recently, various methods to aggregate local descriptors into a vector with less information loss are proposed. These features achieve powerful classification performance even with linear classifiers. The pipeline using BoVW and SPM is divisible into three steps: 1) extracting local descriptors, 2) coding each local descriptor, and 3) pooling all coded descriptors in the same region of an image. BoVW is a method to code each local descriptor by assigning the descriptor to one codeword. The original SPM can be interpreted as average pooling where coded descriptors in the same region are simply summed (averaged). ScSPM [24] uses  $L_1$ -norm regularization for coding to assign each local descriptor to a few codewords. For pooling, [24] presents max pooling where the aggregated feature vector is generated by choosing the maximum value for each codeword. [24, 25] report that the combination of ScSPM and linear SVM achieves surprisingly good classification performance.

BoVW can be regarded as a probability density estimation of a Gaussian mixture model in the space where local descriptors distribute. [26] presents Super-Vector (SV) coding as an extension of BoVW. Because SV approximates each neighborhood of each codeword using linear function, SV approximates a probability density better than BoVW. Moreover, by considering BoVW as a generative model, [27] proposes Fisher Vector (FV) to aggregates local descriptors into a more discriminative vector using Fisher Kernel. FV uses first-order and second-order statistics of the distribution of local descriptors, whereas BoVW uses 0th-order statistics. [28] improves Fisher Vector using  $L_2$  Normalization, Power Normalization, and Spatial Pyramid Matching. Nowadays FV is a standard approach to the large-scale image recognition.

### 2.1.3 Classification and Annotation

As described in the last subsection, linear classifiers achieve state-of-the-art performance for large-scale classification. In existing works for image annotation, however, non-parametric approaches [29, 30, 31, 32, 33] are mainstream. With such approaches, an input image is annotated with the labels of images which are located near the input image according to a certain metric. Although multi-keyphrase estimation using such an approach is conceivable, the complexity for calculating the distances between the input image and all  $N$  training samples is  $O(N)$ . Therefore, the execution time increases with the number of training samples. Moreover, the state-of-the-art methods [31, 32, 33] require the complexity

---

to be within  $O(N) \sim O(N^2)$  for metric learning. Therefore, scalability remains an open question for non-parametric approaches.

## 2.2 Challenge to Associate Multimedia with a Caption

To generate sentences for images, ascertaining relations between labels is an important problem to be solved. Sadghi et al. [34] presents visual recognition with “Visual Phrases”. For example, detecting “person\_riding\_horse” is performed by decoding the results of object detection for each object such as person and horse. However, the following problems exist: (i) binary classification is performed for each phrase, and (ii) detection-based phrase estimation requires manually managed datasets including bounding boxes.

Sentence generation from image contents was started by [35], which was published in 2010. An input image was segmented for each object. Then each segment was homologized to a verbal expression. Finally, an output sentence was generated using these expressions while maintaining correct grammar. Because this approach requires numerous special datasets for each process, extension to a dataset of various images is difficult. Another method [36] uses the geo tag of an input image to retrieve related articles and to summarize them as a sentential caption. This approach is useful for some contents such as landmarks.

Several datasets are used to evaluate generated sentences in recent works. [37] use crowdsourcing by Amazon Mechanical Turk to collect two datasets: a PASCAL Sentence dataset and a Flickr-8k dataset. In both datasets, each image is described manually with around five sentences. IAPR-TC12 [14] consists of 20,000 images with sentential descriptions and light annotations in English, German, and so on. This dataset is used not only for Image Annotation but also for Sentence Generation. The SBU dataset [38] is generated by collecting 1M images from the image sharing website, Flickr. Because the sentences in the web are noisy, [39] presents the novel task called Image Caption Generalization to eliminate the noisy description.

Some methods in the literature reuse the whole sentence associated with the training image that is sought from an input image. In [6], all images are labeled with a triplet:  $\langle \text{object, action, scene} \rangle$ . An image with the same labels estimated from an input image is retrieved. In another method [38], images are labeled according to their objects, stuff, people, and scenes from different datasets. Similarity to estimated labels from an input image and matching of local descriptors are used to search for similar images found among one million images. [40] adopts Kernel CCA to associate existing sentences to input images using their original dataset called Flickr-8k dataset [41]. To use a whole sentence directly, how-



---

ever, vastly numerous images related to all combinations of image contents must be used. Moreover, similar images must be retrieved exactly from such a huge dataset.

Therefore, some works generate a new sentence using one or more templates. In [4], images are explained sententially with respect to the objects' names, number, and their spatial relations by learning objects, stuff, and attributes from different datasets. [2] extended the concepts described earlier in [6] to generate a new sentence learning not only ⟨object, action, and scene⟩, but also a preposition. Whereas most works try to generate sentences from input images, [42] presents a bit different problem: to generate a sentence from an already annotated input image. Given labels corresponding to objects and attributes, a proper action and their relations are estimated to generate a sentence.

However, the authors of [3, 43, 44] report that the use of a template to generate general sentences is suboptimal. [3, 43, 44] alternatively connect the result of object detection using web-scale N-gram model [43] and their generating system by gathering local subtrees for each detected object [3, 44]. Although new sentences can be generated, [2, 4] requires datasets including multiples and [3, 43, 44] requires an extra dataset for object detection.

Previous works described by Feng [45] and us [1] are most related to the present thesis. Feng [45] used an annotation method to generate a sentence from images. In [1], we also apply a method for image annotation, called Canonical Contextual Distance (CCD) [46]. With CCD, distances in image feature space are improved according to labels. [1] modified CCD to use not labels but sentences associated with images. Then images that are similar to an input image are sought; a sentence is generated from the phrases in the sentences associated with those images. The pipeline of [1] is presented in Figure 2.2.

The weaknesses of these methods are the following. First, as one example, [45] requires manual selection of labels, subject, actions, and adjectives. Furthermore, although an output sentence is generated by connecting phrases, [45] regards phrases as grammatical. Because parsing and word tagging are necessary for grammatical phrase extraction, (i) additional training datasets for parsing and tagging are also required and (ii) mistakes of these process adversely affect the generated sentences. Secondly, the approach described in [1], from another perspective, estimates phrases to generate a sentence from similar images' sentences. However, (i) time spent on neighbor search increases linearly with the number of training samples, and (ii) the absence of a grammar model might generate unnatural sentences.

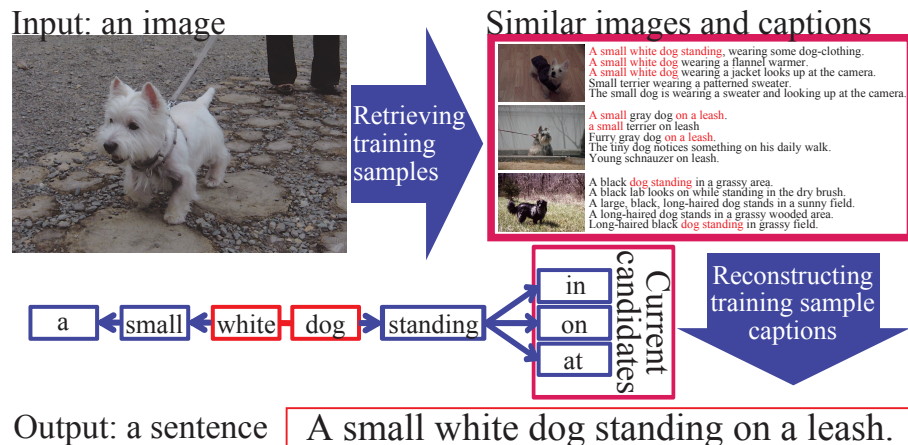


Figure 2.2: Methodology overview of [1].

## 2.3 Multi-Keyphrase Approach for Sentential Description

In general terms, collecting a large amount of data from the web is a common means to understand various images. What we can collect automatically are images associated not with semantically clear labels but with surrounding sentences and uncontrolled words. Indeed, [38] collects a million pairs of an image and a sentential caption from an image sharing website, Flickr, for sentence generation from images.

As described in Section 2.1.3, various studies have specifically addressed image classification and annotation using a large-scale dataset collected from the web. Even though noisy textual information is common around the images on the web, [47, 48] report that the surrounding text can improve similarity among images. In [48], for example, articles and their images in New York Times are collected as a dataset. The performance of similar image searching among these news images is improved by combining similarities among the news articles.

Recent works attempt to describe videos [49, 50, 51, 52, 53] with sentences. However, these approaches are also based on multi-plets and object detection. To generate sentences for videos, studies to generate sentences for images should be undertaken using only pairs of an image and sentences.

Consequently, we strive to develop an approach to generate a new sentence for input images using only pairs of an image and sentences. Our objective is to generate a sentence such as “A man bites a white dog in his arms.” on the right side of Figure 2.3 from the input image on the left side of Figure 2.3. Generally,

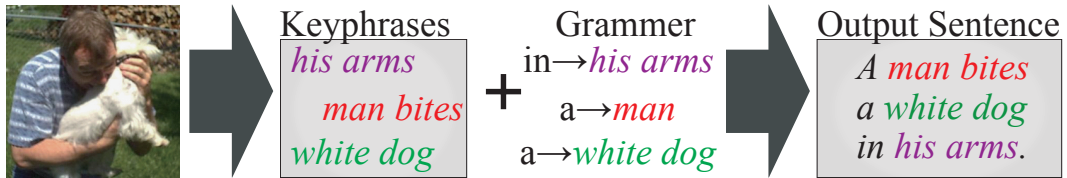


Figure 2.3: Pipeline of the proposed framework. We first estimate some *keyphrases* from an input image. Then a sentential caption is generated by spinning them with an experimental grammar model.

given  $|w_{all}|$  kinds of words, there are  $l^{|w_{all}|}$  sentences in which  $l$  is the length of the sentences. Therefore, directly estimating one of  $l^{|w_{all}|}$  sentences for each input image is impossible. However, once the combinations of a few words such as “man-bites”, “white-dog”, “his-arms” are estimated, a sentence can be generated using an experimental grammar model such as “in” appears frequently after “his arms”, “a” appears frequently before “white dog” and after “bites”. Therefore, we form the following hypothesis:

**Hypothesis** Almost all contents of an image are identifiable with a few descriptive phrases (*keyphrases*). A sentential caption can be generated by connecting these with an experimental grammar model.

Consequently, we present *Multi-keyphrase Problem* to estimate keyphrases<sup>1</sup> which clarify the relations between contents in images. Additionally, keyphrases can narrow the semantic gap between image contents and single words. For example, to learn general “dog”, following dogs should be considered to be in the same class: “white dog”, “black dog”, “running dog”, “sleeping dog”, and so on. Keyphrases can distinguish those dogs by modifying the word “dog”.

From the viewpoint of *Multi-keyphrase Problem*, our previous work [1] estimates keyphrases by retrieving similar images using CCD. Some approaches follow our first work [1] to generate sentences based on phrases, with [54, 55, 56] also proposing a method to estimate several phrases for input images and to generate a sentence by combining them.

However, [55, 56] present methods that entail several problems. First, because [55, 56] adopt metric learning [57], their approaches are not scalable for the data amount. Secondly, [55, 56] use a template to combine these phrases. As the authors of [3, 43, 44] report, template usage is suboptimal to generate general sentences.

<sup>1</sup>As described in this paper, a phrase means just a sequence of words. It has no other grammatical meaning.

---

The authors of [54] introduce integer linear programming instead of the solid templates. Although [55, 56] estimate phrases directly from images, the estimation in [54] is divisible into two steps. First, given an input image, several visual recognition modules are applied to the input: object detection, scene classification, and background (e.g., grass, sky) recognition. Presuming that a dog is detected in the input image, the authors retrieve a visually similar dog from the dataset, and extract phrases with a parser from the associated sentence. The problems of [54] are that they use organized datasets for each attribute and that a similar image search would be naive to ascertain proper phrases because of semantic gap. Additionally, even though [54] uses integer linear programming instead of a solid template, the contents to be described in a sentence are fixed.

As described in this paper, we develop a scalable system to train the relations between keyphrases and images. To estimate keyphrases and to obtain a grammar model, we use only pairs of an image and sentences. Because manual collection of datasets including semantic knowledge such as what is subject, action, and scene is no longer required, we can adopt a large-scale dataset consisting only of images and sentences.

During the last decade, numerous online learning methods for linear classification have also been widely studied [58, 59, 60, 61, 62, 63, 64] to address vast quantities of data which cannot be loaded on RAM at a time. Despite progress in online learning methods, few evaluations of these methods have been reported for large-scale visual recognition. Most online learning methods are evaluated using synthetic datasets and natural language datasets such as document classification. Therefore, before describing our particular pipeline to generate a sentence via keyphrase estimation in Chapter 4, we investigate state-of-the-art online learning methods over various image features using large-scale datasets in Chapter 3.

## Chapter 3

# Investigation of Online Learning Methods for Multiclass Classification

As described in Chapter 2, we develop novel online learning algorithms for phrases. To learn numerous phrases, large datasets consisting of many labels should be used. Therefore, we would like to start with investigation of multiclass classification using large datasets.

Nowadays, for large-scale visual recognition, combinations of high-dimensional features and linear classifiers are widely used. Numerous so-called mid-level features have been developed and mutually compared on an experimental basis. Although various learning methods for linear classification have also been proposed in the machine learning and natural language processing literature, they have rarely been evaluated for visual recognition.

In this chapter, we give guidelines via investigations of state-of-the-art online learning methods of linear classifiers. Many methods have been evaluated using toy data and natural language processing problems such as document classification. Consequently, we gave those methods a unified interpretation from the viewpoint of visual recognition. Results of controlled comparisons illustrate three guidelines that not only help us develop novel algorithms but also changing the pipeline for visual recognition.

### 3.1 Necessity to Investigate Existing Algorithms

By virtue of recent advances in computer science and because of the culture of sharing of multimedia information such as photographs, vast quantities of labeled images have been used for visual recognition [64, 65, 66, 67]. Combinations

---

of high-dimensional features and linear classifiers have been studied specifically. Such high-dimensional features are generated from each image by pooling many mid-level features. Each mid-level feature is coded from a local descriptor. Recently, many techniques for coding and pooling have been proposed and compared using well-known datasets[25, 68].

During the last decade, numerous online learning methods for linear classification have also been widely studied [58, 59, 60, 61, 62, 63, 64] to address vast quantities of data which cannot be loaded on RAM at a time. Given the  $t$ -th training sample,  $\mathbf{x}_t \in \mathbb{R}^d$ , associated with a label,  $y_t \in \mathcal{Y} = \{y_1, \dots, y_m\}$ , the sample is classified with the present weight vector,  $\boldsymbol{\mu}_t^{y_t}$ , as  $\hat{y}_t = \operatorname{argmax}_y \boldsymbol{\mu}_t^y \cdot \mathbf{x}_t$ . Here, bias  $b$  is included in  $\boldsymbol{\mu}_t$  as  $\boldsymbol{\mu}_t^\top \leftarrow [\boldsymbol{\mu}_t^\top, b]$  by redefining  $\mathbf{x}_t^\top \leftarrow [\mathbf{x}_t^\top, 1]$ . The classifiers suffer from a loss when they misclassify a datum and get updated as  $\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \tau_t \mathbf{x}_t$ , where  $\tau_t$  determines the step size. Learning can be performed by holding one datum. Therefore, online learning methods are appropriate for large-scale problems. Additionally, state-of-the-art methods outperform batch learning methods such as Support Vector Machine (SVM), as reported in [69, 70].

Despite progress in online learning methods, few evaluations of these methods have been reported for large-scale visual recognition. Almost all approaches to obtain linear classifiers have been online versions of SVM [58, 71, 72] with a one-versus-the-rest (OVR) manner. Furthermore, the original SVM is not a multiclass classifier. With OVR, we divide training samples into a positive class or a negative class for each label. Then we train binary SVM for each label. When we use OVR, however, the quantities of samples in the two classes (positive and negative) are imbalanced. Moreover, learning with OVR takes more CPU time because OVR might update many more weights than MUL in each step.

The newest learning method seems to perform best for large-scale visual recognition. Nevertheless, this inference is *not valid* because most online learning methods are evaluated using synthetic datasets and natural language datasets such as document classification. In NLP, feature vectors are based on the Bag-of-Words (BoW) model, in which each dimension in the feature vector for each datum represents the presence of a certain word in the datum. In such cases, feature vectors tend to be sparse. Consequently, many researchers devote attention to adaptive learning when the occurrence ratio of each dimension of feature vectors differs. In visual recognition problems, however, feature vectors tend to be much denser than those of NLP problems, as described later. Therefore, we must evaluate state-of-the-art algorithms in large-scale visual recognition.

As described in this chapter, to give guidelines to choose learning methods for large-scale visual recognition, we investigate state-of-the-art online learning methods over various mid-level features using large-scale datasets.

The remainder of this chapter is organized as follows: Section 3.2 introduces related works for large-scale visual recognition. In Section 3.3, we overview state-

---

of-the-art algorithms from various perspectives. Qualitative and quantitative discussions are given, respectively, in Section 3.4 and Section 3.5. From these discussions, three guidelines in Figure 3.10 are obtained.

## 3.2 Related Works for Large-scale Classification

To achieve generic object recognition, large datasets are required because numerous objects with various appearances must be assessed. For example, with ImageNet [18], there are 14 million images for the 20,000 words in WordNet [19]. Therefore, scalability of the data amount is necessary.

For scalability, combinations of high-dimensional features and linear classifiers have been widely studied [66, 67]. Mid-level features have been improved from traditional Bag-of-Visual-Words (BoVW) models [21]. Although a BoVW vector for each descriptor has only one non-zero element, recent mid-level features extract richer information: second-moment [28], first-moment [26, 73], and zero-moment [24, 74, 75] with respect to between descriptors and code words. Those features have been compared with common datasets. Some studies [25, 68] have compared the conventional features in a unified evaluation setting.

Perceptron [63] has started the development of online learning algorithms for linear classification. As described in Section 3.3, these algorithms are divisible into two groups: first-order methods and second-order methods. Perceptron and gradient-based online SVMs [58, 71, 72, 76] are first-order methods. Other first-order methods [59, 77] can adjust the step size automatically. Recently, second-order algorithms [60, 61, 62, 69, 70, 78, 79, 80] have been studied thoroughly for adaptive updates for each dimension using second-order information. They outperform batch SVM.

Although many proposals of mid-level features and their evaluations [25, 68] exist, few works describe investigations of learning methods for linear classifiers. In papers proposing the algorithms, the use of synthetic data and commonly used ML/NLP datasets is typically described. Few reports have described their evaluation. In [81], only first-order algorithms and the averaging technique are evaluated. In [82], linear SVMs (including OVR and multiclass) have been investigated for large-scale problems. In [83], a link between Perceptron and SGD-SVM is discussed, but no quantitative comparison is included. Furthermore, evaluations of these algorithms for visual recognition are rare. In [66, 84], some versions of SGD-SVM are evaluated with ImageNet [18]. [84] also proposed a reweighting OVR.

---

### 3.3 Online Learning Algorithms

In this section, we introduce state-of-the-art online learning methods [58, 59, 60, 61, 62, 63] for linear classification. All update rules of the methods are summarized in Table 3.1.

Fundamentally, each method has been proposed as learning for binary classification. Two commonly used techniques apply binary classifiers themselves to multiclass problems. One is the one-versus-the-rest (OVR) technique described in Section 3.1. The other is the one-versus-one (OVO) technique. With the OVO, we train  $mC_2$  classifiers for all pairs of labels. We use the OVR because OVO requires numerous classifiers for labels of many kinds.

An overview of binary learning is shown in Figure 3.1. In an online learning scheme,  $t$ -th sample,  $\mathbf{x}_t$ , is classified with the  $t$ -th weights,  $\boldsymbol{\mu}_t$ , by checking the sign of the inner product,  $\mathbf{x}_t \cdot \boldsymbol{\mu}_t$ . Samples are permuted randomly for stable learning because arranging samples in label order makes conversion slow. We can verify the prediction by checking whether the margin,  $\gamma_t = y_t(\mathbf{x}_t \cdot \boldsymbol{\mu}_t)$ , is greater than zero or not because of the ground truth,  $y_t = \pm 1$ . However, we seek to enlarge the margin  $\gamma_t$  for stable classification. Therefore, we verify that  $\gamma_t > E$ , where  $E \geq 0$ .

We update  $\boldsymbol{\mu}$  using a step size  $\alpha_t$  as  $\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \alpha_t y_t \mathbf{x}_t$  in first-order algorithms. Recent second-order algorithms use  $\Sigma_t \in \mathbb{R}^{d \times d}$  as confidence information for the dimensions where non-zero values frequently appear not to be updated widely. We must learn  $d \times d$  elements for each label if  $\Sigma_t$  is a full matrix. Therefore, diagonal matrices are commonly used. In Figure 3.1 and Figure 3.2,  $\text{diag}(\mathbf{x}_t)$  is a diagonal matrix having elements of  $\mathbf{x}_t$  as diagonal elements.

An overview of multiclass (MUL) learning is portrayed in Figure 3.2. Given the sample,  $\mathbf{x}_t$ , and its label,  $y_t$ , which now represents the label number, we treat a violating label,  $y'_t = \arg \max_{y \in \mathcal{Y} \setminus y_t} \boldsymbol{\mu}_t^y \cdot \mathbf{x}_t$ . Then we redefine the margin,  $\gamma_t = \boldsymbol{\mu}_t^{y_t} \cdot \mathbf{x}_t - \boldsymbol{\mu}_t^{y'_t} \cdot \mathbf{x}_t$ , and check if  $\gamma_t > E$ .

For multiclass learning, we must learn  $\boldsymbol{\mu}$ s and  $\Sigma$ s comprehensively. However, algorithms proposed for binary classification only handle a pair of one  $\boldsymbol{\mu}$  and one  $\Sigma$ . Therefore, in accordance with [59], we combine all  $\boldsymbol{\mu}_t^y \in \mathbb{R}^d$  and  $\Sigma_t^y \in \mathbb{R}^{d \times d}$  into one vector,  $\mathbf{M}_t \in \mathbb{R}^{dm}$ , and one block-diagonal-matrix,  $\mathbf{S}_t \in \mathbb{R}^{dm \times dm}$ , respectively. Additionally, we replace  $\mathbf{x}_t$  with  $\mathbf{X}_t(y_t) \in \mathbb{R}^{dm}$ .  $\mathbf{X}_t(y_t)$  and  $\mathbf{M}_t$  consist of  $m$  sub-vectors, and  $\mathbf{S}_t$  consists of  $m \times m$  sub-matrices. The  $y_t$ -th sub-vector of  $\mathbf{X}_t(y_t)$  is  $\mathbf{x}_t$ . The others are zero vectors. The  $y_t$ -th vector of  $\mathbf{M}_t$  and the  $y_t$ -th main diagonal matrix of  $\mathbf{S}_t$  are defined respectively as  $\boldsymbol{\mu}_t^{y_t}$  and  $\Sigma_t^{y_t}$ . Learning can then be performed with these:  $\mathbf{X}(y)$ ,  $\mathbf{M}$ , and  $\mathbf{S}$ , using each algorithm. We can obtain the update rules for multiclass classifications through the following replacement schemes in Table 3.1:



---

```

Initialize  $\boldsymbol{\mu}_0 = 0$  and  $\Sigma_0 = I$ 
while classifiers are not converged do
  for  $t = 1, 2, \dots, N$  do
    Receive sample  $\mathbf{x}_t \in \mathbb{R}^d$ .
    Predict  $\hat{y}_t = \text{sign}(\boldsymbol{\mu}_t \cdot \mathbf{x}_t)$ .
    Get true label  $y_t$  and margin  $\gamma_t = y_t(\boldsymbol{\mu}_t \cdot \mathbf{x}_t)$ .
    if  $\gamma_t < E$ 
      Set  $\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \alpha_t y_t \Sigma_t \mathbf{x}_t$ .
      Set  $\Sigma_{t+1}^{-1} = \Sigma_t^{-1} + \beta_t \text{diag}(\mathbf{x}_t)^2$ .
    end if
  end for
end while

```

Figure 3.1: Overview of learning binary classifiers.

- $\gamma_t = y_t(\boldsymbol{\mu}_t \cdot \mathbf{x}_t)$  with  $\gamma_t = \boldsymbol{\mu}_t^{y_t} \cdot \mathbf{x}_t - \boldsymbol{\mu}_t^{y'_t} \cdot \mathbf{x}_t$ .
- $v_t = \mathbf{x}_t^\top \Sigma_t \mathbf{x}_t$  with  $v_t = \mathbf{x}_t^\top (\Sigma_t^{y_t} + \Sigma_t^{y'_t}) \mathbf{x}_t$ .
- $l(\mathbf{x}_t)^2 = \|\mathbf{x}_t\|^2$  with  $l(\mathbf{x}_t)^2 = 2\|\mathbf{x}_t\|^2$ .

Whether a classifier is for binary or for multiclass, the label for a sample  $\mathbf{x}_t$  is predicted as  $\hat{y}_t = \underset{y}{\operatorname{argmax}} \boldsymbol{\mu}_t^y \cdot \mathbf{x}_t$ .

### 3.3.1 Perceptron

Perceptron, which was proposed in [63] more than half a century ago, is a traditional algorithm for use as a linear classifier. Margin  $\gamma_t$  is simply expected to be more than zero. Although the step size is also defined simply as one in [63], we tune the fixed step size  $\alpha_t = C$  for better accuracy.

### 3.3.2 Stochastic Gradient Descent SVM

The objective function of the original SVM, which is batch learning, is:

$$\boldsymbol{\mu} = \underset{\boldsymbol{\mu}}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{\mu}\|^2 + \sum_{n=1}^N \alpha_n \max\{0, 1 - \gamma_n\}, \quad (3.1)$$

where  $N$  is the number of all training samples. This batch version of SVM can be converted to online learning methods by introducing stochastic gradient descent (SGD) [58, 71] or sub-gradient descent [72]. Pegasos [72] used a sub-gradient of the object function with a subset of training samples. When the size

---

```

Initialize  $\boldsymbol{\mu}_0 = 0$  and  $\Sigma_0 = I$ 
while classifiers are not converged do
  for  $t = 1, 2, \dots, N$  do
    Receive sample  $\mathbf{x}_t \in \mathbb{R}^d$ .
    Predict  $\hat{y}_t = \arg \max_{y \in \mathcal{Y}} (\boldsymbol{\mu}_t^y \cdot \mathbf{x}_t)$ .
    Get ...
      true label  $y_t$ ,
      violating label  $y'_t = \arg \max_{y \neq y_t} (\boldsymbol{\mu}_t^y \cdot \mathbf{x}_t)$ ,
      and their margin  $\gamma_t = \boldsymbol{\mu}_t^{y_t} \cdot \mathbf{x}_t - \boldsymbol{\mu}_t^{y'_t} \cdot \mathbf{x}_t$ .
    if  $\gamma_t < E$ 
      Set  $\boldsymbol{\mu}_{t+1}^{y_t} = \boldsymbol{\mu}_t^{y_t} + \alpha_t \Sigma_t^{y_t} \mathbf{x}_t$ .
      Set  $\boldsymbol{\mu}_{t+1}^{y'_t} = \boldsymbol{\mu}_t^{y'_t} - \alpha_t \Sigma_t^{y'_t} \mathbf{x}_t$ .
      Set  $(\Sigma_{t+1}^{y_t})^{-1} = (\Sigma_t^{y_t})^{-1} + \beta_t \text{diag}(\mathbf{x}_t)^2$ .
      Set  $(\Sigma_{t+1}^{y'_t})^{-1} = (\Sigma_t^{y'_t})^{-1} + \beta_t \text{diag}(\mathbf{x}_t)^2$ .
    end if
  end for
end while

```

Figure 3.2: Overview of learning multiclass classifiers.

of this subset becomes one, the algorithm closely simulates the stochastic gradient descent (SGD-SVM) [58, 71] as:

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \alpha_t \nabla (\lambda \|\boldsymbol{\mu}\|^2 + \max\{0, 1 - \gamma_t\}), \quad (3.2)$$

where  $\lambda$  is a hyperparameter that must be tuned manually. As described in [58], there is also a second-order version of SGD using an approximation of the Hessian for the objective function. In this chapter, first-order SGD is used because the first-order version is commonly used for visual recognition [66, 84].

The most important problem is to design step size  $\alpha_t$ . A usual technique [58, 71] states that  $\alpha_t = 1/\lambda(t + t_0)$  with another hyperparameter  $t_0$ . The other [83, 84] is to define that  $\alpha_t = C$  where  $1 \gg C > 0$ .

Another problem is related to regularization. We should tune the parameter  $\lambda$ . One empirical definition is  $\lambda = 1/N$  where  $N$  is the data amount. Using a naive implementation, we must regularize all classifiers whether  $1 - \gamma_t > 0$  or not. One well known approach is to divide  $\boldsymbol{\mu}_t$  into  $l_t \mathbf{r}_t$ , where  $l_t$  is the  $L_2$  norm of  $\boldsymbol{\mu}_t$  and  $\mathbf{r}_t$  is normalized vector of  $\boldsymbol{\mu}_t$ . Consequently, regularization can be performed by multiplying  $l_t$  by  $(1 - \alpha_t \lambda)$ . However, [83] obviates regularization by defining  $\lambda = 0$ . Instead, the authors use early stopping, which is stopping training using a validation dataset. This version of SGD-SVM is the same as a variation of

---

Perceptron called Margin Perceptron [85]. Indeed, [84] uses fixed step size  $\alpha_t = C$  and discards regularization for large-scale datasets. Experimental results in [84] show that SGD-SVM without regularization (Margin Perceptron) achieves similar or superior performance to that of SGD-SVM with  $L_2$  regularization.

### 3.3.3 Passive–Aggressive

The largest benefit of Passive–Aggressive (PA) [59] is that the update coefficient is calculated analytically according to the loss. Here, we sought to decrease the hinge loss,  $1 - \gamma_t$  and not to change the weight radically:

$$\boldsymbol{\mu}_{t+1} = \underset{\boldsymbol{\mu}}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{\mu} - \boldsymbol{\mu}_t\|^2 \text{ s.t. } 1 - \gamma_t = 0. \quad (3.3)$$

The equation presented above is the objective function of PA. *Objective functions of all other algorithms explained later are also defined as a form of each update.*

This problem is solvable analytically with ease. An important shortcoming is that  $\boldsymbol{\mu}_{t+1}$  always classifies  $\mathbf{x}_t$  as  $y_t$ , whether  $y_t$  is correct or not. It is impossible to design large datasets that include no label noise. In [59], therefore, aggressiveness parameter  $C$  is introduced to soften the condition:

$$\boldsymbol{\mu}_{t+1} = \underset{\boldsymbol{\mu}}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{\mu} - \boldsymbol{\mu}_t\|^2 + C(1 - \gamma_t). \quad (3.4)$$

This version is called PA-I [59]. Another version, PA-II, uses the squared hinge loss,  $C(1 - \gamma_t)^2$ , in the second term. According to [59], the accuracies of PA-I and PA-II are mutually close. For these analyses, we used PA-I as PA.

### 3.3.4 Confidence-Weighted

The main difference between Confidence-Weighted (CW) [70] and PA is that CW has the confidence weight  $\Sigma$ , a diagonal  $d \times d$  matrix. If a classifier learns about a certain dimension of feature vectors many times, then the classifier must be more confident about that dimension. In other words, the classifier is expected to update less confident dimensions larger. Such an adaptive update makes convergence faster than the first-order algorithms.

Therefore, CW considers weights as a normal distribution,  $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ . We expect that the  $(t + 1)$ -th weight from  $\mathcal{N}(\boldsymbol{\mu}_{t+1}, \Sigma_{t+1})$  can classify  $\mathbf{x}_t$  correctly with a fixed probability,  $\eta$ . This condition is expressed as  $\gamma_t \geq \phi\sqrt{v_t}$ , where  $v_t = \mathbf{x}_t^\top \Sigma_t \mathbf{x}_t$ , and  $\phi = \Phi^{-1}(\eta)$ .  $\Phi$  is the cumulative function of the normal distribution.

To preserve the current classifiers, Kullback–Leibler divergence is used instead of the squared  $L_2$  norm,  $\|\boldsymbol{\mu} - \boldsymbol{\mu}_t\|^2$ , in PA. Consequently, the objective function

---

is the following:

$$\begin{aligned}
(\boldsymbol{\mu}_{t+1}, \Sigma_{t+1}) &= \underset{\boldsymbol{\mu}, \Sigma}{\operatorname{argmin}} D_{\text{KL}}(\mathcal{N}(\boldsymbol{\mu}, \Sigma) \parallel \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)), \\
\text{s.t. } \phi\sqrt{v_t} - \gamma_t &= 0.
\end{aligned} \tag{3.5}$$

In [70], the solution was approximated, whereas exact updates for binary and multiclass classifications were proposed, respectively, in [60] and [69].

### 3.3.5 Adaptive Regularization of Weight

The salient shortcoming of CW is its poor adaptability to label noise. Therefore, Adaptive Regularization of Weight (AROW) [61] introduces the squared hinge loss as:

$$\begin{aligned}
(\boldsymbol{\mu}_{t+1}, \Sigma_{t+1}) &= \underset{\boldsymbol{\mu}, \Sigma}{\operatorname{argmin}} D_{\text{KL}}(\mathcal{N}(\boldsymbol{\mu}, \Sigma) \parallel \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)) \\
&\quad + C(1 - \gamma_t)^2 + C\mathbf{x}_t^\top \Sigma \mathbf{x}_t,
\end{aligned} \tag{3.6}$$

where  $C$  is a constant parameter to be tuned.<sup>1</sup> The third term aims to converge classifiers faster. To improve the mistake bounds, New AROW (NAROW) [80] tunes  $C$  automatically at each step.

### 3.3.6 Gaussian Herding

Gaussian Herding (NHERD) [62] is a modified version of PA for second-order algorithms. As is true also for CW and AROW, weights in HERD are expressed with normal distributions,  $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ . Additionally, the  $t$ -th update is defined as a linear transformation of the distributions with matrices  $A_t$ . As a result, we obtained the objective function as shown below.

$$\begin{aligned}
(\boldsymbol{\mu}_{t+1}, A_t) &= \underset{\boldsymbol{\mu}, A}{\operatorname{argmin}} \frac{1}{2}(\boldsymbol{\mu} - \boldsymbol{\mu}_t)^\top \Sigma_t^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_t) \\
&\quad + \frac{1}{2}\operatorname{Tr}((A - I)^\top \Sigma_t^{-1}(A - I)\Sigma_t) \\
&\quad + C(1 - \gamma_t)^2 + \frac{C}{2}\mathbf{x}_t^\top A \Sigma_t A^\top \mathbf{x}_t.
\end{aligned} \tag{3.7}$$

---

<sup>1</sup>In [61],  $C$  is expressed as  $\frac{1}{2r}$ , where  $r$  is also a parameter to be tuned. Here, we use  $C$  for unified description.

---

### 3.3.7 Soft Confidence-Weighted

Soft Confidence-Weighted (SCW) [86] solves the problem of CWs poor adaptability to label noise by softening the condition according to the manipulation in PA [59]:

$$\begin{aligned}
 (\boldsymbol{\mu}_{t+1}, \Sigma_{t+1}) = \underset{\boldsymbol{\mu}, \Sigma}{\operatorname{argmin}} & D_{KL}(\mathcal{N}(\boldsymbol{\mu}, \Sigma) \parallel \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)) \\
 & + C \left( \phi \sqrt{\mathbf{x}_t^\top \Sigma_t \mathbf{x}_t} - m_t \right).
 \end{aligned} \tag{3.8}$$

The difference between Equation 3.5 and Equation 3.8 is the same as the difference between Equation 3.3 and Equation 3.4; the condition is added to the equation with aggressiveness parameter  $C$ . Therefore, there are two versions of SCW: SCW-I and SCW-II. For this study, we used SCW-I as SCW.

As described in Section 3.3, diagonal matrices are commonly used to update  $\Sigma$ . As summarized in [62], several methods exist to approximate  $\Sigma$  as a diagonal matrix. Herein, we employ *project* version, in which the approximation is performed by comparing only the diagonal elements in the update equation using the inverse of full matrix  $\Sigma$ . By comparing the diagonal elements in the update equation using the not-inversed matrix  $\Sigma$ , *drop* version can also be derived. Although the *project* version is reported to be slightly better than the others for CW, AROW and NHERD [62], SCW is originally proposed in the *drop* form only. For efficient comparison of the update rule, we derive the *project* version of SCW and introduce it in Table 3.1. In our experiments, we compared both the *drop* version and *project* version, which revealed no significant difference among them. Therefore, we show the results of *drop* version according to the original report [86]. The *drop* form of update for  $\Sigma$  is:

$$\Sigma_{t+1}^{y_t} = \Sigma_t^{y_t} - \beta'_t \Sigma_t \mathbf{x}_t \mathbf{x}_t^\top \Sigma_t^\top, \tag{3.9}$$

$$\Sigma_{t+1}^{y'_t} = \Sigma_t^{y'_t} - \beta'_t \Sigma_t \mathbf{x}_t \mathbf{x}_t^\top \Sigma_t^\top, \tag{3.10}$$

where  $\beta'_t = \alpha_t \phi / (\sqrt{u_t} + v_t \alpha_t \phi)$ , and  $u_t = (-\alpha_t v_t \phi + \sqrt{\alpha_t^2 v_t^2 \phi^2 + 4v_t})^2 / 4$ . Other variables,  $v_t$ ,  $\phi$ , and  $\alpha_t$ , are already defined in Table 3.1.

Table 3.1: Update rules. Variables in Figure 3.1 and Figure 3.2 are shown in the middle columns. The last column shows the parameters to be tuned. For binary classification,  $\gamma_t = y_t(\boldsymbol{\mu}_t \cdot \mathbf{x}_t)$ ,  $v_t = \mathbf{x}_t^\top \Sigma_t \mathbf{x}_t$ , and  $l(\mathbf{x}_t)^2 = \|\mathbf{x}_t\|^2$ . For multiclass classification,  $\gamma_t = \boldsymbol{\mu}_t^{y_t} \cdot \mathbf{x}_t - \boldsymbol{\mu}_t^{y_t} \cdot \mathbf{x}_t$ ,  $v_t = \mathbf{x}_t^\top (\Sigma_t^{y_t} + \Sigma_t^{y_t'}) \mathbf{x}_t$ , and  $l(\mathbf{x}_t)^2 = 2\|\mathbf{x}_t\|^2$ . Whether the classification is binary or multiclass,  $\phi = \Phi^{-1}(\eta)$  ( $\Phi$  is the cumulative function of the normal distribution),  $\psi = 1 + \phi^2/2$ , and  $\zeta = 1 + \phi^2$ .

Method	$E$	$\alpha_t$	$\beta_t$	Parameters
Perceptron [63]	0	$C$	0	$C$
SGD-SVM [58]	1	$C$	0	$C$
PA [59]	1	$\min\{C, (1 - \gamma_t)/l(\mathbf{x}_t)^2\}$	0	$C$
CW [60]	$\phi\sqrt{v_t}$	$\max\left\{0, \frac{1}{v_t\zeta} \left( -\gamma_t\psi + \sqrt{\gamma_t^2 \frac{\phi^4}{4} + v_t\phi^2\zeta} \right)\right\}$	$\frac{2}{-v_t + \sqrt{v_t^2 + 4v_t/(\alpha_t^2\phi^2)}}$	$\eta$
AROW [61]	1	$(1 - \gamma_t)/(v_t + 1/C)$	$C$	$C$
NHERD [62]	1	$(1 - \gamma_t)/(v_t + 1/C)$	$2C + C^2v_t$	$C$
SCW [86]	$\phi\sqrt{v_t}$	$\min\left\{C, \max\left\{0, \frac{1}{v_t\zeta} \left( -\gamma_t\psi + \sqrt{\gamma_t^2 \frac{\phi^4}{4} + v_t\phi^2\zeta} \right)\right\}\right\}$	$\frac{2}{-v_t + \sqrt{v_t^2 + 4v_t/(\alpha_t^2\phi^2)}}$	$C, \eta$

---

## 3.4 Common Qualitative Issues

Update rules of all learning methods are presented in Table 3.1. Again, it is noteworthy that objective functions of PA, CW, AROW, NHERD, and SCW are already shown in their subsections. These algorithms are designed using a form of each update, whereas batch SVM is designed using a total loss. In this section, we investigate two common issues.

### 3.4.1 OVR vs. MUL

Two common choices of OVR are e-OVR, for which the number of negative samples is the same as the number of positive samples, and u-OVR, for which all samples in other classes are selected as negative samples. In [84], reweighting samples for OVR (w-OVR) is proposed. In each iteration,<sup>1</sup> the number of negative samples is limited with respect to the number of positive samples.

[84] experimentally compared MUL and OVR using SGD-SVM. As a result, MUL outperformed e-OVR and u-OVR and w-OVR outperformed multiclass SGD-SVM. Authentically, OVR is easily parallelized because a classifier for each label can be learned independently. However, MUL can also be parallelized easily as described in [87]. In general, more accurate classifiers with less CPU time for learning are preferred. Herein, we compare these OVRs and MUL using state-of-the-art online learning methods.

### 3.4.2 Averaging

With most algorithms, training samples that are learned later strongly influence the classifiers. In [88], the weighted sum of  $\boldsymbol{\mu}_{1\dots T}$  is proposed for testing. Particularly, averaging them as  $\bar{\boldsymbol{\mu}} = \frac{1}{T}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2 + \dots + \boldsymbol{\mu}_T)$  greatly facilitates conversion. The authors of [88] insist that averaging is an approximation of a second-order algorithm. Indeed, averaged (first-order) SGD is proved to be an approximation of Newton-like second-order SGD.

In [66], averaging SGD SVM outperforms SGD SVM for visual recognition. However, averaging weights using other state-of-the-art online learning methods are rarely evaluated for visual recognition.

When averaging the classifiers, summing all  $m$  weights in each step is time consuming. Therefore, only the summations of updates  $\Delta$  are memorized.

$$\begin{cases} \Delta_{t+1}(\boldsymbol{\mu}^{y_t}) = \Delta_t(\boldsymbol{\mu}^{y_t}) + t\alpha_t y_t \sum_t^{y_t} \mathbf{x}_t, & \text{(binary)} \\ \Delta_{t+1}(\boldsymbol{\mu}^{y_t}) = \Delta_t(\boldsymbol{\mu}^{y_t}) + t\alpha_t \sum_t^{y_t} \mathbf{x}_t, \\ \Delta_{t+1}(\boldsymbol{\mu}^{y'_t}) = \Delta_t(\boldsymbol{\mu}^{y'_t}) - t\alpha_t \sum_t^{y'_t} \mathbf{x}_t. & \text{(multiclass)} \end{cases} \quad (3.11)$$

---

<sup>1</sup>Here iteration means learning through all  $T$  samples.

---

Then  $\bar{\boldsymbol{\mu}} = \boldsymbol{\mu}_T - \frac{1}{T} \Delta_T(\boldsymbol{\mu})$ . Using this approach, we can avoid summing all weights in each step. Weights are averaged only in the last iteration.

## 3.5 Experiments on ImageNet

In this section, we present both full results and their highlights. For general evaluation, various subsets of ImageNet[18] are used: (1) the dataset of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2010 and (2) the subset of ILSVRC 2012 dataset<sup>1</sup>. Both ILSVRC 2010 and 2012 datasets include 1.2 million training images, 50,000 validation images, and 150,000 testing images for different sets of 1000 classes.

Parameters of online learning methods are tuned using each group of validation data as follows:  $C$ s in Perceptron, SGD-SVM, PA, CW, AROW, and NHERD are determined by selecting the best one from  $\{2^{-4}, 2^{-2}, 2^0, 2^2, 2^4\}$ ,  $\eta$ s in CW is determined by selecting from  $\{0.5, 0.6, \dots, 0.9\}$ .  $C$  and  $\eta$  in SCW are determined as explained above. Fundamentally, the numbers of iterations are tuned up to 10 for the best accuracy. All evaluations are repeated five times. Herein, we present the results obtained using the mean and the standard deviation.

### 3.5.1 ILSVRC 2010 Dataset

We use the whole ILSVRC 2010 dataset. To train the classifiers, 1.2M training samples are used after the parameter tuning using 50,000 validation samples. Accuracies are evaluated using 150,000 test samples.

For this dataset, SIFT [89] descriptor and Fisher Vector (FV) [28] are used. We extract each descriptor from a regular grid with step size 6 pixels at multiple patch sizes:  $16 \times 16$ ,  $25 \times 25$ ,  $36 \times 36$ ,  $49 \times 49$ , and  $64 \times 64$ . As a result, tens of thousands of descriptors are extracted for each image. For FV, according to [84], we reduce the dimensions of SIFT to 64 using PCA, and obtain a Gaussian Mixture Model (GMM) with 16 components.

### 3.5.2 ILSVRC 2012 Dataset

ILSVRC 2012 also has test samples, but the ground truth is not provided. Therefore, also for efficiency, we extract a subset. For each class, 100 images are extracted from training data. 5000 validation samples are used for validation. The rest are used for testing.

---

<sup>1</sup><http://www.image-net.org/download>



---

We use four local descriptors: SIFT [89], Local Binary Pattern (LBP) [90], GIST [91], and CSIFT [92], and three mid-level features: BoVW, Locality-constrained Linear Coding (LLC) [74], and FV [28]. The size of patch and the grid width are the same as SIFT from ILSVRC 2010. Our main contribution is to compare online learning algorithms. Comparison with these mid-level features are done in [68], although we also contribute somewhat by comparing them using various local descriptors.

LBP [90] is extracted from  $2 \times 2$  cells in each patch. From each cell, a histogram of 256 bins of local patterns is extracted. Combinations of LBP and gradient based descriptor, such as SIFT, are shown to be effective for large-scale visual recognition [66]. GIST [91] is extracted from  $4 \times 4$  cells in each patch. From each cell, responses from 20 Gabor filters are extracted on R, G, and B channels. Usually, GIST is used for a global feature. This report is the first describing the use of GIST as a local descriptor for mid-level features. CSIFT, one variation of color SIFTs, is a 384-dimensional descriptor that has been shown to perform best for visual recognition in [92].

Each mid-level feature is generated from each descriptor. For FV, we reduce the dimensions of descriptors to 64 using PCA, and obtain a GMM with 256 components. For BoVW and LLC, we learn 2048 codewords using k-means. BoVW and LLC are calculated respectively over  $1 \times 1$ ,  $2 \times 2$ , and  $3 \times 1$  cells according to Spatial Pyramid Matching [11]. Additionally, we reduce the memory usage with Product Quantization [73] according to [67]. We divide mid-level features into each of eight dimension vectors, and generate 256 clusters using k-means. All FVs of training samples are quantized and approximated with the centroids of each cluster when learning.

### 3.5.3 Result 1: Accuracies of not Averaged Classifiers

To compare all algorithms explained in Section 3.3, we first use ILSVRC 2010 dataset. In Figure 3.3 the accuracies of all algorithms without averaging are shown<sup>1</sup>. We also evaluate the algorithms using 100k samples extracted in the same way as the ILSVRC 2012 subset. As shown in Figure 3.3, we found the same trend both in the 1.2M images and in the 100k images: The second-order algorithms (right four algorithms in Figure 3.3) tend to outperform first-order algorithms.

To increase the reliability of our evaluations, we assess all algorithms using the ILSVRC 2012 subset. Moreover, we investigate 12 combinations of local descriptors and mid-level features including SIFT+FV. Figure 3.4 represents the

---

<sup>1</sup>In [84], the accuracy of SIFT+FV with the same parameter is around 25%. The accuracies in Figure 3.3 are slightly different, probably because of the difference of SIFT extraction and GMM training.

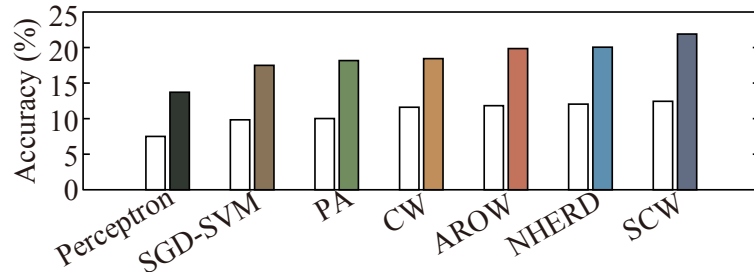


Figure 3.3: Comparison using ILSVRC 2010 1.2M dataset with SIFT+FV. White bars show performance using a 100k subset of ILSVRC 2010.

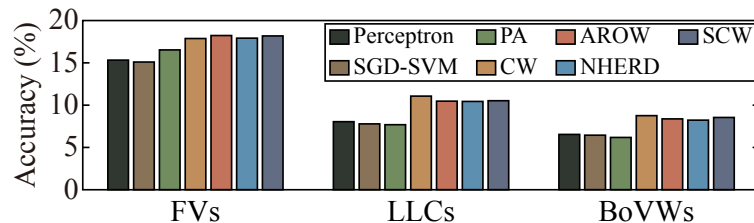


Figure 3.4: Comparison using ILSVRC 2012 subset. Each bar represents the mean accuracy among mid-level features from four descriptors. For example, the accuracy using FVs is the mean of the accuracies using SIFT+FV, LBP+FV, GIST+FV and CSIFT+FV.

accuracies of all algorithms for each mid-level feature. All accuracies for each combination of a local descriptor and a mid-level feature is reported in Table 3.3. In Figure 3.4, to compare the algorithms easily, we average four accuracies from the same mid-level features generated from four descriptors. Again, we can find the superiority of the second-order algorithms. Particularly CW performs best on nine combinations among all 12 combinations of descriptors and mid-level features. AROW and SCW perform best on the three remaining combinations.

### 3.5.4 Result 2: Averaging Does Boost All

When we compare all algorithms without averaging, the second-order algorithms simply seem to outperform the first-order algorithms. However, Figure 3.5 shows that averaging dramatically eliminates the difference of accuracies. The accuracies of second-order algorithms are also boosted, as shown numerically in Table 3.2.

For a comparison of several datasets, we again evaluate the algorithms with

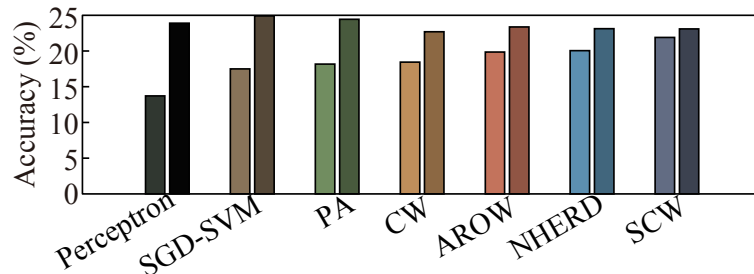


Figure 3.5: Comparison using ILSVRC 2010 1.2M dataset with SIFT+FV. The darker bar for each algorithm shows the accuracy with averaging. The brighter one shows the accuracy without averaging for easy reference.

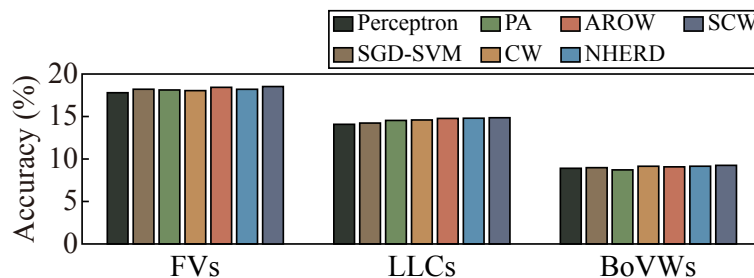


Figure 3.6: Comparison using ILSVRC 2012 subset. Each bar represents the mean accuracy among mid-level features from four descriptors.

averaging on the ILSVRC 2012 subset. To compare the algorithms easily, Figure 3.6 shows the accuracies of all algorithms with averaging for each mid-level feature. All results are shown in Table 3.4. These results show four facts that are little noted in the literature, although averaging itself is a well-known technique. First, second-order algorithms are also boosted for all combinations. Secondly, when averaging is used, SCW performs best instead of CW with many combinations of mid-level features and descriptors. Thirdly, however, the differences among all algorithms are narrowed again. Consequently, first-order algorithms such as Perceptron, SGD-SVM, and PA achieve comparable performance using several combinations including the result obtained using ILSVRC 2010 dataset. Here, the first guideline is concluded: Perceptron can compete against the latest algorithms, but only when averaging is used.

The next question is whether the averaging just hastens the convergence. Online learning algorithms are evaluated in 10 iterations. Therefore, we also stop learning earlier than in the tenth iteration in almost all experiments. To do justice

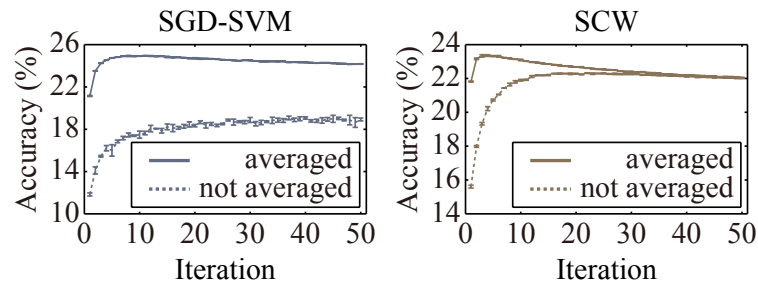


Figure 3.7: Comparison about averaging.

to this inquiry, we continue learning until the 50th iteration on the ILSVRC 2010 dataset.

Figure 3.7 depicts the convergence of SGD-SVM and SCW. The accuracies are evaluated not using training data but using test data. After some iterations, both averaged and not-averaged classifiers seem to reduce their performance on test data, which is true mainly because of an overfit to the training data. Furthermore, the best accuracy of the averaged classifier is better than the not-averaged classifiers, even with many iterations. Averaging not only accelerates the optimization but also improves the generalization accuracy. Therefore, we propose the second guideline: averaging is necessary for any algorithm.

Table 3.2: Accuracy(%) comparison of ILSVRC2010 using SIFT+FV.

	Perceptron	SGD-SVM	PA	CW	AROW	NHERD	SCW
w/o averaging	13.71 ± 0.06	17.49 ± 0.34	18.16 ± 0.17	18.44 ± 0.07	19.86 ± 0.03	20.05 ± 0.12	21.90 ± 0.05
w/ averaging	23.89 ± 0.03	24.92 ± 0.01	24.44 ± 0.02	22.69 ± 0.05	23.37 ± 0.03	23.12 ± 0.02	23.09 ± 0.04

Table 3.3: Accuracy (%) comparison on ILSVRC 2012 *without averaging*. The best accuracy is highlighted for each mid-level feature.

Features	Perceptron	SGD-SVM	PA	CW	AROW	NHERD	SCW
SIFT	17.30 ± 0.34	17.05 ± 0.44	18.72 ± 0.19	<b>20.22</b> ± 0.10	20.10 ± 0.16	19.67 ± 0.20	20.14 ± 0.23
LBP	12.61 ± 0.25	12.60 ± 0.05	13.44 ± 0.37	14.65 ± 0.17	15.98 ± 0.15	15.52 ± 0.22	<b>16.12</b> ± 0.11
GIST	16.08 ± 0.26	15.70 ± 0.35	17.22 ± 0.20	18.39 ± 0.15	<b>18.56</b> ± 0.24	18.46 ± 0.21	18.45 ± 0.23
CSIFT	15.30 ± 0.21	15.04 ± 0.51	16.75 ± 0.30	18.24 ± 0.21	<b>18.28</b> ± 0.22	18.02 ± 0.18	18.01 ± 0.19
SIFT	9.30 ± 0.79	9.01 ± 0.60	8.63 ± 0.67	<b>12.81</b> ± 0.19	11.84 ± 0.56	12.06 ± 0.22	12.12 ± 0.27
LBP	6.52 ± 0.63	6.37 ± 0.36	6.07 ± 0.39	<b>8.98</b> ± 0.50	8.61 ± 0.30	8.43 ± 0.19	8.41 ± 0.49
GIST	10.18 ± 0.64	9.74 ± 0.36	9.96 ± 0.30	<b>13.13</b> ± 0.21	12.41 ± 0.22	12.48 ± 0.51	12.78 ± 0.24
CSIFT	6.18 ± 1.05	6.02 ± 0.20	6.08 ± 0.49	<b>9.35</b> ± 0.34	9.03 ± 0.18	8.77 ± 0.28	8.78 ± 0.56
SIFT	8.04 ± 0.28	7.89 ± 0.17	7.56 ± 0.19	<b>10.78</b> ± 0.17	10.27 ± 0.14	10.20 ± 0.17	10.50 ± 0.15
LBP	4.75 ± 0.21	4.70 ± 0.29	4.48 ± 0.16	<b>6.56</b> ± 0.12	6.14 ± 0.23	6.01 ± 0.14	6.48 ± 0.14
GIST	4.48 ± 0.12	4.32 ± 0.22	4.17 ± 0.19	<b>6.76</b> ± 0.14	6.22 ± 0.17	5.98 ± 0.08	6.28 ± 0.06
CSIFT	8.89 ± 0.29	8.87 ± 0.34	8.52 ± 0.10	<b>10.93</b> ± 0.13	10.87 ± 0.17	10.72 ± 0.24	10.92 ± 0.11

Table 3.4: Accuracy (%) comparison on ILSVRC 2012 *with averaging*. The best accuracy for each mid-level feature is highlighted.

Features	Perceptron	SGD-SVM	PA	CW	AROW	NHERD	SCW
SIFT	19.83 ± 0.22	20.43 ± 0.14	20.24 ± 0.19	20.21 ± 0.09	20.42 ± 0.07	20.13 ± 0.28	<b>20.52 ± 0.10</b>
FV	LBP	14.97 ± 0.27	15.37 ± 0.29	15.11 ± 0.09	16.19 ± 0.13	15.77 ± 0.14	<b>16.43 ± 0.06</b>
	GIST	18.15 ± 0.12	18.65 ± 0.08	18.60 ± 0.07	18.47 ± 0.13	18.68 ± 0.14	<b>18.70 ± 0.17</b>
CSIFT	18.24 ± 0.15	18.34 ± 0.31	18.28 ± 0.25	18.39 ± 0.21	18.44 ± 0.12	18.39 ± 0.04	<b>18.45 ± 0.08</b>
SIFT	16.28 ± 0.16	16.45 ± 0.26	16.85 ± 0.34	17.18 ± 0.18	17.34 ± 0.29	17.32 ± 0.21	<b>17.35 ± 0.18</b>
LLC	LBP	12.06 ± 0.21	12.32 ± 0.12	<b>12.36 ± 0.21</b>	12.00 ± 0.16	12.34 ± 0.22	12.30 ± 0.11
	GIST	15.10 ± 0.13	<b>15.44 ± 0.28</b>	15.16 ± 0.20	14.69 ± 0.11	14.91 ± 0.28	15.08 ± 0.14
CSIFT	12.88 ± 0.26	12.70 ± 0.27	13.78 ± 0.11	14.50 ± 0.22	<b>14.50 ± 0.16</b>	14.49 ± 0.12	14.45 ± 0.18
SIFT	10.54 ± 0.19	10.63 ± 0.14	10.34 ± 0.27	11.14 ± 0.12	10.95 ± 0.13	11.03 ± 0.19	<b>11.26 ± 0.11</b>
BoVW	LBP	6.83 ± 0.26	6.86 ± 0.08	6.72 ± 0.26	6.91 ± 0.16	6.77 ± 0.07	<b>6.94 ± 0.17</b>
	GIST	6.34 ± 0.06	6.36 ± 0.21	6.16 ± 0.23	<b>6.96 ± 0.04</b>	6.81 ± 0.22	6.69 ± 0.06
CSIFT	11.91 ± 0.18	<b>12.06 ± 0.16</b>	11.65 ± 0.12	11.57 ± 0.16	11.77 ± 0.21	12.00 ± 0.20	12.02 ± 0.12

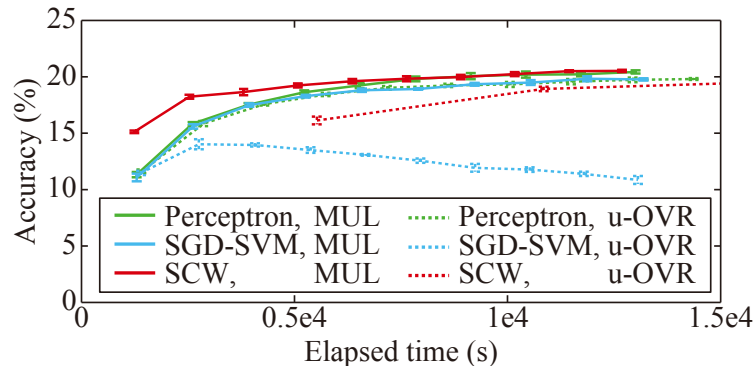


Figure 3.8: Comparison of MUL and OVRs. Dashed lines and solid lines respectively show u-OVR and MUL.

### 3.5.5 Result 3: MUL vs. OVRs

In Figure 3.9, the relation between elapsed time for learning and accuracy using MUL and all OVRs on SIFT+FV of ILSVRC 2012 is shown. The numerical comparison using their best scores is shown in Table 3.5 and Table 3.5.

Figure 3.8 shows the highlights of Figure 3.9 for Perceptron, SGD-SVM, and SCW. First, Perceptron shortly begins to overfit with OVRs. Secondly, especially for second-order algorithms, MUL can converge faster than OVR, mainly because updating the weights becomes the rate-limiting step. Prediction with current weights is rate-limiting for first-order algorithms. Therefore, we propose the third guideline: investigate multiclass learning first.

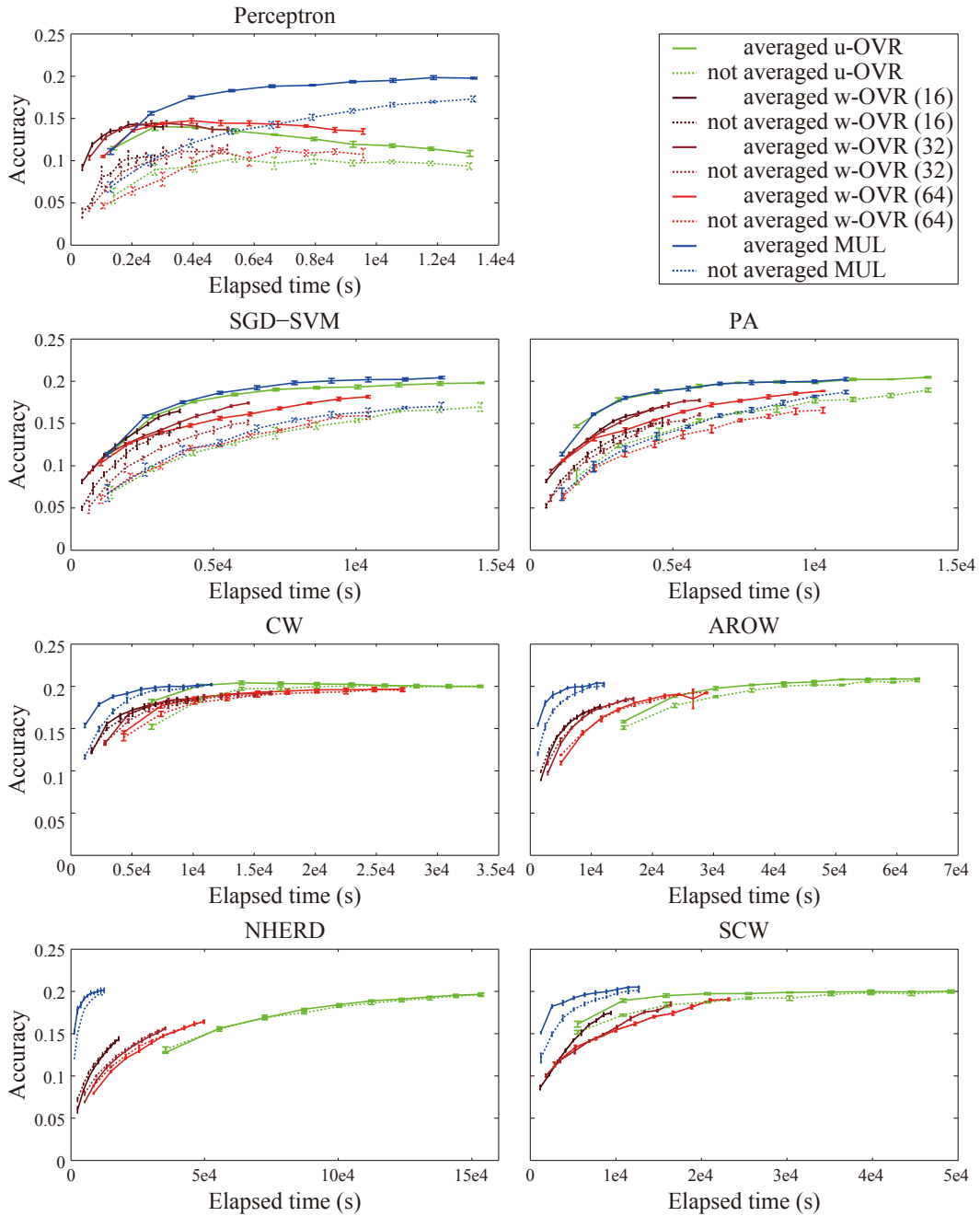


Figure 3.9: Convergence speeds of online learning methods. Solid lines represent accuracies of averaged weights. Dotted lines represent accuracies of non-averaged weights. w-OVR (x) signifies that negative samples are selected randomly x times as often as positive samples.



Table 3.5: Accuracy(%) comparison of ILSVRC2012 using SIFT+FV *without averaging*. MUL and three OVR manners are described in Section 3.4.1. For w-OVR, here, we extracted 32 times as many negative samples as positive samples.

Method	Perceptron	SGD-SVM	PA	CW	AROW	NHERD	SCW
e-OVR	6.58 ± 0.23	8.86 ± 0.41	9.63 ± 0.26	13.14 ± 0.28	11.60 ± 0.18	9.71 ± 0.12	9.69 ± 0.24
w-OVR	10.40 ± 0.81	15.13 ± 0.28	16.06 ± 0.15	19.19 ± 0.20	18.48 ± 0.21	15.72 ± 0.05	18.51 ± 0.23
u-OVR	8.93 ± 0.63	16.96 ± 0.52	18.95 ± 0.23	19.71 ± 0.17	20.74 ± 0.19	19.62 ± 0.21	19.97 ± 0.18
MUL	16.96 ± 0.12	17.05 ± 0.44	18.72 ± 0.19	20.22 ± 0.10	19.96 ± 0.30	19.67 ± 0.20	20.14 ± 0.23

Table 3.6: Accuracy(%) comparison of ILSVRC2012 using SIFT+FV *with averaging*. MUL and three OVR are described in Section 3.4.1. For w-OVR, here, we extracted 32 times as many negative samples as positive samples.

Method	Perceptron	SGD-SVM	PA	CW	AROW	NHERD	SCW
e-OVR	9.98 ± 0.30	10.42 ± 0.06	10.79 ± 0.29	13.06 ± 0.35	11.55 ± 0.23	9.60 ± 0.10	9.71 ± 0.24
w-OVR	14.40 ± 0.33	17.42 ± 0.10	17.74 ± 0.14	19.23 ± 0.14	18.57 ± 0.08	15.58 ± 0.12	18.51 ± 0.23
u-OVR	14.02 ± 0.45	19.81 ± 0.08	20.47 ± 0.09	20.42 ± 0.22	20.88 ± 0.10	19.65 ± 0.12	20.03 ± 0.13
MUL	19.83 ± 0.22	20.43 ± 0.14	20.24 ± 0.19	20.21 ± 0.09	20.42 ± 0.07	20.13 ± 0.28	20.52 ± 0.10

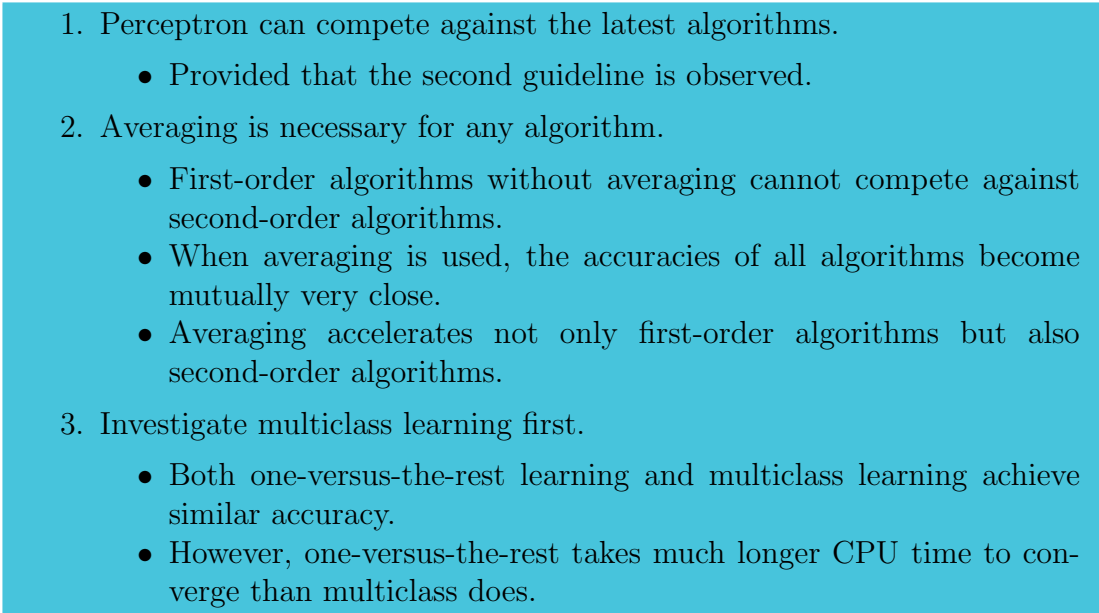
- 
- 
1. Perceptron can compete against the latest algorithms.
    - Provided that the second guideline is observed.
  2. Averaging is necessary for any algorithm.
    - First-order algorithms without averaging cannot compete against second-order algorithms.
    - When averaging is used, the accuracies of all algorithms become mutually very close.
    - Averaging accelerates not only first-order algorithms but also second-order algorithms.
  3. Investigate multiclass learning first.
    - Both one-versus-the-rest learning and multiclass learning achieve similar accuracy.
    - However, one-versus-the-rest takes much longer CPU time to converge than multiclass does.

Figure 3.10: Three guidelines for online learning for large-scale visual recognition.

### 3.5.6 Summarized Guidelines

To realize generic object recognition, large amounts of data are required. Considering scalability, combinations of mid-level features and online learning for linear classifiers are suitable for large-scale visual recognition.

As described in this chapter, we gave qualitative and quantitative comparisons of these online learning algorithms. To date, no report has described a study investigating state-of-the-art algorithms for visual recognition or a study evaluating those algorithms in unified experimental settings. When these algorithms were proposed, toy data and the NLP dataset were used for evaluation. Comparison using conventional settings for visual recognition must be conducted. Finally, this chapter presents three guidelines based on results of image classification, as described in Figure 3.10.

## Chapter 4

# Multi-Keyphrase Problem and Sentence Generation

As described in this paper, we attack a novel problem, the “multi-keyphrase problem”, to address this goal. We hypothesize that image contents can be described with multi-keyphrases, and that a natural sentence can be generated by connecting multi-keyphrases with an experimental grammar model. Existing methods require semantic knowledge such as labels of objects, actions, and scenes. Using these methods, we must strive to prepare a highly organized dataset. Therefore, we propose a novel online learning method for multi-keyphrase estimation. The proposed framework, although simple and scalable, can generate sentences from images with no semantic knowledge. Moreover, the proposed method for multi-keyphrase estimation is applicable to Image Annotation. It achieves state-of-the-art performance. Our experiment using only images and texts demonstrates that the proposed framework is useful for sentence generation from images.

### 4.1 Multi-Keyphrase Estimation as an Annotation Problem

Almost all reported methods [2, 3, 4, 6, 6, 38, 40, 42, 43, 44, 54] rely on assumptions of well-controlled semantic knowledge, objects, actions, scenes, and so on. However, using these methods, it is necessary to manage the label set for each attribute and to associate the labels to each image manually. Consequently, the dataset tends to be small. Moreover, it tends to lack coverage to generate sentences from various images.

In general terms, collecting a large amount of data from the web is a common means to retrieve and process various images. What can be collected automatically are images associated not with semantically clear labels but with surround-

---

ing sentences and uncontrolled words. Indeed, [38] collects a million pairs of an image and a sentential caption from the web for sentence generation from images.

As described in this paper, we present a novel problem, the *Multi-keyphrase Problem*, for sentence generation from images. We seek to learn the relation between *keyphrases* and images from pairs of an image and a sentence, and to obtain a grammar model from all pairs. Consequently, a proposed framework is extendable for sentence generation from large-scale images because a sentence can be generated with no semantic knowledge such as an object, action, or scene.

Given an input image  $I$ ,  $n_p$  keyphrases are estimated. We define the  $i$ -th keyphrase  $K_i$  as:

$$K_i = \{w_1^{K_i}, \dots, w_{n_w}^{K_i}\}, (i = 1, \dots, n_p), \quad (4.1)$$

where  $w$  represents a word. As the first step for multi-keyphrase problem, we regard different word sequences as independent labels. Consequently, as a sub-problem, a multi-keyphrase problem comes down to Image Annotation.

In existing works related to Image Annotation, combinations of metric learning and non-parametric approaches are mainstream.

With non-parametric approaches, an input image is annotated with the labels of images located near the input image according to a certain metric. Although multi-keyphrase estimation using such an approach is conceivable, the complexity for calculating the distances between the input image and all  $N$  training samples is  $O(N)$ . Therefore, the execution time increases with the number of training samples.

Moreover, the state-of-the-art methods require the complexity to be within  $O(N) - -O(N^2)$  for metric learning. Therefore, scalability remains an open question for non-parametric approaches.

To annotate images efficiently, lower time complexity and higher accuracy are required. Therefore, learning one classifier for each keyphrase is apparently a good approach. We need only match a feature of an input image to all classifiers. Therefore, the time for keyphrase estimation is  $O(1)$  (independent of  $N$ ). Although some existing works propose such classifier-based methods [93, 94, 95, 96], their accuracy is inferior to those of non-parametric approaches.

As described in this chapter, we interpret a classifier-based approach as inferior to a non-parametric approach because the learning methods for classifiers are unsuitable for multi-label problems. Existing works mostly use binary classification such as SVM with a one-vs.-the-rest manner. The classifier for a label is obtained by regarding images associated with the label as positive samples and the rest images as negative samples. Furthermore, labels are output according to the scores from the binary classifiers. Nevertheless, no guarantee exists that the output of SVMs for different classifiers will have appropriate scales.

---

Therefore, we modify existing learning methods for a multi-keyphrase problem in which each image is associated with more than one label. The proposed method is formulated by improving online multiclass learning so that more than one label for a training sample is useful efficiently. This method, which is applicable not only to multi-keyphrase estimation but also to Image Annotation, achieves state-of-the-art performance on some benchmark datasets.

The remainder of this chapter is organized as follows: Section 4.2 describes related works. Section 4.3 presents the proposed framework and a practical method for estimating multi-keyphrase. Section 4.4 shows that our method obtains state-of-the-art accuracy on the datasets for Image Annotation.

## 4.2 Related Methods for Annotation

This section introduces related works for Image Annotation, a subproblem of the presented multi-keyphrase problem.

A non-parametric approach using training labels of neighbor samples for annotation, is undertaken in [29].

Recently, almost all existing works have adopted widely various image features. Joint Equal Contribution (JEC) [30] uses color histograms and wavelets, and computes the distances from an input image and all training samples with a proper metric. For example,  $\chi^2$  distance is ideal for histograms, and  $L_1$  distance is ideal for the wavelets. All distances are normalized and are simply summed up to search for neighbor samples. Although simple, [30] achieved state-of-the-art performance in 2008 when the method was proposed. TagProp [31] uses 15 features such as Gist [91], Bag-of-Features representation of SIFT [89], and color histograms. TagProp learns the proper weight of each distance to obtain the best performance, whereas JEC simply adds those distances. Moreover, TagProp learns the weight for each label in neighbor samples because minor labels that are rarely associated with images are not often used for annotation. TagProp applies the logistic discriminant model to learn the weight of labels. Therefore, TagProp, proposed in 2009, is recorded as showing state-of-the-art performance on several benchmark datasets up to 2012. In 2011, [97] 43 proposed features from foreground and background were inferred according to saliency, and now are used along with TagProp. In some cases, [97] yields slightly superior performance to that of TagProp. In [46], Canonical Contextual Distance (CCD) is proposed. With CCD, low-dimensional space is computed using probabilistic canonical correlation analysis with image features and label features so that correlations between image features and label features might be largest. Consequently, distances between image features are improved by virtue of similarities among labels; the CCD performance is comparable to that of TagProp [32].

---

However, the more important problem with a non-parametric approach is the computational cost. As described in Section 4.1, calculating all distances between an input image and all training samples is necessary for annotation. Therefore, the time used for annotation increases linearly with the data amount. Regarding the computational cost for the learning metric, JEC is constant to the data amount because it needs only feature extraction. TagProp and CCD require all distances between training samples for metric learning. Although they make efforts so that learning can be performed only with distances among all samples and a part of all samples, the number of a part of all samples will increase when the whole dataset increases. Consequently, they lack scalability as the training time increases to become more than linear to the data amount.

To realize scalability, the usage of a linear classifier appears to be a promising method. When each image has only a single label, the problem of estimating the label is designated as Image Classification. In this area, as pipeline alternatives to Bag-of-Features of SIFT and Kernel SVM, richer image representation than Bag-or-Features and linear SVM is actively pursued [28, 74]. With these image representations, feature extraction and learning for classification can be done linearly to the data amount. Moreover, the time for label estimation is constant to the data amount.

However, existing work using SVM for annotation is inferior to JEC proposed in the same year.

Binary classification with SVM is inappropriate for multiclass settings such as annotation. As described in Section 4.1, in the one-vs.-the-rest technique, all samples are divided into positive or negative for each class. However, it renders positive and negative samples as highly imbalanced, and no guarantee exists that the output of SVMs for different classifiers will have appropriate scales.

Recent studies of large-scale Image Classification adopt online learning for linear classification as described in Chapter 1. Indeed, Weston [64] proposes an online learning method for Image Annotation. The authors devote their attention to developing an algorithm that fits on a laptop. Although various approximations are included, their method outperforms the approximated nearest neighbor and binary PA.

As described in Chapter 1, we have three guidelines for large-scale Image Classification:

In this chapter, therefore, we propose a novel learning method for training samples with multiple labels by generalizing Passive–Aggressive (PA) [59] to achieve state-of-the-art accuracy and scalability. The method is called Passive–Aggressive with Averaged Pairwise Loss (PAAPL). Moreover, other online algorithms of multi-classification Passive–Aggressive (PA) [59], and Normal distribution HERDing (NHERD) [62]—are investigated.

---

## 4.3 Sentence Generation with Multi-keyphrase Estimation

In this section, we first describe the means to extract the keyphrases to be learned. Secondly we describe the methods used to estimate keyphrases. Finally, we explain the method used for spinning keyphrases into one sentence.

### 4.3.1 Multi-keyphrase Extraction

Learning with noisy keyphrases that are unrelated to images degrades the accuracy of multi-keyphrase estimation. Therefore, we investigate a filter to extract keyphrases from each sentence of an image. We want to select phrases that are (i) related to the image contents and that (ii) appear in many sentences. Consequently, we investigate two filters.

**Local TF-IDF filter.** We regard the typicality of each phrase as a clue to how the phrase relates to the image’s contents. For example, some phrases such as “is a” and “this is” appear frequently though the phrases, but they are apparently irrelevant to the contents. Conversely, “living room” and “black dog” are expected to be typical phrases. Therefore, we regard phrases that appear many times in the sentence (high Term Frequency) and which rarely appear in other sentences (Inverse Document Frequency) as typical. Therefore, top- $n_k$  TF-IDF [98] valued phrases are extracted.

**Global DF filter.** With local TF-IDF filter, overly rare phrases are unsuitable for classifier learning because small samples for a phrase might engender low accuracy. Here we make much of the sample size for each keyphrase to learn it stably. We extract phrases only according to bounds of  $n_d$ , the number of samples in which each phrase appears.

### 4.3.2 Methods of Multi-keyphrase Learning

Whichever filter we use, some noisy phrases exist in extracted keyphrases. To learn the classifiers for each keyphrase from various images, requirements are not only the compatibility of scalability for the data amount and accuracy for keyphrase estimation, but also the tolerability of noise.

Given the  $t$ -th training sample  $\mathbf{x}_t \in \mathbb{R}^d$  associated with a label set  $Y_t$ , a subset of  $\mathcal{Y} = \{1, \dots, n_y\}$ , it is classified with the present weight vector  $\boldsymbol{\mu}_t^{y_i}$  ( $i = 1, \dots, n_y$ )<sup>1</sup> as:

$$\hat{y}_t = \operatorname{argmax}_{y_i} \boldsymbol{\mu}_t^{y_i} \cdot \mathbf{x}_t. \quad (4.2)$$

---

<sup>1</sup>Here, the bias  $b$  is included in  $\boldsymbol{\mu}_t$  as  $\boldsymbol{\mu}_t^\top \leftarrow [\boldsymbol{\mu}_t^\top, b]$  by redefining  $\mathbf{x}_t^\top \leftarrow [\mathbf{x}_t^\top, 1]$

---

If necessary, multiple labels are estimated in score order.

Multi-labeling for one sample is applicable by defining  $n_y > 1$ . Here, hinge-loss  $\ell$  is given as:

$$\begin{aligned} & \ell(\boldsymbol{\mu}_t^{r_t}, \boldsymbol{\mu}_t^{s_t}; (\mathbf{x}_t, Y_t)) \\ = & \begin{cases} 0 & \boldsymbol{\mu}_t^{r_t} \cdot \mathbf{x}_t - \boldsymbol{\mu}_t^{s_t} \cdot \mathbf{x}_t \geq 1 \\ 1 - (\boldsymbol{\mu}_t^{r_t} \cdot \mathbf{x}_t - \boldsymbol{\mu}_t^{s_t} \cdot \mathbf{x}_t) & \text{otherwise} \end{cases}, \end{aligned} \quad (4.3)$$

where  $r_t = \operatorname{argmin}_{r \in Y_t} \boldsymbol{\mu}_t^r \cdot \mathbf{x}_t$  and  $s_t = \operatorname{argmax}_{s \notin Y_t} \boldsymbol{\mu}_t^s \cdot \mathbf{x}_t$ .

In this chapter, we use Passive–Aggressive (PA) and NHERD as baseline methods. Although both are explained in Chapter 3, we describe the formulation of PA because our proposed method is based on PA.

#### 4.3.2.1 Passive–Aggressive

PA is an online learning method for binary and multiclass classification, regression, uniclass estimation, and structure estimation. The salient benefit of PA is that the update coefficient is analytically calculated according to the loss. In contrast, SGD based methods and traditional perceptron require design of the coefficient.

Here we seek to decrease the hinge-loss of multi-classification and not to change the weight radically. Consequently, we obtain the following formulation.

$$\begin{aligned} & \boldsymbol{\mu}_{t+1}^{r_t}, \boldsymbol{\mu}_{t+1}^{s_t} \\ = & \operatorname{argmin}_{\boldsymbol{\mu}^{r_t}, \boldsymbol{\mu}^{s_t}} \|\boldsymbol{\mu}^{r_t} - \boldsymbol{\mu}_t^{r_t}\|^2 + \|\boldsymbol{\mu}^{s_t} - \boldsymbol{\mu}_t^{s_t}\|^2 + C\xi, \end{aligned} \quad (4.4)$$

$$\text{s.t. } \ell(\boldsymbol{\mu}^{r_t}, \boldsymbol{\mu}^{s_t}; (\mathbf{x}_t, Y_t)) \leq \xi \text{ and } \xi \geq 0. \quad (4.5)$$

Therein,  $\xi$  denotes a slack variable representing the bound of the loss.  $C$  signifies a parameter to reduce the negative influence of noisy labels. It can be derived using Lagrange’s method of undetermined multipliers. Therefore we obtain:

$$\boldsymbol{\mu}_{t+1}^{r_t} = \boldsymbol{\mu}_t^{r_t} + \tau_t \cdot \mathbf{x}_t, \quad \boldsymbol{\mu}_{t+1}^{s_t} = \boldsymbol{\mu}_t^{s_t} - \tau_t \cdot \mathbf{x}_t, \quad (4.6)$$

$$\tau_t = \min\{C, \ell(\boldsymbol{\mu}_t^{r_t}, \boldsymbol{\mu}_t^{s_t}; (\mathbf{x}_t, Y_t)) / (2\|\mathbf{x}_t\|^2)\}. \quad (4.7)$$

This PA is called PA-I in [59]. NHERD has a generalized form of PA-II. PA and SGD-SVM have a closed form. Indeed, PA for binary classification and SGD-SVM without  $L_2$  regularization have the same update rule. Differences between SGD-SVM and PA-I here are (1) binary or multi-class, (2) regularization form, and (3) the number of parameter to be tuned. Consequently, not only SGD-SVM but also PA can be regarded as online learning for SVM.



---

### 4.3.2.2 Passive–Aggressive with Averaged Pairwise Loss

Both PA and NHERD are online learning methods for classification, but they present no problem if a sample is associated with multiple labels. Indeed, the Passive–Aggressive Model for Image Retrieval (PAMIR) [99] is proposed by application of PA to image retrieval.

However, they treat only one relevant label and one irrelevant label. Apparently, classifiers of some labels are not well updated. That convergence becomes delayed.

Therefore, we propose a novel online learning algorithm for which multiple labels are attached to one sample. As discussed in Section 4.2, general online learning methods consist of two steps: classification of the  $t$ -th sample, and update of the  $t$ -th classifiers. Given the  $d$ -dimensional weight vectors  $\boldsymbol{\mu}$  for all  $n_y$  labels, the complexity for classification of a sample is  $O(dn_y)$ , whereas the complexity for update of a classifier is  $O(d)$ . If we update all classifiers with given labels  $Y_t$ , then its complexity becomes  $O(d|Y_t|)$ . In Image Annotation and especially sentence generation, we can assume  $n_y \gg |Y_t|$ . Therefore, because classification is the rate-controlling step, total computation time remains much the same whether we update one classifier or  $|Y_t|$  classifiers. Figure 4.1 shows the conceptual difference between hinge-loss and the loss used in the proposed method. Consequently, the proposed PAAPL achieves efficiency by averaging all pairwise losses between relevant and irrelevant labels.

1. Given a  $t$ -th image, define label set  $\bar{Y}_t$  of  $n_y$  labels by selecting highly scored and irrelevant labels.
2. Randomly select one relevant label  $r_t$  from  $Y_t$  and one irrelevant label  $s_t$  from  $\bar{Y}_t$ .
3. Based on a hinge-loss between  $r_t$  and  $s_t$ ,  $1 - (\boldsymbol{\mu}_t^{r_t} \cdot \boldsymbol{x}_t - \boldsymbol{\mu}_t^{s_t} \cdot \boldsymbol{x}_t)$ , update classifiers according to PA.

Additionally, we investigate a way to reduce the complexity  $O(dn_y)$  for the classification step. In [64], the approximation of a loss function by the random selection of labels is an important step for online learning when using less powerful computers. Although random selection might miss incorrectly classified labels at a higher rate, it was verified experimentally that correct classifiers are obtainable eventually. Therefore, we also adopted random selection.

1. Randomly select one relevant label  $r_t$  from  $Y_t$ .
2. Define irrelevant label  $s_t$  with random selection from  $Y_t$  and compute the hinge-loss  $1 - (\boldsymbol{\mu}_t^{r_t} \cdot \boldsymbol{x}_t - \boldsymbol{\mu}_t^{s_t} \cdot \boldsymbol{x}_t)$ . Continue selecting  $s_t$  until the loss becomes positive.
3. If the hinge-loss becomes positive, update classifiers for  $r_t$  and  $s_t$  according to PA; otherwise proceed to the next training sample.

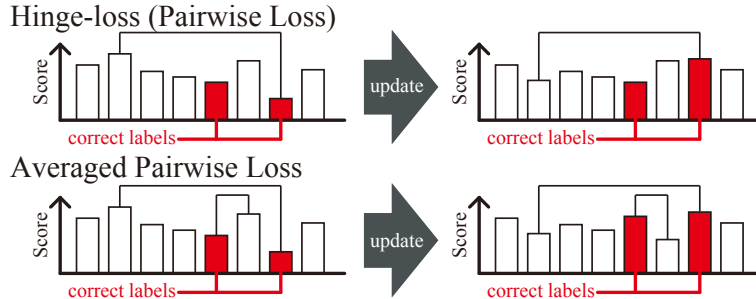


Figure 4.1: Comparison of hinge-loss and averaged pairwise loss.

### 4.3.3 From Multi-keyphrases to a Sentence

Generated sentences for an input image  $I_t$  are expected to include many keyphrases and must be grammatically correct. We can solve a sentence-generation problem by minimizing the sum of costs: phrase cost  $\phi_p(w_1, \dots, w_n)$ , and length cost  $\phi_l(l)$ . The optimization problem is:

$$\{w_1, \dots, w_l\} = \operatorname{argmin}_{w_1, \dots, w_l} \phi_l(l) + \lambda_p \sum \phi_p(w_1, \dots, w_n), \quad (4.8)$$

where  $\lambda_p$  is weight parameter for phrase cost. This problem can be regarded as integer programming, but the length cost complicates this problem. Therefore, we investigated the effects of relaxation using multi-stack beam search modified in [1]. Beam searching saves some high rank candidate sentences in one stack. The stacks are divided according to the length of the generated sentence. The highest-ranked sentence wedged between two edge-of-sentences (EOS), which is a special word representing the edge of a sentence, is chosen as the output. If there are too few contents compared to the target length, keyphrases are expended quickly, leading a large increase in grammar cost and ultimately shortening the generated sentences. Consequently, short sentences are generated for images with little content.

The remainder of this section presents a description of the details of the cost terms/functions.

#### 4.3.3.1 Phrase Cost

The phrase cost is calculated with two probability functions. Given a phrase  $\{w_1, \dots, w_n\}$ , the phrase is evaluated with the keyphrase probability  $P_k(w_1, \dots, w_n | I_t)$  if the phrase is an estimated keyphrase  $\mathcal{K}$  for the input image  $I_t$ . Otherwise, the

---

phrase is evaluated with grammar probability  $P_g(w_{n_g}|w_1, \dots, w_{n_g-1})$ .

$$\phi_p(w_1, \dots, w_n) = \begin{cases} -\log P_k(w_1, \dots, w_n|I_t) & \{w_1, \dots, w_l\} \in \mathcal{K} \\ -\log P_g(w_n|w_1, \dots, w_{n-1}) & \text{otherwise} \end{cases}. \quad (4.9)$$

For grammar probability, we investigate a well-known N-gram model that uses statistical frequency  $P_g(w_n; w_1, \dots, w_{n-1})$  to estimate a word  $w_n$  after a word sequence  $\{w_1, \dots, w_{n-1}\}$ . The N-gram model is a common method for natural language processing (e.g., statistical machine translation).

Alternatively, the combination of two probabilities such as  $\phi_p = -\log P_k - \lambda_g \log_g P_g$  with a weight parameter  $\lambda_g$  is plausible. As described in this paper, however, we simply encourage the maximal use of keyphrases by defining  $P_k(w_1, \dots, w_n|I_t) = 1$  to investigate the influence of keyphrase estimation on the accuracy of generated sentences.

#### 4.3.3.2 Length Cost

Using the phrase cost alone tends to generate sentences that are too short because the cost sum increases concomitantly with sentence length. Therefore, we introduce the following length cost with the target length  $l_0 = 10 + 2$  EOS as:

$$\phi_l(l) = -\log P_l(l), \quad (4.10)$$

where  $P_l \propto \mathcal{N}(l_0, \sigma_0)$ , and  $\sigma_0$  represents the strictness of length.

## 4.4 Evaluation of Multi-keyphrase Estimation

In this section, we evaluate the proposed method for multi-keyphrase estimation. Particularly, we investigate performances of Image Annotation using three benchmark datasets.

### 4.4.1 Experiment Setting

We use the following three de-facto standard datasets. As described in [31], Corel 5k is easy to learn. What must be done is comparison of the performance on more than one dataset.

**Corel 5k.** Corel 5k [8] consists of about 4500 training samples and 500 test samples. Each training sample is associated with 3.4 labels, on average, from 260 labels.

---

**ESP Game.** ESP Game [13] has about 60,000 images that are manually annotated through a game. Existing works for annotation [30, 31] select about 20,000 images. Each image is associated with 4.7 labels, on average, from 268 labels.

**IAPR-TC12.** IAPR-TC12 [14] is released for image retrieval between several languages. Existing works for annotation [30, 31] select about 20,000 images. Each training sample is associated with 5.7 labels, on average, from 291 labels.

As an image feature, we use Fisher Vector (FV) [28] from SIFT [89]. We extracted SIFT descriptor from a regular grid with step size of 6 pixels at multiple scales:  $16 \times 16$ ,  $25 \times 25$ ,  $36 \times 36$ ,  $49 \times 49$ , and  $64 \times 64$ . Then we reduce the dimensions to 64 using PCA, and obtain a Gaussian Mixture Model (GMM) with 256 components. Then FV is calculated respectively over  $1 \times 1$ ,  $2 \times 2$ , and  $3 \times 1$  cells.

As ESP Game and IAPR-TC12 has many data, we reduce the memory usage with Product Quantization (PQ) according to [67]. We divide FV into each of eight dimension vectors, and generate 256 clusters using k-means method. All FVs of training samples are quantized with those clusters, and approximated with the centroids of each cluster when learning with the training sample.

For a fair comparison of annotation method, we also use 15 features provided in the paper of TagProp [31]. To combine different features, we independently learn the classifiers for each feature at first, and estimate labels for test samples with the sum of the scores from classifiers of each feature.

We evaluate the performance for annotation with the following three indicators. Given a sample, we define the number  $a$  as correctly estimated labels,  $b$  as correct labels, and  $c$  as estimated labels.

**Precision ( $P$ ).** The ratio of correctly estimated labels to estimated labels, i.e.,  $P = a/c$ .

**Recall ( $R$ ).** The ratio of correctly estimated labels to correct labels i.e.,  $R = b/c$ .

**F-measure ( $F$ ).** Because a tradeoff exists between Precision and Recall, we use the harmonic average:

$$F = \frac{2 \times P \times R}{P + R} = \frac{2ab}{(a + b)c}. \quad (4.11)$$

Following existing works, we fix  $c = 5$ . Additionally, we multiply all indicators by 100, i.e.,  $0 \leq P, R, F \leq 100$ .

#### 4.4.2 Experiment Result on Benchmark Datasets

We present the result of annotation with Corel 5k in Table 4.1. TagProp43 [97] represents the integration of 43 features proposed in [97] and metric learning

Table 4.1: Comparison of performance in terms of  $P$ ,  $R$ , and  $F$  on Corel 5K.

	$P$	$R$	$F$
JEC [30]	27	32	29
Matrix Factorization [93]	29	29	29
TagProp [31]	33	42	37
CCD [32]	36	41	38
TagProp43 [97]	35	41	37
PAAPL on 15 features	<b>40</b>	<b>57</b>	<b>47</b>
PAAPL on FV	<b>44</b>	<b>62</b>	<b>51</b>

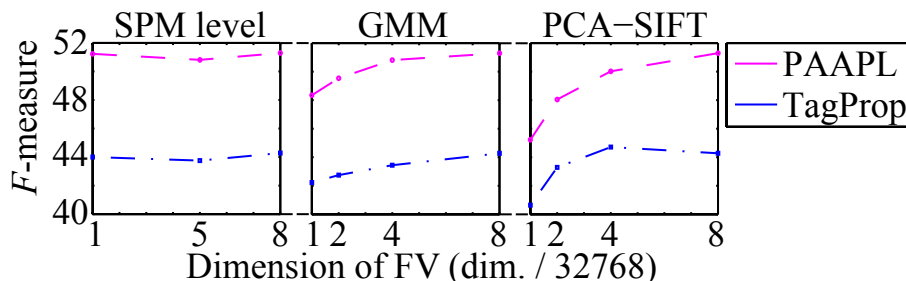


Figure 4.2: Comparison between PAAPL and TagProp with lower dimensional FVs.

with TagProp. PAAPL on 15 features uses 15 features provided in the paper of TagProp [31]<sup>1</sup>. As shown in Table 4.1, we outperform existing works. Even if TagProp’s 15 features are used, the performance of PAAPL is superior to that of TagProp.

Moreover, the combination of proposed PAAPL and FV achieves the best  $F$ -measure of 51. However, the combination of FV and TagProp is not reported. Moreover, high-dimensional features are incompatible with neighbor search. Therefore, we reduce the dimensions of FV in the following manner and compare the performance of PAAPL and TagProp.

**SPM level.** Reduce the SPM cells from  $1 \times 1 + 2 \times 2 + 3 \times 1$  (262144 dim.) to  $1 \times 1 + 2 \times 2$  (163840 dim.), and to  $1 \times 1$  (32768 dim.).

**GMM component.** Reduce the components of GMM from 256 (262144 dim.) to 128 (131072 dim.), to 64 (65536 dim.), and to 32 (32768 dim.).

**PCA-SIFT dimension.** Reduce the dimensions with PCA from 64 (262144 dim.) to 32 (131072 dim.), to 16 (65536 dim.), and to 8 (32768 dim.).

<sup>1</sup>Features and codes of TagProp are provided on <http://lear.inrialpes.fr/pubs/2009/GMVS09/>.

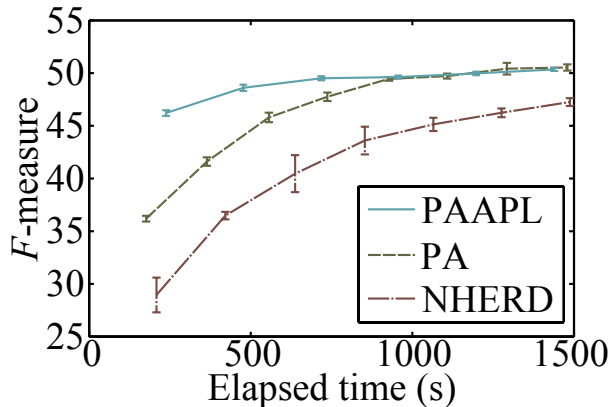


Figure 4.3: Convergence comparison conducted among online learning methods.

A comparison of results is shown in Figure 4.2. First, the combination of FV and TagProp is also superior to existing results on Corel 5k. Secondly, SPM does not affect the performance on Corel 5k because FV for whole image itself has sufficient isolation for this dataset. Thirdly, FV with 32 dim. PCA-SIFT is better than full FV for TagProp. Finally, with all FVs, PAAPL outperformed TagProp.

Next we compare the respective convergence speeds of PAAPL, PA, and NHERD. All methods are implemented using MATLAB. Figure 4.3 shows the performance at every iteration. The proposed PAAPL obtains good performance with few iterations. Such a trend will become pronounced in multi-keyphrase estimation using many more labels. Indeed, such a result is obtained in Section 4.5.

We present comparison results of ESP Game and IAPR-TC12 in Table 4.2. Results show that PAAPL is comparable to existing methods on IAPR-TC12 and that it outperforms them on ESP Game. The computational time for learning is proportional to the data amount. That for annotation is constant in relation to the data amount. Because the proposed method achieves a better score than TagProp does, the scalability of which is a problem, PAAPL is shown to be competitive.

## 4.5 Experiment and Discussion of Sentence Generation

In this section, PASCAL Sentence [6] is used for experimentation. It has 20 categories, each with 50 images. Each image has about five captions of around 10 words each. This dataset is generated using crowdsourcing in [37]. We split

Table 4.2: Comparison of performance in terms of  $P$ ,  $R$ , and  $F$  on ESP Game and IAPR-TC12.

	ESP Game			IAPR-TC12		
	$P$	$R$	$F$	$P$	$R$	$F$
JEC [30]	22	25	23	28	29	28
TagProp [31]	39	27	32	46	35	40
CCD [32]	36	24	29	44	29	35
TagProp43 [97]	43	23	30	-	-	-
PAAPL on 15 features	32	<b>35</b>	<b>33</b>	40	<b>38</b>	39
PAAPL on FV	36	<b>40</b>	<b>38</b>	38	<b>36</b>	37

Table 4.3: Statistics of all phrases and filtered phrases with local TF-IDF. The upper side of each cell presents the number of training samples including at least one keyphrase. The lower side presents the number of keyphrases.

Local TF-IDF					All
top-1	top-3	top-5	top-7	top-9	$\infty$
900	900	900	900	900	900
764.6	1983.6	3117.0	4504.2	6111.0	15443.2

the dataset into training data and test data according to [1]: 45 pairs of an image and a text per category are chosen as training samples. All splits are defined randomly five times. Then the scores are averaged.

We regard to [1], which also uses only PASCAL Sentence, as a baseline. Therefore, we extract the same features and perform the same dimension reduction as a preprocess. As an image feature for the shape information, we extracted a SIFT descriptor from a regular grid with step size 6 pixels at  $16 \times 16$  scale. Then LLC with 1024 codewords is calculated respectively over  $1 \times 1$ ,  $2 \times 2$ , and  $4 \times 4$  cells. For the texture information, we extract HLAC [100], Gist [91], and LBP [90].

Additionally, as [1] does, we extract word TF-IDF vectors as text features and reduce the dimension of image feature with Canonical Contextual Distance (CCD) [46]. Because CCD is a name for a metric, we refer to compressed features as Canonical Contextual Coordinate ( $C^3$ ). Keyphrases are estimated through neighbor search in this space in [1]. Therefore, we refer to [1] as  $C^3$ +knn.

We extract two-word sequences as keyphrases and learn the classification by PA, NHERD, and PAAPL with 10 iterations on  $C^3$ . Then keyphrases are connected with the grammar model consisting of a bigram and trigram. We adopt standard back-off smoothing because the training samples are too few to esti-

Table 4.4: Statistics of filtered phrases with global DF. The upper side of each cell presents the number of training samples include at least one keyphrase. The lower side presents the number of keyphrases.

Bound		Upper		
		300	200	100
Lower	5	900.0	900.0	900.0
		1226.6	1221.8	1207.8
	10	900.0	900.0	899.6
		500.6	495.8	481.8
	50	889.0	863.6	694.6
		43.4	38.6	24.6

Table 4.5: Accuracy of multi-keyphrase estimation and of generated sentence for each method.

	<i>P</i>	<i>R</i>	<i>F</i>	BLEU	NIST
C <sup>3</sup> +knn (baseline)	2.6	0.6	1.0	3.59±0.82	2.03±0.04
C <sup>3</sup> +knn+N-gram	2.6	0.6	1.0	3.68±0.67	2.01±0.03
C <sup>3</sup> +PA+N-gram	6.4	1.5	2.5	4.90±1.50	2.31±0.20
C <sup>3</sup> +NHERD+N-gram	6.4	1.5	2.4	4.99±1.07	2.20±0.21
C <sup>3</sup> +PAAPL+N-gram (proposed)	<b>20.2</b>	<b>4.9</b>	<b>7.8</b>	<b>6.15±2.07</b>	<b>2.34±0.11</b>
FV+PAAPL+N-gram (proposed)	<b>26.9</b>	<b>6.4</b>	<b>10.4</b>	<b>7.52±1.89</b>	<b>2.65±0.18</b>

mate an N-gram accurately. We define  $\lambda_p = 0.05$ , and estimate five keyphrases. Alternatively, we extract FV with the same protocol in Section 4.4 and generate sentences through multi-keyphrase learning.

Keyphrases are extracted according to the filter described in Section 4.3. The statistics of the number of keyphrases and usable training samples are presented in Table 4.3 and Table 4.4: for statistics with no filter and local TF-IDF filter, and for the statistics with global DF filter, respectively. The number of top phrases to be used,  $n_k$ , is defined 1, 3, 5, 7, or 9. The bounds of  $n_d$ , the number of samples where each phrase appears are defined so that the number of samples for each phrase might be ensured and so that overfrequent (and meaningless) phrases such as “is-a” might be eliminated.



---

### 4.5.1 Examples of Generated Sentences

Figure 4.4 presents some examples of generated sentences. As shown, sentences can be generated using as many keyphrases as the method can. We can learn keyphrases about spatial relations (e.g., “in front”) because we extract features using Spatial Pyramid and grid cells and preserve spatial information. Moreover, words that do not appear in the keyphrases are useful in interpolated from keyphrases found using the N-gram model.

### 4.5.2 Automatic Evaluation

Automatic evaluation for generated sentence is important and difficult. Some studies [2, 4, 6, 35, 38, 42, 43, 54, 55, 56] evaluate systems that assess sentences using humans instead of automatic evaluation. However, it is extremely difficult to compare the performance without automatic evaluation. Automatic evaluation is difficult mainly because of the large variety of representations for one image. Even if each image has one correct sentential caption, as do other benchmark datasets for other image recognition, processing word order variation and synonyms appropriately is difficult.

Some works [4, 36, 38] use automatic evaluations for statistical machine translation. In addition, for statistical machine translation, evaluation adopting word order variation and synonyms is ideal. Evaluation methods such as BLEU [101] and NIST [102] subsume that various reference translations are associated with one test datum. The translation output is evaluated in terms of the degree to which word sequences are common in output and references. Consequently, because word order variation and synonyms are partly adopted, correlation between automatic evaluations and human evaluations exists. [4] shows low correlation between BLEU and human evaluation, but this is true because references of the types in PASCAL Sentence and their Baby Talk differ greatly: PASCAL Sentence describes the main content of each image, whereas Baby Talk describes all objects’ names, numbers, and their local relations.

Therefore, we use BLEU and NIST for evaluation<sup>1</sup> with the default order of N-grams for BLEU (up to 4-gram) and NIST (up to 5-gram). Using multiple evaluations, the performance comparison is more reliable.

---

<sup>1</sup>Particularly we use the evaluation tool provided by NIST (<http://www.itl.nist.gov/iad/mig/tools/>).

	Input Image	Keyphrases	Sentence	Input Image	Keyphrases	Sentence
Successfully Generated		field EOS front of in front a black tracks EOS	A black and white cow in front of a man.		group of a group EOS two at a front of	Front of a group of people sitting at a table.
		and a sitting on a woman in front front of	Front of a woman in front of people sitting on.		a brown water EOS field EOS brown horse a horse	Sandy field with a brown horse standing in a horse.
Keyphrases are Appropriate		and a room with a black computer EOS a computer	Room with a computer and a black and white photo.		front of in front street EOS the street a city	Front of a city skyline in front of the street.
		front of in front is parked bus is a large	E1 bus is parked in front of a large black.		and a room with EOS an dining room front of	A furnished dining room with gardens in front of a.
		a table table EOS at a a group a man	Table with a group of a man in a table.		room with living room and a a small table EOS	Room with a small black and a black and white.
Humorous Mistakes		close up close-up of a close a brown the camera	A brown horse with a close up of the camera.		grass EOS a field a white field EOS the grass	A white cat laying on a field in the grass.
		at a a group group of a man and a	At a man and a group of people sitting around.		sitting on at a a computer and a a table	At a blackbird sitting on a table and a computer

Figure 4.4: Examples of estimated keyphrases and generated sentences. The first row depicts successful examples. The second row have partly correct examples thanks to appropriate keyphrases. The last row includes humorous mistakes.

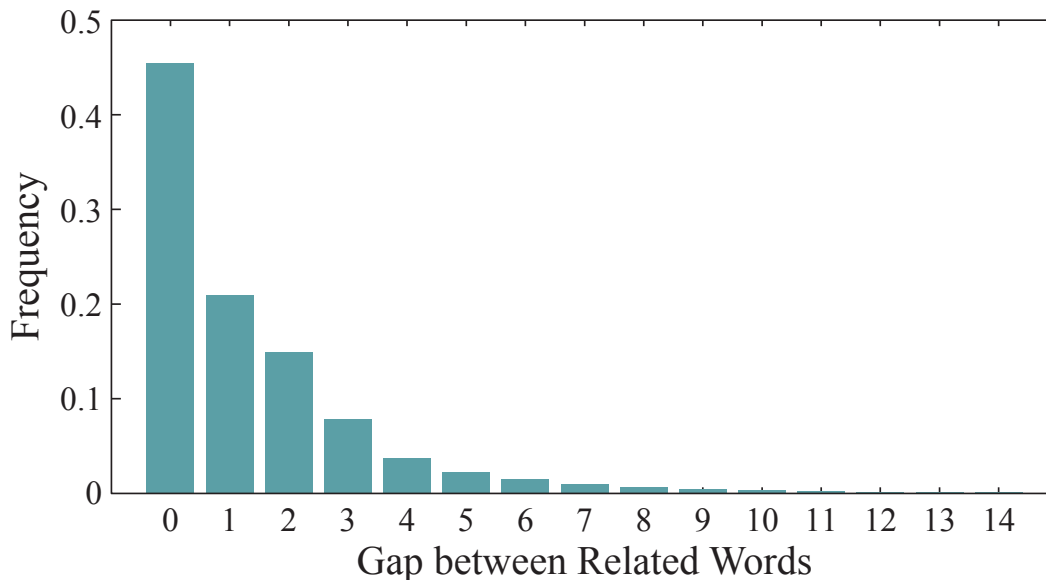


Figure 4.5: Frequencies of the word gaps between grammatically related words.

### 4.5.3 Discussion about Keyphrase Extraction

In this subsection, we discuss the definition of the keyphrases and how to extract them from the experimentally obtained results.

In this chapter, we define keyphrases as sequences of two continuous words. However, it is not clear that sequences of two continuous words have the most relations among objects, actions, and attributes in the input images.

Therefore, we first investigate whether the most relations in the sentences are included in two continuous words or not. Particularly, we parse all sentences in PASCAL Sentence dataset using Stanford Parser [103]. The frequencies of the word gaps between grammatically related words are shown in Figure 4.5. About half of the relations between two words are extracted from two continuous words. Table 4.6 shows the relations which frequently occur in two continuous words. We also find that the relations extracted from two discontinuous words include many “prep\_\*” relations. For example, relation “prep\_in” is found from “airplane in flight”. Although “airplane” and “flight” are distant from each other, these three words can be restored by estimating two keyphrases, “airplane in” and “in flight”. Consequently, “prep\_\*” relations can also be represented by two continuous words. Over half of the relations are included in two continuous words. Therefore, we use these continuous words to extract the relations.

Table 4.6: Frequent grammatical relations in two continuous words. “Freq.” shows the frequency of each relation in two continuous words against “Total”, which is the number of all occurrences of each relation. The definition of “Relation” is described in [5].

Freq.	Total #	Relation	Examples
100.0	15	number	two yellow, four blue, two black, four brown, three wheeled
100.0	2	discourse	saying welcome, trains amble
100.0	1	mwe	more than
100.0	1	possessive	TV’s
98.6	216	auxpass	is turned, are displayed, is stacked, is shown, is surrounded
97.7	43	expl	there are, there is
96.6	384	prt	gear down, taken off, close up, travels down, lifted off, trail behind
94.5	509	aux	are conversing, are waiting, has taken, is flying, is sitting
91.3	46	pcomp	from below, in to, from behind, by for, on to
86.4	2457	nn	metal gate, link fence, sandy field, bird feeder, american eagle
80.0	5	quantmod	about nine, than five, a few, only few
75.5	1711	vmod	things growing, retro living, tool shed, boy pointing, uniforms looking
74.3	35	acomp	wearing white, stands next, looks sad, holding newborn, train idle
73.5	34	npadvmod	little disheveled, mouth open, one black, plants in, dairy somewhere
73.0	905	num	two phones, one hand, two people, two bicyclists, one leg
72.6	5323	amod	moving bicycle, land smiling, left leg, rocky path, clear day
71.9	352	poss	his bicycle, his stomach, its kickstand, its wings, her head
66.7	3	cc	table and, in and, grow and
61.5	78	prep	sheep on, played with, picture of, filled with, learning in
60.3	330	advmod	sitting together, couch outside, very old, visible just, on tracks
55.2	9141	det	coming forward, lay quietly, sit idly, very crowded, placed right
51.2	201	xcomp	screen showing, phone sitting, train traveling, airliner taking, helmet posing
48.7	113	cop	is open, is white, is painted, are ready, is happy
46.3	2897	nsubj	bus drives, snow covered, people walk, SUV smashed, cat lounges

---

The performance of multi-keyphrase estimation and accuracy of generated sentence of all compared methods including the baseline is shown in Table 4.5. For a fair comparison to the baseline, we show the results when using all phrases as keyphrases. PA and NHERD are superior to knn in terms of multi-keyphrase estimation, but PAAPL clearly outperforms other methods because the convergence of PA and NHERD with normal hinge-loss becomes much delayed in the situation where one image has multiple phrases. For the accuracy of generated sentences, PAAPL achieves the best score. Merely applying a grammar model to baseline [1] does not affect the accuracy. Although PA and NHERD outperform knn in terms of keyphrase estimation, the accuracy of the generated sentence does not increase greatly.

In this chapter, we use filters based on frequencies only. The objective to filter phrases according to their frequencies is to eliminate overly frequent (and meaningless) phrases such as “is-a”. Therefore, we would like to eliminate all meaningless phrases.

Figure 4.6 shows the result of PAAPL with FV and each keyphrase filter. As it is shown, global DF filter with  $300 \geq n_d \geq 10$  yields the best performance. Although the accuracy increases with larger  $n_k$  of local TF-IDF filter, Table 4.3 and Table 4.4 show that the number of keyphrases from local TF-IDF filter with  $n_k = 9$  is much larger than that from global DF filter with  $300 \geq n_d \geq 10$ . It can be said that a phrase should be regarded as a keyphrase when the sample numbers which contain the phrase are greater than the lower bound.

What are meaningless phrases? Indeed, “is-a” is a meaningless phrase. Actually, this is not because “is-a” is overly frequent but because “is-a” consists of an auxiliary verb and an article. Therefore, meaningless phrases are apparently found by considering whether each word in the phrase is meaningless or not. For example, the following word classes can be regarded as senseless.

1. articles (a/an/the)
2. prepositions (about/in/upon)
3. auxiliary verbs (be/do/have)
4. pronouns (it/this/I)
5. numericals (one/two/three)
6. interrogatives (who/where/when)

Additionally, other meaningless words exist such as abstract verbs (come/have/take). Although new words are born every year, the meaningless words described here are apparently senseless and will be so for a long time.

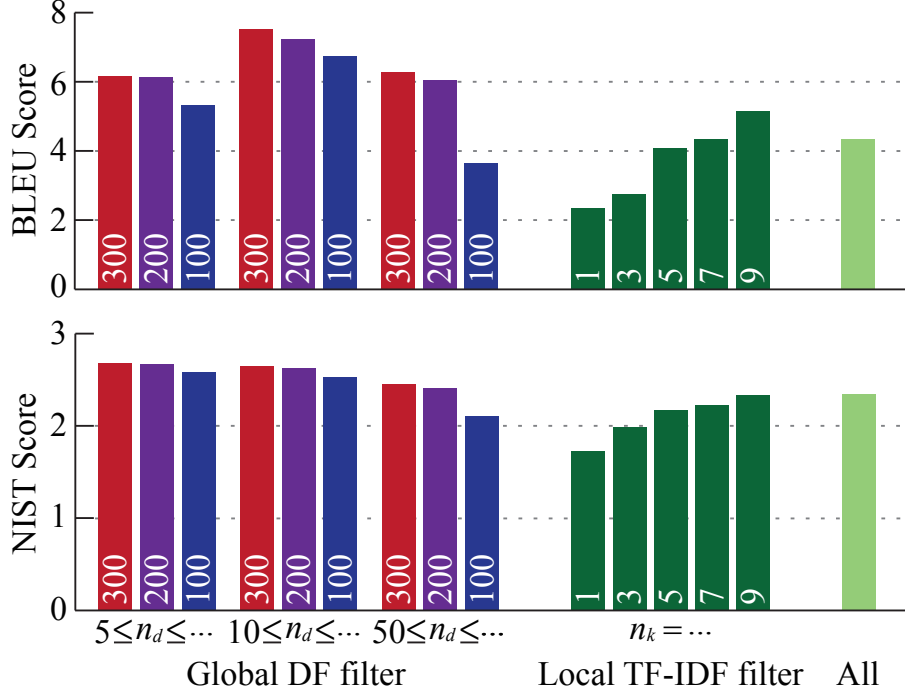


Figure 4.6: Accuracy of generated sentences with each filter for keyphrase extraction.

In the literature on Information Retrieval, such meaningless words are called stop words and are defined for each retrieval system. Therefore, we present another filter, Stop Word Filter (SWF), based on the rate of stop words. Particularly, phrases are discarded if more than half of words in the phrases are stop words. In Chapter 6, we evaluate the effect of this filter.

Although the proposed method achieves the best result, the performance still must be improved mainly because the keyphrases are large compared to the number of training samples. In [43], manually generated sentences for images achieve BLEU 1 of almost 50. Low accuracy of multi-keyphrase estimation harms the eventual performance of generated sentences.

The simplest means to cope with the shortage of training samples is collecting a larger dataset. The proposed method is applicable for a large dataset. Therefore, the accuracy of generated sentences will be improved when numerous data consisting of pairs of an image and a sentence are collected.

Another problem for the keyphrase approach using PAAPL is that  $O(dn_y)$  parameters are necessary for learning all classifiers, which would require not only too much space complexity but also produce a shortage of training samples.

---

## 4.6 Summary of Proposed Approach for Sentential Description

In this chapter, we propose a novel approach to generate sentences from images. We present *Multi-keyphrase Problem* to estimate keyphrases and to generate a sentence by connecting the keyphrases using a grammar model. Our method merely requires pairs of an image and an associated sentence. Manual preparation of semantic knowledge such as subjects, actions, and scenes is not necessary.

Because we consider different word sequences as independent phrases, the multi-keyphrase problem is reduced to Image Annotation. Therefore, we propose novel online learning called PAAPL for training samples with multiple labels.

Experimental results demonstrate that sentences can be generated with the proposed framework. The accuracy of generated sentences is better than that of the existing work. Moreover, the proposed PAAPL method is superior in terms of scalability and performance on Image Annotation. However, its accuracy remains low, mainly because there are many more *keyphrases* than labels in usual annotation datasets. These numerous *keyphrases* require too many parameters for classifiers. Therefore, the space complexity would be a problem. Our next work is therefore the development of a learning method to learn many labels more correctly with fewer parameters.

# Chapter 5

## CoSMoS: Common Subspace for Model and Similarity

Image Annotation is a widely confronted problem of associating input images with multiple labels. To estimate keyphrases for images, many more labels should be learned because keyphrases are combinations of labels. This chapter introduces a subspace in which (a) all feature vectors associated with the same label should be mapped as mutually close and (b) classifiers for each label are learned. To learn such a subspace, we propose a novel online learning method called Common Subspace for Model and Similarity (CoSMoS).

To learn linear classifiers, most methods can be grouped into two approaches: **model**-based methods such as SVM to learn linear weight vectors and **similarity**-based methods to learn metric in the feature space. This method can be regarded as a combination of both methods.

Experimental results obtained using three de-facto standard datasets for Image Annotation show that the proposed method achieves state-of-the-art performance and superior scalability.

### 5.1 Subspace for Image Recognition

As described in Chapter 4, methods for Image Annotation are divisible into two groups: (a) non-parametric approaches by which an input image is annotated with the labels of images located near the input image according to a certain metric and (b) classification approaches with which each classifier for each label scores how relevant the input image is to each label. For Image Annotation, non-parametric approaches are mainstream. In Chapter 4, we modify Passive Aggressive (PA) [59], which is an existing online learning method for multiclass classification, by devoting attention to the fact that each datum has more than



---

one label. The proposed method, Passive–Aggressive with Averaged Pairwise Loss (PAAPL), achieves state-of-the-art performance and scalability with the benchmark datasets for Image Annotation.

The most important problem to learn a large amount of labels such as keyphrases is that we should treat a large amount of parameters for classifiers. Learning methods for linear weights including PAAPL require  $d$ -dimensional weight vectors for all  $n_y$  labels. This problem is not confined to **model**-based methods. State-of-the-art methods for **similarity**-based annotation such as TagProp [31] and 2PKNN [33] require distance functions for each label. Because overly numerous parameters create the need for a huge training dataset, accuracy of keyphrase estimation would suffer.

A useful technique to reduce a large number of parameters is the use of subspace learning methods including traditional Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). Web Scale Annotation by Image Embedding (WSABIE) [64] approximates linear classifiers using a low-rank matrix. One version of Large Scale Metric Learning (LSML) [104, 105] uses mean vector for each class (Class Mean) to classify the input datum by searching for the nearest Class Mean. In fact, LSML obtains a subspace in which all vectors belonging to the same class come close to their mean. LDA and its variation, Canonical Correlation Analysis (CCA), can be regarded as similar methods in the sense that a subspace where feature vectors associated with the same label come mutually close is learned. Generally speaking, CCA can maximize the correlation between image features and labels in the latent space. Indeed, Canonical Contextual Distance (CCD) [32] uses the nature of CCA for Image Annotation.

This thesis presents assessment of methods to learn a distance function (or a similarity function) so that the feature vectors having the same label would come as mutually close as **similarity**-based methods. At the same time, we refer to methods to learn linear weights for each label as **model**-based methods. Their illustrations are shown in Figure 5.1. It is noteworthy that (a) non-parametric approaches and (b) classification approaches are distinguished by the form of recognition. They are not respectively synonymous with **similarity**-based methods and **model**-based methods, which are distinguished by the form of learning. For example, both CCD and LSML are **similarity**-based learning methods, although CCD and LSML respectively use a non-parametric approach and a classification approach for annotation.

A problem of **model**-based subspace learning such as WSABIE is that the constraints of the subspace are not defined clearly. However, **similarity**-based subspace learning methods have a constraint that the feature vectors with the same label would come close mutually. However, **similarity**-based methods cannot obtain clear rules to classify input data in the subspace. With LSML, which obtains a subspace where the feature vectors with the same label would come

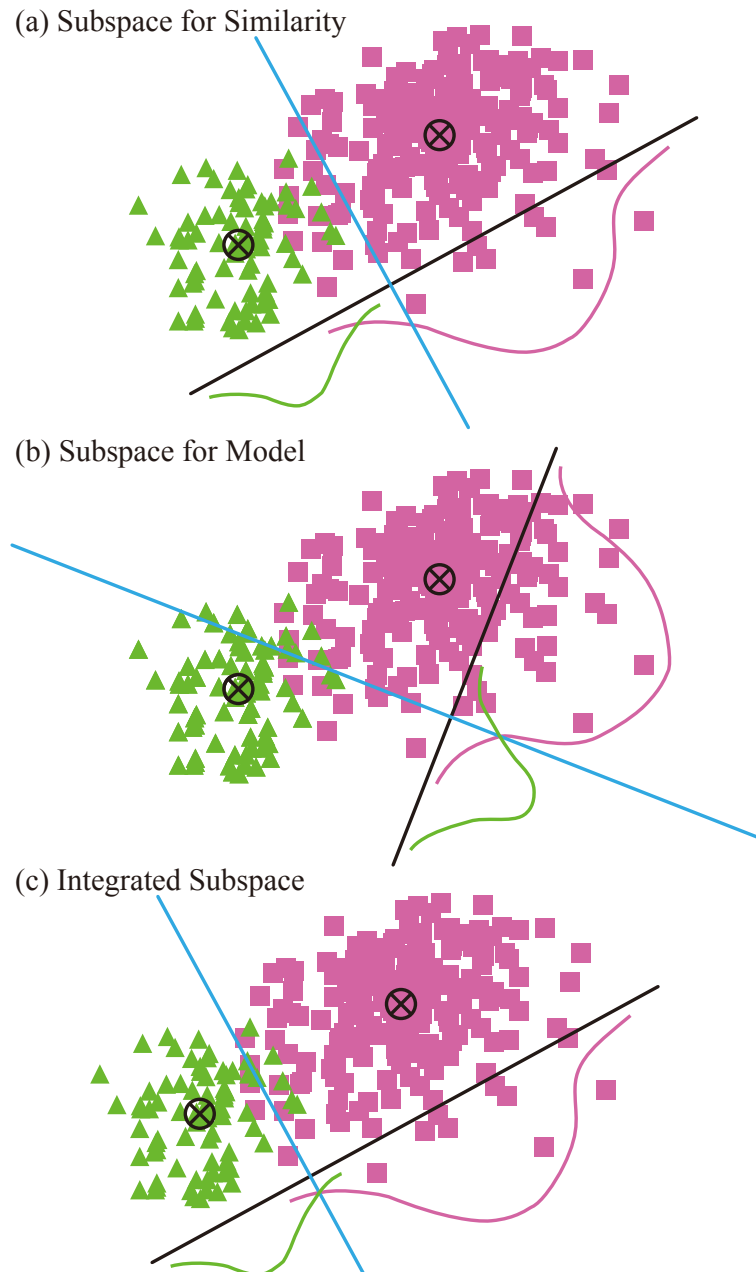


Figure 5.1: Simple overview of subspace learning. Now we would like to obtain one-dimensional subspace (black line) given training samples in two-dimensional feature space. The blue line orthogonal to each subspace is the decision plane between the green triangle class and purple rectangle class. Two crossed circles are the mean of each class.

---

close to their Class Mean, input data are classified according to the nearest Class Means. Therefore, if the distribution for each label is biased, such a classification rule would misclassify the input data.

Therefore, in this chapter, we introduce a combined form of subspace learning methods. We would like to obtain a subspace in which (a) all feature vectors associated with the same label should be mapped as mutually close and (b) classifiers for each label are learned. The proposed method, called Common Subspace for Model and Similarity (CoSMoS) annotates images using not only similarities to Class Means but also linear weight vectors in the subspace. By incorporating **model**-based learning and **similarity**-based learning, stable and accurate annotation can be achieved.

## 5.2 Related Subspace Learning Methods

In fact, WSABIE [64] aims to achieve Image Annotation using limited computation. Subspace is introduced to make their memory usage fit for a laptop in the ideal case. WSABIE estimates a label  $\hat{y}$  for the feature vector  $\mathbf{x}_t$  of an input datum as:

$$\hat{y} = \operatorname{argmax}_y \boldsymbol{\delta}^y \top M^\top S \mathbf{x}_t, \quad (5.1)$$

where  $S, M \in \mathbb{R}^{D \times d}$  respectively denote a projection matrix to the subspace and linear classifiers. The  $i$ -th column vector is the classifier  $\boldsymbol{\mu}^{y_i} \in \mathbb{R}^D$  for label  $y_i$ .  $\boldsymbol{\delta}^y \in \mathbb{R}^{n_y}$  is a binary vector the  $y$ -th element of which is one and the others are zero. The names of some variables here differ from those presented in [64] for reasons of consistency. WSABIE also introduces Weighted Approximate-Rank Pairwise Loss, an approximation of Ordered Weighted Pairwise Classification Loss [106]. This loss function can weigh lower-ranked positive labels heavily. By adopting stochastic gradient descent, subspace  $S$  and linear weights  $M$  are learned.

LSML [104, 105] combine the means (Class Mean) of feature vectors belonging to the same class and similarity learning. The classification rule of LSML is the following.

$$\hat{y} = \operatorname{argmax}_y \tilde{\mathbf{x}}^y \top S^\top S \mathbf{x}_t - b^y. \quad (5.2)$$

Therein,  $b^y$  is a bias for label  $y$ . In [105],  $\tilde{\mathbf{x}}^y$  can be chosen from (i) all training samples (i.e., non-parametric approach), (ii) single Class Mean for each label, or (iii) multiple Class Means for each label. This thesis presents discussion mainly of (ii) single Class Mean for each label because we specifically examine linear classification.

The authors of [104, 105] report that LSML is related to WSABIE. Particularly,  $\tilde{\mathbf{x}}^y \top S^\top$  in LSML corresponds to  $\boldsymbol{\delta}^y \top M^\top$  in WSABIE. The LSML method uses logistic loss, which means that all labels  $y \notin Y_t$  not attached to  $t$ -th feature  $\mathbf{x}_t$  should be scored as zero. However, this loss function might be too strict because some labels are not attached but are relevant to the image.

Canonical Contextual Distance (CCD) [32, 46, 107] is a method for Image Annotation based on a probabilistic interpretation of CCA (PCCA) [108]. Given image feature vector  $\mathbf{x} \in \mathbb{R}^d$  and the feature vector of associated labels  $\mathbf{y} \in \mathbb{R}^{n_y}$  as a binary vector the  $y$ -th element of which is one and the others are zero, then because inner products between two label vectors represents the co-occurrence of labels, binary representation for labels is reasonable. Actually, PCCA obtains latent variables by considering both similarities among image features and similarities among label features. First, probabilistic distributions among image features, label features, and latent variables are defined using Gaussians as described below.

$$\mathbf{z} \sim \mathcal{N}(0, I_D), \min\{d, n_y\} \geq D \geq 1, \quad (5.3)$$

$$\mathbf{x}|\mathbf{z} \sim \mathcal{N}(M_x \mathbf{z} + \mathbf{m}_x, \Psi_x), M_x \in \mathbb{R}^{d \times D}, \Psi_x \succeq 0, \quad (5.4)$$

$$\mathbf{y}|\mathbf{z} \sim \mathcal{N}(M_y \mathbf{z} + \mathbf{m}_y, \Psi_y), M_y \in \mathbb{R}^{n_y \times D}, \Psi_y \succeq 0. \quad (5.5)$$

In actuality, as [108] proves, the solutions of maximum likelihood estimation of these models are identical to the solutions of CCA. Therefore, posteriori distribution to latent variables is based on input images as follows. Define the distribution of the latent variable  $\mathbf{z}$  when only an image feature vector  $\mathbf{x}$  is given as  $p(\mathbf{z}|\mathbf{x})$ . Similarly, define the distribution of  $\mathbf{z}$  when both an image feature  $\mathbf{x}$  and a label feature  $\mathbf{y}$  are given as  $p(\mathbf{z}|\mathbf{x}, \mathbf{y})$ . These distributions are also Gaussians, the means and the variances of which can be derived analytically as presented below.

$$\hat{\mathbf{z}}_x = E(\mathbf{z}|\mathbf{x}) = M_x^\top A^\top (\mathbf{x} - \mathbf{m}_x), \quad (5.6)$$

$$\hat{\Phi}_x = \text{var}(\mathbf{z}|\mathbf{x}) = I - M_x M_x^\top, \quad (5.7)$$

$$\begin{aligned} \hat{\mathbf{z}}_{xy} &= E(\mathbf{z}|\mathbf{x}, \mathbf{y}) = \\ &\begin{pmatrix} M_x \\ M_y \end{pmatrix}^\top \begin{pmatrix} (I - \Lambda^2)^{-1} & -(I - \Lambda^2)^{-1} \Lambda \\ -(I - \Lambda^2)^{-1} \Lambda & (I - \Lambda^2)^{-1} \end{pmatrix} \begin{pmatrix} A^\top (\mathbf{x} - \bar{\mathbf{x}}) \\ B^\top (\mathbf{y} - \bar{\mathbf{y}}) \end{pmatrix}, \end{aligned} \quad (5.8)$$

$$\begin{aligned} \hat{\Phi}_{xy} &= \text{var}(\mathbf{z}|\mathbf{x}, \mathbf{y}) = I - \\ &\begin{pmatrix} M_x \\ M_y \end{pmatrix}^\top \begin{pmatrix} (I - \Lambda^2)^{-1} & -(I - \Lambda^2)^{-1} \Lambda \\ -(I - \Lambda^2)^{-1} \Lambda & (I - \Lambda^2)^{-1} \end{pmatrix} \begin{pmatrix} M_x \\ M_y \end{pmatrix}, \end{aligned} \quad (5.9)$$

Therein,  $\bar{\mathbf{x}}$  and  $\bar{\mathbf{y}}$  respectively signify means of all image feature vectors  $\mathbf{x}$  and all label feature vectors  $\mathbf{y}$ .  $A$  and  $B$  are projection matrices, and  $\Lambda$  is a diagonal matrix with the first  $D$  canonical correlations on its diagonal components.  $D$

---

represents the subspace dimension, called canonical space. These matrices are calculable using plain CCA as the solutions of generalized eigenvalue problems as:

$$C_{xx}C_{yy}^{-1}C_{yx}A = C_{xx}A\Lambda^2 (A^\top C_{xx}A = I_D), \quad (5.10)$$

$$C_{yx}C_{xx}^{-1}C_{xy}A = C_{yy}B\Lambda^2 (B^\top C_{yy}B = I_D), \quad (5.11)$$

where  $C = \begin{pmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{pmatrix}$  is a covariance matrix of training samples.  $M_x, M_y \in \mathbb{R}^{D \times D}$  are matrices such that  $M_x M_y^\top = \Lambda$ . CCD uses diagonal matrices as:

$$M_x = \Lambda^\beta, M_y = \Lambda^{1-\beta} (0 < \beta < 1). \quad (5.12)$$

To learn a latent space, both image feature vectors and the label feature mutually interact as a supervisory signal. As a result, the obtained subspace represents both images and labels efficiently.  $\beta$  is a hyperparameter to balance the contributions of the images and the labels to estimate the latent variables.

Canonical Contextual Distance (CCD) is a distance function between the probabilistic distributions introduce above. [107] presents 1-view CCD (CCD1) and 2-view CCD (CCD2). Whereas CCD1 uses  $p(\mathbf{z}|\mathbf{x})$  to calculate the distances, CCD2 uses  $p(\mathbf{z}|\mathbf{x}, \mathbf{y})$ . Therefore, CCD2 considers both similarities among image feature vectors and similarities among label feature vectors. This chapter presents a description of CCD2 as a more related method.

As described above, each sample can be represented as a Gaussian in the latent space. Therefore, we use Kullback–Leibler (KL) divergence to calculate the distances among these distributions. Given an input image feature  $\mathbf{x}_q$  and training samples  $(\mathbf{x}_t, \mathbf{y}_t)$ , their distance is calculable as KL divergence as:

$$\begin{aligned} \text{KL}(p(\mathbf{z}|\mathbf{x}_q)||p(\mathbf{z}|\mathbf{x}_t, \mathbf{y}_t)) &= \frac{1}{2} \log \frac{|\Phi_{xy}|}{|\Phi_x|} - \frac{D}{2} \\ &+ \frac{1}{2} \text{tr}(\Phi_{xy}^{-1}\Phi_x) + (\hat{\mathbf{z}}_q - \hat{\mathbf{z}}_t)^\top \Phi_x^{-1} (\hat{\mathbf{z}}_q - \hat{\mathbf{z}}_t). \end{aligned} \quad (5.13)$$

Because the first three terms are constant values, CCD is defined as a Euclidean distance as:

$$\text{CCD}((\mathbf{x}_i, \mathbf{y}_i), \mathbf{x}_q) = |\mathbf{r}_q - \mathbf{r}_t|^2, \quad (5.14)$$

where  $\mathbf{r}_q = \Phi_{xy}^{-1/2} \hat{\mathbf{z}}_{x_q}$  and  $\mathbf{r}_t = \Phi_{xy}^{-1/2} \hat{\mathbf{z}}_{x_i y_i}$  respectively represent coordinates of an input image and training samples in the latent space. In [32], the authors show experimentally that CCA has better annotation performance than PCA, Partial Least Squares (PLS), and plain CCA.

---

However, CCD entails several problems. The first is complexity of CCA. Although the complexity is independent of the sample size  $N$ , solving CCA requires  $O(d^3)$  complexity for the dimension  $d$  of the image feature vector. Moreover, the stability of CCA must be considered. If  $C_{xx}$  or  $C_{yy}$  is semidefinite, then inverting these matrices causes instability. One solution is to add a regularization term. Particularly by redefining  $C_{xx}$  as  $C_{xx} + \gamma I$ , image feature vectors have a bit of white noise. However, we should tune a hyperparameter  $\gamma > 0$ . The other solution is adopting PCA as a preprocess. By reducing the dimensions of the image feature vector, dimensions for which the variances are small are ignored. Moreover, dimension reduction contributes to the complexity of CCA. However, the dimension reduced by PCA is also a hyperparameter to be tuned. Finally, maximizing correlation between image features and label features is not equivalent to maximizing the annotation performance. Correlation maximization merely improves annotation performance indirectly by making image feature vectors with the same label come mutually close.

Additionally, although CCD has complexity that is independent of the sample size  $N$ , [32] uses Kernel PCA with a subset of training samples as a preprocess to improve similarities among image features to achieve comparable performance to that of TagProp [31]. Consequently, the learning has complexity of at least  $O(N)$ .

Here, we consider annotation using a Class Mean like LSML. Given Class Means of image feature vectors and label feature vectors for the label  $y$  as  $\tilde{\mathbf{x}}^y$  and  $\tilde{\mathbf{y}}^y$ , respectively, then the CCD between Class Means and input image feature is:

$$\text{CCD}((\tilde{\mathbf{x}}^y, \tilde{\mathbf{y}}^y), \mathbf{x}_q) = |\mathbf{r}_q - \mathbf{r}_t|^2, \quad (5.15)$$

$$= \tilde{\mathbf{x}}^{y\top} S^\top S \mathbf{x}_q + \tilde{\mathbf{y}}^{y\top} M^\top S \mathbf{x}_q - b^y, \quad (5.16)$$

where  $S$  and  $M$  are projection matrices derived from CCD, and  $b^y$  is a scalar representing the bias for the label  $y$ . Therefore, the input image is classified according to the sum of (a) similarity between Class Mean and the input image feature  $\mathbf{x}_q$  and (b) inner product of the input image feature and linear weight classifier for each class in the subspace. By designing a proper loss function based on this classification rule, we can learn a subspace in which (a) all feature vectors associated with the same label should be mapped as mutually close and (b) classifiers for each label are learned.

### 5.3 Methodology

This section presents a description of the proposed method: Commons for similarity and Model (CoSMoS). We define a classification rule according to the sum of (a) similarity between the Class Mean and the input image feature and (b)

---

inner product of the input image feature and linear weight classifier for each class in the subspace. We optimize this classification by introducing the averaged pairwise loss and averaged stochastic gradient descent.

### 5.3.1 Classification Rule and Objective Function

In this subsection, we develop CoSMoS as an online learning method. Given  $t$ -th image feature  $\mathbf{x}_t \in \mathbb{R}^d$  associated with a set of label  $Y_t \subset \mathcal{Y}$ , where  $\mathcal{Y}$  is a set of all labels and  $n_y$  is the number of all labels. We define the label feature vector  $\mathbf{y} \in \mathbb{R}^{n_y}$  as a binary vector, the  $i$ -th element of which is one if the image is associated with  $y_i$ , otherwise zero. Let us define Class Means of image feature vectors and label feature vectors for label  $y$  as  $\tilde{\mathbf{x}}^y$  and  $\tilde{\mathbf{y}}^y$ , respectively. The classification rule is defined as presented below.

$$\hat{y} = \operatorname{argmax}_y \tilde{\mathbf{x}}^y \top S^\top S \mathbf{x}_t + \tilde{\mathbf{y}}^y \top M^\top S \mathbf{x}_t - b^y. \quad (5.17)$$

By introducing  $U \equiv (S \ M)$ , this rule becomes:

$$\hat{y} = \operatorname{argmax}_y \theta^y(\mathbf{x}_t) \equiv \operatorname{argmax}_y \begin{pmatrix} \tilde{\mathbf{x}}^y \\ \tilde{\mathbf{y}}^y \end{pmatrix}^\top U_t^\top U_t \begin{pmatrix} \mathbf{x}_t \\ \mathbf{0} \end{pmatrix} - b^y. \quad (5.18)$$

Most works for multiclass classification [59, 62] use pairwise loss  $\ell \equiv 1 - \min_{c \in Y_t} s^y(\mathbf{x}_t) + \max_{c \notin Y_t} s^y(\mathbf{x}_t)$ . As described in Chapter 4. However, using all labels attached to one image would fasten the convergence of learning. Therefore, we introduce Averaged Pairwise Loss also for CoSMoS.

$$\ell_t((S_t, M_t, \mathbf{b}_t); (\mathbf{x}_t, Y_t)) \equiv 1 - \frac{1}{|S_t|} \sum_{c \in S_t} s^y(\mathbf{x}_t) + \frac{1}{|S'_t|} \sum_{c \in S'_t} s^y(\mathbf{x}_t), \quad (5.19)$$

Therein,  $\mathbf{b} \in \mathbb{R}^{|\mathcal{Y}|}$  is a vector of bias, the  $i$ -th element of which is the bias of the label  $y_i$ . Additionally, the members of  $S_t \subset Y_t$  and  $S'_t \subset \mathcal{Y}/Y_t$  are chosen as explained below.

1. Pick up  $y \in Y_t$  with the minimum score  $\theta_y$ .
2. Pick up  $y' \notin Y_t$  with the maximum score  $\theta^{y'}$ .
3. If  $\theta^y < \theta^{y'} + 1$ , then (i) add  $y$  and  $y'$  respectively to  $S_t$  and  $S'_t$ , (ii) remove  $y$  and  $y'$  from  $Y_t$ , and (iii) go back to the first step. Otherwise, the current  $S_t$  and  $S'_t$  are fixed.

---

For a simple formulation, we use  $\mathbf{g}_t \in \mathbb{R}^{n_y}$  and define the  $i$ -th element  $g_{t,i}$  as:

$$g_{t,i} = \begin{cases} 1/|S_t| & y_i \in S_t \\ -1/|S'_t| & y_i \in S'_t \\ 0 & \text{otherwise} \end{cases}. \quad (5.20)$$

Now (5.19) can be rewritten as:

$$\ell_t((S_t, M_t, \mathbf{b}_t); (\mathbf{x}_t, Y_t)) = 1 - \mathbf{g}_t^\top \left( \begin{pmatrix} \tilde{X} \\ \tilde{Y} \end{pmatrix}^\top U_t^\top U_t \begin{pmatrix} \mathbf{x}_t \\ \mathbf{0} \end{pmatrix} - \mathbf{b}_t \right), \quad (5.21)$$

where  $\tilde{X}$  and  $\tilde{Y}$  are matrices consisting of Class Means, the  $i$ -th column vector of which corresponds to the Class Mean  $\tilde{\mathbf{x}}^{y_i}$  and  $\tilde{\mathbf{y}}^{y_i}$  for the label  $y_i$ . Consequently, the objective function  $\mathcal{L}(U, \mathbf{b})$  we would like to minimize is determined as a following cumulative loss.

$$\mathcal{L}(U, \mathbf{b}) = \sum_{t=1}^T \left( 1 - \mathbf{g}_t^\top \left( \begin{pmatrix} \tilde{X} \\ \tilde{Y} \end{pmatrix}^\top U^\top U \begin{pmatrix} \mathbf{x} \\ \mathbf{0} \end{pmatrix} - \mathbf{b} \right) \right). \quad (5.22)$$

To minimize the objective function, we adopt averaged stochastic gradient descent. Update rules for the matrix  $U_t$  and the bias  $\mathbf{b}_t$  are defined using learning rate  $\eta_t$  as:

$$U_{t+1} = U_t \left( I_{d+n_y} + \eta_t \begin{pmatrix} \mathbf{x}_t \\ \mathbf{0} \end{pmatrix} \mathbf{g}_t^\top \begin{pmatrix} \tilde{X} \\ \tilde{Y} \end{pmatrix}^\top + \eta_t \begin{pmatrix} \tilde{X} \\ \tilde{Y} \end{pmatrix} \mathbf{g}_t \begin{pmatrix} \mathbf{x}_t \\ \mathbf{0} \end{pmatrix}^\top \right), \quad (5.23)$$

$$\mathbf{b}_{t+1} = \mathbf{b}_t - \eta_t \mathbf{g}_t. \quad (5.24)$$

As described in [64, 109],  $U$  is initialized randomly with mean 0 and standard deviation  $1/\sqrt{d+n_y}$ .

## 5.4 Experiments for Image Annotation

To evaluate CoSMoS, this section reports the result of Image Annotation using three de-facto standard datasets as performed in Chapter 4.

### 5.4.1 Dataset

We use three datasets: Corel 5k, ESP Game, and IAPR-TC12 as used in Chapter 4.



---

**Corel 5k.** Corel 5k [8] consists of about 5000 images for 260 labels. The dataset is divided into 4500 training samples and 500 testing samples. Each image has around 3.4 labels.

**ESP Game.** ESP Game [13] has 60,000 images collected via an online game where players label images as other players do. Existing works [31, 32, 33, 96, 110] examining Image Annotation use around 20,000 images for 268 labels. Each image has around 4.7 labels.

**IAPR-TC12.** IAPR-TC12 [14] is provided for research of image retrieval using multilingual labels and sentences. Existing works [31, 32, 33, 96, 110] assessing Image Annotation use 20,000 images for 291 labels. Each image has around 5.7 labels.

To make a fair comparison, we use the image features of 15 types provided by TagProp [31]. There are seven global image features: GIST [91] and color histograms in RGB, HSV, and LAB. Three color histograms are extracted not only from the whole image but also from three horizontal regions of the image and concatenated. The other eight features are the Bag-of-Visual-Words (BoVW) model using SIFT [89] and hue descriptors according to [111]. Each descriptor is extracted from regular grid cells or from interest points. These BoVW are also extracted from the whole image or from three horizontal regions. These features are widely used to evaluate annotation methods [30, 31, 32, 96, 110].

Parameter  $\eta_t$  of learning methods WSABIE, LSML, CMM and its variations are tuned by selecting the best one from  $\{2^{-3}, 2^{-4}, \dots, 2^{-7}\}$ . To combine different features, in this paper, we learn the models for each feature independently first. Later we estimate labels for test samples with the sum of the scores from classifiers of respective features.

Following existing works, we estimate five labels for each test sample. We evaluate the performance for annotation using the following three indicators:

**Precision ( $P$ )** The ratio of correctly estimated labels to estimated labels.

**Recall ( $R$ )** The ratio of correctly estimated labels to correct labels.

**F-measure ( $F$ )** The harmonic average of Precision and Recall.

## 5.4.2 Comparison to State-of-the-art Methods

First, we compared CMM to the existing state-of-the-art-methods on three datasets. To elicit the best performance of CMM, we determined the dimension  $D$  of subspace four times as large as the number of labels in each dataset.

Table 5.1 presents comparisons to state-of-the-art methods using three visual annotation datasets. As described before, CoSMoS is related to CCD [32], which

Table 5.1: Comparison of annotation performances among state-of-the-art methods.

Dataset		MBRM [111]	JEC [30]	TagProp [31]	CCD [32]	2PKNN [33]	PAAPL [110]	K SVM-VT [96]	CoSMoS (ours)
Corel	<b>P</b>	0.24	0.27	0.33	0.36	<b>0.44</b>	0.40	0.32	<b>0.41</b>
	<b>R</b>	0.25	0.32	0.42	0.41	0.46	<b>0.57</b>	0.42	<b>0.58</b>
	<b>F</b>	0.25	0.29	0.37	0.38	0.45	<b>0.47</b>	0.36	<b>0.48</b>
ESP	<b>P</b>	0.18	0.22	0.39	0.36	<b>0.53</b>	0.32	0.33	<b>0.35</b>
	<b>R</b>	0.19	0.25	0.27	0.24	0.27	<b>0.35</b>	0.32	<b>0.39</b>
	<b>F</b>	0.19	0.23	0.32	0.29	<b>0.36</b>	0.33	0.33	<b>0.37</b>
IAPR	<b>P</b>	0.24	0.28	0.46	0.44	<b>0.54</b>	0.40	0.47	<b>0.42</b>
	<b>R</b>	0.23	0.29	0.35	0.29	0.37	<b>0.38</b>	0.29	<b>0.40</b>
	<b>F</b>	0.24	0.29	0.40	0.35	<b>0.44</b>	0.39	0.36	<b>0.41</b>

uses a probabilistic interpretation of CCA [108]. Although 2PKNN [33] and PAAPL [110] are superior to CCD, CoSMoS can also achieve state-of-the-art performance. TagProp and 2PKNN consists of **similarity**-learning and non-parametric annotation. These methods require  $O(N^2)$  complexity for learning and  $O(N)$  complexity for annotation against sample size  $N$ .

Particularly, CMM outperforms PAAPL for all evaluations on all datasets. In addition, CMM outperforms 2PKNN for recall on all datasets. Again, a tradeoff prevails between precision and recall. CMM outperforms 2PKNN for the  $F$ -measure, which is a kind of average of recall and precision, on Corel and ESP. The features used for 2PKNN in [33] are not exactly the same as the features used in other works [30, 31, 32, 96, 110].

### 5.4.3 Comparison to Subspace Learning Methods

As described in Section 5.3, CoSMoS is related not only to CCD [32, 46, 107] but also to LSML [104, 105] and WSABIE [64, 109]. Because LSML and WSABIE are not evaluated on these datasets, we implement and evaluate them using Corel 5k.

Additionally, although LSML and WSABIE are closely related to CoSMoS, there are several differences. To investigate which factor of CoSMoS contributes, we also implement and evaluate the following variations of CoSMoS.

Table 5.2: Comparison of annotation performances among subspace learning methods produced using Corel 5k.

Method	P	R	F
WSABIE	0.39	0.55	0.45
LSML	0.36	0.51	0.42
CoSMoS with Regularization	0.34	0.49	0.40
CoSMoS w/o Model	0.41	0.58	0.48
CoSMoS w/o Similarity	0.41	0.58	0.48
CoSMoS	0.41	0.58	0.48

**CoSMoS w/o Model** To ascertain the effect of integrating subspaces for model and similarity, we discard the model from CoSMoS: linear weight  $M$  is eliminated. The classification rule  $\hat{y} = \operatorname{argmax}_y \tilde{\mathbf{x}}^y S_t^\top S_t \mathbf{x}_t - b^y$  is exactly the same as LSML. Projection matrix  $S_t$  is learned as:

$$S_{t+1} = S_t(I_d + \eta_t \mathbf{x}_t \mathbf{g}_t^\top \tilde{X}^\top + \eta_t \tilde{X} \mathbf{g}_t \mathbf{x}_t^\top). \quad (5.25)$$

**CoSMoS w/o Similarity** Next we eliminate the part of similarity to Class Mean  $\tilde{\mathbf{x}}^y$ . In other words, we use a classification as  $\hat{y} = \operatorname{argmax}_y \tilde{\mathbf{y}}^y M_t^\top S_t \mathbf{x}_t - b^y$ . If we also eliminate the bias term, then this is identical to the classification rule of WSABIE. Projection matrices  $S_t$  and  $M_t$  are updated as:

$$S_{t+1} = S_t + \eta_t M_t \mathbf{y}_t \mathbf{g}_t^\top \tilde{Y}^\top, \quad (5.26)$$

$$M_{t+1} = W_t + \eta_t P_t \mathbf{x}_t \mathbf{g}_t^\top \tilde{Y}^\top. \quad (5.27)$$

**CoSMoS with Regularization Form** The original CoSMoS discards the regularization form from the objective function. If a squared Frobenius-norm for  $U$  and a squared  $L_2$  norm for  $\mathbf{b}$  are introduced, a new objective function  $\mathcal{L}'$  is defined as:

$$\mathcal{L}'(U, \mathbf{b}) = \frac{\lambda}{2} \|U\|_F^2 + \frac{\lambda}{2} \|\mathbf{b}\|_2^2 + \mathcal{L}(U, \mathbf{b}). \quad (5.28)$$

Therefore, we can achieve regularization by subtracting  $\eta_t \lambda U_t$  and  $\eta_t \lambda \mathbf{b}_t$  in  $t$ -th step. In the experiments, we determined  $\lambda = 1/N$ , where  $N$  is the number of training samples.

Table 5.2 shows annotation performance obtained using Corel 5k. WSABIE and LSML are slightly inferior to CoSMoS. Additionally, CoSMoS with regularization reduces the accuracy of the original CoSMoS. However, both CoSMoS

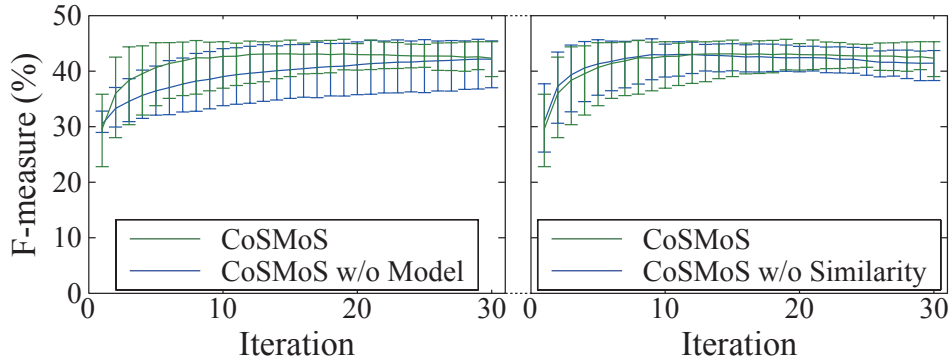


Figure 5.2: Comparison among CoSMoS variations using 16-dimensional subspace.

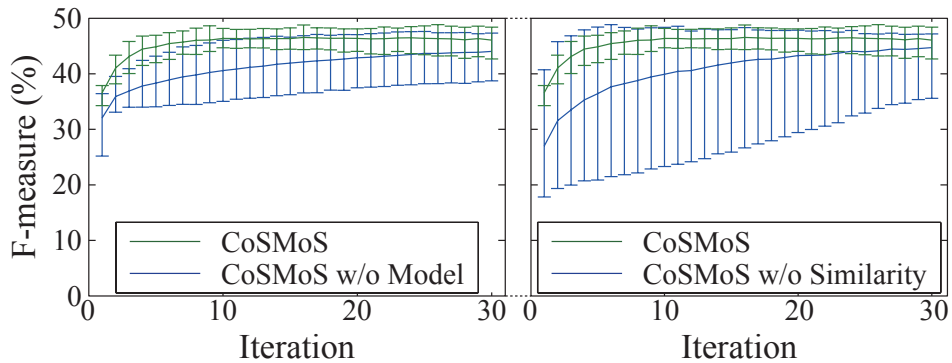


Figure 5.3: Comparison among CoSMoS variations using 1024-dimensional subspace.

w/o Model and CoSMoS w/o Similarity achieve similar performance to that of the original CoSMoS.

(a) Should the combinations of subspace where all feature vectors are associated with the same label be mapped as mutually close? (b) Are the learned classifiers for each label meaningless? Figure 5.2 and Figure 5.3 show the convergence of CoSMoS and its variations with  $\eta_t = \{2^{-3}, 2^{-4}, \dots, 2^{-7}\}$  and  $D = 16, 1024$ . The top and the bottom of each bar respectively represent the minimum and the maximum performance with  $\eta_t = \{2^{-3}, 2^{-4}, \dots, 2^{-7}\}$  in each iteration. Lines are shown by connecting their averaged performance.

First, CoSMoS is superior to its variations in terms of the convergence speed. CoSMoS converges markedly faster than the CoSMoS w/o Model. This difference becomes larger when the dimension of the subspace is large. In the high-

---

dimensional subspace, CoSMoS w/o Similarity converges more slowly than the original CoSMoS. In low-dimensional subspace, CoSMoS w/o Similarity begins to reduce the annotation performance.

Moreover, in comparison to the original CoSMoS, the performance of the variations of CoSMoS varies widely when the learning rate  $\eta$  changes. Especially, CoSMoS w/o Similarity is sensitive to the learning rate in a high-dimensional subspace. The original CoSMoS is less sensitive to the learning rate especially in the high-dimensional subspace. As described in Chapter 3, learning rate *eta* is a hyperparameter to be tuned. By combining the constraint from **similarity**-based learning and linear weight vectors from **model**-based learning, CoSMoS realizes a stable subspace learning method.

## 5.5 Towards Sentential Description for Images

In this chapter, we propose novel online learning using subspace for annotating images with numerous labels. Particularly we introduce a subspace in which (a) all feature vectors associated with the same label should be mapped as mutually close and (b) classifiers for each label are learned. To learn such subspace, we propose a novel online learning method called Common Subspace for Model and Similarity (CoSMoS). This method can be regarded as a combination of **model**-based methods such as SVM to learn linear weight vectors and **similarity**-based methods to learn metric in the feature space. We also report that Canonical Contextual Distance (CCD) [107] implicitly optimizes the subspace in a similar policy. By introducing Averaged Pairwise Loss and averaged stochastic gradient descent, we explicitly optimize the subspace and achieve state-of-the-art performance and scalability in benchmark datasets for Image Annotation.

As described in Chapter 4, to estimate keyphrases as an Image Annotation problem, consideration for a large amount of keyphrases is necessary. CoSMoS not only reduces the number of parameters by introducing subspace: it also achieves high accuracy for Image Annotation. Moreover, the learning is stable in relation to the learning rate. The next chapter reports the results of sentence generation using a combination of CoSMoS and sentence generation described in Chapter 4.

## Chapter 6

# Evaluation of Sentential Description for Images

We evaluated our methodology using three datasets: PASCAL Sentence, IAPR-TC12, and SBU. PASCAL Sentence consists of 1000 pairs of an image and around 5.0 sentences. IAPR-TC12 consists of 19,963 pairs of an image and around 1.8 sentences. Both datasets are compiled manually and described. SBU consists of 1M pairs of an image and a sentence. These images and sentences are collected from Flickr. When we train our system and test the performance, these datasets are respectively divided into training and testing images. We use the same criteria to divide them and repeat it five times.

As described in Section 4.5.2, we use BLEU [101] and NIST [102] for automatic evaluation for generated sentence. “BLEU x” means that the cumulative product of N-gram match rate is used from unigram to x-gram. Similarly, “NIST x” means that the cumulative sum of N-gram match rate is used from unigram to x-gram, which is the standard setup for NIST. Both BLEU and NIST have length penalties to make a fair evaluation of all sentences including overly short sentences. The ceiling on BLEU is one because this score is a kind of match rate. Because NIST weighs rear expressions, the ceiling is unclear.

We use the Fisher Vector (FV) [28] with SIFT [89]. We extract a SIFT descriptor from a regular grid with step size 6 pixels at multiple scales:  $16 \times 16$ ,  $25 \times 25$ ,  $36 \times 36$ ,  $49 \times 49$ , and  $64 \times 64$ . Then the dimensions are reduced to 64 using PCA.

For PASCAL Sentence and IAPR-TC12, we obtain a Gaussian mixture model with 256 components. Then FV is calculated respectively over  $1 \times 1$ ,  $2 \times 2$ , and  $3 \times 1$  cells.

For SBU, because this is a large-scale dataset, we obtain a Gaussian mixture model with 16 components to make all FVs fit for the memory space. Then FV is calculated respectively over the whole image without SPM.

Table 6.1: Statistics of datasets. IAPR-TC has 20,000 images, but 37 images are not associated with sentences. For PASCAL Sentence, we extract not only keyphrases with SWF but also keyphrases as described in Chapter 4.

Dataset	# of images	# of sentence per image	sentence length	# of keyphrases per sentence
PASCAL Sentence	1,000	$5.00 \pm 0.06$	$9.79 \pm 1.99$	$2.53 \pm 1.21$ ( $4.01 \pm 1.49$ )
IAPR-TC12	19,963	$1.76 \pm 0.84$	$15.36 \pm 6.79$	$5.31 \pm 3.26$
SBU	981,450	$1.00 \pm 0.01$	$12.02 \pm 5.93$	$3.45 \pm 2.76$

Keyphrases are extracted from phrases consisting of two continuous words. For PASCAL Sentence, as described in Chapter 4, keyphrases associated with not fewer than 10 images are extracted. Therefore, we extract keyphrases occurring at the same frequency from IAPR-TC12 and SBU. As described in Section 4.5.3, we also investigate another filter to discard phrases for which more than half of the words are meaningless. To estimate keyphrases, we train CoSMoS with 128-dimensional subspace in 10 iterations. The learning rate  $\eta$  is determined by selection from  $\{2^{-3}, 2^{-4}, 2^{-5}, 2^{-6}\}$ .

Sentences are generated using top-ten scored keyphrases for each image. From the experimentally obtained results in Chapter 4, the parameters described in Section 4.3.3 are fixed as follows. Statistics of these datasets are shown in Table 6.1. Because the SBU dataset itself contains not images but URLs of images, we download 981,450 existing images. The coefficient  $\lambda_p = 0.001$  for phrase cost. Because sentences in PASCAL Sentence consist of 10 words on average, the desired length  $l_0 = 10$  and the length strictness  $\sigma_0 = 0.1$ . We use bigrams and trigrams extracted from each dataset as grammar knowledge. Our modified multi-stack beam search is performed by preserving the top-five scored candidates in each stack.

For evaluation, every dataset is divided into training and testing samples. For each dataset, we follow the same experimental setup used in most previous works. Both PASCAL Sentence and IAPR-TC12 are divided into 90% training samples and 10% testing samples. For SBU, 500 testing images are extracted randomly. For each dataset, we repeat the division five times.



**Ours:** Group of people sitting at a table with a dinner.

**Corpus-Guided:** Three people are showing the bottle on the street.

**Midge:** People with a bottle at the table.

**BabyTalk:** This is a picture of three persons, one bottle and one diningtable. The first rusty person is beside the second person. The rusty bottle is near the first rusty person, and within the colorful diningtable. The second person is by the third rusty person. The colorful diningtable is near the first rusty person, and near the second person, and near the third rusty person.

Figure 6.1: Qualitative comparison. A common input image is shown in the upper left. We compare our result with Corpus-Guided [2], Midge [3], and BabyTalk [4].

## 6.1 PASCAL Sentence

Figure 6.2 presents some good examples of generated sentences. As shown, keyphrases estimated correctly contribute to the generation of appropriate sentences. However, as the bottom one in Figure 6.2, even if a few keyphrases (“decker bus” and “a bus”) are incorrect, our method based on the modified multi-stack beam search automatically selects keyphrases to the greatest extent possible. As a result, such incorrect keyphrases are ignored and a sentence “A living room with a view of a television.” is generated.

Figure 6.3 presents some partially incorrect examples. Generally, two groups of mistakes exist: grammatical and semantical. For example, for the first image in Figure 6.3, the estimated keyphrases are appropriate. However, the connection of those keyphrases is not correct in grammar. As a typical mistake, there are two keyphrases “a sheep” and “sheep standing” in the generated sentence. Usage of both keyphrases leads to a somewhat strange sentence. One solution is to discard phrases resembling the phrases already used in the sentence being generated. For example, in the generated sentence for the middle image in Figure 6.3, “the blue sky” and “the air” are overlapping. Therefore, it is necessary to discard not only phrases including the same word but also phrases including similar words.

Qualitative comparison to previous works is presented in Figure 6.1. The sentences generated by existing works are selected from results presented in those papers. Corpus-Guided [2] incorrectly describes that the input image is taken on the street. The sentence of Midge [3] is the best in these sentences from existing works. However, particularly addressing a bottle rather than a dinner is a bit strange for this image. The sentence of BabyTalk [4] is generally long and re-



Table 6.2: Automatic evaluation for output sentences using PASCAL Sentence dataset. Scores in parentheses are computed by matching synonyms.

	BLEU 1	BLEU 2	BLEU 3	BLEU 4	NIST 5
Kulkarni et al. [4]	0.25 (0.30)	-	-	-	-
Yang et al. [2]	(0.41)	(0.13)	(0.03)	-	-
Verma et al. [56]	0.36 (0.43)	-	-	-	-
Gupta et al. [55]	(0.54)	(0.23)	(0.07)	-	-
FV+PAAPL+naive filter [110]	-	-	-	0.07	2.65
FV+CoSMoS+naive filter	0.53	0.32	<b>0.19</b>	<b>0.11</b>	3.37
FV+CoSMoS+SWF	<b>0.56</b>	<b>0.33</b>	<b>0.19</b>	<b>0.11</b>	<b>3.45</b>

dundant. Most objects founded in the picture are described as “rusty”. Whereas other works generate incorrect or verbose sentences, our system generates a proper sentence.

Qualitative comparison with a single input is insufficient. Therefore, we also compared the accuracy of generated sentence using automatic evaluation. Table 6.2 shows a quantitative comparison. Scores in parentheses are computed by matching synonyms. In general, matching not only the same word but also its synonyms stretches the score. The table also shows that our framework can generate more-accurate sentences.

We also compared our method with several from earlier studies. The first is the accuracy using the exactly same pipeline in Chapter 4. The second score is obtained by estimating keyphrases not with PAAPL but with CoSMoS. Consequently, CoSMoS can estimate keyphrases more accurately than PAAPL. The third is obtained by filtering phrases including too many meaningless words described in Section 4.5.3. Usage of Stop Word Filter (SWF) contributes to BLEU 1/2 and NIST 5, which is natural because eliminating meaningless phrases from keyphrase contributes directly to the matching rate of unigrams and bigrams. Therefore, we use SWF for Keyphrase extraction in the following experiments.

Input Image	Estimated Keyphrases	Generated Sentence & Ground Truth
	a group group of of people people sitting a living	<p><b>A living room with a group of people sitting.</b></p> <p>Four Asian young people sitting in a den or living room. Group of kids sitting on brown couch together. The girls watch highlights of the Laker game. Three girls are sitting on a couch and smiling next to one sitting on the floor. Three young teenagers sitting on a couch with another sitting on the floor.</p>
	airplane is a grassy grass in a field grassy field	<p><b>A grassy field in front of a body of water.</b></p> <p>A beautiful lake surrounded by trees with two small boats on the beach. A boat is shown in the foreground of a river-side scene, with mountains and trees in the background. A small watercraft at the edge of a lake. The great outdoors with blue water, evergreen trees, and heavy clouds. Two boats docked on a lakeside.</p>
	a group of people group of people sitting the table	<p><b>Group of people sitting at a table with a dinner.</b></p> <p>A family sitting down for dinner. Three people posing at the dinner table. three people sitting at a table with food and wine Three people sitting at dinner table smiling at camera. Two elderly women sit at a dinner table with a man.</p>
	a desk desk with computer and a computer table with	<p><b>Table with a desk with a computer and a chair.</b></p> <p>An office cube has a desktop computer, a cluttered desk, and a blue office chair. An office workstation with personal belongings. Office desk with computer monitor on. Office space with desktop computer and blue office chair. The office has computer on the table and other miscellaneous items.</p>
	room with decker bus a room desk with a bus	<p><b>A living room with a view of a television.</b></p> <p>A television and entertainment center. A wide screen TV in a green entertainment center. A wide screen TV sits in a living room. Entertainment center with TV and picture frames. The entertainment center of the living room includes a TV, plants, and several baskets.</p>

Figure 6.2: Good examples of estimated keyphrases and generated sentences for the PASCAL Sentence dataset. The first column shows input images. The second column shows estimated keyphrases for each input image. The third column shows the generated sentence at the top and ground truth in the dataset at the bottom. Red-colored words in generated sentences derive from estimated keyphrases.

Input Image	Estimated Keyphrases	Generated Sentence & Ground Truth
	the sheep sheep in field of the grass a sheep	<b>A sheep in grassy field of sheep standing on grass.</b> A sheep A sheep standing angst wild flowers and grasses. A white sheep standing on a field next to wild flowers The white sheep is standing next to the plant. Two lambs stand on a grassy hill.
	bus parked is parked white bulldog parked in parked on	<b>E1 bus is parked near a white bus parked.</b> A large white "Victory Liner" bus with red and yellow trim is in a parking lot. A side view of a passenger bus. A Victory Liner bus is white with red and gold stripes. A white tour bus with red, orange and yellow stripes. The bus has a red, orange and yellow design on it.
	the sky is flying blue shopping flying in airplane is	<b>Airplane is flying in the blue sky above the air.</b> A D-ERFW-6 in flight. An army green plane flying in the sky. An old fighter plane flying with German military markings. A small green and yellow plane in the sky. A WWII fighter plane with its landing gear down.
	horse grazing horse whose grass in a horse brown horned	<b>Brown horse with a horse in a field with trees.</b> A brown horse and a tan calf in a pasture. A brown horse standing near a fowl in a grassy paddock. Adult horse and foal standing in middle of grass arena. Two horses are grazing in a field. Two horses in fenced in field.
	living room room with woman with and woman holding a	<b>A buddy holding a woman with a living room.</b> A woman has a bird on her shoulder, and another bird on her head A woman with a bird on her head and a bird on her shoulder. A women sitting at a dining table with two small birds sitting on her. A young Asian woman sitting at a kitchen table with a bird on her head and another on her shoulder. Two birds are perched on a woman sitting in a kitchen.

Figure 6.3: Partially incorrect examples of estimated keyphrases and generated sentences for the PASCAL Sentence dataset. The first column shows input images. The second column has estimated keyphrases for each input image. The third column shows the generated sentence at the top and ground truth in the dataset at the bottom. Red-colored words in generated sentences come from estimated keyphrases.

Table 6.3: Automatic evaluation for output sentences using IAPR-TC12 dataset. Scores in parentheses are computed by matching synonyms.

Method	BLEU				NIST
	1	2	3	4	5
Gupta et al. [55]	0.15 (0.21)	0.06 (0.07)	0.01 (0.01)	-	-
Ours	<b>0.60</b>	<b>0.40</b>	<b>0.28</b>	<b>0.20</b>	<b>3.73</b>

## 6.2 IAPR-TC12

Figure 6.4 presents some examples of generated sentences for the IAPR-TC12 dataset. Three images at the top are regarded as correct examples, although the other two images in the bottom are regarded as incorrect.

Table 6.3 presents a quantitative comparison. The table also shows that our framework can generate more accurate sentences. We achieve state-of-the-art performance over these datasets.

As described in this thesis, the PASCAL Sentence dataset is generated using PASCAL VOC dataset, where objectives of 20 kinds are treated. Each image in PASCAL Sentence is associated with about five sentences. As used in evaluation for Image Annotation, IAPR-TC12 has 291 labels. Although each image in IAPR-TC12 is associated only with 1.8 sentences, we find that the performance is comparable to those using PASCAL Sentence. The relations between images and keyphrases are apparently learned by CoSMoS.

## 6.3 SBU

Figure 6.5 presents some examples of generated sentences for the SBU dataset. Three images in the top are regarded as correct examples, whereas the other two images in the bottom are regarded as incorrect. Although there are many noisy descriptions as [39] tries to generalize them, the results show that our methodology to generate sentences for images can generate sentences from a large-scale dataset collected from the web.

Table 6.4 presents a quantitative comparison. These experimentally obtained results show the superiority of our framework. However, the scores are inferior to scores obtained using PASCAL Sentence and IAPR-TC12. There are mainly two reasons. First, the SBU dataset with 1M images naturally handles many more contents than the other datasets with at most thousands of images. Secondly, because testing images are also images in SBU, the sentences are sometimes improper for ground truth.

Input Image	Estimated Keyphrases	Generated Sentence & Ground Truth
	the background mountain range mountains in range in snow-covered summit	<p><b>A green trees and brown mountain range in the background.</b></p> <p>A steep, grey canyon in the middle of a green valley with trees and houses, and a brownish, bald mountain in the background.</p>
	the background tourists are standing are sitting the middle	<p><b>Tourists are standing on the middle of a flat desert.</b></p> <p>People are looking at rocks in the middle of a desert landscape.</p>
	the background a brown landscape with a flat brown monument	<p><b>Green trees and a flat landscape with a brown mountains.</b></p> <p>A gravel road through a dry and bald plateau. There are mountains in the distant background.</p>
	made of in front a grey front of grey-brown stones	<p><b>Dark brown mountains in front of a grey and brown.</b></p> <p>A man is standing in front of a wall made of large, grey bricks with two small windows.</p>
	a dark-skinned wearing a white a red in front	<p><b>A dark-skinned boy wearing a grey sky in front.</b></p> <p>A dark-skinned, dark-haired boy with a big smile wearing a red cap and a dark blue anorak. Short grass in background.</p>

Figure 6.4: Examples of estimated keyphrases and generated sentences for the IAPR-TC12 dataset. The first column presents input images. The second column has estimated keyphrases for each input image. The third column shows the generated sentence at the top and ground truth in the dataset at the bottom. Red-colored words in generated sentences come from estimated keyphrases. Three images in the top are regarded as correct examples, whereas the other two images in the bottom are regarded as incorrect.

Table 6.4: Automatic evaluation for output sentences using SBU dataset. Scores in parentheses are computed by matching synonyms.

Method	BLEU				NIST
	1	2	3	4	5
Ordonez et al. [38]	0.13	-	-	-	-
Kuznetsova et al. [54]	0.11 (0.11)	-	-	-	-
Ours	<b>0.20</b>	<b>0.09</b>	<b>0.04</b>	<b>0.02</b>	<b>1.15</b>

As described in Chapter 1, our objective is to generate sentences using only pairs of an image and sentences collected from the web. Actually, we find that web images and their sentences are useful for training our system. However, using those pairs is inappropriate for automatic evaluation for generated sentences. Automatic evaluation should be performed using human-generated references. Therefore, we evaluate the ability to generate sentences for images in PASCAL Sentence dataset after learning the pairs in the SBU dataset later.

## 6.4 Discussion

In this section, we first compare our performance to human-generated sentences. Secondly, we evaluate our method to generate sentences from estimated keyphrases by introducing oracle keyphrase estimation. Thirdly, we apply the system trained on SBU to PASCAL Sentence. Simultaneously, we investigate the increase of performance by enlarging a dataset.

### 6.4.1 Evaluating human-generated sentences

We have evaluated our methodology using automatic evaluation. The results show that our system is superior to other works. They use external datasets for special problems for visual recognition such as object detection, scene recognition, and action detection. Such datasets should be compiled according to one aspect. Therefore, automatic collection from the web is difficult. In other words, collecting images and extracting only those words representing its scene or action is a difficult problem. Our methods use only pairs of an image and sentences. Such data can be collected easily from the web. In fact, we can generate sentences using SBU dataset, which is collected from Flickr.

However, it remains unknown whether our system can generate sentences perfectly or not. Although the upper bound of BLEU is one, even humans are unable to generate a new sentence scoring one with BLEU unless exactly the


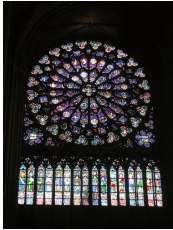



Input Image	Estimated Keyphrases	Generated Sentence & Ground Truth
	blue sky clock tower blue in sky EOS tower in	<b>Clock tower in the city of the blue sky.</b> The clock tower at Sydney Uni against a perfect blue sky.
	stained glass glass window window in in St. the church	<b>Stained glass window in the church in St. Vitus Cathedral.</b> Stained glass window in Notre Dame.
	the mountain the sun the summit a photo took a	<b>Took a photo of the mountain in the sun sets.</b> Sunrise over the mountains on the way home on the coach.
	an old a house the middle house on old house	<b>An old house on a house in the middle.</b> Found this old passenger train coach in LaSalle IL.
	the corner the door house EOS place EOS door to	<b>In the corner of the door to the second floor.</b> Photo class. The assignment was 'Identity'. Anyway, these are some books on the shelf in my room.

Figure 6.5: Examples of estimated keyphrases and generated sentences for the SBU dataset. The first column shows input images. The second column has estimated keyphrases for each input image. The third column shows the generated sentence at the top and ground truth in the dataset at the bottom. Red-colored words in generated sentences derive from estimated keyphrases. Three images at the top are regarded as correct examples, although the other two images at the bottom are regarded as incorrect.

Table 6.5: Automatic evaluation for human-generated sentences using a PASCAL Sentence dataset. Scores in parentheses are computed by matching synonyms.

		BLEU 1	BLEU 2	BLEU 3	BLEU 4	NIST 5
Human	[4, 43]	0.50	-	-	-	-
	[42]	0.64 (0.66)	0.42 (0.44)	0.24 (0.26)	-	-
	Ours	0.64	0.43	0.31	0.23	6.27
Computer	Ours	0.56	0.33	0.19	0.11	3.45

same sentence is rewritten.

Therefore, we evaluate human-generated sentences on the dataset. Particularly we evaluate the sentences in PASCAL Sentence in a leave-one-out manner.

Table 6.5 shows the performance of human-generated sentences on PASCAL dataset. Two existing reports describe human-generated sentences. The performance we obtain is similar to that described in [42]. In comparison to human-generated sentences, the generated sentences from our system can use sufficient vocabularies because our BLEU 1 score is comparable to that produced by humans. However, shortage of BLEU 4 reflects that the connection of these vocabularies must be improved.

## 6.4.2 Oracle keyphrase estimation

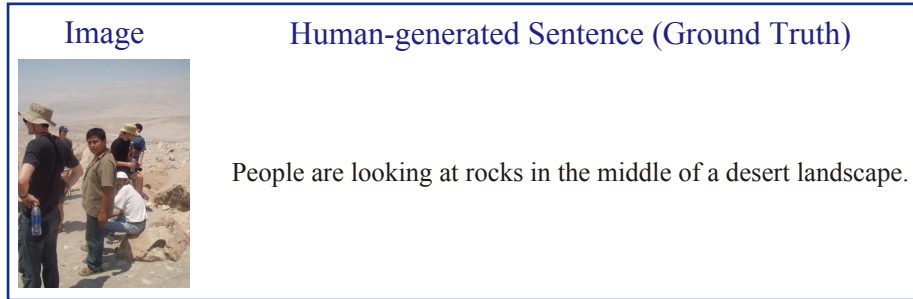
As described in this thesis, our methodology consists of two steps: keyphrase estimation from input images and sentence generation from those keyphrases. Both are challenging problems. Therefore, we would like to evaluate not only CoSMoS but also the modified multi-stack beam search and the cost functions.

Therefore, we generate sentences “oracle” keyphrases extracted from the ground truth sentences. For example shown in Figure 6.6, in the usual problem setting of sentence generation for images, a sentence is generated from the input image at the left side. Here, we evaluate our sentence generation system by generating a sentence from correct keyphrases that exist in the ground truth.

Table 6.6 presents the result of sentence generation from oracle keyphrases. Table 6.5 and Table 6.6 show that our sentences generated from oracle keyphrases and reference sentences are more mutually similar than a human-generated sentence and another sentence generated by humans.

This experiment is related to [42], where sentences are generated from pairs of an image and labels for objects and attributes. Therefore, our sentences from oracle keyphrases are more accurate because the oracle keyphrases inform us not only of objects and attributes but also their relations.





(i) Usual evaluation setting



People are looking at rocks in the middle of a desert landscape.

(ii) Special setting with oracle keyphrases

looking at  
at rocks  
rocks in  
desert landscape



People are looking at rocks in the middle of a desert landscape.

Figure 6.6: Rough illustration of evaluation for sentence generation from oracle keyphrases.

Table 6.6: Automatic evaluation for output sentences from oracle keyphrases and estimated keyphrases. We use PASCAL Sentence and IAPR-TC12 because both have human-generated sentences.

Dataset		BLEU				NIST
		1	2	3	4	5
PASCAL Sentence	CoSMoS	0.56	0.33	0.19	0.11	3.45
	Gupta et al. [42]	0.74	0.55	0.35	-	-
	Oracle	0.82	0.71	0.56	0.42	7.64
	Upper bound	1	1	1	1	(15.1)
IAPR-TC12	CoSMoS	0.60	0.40	0.28	0.20	3.73
	Gupta et al. [42]	0.33	0.18	0.07	-	-
	Oracle	0.74	0.61	0.48	0.37	6.26
	Upper bound	1	1	1	1	-



Learned using SBU

It is a picture of the boat in the water.

Learned using PASCAL Sentence

A boat with trees in the ocean with a river.

Figure 6.7: Comparison of two sentences generated by learning different datasets. The one at the top is generated after learning SBU dataset consisting of 1M web images, the other at the bottom is generated after learning PASCAL Sentence consisting of 1K well-organized images.

### 6.4.3 Describing PASCAL Sentence using SBU

We generate sentences for images of PASCAL Sentence after learning pairs of an image and a sentence of SBU. The objective of this experiment is to investigate (i) if we can generate sentences using not manually organized data but automatically collected web data, and (ii) the impact of dataset size of SBU.

First, we train our system using about 1M images from SBU and generate sentences for 100 PASCAL Sentence images. One example is shown in Figure 6.7. Even if the dataset is not manually organized, we can generate a sentence using numerous images compiled from the web. Secondly, we reduce the number of images from SBU and generate sentences similarly. Particularly we use 1K, 10K, 100K, and about 1M images.

Figure 6.8 and Figure 6.9 respectively present examples of sentences generated using a varying number of SBU datasets respectively for PASCAL Sentence images and for SBU images. All sentences are generated using the same grammar model extracted from 1M sentences in SBU. Therefore, the improvement of these sentences derives from the improvement of keyphrase estimation. These results demonstrate that sentences can be improved when the number of images is increased.

Figure 6.10 and Figure 6.11 respectively show BLEU and NIST scores up to the full-size dataset. Although the BLEU scores obtained using SBU are inferior to that using PASCAL Sentence itself, the scores are still higher than those of several existing works. Moreover, as this figure shows, increasing the dataset improves especially NIST score. Because NIST emphasizes less frequent N-grams and NIST improvement also mean that our system comes to learn less frequent keyphrases when the number of images is increased.



Input Image	# of Images	Generated Sentence
	1K	Is a train station in the lake in the small.
	10K	All the lake in the water is a shot.
	100K	View of the lake in the water in a boat.
	1M	It is a picture of the boat in the water.
	1K	Building a 5D2 from a bar in the evening sky.
	10K	To my desk in the box in the little girl.
	100K	Fienile master bedroom window in the house in my office.
	1M	Desk in the kitchen table in the wall.

Figure 6.8: Examples of sentences generated for PASCAL Sentence images using a varying number of SBU datasets.



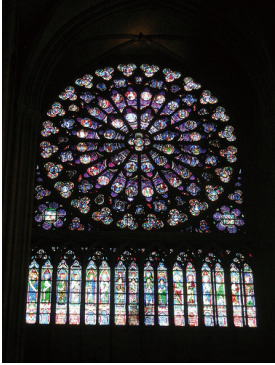

Input Image	# of Images	Generated Sentence
	1K	On the 13h floor in the beach at the park.
	10K	Hat in the roof of the castle in the tower.
	100K	Like the backfgnd of our house in the bottom.
	1M	An office building near the roof of the building.
	1K	On the house in the other side of the grass.
	10K	Water in arugut river in 003b this is the sun.
	100K	Is a stone wall in the reflection of the water.
	1M	Edge of the water in a shot in the background.
	1K	Stained glass window in aanbouw cofferdam for a field.
	10K	Window in the ossuary glass windows in St. Louis Missouri.
	100K	Stained glass in the tower of the church in St.
	1M	Stained glass window in the church in St. Vitus Cathedral.
	1K	The best=est dog in a little girl in the water.
	10K	Through the bubban covered in a tree in my office.
	100K	Loved the contrast of a picture of my favorite tree.
	1M	Of a bird on a tree in the blue sky.

Figure 6.9: Examples of sentences generated for SBU images using a varying number of SBU datasets.

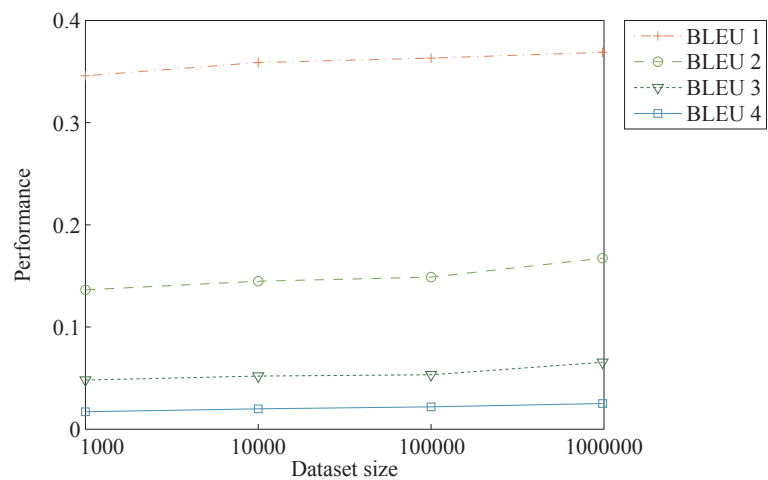


Figure 6.10: Impact of the dataset size evaluated using the BLEU score.

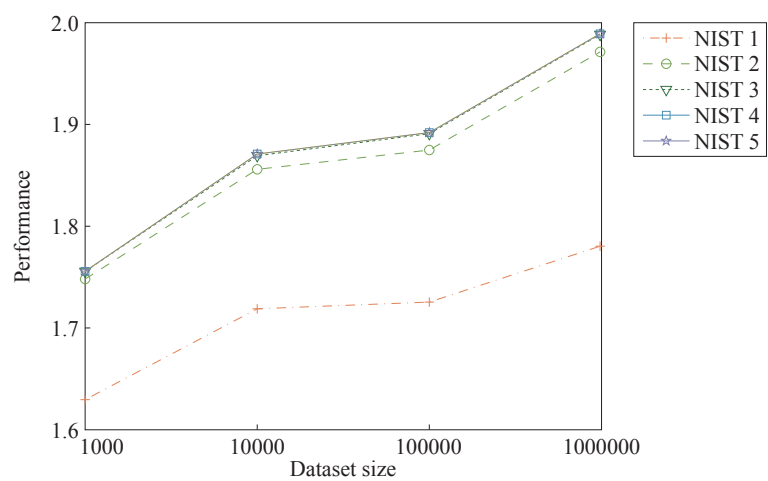


Figure 6.11: Impact of the dataset size evaluated using the NIST score.

# Chapter 7

## Conclusion and Future Work

### 7.1 Conclusion

Generating sentences to explain images is an ultimate goal of generic object recognition. Most existing works require semantic knowledge such as ⟨object, action, scene⟩. Such labels with attributes should be labeled manually. Therefore, compiling a large-scale dataset is difficult. In this thesis, we develop a system to generate sentences for images using only pairs of an image and sentences. To realize sentence generation using only images and sentences, we present a *Multi-keyphrase Problem* to estimate keyphrases and to generate a sentence by connecting the keyphrases using a grammar model.

### Guidelines of Online Learning Methods for Large Scale Visual Recognition (Chapter 3)

As described herein, we gave qualitative and quantitative comparisons of these online learning algorithms. To date, no report has described a study investigating state-of-the-art algorithms for visual recognition or a study evaluating those algorithms in unified experimental settings. When these algorithms were proposed, toy data and the NLP dataset were used for evaluation. Comparison using conventional settings for visual recognition must be conducted. Finally, this chapter presents three guidelines based on results of image classification as the following.

1. Perceptron can compete against the latest algorithms.
  - Provided that the second guideline is observed.
2. Averaging is necessary for any algorithm.
  - First-order algorithms w/o averaging cannot compete against second-order algorithms.

- 
- When averaging is used, the accuracies of all algorithms mutually converge.
  - Averaging accelerates not only first-order algorithms, but also second-order algorithms.
3. Investigate multiclass learning first.
    - Both one-versus-the-rest learning and multiclass learning achieve similar accuracy.
    - However, one-versus-the-rest takes much longer CPU time to converge than multiclass does.

## Sentence Generation via Keyphrase Estimation (Chapter 4)

Existing methods to describe images sentimentally require semantic knowledge such as labels of an object, action, or scene. Using these methods, we must strive to prepare a highly organized dataset.

In this chapter, we propose a novel approach to generate sentences from images. We present *Multi-keyphrase Problem* to estimate keyphrases and to generate a sentence by connecting the keyphrases using a grammar model. Our method requires only pairs of an image and an associated sentence. Manual preparation of semantic knowledge such as subjects, actions, and scenes is not necessary. Therefore, we propose a novel online learning method for multi-keyphrase estimation: Passive-Aggressive with Averaged Pairwise Loss (PAAPL).

The proposed framework, although simple and scalable, can generate sentences from images with no semantic knowledge. Experimental results demonstrate that sentences can be generated using the proposed framework. The accuracy of generated sentences is better than that of existing methods. Moreover, PAAPL, which is proposed for multi-keyphrase estimation, which is applicable to image annotation, is superior in terms of scalability and performance on image annotation.

However, its accuracy remains low, mainly because there are many more *keyphrases* than labels in usual annotation datasets. These many *keyphrases* require too many parameters for classifiers. Therefore, the space complexity presents a problem. Our next work is therefore, the development of a learning method to learn many labels more correctly with fewer parameters.

## CoSMoS: Common Subspace for Model and Similarity (Chapter 5)

This chapter presents a novel subspace method that simultaneously (1) narrows a semantic gap by learning a subspace where images with the same label become

---

close mutually, and (2) learns models as linear weight vectors in the subspace for each label.

The proposed method can be regarded as the integration of two approaches: **model** learning such as SVM using a linear weight, and **similarity** learning among images. Learning the mapping via classification loss enables CoSMoS to weight according to the performances from *model* and *similarity* for each class.

We also report that Canonical Contextual Distance (CCD) [107] implicitly optimizes the subspace in a similar policy. By introducing Averaged Pairwise Loss and averaged stochastic gradient descent, we explicitly optimize the subspace and achieve state-of-the-art performance and scalability in benchmark datasets for image annotation.

Experimental results for three datasets for image annotation show that the proposed CoSMoS achieves state-of-the-art performance.

## Evaluation of Sentential Description for Images (Chapter 6)

We evaluated our methodology using three datasets: PASCAL Sentence, IAPR-TC12, and SBU. The results show that our system can generate sentences more accurately than the other works. The other works use external datasets for special problems for visual recognition such as object detection, scene recognition, and action detection. Our methods use only pairs of an image and sentences. Such data can be collected easily from the web. Our system can generate appropriate sentences for images using the SBU dataset consisting of 1M images, which is collected from Flickr. The scalability of our system and experimentally obtained results with varying size of dataset show that the accuracy increases when the dataset increases.

## 7.2 Unsolved Problems and Future Works

In this thesis, we present the *Multi-keyphrase Problem* to generate sentences for images. From a thorough comparison of online learning methods for large-scale visual classification, we propose a novel online learning method for keyphrase estimation. This method is shown to be capable of learning many labels accurately.

Because keyphrase estimation is a bottleneck to generate sentences, the proposal of CoSMoS in this thesis contributes greatly to the performance of the generated sentences. Therefore, our future works will address the following topics.



---

## Discontinuous Keyphrases

Our objective of the use of keyphrases is learning not only objects, actions, and attributes but also their relations. For simplicity, continuous phrases are used as keyphrases, which is reasonable because the order of words represents relations among English words. A continuous phrase naturally represents the relations among the words in the phrase. However, some distantly positioned words are inferred as representing important relations.

For example, as described in Chapter 2, our objective is to generate a sentence such as “A man bites a white dog in his arms.” from keyphrases such as “man-bites”, “white-dog”, and “his-arms”. Actually, continuous phrases can treat these relations. However, relations such as “bites—dog” cannot be extracted. Although we can manage to extract the relation by introducing four-word phrases such as “bites a white dog”, the use of such long phrases is difficult because the number of phrases would be too large and because the frequency of long phrases would be too low to train classifiers.

Therefore, one avenue of future work is the investigation of discontinuous keyphrases. Such a tide is also happened in literature of Natural Language Processing. [112] proposes phrase-based machine translation. Their “phrases” also represent a sequence of words and extraction from a parallel corpus. Afterward, [113] proposed hierarchical phrase-based translation and achieved superior performance. The hierarchical structures of sentences are not linguistically syntax-based. As [113] reports, hierarchical phrases are extracted from a parallel corpus instead of extraction of syntax structures as in truly syntax-based translation [114]. Moreover, [115] proposes a non-hierarchical approach to extract discontinuous phrases.

The greatest problem hindering extraction of discontinuous keyphrases is that the input structure differs greatly from that of output. In machine translation, both input and output are sentences, i.e., sequences of symbols. In sentence generation from images, only output is a sequence of symbols whereas input is a 2D array of real numbers, i.e., pixels. Therefore, finding the alignment between pixels and discontinuous phrases is difficult. Without such an alignment, we should consider all combinations among all words.

A possible solution to introduce discontinuous keyphrases is a top-down definition of keyphrases based on parts of speech such as “(Noun)–(Verb)”. As extracted in [54, 55, 56], we can extract phrases using a parser. With these discontinuous keyphrases, we should formulate a novel search method to connect these keyphrases.

---

## More Sophisticated Sentence Generation from Keyphrases

As the experimentally obtained results in Chapter 6 demonstrate, sentence generation from estimated keyphrases must be improved.

One solution is the use of more sophisticated grammar model. By collecting more sentences from an external dataset, it is possible to generate a more stable grammar model. The reason we extract a grammar model from only each dataset is that we would like to evaluate the performance using only each dataset with no external data for fair comparison.

When discontinuous keyphrases are extracted as described in the last subsection, the proposed multi-stack beam search should be modified further to treat those keyphrases as in [115].

Additionally, as shown in Figure 6.3, resolving redundant expression such as “A sheep in a grassy field of sheep standing on grass.” is preferred. The proposed beam search encourages the use of estimated keyphrases. Improved search should not only encourage the use of some words but also discourage the inclusion of some words overlapping the sentence that is generated. One naive solution is consideration of all synonyms as one word. Because all synonyms such as “Mac Air” and “ultrabook” cannot be organized manually, however, the use of an existing synonym dictionary is suboptimal. Moreover, overlapping expressions are not limited to synonyms. For example, the use of too many verbs in a sentence and the use of too many words modifying the same word should be avoided.

## Toward Sentence Generation for Individual Users

Generally, sentences for one image should vary among individual users. Our future work should address adaptation to individuals. To adapt sentences to each user, users should input some feedback to the system. For usability, adapting with a slight amount of feedback is preferred. We believe that our keyphrase estimation with CoSMoS is suitable for learning with a small amount of feedback. Such adaptation is necessary for the true goal of interpreting objects and events in the real world for life-log systems and robots working in our dwelling environment.

# References

- [1] Yoshitaka Ushiku, Tatsuya Harada, and Yasuo Kuniyoshi. Automatic sentence generation from images. In *Proceedings of ACM International Conference on Multimedia*, pages 1533–1536, 2011. [viii](#), [12](#), [13](#), [14](#), [45](#), [50](#), [56](#)
- [2] Yezhou Yang, Ching Lik Teo, Hal Daumé III, and Yiannis Aloimonos. Corpus-guided sentence generation of natural images. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 444–454, 2011. [ix](#), [12](#), [38](#), [52](#), [75](#), [76](#)
- [3] Margaret Mitchell, Jesse Dodge, Amit Goyal, Kota Yamaguchi, Karl Stratos, Xufeng Han, Alyssa Mensch, Alex Berg, Tamara Berg, and Hal Daumé III. Midge: Generating image descriptions from computer vision detections. In *Proceedings of European Chapter of the Association for Computational Linguistics*, 2012. [ix](#), [12](#), [14](#), [38](#), [75](#)
- [4] Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Baby talk: Understanding and generating image descriptions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011. [ix](#), [12](#), [38](#), [52](#), [75](#), [76](#), [83](#)
- [5] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *Proceedings of Language Resources and Evaluation Conference*, pages 449–454, 2006. [xii](#), [55](#)
- [6] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *Proceedings of European Conference on Computer Vision*, pages 15–28, 2010. [1](#), [11](#), [12](#), [38](#), [49](#), [52](#)
- [7] Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S.

## REFERENCES

---

- Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990. 6
- [8] Pinar Duygulu, Kobus Barnard, Nando De Freitas, and David Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of European Conference on Computer Vision*, pages 349–354, 2002. 6, 8, 46, 68
- [9] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Greg Sanders, Wessel Kraaij, Alan F. Smeaton, and Georges Queenot. Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2013*, 2013. 7
- [10] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop on Generative-Model Based Vision*, volume 106, 2004. 8
- [11] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, 2006. 8, 9, 28
- [12] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007. 8
- [13] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of ACM International Conference on Human Factors in Computing Systems*, pages 319–326, 2004. 8, 47, 68
- [14] Michael Grubinger, Paul Clough, Henning Müller, and Thomas Deselaers. The iapr tc-12 benchmark: A new evaluation resource for visual information systems. In *International Workshop OntoImage*, pages 13–23, 2006. 8, 11, 47, 68
- [15] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: A real-world web image database from national university of singapore. In *Proceedings of ACM International Conference on Image and Video Retrieval*, 2009. 8
- [16] 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–70, 2008. 8

## REFERENCES

---

- [17] Xin-Jing Wang, Lei Zhang, Ming Liu, Yi Li, and Wei-Ying Ma. Arista – image search to annotation on billions of web photos. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2987–2994, 2010. [9](#)
- [18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. [9](#), [18](#), [27](#)
- [19] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. [9](#), [18](#)
- [20] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010. [9](#)
- [21] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Proceedings of European Conference on Computer Vision Workshop on Statistical Learning in Computer Vision*, pages 1–22, 2004. [9](#), [18](#)
- [22] Gert R. G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004. [10](#)
- [23] Alain Rakotomamonjy, Francis R. Bach, Stéphane Canu, and Yves Grandvalet. Simplemkl. *Journal of Machine Learning Research*, 9:2491–2521, 2008. [10](#)
- [24] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1794–1801, 2009. [10](#), [18](#)
- [25] Y-Lan Boureau, Francis Bach, Yann LeCun, and Jean Ponce. Learning mid-level features for recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2559–2566, 2010. [10](#), [17](#), [18](#)
- [26] Xi Zhou, Kai Yu, Tong Zhang, and Thomas S. Huang. Image classification using super-vector coding of local image descriptors. In *Proceedings of European Conference on Computer Vision*, pages 141–154, 2010. [10](#), [18](#)

## REFERENCES

---

- [27] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. [10](#)
- [28] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *Proceedings of European Conference on Computer Vision*, pages 143–156, 2010. [10](#), [18](#), [27](#), [28](#), [41](#), [47](#), [73](#)
- [29] Jiwoon Jeon, Victor Lavrenko, and Raghavan Manmatha. Automatic image annotation and retrieval using cross-media relevance models categories and subject descriptors. In *Proceedings of ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 119–126, 2003. [10](#), [40](#)
- [30] Ameesh Makadia, Vladimir Pavlovic, and Sanjiv Kumar. A new baselines for image annotation. In *Proceedings of European Conference on Computer Vision*, pages 88–105, 2008. [10](#), [40](#), [47](#), [48](#), [50](#), [68](#), [69](#)
- [31] Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek, and Cordelia Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *Proceedings of IEEE International Conference on Computer Vision*, pages 309–316, 2009. [10](#), [40](#), [46](#), [47](#), [48](#), [50](#), [60](#), [65](#), [68](#), [69](#)
- [32] Hideki Nakayama. *Linear Distance Metric Learning for Large-scale Generic Image Recognition*. PhD thesis, The University of Tokyo, 2011. [10](#), [40](#), [48](#), [50](#), [60](#), [63](#), [64](#), [65](#), [68](#), [69](#)
- [33] Yashaswi Verma and C. V. Jawahar. Image annotation using metric learning in semantic neighbourhoods. In *Proceedings of European Conference on Computer Vision*, pages 836–849, 2012. [10](#), [60](#), [68](#), [69](#)
- [34] Mohammad Amin Sadeghi and Ali Farhadi. Recognition using visual phrases. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1745–1752, 2011. [11](#)
- [35] Benjamin Z. Yao, Xiong Yang, Liang Lin, Mun Wai Lee, and Song-Chun Zhu. I2t: Image parsing to text description. *Proceedings of the IEEE*, 98(8):1485–1508, 2010. [11](#), [52](#)
- [36] Ahmet Aker and Robert Gaizauskas. Generating image descriptions using dependency relational patterns. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, number July, pages 1250–1258, 2010. [11](#), [52](#)

## REFERENCES

---

- [37] Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. Collecting image annotations using amazon’s mechanical turk. In *Proceedings of NAACL HLT Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, number June, pages 139–147, 2010. [11](#), [49](#)
- [38] Vicente Ordonez, Girish Kulkarni, and Tamara L Berg. Im2text: Describing images using 1 million captioned photographs. In *Advances in Neural Information Processing Systems*, pages 1–9, 2011. [11](#), [13](#), [38](#), [39](#), [52](#), [81](#)
- [39] Polina Kuznetsova, Vicente Ordonez, Alexander Berg, Tamara Berg, Yejin Choi, and Stony Brook. Generalizing image captions for image-text parallel corpus. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, 2013. [11](#), [79](#)
- [40] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013. [11](#), [38](#)
- [41] Micah Hodosh, Peter Young, Cyrus Rashtchian, and Julia Hockenmaier. Cross-caption coreference resolution for automatic image understanding. In *Proceedings of Conference on Computational Natural Language Learning*, number July, pages 162–171, 2010. [11](#)
- [42] Ankush Gupta and Prashanth Mannem. From image annotation to image description. In *Proceedings of International Conference on Neural Information Processing*, pages 1–8, 2012. [12](#), [38](#), [52](#), [83](#), [84](#)
- [43] Siming Li, Girish Kulkarni, Tamara L. Berg, Alexander C. Berg, and Yejin Choi. Composing simple image descriptions using web-scale n-grams. In *Proceedings of Conference on Computational Natural Language Learning*, 2011. [12](#), [14](#), [38](#), [52](#), [57](#), [83](#)
- [44] Margaret Mitchell, Xufeng Han, and Jeff Hayes. Midge: Generating descriptions of images. In *Proceedings of International Natural Language Generation Conference*, number May, pages 131–133, 2012. [12](#), [14](#), [38](#)
- [45] Yansong Feng and Mirella Lapata. How many words is a picture worth? automatic caption generation for news images. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, 2010. [12](#)
- [46] Hideki Nakayama, Tatsuya Harada, and Yasuo Kuniyoshi. Evaluation of dimensionality reduction methods for image auto-annotation. In *Proceedings of British Machine Vision Conference*, pages 94.1–94.12, 2010. [12](#), [40](#), [50](#), [63](#), [69](#)

## REFERENCES

---

- [47] Matthew B. Blaschko and Christoph H. Lampert. Correlational spectral clustering. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. [13](#)
- [48] Yoshitaka Ushiku, Tatsuya Harada, and Yasuo Kuniyoshi. Improving image similarity measures for image browsing and retrieval through latent space learning between images and long texts. In *Proceedings of International Conference on Image Processing*, pages 2365–2368, 2010. [13](#)
- [49] Andrei Barbu, Alexander Bridge, Zachary Burchill, Dan Coroian, Sven Dickinson, Sanja Fidler, Aaron Michaux, Sam Mussman, Siddharth Narayanaswamy, Dhaval Salvi, Lara Schmidt, Jiangnan Shangguan, Jeffrey Mark Siskind, Jarrell Waggoner, Song Wang, Jinlian Wei, Yifan Yin, and Zhiqi Zhang. Video in sentences out. In *Proceedings of Conference on Uncertainty in Artificial Intelligence*, pages 102–112, 2012. [13](#)
- [50] Duo Ding, Florian Metze, Shourabh Rawat, Peter F. Schulam, and Susanne Burger. Generating natural language summaries for multimedia. In *Proceedings of International Natural Language Generation Conference*, pages 128–130, 2012. [13](#)
- [51] Haonan Yu and Jeffrey Mark Siskind. Grounded language learning from video described with sentences. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 53–63, 2013. [13](#)
- [52] Pradipto Das, Chenliang Xu, Richard F. Doell, and Jason J. Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013. [13](#)
- [53] Niveda Krishnamoorthy, Girish Malkarnenkar, Raymond Mooney, Kate Saenko, and Sergio Guadarrama. Generating natural-language video descriptions using text-mined knowledge. In *Proceedings of NAACL HLT Workshop on Vision and Language*, number June, pages 10–19, 2013. [13](#)
- [54] Polina Kuznetsova, Vicente Ordonez, Alexander C. Berg, Tamara L. Berg, and Yejin Choi. Collective generation of natural image descriptions. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 359–368, 2012. [14](#), [15](#), [38](#), [52](#), [81](#), [92](#)
- [55] Ankush Gupta, Yashaswi Verma, and C. V. Jawahar. Choosing linguistics over vision to describe images. In *Proceedings of AAAI Conference on Artificial Intelligence*, 2012. [14](#), [15](#), [52](#), [76](#), [79](#), [92](#)



- 
- [56] Yashaswi Verma, Ankush Gupta, Prashanth Mannem, and C.V. Jawahar. Generating image descriptions using semantic similarities in the output space. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop on Language for Vision*, pages 288–293, 2013. [14](#), [15](#), [52](#), [76](#), [92](#)
- [57] Kilian Q. Weinberger, John Blitzer, and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems*, 2006. [14](#)
- [58] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of International Conference on Computational Statistics*, number x, pages 177–187, 2010. [15](#), [17](#), [18](#), [19](#), [20](#), [21](#), [25](#)
- [59] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive–aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006. [15](#), [17](#), [18](#), [19](#), [22](#), [24](#), [25](#), [41](#), [43](#), [59](#), [66](#)
- [60] Koby Crammer, Mark Dredze, and Fernando Pereira. Exact convex confidence-weighted learning. In *Advances in Neural Information Processing Systems*, pages 345–352, 2008. [15](#), [17](#), [18](#), [19](#), [23](#), [25](#)
- [61] Koby Crammer, Alex Kulesza, and Mark Dredze. Adaptive regularization of weight vectors. In *Advances in Neural Information Processing Systems*, volume 22, pages 414–422, 2009. [15](#), [17](#), [18](#), [19](#), [23](#), [25](#)
- [62] Koby Crammer and Daniel D. Lee. Learning via gaussian herding. In *Advances in Neural Information Processing Systems*, pages 1–9, 2010. [15](#), [17](#), [18](#), [19](#), [23](#), [24](#), [25](#), [41](#), [66](#)
- [63] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization. *Psychological Review*, 65(6):386–408, 1958. [15](#), [17](#), [18](#), [19](#), [20](#), [25](#)
- [64] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *Proceedings of International Joint Conference on Artificial Intelligence*, number x, pages 2764–2770, 2011. [15](#), [16](#), [17](#), [41](#), [44](#), [60](#), [62](#), [67](#), [69](#)
- [65] Jia Deng, Alexander C Berg, Kai Li, and Li Fei-fei. What does classifying more than 10,000 image categories tell us? In *Proceedings of European Conference on Computer Vision*, pages 71–84, 2010. [16](#)

- 
- [66] Yuanqing Lin, Fengjun Lv, Shenghuo Zhu, Ming Yang, Timothee Cour, Kai Yu, Liangliang Cao, and Thomas Huang. Large-scale image classification: Fast feature extraction and svm training. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1689–1696, 2011. [16](#), [18](#), [21](#), [26](#), [28](#)
- [67] Jorge Sánchez and Florent Perronnin. High-dimensional signature compression for large-scale image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1665–1672, 2011. [16](#), [18](#), [28](#), [47](#)
- [68] Ken Chatfield, Victor Lempitsky, Andrea Vedaldi, and Andrew Zisserman. The devil is in the details: An evaluation of recent feature encoding methods. In *Proceedings of British Machine Vision Conference*, number 1, pages 76.1–76.12, 2011. [17](#), [18](#), [28](#)
- [69] Koby Crammer, Mark Dredze, and Alex Kulesza. Multi-class confidence weighted algorithms. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 496–504, 2009. [17](#), [18](#), [23](#)
- [70] Mark Dredze, Koby Crammer, and Fernando Pereira. Confidence-weighted linear classification. In *Proceedings of International Conference on Machine Learning*, pages 264–271, New York, New York, USA, 2008. [17](#), [18](#), [22](#), [23](#)
- [71] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*, 2007. [17](#), [18](#), [20](#), [21](#)
- [72] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of International Conference on Machine Learning*, pages 807–814, 2007. [17](#), [18](#), [20](#)
- [73] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, 2010. [18](#), [28](#)
- [74] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3360–3367, 2010. [18](#), [28](#), [41](#)
- [75] Kai Yu, Tong Zhang, and Yihong Gong. Nonlinear learning using local coordinate coding. In *Advances in Neural Information Processing Systems*, volume 22, pages 2223–2231, 2009. [18](#)

- 
- [76] John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10:2899–2934, 2009. [18](#)
- [77] Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3(4-5):951–991, 2003. [18](#)
- [78] Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. A second-order perceptron algorithm. *SIAM Journal on Computing*, 34:640–668, 2005. [18](#)
- [79] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *Proceedings of Conference on Learning Theory*, pages 257–269, 2010. [18](#)
- [80] Francesco Orabona and Koby Crammer. New adaptive algorithms for online classification. In *Advances in Neural Information Processing Systems*, pages 1840–1848, 2010. [18](#), [23](#)
- [81] Vitor R. Carvalho and William W. Cohen. Single-pass online learning: Performance, voting schemes and online feature selection. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Datamining*, pages 548–553, 2006. [18](#)
- [82] S. Sathiya Keerthi, S. Sundararajan, Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. A sequential dual method for large scale multi-class linear svms. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Datamining*, pages 408–416, 2008. [18](#)
- [83] Ronan Collobert and Samy Bengio. Links between perceptrons, mlps and svms. In *Proceedings of International Conference on Machine Learning*, New York, New York, USA, 2004. [18](#), [21](#)
- [84] Florent Perronnin, Zeynep Akata, Zaid Harchaoui, and Cordelia Schmid. Towards good practice in large-scale learning for image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3482–3489, 2012. [18](#), [21](#), [22](#), [26](#), [27](#), [28](#)
- [85] Richard O. Duda and Peter E. Hart. *Pattern classification and scene analysis*. 1973. [22](#)
- [86] Jialei Wang, Peilin Zhao, and C.H. Steven Hoi. Exact soft confidence-weighted learning. In *Proceedings of International Conference on Machine Learning*, pages 121–128, 2012. [24](#), [25](#)

- 
- [87] Keith B. Hall, Scott Gilpin, and Gideon Mann. Mapreduce/bigtable for distributed optimization. In *Proceedings of Neural Information Processing Systems Workshop on Learning on Cores, Clusters, and Clouds*, 2010. 26
- [88] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992. 26
- [89] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, (2):91–110, 2004. 27, 28, 40, 47, 68, 73
- [90] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1):51–59, 1996. 28, 50
- [91] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001. 28, 40, 50, 68
- [92] Koen E. A. van de Sande, Theo Gevers, and Cees G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010. 28
- [93] Nicolas Loeff and Ali Farhadi. Scene discovery by matrix factorization. In *Proceedings of European Conference on Computer Vision*, pages 451–464, 2008. 39, 48
- [94] Bharath Hariharan, Lihi Zelnik-Manor, S. V. N. Vishwanathan, and Manik Varma. Large scale max-margin multi-label classification with priors. In *Proceedings of International Conference on Machine Learning*, pages 423–430, 2010. 39
- [95] Serhat Selcuk Bucak, Rong Jin, and Anil K. Jain. Multi-label learning with incomplete class assignments. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2801–2808, 2011. 39
- [96] Yashaswi Verma and C. V. Jawahar. Exploring svm for image annotation in presence of confusing labels. In *Proceedings of British Machine Vision Conference*, number c, pages 1–11, 2013. 39, 68, 69
- [97] Supheakmongkol Sarin, Michael Fahrmar, Matthias Wagner, and Wataru Kameyama. Holistic feature extraction for automatic image annotation. In *Proceedings of International Conference on Multimedia and Ubiquitous Engineering*, pages 59–66, 2011. 40, 47, 48, 50

- 
- [98] Gerard Salton and Chung S. Yang. On the specification of term values in automatic indexing. *Journal of documentation*, 29(4):351–372, 1973. [42](#)
- [99] David Grangier, Florent Monay, and Samy Bengio. A discriminative approach for the retrieval of images from text queries. In *Proceedings of European Conference on Machine Learning*, pages 162–173, 2006. [44](#)
- [100] Nobuyuki Otsu and Takio Kurita. A new scheme for practical flexible and intelligent vision systems. In *Proceedings of IAPR Workshop on Computer Vision Special Hardware and Industrial Applications*, pages 431–435, 1988. [50](#)
- [101] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 311–318, 2002. [52](#), [73](#)
- [102] George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of International Conference on Human Language Technology Research*, pages 138–145, Morristown, NJ, USA, 2002. [52](#), [73](#)
- [103] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 423–430, 2003. [54](#)
- [104] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *Proceedings of European Conference on Computer Vision*, pages 488–501, 2012. [60](#), [62](#), [63](#), [69](#)
- [105] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Large scale metric learning for distance-based image classification. Technical report, LEAR - INRIA and TVPA - XRCE, 2012. [60](#), [62](#), [63](#), [69](#)
- [106] Nicolas Usunier, David Buffoni, and Patrick Gallinari. Ranking with ordered weighted pairwise classification. In *Proceedings of International Conference on Machine Learning*, pages 1057–1064, 2009. [62](#)
- [107] Hideki Nakayama, Tatsuya Harada, and Yasuo Kuniyoshi. Canonical contextual distance for large-scale image annotation and retrieval. In *Proceedings of ACM Workshop on Large-Scale Multimedia Retrieval and Mining*, pages 3–10, New York, New York, USA, 2009. [63](#), [64](#), [69](#), [72](#), [91](#)

## REFERENCES

---

- [108] Francis R. Bach and Michael I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical report, Dept. Statist., Univ. California, Berkeley, 2005. [63](#), [69](#)
- [109] Jason Weston, Samy Bengio, and Nicolas Usunier. Large scale image annotation: Learning to rank with joint word-image embeddings. *Machine Learning*, 81:21–35, 2010. [67](#), [69](#)
- [110] Yoshitaka Ushiku, Tatsuya Harada, and Yasuo Kuniyoshi. Efficient image annotation for automatic sentence generation. In *Proceedings of ACM International Conference on Multimedia*, pages 549–558, 2012. [68](#), [69](#), [76](#)
- [111] S. L. Feng, R. Manmatha, and Victor P. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2004. [68](#), [69](#)
- [112] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Morristown, NJ, USA, 2003. [92](#)
- [113] David Chiang. A hierarchical phrase-based model for statistical machine translation. *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 263–270, 2005. [92](#)
- [114] Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 523–530, 2001. [92](#)
- [115] Michel Galley and C.D. Manning. Accurate non-hierarchical phrase-based translation. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, number June, pages 966–974, 2010. [92](#), [93](#)

# Publications

## Journal

1. 牛久祥孝, 原田達也, 國吉康夫, “画像・長文からの潜在空間獲得による画像間類似度の改善,” 情報処理学会論文誌, Vol. 52, No. 12, pp.3496-3503, 2011.

## Reviewed Conference

1. Yoshitaka Ushiku, Tatsuya Harada, and Yasuo Kuniyoshi, “Efficient Image Annotation for Automatic Sentence Generation,” Proceedings of ACM International Conference on Multimedia (ACMMM 2012), pp.549-558, Nara, Japan, Oct., 2012.
2. Yoshitaka Ushiku, Tatsuya Harada, and Yasuo Kuniyoshi, “Understanding Images with Natural Sentences,” Proceedings of ACM International Conference on Multimedia (ACMMM 2011), pp.679-682, Scottsdale, Arizona, Nov., 2011.
3. Yoshitaka Ushiku, Tatsuya Harada, and Yasuo Kuniyoshi, “Automatic Sentence Generation from Images,” Proceedings of ACM International Conference on Multimedia (ACMMM 2011), pp.1533-1536, Scottsdale, Arizona, Nov., 2011.
4. Tatsuya Harada, Yoshitaka Ushiku, and Yasuo Kuniyoshi, “Discriminative Spatial Pyramid,” Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011), pp.1617-1624, Colorado Springs, Colorado, June, 2011.
5. Yoshitaka Ushiku, Tatsuya Harada, and Yasuo Kuniyoshi, “Improving Image Similarity Measures for Image Browsing and Retrieval Through Latent Space Learning Between Images and Long Texts,” Proceedings of International Conference on Image Processing (ICIP 2010), pp.2365-2368, Hong Kong, China, Sep., 2010.

## Un-reviewed Conference

1. Yoshitaka Ushiku, Hiroshi Muraoka, Sho Inaba, Teppei Fujisawa, Koki Yasumoto, Naoyuki Gunji, Takayuki Higuchi, Yuko Hara, Tatsuya Harada, and Yasuo Kuniyoshi, “ISI at ImageCLEF 2012: Scalable System for Image Annotation,” CLEF Evaluation Labs and Workshop, Online Working Notes (CLEF 2012), Rome, Italy, Sep., 2012.

## Reviewed Domestic Conference

1. 牛久祥孝, 原田達也, 國吉康夫, “キーフレーズ推定と文法モデルによる画像説明文生成,” 画像の認識・理解シンポジウム (MIRU 2012), OS3-01, 福岡, 8月, 2012.
2. 稲葉翔, 村岡宏是, 山下裕也, 牛久祥孝, 金崎朝子, 原田達也, 國吉康夫, “学習時間に着目した効率的な大規模画像分類,” 画像の認識・理解シンポジウム (MIRU 2012), OS6-03, 福岡, 8月, 2012.
3. 牛久祥孝, 原田達也, 國吉康夫, “画像・文章間の類似度学習による画像説明文の自動生成,” 画像の認識・理解シンポジウム (MIRU 2011), pp.365-372, 金沢, 7月, 2011.
4. 牛久祥孝, 原田達也, 國吉康夫, “画像・長文からの潜在空間獲得による画像間類似度の改善,” 画像の認識・理解シンポジウム (MIRU 2010), pp.1153-1160, 旭川, 7月, 2010.

## Un-reviewed Domestic Conference

1. 金崎朝子, 稲葉 翔, 牛久祥孝, 山下裕也, 村岡宏是, 原田達也, 國吉康夫, “大規模画像データセットを用いたマルチクラス物体検出器の同時学習 - 物体毎に特化した負例クラスの導入 -, ” 電子情報通信学会技術研究報告 (PRMU), pp.105-112, 東京, 9月, 2012.
2. 牛久祥孝, 山下裕也, 井村純, 中山英樹, 原田達也, 國吉康夫, “複数画像特徴とクラスラベルの相関に着目した距離計量による大規模画像分類,” 電子情報通信学会技術研究報告 (PRMU), pp.1-6, 埼玉, 2月, 2011.
3. 牛久祥孝, 中山英樹, 原田達也, 國吉康夫, “Web 画像と文章の大域的特徴から得る潜在的意味に基づくデータ検索 Web 上での一般画像認識実現への新たなアプローチを目指して,” 電子情報通信学会技術研究報告 (PRMU), pp.45-50, 金沢, 11月, 2009.



## Others

1. **(Competition)** Naoyuki Gunji, Takayuki Higuchi, Koki Yasumoto, Hiroshi Muraoka, Yoshitaka Ushiku, Tatsuya Harada, and Yasuo Kuniyoshi, Got the first place in the fine-grained classification task and the second place in the classification task, ImageNet Large Scale Visual Recognition Challenge 2012 (in conjunction with ECCV 2012), Florence, Italy, Oct., 2012.
2. **(Invited talk)** Yoshitaka Ushiku, Tatsuya Harada, and Yasuo Kuniyoshi, “Efficient Image Annotation for Automatic Sentence Generation,” Greater Tokyo Area Multimedia/Vision Workshop, Tokyo, Aug., 2012.
3. **(Competition)** Tatsuya Harada, Asako Kanezaki, Yoshitaka Ushiku, Yuya Yamashita, Sho Inaba, Hiroshi Muraoka, and Yasuo Kuniyoshi, Got the third place in the classification task and the second place in the detection task, ImageNet Large Scale Visual Recognition Challenge 2011 (in conjunction with ICCV 2011), Barcelona, Spain, Nov., 2011.
4. **(Competition)** Tatsuya Harada, Hideki Nakayama, Yoshitaka Ushiku, Yuya Yamashita, Jun Imura, and Yasuo Kuniyoshi, Got the 3rd place, ImageNet Large Scale Visual Recognition Challenge 2010 (in conjunction with ECCV 2010), Crete, Greece, Sep., 2010.

## Awards

1. 2010年 PRMU 研究奨励賞
2. 2011年 MIRU 2011 インタラクティブセッション賞
3. 2011年 ACMMM 2011 Special Prize on the Best Application of a Theoretical Framework
4. 2012年 MIRU 2012 シングルトラックオーラルセッション採択 (ベストペーパー候補論文)