

Zero-shot Image Classification



The
University
Of
Sheffield.

Yang Long

Electronic and Electrical Engineering
University of Sheffield

This dissertation is submitted for the degree of
Doctor of Philosophy

I would like to dedicate this thesis to the intelligent.

Declaration

Parts of this thesis have been taken from published academic conference/journal papers. All of these papers were primarily written by me, Yang Long, during and as a result of my Ph.D. study. Involved papers are listed as follows.

Yang Long, and Ling Shao, Learning to Recognise Unseen Classes by A Few Similes, ACM Multimedia, 2017 (Chapter 7).

Yang Long, Li Liu, Ling Shao, Fumin Shen, Guiguang Ding, and Jungong Han, From Zero-shot Learning to Conventional Supervised Classification: Unseen Visual Data Synthesis, CVPR, 2017 (Chapter 5).

Yang Long and Ling Shao, Describing Unseen Classes by Exemplars: Zero-Shot Learning Using Grouped Simile Ensemble, WACV, 2017 (Chapter 6).

Yang Long, Liu Liu, and Ling Shao, Towards Fine-Grained Open Zero-Shot Learning: Inferring Unseen Visual Features from Attributes, WACV, 2017 (Chapter 4).

Yang Long, Fan Zhu, and Ling Shao, Recognising occluded multi-view actions using local nearest neighbour embedding, Computer Vision and Image Understanding 144 (2016): 36-45 (Chapter 2).

Yang Long, Liu Liu, and Ling Shao, Attribute Embedding with Visual-Semantic Ambiguity Removal for Zero-shot Learning, BMVC, 2016 (Chapter 3).

Yang Long
November 2017

Acknowledgements

Firstly, I would like to express my sincerest thanks to my supervisor, Prof. Ling Shao, who has provided me patient and inspiring direction towards the academic field. He is gentle and gives me so much freedom to explore the mysteries of computer vision field.

I would also thank my family, especially my mother, Yan Huang, who has provided every possible material and spiritual support for my study. Another significant person who I would like to thank to is my wife, JianQin Zhao. She has spent all her time and energy to take care of me and our daughter, KaiLi Long. My achievement cannot be apart from their continued love and encouragement.

I would also like to appreciate the help and inspirations from my dear colleges. Li Liu, Fan Zhu, Di Wu, as senior members, have provided many helpful supports. Also, I have collaborated with HaoFeng Zhang, Ziyun Cai, Shidong Wang, and Yao Tan with loads of joys. To BingZhang Hu, YuMing Shen, Yi Zhou, JiaoJiao Zhao, Lining Zhang, Heng Liu, and all of whom that could not be fully listed here, I will extend my sincere thanks to all of you. Besides, I would specially thank Daniel Organisciak for many proof readings and hope he could recover very soon.

Finally, I would thank all of my friends during my study in the UK. You let me know that life is not only about the niggles in front of our eyes, but also poem and field far away. A short list will be available on my website that will come very soon.

Abstract

Image classification is one of the essential tasks for the intelligent visual system. Conventional image classification techniques rely on a large number of labelled images for supervised learning, which requires expensive human annotations. Towards real intelligent systems, a more favourable way is to teach the machine how to make classification using prior knowledge like humans. For example, a palaeontologist could recognise an extinct species purely based on the textual descriptions. To this end, *Zero-Shot Image Classification (ZIC)* is proposed, which aims to make machines that can learn to classify unseen images like humans. The problem can be viewed from two different levels. Low-level technical issues are concerned by the general Zero-shot Learning (ZSL) problem which considers how to train a classifier on the unseen visual domain using prior knowledge. High-level issues incorporate how to design and organise visual knowledge representation to construct a systematic ontology that could be an ultimate knowledge base for machines to learn.

This thesis aims to provide a thorough study of the ZIC problem, regarding models, challenges, possible applications, *etc.* Besides, each main chapter demonstrates an innovative contribution that is creatively made during my study. The first is to solve the problem of *Visual-Semantic Ambiguity*. Namely, the same semantic concepts (*e.g.* attributes) can refer to a huge variety of visual features, and vice versa. Conventional ZSL methods usually adopt a one-way embedding that maps such high-variance visual features into the semantic space, which may lead to degraded performance. As a solution, a dual-graph regularised embedding algorithm named *Visual-Semantic Ambiguity Removal (VSAR)* is proposed, which can capture the intrinsic local structure of both visual and semantic spaces. In the intermediate embedding space, the structural difference is reconciled to remove the ambiguity.

The second contribution aims to circumvent costly visual data collection for conventional supervised classification using ZSL techniques. The key idea is to synthesise visual features from the semantic information, just like humans can imagine features of an unseen class from the semantic description of prior knowledge. Hereafter, new objects from unseen classes can be classified in a conventional supervised framework using the inferred visual

features. To overcome the correlation problem, we propose an intermediate Orthogonal Semantic-Visual Embedding (OSVE) space to remove the correlated redundancy. The proposed method achieves promising performance on fine-grained datasets.

In the third contribution, the graph constraint of VSAR is incorporated to synthesise improved visual features. The orthogonal embedding is reconsidered as an *Information Diffusion* problem. Through an orthogonal rotation, the synthesised visual features become more discriminative. On four benchmarks, the proposed method demonstrates the advantages of synthesised visual features, which significantly outperforms state-of-the-art results.

Since most of ZSL approaches highly rely on expensive attributes, the fourth contribution of this thesis explores a more feasible but more effective *Semantic Simile* model to describe unseen classes. From a group of similes, *e.g.* an unknown animal has the same parts of a wolf, and the colour looks like a bobcat, implicit attributes are discovered by a graph-cut algorithm. Comprehensive experimental results suggest the simile-based implicit attributes can significantly boost the performance.

To maximumly reduce the cost of building ontologies for ZIC, the final chapter introduces a novel scheme, using which ZIC can be achieved by only a few similes of each unseen class. No annotations of seen classes are needed. Such an approach finally sets ZIC attribute-free, which significantly improve the feasibility of ZIC. Unseen classes can be recognised using a conventional setting without expensive attribute ontology.

It can be concluded that the methods introduced in this thesis provide fundamental components of a zero-shot image classification system. The thesis also points out four core directions for future ZIC research.

Key Words: Zero-shot Learning, Attributes, Similes, Feature Embedding, Graph Regularisation, Graph Cut, Deep Neural Network, Ontological Engineering, Image Classification, Machine Learning, Artificial Intelligence, Computer Vision.

Table of contents

Table of contents	xi
List of figures	xv
List of tables	xix
1 Introduction	1
1.1 Motivations	2
1.2 Context	5
1.3 Contributions and Thesis Outline	8
2 Background	11
2.1 Image Classification	11
2.2 Attribute Learning and Ontology	13
2.3 Zero-shot Learning	15
2.3.1 Learning Framework	15
2.3.2 Knowledge Representation	16
2.3.3 Assumptions and Scenarios	17
3 Visual-Semantic Ambiguity Removal	21
3.1 Introduction	21
3.2 Visual-Semantic Ambiguity Removal	23
3.2.1 Latent Attribute Embedding	24
3.2.2 Dual-graph Regularisation	25
3.2.3 Optimisation Strategy	26
3.2.4 Zero-shot Image Classification	28
3.3 Experiments	28
3.3.1 Comparison with the state-of-the-arts	28
3.3.2 Algorithm analysis	29

3.3.3	Visual-semantic ambiguity removal	31
3.4	Conclusion and future work	31
4	Towards Open Zero-shot Learning	33
4.1	Introduction	33
4.2	Related work	36
4.3	Visual Feature Inference	37
4.3.1	Problem setup	37
4.3.2	Orthogonal Semantic-Visual Embedding	38
4.3.3	Optimisation Strategy	39
4.3.4	Zero-shot Recognition	42
4.4	Experiments	42
4.4.1	Setup	42
4.4.2	Benchmark Comparison	43
4.4.3	Fine-grained Open Zero-shot Learning	47
4.5	Conclusion	48
5	Zero-shot Data Synthesis	51
5.1	Introduction	51
5.2	Related work	54
5.3	Approach	57
5.3.1	Preliminaries	57
5.3.2	Unseen Visual Data Synthesis	58
5.3.3	Optimisation Strategy	61
5.3.4	Zero-shot Recognition	65
5.4	Experiments	65
5.4.1	Setup	65
5.4.2	Comparison with the State-of-the-art methods	67
5.4.3	Detailed Evaluations	68
5.4.4	Further Discussions	72
5.5	Conclusion	74
6	Beyond Explicit Attributes	75
6.1	Introduction	75
6.2	Related Work	77
6.3	Approach	78
6.3.1	Simile Annotation	78

6.3.2	Preliminary	79
6.3.3	Implicit Attributes Discovery	79
6.3.4	Grouped Simile Ensemble	81
6.3.5	Zero-shot Classification	84
6.4	Experiments and Results	84
6.4.1	Implicit Attribute Discovery	85
6.4.2	Compared to State-of-the-art methods	86
6.4.3	Detailed Analysis	87
6.5	Conclusion	90
7	Towards Affordable Ontology	93
7.1	Introduction	93
7.2	Related Work	95
7.3	Approach	97
7.3.1	Preliminary	98
7.3.2	Match Kernel Embedding	98
7.3.3	Simile Quantification	99
7.3.4	Inferring Complete Class-level Prototype	100
7.3.5	Zero-shot Classification	102
7.4	Experiments	103
7.4.1	Simile Annotations	105
7.4.2	Compared to Published Results	106
7.4.3	Detailed Analysis	107
7.5	Conclusions	109
8	Conclusion and Future Work	111
8.1	Learning and Data Synthesis	111
8.2	Simile Ontology	112
8.3	Future Research Interests	113
	References	115
	Appendix A Glossary	127
	Appendix B Notation	129

List of figures

1.1	Computer vision involves various technologies at different levels.	1
1.2	Illustration of the ImageNet that is organised based on Wordnet.	2
1.3	Illustration of human-like zero-shot learning.	3
1.4	Left: Raw RGB Image; right: High-level class labels of car parts.	6
3.1	An intuitive illustration of VSAR (best viewed in colour). Visual Ambiguity (in blue oval): the image of a carriage is taken with a building background. It cannot recover the semantic distance (blue question mark) to the building category. Semantic Ambiguity (in red oval): the cup printed with a wolf and the cup-like building share the same semantic expression which can lead to a large visual variance (the red question mark). After embedding to the latent attribute space using VSAR, such ambiguity is mitigated.	22
3.2	Confusion matrix of ZSL performance on aPY (left) and AwA (right). . . .	29
3.3	Evaluating each term of the loss function in Eq. 3.3 (left) and the performance curve respects to the dimension K of the latent attribute space (right).	30
3.4	Examples of successful semantic ambiguity removal on aPY (left) and the visual ambiguity removal on AwA (right).	31
4.1	Comparison between our procedure (Red) and the conventional ZSL framework (Blue). Fine-grained classes are often compact and non-describable in the attribute space. Our OSVE can discover tiny visual differences between different instances under the same attribute so as to infer discriminative visual features for unseen classes from fine-grained open candidates.	34
4.2	An example of the convergence situations shows the loss with respect to the number of iterations. Term 1 and 2 corresponds to the reconstruction errors to visual and semantic spaces. Term 3 accounts how orthogonal is the embedding space.	41

4.3	A. overall accuracies of baseline methods by substituting key components of the proposed framework. B. ROC curves of our method on the two datasets. For clarity, only 10 of the 50 unseen classes on CUB are shown.	44
4.4	Comparing the data distribution between real (A) and inferred (B) visual features of unseen classes. Note that t-SNE can result in slight distortion and colour differences.	44
4.5	Performance curve with respect to the dimension K of the intermediate embedding space.	45
4.6	Open ZSL 2: test by increasing number of unseen classes using different size of training sets.	48
4.7	Top-5 nearest neighbours of the query image under conventional and open ZSL. Correct and incorrect matches are shown in green and red respectively. Corresponding seen/unseen splits are shown on the right.	49
5.1	Given a conceptual description, human can imagine the outline of the scene by combining previous seen visual elements.	52
5.2	Comparison of supervised and zero-shot classifications and existing ZSL frameworks. (A) a typical supervised classification: the training samples and labels are in pairs; (B) a zero-shot learning problem: without training samples, the classes C and D cannot be predicted; (C) Direct-Attribute Prediction model uses attributes as intermediate clues to associate visual features to class labels; (D) label-embedding: the attributes are concatenated as a semantic embedding; (E) we use semantic embedding to synthesise unseen visual data.	54
5.3	An illustration of our framework of unseen data synthesis. Unseen classes are represented by semantic attributes as inputs. We train a model that maps the semantic space to the visual data space to synthesise training data for these unseen classes. The crosses in the visual spaces denote test feature points.	56
5.4	Objective function convergence on the AwA dataset.	62
5.5	Some random image and attribute examples of the 4 datasets.	64
5.6	Normalised variances of the synthesised data <i>w.r.t.</i> dimensions. Variance of each dimension is sorted in descending order. We make a comparison between the synthesised data variances ‘with’ (green) and ‘without’ (red) diffusion regularisation. The variances of real data (blue) are computed from real unseen data as references.	68

5.7	T-SNE of the real and synthesised visual features of unseen classes: (A) real visual features; (B) synthesised visual features; (C) Since t-SNE of different data is not aligned, we also show the distribution of mixed real and synthesised visual features.	69
5.8	The performance with respect to the Graph regularisation and Diffusion regularisation. The results are under the scenario of CA and using NN classifier.	70
5.9	Success and Failure cases of nearest neighbour matching. The query visual feature is synthesised from its attribute description. We find top-5 nearest neighbours of the query feature from the real instances. It is a match if the nearest instance and the test image have the same label.	71
6.1	A new class can be described by similes of seen classes without extra attribute concepts involved. We use semantic grouping to make the similes more discriminative. Similes are more natural to describe complex concepts, <i>e.g. behaviour</i> or <i>domestic</i>	76
6.2	An example of simile annotation process: whose <i>colour</i> is similar to <i>antelope</i> . The annotator is asked to choose a number of most similar exemplars. We achieve averaged similarities among all of the annotator's associations.	80
6.3	Implicit attribute discovery. Under each simile group, the associated exemplars of each class satisfy a k -nn graph (left). Red vertices indicate <i>unseen</i> classes. Our algorithm can cut the weakest edges and cluster the classes with similar implicit attributes (right).	81
6.4	Visualisation of eigenvalues. We demonstrate the example from the simile group of <i>activity</i> in the AWA dataset. The k -NN graph of similes has two disconnected subsets (one zero eigenvalue). However, we could find roughly four more layers, which indicates that the optimal value for m is 5. .	82
6.5	Examples of images annotated by similes under different groups in AWA (upper) and aPY (lower).	85
6.6	Partial results of graph-cut class-clustering. Images with in the same colour of frames are from the same cluster.	88
6.7	Implicit Attribute Prediction Precision on AWA and aPY. Results are shown by different simile groups.	89
7.1	Given some similes as clues, humans can easily make classification for the unseen classes.	94
7.2	Comparison between ZSL frameworks. The gray bars between attributes and class labels denote the human-defined class-attribute matrices.	95

7.3	An overview of training stages. Each unseen class gains a class-level prototype by inferring from the paired similes.	96
7.4	(A) Raw visual feature distribution of the 10 unseen classes in AwA. (B) After MKE, non-discriminative points (red circle in (A)) are separated. Before the test, we aim to infer the centroid of each class as the prototype.	99
7.5	Illustration of the idea to infer complete MKE representation (numbers are only for demonstration purposes).	101
7.6	Illustration of test phase: a test image can be converted into the MKE space and compared to the inferred MKE prototypes of unseen classes to make a prediction.	103
7.7	Examples of top-5 similes from human annotations (upper) versus top-5 classes with largest MKE values (lower).	104
7.8	Simile error tolerance: ZSL performance <i>w.r.t.</i> different combinations (upper). Amount of supervision: ZSL performance <i>w.r.t.</i> different numbers of similes (Lower). Best viewed in colours.	106
8.1	Zero-shot Learning and other learning framework.	114

List of tables

3.1	Compare with the published state-of-the-art methods.	29
4.1	Key statistics of CUB and SUN datasets.	42
4.2	Comparison to state-of-the-art methods for both datasets. Results are overall accuracies in %.	43
4.3	Results (in %) of Open ZSL 1: add extra seen classes as candidates or add instances from seen classes for testing.	47
5.1	Key statistics of the four datasets.	64
5.2	Comparison with state-of-the-art methods.	66
5.3	Detailed analysis of key aspects of the proposed method.	66
5.4	Comparison with published results on GZSL.	73
5.5	Comparison with published results on the ImageNet Dataset.	74
6.1	Statistics of simile annotation on AwA and aPY datasets.	79
6.2	Compared to the state-of-the-arts using deep features.	86
6.3	Compared to baseline methods using low-level features.	87
6.4	Evaluating GSE on different settings.	90
7.1	Dataset statistics	103
7.2	Simile annotation evaluation using hit rate.	103
7.3	Main comparison with the state-of-the-art results.	105
7.4	Compared to supervised results (%) on Caltech 101.	105
7.5	Averaged inference time (s) for each unseen class	106
7.6	Compared to state-of-the-art methods on ZSL and GZSL scenarios. Performances (%) under different Auxiliary Information (AI) are compared, in terms of MKE and Attributes.	107
7.7	Comparing the performance upper bounds (%) of ZSL and GZSL using MKE and raw features.	109

Chapter 1

Introduction

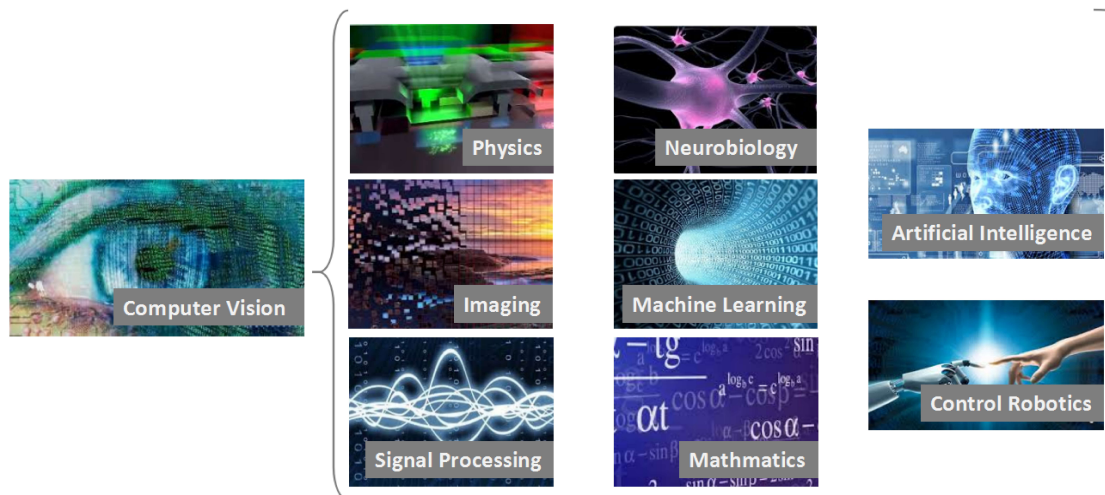


Fig. 1.1 Computer vision involves various technologies at different levels.

Vision is an important perceptive approach. Computer vision community has dedicated to improve the visual system of machines and meanwhile to make it more interpretable. As shown in Fig. 1.1, computer vision involves various technologies at different levels. Natural light through the cameras is firstly converted into digital signals. Afterwards, visual features are extracted for various applications, such as Detection [38], Recognition [138], Segmentation [120], Content Based Information Retrieval (CBIR) [123], Tracking [56], Simultaneous Localisation and Mapping (SLAM) [30]. These techniques are important premises for realistic artificial intelligence and robotic controls.

We hope the machine vision is not only a sensor for automatic control, but can also make sense of the real world and achieve concept-level understanding like humans. To this



Fig. 1.2 Illustration of the ImageNet that is organised based on Wordnet.

end, an essential technique for computer vision is image classification, which involves making high-level predictions of the given image, such as the category, location, attribute, and relationship. Nowadays, the most popular approach for image classification is supervised learning. Namely, we provide many positive and negative instances that are called *training examples*. Each training example is tagged with one or multiple labels denoting the corresponding concepts. The training process is to discover the features and the rules and how these pieces of information are related to the target concepts. For example, we can train a classifier to recognise dogs and cats. The machine first extracts the discriminative features from the training examples of dogs and cats and then discovers how these features are different to each other. At the test time, dogs and cats can be recognised by directly applying the trained classifier to the input image. Using large-scale datasets, such as the ImageNet [32], machines can recognise roughly 24K classes that cover most of the general categories in our human life.

1.1 Motivations

Despite the promising progress, existing supervised learning framework is not easy to be applied for many real-life problems. Firstly, in our real life, new concepts and labels are continually generated every day. Existing supervised training highly relies on sufficient well-labelled training examples for each new class, which requires very costly human annotations. Although an alternative approach is to utilise the online search engine, the tags are often based on the semantic content and may not adequately describe the visual concepts. Moreover, the tags can be very noisy due to different purposes of internet users. Therefore, acquiring sufficient high-quality training examples for ever-increasing new classes is infeasible. Secondly, a basic class can have many finer-grained subclasses. For example, the

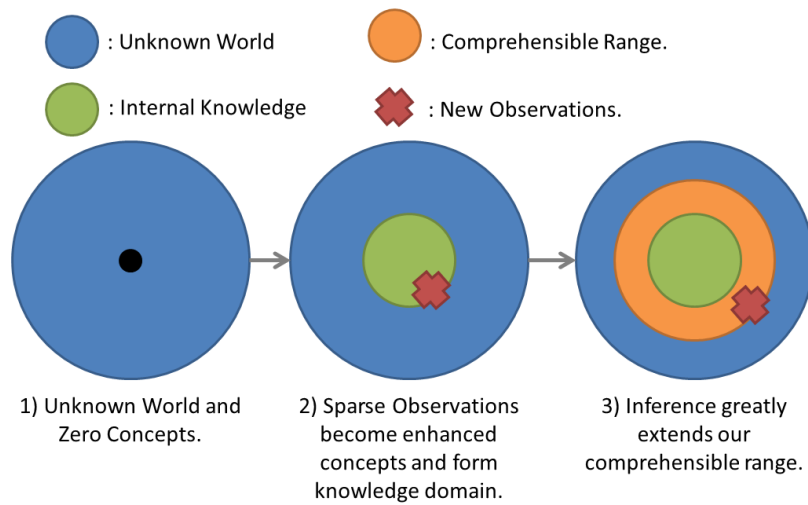


Fig. 1.3 Illustration of human-like zero-shot learning.

ImageNet mentioned above has 30 mushroom synsets with roughly 1,000 images. However, such a level of granularity is still far behind nature which has approximately 14,000 species of mushrooms. The increased number of required training examples from coarse to fine classes will be huge. Thirdly, sometimes, it is impossible to obtain an even one-shot image for training. For instance, one would recognise an unseen criminal purely based on the description from witnesses. For the reasons mentioned above, the problem of *‘training example hunger’* has severely restricted the possible applications of supervised learning.

To this end, Zero-shot Learning (ZSL) is proposed, which aims to predict the labels of unseen images without the burden of collecting training examples. From a higher view, as shown in Fig. 1.3, our human life can be regarded as a ZSL process. In the beginning, everyone is born with ‘Zero’. With our growth, we gradually accumulate our observations with paired sparse concepts that come from supervision. For example, we could learn what is an ‘apple’ by seeing it while repetitively hear the pronunciation from our parents. These occurrences may form some early perceptions that are denoted by the green circle. Afterwards, we discover the potential associations between concepts. For instance, we found apple is ‘red’ that also applies to ‘strawberry’. Such connections result in the generation of knowledge that allows us to infer the unobserved world through prior human knowledge. For example, when we learn an unseen fruit that is red, smaller than strawberry and looks like an apple, we could imagine how does a ‘cherry’ look like. Such inference significantly extends our comprehensible range of the unobserved world. The fast inheritance of human knowledge also makes new generations can more efficiently explore the unknown world. Inspired by this fact, we could design a similar ZSL system for image classification and make it possible to utilise the vast amount of our prior human knowledge to mitigate the short-

age and difficulty of acquiring training images. Furthermore, the ZSL visual system can be more interpretable and flexible that provides possibilities for concept-level human-machine interaction.

To achieve practical intelligent Zero-shot Image Classification systems, we consider four fundamental technologies, which are summarised as follows.

- **Visual Feature Extraction** To deal with large-scaled ever-growing new classes, we need to extract high-capability features that can differentiate one image from all of the rest. Meanwhile, the features are better to capture the semantic structure of category labels so that the human-machine gap can be minimal. For example, an ‘apple’ should be close to an ‘orange’ rather than a ‘car’ in the visual feature space.
- **Interpretable Knowledge Representation** ZIC is about how to interpret unseen visual features. Although we can have knowledge about the unseen classes, it would be a challenge to convert the knowledge into machine-understandable views. The representation should be dense, informative, and visual-sensitive so that the machine can maximumly infer the visual features that are close to our human perception.
- **Ontological Engineering and Dataset** This is a higher-level requirement of knowledge representation. We need to not only teach machines how we human perceive the world, but also design a proper structure that can systematically organise our human knowledge.
- **Machine Learning** During the past decades, it has been widely acknowledged that rule-based intelligent systems might not be competent to deal with the gigantic information in the complex environment. Therefore, so far, the learning-based scheme is the most promising methodology towards the real intelligent system. Particularly, a challenge is how to achieve incremental training and let the machine can grow with the progress of the human society and the expanding requirement of automation.

Motivated by the above four concerns, this thesis dedicates to discuss the past, present and future of Zero-shot Image Classification (ZIC). Excepting reviewing the current state of the above four aspects, there are four innovative contributions of the thesis, which focus on addressing crucial problems, such as mathematical modellings, methodologies, and other detailed engineering challenges of ZIC.

1.2 Context

A realistic image classification system involves many complex processes. We first need to detect the interest object and localise it in the image. We then crop the image or make segmentations to distinguish the target from the background. However, it is worth noting that for research purposes, the focus of image classification only focuses on predicting the label of the test image, either cropped or non-cropped. Normally, preprocessing such as detection or image segmentation is out of the scope of image classification.

This section briefly introduces the development of related techniques of image classification. However, more detailed background knowledge that is frequently used in the main chapters are not included here but will be discussed in the next chapter.

1. Intelligent Visual System As shown in Fig. 1.4, a digital image consists of discrete numbers ranging from 0 to 255 indicating the pixel intensities in each of the RGB channels. It can be seen that the resulted matrix is not as straightforward as what we humans can see in the image. Therefore, a core technique is how to extract invariant features from noisy input images. During the last decade, a large number of heuristic low-level feature descriptors have been proposed [29, 35, 88, 132]. These features attempt to simulate the mechanism of human vision so that the data can be presented closer to human visual perceptions. Meanwhile, machine learning approaches aim at computing the statistical features between low-level visual signal and a human-defined task space. Some of the learning methods focus on discovering the intrinsic data structure, such as scattering [126], or clustering [130], while some approaches directly interpret data with task labels, e.g., classification [26]. The most impressive progress comes from the technique of Deep Neural Network [54]. Since the first Caffe model [62] becomes open-sourced, an explosively increasing number of pre-trained deep models using large-scaled datasets app. These technologies have significantly improved the performance of image classification.

On the other hand, the learning-based visual feature extraction approaches highly rely on a large number of well-labelled training data. Although the low-cost digital devices have made big data available through the internet, the labels are very noisy and not competent for many practical tasks. For example, the labels may refer to the usages rather the visual appearances; an image can be interpreted by multiple labels or from different views *etc.*

To this end, Zero-shot Image Classification (ZIC) is proposed in 2009 by Lampert *et.al* [72]. Despite some Earlier work [74], [72] proposes a standard framework that inspires most of later approaches. The key idea is to learn a knowledge model on the well-labelled source domain and make the model generalise and explore unlabelled or unseen classes. The ultimate challenge is to extract universal visual features that have the capability to represent all of our human knowledge. However, the structure of visual and semantic features are

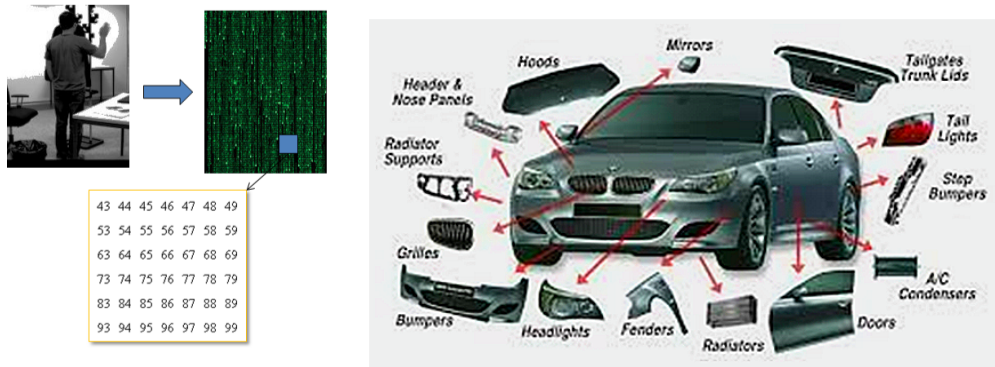


Fig. 1.4 Left: Raw RGB Image; right: High-level class labels of car parts.

very heterogeneous. Such transfer is harder than conventional domain adaptation problem (*e.g.* transferring from one view of a camera to another). Also, new unseen classes are ever-growing, and human knowledge is huge comparing to acquired visual training data. Therefore, existing ZIC methods only focus on an unrealistic assumption of a small close set of unseen classes. For example, we have known the test image must come from the class of either dogs or cats and do not need to consider the other classes. So far, how to make generalised ZIC on large-scaled open dataset remains difficult.

Knowledge Representation In order to recognise unseen classes, one of the fundamental problems is how to explore the potential knowledge using the obtained training source maximumly. Compared to conventional supervised approaches that only associate visual features with labels, knowledge representation aims to capture the relationship between discrete concepts, *e.g.* organise them into a graph. In this way, the learnt model can generalise to new unseen concepts through inference.

However, in the context of ZIC the main difficulty is to describe the visual relationship between concepts, *i.e.* how do images of a class look like the others. Despite the existence of many advanced language models, such as Word2Vec [97] or textual models [36], the performance is still not competent for realistic applications. The fundamental problem is that most of these models are designed for capturing the semantic relationships rather than for visions. For example, the most popular Word2Vec model is trained using Wikipedia or news. Although these source domains are huge, they are not particular descriptions for visual features. The resultant knowledge model will be highly heterogeneous compared to expected visual relationships. Another difficulty is the limitation of our language power, especially for those intangible or emotional feelings from our visual perceptions. A widely-used existing representations for ZIC are visual attributes [37, 58, 72, 155], which can be a colour ‘red’, a description for parts ‘has hands’, or even some indirect information, such as ‘living in the water’. State-of-the-art ZIC results [142] are typically based on multi-

modal frameworks that jointly consider the visual and attribute domains. The resultant representations can effectively associate visual features with semantic visual attributes and outperform methods using other knowledge models.

Ontological Engineering Despite the advantages of attribute-based ZIC methods, their implementation is highly restricted by existing ontological engineering. The expected attributes should be able to represent the required knowledge of all test classes. However, for large-scaled image datasets, such as ImageNet, it is infeasible to associate each class with sufficient attributes. To this end, there are two main difficulties: design and annotation.

To design attributes for ZIC is ambiguous work. For example, it is easy to differentiate a zebra from normal horses by the textures. However, textures may be not the best attributes when distinguishing horses from bulls. The fundamental problem is that we do not have explicit knowledge that can direct us what attributes are adequate for the classification tasks. Therefore, existing datasets [72, 131] often requires expert knowledge on specific domains. Furthermore, the expert knowledge may be not directly sensitive to visual features. For example, the first ZSL approach DAP [72] is proposed on the AwA dataset. 50 wild animal classes are associated with 85 attributes that are designed by zoologists. One of the attribute groups is ‘character’, such as ‘fierce’, ‘smart’ and ‘domestic’. Classes that share these attributes may have very different visual appearances, which leads to severely high variances. Last but not least, some implicit attributes that can hardly be expressed, which breaks the preliminary of existing attribute-based methods.

Annotating attribute training samples is also expensive. Compared to the conventional supervised classification that requires one label for each category, attribute-based ZIC needs a list of labels indicating the presence or absence of the corresponding attributes. Such requirement worsens the original purpose of mitigating annotation expense of large-scaled image classification.

Considering the above shortages of attributes, more advanced ontological engineering is vital for a new generation of ZIC techniques. Some recent work such as Visual Genome [69], and Activity Net [18], has incorporated attributes into a concept net that consists of verbs, nouns, adjectives, etc. These datasets are proposed to address more complex visual problems. However, the total number of involved concepts is still much smaller than the whole ImageNet dataset with 21K categories. Associations between such a large number of classes are out of the limit for existing ontological engineering.

Machine Learning One of the most popular frameworks is supervised learning. All of the training data is labelled (which is known as supervised). The machine is trained to learn a mapping to satisfy our particular criterion which can be one or multiple class labels. The performance can be measured by the classification accuracy. In semi-supervised learning,

only part of the training data is labelled. Unlabelled data is complementary to the trained model as long as it can improve the performance. For unsupervised learning, however, the performance can be hardly measured by classification accuracy because all of the training data is unlabelled. Therefore, additional rules need to be specified as implicit criteria. Evaluation is then based on whether the learnt model can indirectly make contributions to the classification accuracy.

ZIC is indeed a supervised problem since our goal is to predict the labels of images. For example, we first teach the machine what is food by specifying some categories with training examples. But there is no meat during the training. However, from knowledge view, ZIC is semi-supervised, *i.e.* seen classes are supervised by knowledge representation while unseen classes do not have. If we consider unlabelled visual features from unseen classes are available, the transductive ZSL utilises can be viewed as a special case of semi-supervised learning. Furthermore, unlabelled data of other food could be complementary information to improve the knowledge domain. ZIC assumes that the acquired prior knowledge can be shared to unseen classes, *e.g.* how does meat relate to vegetables. Such a scenario can be viewed as a special case of transfer learning. Conventionally, the transfer is only considered to happen within visual domains, such as view changing. However, ZIC requires a more challenging transfer that is from knowledge to visual domain. It is worth noting that the machine learning scenarios mentioned above are not exclusive to each other. Although the original purposes are different, these methods can be complementary to each other to improve the performance together.

1.3 Contributions and Thesis Outline

The rest of the thesis is organised as follows. In the next chapter, a comprehensive literature review will be provided for frequently-used techniques. Also, a brief survey of the development of zero-shot learning will be given, from which we can see there are two streams of contributions that are made by this thesis.

The first stream considers the learning problem that is summarised as follows:

- In chapter 3, we investigate the problem of visual-semantic ambiguity. Namely, some instances that look very similar may have distinctive semantic meanings, whereas large varieties of categories may share the same attribute. Direct mapping from visual to semantic space results in degraded performance. Therefore, there should exist a latent space that can mitigate the heterogeneousness. We model such a problem as a graph, using which we can maximumly preserve the relationships in both visual and semantic spaces.

- In chapter 4, we consider a more challenging fine-grained ZSL problem. Instead of classifying general categories, *e.g.* cats or dogs, fine-grained classification focuses on specific subcategories that are close to each other, *e.g.* bistro or restaurant. Since these classes have very similar attributes, making classification on the conventional semantic space is very difficult. We propose a visual-semantic orthogonal embedding algorithm that inversely projects semantic attributes into the visual feature space so that small discriminative features can be captured. Such a scheme also benefits a more realistic scenario called Open ZSL, *i.e.* the number of unseen classes is much larger than that of conventional ZSL. Also, we upgrade the visual features by deep learning features and achieve the state-of-the-art performance on both conventional and open ZSL scenarios.
- Finally in chapter 5, we combine the above graph and orthogonal models to synthesise unseen visual features. Such a technique ensures that we can achieve training examples for unseen classes as long as we have the prior knowledge. The ZIC problem is addressed by converting it back to conventional supervised classification. The proposed method steadily outperforms the state-of-the-art methods on all of the four benchmarks.

Besides learning problems, we seek for better knowledge representation and ontology for ZIC rather than visual attributes:

- In chapter 6, we propose an alternative to visual attributes for ZIC which is known as Similes, *e.g.* panda looks like a bear with similar colour as a killer whale. Unseen classes can be associated with seen classes using similes without involving extra attributes bothering the attribute design. Through a novel graph-cut scheme, implicit attributes that can hardly be expressed can also be described for ZIC. The resultant grouped similes consist of an ensemble of classifiers, which can significantly improve the state-of-the-art results.
- In chapter 7, the annotation cost of simile is further reduced to an affordable degree. High ZIC performance can be achieved by a powerful inference scheme using Gaussian Process Regression. Each unseen class is annotated by only a few sparse similes, and there is no need to provide any similes for seen classes at all. The proposed method enables us to implement ZIC tasks on any conventional supervised scenarios without the burden of attribute ontology. Extensive experiments manifest that similes can better represent complex visual associations and lead to remarkable performance gain over attribute-based ZIC methods.

In the final chapter seven, the highlights of the thesis are concluded. Furthermore, the conclusion suggests how we could apply the proposed methods in this thesis to a wider range of tasks to make a real impact on the society. Finally, potential future work is advised for whom may continue work on the field of ZIC.

Chapter 2

Background

We first review basic techniques for ZIC, in such as feature extraction and classification. Furthermore, frequently-used ontologies for ZIC, such as visual attributes, word embedding, and knowledge graph, are introduced. Finally, we discuss some variations of ZIC scenarios and assumptions.

2.1 Image Classification

Digital visual data are in the form of pixels, such as images and videos. However, such representation is not straightforward to reflect the contents. As shown in Fig. 1.4, the human in the image is represented by a huge matrix of numbers. To extract informative features, existing methods can be divided into two branches: heuristic visual features and automatic learnt features. After feature extraction, classification rules are applied to predict the labels. In the following, frequently-used feature representation and classification methods are both reviewed.

Heuristic Features Pixel-wise visual data is sensitive to tiny variations, such as motion, illumination, viewpoint, etc. During the past decade, local features are more popular in the computer vision community due to the local invariance. A representative method is known as SIFT in [88]. Its success can be credited by heuristically implementing the mechanism of human vision detection. Key points are detected through Difference-of-Gaussian (DoG) process and localised by Hessian Matrix, which is robust to rotation and scale variations. In general, methods like SIFT attempt to extract local features at the detected regions of interests (ROIs). SURF [12] achieves invariance that similar to SIFT but can further speed up by processing the Hessian Matrix before the pyramid of (DoG). Besides keypoint detection, HOG [29] is proposed to detect objects. It quantises the angles of the pixel gradients within a certain region into bins of a histogram. The target object can be detected by sliding

the detectors and match the histogram of each window to the reference histogram. However, local descriptors are prone to neglect the holistic relationships, such as their relative spatial positions. Pyramid representations, such as PHOG, incorporate local features by embedding gradually scaled-up cells of local descriptors into a holistic one [16]. Such a scheme is recently extended to model deformable object parts [38], which has achieved state-of-the-art performance on detection. In contrast to gradient-based descriptors, LBP [2] directly encode local pixels into binary codes, which are then statistically quantised by a histogram. Together with the spatial pyramid, LBP has shown powerful descriptiveness for face recognition [2]. Other than spatial information, Bag-of-words [28] (BoW) methods achieve holistic representations by regarding each local feature as a visual word. Each word is often obtained by clustering nearby training examples, such as K-means [89]. The clustered examples are assumed as the same word. Most of the video local descriptors, such as Cuboid [35], HOG3D [65], Dense Trajectory [132] and LBP-TOP [93], can be smoothly introduced to the BoW framework and achieve state-of-the-art action recognition results. In defence, holistic features also achieve progress on some other tasks. For example, GIST [102] introduces Gabor filters on different scales and orientations, and the feature maps are average pooled on the spatial cells, which has achieved promising performance on scene classification tasks. In addition, some inherently holistic features, such as colours and textures, are better encoded by holistic methods [92].

The above methods have been successfully applied to conventional supervised single-label image classification. However, for ZIC, we need to predict a more complex knowledge representation that requires richer visual information. For instance, the attribute "wild" could involve various visual features, like the scene, colour or local parts for wild animals. Typically, a single heuristic feature is believed not to be competent to represent a vast number of different attributes. The typical solution is to combine various heuristic visual features into a rich visual representation of multiple attribute spaces using learning approaches.

Feature Learning Feature learning methods can be supervised or unsupervised. Generally, they learn an objective function that can project the input data to a human-defined space to satisfy particular criteria. The learnt features are more discriminative compared to heuristic methods since the models are more related to samples rather than prior rules. For example, PCA has been successfully applied to face recognition for almost thirty years, which is a well known unsupervised approach dubbed Eigenface [126]. Given a large number of training examples, the principle components of each face can be usually concentrated into its first several eigenvectors that preserve most of the energy. After PCA embedding, the feature dimension drops remarkably, and the embedded points of different faces are scattered. In other words, the embedded features are informative and discriminative. However, due to

there is no supervision at all, PCA is not task sensitive. It can only scatter the points based on the intrinsic data structure. LDA [13], in contrast, is a supervised feature embedding algorithm. It can crush points with the same label into compact space while widening the distances between points from different classes. After embedding, the feature dimension is reduced to the number of classes minus one. However, the original data structure is completely distorted. Generally, such spectrum embedding methods are still heuristic. The learning objective could be some task-independent criteria, such the variance, correlation. Recently, deep learning features are getting dominant [54, 75, 129] due to its power to unifying feature extraction with image classification. CNN [75] can effectively extract local variations. Different scales from coarse to fine are also considered through pooling scheme. After convolutional layers, local patches are concatenated into a global representation. A deep fully-connected structure like that in DBN [54] is followed to extract higher-level information towards class labels. Another popular unsupervised deep framework is known as SAE [129]. Without supervision, the learning objective is first to compress the input image and then reconstruct it, through which discriminative features are captured. The key advantage of the deep feature is that it is naturally designed for large-scale data. By pre-training the deep model on large-scaled datasets, such as the ImageNet, the extracted features can be directly generalised to other tasks and results in state-of-the-art performance.

Classification The most straight forward classifier could be the NN classifier [27]. The query point is matched to the nearest neighbour in the training set. The underlying principle is called Kernel Density Estimation (KDE), where each training example stands for a radius estimation of the probability distribution. NBNN [15] extends such a simple scheme with image-to-class distance, which achieves state-of-the-art results. While the size of the training set is small, SVM [26] is powerful due to the maximum margin scheme. In contrast to these conventional classifiers, Deep Neural Networks demonstrate the power of end-to-end models. Input visual data is classified by forward propagation through multiple layers of non-linear projections. Besides, ensemble learning attempts to combine multiple classifiers into an ensemble model [34] to make the classifiers complementary to each other.

2.2 Attribute Learning and Ontology

Variations of Attributes Learning visual attributes extends the standard training labels to abstract semantic concepts [39]. Such progress on comprehension richly broadens the range of visual tasks [37, 71, 72, 115, 117]. One of the fundamental questions is how to obtain these attributes? A commonly agreed method is pre-defined category-attribute prediction

matrix by specialists [72]. However, there are several drawbacks. Firstly, the category-level attributes are not sensitive to instance variances. For example, we can annotate the category of pandas as 'eating bamboo'. However, there is no guarantee that all of the pandas in the test images are eating bamboo. Secondly, some realistic applications may not have systematic domain knowledge. Defining category-attribute association becomes ambiguous. As an alternative solution, in [37], each image is annotated with a unique attribute code. Moreover, both of [37, 72] involve exhaustive human annotations that are either achieved from volunteers or through the Amazon Mechanical Turk. To reduce the cost, automatic attributes mining through website text information fills in this gap [14, 115]. However, pre-defining the attribute list is still required. Therefore, attributes should be used with respect. Unfortunately, only a few existing research consider the specific characteristic of attributes where as most of the rest just ignore what types of attributes are they using. Particularly, phrases and adjectives are commonly used as attributes, *e.g.* 'has hands', and 'red'. Attributes can also be relative [104], *e.g.* 'He is taller than her'. Also, attributes could have grouped or hierarchical relationship [58]. More generally, the conceptual attributes and the objects can be organised into a knowledge-based hierarchical structure, which is known as ontology engineering [32, 100]. By this step, the fundamental problem is human knowledge itself since visual perception may not be adequately described by semantics or organised systematically. Existing work has realised the knowledge-based explicit attributes could be unreliable or debating [4, 57]. However, these work attempt to address the problem through machine learning aspects regardless the fundamental semantic paradox. To circumvent the conflict between visual and semantic relationship, the data-driven attributes aim to implicitly reconcile the visual-semantic gap [151]. Yet, the misuse of attributes still require future investigation in the future.

Extensive Applications Attributes have been used for describing images [37, 39] and for retrieval [121]. The most common usage of attributes is classification [6, 63, 116, 153]. The underlying rationale is that attribute is a higher-level abstraction of specific categories. Instead of mapping instances to a label, attribute learning aims to discover more general labels that specify the characteristics shared by many classes. In this way, learning attributes can estimate the distribution of completely unseen categories, which is the problem we concern in ZIC. Apart from classification tasks, attribute can improve the interpretation of the model. Especially for deep learning, the surprisingly high performance can hardly be explained by the single value of accuracy. Analysing the huge number of parameters in the model is also intractable. Therefore, attributes can be added as the auxiliary loss to provide interpretation with the co-occurrence. For example, if an image is confidently predicted as

a bear, but the colour attributes are predicted as black and white. It is likely that the test image is a panda, even though the corresponding label is not included in the training set.

2.3 Zero-shot Learning

Zero-shot learning is getting increasing attentions. This section aims to coarsely review the development of ZSL so as to have an overview of the previous pioneering work. The differences to the contributions of this thesis will not be discussed here but will be covered in later specific chapters. Generally speaking, the contributions to ZSL can be summarised into three directions: low-level, mid-level, and high-level. The low-level focuses on more general machine learning concerns, ZSL is a generalisation of special case for conventional supervised methods, such as Error Correction Output Codes [110], Fuzzy Inference [82], active learning [143]. In the mid-level, ZSL techniques focuses on framework design [51, 52, 108, 145, 147, 148] and knowledge representation [1, 47, 136]. Also, it has highly related inspiration to one/few-shot learning [134] and other forms of transfer [23, 106, 107]. The intersection between vision and linguistics is also the unique charm of ZSL, such as relative models [25, 125]. For higher-level, ZSL has various applications. It has been successfully applied to remote sensing images [77], robotic control [33], Unseen gesture recognition [90], First-person Decomposition [154], and Land Cover Prediction [61]. Expanding specific introduction for such various techniques has exceeded the scope of this thesis. We will therefore review the most related three aspects to my thesis, which are *learning framework*, *knowledge representation*, and *scenarios and assumptions*. From the review we can see that the blossom of ZSL is just starting.

2.3.1 Learning Framework

The earliest ZSL techniques can refer to zero-data learning [74] using templates or some earlier transferring approaches [98] using fMRI images. In 2009, Lampert *et al.* [72] firstly applied ZSL for image classification tasks using an attribute-based protocol. Zoological knowledge is encoded into a class-attribute matrix that can be used as to recognise unseen classes that have no examples during training. Such a ZIC task can be achieved by two models that are named Direct Attribute Prediction (DAP) and Indirect Attribute Prediction (IAP). The basic idea is to train an attribute model using training examples and assume the attribute knowledge can hold for unseen classes. It is worth noting that, although most of existing state-of-the-art results are based on DAP, in [63], IAP model is defended as more applicable for the ever-growing concept space. In the future, IAP related models might gain

popularity again for generalised zero-shot learning.

Since ZSL and One-shot learning is closely related in [153], these two problems are unified into a probabilistic framework. The idea of pooling using Bag-of-words is adopted to suppress the variances during knowledge transfer. Besides, more probabilistic models, such as kernel and metric learning [96, 135] are proposed to address the ZSL problem.

The development of ZSL is prominent in 2013. The main contribution is to convert ZSL into an embedding problem. The idea is, rather than predicting each attribute as an independent task, the whole attribute space can be viewed as an embedding of the label space. Such an idea inspires many later frameworks, such as subspace learning, manifold learning, multi-modal learning *etc.* [6, 41, 42, 101, 150]. The embedding framework also enables many deep models can be applied to the problem [40, 124]. However, since the problem is simplified into finding the correlations between two or more embeddings of different modalities, the unique challenges of ZSL are incorporated into general transfer learning problems.

Therefore, the discussion of learning framework of ZSL can be considered from a more general perspective with a larger volume of literatures on transfer learning. But the unique focuses of ZSL shrink to knowledge representations and various scenarios and assumptions that will be reviewed next.

2.3.2 Knowledge Representation

Besides class-specific attributes, [37] adopts instance-level attribute annotations. Each image is annotated by a list of attributes. The advantage is that they do not rely on expert knowledge. But the disadvantage is the extremely high cost for annotations. In [103], a match between semantic word and the corresponding fMRI image of the neural activity is considered. In order to achieve the knowledge representation, both word embeddings and semantic attributes are utilised. After emphasizing the importance of ZSL, [115] firstly concerns the burden of constructing ontologies. The consequent problems are two-fold: class-attribute associations and attributes mining, which concern the annotation and design cost, respectively. Three measures: hierarchical path length of the WordNet, Word2Vec similarity, and World Wide Web hit-count are proposed to count the importance of each attribute to a class.

Another creative work is presented by [104], which introduces an intuitive form of relative attributes. Rather than thinking the presence/absence of an attribute, [104] considers the degree of the attribute into a ranking list of corresponding categories. However, due to relative attributes require more expensive human annotations, the related approaches are not the main focuses in the ZSL field.

More concerns of attributes for ZSL concentrate in 2014. So far, the attribute models are trained by counting the co-occurrence [94] to visual features (like most of the conventional supervised methods). However, [58] believes the correlation between different attributes can harm the co-occurrence-based scheme. For example, ‘iron’ is always accompanying with ‘black’ in the training set. Then, it is unlikely to simultaneously recognise an object with ‘white iron’. More severely, we cannot guarantee the learnt model is exactly targeting the attributes that we want the machine to learn, despite high recognition performance. Besides the debate among the semantic characteristic, the attribute annotation may be not flawless either [57]. Especially for those instance-level approaches, the annotation often requires a collaborative website, such as Amazon Mechanical Turk. The quality of annotations may not hold for different annotators.

For other knowledge representation, [36] proposes a textual model that can directly manipulate ZSL by semantic descriptions. Also, [151] attempts to dispense from the attribute by directly assigning seen classes as positive and negative to unseen classes. [95] follows [114] and uses the hierarchy of ImageNet to infer exemplars of unseen classes. Specifically, they average the ancestor nodes of the unseen class and use the mean vector as the exemplar representation.

Except considering the ‘horizontal’ split between the source and target domain, most of the rest continuously follow the multi-view assumption [43, 149] and attempts to find a solution through manifold learning [44, 55]. To this point, more and more approaches are not satisfied with visual attributes [5, 79, 80]. Textual [76], hierarchical [7] and prototype [60] models gain popularity. [5] provides a comprehensive evaluation of different types of knowledge models, regarding attributes, word2vector, and hierarchy. An interesting topic is to utilise human gaze on ZSL [64]. [3] also utilise strong supervision for ZSL.

2.3.3 Assumptions and Scenarios

Since knowledge transfer has been accepted as the standard approach for ZIC, in 2011, [114] argues that existing methods are surprisingly restricted to small close-sets. To improve, they attempt ZSL on the 1K object classes of the subset of ImageNet. Another interesting work [91] unifies attribute learning with image description, which also aims to weight attributes for each class automatically. Rather than purely mining the associations within the semantic information, [91] also takes visual features into account and optimises a joint distributional loss.

Aside from learning frameworks, [119] argues the explicit attributes may not be sufficient to represent visual features. Therefore, attribute augmentation aims to incorporate data-driven attributes to enrich the semantic representation. Furthermore, [113] develops

the leverage unlabelled data in [153] and introduce Transductive ZSL settings and followed by various models [41] for ZSL frameworks. Interestingly, most of transductive settings focuses on sequential visual problems, such as activity recognition [24].

The success of ZSL in computer vision has aroused the attention of artificial intelligence field [46]. The benchmark of ZSL is rocketed due to adopting deep learnt features [155]. Developing from knowledge transfer, ZSL is then modelled as a domain adaptation problem [66, 116]. The source domain consists of visual examples with annotated attribute knowledge. The test domain only contains knowledge view. And the problem is to predict the visual test domain beforehand. Such adaptation is theoretically analysed by ESZSL [116] which is believed as the simplest state-of-the-art method for ZSL. A similar approach is [81]. Apart from the model, however, the key conclusion is that ‘if the knowledge representation of unseen classes is completely orthogonal to that of source domain, the learnt model will have the minimal effectiveness’. More models [31, 73] and applications [9, 140] of ZSL are proposed as well.

Most approaches further modify the embedding schemes [17, 48, 99, 112, 141, 156, 157] with more specific constraints. More unified joint models are proposed to associate unseen classes using texts [8, 109, 112]. [45] encodes pair-wise relationships for human activity recognition. One of the novel ideas starts to make manipulations on the visual domain, which is shared by [20, 21, 146]. [21, 146] aims to infer augmented data for unseen classes using prior knowledge while [20] goes one step further to infer the expected classifiers for unseen visual data. However, the key conclusions of [20, 22] argues that the main problem of existing ZSL is not the model but the definition. At the point, most of existing approaches are restricted to a small close-set of unseen classes. [20] carries out the first ZSL evaluation on whole ImageNet dataset. The trained model using 1K classes is generalised to 20K+ unseen classes. Later on, in [22], they suggest that anomaly detection can benefit such large-scaled Generalised ZSL (GZSL) problems.

Despite the huge amount of study devoted to ZSL, the benchmark is not clear enough. Since ZSL often involves a complex system of various feature extractors, semantic representations, models, setting, etc., we may feel confused which is the fundamental issue for ZSL. To this end, in 2017, a new benchmark is proposed [142] and mainly compares conventional and generalised ZSL. Different models are fed with the same visual feature as input. Yet, it might be ambitious to use a unified measure that can include unpredictable new attempts in the future due to the large variety of models and also the applicable range.

From the literature review, we can see the unique contribution of the thesis is to convert ZSL into conventional supervised classification via various models. Also, similes, as a promising knowledge representation, are creatively proposed by this thesis. More specific

discussion of the differences will be given in later chapters, depending on each individual contribution.

Chapter 3

Visual-Semantic Ambiguity Removal

Conventional ZSL methods recognise an unseen instance by projecting its visual features to a semantic space that is shared by both seen and unseen categories. However, we observe that such a one-way paradigm suffers from the *visual-semantic ambiguity* problem. Namely, the semantic concepts (e.g. attributes) cannot explicitly correspond to visual patterns, and vice versa. Such a problem can lead to a huge variance in the visual features for each attribute. In this chapter, we investigate how to remove such semantic ambiguity according to visual training examples. In particular, we propose (1) a novel latent attribute space to mitigate the gap between visual features and semantic attributes; (2) a dual-graph regularised embedding algorithm called *Visual-Semantic Ambiguity Removal* (VSAR) that can simultaneously extract the shared components between visual and semantic information and mutually align the data distribution based on the intrinsic local structures of both spaces; (3) a new zero-shot recognition framework that can deal with both instance-level and category-level ZSL tasks. We validate our method on two popular zero-shot learning datasets, AwA and aPY. Extensive experiments demonstrate that our proposed approach significantly performs the state-of-the-art methods.

3.1 Introduction

Conventional ZSL methods [4, 40, 72] directly map visual features to a human-interpretable semantic space and the labels are inferred through human knowledge. However, an inevitable issue of using semantic attributes is the *ambiguity* problem. In linguistics, a concept is considered ambiguous if its extension is deemed lacking in clarity. It is the uncertainty about which objects belong to the concept or which exhibit characteristics that have this predicate. In the context of ZSL, **Visual-Semantic Ambiguity** refers to the situation that a semantic concept (e.g. an attribute) cannot clearly correspond to a certain pattern of visual

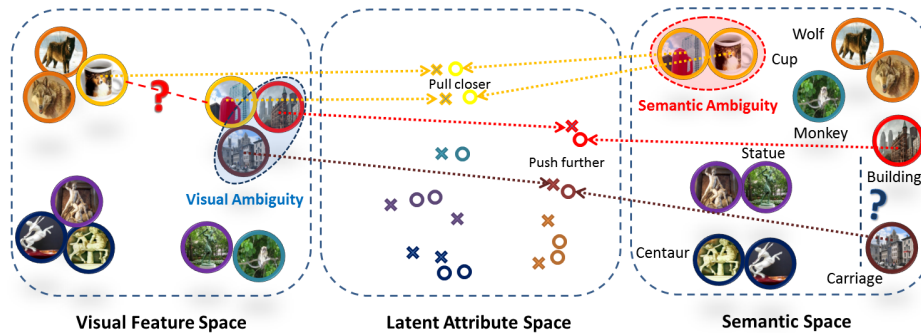


Fig. 3.1 An intuitive illustration of VSAR (best viewed in colour). Visual Ambiguity (in blue oval): the image of a carriage is taken with a building background. It cannot recover the semantic distance (blue question mark) to the building category. Semantic Ambiguity (in red oval): the cup printed with a wolf and the cup-like building share the same semantic expression which can lead to a large visual variance (the red question mark). After embedding to the latent attribute space using VSAR, such ambiguity is mitigated.

data, and vice versa. Therefore, the paradox is that how much difference of the visual patterns can we tolerate for each semantic concept? Alternatively, should we split the concept into sub-concepts to fit the visual data? This is known as the Sorites Paradox that can lead to two extreme solutions. (1) We can accept all instances as if they have the same attribute. Jayaraman and Grauman [57] also study this problem. They provide an extreme example that the concept ‘bumpy’ is assigned to both ‘bumpy road’ and ‘bumpy rash’ which can lead to unreasonable classification results. Unfortunately, most of the existing methods accept this solution regardless the large variation of the attribute. (2) We could refuse any ambiguity and give every instance a unique attribute. For example, compared to ‘smile’, ‘Mona Lisa’s smile’ is clearly referring to a unique visual pattern with no ambiguity. However, it is infeasible to treat everything as unique and assign a new concept to it.

Instead of debating on what is or is not ambiguous, in this chapter, we propose a latent attribute space to mitigate the visual-semantic ambiguity using a novel algorithm named *Visual-Semantic Ambiguity Removal* (VSAR). We measure the visual-semantic ambiguity by the reconstruction error and mitigate it in the latent attribute space. Intuitively, if a semantic concept refers to multiple variations of visual features, it should be split into different regions in the latent attribute space. From the visual aspect, if two close feature points are labelled by different attributes, we should find lower-dimensional subspaces so that they can be discriminated after embedding. Specifically, this is modelled by a graph regularised embedding function that can minimise the reconstruction errors in both visual and semantic spaces. Meanwhile, the regularisation can preserve the discriminative information for recog-

nising unseen categories. We illustrate this idea in Fig. 3.1. Our contribution is three-fold: (1) VSAR can simultaneously remove the ambiguity between visual and semantic information; (2) extensive experimental results suggest the important role of visual-semantic ambiguity to the performance improvement; (3) we introduce a unified framework that can deal with both category-level (AwA dataset) and instance-level (aPY dataset) zero-shot recognition tasks without adjusting the paradigm.

Related Work Since learning visual attributes [39] is proposed, extensive studies [37, 41, 91] have been conducted on how to use attributes as an intermediate representation for ZSL tasks. One interesting direction is to investigate the properties of attributes, such as the label co-occurrence property [94], the relativeness [104], the unreliability [57], and the correlation problem [58] of human-nameable attributes. All of these are semantic properties and therefore suffer from the semantic-visual ambiguity problem. Due to this problem, some work turns to abandon human-nameable attributes and discovers data-driven attributes [70, 151]. However, for ZSL, these methods cannot exploit existing attribute ontologies. Hence, the feasibility is limited. Another trend is based on the embedding framework [4, 80, 150]. All these methods follow the restricted one-way paradigm that suffers from the ambiguity between low-level instances and high-level semantic concepts and labels. Another direction of ZSL is the transductive model [41, 66, 113]. Unlabelled target domain data is collected for learning a transfer function. However, this setting slightly differs from the original ZSL purpose because the target domain may be strictly inaccessible. In contrast, our method can exploit the extensive existing attribute ontology while also stressing the existence of visual-semantic ambiguity and removing it through a learning process.

Some related work also adopts the intermediate embedding [53, 146]. Since we aim to recognise unseen classes using nearest neighbour scheme, the key challenge is to preserve the local structure of the data. Therefore, our unique contribution to [53, 146] is the proposed dual-graph that can efficiently capture the relationship between data. This shares the idea of manifold learning as [145] for transductive ZSL. In contrast, our model is purely inductive that follows conventional ZSL settings.

3.2 Visual-Semantic Ambiguity Removal

Problem setup: The training data is in N 3-tuples of ‘seen’ samples, attributes, and category labels: $(\mathbf{x}_1, \mathbf{a}_1, y_1), \dots, (\mathbf{x}_N, \mathbf{a}_N, y_N) \subseteq \mathbf{X}_s \times \mathbf{A}_s \times \mathbf{Y}_s$, where \mathbf{X}_s is a D -dimensional feature space $\mathbf{X}_s = [\mathbf{x}_{dn}] \in \mathbb{R}^{D \times N}$, \mathbf{A}_s is a M -dimensional attribute space $\mathbf{A}_s = [\mathbf{a}_{mn}] \in \mathbb{R}^{M \times N}$, and $y_n \in \{1, \dots, C\}$ consists of C discrete categories. The bold typeface indicates a space. We use subscript u to denote information of ‘unseen’ space and *hat* denotes information

related to ‘unseen’ samples. During testing, the preliminary knowledge is in \hat{C} pairs of ‘unseen’ category-level attributes and labels: $(\hat{a}_1, \hat{y}_1), \dots, (\hat{a}_{\hat{C}}, \hat{y}_{\hat{C}}) \subseteq \mathbf{A}_u \times \mathbf{Y}_u$, $\mathbf{Y} \cap \mathbf{Y}_u = \emptyset$, $\mathbf{A}_u = [\mathbf{a}_{m\hat{c}}] \in \mathbb{R}^{M \times \hat{C}}$. The goal is to learn a classifier, $f: \mathbf{X}_u \rightarrow \mathbf{Y}_u$, where the samples in \mathbf{X}_u are completely unavailable during training. Such a problem is known as zero-shot learning.

Latent Attribute Embedding: We aim to discover a latent attribute embedding space \mathbf{V} shared by both visual and semantic spaces \mathbf{X} and \mathbf{A} to mitigate the visual-semantic ambiguity. During testing, both \mathbf{X}_u and \mathbf{A}_u can be embedded into \mathbf{V} .

Zero-shot Recognition: Instead of typical two-step prediction $\mathbf{X}_u \rightarrow \mathbf{A}_u \rightarrow \mathbf{Y}_u$, our embedding is two-way from both \mathbf{X}_u and \mathbf{A}_u . Because attribute space \mathbf{A}_u and label space \mathbf{Y}_u are in pairs, we can firstly embed the known \mathbf{A}_u to \mathbf{V} as a knowledge domain. During testing, an unseen image \hat{x} is also embedded to \mathbf{V} so that we can compute the index, i.e., $\mathbf{X}_u \rightarrow \mathbf{V} \leftarrow \mathbf{A}_u \leftarrow \mathbf{Y}_u$.

3.2.1 Latent Attribute Embedding

This is the core component to deal with the visual-semantic ambiguity. We require \mathbf{X}_s and \mathbf{A}_s to compute \mathbf{V} . *In the following, we drop the subscript s for convenience, i.e. we replace $\{\mathbf{X}_s, \mathbf{A}_s, \mathbf{Y}_s\}$ by $\{\mathbf{X}, \mathbf{A}, \mathbf{Y}\}$.* Typically, each dimension \mathbf{a}_m denotes a human-nameable concept, where $M \ll D$. The attribute notions here are instance-level. For the category-level, we can simply set the same attribute vectors to the instances within the same class. For embedding, many previous approaches are based on a forward matrix transformation, i.e. \mathbf{X} to \mathbf{A} . However, because of the visual-semantic ambiguity, the variance in \mathbf{X} is large. Therefore, the forward embedding is difficult to be reconstructed by a backward inverse matrix transformation from \mathbf{A} . Therefore, we insert an intermediate latent attribute space \mathbf{V} between \mathbf{X} and \mathbf{A} , where $\mathbf{V} = [v_{kn}] \in \mathbb{R}^{K \times N}$. K is the dimension of the embedding space. A straightforward setting is $M \leq K \leq A$. However, we stress that K can be any positive whole number. Specifically, we introduce our loss function as:

$$J = \|\mathbf{X} - U_1 \mathbf{V}\|_F^2 + \alpha \|\mathbf{A} - U_2 \mathbf{V}\|_F^2, \quad (3.1)$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix, which estimates the Euclidean distance between two matrices. The shared embedding space \mathbf{V} is decomposed from both \mathbf{X} and \mathbf{A} , where $U_1 = [u_{1dk}] \in \mathbb{R}^{D \times K}$ and $U_2 = [u_{2mk}] \in \mathbb{R}^{M \times K}$ are the basis matrices of the visual feature and attribute space, respectively.

Using Eq. 3.1, it becomes easier to understand the properties of the latent attribute space and how it could mitigate the visual-semantic ambiguity. Optimising Eq. 3.1 aims to minimise the reconstruction errors that are from \mathbf{V} to \mathbf{X} and from \mathbf{V} to \mathbf{A} , respectively. To achieve the optimal solution, U_1 and U_2 should preserve the principal components between \mathbf{X} and \mathbf{A} . This differs from unsupervised methods, such as PCA, that only analyse the data structure in a single domain. Our Eq. 3.1 can reduce the variance of the embedded data that comes from both visual and semantic domains. α is a reliability parameter that can balance the strengths of the two terms.

3.2.2 Dual-graph Regularisation

The above Eq. 3.1 can reduce the difference between the data structures of \mathbf{X} and \mathbf{A} . However, it cannot preserve the discriminative information. For instance, if the gap between \mathbf{x}_n and \mathbf{a}_n is too large, their corresponding weights tend to be minimised to very small values. As a result, the learnt latent attributes are the principal components that are shared by all of the categories. For the purpose of ZSL, we need to preserve the intrinsic geometrical structure so that the learnt representation is discriminative.

We achieve this goal by taking the local invariance assumption and model the problem through a spectral graph approach named *Dual-graph Regularisation*. In particular, this is a combination of two supervised graphs that model the relationship between \mathbf{X} and \mathbf{Y} , and \mathbf{A} and \mathbf{Y} . The main criteria is to preserve the local structures. Therefore, we need the two graphs to simultaneously estimate the data structures of both spaces. Each graph has N vertices that correspond to N data points in the training set. As mentioned earlier, our method can effectively handle ZSL tasks for both instance-level and category-level attribute scenarios. In particular, for *instance-level attributes*, we put an edge between each data point \mathbf{x}_n or \mathbf{a}_n and its p nearest neighbours. For each pair of the vertices s_i and s_j in the weight matrix, $w_{ij} = 1$ if and only if s_i and s_j are connected by an edge, otherwise, $w_{ij} = 0$. As a result, we can separately compute two weight matrices $W_{\mathbf{X}}$ and $W_{\mathbf{A}}$.

It is noteworthy that for *category-level attributes*, $W_{\mathbf{A}}$ is computed slightly different. Every vertex in the same category are connected by a normalised edge, i.e. $w_{ij} = p/n_c$, if and only if \mathbf{a}_i and \mathbf{a}_j are from the same category c , where n_c is the size of category c .

In the embedding space \mathbf{V} , we expect that if the s_i and s_j in both graphs are connected, each pair of embedded points \mathbf{v}_i and \mathbf{v}_j are also closed to each other. However, for the *visual-*

semantic ambiguity problem, $W_{\mathbf{X}}$ and $W_{\mathbf{A}}$ usually give contradictory results. To compromise such conflict, we use the same reliability parameter α in Eq. 3.1 to linearly combine the two graphs, i.e. $W_{ij} = W_{\mathbf{X}_{ij}} + \alpha W_{\mathbf{A}_{ij}}$. The resulted regularisation is:

$$\begin{aligned} \mathbf{R} &= \frac{1}{2} \sum_{i,j=1}^N \|v_i - v_j\|^2 w_{ij} \\ &= \text{Tr}(\mathbf{V} \mathbf{D} \mathbf{V}^T) - \text{Tr}(\mathbf{V} \mathbf{W} \mathbf{V}^T) = \text{Tr}(\mathbf{V} \mathbf{L} \mathbf{V}^T), \end{aligned} \quad (3.2)$$

where D is the degree matrix of W , $D_{ii} = \sum_i w_{ij}$. L is known as graph Laplacian matrix $L = D - W$ and $\text{Tr}(\cdot)$ computes the trace of a matrix. We combine Eq. 3.1 and 3.2 using a regularisation parameter λ to control the balance between reconstruction error and local structure preservation. The final goal is to optimise the following equation:

$$J = \|\mathbf{X} - U_1 \mathbf{V}\|_F^2 + \alpha \|\mathbf{A} - U_2 \mathbf{V}\|_F^2 + \lambda \text{Tr}(\mathbf{V} \mathbf{L} \mathbf{V}^T), \quad (3.3)$$

3.2.3 Optimisation Strategy

Each term of the above Eq. 3.3 is convex, but the combined expression of U_1, U_2, \mathbf{V} is non-convex. To our best knowledge, there is no direct solution to find the global optima. Instead, we adopt an alternating optimisation strategy to find the local minima for each term separately as a relaxed solution. Specifically, the whole task is in turn separated into three sub-problems.

1. sub-problem U_1 : Suppose we compute the partial derivative of the overall loss function J with respect to U_1 , U_2 and \mathbf{V} are fixed as constants. It then becomes a standard least squares problem. Let the partial derivative equal to zero, we have the closed form solution:

$$\begin{aligned} \frac{\partial J}{\partial U_1} &= -2\mathbf{X}\mathbf{V}^T + 2U_1\mathbf{V}\mathbf{V}^T = 0, \\ U_1 &= \mathbf{X}\mathbf{V}^T (\mathbf{V}\mathbf{V}^T)^{-1}. \end{aligned} \quad (3.4)$$

2. sub-problem U_2 : Similar to the sub-problem 1, we can fix U_1 and \mathbf{V} , and compute the partial derivative of J with respect to U_2 . The corresponding solution is:

$$U_2 = \mathbf{A}\mathbf{V}^T (\mathbf{V}\mathbf{V}^T)^{-1}. \quad (3.5)$$

Since we do not expect any prior bias from the unnormalised magnitudes of the training

data, the basis vectors in the matrices should be normalised to unit vectors via:

$$u_{1_{dk}} \leftarrow \frac{u_{1_{dk}}}{\sqrt{\sum_d u_{1_{dk}}^2}} \quad u_{2_{mk}} \leftarrow \frac{u_{1_{mk}}}{\sqrt{\sum_m u_{2_{mk}}^2}}.$$

3. sub-problem V: Fix U_1 and U_2 , we can then update \mathbf{V} . Applying the matrix properties $Tr(AB) = Tr(BA)$ and $Tr(A^T) = Tr(A)$, and we set the partial derivative respect to \mathbf{V} to zero:

$$\frac{\partial J}{\partial \mathbf{V}} = 2((U_1^T U_1 + \alpha U_2^T U_2) \mathbf{V} + \mathbf{V}(\lambda L) - (U_1^T \mathbf{X} + \alpha U_2^T \mathbf{A})) = 0. \quad (3.6)$$

Since space U_1 , U_2 and L are disjointed, this forms a typical Sylvester equation that has the unique solution for \mathbf{V} . We use the *lyap()* function in MATLAB to solve this problem.

Batch sampling scheme: In practice, the computational complexity of solving the Eq. 3.6 is $\mathcal{O}(N^3)$. To improve the efficiency, we adopt a batch sampling scheme like the deep learning strategy. The whole training set is divided into t batches by randomly sampling training instances from each categories. The size of each batch roughly equals to $\frac{N}{t}$. As a result, the computational complexity is reduced to $\mathcal{O}\left(t \left(\frac{N}{t}\right)^3\right)$, where $\left(\frac{N}{t}\right)^3 \ll N^3$. Each batch is in turn used to optimise the loss function in Eq. 3.3. We turn to the next batch until it converges on the previous batch. The whole learning procedure is summarised in Algorithm 1.

Algorithm 1 : Visual-Semantic Ambiguity Removal

Input: $\{\mathbf{X}, \mathbf{A}, \mathbf{Y}\}$, α , λ , K , p , number of batch t .

Output: \mathbf{V} , U_1 , and U_2 .

- 1: Initialisation: random batch sampling $\{\mathbf{X}_1, \mathbf{A}_1, \mathbf{Y}_1\} \dots \{\mathbf{X}_t, \mathbf{A}_t, \mathbf{Y}_t\}$, random initial matrix \mathbf{V} .
 - 2: **for** each batch **do**
 - 3: Compute the graph Laplacian matrix L using Eq. 3.2;
 - 4: **while** Eq. 3.3 is not converged **do**
 - 5: Update U_1 by Eq. 3.5, then normalise U_1 by $u_{1_{dk}} \leftarrow \frac{u_{1_{dk}}}{\sqrt{\sum_d u_{1_{dk}}^2}}$;
 - 6: Update U_2 by Eq. 3.5, then normalise U_2 by $u_{2_{mk}} \leftarrow \frac{u_{1_{mk}}}{\sqrt{\sum_m u_{2_{mk}}^2}}$;
 - 7: Update \mathbf{V} by Eq. 3.6;
 - 8: **end while**
 - 9: **end for**
 - 10: **return** \mathbf{V} , U_1 , and U_2 ;
-

3.2.4 Zero-shot Image Classification

Once we obtain the latent attribute embedding \mathbf{V} of the seen data, performing ZSL is straightforward via *least-square approximation* between \mathbf{V} and $\{\mathbf{A}, \mathbf{X}\}$. During the test, the given informations are the unseen category names and their attributes in pairs: $\{\mathbf{Y}_u, \mathbf{A}_u\}$. We firstly embed all unseen attributes \mathbf{A}_u into the latent embedding space as references: $\mathbf{V}_u = \mathbf{V}\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}_u$. Given a test unseen instance \hat{x} , its embedded latent attribute representation is: $\hat{v} = \mathbf{V}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\hat{x}$. Finally, we adopt a simple NN classifier to predict the category label \hat{c} :

$$\hat{c} = \arg \min_c \|\hat{v} - \mathbf{v}_c\|^2, \text{ where } \mathbf{v}_c \in \mathbf{V}_u. \quad (3.7)$$

3.3 Experiments

Datasets and Settings. We choose two of the most popular datasets for evaluating ZSL tasks. (a) **AwA dataset** [72] is one of the earliest work that particularly proposed for ZSL tasks. Many published results are based on this dataset. Each animal category in AwA is labelled by an attribute signature. (b) **aPY dataset** [37] is an instance-level attribute dataset that each image has a unique attribute signature. In contrast to AwA, aPY covers a more various range of categories, including human, artificial objects, buildings, as well as animals. For comparison reason, we adopt the base features that are provided by the datasets. We carefully follow the standard settings on both of the datasets. In particular, the training/test splits are 40/10 and 20/12 on AwA and aPY dataset, respectively. The optimal reliability parameter α for each dataset is selected from one of $\{0.1, \dots, 0.5, \dots, 0.9\}$ with the step of 0.1 which yields the best performance by 10-fold cross-validation on the training data. For λ and p , cross-validation is still deployed and finally fixed as $\lambda = 0.03$ and $p = 10$. Optimal k is achieved by cross-validation and its effect is evaluated later.

3.3.1 Comparison with the state-of-the-arts

We summarise our comparison in Table 3.1, where the hyphen indicates the existing method has not tested on the datasets in their original publication. Our method significantly outperforms the previous published results and can achieve state-of-the-art performance comparing to most recent approaches. From the confusion matrices in Fig. 3.2 we can see that the recognition rate to each category tends to be averaged. Such a result indicates the performance of our proposed method is stable and reliable. It is also worth noting that, due to the attributes of the two datasets are not both category-level or instance-level, all of the compared methods have to adjust the framework to fit such different settings. In comparison,

Method	aPascal&aYahoo	Animals with Attributes
Farhadi <i>et al.</i> [37]	32.5	-
Mahajan <i>et al.</i> [91]	37.93	-
Akata <i>et al.</i> [4]	-	43.5
Fu <i>et al.</i> [58]	-	47.1
Lampert <i>et al.</i> [72]	19.1	40.5
Jayaraman and Grauman [57]	26.02 ± 0.05	43.01 ± 0.07
Romera-Paredes and Torr [116]	27.27 ± 1.62	49.30 ± 0.21
our VSAR	39.42 ± 0.27	51.75 ± 0.43

Table 3.1 Compare with the published state-of-the-art methods.

our VSAR approach can deal with both of the situations.

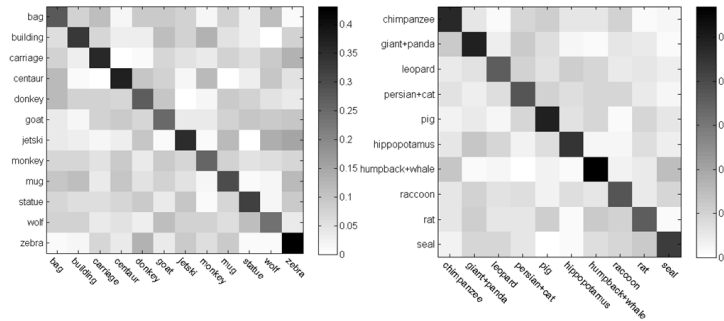


Fig. 3.2 Confusion matrix of ZSL performance on aPY (left) and AwA (right).

3.3.2 Algorithm analysis

Effects of terms in VSAR. To understand the success of our VSAR algorithm, the first important question is how does each terms in our VSAR algorithm work for ZSL. Thus, we separately strip-down each term in Eq. 3.3 into three baseline models. The first model is referred as *X-to-A*, in which we remove the second term of Eq. 3.3 and let the visual space \mathbf{X} directly map to the semantic space, i.e. $\mathbf{V} = \mathbf{A}$. This is exactly a DAP procedure that, during the test, the image is firstly mapped to the semantic space and then classified to the label space. The second model is referred as *A-to-X*. This is an interesting scenario that investigates whether we could regenerate the original visual features given just the semantic representations. Specifically, we train the model by setting $\mathbf{V} = \mathbf{X}$ and remove the first term in Eq. 3.3. During the test, we firstly project all attributes of the unseen categories/instances to \mathbf{X} . A test image is then classified in this embedding space using Eq. 3.7. In the third model that is denoted as *No-Graph*, we explore the importance of our dual-graph regularisations. Specifically, we train the model by setting $\lambda = 0$.

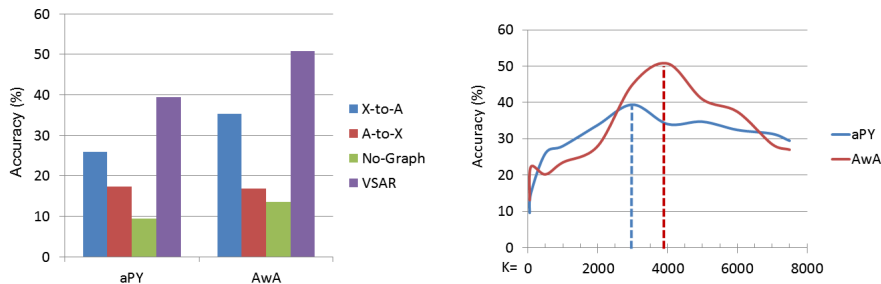


Fig. 3.3 Evaluating each term of the loss function in Eq. 3.3 (left) and the performance curve respects to the dimension K of the latent attribute space (right).

In Fig. 3.3. it can be seen that our full model significantly outperforms all of the baseline methods. In addition, we find the performance of the third model is roughly equal to random guess. Such a failure case matches our previous expectation that, without regularisation, Eq. 3.1 tend to discover the principle components rather than discriminating the categories. It is also noticeable that the *A-to-X* method gets better result on the aPY dataset than that on AwA. We ascribe this to the instance-level attributes. Such a result implies that it is feasible to generate visual features of each image from its semantic representations in future work.

Number of latent attributes. Another important issue is how many latent attributes K are required for the embedding space. Does a larger number of K always give better results? To investigate this question, we gradually increase K from 50, 85, 500, 1000, and 1000 per further step. We show the result in Fig. 3.3 (left). Generally speaking, a larger K tends to benefit the performance. However, we point out that there is an optimal K that gives the peak result. After that, the performance gradually degrades while we further increase K . This problem is severer on AwA than that on aPY. This is because when K goes too large, this can be viewed as an spectral over-fitting problem [158]. Since the attributes of AwA is category-level, the variance of its semantic space is much smaller than its visual space, which results in that the model on the AwA is more likely to over-fitting.

Efficiency Our implementation is conducted in Matlab 2014a environment that is installed on a 12-core Linux system with 400G memory. The test time is done within a second. The training process takes roughly half an hour (i.e. number of batches $t = 15$) to get a converged model. Most of the time is used for solving the Eq. 3.6. We stress our contribution of using the batch sampling scheme, whereas directly solving the Eq. 3.7 without the batch sampling scheme can take up to 10 hours.

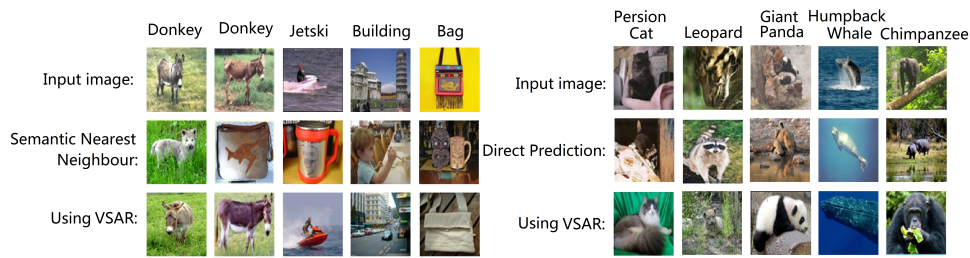


Fig. 3.4 Examples of successful semantic ambiguity removal on aPY (left) and the visual ambiguity removal on AwA (right).

3.3.3 Visual-semantic ambiguity removal

In this section, we investigate what kinds of visual-semantic ambiguity are removed using our algorithm. This question can be considered from two aspects. Firstly, we consider the semantic ambiguity between different categories. On aPY dataset, we find such a semantic ambiguity problem is very severe. We use the provided “ground truth” attribute labels as the representation for each image. We then search the nearest neighbour for each image like an 1-NN classification. We find that only 67.17% of the nearest neighbours can match their original categories. Such a result implies that even if the conventional attribute classifiers can give perfect predictions, the overall recognition rate is only 67.17%. In Fig. 3.4 (left), we show that our VSAR is able to remove some of the semantic ambiguities. For example, in the second columns, the test image ‘donkey’ is misclassified as a ‘bag’ because the material and the logo of the bag possesses the same attributes to the donkey. However, in the visual space, such two instances are very distinctive. Therefore, using VSAR, our method successfully removes the ambiguity and gives the correct nearest neighbour. On the AwA dataset, the semantic ambiguity does not exist because all of the images in one category share the same attributes. Therefore, we consider the problem of visual ambiguity, i.e. the extracted low-level features from different categories are confused to each other. Specifically, we compare our method with the DAP framework using the X-to-A model. In Fig. 3.4 (right), we show some prediction errors in DAP can be corrected using VSAR. Such an ability contributes to the remarkable performance improvement (39.42% to 51.75%) in Fig. 3.3.

3.4 Conclusion and future work

We can conclude that the visual-semantic ambiguity is a common issue in ZSL tasks. Our results on both datasets support that ambiguity removal can significantly benefit the recognition performance. The proposed VSAR is an unified framework that can deal with various

semantic inputs, such as category-level and instance-level attributes. Instead of treating ZSL as a multi-label classification task, we adopt an embedding approach without struggling with the effectiveness of each attribute concept. Due to this property, our method can be simply applied to various existing intermediate semantic representations, such as data-driven attribute [151] or word-vector [124]. In the future, we plan to extend our visual-semantic constrains to multilateral in order to simultaneously incorporate multiple types of visual, semantic, as well as hierarchical label information.

In this chapter, our experiments are based on heuristic visual feature. In later chapters, we will see a significant performance boost using deep features. Also, the proposed graph regularisation is later on adopted for inferring unseen visual features, which further proves the effectiveness of VSAR.

Chapter 4

Towards Open Zero-shot Learning

Existing Zero-shot Learning can leverage attributes to recognise unseen instances. However, the training data is limited and cannot adequately discriminate fine-grained classes with similar attributes. In this chapter, we propose a complementary procedure that inversely makes use of attributes to infer discriminative visual features for unseen classes. In this way, ZSL is fully converted into conventional supervised classification, where robust classifiers can be employed to address the fine-grained problem. To infer high-quality unseen data, we propose a novel algorithm named *Orthogonal Semantic-Visual Embedding (OSVE)* that can discover the tiny visual differences between different instances under the same attribute in an orthogonal embedding space. On two fine-grained benchmarks, CUB and SUN, our method remarkably improves the state-of-the-art results under standard ZSL settings. We further investigate the *Open* ZSL problem where the number of seen classes is significantly smaller than that of unseen classes. Substantial experiments manifest that the inferred visual features can be successfully fed to SVM which can effectively discriminate unseen classes from fine-grained open candidates.

4.1 Introduction

Zero-Shot Learning [72, 74, 103, 124] aims to train semantic models that can generalise to new classes without acquiring unseen visual data at training stage. The standard paradigm of ZIC framework is shown in Fig. 4.1 (blue path), where a closed-set of seen instances are used to learn a visual-semantic mapping. During the test, images from unseen classes can be firstly mapped to the semantic space and predictions can be made by choosing one of the candidates that are pre-defined by attribute descriptions. However, while new semantics and unseen classes can be incrementally added to the system, the training data is restricted to the closed-set of seen classes without expansion. Under such a framework, there are mainly

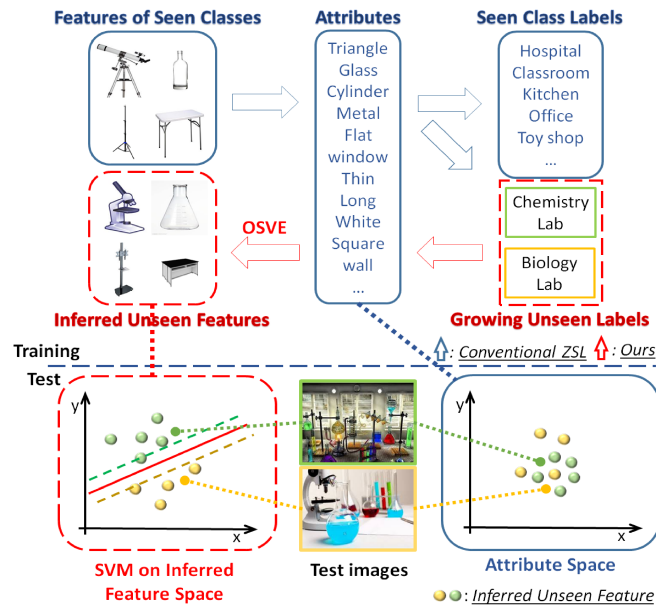


Fig. 4.1 Comparison between our procedure (Red) and the conventional ZSL framework (Blue). Fine-grained classes are often compact and non-describable in the attribute space. Our OSVE can discover tiny visual differences between different instances under the same attribute so as to infer discriminative visual features for unseen classes from fine-grained open candidates.

two problems impeding existing ZIC methods from scaling-up. The first is the *Fine-grained* problem. Namely, the classes are close in the taxonomy, which results in very similar semantic descriptions. Due to existing methods rely on visual-semantic mapping, unseen classes with similar attributes cannot be adequately discriminated. The second is known as the *Open Zero-shot Learning* problem which removes two main unrealistic restrictions of conventional ZIC: 1) all of the candidates for test image must come from unseen classes; 2) the number of seen classes is larger than unseen classes. The first restriction is caused by the correlation problem during attribute designing the results in two attributes A and B may be present or absent together all the time during training. As a result, the test image with only attribute A will be predicted as $A \& B$ that is biased towards the seen classes. The second restriction considers the limited size of the training set. Without various seen instances, the learnt semantic model can hardly adapt to unseen classes from a large number of candidates.

In this chapter, we propose a complementary approach that inversely infers visual data to train discriminative models for unseen classes. Our method is inspired by the fact that we human can roughly imagine the appearance of unseen objects by associating previous seen classes. Accordingly, as shown in Fig. 4.1 (Red), our method can inversely infer discriminative visual features from attribute descriptions of unseen instances. In this way, inferred features can be used to train classifiers for unseen classes as conventional supervised learn-

ing, *e.g.* SVM. Such a new framework has two potential advantages. Firstly, the training set can be expanded to new unseen classes, which can benefit the open ZSL problem if the number of unseen classes becomes large. Secondly, our classifier is now trained on the original visual feature space without quantisation to the attribute space that is often too compact for the fine-grained problem. For example, the *Biology Lab* and *Chemical Lab* are not discriminative in the semantic space since they share most of the attributes. But, in the visual space, we can enlarge tiny differences between various instances with the same attribute, which, consequently, make fine-grained classes more discriminative. To confirm this argument, we conduct SVM on both visual features and attributes to predict the class labels in comparison. On SUN which has 717 fine-grained classes, SVM on VGG-19 features achieves 89.8 % overall accuracy, whereas SVM on the corresponding attributes results in only 72.4 %.

In spite of that our idea is simple and intuitive, there are two main unsolved technical issues. 1) *Semantic-visual discrepancy*: since attributes are compact high-level representations whereas visual data is usually long-tailed low-level features, the data structure in the two spaces are distinctive. Two close points in the attribute space can be far away in the visual feature space, and vice versa. Due to the structural difference, normal embedding processes are prone to learn the principal components between the two spaces, by which the learnt feature distribution is concentrated and not discriminative. 2) *Semantic correlation*: like that in the conventional ZSL framework, different attributes may be assigned to the same pattern of visual features. As a result, the inferred unseen features are prone to fall into the clusters of seen features. Considering the above two problems, we propose a novel Orthogonal Semantic-Visual Embedding (OSVE) algorithm to infer visual features from attributes. The key idea is to find an intermediate embedding space that can compromise the structural difference between the visual and semantic space. Meanwhile, we hope to remove the correlations between different attributes, and between seen and unseen classes. To this end, our algorithm jointly optimise the semantic-visual reconstruction error and the orthogonalisation, where the redundancy can be removed in the orthogonal embedding space so that the remaining bases are then decorrelated. We summarise our contributions as follows.

i. We propose to inversely infer discriminative visual features from the attributes of unseen classes. Such a framework can make the training set grow with newly added unseen classes in the open ZSL problem. Typical powerful classifiers, such as SVM, can be employed directly in the feature space rather than the attribute space to improve the fine-grained recognition performance.

ii. We propose a novel OSVE algorithm that can effectively infer visual features and meanwhile remove the correlations. On two benchmarks, our OSVE outperforms state-of-the-art methods under conventional ZSL scenarios.

iii. We further investigate two sets of Fine-grained Open ZSL tasks. On both sets of tasks, our OSVE demonstrates promising recognition performance. Extensive experiments manifest that our algorithm can successfully capture the significant visual features from the attributes of unseen classes.

The rest of the chapter is organised as follows. In Section 2, we review related ZSL approaches. In Section 3, our algorithm is formalised and introduced. We provide extensive experiments on both conventional and fine-grained open ZSL settings in Section 4. In the last Section 5, we conclude our work and state some possible future work.

4.2 Related work

We compare our paradigm and that of conventional ZSL in Fig. 4.1. Most of previous ZSL work is based on (or similar to) the framework called Direct Attribute Prediction (DAP) [72, 73, 91, 153]. For each attribute, a binary classifier is trained using all of the seen classes. During the test, a prediction can be made by Maximum-a-Posteriori criteria over all of the outputs of the binary classifiers. The main drawback of such framework is the correlation problem that reported in [58]. Besides, the human-defined attribute list can be unrealistic and noisy and need to be selected [37, 57, 84, 86]. Therefore, many previous work seeks for an effective form of semantic representation such as class taxonomies [87, 95, 114], or textual features [94, 115]. However, due to other semantic sources cannot provide direct and compact descriptions to the visual appearances, semantic attributes remain the most popular side information for ZSL learning.

A recent trend of ZSL methods adopts the framework of Attribute-Label Embedding (ALE) that jointly estimate all of the attributes by an embedding function from visual to attribute space. Such a framework skilfully avoid the correlation problem or attribute selection since the embedding can optimise the weight of each attribute. Moreover, such a framework be straightforwardly combined with Deep Neural Network [112]. The much recent research adopts the embedding approach and demonstrates state-of-the-art performance [5, 66, 111, 116, 155, 156]. The remaining challenges so far is to break the restrictions of conventional ZSL settings. [42, 113] focus on transductive settings which view ZSL as a domain adaptation problem. These methods are based on the assumption that unlabelled data of unseen classes can be obtained. Reed *et al.* [112] addresses fine-grained ZSL by a Deep Symmetric Structured Joint Embedding (DA-SJE). Zhang and Saligrama [155] investigate how their method can withstand the reduction of the training set size. Our settings can be viewed as *Transductive Labels* when we use instance-level unseen attributes. Such

an assumption is particular useful for fine-grained problem, *i.e.* the class cannot properly summarises the attributes of the huge variations of instances.

Aside of ALE, some work also considers the drawbacks of direct mapping from visual to semantic spaces. Accordingly, latent attributes [6, 119, 141, 152] aims to discover the statistical relationships between visual and semantic features so as to eliminate the human bias in the attributes. Yu *et al.* [151] use one-to-one classifiers to estimate the similarity of between pair of classes. [6, 84, 118] aim to remove the visual-semantic ambiguity through an intermediate embedding space. [141] proposes bilinear joint embeddings to mitigate the distribution difference between visual and semantic spaces. In [20], classifiers of unseen classes are directly estimated by aligning the manifolds of seen classes.

In comparison to previous methods, our work aims to simultaneously address both fine-grained and open ZSL problems using a unified framework. Our work also adopts attributes as the side information and shares the idea of latent embedding, but our method is inverse and complementary to existing work. While most of the previous methods focus on visual to semantic embedding, our approach focuses on semantic-visual embedding, which is more challenging and requires more consideration. We also consider the imperfection of human-designed attributes, for which we propose a novel orthogonalised embedding approach. The most related work is [21] that attempts to predict visual exemplars for unseen classes. However, their output is a single point in the semantic embedding space, whereas our method can infer instance-level visual features, the number of which equals to that of unseen instances. In short, our unique contribution is to convert ZSL problem into the conventional supervised classification for fine-grained open ZSL using orthogonalised latent embedding.

4.3 Visual Feature Inference

4.3.1 Problem setup

The training set contains samples, attributes, and class labels that are in 3-tuples: $(\mathbf{x}_1, \mathbf{a}_1, y_1), \dots, (\mathbf{x}_N, \mathbf{a}_N, y_N) \subseteq \mathbf{X}_s \times \mathbf{A}_s \times \mathbf{Y}_s$, where N is the number of training samples; $\mathbf{X}_s = [\mathbf{x}_{dn}] \in \mathbb{R}^{D \times N}$ is a D -dimensional feature space; $\mathbf{A}_s = [\mathbf{a}_{mn}] \in \mathbb{R}^{M \times N}$ is a M -dimensional attribute space; and $y_n \in \{1, \dots, C\}$ consists of C discrete class labels. In order to deal with fine-grained open ZSL, we use instance-level attributes, *i.e.* each image is paired with a unique attribute signature. Suppose there are \hat{N} pairs of ‘unseen’ attributes from \hat{C} discrete classes: $(\hat{\mathbf{a}}_1, \hat{y}_1), \dots, (\hat{\mathbf{a}}_{\hat{N}}, \hat{y}_{\hat{N}}) \subseteq \mathbf{A}_u \times \mathbf{Y}_u$, where $\mathbf{Y}_u \cap \mathbf{Y}_s = \emptyset$, $\mathbf{A}_u = [\mathbf{a}_{m\hat{n}}] \in \mathbb{R}^{M \times \hat{N}}$. The goal of zero-shot learning is to learn a classifier, $f: \mathbf{X}_u \rightarrow \mathbf{Y}_u$, where the samples in \mathbf{X}_u are completely unavailable during training. Again, we use *Bold* typeface to indicate a space. Subscript s

and u refer to ‘seen’ and ‘unseen’. *hat* denotes the variables that are related to ‘unseen’ samples.

Semantic-Visual Embedding: We aim to infer the visual features of unseen classes given the semantic attributes. Specifically, we learn a embedding function on the training set $f : \mathbf{A}_s \rightarrow \mathbf{X}_s$. After that, we are able to infer \mathbf{X}_u though: $\mathbf{X}_u = f(\mathbf{A}_u)$.

Zero-shot Recognition: Using the inferred visual features, we can directly estimate the probability distribution of the unseen classes. It is straightforward to employ existing supervised classification methods, *i.e.* $f : \mathbf{X}_u \rightarrow \mathbf{Y}_u$.

4.3.2 Orthogonal Semantic-Visual Embedding

Conventional ZSL methods minimise the single classification error of each attribute. Due to the attributes are separately learnt, as aforementioned, such a framework highly depends on the quality of the designed attributes. A better approach is to regard all of the attributes as an embedding of the class label [4]. Then, an objective function is learnt to simultaneously minimise the multi-class error and also consider the relationship between different attributes. A typical multi-attributes regressor can be formalised as the following problem:

$$\min_W \mathbb{L}(W\mathbf{X}_s, \mathbf{A}_s) + \lambda\Omega(W), \quad (4.1)$$

where W is the mapping matrix, \mathbb{L} is a loss function, and Ω is a regularisation term with its hyper-parameter λ . During the test, an unseen instance can be directly mapped to the attribute space by: $\hat{a} = W\hat{x}$.

However, due to W is learnt using only the training data, the inferred attributes \hat{a} are prone to be biased towards the ‘seen’ attributes \mathbf{A}_s . Since the number of dimension of the visual feature is dominantly large, *i.e.*, $D \gg M$, the mapped semantic data may discard important information to distinguish fine-grained unseen classes. Inspired by the idea that a human can imagine the visual appearance of an unseen object through given semantic descriptions, we proposed to infer the visual feature of the unseen classes by reversely learning a mapping function from semantic space to the visual feature space. In the process of dimension augmentation from semantic to visual space training set, the representation gains more information. Such information can benefit unseen attributes to infer more discriminative visual features:

$$\min_W \mathbb{L}(W\mathbf{A}_s, \mathbf{X}_s) + \lambda\Omega(W). \quad (4.2)$$

The loss term accounts the reconstruction error between the semantic input and visual output; whereas the regularisation ensures the discrimination to unseen classes. Such a framework provides a direct mapping to the visual space without computing a pseudo-inverse matrix that can lead to information loss. Before the test, it is straightforward to infer the visual features of unseen classes using their class attributes:

$$\mathbf{X}_u = W\mathbf{A}_u. \quad (4.3)$$

In spite of the simplicity of the above framework, several problems are worth noting. Firstly, in practice, there is often a huge gap between visual and semantic spaces. Compared to the compact attribute representation, the variance of visual data is usually larger due to outliers and noise. Also, the data distribution of the two spaces is distinctive. Thus, directly mapping from semantic to visual space can lead to inferior performance. We propose to insert a latent embedding space \mathbf{V} to reconcile the semantic space with the visual feature space, where $\mathbf{V} = [v_{kn}] \in \mathbb{R}^{K \times N}$, and K is an adjustable number of dimension of \mathbf{V} . Secondly, in order to learn discriminative features, we need to remove the correlation between each attribute so as to ensure better generality. For this purpose, the embedding space should be strictly orthogonal. If we consider a multi-variable linear regression model, the loss function can be defined as:

$$J = \|\mathbf{X}_s - W_1\mathbf{V}\|_F^2 + \|\mathbf{V} - W_2\mathbf{A}_s\|_F^2 + \lambda\|W_1\|_F^2 + \lambda\|W_2\|_F^2, \text{ s.t. } \mathbf{V}\mathbf{V}^T = I, \quad (4.4)$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix, which estimates the Euclidean distance between two matrices. The latent embedding space \mathbf{V} is decomposed from \mathbf{X} , and \mathbf{A} is decomposed from \mathbf{V} . $W_1 = [w_{1_{dk}}] \in \mathbb{R}^{D \times K}$ and $W_2 = [w_{2_{km}}] \in \mathbb{R}^{K \times M}$ are embedding matrices. The above Eq. 4.4 helps us to understand our approach. The embedding space can preserve the principal components between the visual and semantic spaces. Meanwhile, the data structure is scattered so that the inferred features can be discriminative and decorrelated to the original attributes. However, because of the fast decay of eigenvalues, the strict orthogonal constraint can impair the reconstruction of the visual features. Therefore, we relax the constraint. The overall loss function is:

$$J = \|\mathbf{X}_s - W_1\mathbf{V}\|_F^2 + \|\mathbf{V} - W_2\mathbf{A}_s\|_F^2 + \lambda\|W_1\|_F^2 + \lambda\|W_2\|_F^2 + \beta\|\mathbf{V}\mathbf{V}^T - I\|_F^2. \quad (4.5)$$

4.3.3 Optimisation Strategy

Each term of the above Eq. 4.5 is convex. However, It is non-convex in W_1, W_2, \mathbf{V} all together. To our best knowledge, there is no direct solution to find the global optima. Here, we

adopt an alternating optimisation strategy to find the local minima for each term separately. Specifically, the whole task is in turn separated into three sub-problems.

1. W_1 -step: Suppose we compute the partial derivative of the overall loss function J with respect to W_1 , then W_2 and \mathbf{V} are fixed as constants. The loss function becomes a standard least squares problem. Let the partial derivative equal to zero; then we have the closed form solution:

$$\begin{aligned} \min_{W_1} \quad & \|\mathbf{X}_s - W_1 \mathbf{V}\|_F^2 + \lambda \|W_1\|_F^2 \\ \frac{\partial J}{\partial W_1} = \quad & -2(W_1 \mathbf{V} - \mathbf{X}_s) \mathbf{V}^T + 2\lambda W_1 = 0 \\ W_1 = \quad & \mathbf{X}_s \mathbf{V}^T (\mathbf{V} \mathbf{V}^T + \lambda I)^{-1}. \end{aligned} \quad (4.6)$$

2. W_2 -step: Similar to the step 1, we can fix W_1 and \mathbf{V} , and compute the partial derivative of J with respect to W_2 . The corresponding solution is:

$$\begin{aligned} \min_{W_2} \quad & \|\mathbf{V} - W_2 \mathbf{A}_s\|_F^2 + \lambda \|W_2\|_F^2 \\ \frac{\partial J}{\partial W_2} = \quad & -2(W_2 \mathbf{A}_s - \mathbf{V}) \mathbf{A}_s^T + 2\lambda W_2 = 0 \\ W_2 = \quad & \mathbf{V} \mathbf{A}_s^T (\mathbf{A}_s \mathbf{A}_s^T + \lambda I)^{-1}. \end{aligned} \quad (4.7)$$

3. \mathbf{V} -step: \mathbf{V} should be solved carefully. Since \mathbf{V} is related to all of the three terms, it balances how accurate can we infer the visual feature and how discriminative can the inferred features generalise to unseen data. We propose to solve \mathbf{V} as an independent sub-problem inside the overall optimisation. Fix W_1 and W_2 , we can get the partial loss function J_v for \mathbf{V} . We then set the partial derivative respect to \mathbf{V} to zero:

$$\begin{aligned} \min_{\mathbf{V}} J_v = \quad & \|\mathbf{X}_s - W_1 \mathbf{V}\|_F^2 + \|\mathbf{V} - W_2 \mathbf{A}_s\|_F^2 + \beta \|\mathbf{V} \mathbf{V}^T - I\|_F^2 \\ \frac{\partial J_v}{\partial \mathbf{V}} = \quad & 2W_1^T (W_1 \mathbf{V} - \mathbf{X}_s) + 2(\mathbf{V} - W_2 \mathbf{A}_s) + 2\beta (\mathbf{V} \mathbf{V}^T - I) \mathbf{V}. \end{aligned} \quad (4.8)$$

Adaptive Gradient Descent: In order to solve the optimal \mathbf{V} , we adopt the adaptive gradient decent strategy to solve Eq. 4.8. We introduce τ to control the learning rate. If J_v keeps converging, τ is increased to accelerate the process. Once J_v becomes diverged, τ is reduced correspondingly to increase the tolerance. Such a strategy is vital for keeping the balance between reconstruction and orthogonalisation. As shown in Fig. 4.2, the solver firstly focuses on optimising the semantic reconstruction and the orthogonality. After 200 iterations, the learning rate becomes over large that causes the loss of visual reconstruction increased

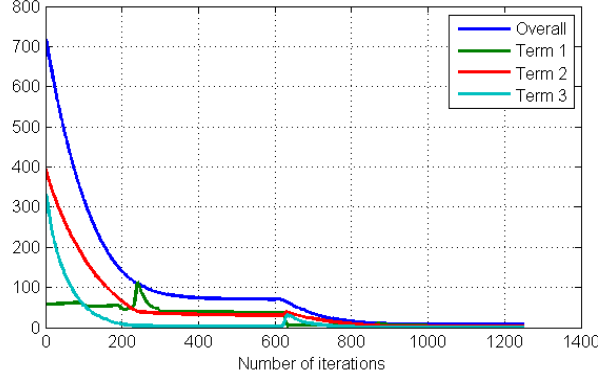


Fig. 4.2 An example of the convergence situations shows the loss with respect to the number of iterations. Term 1 and 2 corresponds to the reconstruction errors to visual and semantic spaces. Term 3 accounts how orthogonal is the embedding space.

dramatically. Thus, τ is immediately reduced so that the three terms start to be optimised together again. Without such an adaptive scheme, it is unable to control the unpredictable divergence of any of the terms. The whole learning procedure is summarised in Algorithm 2.

$$\mathbf{V}_{t+1} = \mathbf{V}_t - \tau \frac{\partial J_v}{\partial \mathbf{V}} \quad (4.9)$$

$$\tau_{t+1} = \begin{cases} 1.2\tau & \text{if } J_{v_{t+1}} < J_v \\ 0.5\tau & \text{otherwise} \end{cases} \quad (4.10)$$

Algorithm 2: OSVE

Input: $\{\mathbf{X}, \mathbf{A}, \mathbf{Y}\}, K, \lambda, \beta, \tau$.

Output: W_1 , and W_2 .

- 1: Initialisation: random initial matrix \mathbf{V} .
 - 2: **while** Rq. 4.5 is not converged **do**
 - 3: Update W_1 by Eq. 4.6;
 - 4: Update W_2 by Eq. 4.7;
 - 5: **while** Eq. 4.8 is not converged **do**
 - 6: Update \mathbf{V} by Eq. 4.9;
 - 7: Update τ by Eq. 4.10;
 - 8: **end while**
 - 9: **end while**
 - 10: **return** W_1 , and W_2 ;
-

Table 4.1 Key statistics of CUB and SUN datasets.

Dataset	CUB	SUN
# of Attributes	312	102
Attribute Type	Binary	Continues
Annotation Level	per image & per class	per image
# of Total Images	11788	13430
Seen/Unseen Split	150/50	707/10

4.3.4 Zero-shot Recognition

Once we obtain the embedding matrices W_1 and W_2 , the visual features of unseen classes can be easily inferred from their attributes:

$$\mathbf{X}_u = W_1 * W_2 * \mathbf{A}_u. \quad (4.11)$$

It is noticeable that for instance-level attributes, \mathbf{X}_u contains as many instances as the test set. The zero-shot recognition task now becomes a conventional classification problem. Thus, any existing supervised classifier, *e.g.* SVM, can be applied. Since we focus on the quality of the inferred features, we compare NN to SVM s well. For NN approach, given a test unseen instance \hat{x} , we can predict its class label \hat{c} by:

$$\hat{c} = \arg \min_c \|\hat{x} - \mathbf{x}_{\hat{n}}\|^2, \text{ where } \mathbf{x}_{\hat{n}} \in \mathbf{X}_u, y_{\hat{n}} = c \in \mathbf{Y}_u. \quad (4.12)$$

4.4 Experiments

We first introduce the datasets, on which we compare our approach to existing state-of-the-art methods. Since the published results are obtained on different settings, in terms of visual features, seen/unseen splits, and semantic side information, we aim to provide a fair comprehensive comparison to most of the outstanding models. We also provide detailed self-comparisons to baseline methods so as to verify the claims we made above. Finally, we investigate our method on the fine-grained open ZSL tasks.

4.4.1 Setup

Datasets and Settings Our method is evaluated on two fine-grained datasets, Caltech-UCSD Birds (CUB) [131] and SUN attribute (SUN) [105]. We summarise the key statistics in Table 4.1. For CUB, there are 11788 images from 200 classes of birds. Many bird species can be hardly differentiated by humans. The usual Seen/Unseen split for ZSL is 150/50. For

Table 4.2 Comparison to state-of-the-art methods for both datasets. Results are overall accuracies in %.

Caltech-UCSD Birds				SUN attribute		
Methods	SI	Shallow features	Deep features	Methods	Shallow features	Deep features
DAP[72]	A	10.5	31.4	DAP[72]	52.50	72.00
AHLE[4]	A+H	18.0	27.3	ZSRwUA[57]	56.18	-
SJE[5]	A+W+H	19.0	47.1	ESEZL[116]	65.75	82.10
UDA[66]	A+W	28.1	40.6	SSE[155]	-	82.50
DS-SJE[112]	A+W+H	-	56.8	JLSE[156]	-	83.83
OSVE+NN	A	20.2	45.2	OSVE+NN	56.96	76.21
OSVE+SVM	A	28.9	60.1	OSVE+SVM	70.59	83.23

SI: side informations, A: attributes, H: hierarchy, W: word2vec.

SUN, the number of classes is 717, which is larger than that of CUB. The total number of images is 13430. Some classes are close on both semantic meanings and visual appearances, *e.g. theatre* and *ballroom*.

Visual Features Existing methods differ in adopted visual features. To make a comprehensive comparison, we implement our method using both shallow features that are released by the datasets and deep features extracted using VGG-19 and released by [155].

Semantic Attributes Both of the datasets now provide instance-level attributes. Each test image is paired with a unique attribute signature based on the actual visual appearance, which is different from the class-level attributes that let all of the images in a class share the same attribute signature. Our method benefits from such a scenario for open ZSL for the reason that, if the number of training classes is small, our algorithm can still discover the differences between instances under the same attribute.

Zero-shot Cross-validation We obtain the optimal hyper-parameters through a new cross-validation strategy. Since we aim to address ZSL problems, traditional cross-validation for multi-label classification is not helpful because all of the seen classes are used for both training and validation. Therefore, we propose a novel leave-one-fold-out strategy. The seen classes are divided into ten disjointed folds. We use one fold as unseen validation set and train models on remaining folds. We choose the set of hyper-parameters which can lead to the highest mean accuracy on all of the ten folds. We fix this set of parameters for the following experiments.

4.4.2 Benchmark Comparison

Comparison to State-of-the-art Methods We first compare to previous published results. Due to few methods are evaluated on both of the datasets, separate the results by the two datasets. We summarise our comparison in Table 4.2.

For CUB, we compare to five methods. DAP [72] is the most common ZSL framework

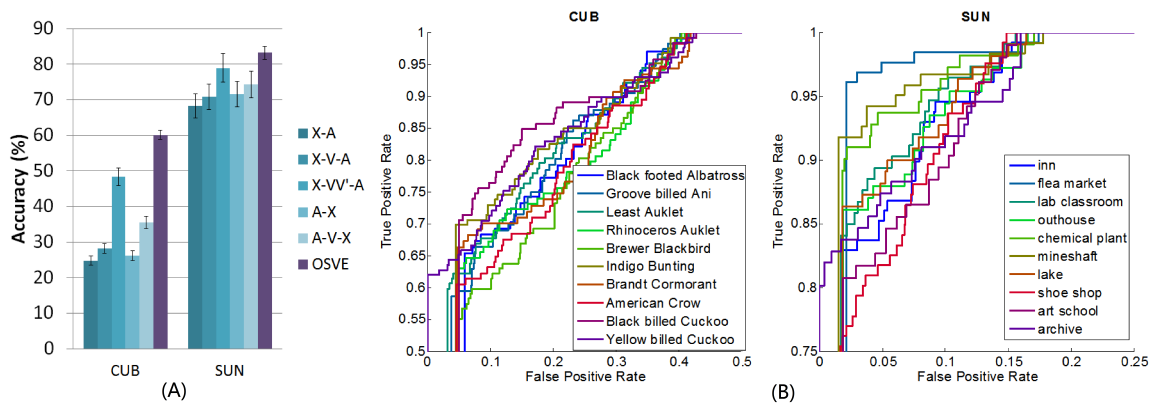


Fig. 4.3 A. overall accuracies of baseline methods by substituting key components of the proposed framework. B. ROC curves of our method on the two datasets. For clarity, only 10 of the 50 unseen classes on CUB are shown.

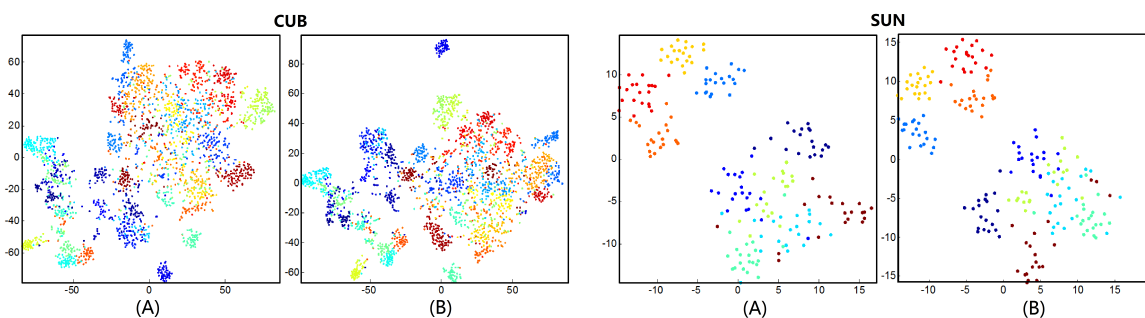


Fig. 4.4 Comparing the data distribution between real (A) and inferred (B) visual features of unseen classes. Note that t-SNE can result in slight distortion and colour differences.

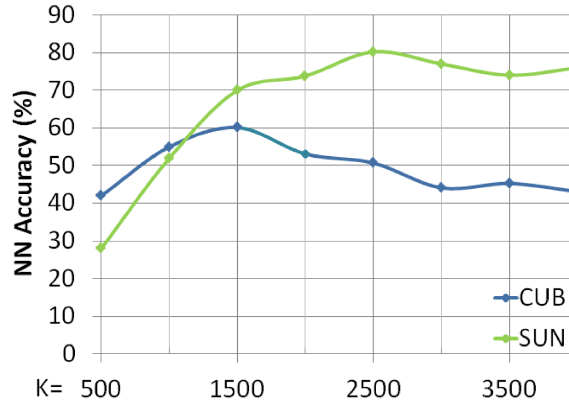


Fig. 4.5 Performance curve with respect to the dimension K of the intermediate embedding space.

that trains binary SVM classifier for each attribute separately and makes a prediction by Maximum-a-Posteriori. AHLE [4] and SJE [5] both adopt a bilinear compatibility function to make visual to semantic embedding using hierarchical information. But SJE incorporates 1K-dim GoogleNet and textural features. DS-SJE [112] use deep learning to substitute the embedding function and gives state-of-the-art results. UDA [66] views ZSL as a domain adoption problem. Although their setting is slightly different that uses unlabelled unseen data, we still make a comparison because we use inferred unseen data for classification. For both shallow and deep features, our method achieves significant improvements over all of the published results. It is noticeable that our method only uses attributes as side information. Also, the results of using Nearest Neighbour (NN) classifier are slightly lower than that of using SVM, which is caused by that the inferred features become more discriminative after the orthogonal embedding. However, the data structure can be slightly different to real distribution.

For SUN, DAP is also compared. ZSLwUA [57] considers the unreliability of human-defined attributes and make predictions by random forest. ESEZL [116] combines visual-attribute and attribute-label embedding into one joint function. SSE [155] and JLSE [156] are similarity-based approaches that jointly learn a dictionary learning function for both visual and attribute domains. Note that all of the compared methods use attributes as side information. Using deep features, ESEZL, SSE, and JLSE achieves state-of-the-art results. Our result is only 0.5% lower than that of JLSE. However, using shallow features, our method is 5% higher than other methods. Again, we observe that using SVM can significantly boost the performance, which benefits from using inferred visual features.

Fig. 4.3 (B) depicts the resulting ROC curves of our results on the two datasets. One can see that the performances on all classes are balanced and reasonable.

Analysis To understand how each part of our approach contributes to the overall performance, we also implement a set of baseline methods. We summarise the results in Fig. 4.3 (A). All of the baseline methods are implemented using deep features. The first three baselines examine the conventional visual-attribute embeddings. We train SVM using the attributes of unseen instances. During the test, images are mapped to the attribute space and classified by the trained attribute-SVM. X-A directly learns a mapping from visual features. X-VV'-A is the inverse version of the proposed method, where we insert an intermediate latent embedding spaces with orthogonal constraints. To see the effect of orthogonality, we remove the orthogonal constraint in X-V-A as a reference. Similarly, for the later three methods using attribute-visual embedding, we compare to A-X that directly maps attributes to the visual space without orthogonalised embedding space. A-V-X is implemented by removing the orthogonal constraint in Equation 5.

We observe the orthogonality contributes the most to the overall performance. Also, embedding from attribute to visual space significantly boosts the performance, which verified our statements that, fine-grained classes are more discriminable in the visual space due to the semantic representations are too close. Another conclusion can be made that inserting an intermediate embedding space is helpful to compromise the data structural differences to some extents. Although without orthogonal constraints, the results of X-V-A and A-V-X are higher than that without the \mathbf{V} space.

How many dimensions do we need for \mathbf{V} ? Since orthogonalisation can effectively remove the redundant information, each dimension of the orthogonal space indicate a reliable component. In Fig. 4.5, we show the recognition rates vary with respect to the dimension K of the embedding space for the two datasets. It can be seen that best results are given with K equals to 1500 and 2500 respectively. Since the classes in SUN are more various than that in CUB, higher dimensional \mathbf{V} can give better results in general.

Data Distribution of inferred Visual Features One of the fundamental questions is whether our inferred visual features are close to the real data. In Fig. 4.4, we demonstrate the data distribution of real and inferred visual features using t-SNE. Although t-SNE can result in slight distortion and colour changes, we can still recognise the data structures are preserved. The only difference is that some of the inferred visual features are shown further than the real data. For example, the blue cluster of points at the top in CUB is pulled further by t-SNE, which is because our OSVE can reduce the correlations and make the inferred data more discriminative.

Table 4.3 Results (in %) of Open ZSL 1: add extra seen classes as candidates or add instances from seen classes for testing.

Dataset	#Extra Seen	For Candidate	Add to Test
CUB	50	56.5	51.9
	100	52.7	43.2
	150	47.1	36.8
	0	60.1	
SUN	10	79.98	76.63
	100	74.38	70.47
	300	65.53	59.81
	500	61.72	54.26
	707	58.42	49.59
	0	83.23	

4.4.3 Fine-grained Open Zero-shot Learning

There are two restrictions for conventional ZSL settings that are not realistic. 1) The test images can only come from unseen class. 2) The number of seen class is substantially larger than that of unseen classes. By breaking the restrictions, we investigate two scenarios of open zero-shot learning, both of which widely exist in real-world applications. **Scenario 1:** *Test images come from a mixture of seen and unseen classes.* **Scenario 2:** *Testing by a large number of unseen classes using a small training set.*

For scenario 1, the seen/unseen splits are the same (150/50 for CUB and 707/10 for SUN). But we use half of each seen class for training and the other half for testing. Before the test, we infer the visual features for both seen and unseen test images, using which we train SVM classifiers. In this way, the seen classes are added as candidates, *i.e.* test unseen image now may be misclassified to seen classes. We also add images from seen classes for testing. The potential challenge is that the seen classes may be misclassified into unseen classes. We summarise our results in Table 4.3. We show the results of conventional ZSL (0 extra seen) as references. It can be seen that by testing on the whole datasets (200 classes in CUB and 717 classes in SUN), our method can still lead to acceptable results.

For scenario 2, we investigate how our method can withstand a significant reduction of seen class number and an increasing unseen class number. Our results are summarised in Fig. 4.6. Results using a various size of training sets are shown in different colours of lines. We gradually add remaining classes as unseen classes for testing and see the trend of overall recognition rates. We observe the result on the most extreme splits (10/190) on CUB is only 8% lower than that of 10/50. For SUN, increasing the number of unseen classes from 10 to 100 only result in 15% recognition drop in average. Under the extreme setting on SUN

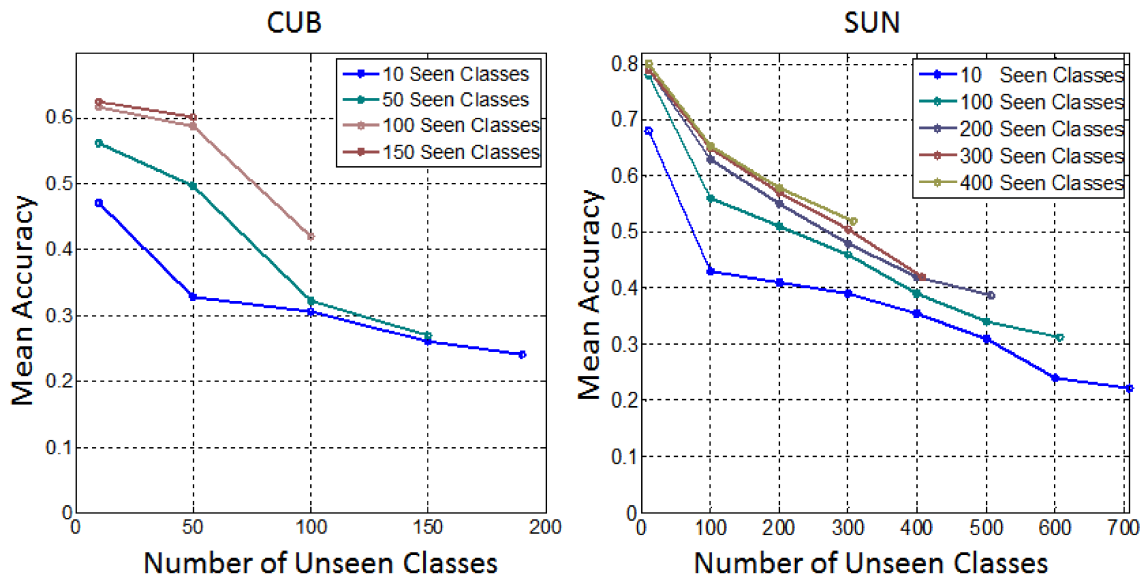


Fig. 4.6 Open ZSL 2: test by increasing number of unseen classes using different size of training sets.

(10/707), we achieve 22.4% recognition rate, where the random guess is only 0.14%.

Qualitative Results As shown in Fig. 5.9, given a query unseen instance, we infer its visual feature and examine what do the original images of the nearest features look like. We compare the results under conventional and extreme open ZSL settings. It can be seen that the tasks are difficult even for humans. The inferred visual features can still retrieve the most visually similar instances.

4.5 Conclusion

In this Chapter, we proposed a novel semantic-visual embedding framework that was inverse to conventional ZSL frameworks. Using inferred visual features, we could convert the ZSL problem into conventional supervised classification and employ powerful classifiers for fine-grained open ZSL. On standard seen/unseen settings, our method achieved significant improvements over the state-of-the-art results. Furthermore, we challenged two scenarios of open ZSL tasks, on both of which our method manifested promising performance. Also, the inferred visual features were shown under the same data distribution as real data. The success of our method owes to the orthogonal embedding space that can jointly compromise the structural differences between visual and attribute spaces and remove the redundant correlations simultaneously.



Fig. 4.7 Top-5 nearest neighbours of the query image under conventional and open ZSL. Correct and incorrect matches are shown in green and red respectively. Corresponding seen/unseen splits are shown on the right.

For future work, our method is helpful to synthesise visual data for rare unseen classes. Our method can also be applied to incremental ZSL frameworks that can mutually infer new attributes and visual data in a large-scale recognition system. In the next chapter, we combine the advantages of graph and orthogonal regularisation for unseen data synthesis.

Chapter 5

Zero-shot Data Synthesis

The last two chapters demonstrate how to utilise spectral graph to remove visual-semantic ambiguities and how to infer unseen visual data using the orthogonal embedding for fine-grained ZIC. This chapter combines these techniques into a unified framework. Comprehensive experiments show that the hybrid model can effectively address most of existing ZIC scenarios, including conventional ZIC, Open ZSL, Generalised ZIC, *etc.* The proposed method steadily outperforms the state-of-the-art methods on all of the considered scenarios.

5.1 Introduction

Zero-shot Learning aims to leverage a closed-set of semantic models that can generalise to an ever growing set of new classes [72, 74, 103, 116]. Since semantic information can be obtained through human knowledge, new classes can be dynamically created without collecting any new visual data. The common paradigm is inspired by that humans can identify new things by just knowing the conceptual descriptions since we could associate the concepts to our previous knowledge. Following such an idea, the first step of ZSL is to train a prediction model that can map visual data to a semantic representation. Hereafter, new categories can be recognised by only knowing their semantic descriptions. Existing ZSL studies fall into two main streams: prediction models and semantic representation designs. The former stream develops advanced models that aim to predict human knowledge accurately from visual data, *e.g.* the probabilistic model DAP and IAP [63, 72, 73]. More recent studies take advantages of an embedding approach as middle layers between low-level features and class labels [4, 40, 44, 80, 101, 116]. Besides, some novel works study how to directly construct classifiers for unseen classes [36, 94, 135]. The latter stream focuses on how to effectively represent human knowledge that can generalise to novel classes, such as human-nameable attributes [37, 58, 72, 104, 119], word vectors [103, 124], textual descriptions [115], and



Fig. 5.1 Given a conceptual description, human can imagine the outline of the scene by combining previous seen visual elements.

class similarities [151, 155].

The methods mentioned above share a common shortage that the training visual examples cannot be expanded while the semantic information is increasing and new classes are added. Since new concepts are ever growing, it is inevitable to collect training data for new semantic models. In this chapter, we propose to synthesise training data for unseen classes. Our idea is inspired by the ability of imagination of human beings. As illustrated in Fig. 5.1, given a semantic description, humans can associate familiar visual elements and then imagine an approximate scene. It is worth noting that our method differs from image synthesis in [74] since the synthesised images from semantics can hardly cover the large variation of visual appearances. Instead, we synthesise discriminative low-level features to train supervised classifiers for ZSL. Such an approach provides a direct interface between ZSL tasks and conventional supervised classifiers. Moreover, it enables the information mutually flow between high-level concepts and low-level visual features. In this way, the training set can be expanded to as large as the semantic representations.

Despite the simplicity of the idea, we confront two main technical issues. The first is the *visual-semantic discrepancy*. Since the visual and semantic features differ in the extracted sources and means, the data distributions of the two data spaces can be significantly discrepant. Two close points in one space can be far away in the other space. For example, as reported in [43], the same attribute ‘HasTail’ may have a great difference between the visual appearances of ‘Zebra’ and ‘Pig’. However, rather than concerning the ‘domain-shift problem’ for the recognition task in [43], instead, we hope the model can effectively capture the semantic-visual correlation so that the synthesised visual data can preserve the intrinsic

structure as close as the real data.

The second issue is the *variance decay*. Due to that the number of visual feature dimensions is usually much larger than that of semantic representations, the learnt projection is prone to be imbalanced, *i.e.* the variances of the projected dimensions vary severely [133]. As shown in Fig. 5.6, comparing to the real data, we observe that the synthesised data using linear projection suffers from remarkable variance decay. The variances of most of the projected dimensions are extremely low, which indicates they gain little information. Such projections can lead to degraded performance owing to the great number of redundant dimensions. Therefore, the challenge is how to make the information diffuse to most of the dimensions of the synthesised data with a balanced projection. To the best of our knowledge, this issue has not been identified in previous ZSL literature.

To address the above issues, we propose a novel embedding algorithm named *Unseen Visual Data Synthesis* (UVDS) that projects semantic features to the high-dimensional visual feature space. In particular, for the first issue, we introduce a latent embedding space to reconcile the structural difference between the visual and semantic spaces. We use the *dual-graph* (GR) in Chapter 3 to preserve the local structure of both visual and semantic spaces. For the second problem, we propose a novel *Diffusion Regularisation* (DR) that explicitly makes the information diffuse to all dimensions of the synthesised data. Specifically, we use the variances as the measurement to force information to diffuse over the dimensions of the synthesised data. We prove that such a scheme is equivalent to finding an orthogonal rotation transformation. Also, we discover an elegant form of such an orthogonal rotation using the $\ell_{2,1}$ norm regularisation with efficient solutions.

In addition to the above two problems, the synthesised data should also be discriminative for the ZSL task. A direct regression model can be viewed to suffer from the *over-fitting problem*, *i.e.* the trained model can achieve high performance on the synthesised data of seen classes but will dramatically degrade on the synthesised unseen data. We empirically show that the above GR and DR can mitigate the over-fitting problem in a complementary manner: DR does not harm the local structure preservation but instead benefits the data synthesis by eliminating the redundant correlations in the semantic space through the orthogonal rotation. The main contributions of this chapter are summarised below:

- An intuitive framework that enables us to synthesise unseen data from the given semantic attributes. The synthesised data can be straightforwardly fed to typical classifiers and lead to the state-of-the-art performance on four benchmark datasets.
- A novel diffusion regularisation that can explicitly make information diffuse to each dimension of the synthesised data. We achieve information diffusion by optimising

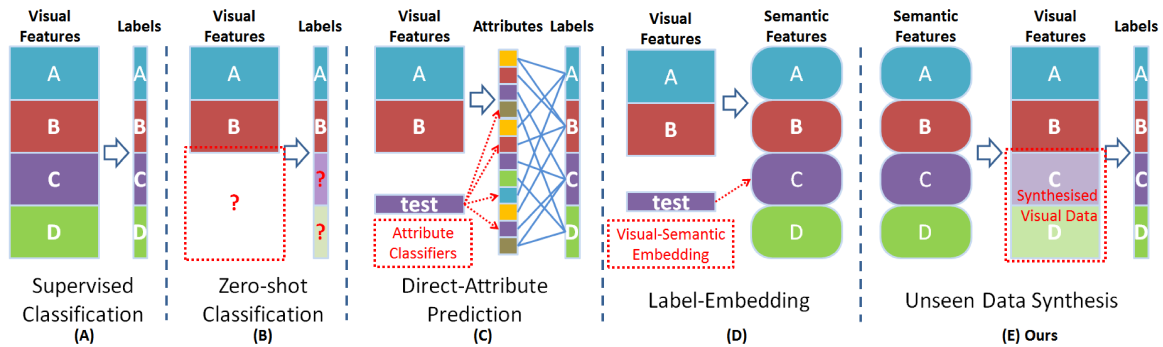


Fig. 5.2 Comparison of supervised and zero-shot classifications and existing ZSL frameworks. (A) a typical supervised classification: the training samples and labels are in pairs; (B) a zero-shot learning problem: without training samples, the classes *C* and *D* cannot be predicted; (C) Direct-Attribute Prediction model uses attributes as intermediate clues to associate visual features to class labels; (D) label-embedding: the attributes are concatenated as a semantic embedding; (E) we use semantic embedding to synthesise unseen visual data.

an orthogonal rotation problem. We provide an efficient optimisation strategy to solve this problem together with the data structural preservation and data reconstruction.

The rest of the chapter is organised as follows. We review existing ZSL methods and related work in Section 2. The proposed algorithm is described in detail in Section 3. The experimental results are demonstrated in Section 4. Finally, we make a conclusion and discuss possible future works in Section 5.

5.2 Related work

Zero-shot Recognition Schemes: We summarise previous ZSL schemes in Fig. 5.2, in contrast to conventional supervised classification (Fig. 5.2(A)). Since collecting well-labelled visual data for novel classes is expensive, as shown in Fig. 5.2(B), zero-shot learning techniques [72, 74, 103] are proposed to recognise novel classes without acquiring the visual data. Most of the early works are based on the Direct-Attribute Prediction (DAP) model [72]. Such a model utilises semantic attributes as intermediate clues. A test sample is classified by each attribute classifier in turn, and the class label is predicted by probabilistic estimation. Admitting the merit of DAP, there are some concerns about its deficiencies. [58] points out that the attributes may correlate to each other resulting in significant information redundancy and poor performance. The human labelling involved in attribute annotation may also be unreliable [57].

To circumvent learning independent attributes, embedding-based ZSL frameworks (Fig. 5.2(C)) are proposed to learn a projection that can map the visual features to all of the

attributes at once. The class label is then inferred in the semantic space using various measurements [4, 6, 44, 79, 95, 124]. Since the attribute vectors are regarded as whole semantic representations, attributes are used for transductive ZSL settings [41, 43, 66, 80, 113, 149]. However, these methods involve the data of unseen classes to learn the model, which to some extent breaches the strict ZSL settings. Recent work [116, 141] combines the embedding-infering procedure into a unified framework and empirically demonstrates better performance. The closest related work is [20], which takes one-step further to synthesise classifiers for unseen classes. Our method is also different from DS-SJE [112], in terms of learning objective, regularisation, and the potential applications. DS-SJE seeks to learn a compatibility function for both images and texts, whereas our objective function aims to reconstruct the visual features from semantic attributes. Also, our method learns with GR and DR that are not considered in DS-SJE. The inferred visual features can be applied to conventional supervised classifiers, which differs our method from other previous work.

Our method takes the advantages of semantic embedding. However, our purpose is entirely different from existing work. As discussed earlier, owing to the fact that the semantic information is ever growing, it is inevitable to collect visual training data for newly added concepts. Since it is easier to obtain semantic information from the Internet, our method can expand the number of visual feature vectors to as many as the semantic instances.

Semantic Side Information: ZSL tasks require to leverage side information as intermediate clues. Such frameworks not only broaden the classification settings but also enable various information to aid visual systems. Since textual sources are relatively easy to obtain, [94, 115] propose to estimate the semantic relatedness of the novel classes from the text. [36, 76, 76] learn pseudo-concepts to associate novel classes using Wikipedia articles. Recently, lexical hierarchies in the ontology engineering are also exploited to find the relationships between classes [5, 7, 114].

Although various side information is studied, attribute-based ZSL methods still gain the most popularity. One reason is that attributes often give prominent classification performance [55, 151, 153, 155, 156]. For another reason, attribute representation is a compact way that can further describe an image by concrete words that are human-understandable [3, 37, 45, 81]. Various types of attributes are proposed to enrich applicable tasks and improve the performance, such as relative attributes [104], class-similarity attributes [151], and augmented attributes [119].

We evaluate our method using attributes and Word2vectors. Since our proposed framework is embedding-based, it can easily exploit most of existing semantic side information.

Structure-Preserving Projection: Structure-preserving projection is well-studied in unsupervised learning [19]. A spectral graph is constructed to preserve the original data structure.

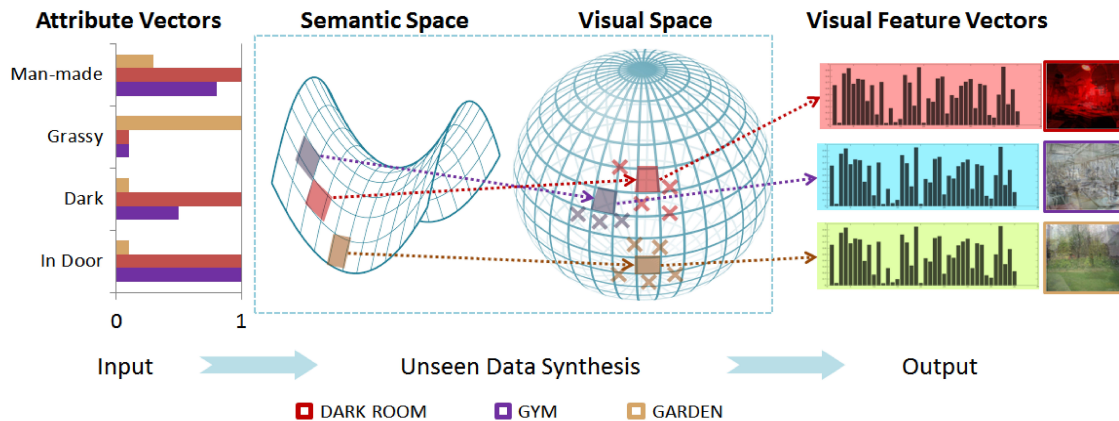


Fig. 5.3 An illustration of our framework of unseen data synthesis. Unseen classes are represented by semantic attributes as inputs. We train a model that maps the semantic space to the visual data space to synthesise training data for these unseen classes. The crosses in the visual spaces denote test feature points.

[159] extends such an idea to multi-view classification to preserve the intrinsic data structures of multiple modalities. The most common approach is to use local neighbourhood graphs for each view independently [41]. [160] generalises a single graph to a multi-graph with random walks between the connections. The graph-based approach is adopted in [43] for transductive ZSL. They estimate the pairwise similarity between training data and unlabelled unseen data using heterogeneous hyper-graphs.

In contrast to these methods, we strictly follow the ZSL setting that excludes data of unseen classes from the training set. Such a setting increases the difficulty since the visual structure of unseen classes can be distinctive from the given semantic data structure. As a solution, we propose to insert a latent embedding space to reconcile the data structure discrepancy between the visual and semantic spaces. A dual-graph is then constructed to find a balanced structure between the two spaces.

Data Rotation for Information Diffusion: Because information diffusion has not aroused attentions in the ZSL field, we discuss related work in a broader context. Data Rotation aims to find a balanced projection that makes the information diffuse to all dimensions of the synthesised data. Such an issue is initially concerned with unsupervised learning methods [49, 67, 83] since imbalanced projection can lead to inferior retrieval performance. In [49], data rotation is adopted to minimise the quantisation error. [67] achieves information diffusion by minimising the reconstruction error of the covariance matrix. [144] uses perfectly diffused data as referencing source to find the rotation so that the projected data can also be well diffused.

We share the consideration of these previous works, yet our proposed method is entirely

different from them. Firstly, our ZSL task is fully supervised. We aim to synthesise visual features rather than finding an optimal subspace of original features. Secondly, none of the previous works utilise variance as measurement and explicitly control the information diffusion. In our experiments, we demonstrate that the synthesised data can achieve more balanced dimensions even comparing to the real data. The improved performance can also prove the effectiveness of our method.

5.3 Approach

ZSL tasks generally involve three steps: training, inference, and test. Some of previous methods may combine inference with training or test. In our framework, the training only requires data of seen classes. The attributes of unseen classes are required at the inference stage to synthesis visual features. Finally, we use the synthesised features for ZSL classification.

5.3.1 Preliminaries

The training set contains visual features, attributes, and seen class labels that are in 3-tuples: $(\mathbf{x}_1, \mathbf{a}_1, y_1), \dots, (\mathbf{x}_N, \mathbf{a}_N, y_N) \subseteq \mathbf{X}_s \times \mathbf{A}_s \times \mathbf{Y}_s$, where N is the number of training samples; $\mathbf{X}_s = [\mathbf{x}_{nd}] \in \mathbb{R}^{N \times D}$ is a D -dimensional feature space; $\mathbf{A}_s = [\mathbf{a}_{nm}] \in \mathbb{R}^{N \times M}$ is an M -dimensional attribute space; and $y_n \in \{1, \dots, C\}$ consists of C discrete class labels. During the test, the given attributes can be either *category-level* or *instance-level*. In our framework, we aim to cope with both of the scenarios using a unified paradigm. Given \hat{N} pairs of unseen instances with semantic attributes from \hat{C} discrete categories: $(\hat{\mathbf{a}}_1, \hat{y}_1), \dots, (\hat{\mathbf{a}}_{\hat{N}}, \hat{y}_{\hat{N}}) \subseteq \mathbf{A}_u \times \mathbf{Y}_u$, where $\mathbf{Y}_u \cap \mathbf{Y}_s = \emptyset$, $\mathbf{A}_u = [\mathbf{a}_{\hat{n}m}] \in \mathbb{R}^{\hat{N} \times M}$, the goal of zero-shot learning is to learn a classifier, $f: \mathbf{X}_u \rightarrow \mathbf{Y}_u$, where the samples in \mathbf{X}_u are completely unavailable during training. We use *Calligraphic* typeface to indicate a space. Subscripts s and u refer to ‘seen’ and ‘unseen’. *hat* denotes the variables that are related to ‘unseen’ samples.

Unseen Visual Data Synthesis: We aim to synthesise the visual features of unseen categories by the given semantic attributes. Specifically, we learn an embedding function on the training set $f': \mathbf{A}_s \rightarrow \mathbf{X}_s$. After that, we are able to infer \mathbf{X}_u through: $\mathbf{X}_u = f'(\mathbf{A}_u)$.

Zero-shot Recognition: Using the synthesised visual features, it can directly estimate the probability distribution of the unseen categories. It is straightforward to employ conventional supervised classifiers, *e.g.* SVM, to predict the labels of unseen classes $f_{\text{SVM}}: \mathbf{X}_u \rightarrow \mathbf{Y}_u$.

5.3.2 Unseen Visual Data Synthesis

Traditional ZSL methods minimise the single classification error of each attribute. Due to that, the attributes are separately learnt, as aforementioned, such a framework highly depends on the quality of the designed attributes. Recently, there is a new scheme that addresses ZSL by an embedding approach [4]. In particular, an objective function is learnt to minimise the multi-class error simultaneously and consider the relationship between different attributes. A typical multi-attributes classifier can be learnt by the following problem:

$$\min_P \mathbb{L}(\mathbf{X}_s P, \mathbf{A}_s) + \lambda \Omega(P), \quad (5.1)$$

where P is the projection matrix, \mathbb{L} is a loss function, and Ω is a regularisation term with its hyper-parameter λ . It is common to choose $\Omega(P) = \|P\|_F^2$. During the test, an unseen instance can be directly mapped to the attribute space by $\hat{a} = \hat{x}P$.

However, due to the fact that P is learnt using only the training data, the inferred attributes \hat{a} are prone to be biased towards the ‘seen’ attributes \mathbf{A}_s . Inspired by the idea that a human can imagine the visual appearance of an unseen object through given semantic descriptions, we propose to synthesise visual features by reversely learning a mapping function from the semantic space to the visual feature space:

$$\min_P \mathbb{L}(\mathbf{A}_s P, \mathbf{X}_s) + \lambda \Omega(P). \quad (5.2)$$

The loss term accounts for the reconstruction error between the semantic input and the visual output; whereas the regularisation ensures the discrimination to unseen classes. Such a framework provides a direct mapping to the visual space without computing a pseudo-inverse matrix and therefore avoids information loss. Before the test, it is straightforward to infer the visual features of unseen classes using their class attributes:

$$\mathbf{X}_u = \mathbf{A}_u P. \quad (5.3)$$

Visual-Semantic Structure Preservation In spite of the simplicity of the above framework, several problems are worth noting. Firstly, in practice, there is often a huge gap between visual and semantic spaces. In pursuance of minimum reconstruction error, the model tends to learn principal components between the two spaces. Consequently, the synthesised data would be not discriminant enough for ZSL purposes. Secondly, such a regression-based framework does not discover the intrinsic topological structure. As a result, the synthesised data may gain an entirely different feature distribution to the original visual features. Thus, directly mapping from semantic to visual space can lead to inferior performance. We

propose to introduce an auxiliary latent-embedding space \mathbf{V} to reconcile the semantic space with the visual feature space, where $\mathbf{V} = [v_{nd}] \in \mathbb{R}^{N \times D}$. In this way, instead of $\Omega(P)$, we can let \mathbf{V} preserve the intrinsic data structural information of both visual and semantic spaces:

$$J = \|\mathbf{X}_s - \mathbf{V}Q\|_F^2 + \|\mathbf{V} - \mathbf{A}_sP\|_F^2 + \lambda\Omega_1(\mathbf{V}), \quad (5.4)$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix, which estimates the Euclidean distance between two matrices. The latent-embedding space \mathbf{V} is decomposed from \mathbf{X} and \mathbf{A} is then decomposed from \mathbf{V} , where $Q = [q_{d'd}] \in \mathbb{R}^{D \times D}$ and $P = [p_{md}] \in \mathbb{R}^{M \times D}$ are two projection matrices. Ω_1 is a *dual-graph* that is introduced next.

In detail, we take the *Local Invariance* [19] assumption and solve the problem through a spectral *Dual-Graph* approach. This is a combination of two supervised graphs that aim to simultaneously estimate the data structures of both \mathbf{X} and \mathbf{A} . The graph of visual space $W_{\mathbf{X}} \in \mathbb{R}^{N \times N}$ has N vertices $\{g_1, \dots, g_N\}$ that correspond to N data points $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ in the training set. The semantic graph $W_{\mathbf{A}} \in \mathbb{R}^{N \times N}$ has the same number of vertices. As mentioned earlier, the attributes for ZSL tasks can be instance-level or category-level. In particular, for *instance-level attributes*, we construct k -nn graphs for both visual and semantic spaces, *i.e.* put an edge between each data point \mathbf{x}_n (or \mathbf{a}_n) and each of its k nearest neighbours. For each pair of the vertices g_i and g_j in the weight matrix (not differ in $W_{\mathbf{X}}$ and $W_{\mathbf{A}}$), $w_{ij} = 1$ if and only if g_i and g_j are connected by an edge, otherwise, $w_{ij} = 0$. As a result, we can separately compute the two weight matrices $W_{\mathbf{X}}$ and $W_{\mathbf{A}}$. It is noteworthy that, for *category-level attributes*, $W_{\mathbf{A}}$ is computed in a slightly different way. Every vertex in the same category is connected by a normalised edge, *i.e.* $w_{ij} = k/n_c$, if and only if \mathbf{a}_i and \mathbf{a}_j are from the same category c , where n_c is the size of category c .

In the embedding space \mathbf{V} , we expect that if g_i and g_j in both graphs are connected, each pair of embedded points v_i and v_j are also close to each other. However, sometimes $W_{\mathbf{X}}$ and $W_{\mathbf{A}}$ are not always consistent due to the visual-semantic gap. To compromise such conflicts, we compute the mean of the visual and attribute graphs, *i.e.* $W_{ij} = \frac{1}{2}(W_{\mathbf{X}_{ij}} + W_{\mathbf{A}_{ij}})$. The resulted regularisation is:

$$\begin{aligned} \Omega_1(\mathbf{V}) &= \frac{1}{2} \sum_{i,j=1}^N \|v_i - v_j\|^2 w_{ij} \\ &= \text{Tr}(\mathbf{V}^T \mathbf{D} \mathbf{V}) - \text{Tr}(\mathbf{V}^T \mathbf{W} \mathbf{V}) = \text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}), \end{aligned} \quad (5.5)$$

where \mathbf{D} is the degree matrix of W , $\mathbf{D}_{ii} = \sum_j w_{ij}$. L is known as graph Laplacian matrix $L = \mathbf{D} - \mathbf{W}$ and $\text{Tr}(\cdot)$ computes the trace of a matrix.

Diffusion Regularisation Another fundamental problem is *Redundant projections*. Com-

pared to the compact attributes, the variance of visual data is usually larger and more informative. However, when we learn visual features from the attributes, in particular when projecting \mathbf{A} to \mathbf{V} using \mathbf{P} , the dimension difference $D \gg M$ will lead the learning algorithm to pick the directions with low variances progressively. As shown in Fig. 5.6, most of the information (variance) is contained in a few projections. As a result, the remaining dimensions of the synthesised data experience a dramatic variance decay, which indicates the learnt representation is severely redundant. To address the problem, we may expect the concentrated information can effectively diffuse to all of the learnt dimensions through an adjustment rotation [59]. Therefore, we modify the rotating matrix \mathbf{Q} in Eq. (5.4). Here, we consider an orthogonal rotation, *i.e.* $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$, since it is easy to show that $\text{Tr}(\mathbf{Q}^T \mathbf{P}^T \mathbf{A}^T \mathbf{A} \mathbf{P} \mathbf{Q}) = \text{Tr}(\mathbf{P}^T \mathbf{A}^T \mathbf{A} \mathbf{P})$. This is an intuitive idea that we can rotate the whole feature space by changing the coordinates through the orthogonal transformation. In this way, the high-variance can diffuse to lower-variance dimensions without changing the overall variance. Such a property is reported in [49] that is known as ITQ, which aims to learn similarity-preserving binary codes. By solving an orthogonal Procrustes problem, the whole feature space is rotated according to the coordinates without changing the structure. Although the values of each dimension are changed, the overall data structure in the semantic \mathbf{A} is completely preserved. Next, we show how the rotation can control variance diffusion.

From Eq. (5.4), the optimal synthesised data is $\mathbf{X} = \mathbf{V}\mathbf{Q}$, where $\mathbf{V} = \mathbf{A}\mathbf{P}$. We first prove that the overall variance does not change after rotation. The attribute data \mathbf{A}_s is centralised, *i.e.* $\sum_{n=1}^N \mathbf{a}_n = \mathbf{0}$. The original variance Γ of \mathbf{V} is $\Gamma = N\sigma_d$, where $\sigma_d = \sum_{n=1}^N v_{nd}^2 / N$ denotes the variance of the d -th dimension. After rotation \mathbf{Q} , we have the new variance of each dimension σ'_d and the sum of variance of each dimension is Γ' . We show $\Gamma = \Gamma'$ in the following:

$$\begin{aligned}
\Gamma &= N \sum_{d=1}^D \sigma_d = \sum_{d=1}^D \sum_{n=1}^N v_{nd}^2 = \|\mathbf{V}\|_F^2 = \text{Tr}(\mathbf{V}\mathbf{V}^T) \\
&= \text{Tr}(\mathbf{V}\mathbf{Q}\mathbf{Q}^T \mathbf{V}^T) = \|\mathbf{V}\mathbf{Q}\|_F^2 \\
&= \sum_{d=1}^D \sum_{n=1}^N x_{nd}^2 = N \sum_{d=1}^D \sigma'_d = \Gamma'.
\end{aligned} \tag{5.6}$$

We hope the overall variance Γ tends to equally diffuse to all of the learnt dimensions in order to recover the real data distribution of \mathbf{X} . In other words, the variance of diffused standard deviations Π in the synthesised data should be small, *i.e.* $\Pi = 1/D \sum_{d=1}^D (\pi_d - \bar{\pi})^2$, where $\pi_d = \sqrt{\sigma'_d}$ and $\bar{\pi}$ is the mean of all standard deviations. According to the above Eq. (5.6), we have ε , *i.e.* $\sum_{d=1}^D \pi_d^2 = \sum_{d=1}^D \sigma'_d = \sum_{d=1}^D \sigma_d = \varepsilon$. Since the sum of standard deviations does not change after rotation Eq. (5.6), minimising the variance of diffused

standard deviations can make high variances diffuse to dimensions with low variances. Next, we show how to minimise Π in our learning framework to find the orthogonal rotation. We first rewrite Π :

$$\begin{aligned}
\Pi &= \frac{1}{D} \sum_{d=1}^D (\pi_d - \bar{\pi})^2 \\
&= \frac{1}{D} \sum_{d=1}^D \pi_d^2 + \bar{\pi}^2 - \frac{2}{D} \sum_{d=1}^D \pi_d \bar{\pi} \\
&= \frac{\varepsilon}{D} - \frac{1}{D^2} \left(\sum_{d=1}^D \pi_d \right)^2.
\end{aligned} \tag{5.7}$$

The first term $\frac{\varepsilon}{D}$ of the above equation is a constant. Thus, the problem of minimising Π is equivalent to maximise the sum of diffused standard deviations in the bracket of the second term of Eq. (5.7). Furthermore, such a maximisation can be converted into the problem of optimising the orthogonal rotation:

$$\begin{aligned}
\sum_{d=1}^D \pi_d &= \sum_{d=1}^D \sqrt{\sigma'_d} = \sum_{d=1}^D \sqrt{\sum_{n=1}^N x_{nd}^2 / N} \\
&= \frac{1}{\sqrt{N}} \|\mathbf{X}^T\|_{2,1} = \frac{1}{\sqrt{N}} \|Q^T \mathbf{V}^T\|_{2,1},
\end{aligned} \tag{5.8}$$

where $\|\cdot\|_{2,1}$ is the $\ell_{2,1}$ norm of a matrix. According to Eq. (5.7) and Eq. (5.8), we can simply maximise $\|Q^T \mathbf{V}^T\|_{2,1}$ to maximise Π with the optimal Q for the purpose of information diffusion. Finally, we can combine the diffusion regularisation with Eq. (5.4) and Eq. (5.5) to form the overall loss function. Such a function aims to minimise the reconstruction error from attributes to visual features, meanwhile preserve the data structure and enable the information to diffuse to all dimensions:

$$\begin{aligned}
\min_{P, Q, \mathbf{V}} J &= \|\mathbf{X}_s - \mathbf{V}Q\|_F^2 + \|\mathbf{V} - \mathbf{A}_s P\|_F^2 + \lambda \text{Tr}(\mathbf{V}^T L \mathbf{V}) \\
&\quad - \beta \|Q^T \mathbf{V}^T\|_{2,1}, \quad s.t. \quad Q Q^T = I.
\end{aligned} \tag{5.9}$$

5.3.3 Optimisation Strategy

The key of our optimisation is to find a proper solution for the latent-embedding space \mathbf{V} . From the above Eq. (5.9), it can be seen that \mathbf{V} simultaneously accounts for the reconstruction error, structure preservation, and diffusion regularisation. However, the problem raised in Eq. (5.9) is a non-convex optimisation problem. To the best of our knowledge, there is no

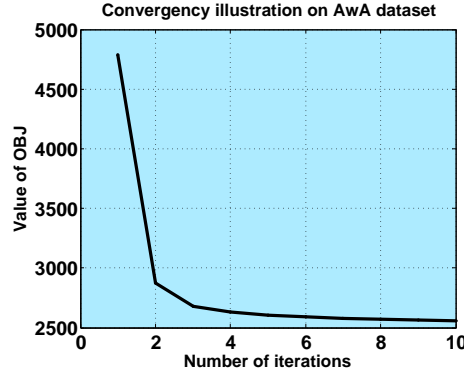


Fig. 5.4 Objective function convergence on the AWA dataset.

direct way to find its optimal solution. Hereby, we propose an iterative scheme by using the alternating optimisation to obtain the local optimal solution. Specifically, we iteratively update \mathbf{V} , \mathbf{Q} , and \mathbf{P} in an alternate manner. In this way, the optimisation becomes analytic and tractable for each variable with the associated sub-problem. It is noted that some variables are first heuristically initialised before our proposed optimisation. Specifically, we initialise $\mathbf{Q} = \mathbf{I}$ and $\mathbf{V} = \mathbf{X}_s$. Such an initialisation equals to start from the simple problem in Eq. (5.2). The initialisation of \mathbf{P} can be obtained via $\mathbf{P} = (\mathbf{A}_s^T \mathbf{A}_s)^{-1} \mathbf{A}_s^T \mathbf{V}$. The whole alternate procedure of the proposed UVDS is listed as follows.

1. V-step: By fixing \mathbf{P} and \mathbf{Q} , we can reduce Eq. (5.9) to the following sub-problem:

$$\min_{\mathbf{V}} \|\mathbf{X}_s - \mathbf{V}\mathbf{Q}\|_F^2 + \|\mathbf{V} - \mathbf{A}_s \mathbf{P}\|_F^2 + \lambda \text{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) - \beta \|\mathbf{Q}^T \mathbf{V}^T\|_{2,1} \quad (5.10)$$

The minimal \mathbf{V} can be obtained by setting the partial derivative of Eq. (5.10) to zero and we have

$$\frac{\partial J}{\partial \mathbf{V}} 2(\mathbf{V}\mathbf{Q} - \mathbf{X})\mathbf{Q}^T + 2(\mathbf{V} - \mathbf{A}\mathbf{P}) + 2\lambda \mathbf{L}\mathbf{V} - \beta \mathbf{V}\mathbf{Q}\mathbf{E}\mathbf{Q}^T = 0, \quad (5.11)$$

where $\mathbf{E} = \text{diag}(e_1, \dots, e_d, \dots, e_D) \in \mathbb{R}^{D \times D}$ and the d -th element of \mathbf{E} is $e_d = 1/(\sqrt{N}\pi_d)$. By merging the like terms, Eq. (5.11) can be rewritten as

$$\mathbf{V}(2\mathbf{Q}\mathbf{Q}^T + 2\alpha\mathbf{I} + \beta\mathbf{Q}\mathbf{E}\mathbf{Q}^T) + (2\lambda\mathbf{L})\mathbf{V} - (\mathbf{X}\mathbf{Q}^T + 2\mathbf{A}\mathbf{P}) = 0, \quad (5.12)$$

which is a typical Sylvester equation so that \mathbf{V} can be efficiently solved by the *lyap()* function in the MATLAB toolbox.

2. Q-step: By fixing \mathbf{P} and \mathbf{V} , we can reduce Eq. (5.9) to the following sub-problem:

$$\min_{\mathbf{Q}} \|\mathbf{X}_s - \mathbf{V}\mathbf{Q}\|_F^2 - \beta \|\mathbf{Q}^T \mathbf{V}^T\|_{2,1}, \quad \text{s.t. } \mathbf{Q}\mathbf{Q}^T = \mathbf{I} \quad (5.13)$$

Since we need to solve Q with the orthogonality constraint in Eq. (5.13), we adopt the gradient flow in [137] which is an iterative scheme that can optimise orthogonal problems with a feasible solution. Such an iterative scheme can minimise Eq. (5.13) until it arrives at a stationary solution. Specifically, given the orthogonal rotation Q_t during the t -th iterative optimisation, a better solution of Q_{t+1} is updated via *Cayley transformation*:

$$Q_{t+1} = H_t Q_t, \quad (5.14)$$

where H_t is the *Cayley transformation* matrix and defined as

$$H_t = \left(I + \frac{\tau}{2} \Phi_t \right)^{-1} \left(I - \frac{\tau}{2} \Phi_t \right), \quad (5.15)$$

where I is the identity matrix, $\Phi_t = \Delta Q_t^T - Q_t \Delta^T$ is the skew-symmetric matrix, τ is an approximate minimiser satisfying Armijo-Wolfe conditions [139] and Δ is the partial derivative of Eq. (5.13) with respect to Q as

$$\Delta_t = \mathbf{V}^T (\mathbf{V} Q_t - \mathbf{X}_s) - \beta \mathbf{V}^T \mathbf{V} Q_t E., \quad (5.16)$$

where the diagonal matrix E is defined the same as that in Eq. (5.11). In this way, for the Q -step, we repeat the above formulation to update Q until achieving convergence. Generally, we set $t = 30$ for Q updating in the Q -step. A similar proof of the updating procedure with the orthogonality constraint can be observed in [137].

3. P-step: By fixing Q and V , we can reduce Eq. (5.9) to the following sub-problem:

$$\min_P \alpha \|\mathbf{V} - \mathbf{A}_s P\|_F^2. \quad (5.17)$$

The resulted equation is derived by a standard least squares problem with the following analytical solution:

$$P = (\mathbf{A}_s^T \mathbf{A}_s)^{-1} \mathbf{A}_s^T \mathbf{V}. \quad (5.18)$$

Note that $(\mathbf{A}_s^T \mathbf{A}_s)^{-1}$ is not always full rank, especially in which all of the instances share the class-level attributes. Therefore, we use Moore-Penrose pseudo inverse of matrix instead. We have so far described our optimisation of each step for Eq. (5.9) in detail. As mentioned above, to obtain a local optimal solution, we adopt an alternate optimisation scheme, in which we repeat t times to solve \mathbf{V} sub-problem, Q sub-problem and P sub-problem in sequence. In our experiments, ten iterations in overall alternate optimisation are proved to be enough for convergence as shown in Fig. 5.4. The proposed UVDS approach is depicted in Algorithm. 3.

Algorithm 3: Unseen Visual Data Synthesis (UVDS)**Input:** Training set $\{\mathbf{X}_s, \mathbf{A}_s, \mathbf{Y}_s\}$, k for k -nn graph**Output:** P , Q and \mathbf{V}

- 1 Initialise $Q = \mathbf{I}$, $\mathbf{V} = \mathbf{X}_s$ and $P = (\mathbf{A}_s^T \mathbf{A}_s)^{-1} \mathbf{A}_s^T \mathbf{V}$, where $\mathbf{I} \in \mathbb{R}^{D \times D}$ is the identity matrix.
- 2 **Repeat**
- 3 **V-Step:** Fix P , Q and update \mathbf{V} using Eq. (5.12).
- 4 **Q-Step:** Fix P , \mathbf{V} and update Q by following steps:
- 5 **for** $t = 1 : \text{max iterations}$ **do**
- 6 Compute the gradient Δ_t using Eq. (5.16);
- 7 Compute the skew-symmetric matrix Φ_t ;
- 8 Compute the Cayley matrix H_t using Eq. (5.15);
- 9 Compute the Q_{t+1} using Eq. (5.14);
- 10 **if** convergence, **break**;
- 11 **end**
- 12 **P-Step:** Fix \mathbf{V} , Q and update P using Eq. (5.18).
- 13 **Until** convergence
- 14 **Return** $f_{UVDS}(x) = xPQ$

Table 5.1 Key statistics of the four datasets.

Dataset	# of attributes	Attribute Type	Annotation Level	# of Seen Classes	# of unseen classes	# of total images
AwA	85	Both	per class	40	10	30475
CUB	312	Binary	Both	150	50	11788
aPY	64	Binary	per image	20	12	15339
SUN	102	Continues	per image	707	10	14340



Fig. 5.5 Some random image and attribute examples of the 4 datasets.

5.3.4 Zero-shot Recognition

Once we obtain the embedding matrices P and Q , the visual features of unseen classes can be easily synthesised from their semantic attributes:

$$\mathbf{X}_u = \mathbf{A}_u P Q. \quad (5.19)$$

It is noticeable that for instance-level attributes, \mathbf{X}_u contains as many instances as the test set. The zero-shot recognition task now becomes a typical classification problem. Thus, any existing supervised classifier, *e.g.* SVM, can be applied to learn a mapping function: $\mathbf{Y}_u = f_{svm}(\mathbf{X}_u)$.

For category-level, only a prototype feature of each category is synthesised. Either few-shot learning techniques or the simplest Nearest Neighbour (NN) algorithm can be adopted: $\hat{y} = \arg \min_i \|\hat{x} - \hat{a}_i P Q\|_2^2$, where \hat{x} is a test image, \hat{a}_i is the class-level attribute vector of the i -th unseen class, and \hat{y} is the final prediction. Since we focus on the quality of the synthesised features, we simply use NN and SVM for instance-level tasks and NN for category-level tasks.

5.4 Experiments

We provide a comprehensive comparison with both classic and recent state-of-the-art methods on four benchmark datasets: Animals with Attributes (AwA) [72], aPascal & aYahoo (aPY) [37], Caltech-UCSD Birds-200-2011 (CUB)[131], and SUN Attribute (SUN) [105]. Key characteristics of these datasets are summarised in Table 5.1. Furthermore, we verify the statements we made in this chapter by comparing to a variety of baselines.

5.4.1 Setup

Settings We strictly follow published seen/unseen splits. For AwA [72] and aPY [37], we follow the standard 40/10 and 20/12 splits like most of existing methods. For CUB, we follow [4] to use the 150/50 setting. For SUN, we use the simple 707/10 setting as reported in [57, 116, 155]. Methods under different settings [20, 43, 113], or using other various semantic information [3, 5, 104, 151] are not compared with.

Semantic Attributes The attribute annotation levels of the four datasets are different. In CUB, aPY, and SUN, each image is annotated by a unique attribute signature. In AwA, all of the images within one class share the same attribute signature. We compute such class-level attributes for aPY and SUN by averaging the image-level attributes for each class.

Table 5.2 Comparison with state-of-the-art methods.

Method	Feature	AI	EP	Animals with Attributes	Caltech-UCSD Birds	aPascal&aYahoo	SUN Attribute
Lampert <i>et al.</i> [72]	V	CA	PC	57.23	-	38.16	72.00
Akata <i>et al.</i> [5]	G	CA	PC	66.7	50.1	-	-
Romera-Paredes and Torr [116]	V	CA	PC	75.32±2.28	-	24.22±2.89	82.10±0.32
Zhang and Saligrama [155]	V	CA	PI	76.33±0.83	30.41±0.20	46.23±0.53	82.50±1.32
Zhang and Saligrama [156]	V	CA	PI	80.46±0.53	42.11±0.55	50.35±2.97	83.83±0.29
GAN	V	CA	PC	62.40±0.85	40.52±0.95	24.28±0.44	68.85±0.72
Ours+CA	V	CA	PC	82.12±0.12	44.90±0.88	42.25±0.54	80.50±0.75
Ours+SVM	V	IA	PC	-	45.72±1.23	53.21±0.62	86.50±1.75
Lampert <i>et al.</i> [72]	V	W2V	PC	42.82±0.81	24.52±0.68	24.52±0.28	65.28±0.57
Akata <i>et al.</i> [5]	V	W2V	PC	56.25±0.74	30.28±0.56	29.28±0.86	70.70±0.65
Romera-Paredes and Torr [116]	V	W2V	PC	58.29±0.58	28.47±0.76	32.67±0.58	72.65±0.78
Zhang and Saligrama [155]	V	W2V	PC	57.49±1.82	29.68±0.84	34.95±1.47	74.19±0.83
GAN	V	W2V	PC	48.34±0.69	25.33±0.82	27.48±0.74	68.58±0.89
Ours+CA	V	W2V	PC	62.88±0.76	32.14±0.47	35.82±0.45	76.98±0.46

Feature: VGG-19 (V) and GoogLeNet-1K (G); Auxiliary Information (AI): Class-level Attributes (CA), Instance-level Attributes (IA), and Word2Vec (W2V); Evaluation Protocol (EP): Per-class accuracy (PC) and Per-image accuracy (PI).

Table 5.3 Detailed analysis of key aspects of the proposed method.

Scenario	Dataset	CUB				SUN				aPY			
	Test Domain	Seen		Unseen		Seen		Unseen		Seen		Unseen	
Prototype-based	Baseline	CA	MF	CA	MF	CA	MF	CA	MF	CA	MF	CA	MF
	Linear Regression	66.82	64.34	27.28	30.31	88.85	89.12	63.00	64.50	52.42	55.35	17.96	21.63
	GR-only ($\beta = 0$)	65.79	65.53	38.82	40.42	89.67	88.41	75.50	76.00	59.38	57.75	25.75	28.86
	DR-only ($\lambda = 0$)	66.32	67.98	37.75	40.64	90.31	89.85	74.00	77.50	57.96	58.32	30.28	32.46
	Ours	67.47	68.43	44.90	44.90	92.32	89.88	80.50	78.50	62.75	64.88	42.25	41.97
Sample-based	Baseline	NN	SVM	NN	SVM	NN	SVM	NN	SVM	NN	SVM	NN	SVM
	Linear Regression	64.57	67.44	22.36	26.57	90.79	92.27	72.50	77.00	43.75	44.42	13.48	15.96
	GR-only ($\beta = 0$)	61.38	66.88	32.65	38.58	88.42	91.91	74.50	80.00	53.34	57.08	22.74	25.59
	DR-only ($\lambda = 0$)	62.44	68.94	36.93	42.24	88.34	90.47	78.00	84.00	55.05	53.41	23.68	24.22
	Ours	63.78	70.32	39.82	45.72	89.85	93.23	78.50	86.50	54.35	69.75	38.49	53.21

CA: Class-level attributes, MF: Mean of synthesised features, GR: Graph regularisation, and DR: Diffusion regularisation. Best results are in bold.

Yet, it is impossible to get the image-level attribute descriptions for AWA. The resultant class-level attributes for the four datasets are in real numbers, whereas the image-level attribute signatures of CUB, aPY, and SUN are binary. We also implement evaluations using Word2Vec features as the auxiliary information. Each class name is encoded into a vector as the class-level semantic representation.

Visual Features The adopted visual features of existing methods mainly differ in deep models. Since most of previous methods are based on the 4096-dimensional CNN features extracted by [155] for the four datasets using the “Image-net-vgg-verydeep-19” model [122], most of our evaluations are based on the same model.

Implementation Parameters Half of the data in each class in the training sets are used as the validation set. We use 10-fold cross-validation to obtain the optimal hyper-parameters λ and β . k is fixed to 10 for the k -nn graph.

5.4.2 Comparison with the State-of-the-art methods

Table 5.2 summarises our comparison to the published results of state-of-the-art methods on the benchmark datasets. The hyphens indicate that the compared methods were not tested on the corresponding datasets in the original papers. The comparisons are mainly divided into two sections. In the first section, all of the compared methods were tested using human-annotated attributes. In the second section, W2V class-label embeddings [97] are employed as the class-level semantic features. We implement the state-of-the-art methods using their published codes. For all of the four datasets, we first evaluate our method using class-level attributes. In this scenario, each unseen class gains a synthesised visual feature prototype from the class attribute signature. The test unseen images are predicted by the NN classification using these prototypes. When image-level attributes are available in CUB, aPY, and SUN, we further conduct experiments using SVM classifiers. The visual feature vector of each unseen image is synthesised by the proposed UVDS and then fed to train SVM models. During the test, visual features that are extracted from the unseen image are fed to the trained SVM to get the prediction.

In the first section, our method outperforms published results on three of the four datasets. Note that on aPY, using synthesised instance-level features with SVM provides a significant performance boost. The evidence can also be found on SUN. This is because that on aPY and SUN, the class-level attributes may not well conclude the features of all of the instances in each class, *e.g.* different style of room. Thus, the individualised synthesised visual features with the SVM classifier can make significant improvement. However, using finer-defined attributes, such as on AWA and CUB, CA can also results in similar performance to that of using instance-level features with SVM. In the second section, the performance based on W2V degraded severely due to the coarse description of the class labels. Our method achieves the best results on all of the datasets. The success can be considered from two aspects. Firstly, although the W2V feature space is heterogeneous to the visual space, our GR can adjust the synthesised features to mitigate such a difference. Secondly, from the Fig. (5.7) can be seen, the synthesised features are more discriminative than the real visual features, which can withstand some performance degradation.

In addition, we also consider the recent Generative Adversarial Network (GAN) as a comparable baseline, which can also synthesise unseen visual features using attributes. We adopt the conditional GAN framework to synthesise features that are sensitive to different class labels. The model is trained as follows. Firstly, semantic representations (attributes or W2V) with noise are used to generate visual features (GN) that are classified as ‘real’ by the discriminative network (DN). Secondly, the synthesised with real visual features are used to train two networks, which are DN and a classification network (CN), respectively. DN

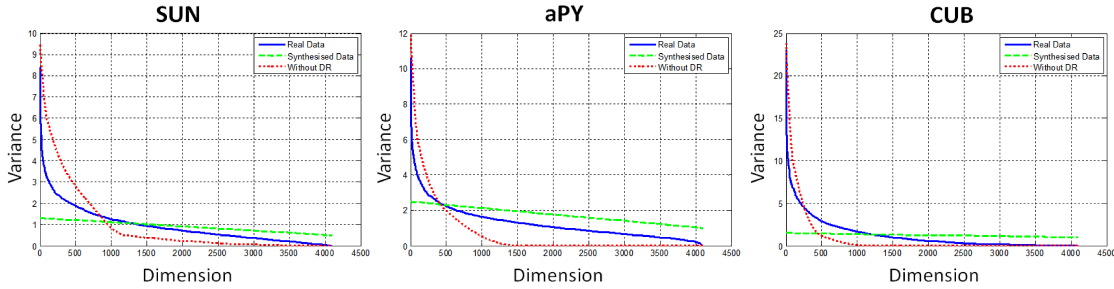


Fig. 5.6 Normalised variances of the synthesised data *w.r.t.* dimensions. Variance of each dimension is sorted in descending order. We make a comparison between the synthesised data variances ‘with’ (green) and ‘without’ (red) diffusion regularisation. The variances of real data (blue) are computed from real unseen data as references.

accounts to differ the real and false features while CN makes the features are discriminative to the corresponding classes. The loss of DN and CN are added together. We repeat the above two step until convergence. During the inference, we input the class-level semantic representations (attributes or W2V) of unseen classes to generate unseen visual features. We then use the generated samples to train SVM to classify unseen instances at the test phase. However, due to the sizes of the datasets are not very large, the results of GAN are inferior to conventional methods. Therefore, how to apply GAN on ZSL task requires further investigation.

5.4.3 Detailed Evaluations

To further understand the success of our UVDS algorithm and verify our statements that are made above, we compare to variations of our methods as baselines under different scenarios. Since AwA only provides class-level attributes, we conduct the remaining experiments on CUB, SUN, and aPY.

Baseline methods The primary purpose of our comparison is to understand the effect of each term in Eq. (5.9). The first baseline method is simply *Linear Regression* ($\beta = 0, \lambda = 0$) that we solve Eq. (5.2) and synthesise prototypes of unseen classes using Eq. (5.3). The second and third methods are denoted as *Graph-Regularisation (GR) only* ($\beta = 0$) and *Diffusion-Regularisation (DR) only* ($\lambda = 0$). In this pair of comparison, we aim to study the characters of each term and how they contribute to the overall performance. For both of the methods, we use the same cross-validation as our proposed method to tune λ and β . In order to discuss the over-fitting problem, we also use the validation set as test for seen classes.

Since existing zero-shot learning methods differ in the annotation level of the semantic

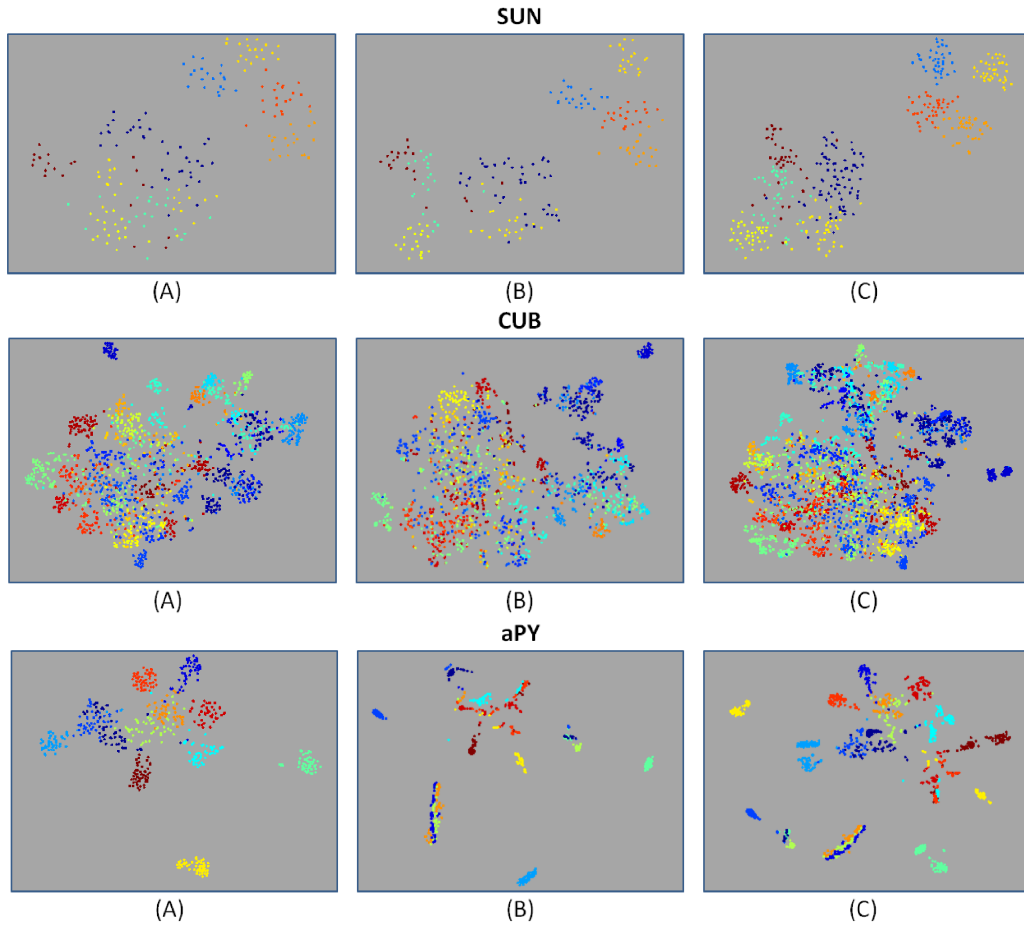


Fig. 5.7 T-SNE of the real and synthesised visual features of unseen classes: (A) real visual features; (B) synthesised visual features; (C) Since t -SNE of different data is not aligned, we also show the distribution of mixed real and synthesised visual features.

attributes, we also investigate how such scenarios can affect the performance. The first scenario is *prototype-based*, *i.e.* each unseen class gains only one visual prototype. There are two possible ways to obtain the class-level prototype: (1) we can compute the mean of image-level attributes in each class and use the averaged class-level attributes (CA) to synthesise one visual prototype for each class; (2) we can first synthesise the visual features from the image-level attributes and use the mean of the features (MF) as the class prototype. During the test, we use NN classification to predict the label for the test image. The second scenario is *sample-based*, *i.e.* each unseen image has one unique attribute description. In this scenario, we can fully synthesise all of the visual features of unseen classes and use them as training examples. We show how an advanced classifier, *e.g.* SVM, can further boost the performance. We summarise the results of our self-comparison in Table 5.3. Based on the outcomes, we can verify the following statements that are made in the context above.

Generalisation to Unseen Data From Table 5.3, we can see that linear regression can

achieve acceptable performance when tested on seen classes. On two datasets, CUB and SUN, the synthesised visual features by the linear regression method are even better than the comparative methods using simple NN classifiers. However, a remarkable drop of recognition rates (32.21% on CUB and 18.29% on SUN) can be found when tested on unseen classes. In average, the performance degradation of unseen class recognition using the linear regression method is about 20%. This is a typical over-fitting problem since we tune the best parameters on the seen set but the trained model cannot well generalise to unseen classes. In comparison, the proposed method can achieve the best performance in most of the situations. Meanwhile, the proposed method can also smoothly generalise to unseen classes. In the case of the SUN dataset, the recognition rate of unseen classes using the SVM classifier is only 3.38% lower than the MF scenario on seen classes (89.88%). The other two baseline methods GR-only and DR-only achieve similar performances on the seen classes and once is higher than the proposed method (55.05% of DR-only on aPY using NN classifier). On unseen classes, the two baseline methods are all better than linear regression without regularisation but lower than the proposed method using both regularisations. Such results suggest that the proposed method can significantly eliminate the bias to the seen training data.

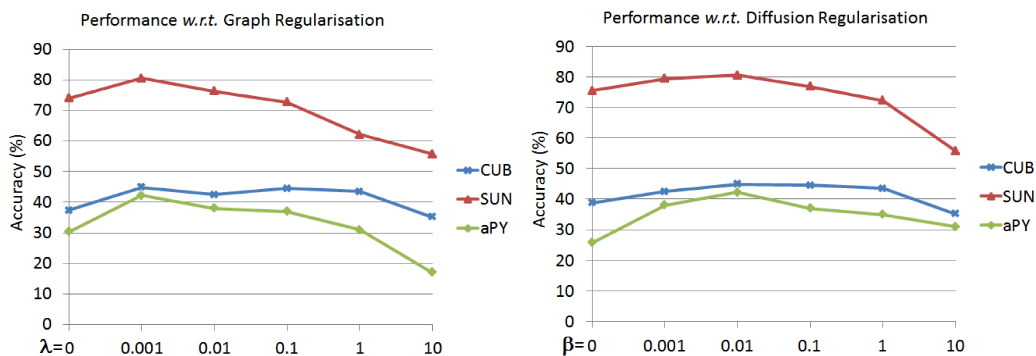


Fig. 5.8 The performance with respect to the Graph regularisation and Diffusion regularisation. The results are under the scenario of CA and using NN classifier.

Effect of Regularisations In Table 5.3, we can see both of the regularisations can significantly boost the performance comparing to the linear regression method. In most cases, the DR-only method is slightly better than the GR-only method. This suggests the importance of the balanced features. Also, we observe the performance of using both of the regularisations is always better than using one of them on the unseen set. To further understand the relationships between GR and DR, in Fig. 5.8, we fix $\lambda = 0.001$ and show the performance varies with β . In turn, we fix $\beta = 0.1$ to see the trend of performance with respect to λ . It can be seen that in most cases, adding the other regularisation can benefit the performance

Class Label	Success Cases						Failure Cases					
Flea Market												
Shoe Shop												
Lab&Classroom												
Donkey												
Centaur												
Bag												
Brandt Cormorant												
Pacific Loon												
Pomarine Jaeger												

 Test Image
 Matched Instance
 Mismatched Instance

Fig. 5.9 Success and Failure cases of nearest neighbour matching. The query visual feature is synthesised from its attribute description. We find top-5 nearest neighbours of the query feature from the real instances. It is a match if the nearest instance and the test image have the same label.

(compared to the case of $\beta = 0$ or $\lambda = 0$ at the beginning of the curve). The exception is only when the other regularisation is over-weighted, *e.g.* $\lambda = 10$. Such a result indicates the two regularisations are not redundant but well complementary to each other.

Class-level attributes or Mean of Features In the case that only class-level attributes are provided, there is no other optional scenario. However, if the provided attributes are image-level, we could use the mean of the attributes for each class to compute prototypes (CA). Alternatively, we could synthesise visual feature for each image first and then compute the mean of the features for each class (MF). When comparing these two scenarios in Table 5.3, interestingly, the performance difference between the two methods is insignificant. The results of MF on the aPY dataset tend to be better than those of CA, whereas, on the SUN dataset, the results of CA are slightly higher than those of MF. We assume the potential reason is due to the quality of the attribute annotations since the attributes in aPY are reported

not very reliable [57]. Such results also show the positive side of our method that we could confidently use the class-level attributes even though there are no image-level attributes available, *e.g.* the AWA dataset.

Advance of using SVM One encouraging reason for synthesising unseen data is to be used for training supervised classifiers. In Table 5.3, the performance of using NN classification under the sample-based scenario is somewhat worse than that under the prototype-based scenarios (CA and MF). After using SVM classifiers, the performance is remarkably boosted and achieves the highest ZSL recognition rate among all of the scenarios. This is a promising result that substantially demonstrates the advantage of using synthesised training data for advanced classifiers.

5.4.4 Further Discussions

This section mainly investigates three key aspects of the proposed method: (1) what are the outcomes of the diffusion regularisation? (2) what kind of visual features are synthesised? and (3) how is the performance on other ZSL scenarios, *e.g.* Generalised and large-scale ZSL? We answer these questions based on the following experimental analysis.

In Fig. 5.6, we show the variance of each dimension of the synthesised data. The variances are sorted in descending order. We compare with the real unseen data and the synthesised data without diffusion regularisation ($\beta = 0$). It is noticeable that, in the synthesised data without DR, most variances are concentrated in a few dimensions (roughly 1000, 1500, and 500 on SUN, aPY, and CUB) while most of the remaining dimensions gain very low variances. In comparison, the variances of our proposed synthesised data and real data are more informative. Furthermore, thanks to the DR, the variances in our proposed data are even more balanced than real data. In other words, each of the dimension gains the equal amount of information. Such quantitative evidence explains the success of our proposed method in the ZSL recognition task.

In Fig.(5.9), we provide some qualitative results of our method. We use the synthesised features as queries and retrieve real images from the unseen datasets. In Fig. 5.9, we show some success cases that most of the top-5 results are with the same class labels. Particularly, the third result of *Bag* is the same paired image of the attributes that are used to synthesise the data. Such results demonstrate that the synthesised data is close to the samples from the same class in the feature space. On the contrary, we also provide some failure cases that the top-1 retrieval result is not with the same class label. Some of them are due to the ambiguity of the semantic meaning, *e.g.* the *flea market* has many similar attributes to the *shoe shop*. Some other cases, *e.g.* the CUB dataset, the real data of the birds are not distinctive to the other classes. Therefore, the NN-based retrieval gives a mixture of true-positives and false-

positives. Such failures due to the ambiguity of the visual feature are not common cases. We can still achieve 45.72% overall recognition rate on the CUB dataset.

Fig.(5.7) shows the distribution of the synthesised (B) and real features (A) of the unseen classes using t-SNE. On SUN and CUB, after mixing both of the features together (C), most classes are discriminative, which means the synthesised features capture the same distribution of the real unseen classes. On aPY, however, the synthesised features look more discriminative than the real features. This can be ascribed to the orthogonal constraint that makes the structure-preserving of the graph constraint sacrifice for the performance. After mixing the real and synthesised features together, intraclass points can be easily discriminated, which supports the effectiveness of the synthesised features.

Finally, we evaluate our method under Generalise ZSL (GZSL) scenarios (Table 5.4) and on large scale datasets (Table 5.5) using the class-level attributes (CA). For the former one, we investigate the four scenarios proposed in [22]: $U-U$ is the conventional unseen-to-unseen ZSL; $S-S$ is the traditional supervised classification; $U-T$ and $S-T$ are two types of GZSL that evaluate whether learnt unseen/seen models are confused to each other. On AWA, our method outperforms the state-of-the-art methods on three of the four scenarios. Only on $S-T$ our result is slightly lower than that of [22]. The seen/unseen balance can be viewed as an over-fitting problem: while we sacrifice the performance on seen classes ($S-T$), the performance on GZSL on unseen classes $U-T$ is significantly boosted. The evidence can also be found on CUB dataset. Although our model performs slightly worse on the seen classes, a better trade-off is achieved, which results in 6.2% performance gain on the $U-T$ scenario on CUB.

Table 5.4 Comparison with published results on GZSL.

Method	AwA				CUB			
	U-U	S-S	U-T	S-T	U-S	S-S	U-T	S-T
DAP[72]	51.1	78.5	2.4	77.9	38.8	56.0	4.0	55.1
IAP[72]	56.3	77.3	1.7	76.8	36.5	69.6	1.0	69.4
ConSE[101]	63.7	76.9	9.5	75.9	35.8	70.5	1.8	69.9
SynC[22]	73.4	81.0	0.4	81.0	54.4	73.0	13.2	72.0
Ours	82.1	93.3	15.8	79.6	44.90	68.2	19.4	66.5

For the large scale ZSL, we follow the settings of [142] on the ImageNet dataset. We extracted the same VGG-19 features as that for the four ZSL benchmarks. For class-level attributes, we use the W2V features provided by [20]. Our method consistently outperforms the published results, from which we can see the prominence synthesised features. However, there is still a large room for improvements. We argue that, for most of ZSL scenarios, the number of unseen classes should be at least smaller than that of training classes. Such

inverted ZSL with significantly larger number of test classes requires reconsideration of the framework. One possible way is to incrementally synthesise unseen visual features and then fine-tune the model using both real and synthesised features like a semi-supervised learning framework.

Table 5.5 Comparison with published results on the ImageNet Dataset.

Method	Hierarchy		Most Populated			Least Populated			AH
	2H	3H	500	1K	5K	500	1K	5k	20K
ConSE[101]	7.63	2.18	12.33	8.31	3.22	3.53	2.69	1.05	0.95
DEWISE[40]	5.25	1.29	10.36	6.68	1.94	4.23	2.86	0.78	0.49
SJE[5]	5.31	1.33	9.88	6.53	1.99	4.93	2.93	0.78	0.52
ESZSL[116]	6.35	1.51	11.91	7.69	2.34	4.50	3.23	0.94	0.62
SYNC[20]	9.26	2.29	15.83	10.75	3.42	5.83	3.52	1.26	0.96
Ours	10.15	2.47	15.96	11.28	4.12	6.06	3.74	1.49	1.02

5.5 Conclusion

In this chapter, we proposed a novel algorithm that synthesises visual data for unseen classes using semantic attributes. The attributes are regarded as a full representation and embedded into the visual feature space. From the experiments, we can see that directly embedding using regression-based models can lead to low zero-shot recognition rates. We ascribed such direct synthesised data to three problems, in terms of imbalanced variances, overfitting, and indiscrimination. In correspondence, we introduced a latent structure-preserving space with the diffusion regularisation as the objective function. As a result, we observed that the proposed algorithm could significantly benefit the performance on unseen class recognition. Our approach outperformed the state-of-the-art methods on all of the four benchmark datasets.

For future work, a worthy attempt is to synthesise instance-level features so that the SVM-based framework can be widely applied. For another, our qualitative experiments give positive results since we have shown the synthesised features are close to the real features in the same class. In the future, the synthesised data can be leveraged for more applications such as image retrieval or unseen image reconstruction. Also, how to address the inverted ZSL with larger number of test classes requires further investigation.

Chapter 6

Beyond Explicit Attributes

6.1 Introduction

Zero-shot recognition is an attractive new task that has recently aroused increasing attentions [7, 72, 103, 111, 116]. It has made it possible to recognise a new category without acquiring training examples beforehand. Compared to traditional methods, zero-shot techniques leverage intermediate semantic models that are shareable to both seen and unseen classes. Such a technique can have wide real-world applications. First, we can now recognise many novel categories for which the visual instances are difficult to be obtained. For example, one may wish to recognise rare animals using only textual descriptions in the book. Second, in the big-data era, the number of required target categories can be enormous. Zero-shot learning (ZSL) can effectively alleviate the burden of collecting training data. Third, for many traditional methods, it is inevitable to retrain the whole model again when we need to add new categories. In zero-shot approaches, the trained model can be shareable for any newly added categories so as to avoid re-training.

One of the fundamental premises for existing ZSL frameworks is the effectiveness of the semantic models. Previous methods [20, 57, 72, 116] widely adopt human nameable attributes as the semantic representations and demonstrate promising results. However, using human nameable attributes can also suffer from several problems. Firstly, deciding an attribute list for ZSL is an ambiguous task. It is easy to consider some visual semantic groups, such as *colours*, *textures* and *parts*. However, more complex attributes, *e.g.* some intangible visual effects, can be hardly described by specific words. Secondly, the designed attribute list is not guaranteed to be discriminative for ZSL. For one thing, semantic attributes may not be visually describable, *e.g.* *domestic* and *carnivorous* in the AWA dataset. Consequently, we can hardly find a converged model for such attributes due to the large variety of visual patterns. Another common issue is known as the *correlation problem* [58].

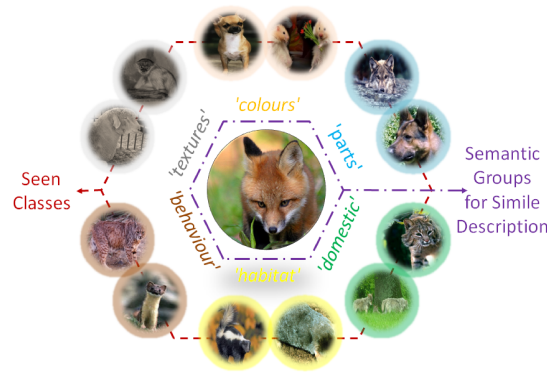


Fig. 6.1 A new class can be described by similes of seen classes without extra attribute concepts involved. We use semantic grouping to make the similes more discriminative. Similes are more natural to describe complex concepts, *e.g. behaviour* or *domestic*.

Namely, different attributes can be highly correlated to each other and are always present or absent together among the whole training set. It then becomes impossible to differentiate these attributes from each other since they share the same positive and negative samples.

Simile is a figure of speech that directly compares two exemplars. In this paper, we propose to use similes instead of explicit attributes. Our idea is motivated by [70] that makes use of similes to describe human faces, *e.g. the glasses on the query face looks like Harry potter's*. However, only similes are not competent for ZSL tasks due to the number of *seen* classes is limited compared to faces. Therefore, we go one step further: we propose a novel graph-cut algorithm that can discover the shared attributes possessed by the similes of exemplars without explicit names. We call such attributes *Implicit Attributes*. Furthermore, to achieve more discriminative semantic models for ZSL tasks, our similes are under different semantic groups, *i.e.* from various aspects such as *colours*, *shapes* and *parts*. We propose a unified framework named Grouped Simile Ensemble (GSE) that can recognise unseen objects by an ensemble model of simile groups. Our method aims to automatically balance the weights between different simile groups, just like we humans can easily find more important attributes to distinguish things. For example, it is easier to differentiate a *panda* from a *bear* by *colours* rather than *shapes*.

Our framework can be briefly summarised as follows. Firstly, we manually annotate both seen and unseen categories by similes under different groups, from which we can discover the implicit attributes by our graph-cut algorithm. We then train our GSE model using training images and the discovered implicit attributes. During the test, our GSE model can find the most important attributes to make predictions for unseen classes. We claim four desired properties of the proposed framework:

- Similes do not involve many additional concepts like explicit attributes. Only the names of seen classes are used. Also, a simile is visually representable by exemplars.

It is natural to describe complex visual appearances by the similarities to training exemplars.

- Our graph-cut algorithm is aware of how many implicit attributes exist in the similes. Each attribute is trained by non-overlapped exemplars to prevent the correlation problem.
- Our GSE model can automatically weigh the significance of different simile groups during the test. On two benchmark datasets, our method achieves state-of-the-art ZSL recognition performance.

The remaining paper is arranged as follows: in Section 2, we review related zero-shot methods; in Section 3, we illustrate our framework and derive the formulations of our ensemble model; we provide extensive evaluations in Section 4; finally, we conclude our findings in Section 5.

6.2 Related Work

Zero-shot learning frameworks The key technique of ZSL is to find an intermediate clue that can generalise to unseen classes. Larochelle *et al.* [74] propose a template-based framework that can depict new classes by manually defined templates. Recently, learning visual attributes [39, 104] gains popularity. In [72], attribute classification is utilised as a mid-level task. During the test, the posterior probability of each attribute is estimated separately by pre-trained classifiers; and the final prediction is made by Maximum a Posteriori (MAP) criteria. Since attribute classifiers are trained separately, such frameworks suffer from the correlation problem [58] and unreliable annotations [57]. In [4], Akata *et al.* propose an embedding-based framework that regards all of the defined attributes as a whole representation. Many recent approaches adopt such an embedding manner and achieve promising results [6, 20, 41, 43, 66, 84, 116, 152]. Besides, similarity-based frameworks also adopt the embedding approach [20, 85, 87, 118, 155, 156]. But the semantic space aims to associate unseen to seen classes. Although these methods have empirically shown improved performance, their embeddings are not human-understandable like the attribute-based methods, *e.g.* they cannot tell which attribute makes the recognition failure like [37]. In comparison to existing methods, our method adopts the advantages of using embedding approaches that can effectively map visual features to the semantic spaces. Furthermore, our embeddings are also interpretable since each simile group has an explicit meaning.

Variations of Semantic information ZSL recognition relies on how to represent unseen classes by prior human knowledge accurately. The representation must be **1)** generalisable,

i.e. the trained model on seen classes is also effective on unseen classes; **2)** visual-related, the gap between the semantic and visual spaces should be small enough to train a stable model. According to these requirements, learning visual attributes has gain most popularity [20, 42, 55, 72, 96, 104, 151]. However, attribute annotations are very expensive, especially for image-level tasks. Also, the involved attributes in the list require careful design. Different datasets often cannot share the learnt attribute models. Such issues make using attributes impractical. As a low-cost solution, text-based semantic features is proposed [36, 94, 115, 150]. However, the textual description from the Internet can be noisy and not directly related to the visual appearance. Another mainstream of semantic representations is similarity-based. Class-wise similarities can be obtained by either human annotators [70, 151] or based on the textual descriptions [155, 156]. Our simile description also shares the idea of similarity comparison. However, none of the existing methods make use of grouping so that the similes can be precisely interpreted. Furthermore, we require the annotators try to make similes based on the visual appearance rather than the semantics so that the visual-semantic gap can be mitigated. Although similes can also be achieved by semantic similarities, such as [114, 115], the accuracy suffers from the semantic-visual gap and therefore is not comparable to our direct similes from human annotators.

6.3 Approach

We first introduce how to annotate classes by similes. Then we formalise the whole framework. The first step of our approach is to discover the implicit attributes from the similes a graph-cut algorithm. Our second step is to train a robust GSE model. Finally, we show how to make predictions using the GSE model during the ZSL test.

6.3.1 Simile Annotation

We aim to annotate both *seen* and *unseen* classes by similes of *seen* exemplars. We illustrate the annotation process in Fig. 6.2. For each target class under annotating, we ask the annotator first meditate its visual appearance from a semantic aspect for ten seconds, *e.g.* *colour, parts, or, shape*. Afterwards, our program starts to flash random exemplars from different *seen* classes, ten images per time. The annotator is asked to choose the most similar exemplars. We accumulate the choices and find the top k most similar classes. Such a process is repeated for all classes under different simile groups. In average, we present ten exemplars from each *seen* class. Key statistics of our simile annotation is summarised in Table 6.1.

Table 6.1 Statistics of simile annotation on AwA and aPY datasets.

items	AwA	aPY
Number of Classes	50	32
Number of Simile Groups	9	5
Number of Images per Flash	10	10
Average Annotating Time	2.5 hours	1 hour

6.3.2 Preliminary

Problem: The training set is in pairs of samples and labels: $(x_1, y_1), \dots, (x_N, y_N) \subseteq \mathbf{X} \times \mathbf{Y}$, where \mathbf{X} is an arbitrary feature space and $y_n \in \{1, \dots, C\}$ consists of C discrete categories. In the test domain, only names of L *unseen* classes are provided without any instances, i.e. $\mathbf{Z} = \{z_1, \dots, z_L\}$. The goal is to learn a classifier, $f : \mathbf{X} \rightarrow \mathbf{Z}$. It is noticeable that $\mathbf{Z} \cap \mathbf{Y} = \emptyset$. Such a problem is known as the Zero-shot classification.

Discovering implicit attributes from similes: After simile annotation in Section 6.3.1, any class $j \in \mathbf{Y} \cup \mathbf{Z}$ can be interpreted by a set of similarity-based exemplars from the training set, i.e. $\mathbf{NN}_j \in \mathbf{Y}$, which can form an undirected graph. Using graph-cut, we can discover what are the implicit attributes that make the classes similar to each other. This is conducted under G different simile groups. For each group: $f_1^{(g)} : \mathbf{NN}^{(g)} \rightarrow \mathbf{A}^{(g)}$. As a result, each category gains an attribute signature in each simile group: $\mathbf{A}_j^{(g)} = (a_1, \dots, a_{m_g}) \in \mathbb{R}^{m_g}$, where $j \in \mathbf{Y} \cup \mathbf{Z}$, and m_g is the total number of discovered implicit attributes.

Base feature extraction and GSE: Low-level features are extracted and concatenated to form a base visual space. We train ensemble models for different simile groups. Each model aims to embed the visual features from *seen* classes to their corresponding implicit attribute space: $f_2^{(g)} : \mathbf{X} \rightarrow \mathbf{A}^{(g)}$.

Zero-shot classification: Given a query instance, it is firstly represented by GSE using f_2 . Our final ensemble mechanism aims to make predictions for instances from both *seen* and *unseen*: $f_3 : (\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(G)}) \rightarrow \mathbf{Y} \cup \mathbf{Z}$.

6.3.3 Implicit Attributes Discovery

Implicit attributes are shared attributes of a group of exemplars without explicit names. Our implicit attributes are under different semantic groups, such as colour, shape, and texture. For instance, one attribute could be a mixture of colours that is possessed by *zebra*, *panda*,

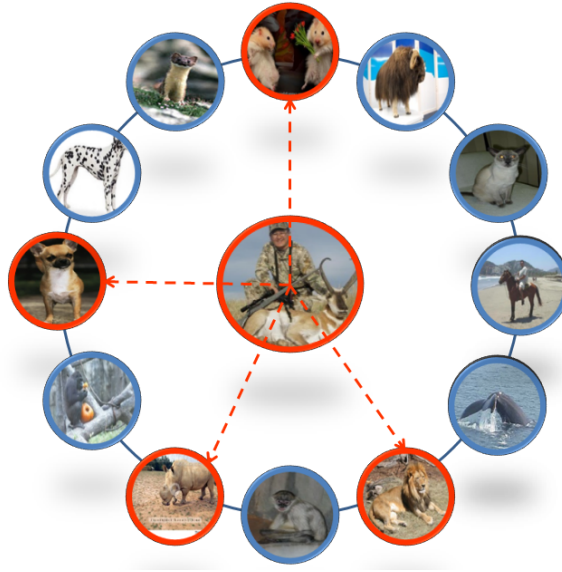


Fig. 6.2 An example of simile annotation process: whose *colour* is similar to *antelope*. The annotator is asked to choose a number of most similar exemplars. We achieve averaged similarities among all of the annotator's associations.

and *dalmatian*. Furthermore, some implicit attributes are even intangible but can be only expressed by similes. The number of such implicit attributes can be arbitrary. Our motivation of using graph-cut aims to scope the various implicit attributes by several clusters. Within each cluster, the simile of exemplars can have very close visual attributes so that we can train stable models for them.

The simile annotation introduced in Section 6.3.1 naturally satisfies a class-level undirected k -nearest neighbour graph. In the graph, each vertex v_c corresponds to a class from $\mathbf{Y} \cup \mathbf{Z}$. Fig. 6.3 illustrates such a problem intuitively. v_{c_1} and v_{c_2} are connected if and only if v_{c_2} is a member of similes \mathbf{NN}_{c_1} of class c_1 . In this way, if v_{c_1} and v_{c_2} are mutually nearest neighbours, the weight of the edge in between is 2. Similarly, if v_{c_1} and v_{c_2} are not mutually nearest neighbours but connected, the weight of the edge in between is 1. Since $\mathbf{NN} \in \mathbf{Y}$, the achieved graph has the same dimension as the number of *seen* classes: $W \in \{0, 1, 2\}^{C \times C}$. Cutting such a graph clusters the seen classes. Each cluster possesses a visually similar implicit attribute.

According to [130], graph cut can be approximated through the spectral clustering approach in order to improve the efficiency. The unnormalised graph Laplacian matrix is defined as:

$$L = D - W, \quad (6.1)$$

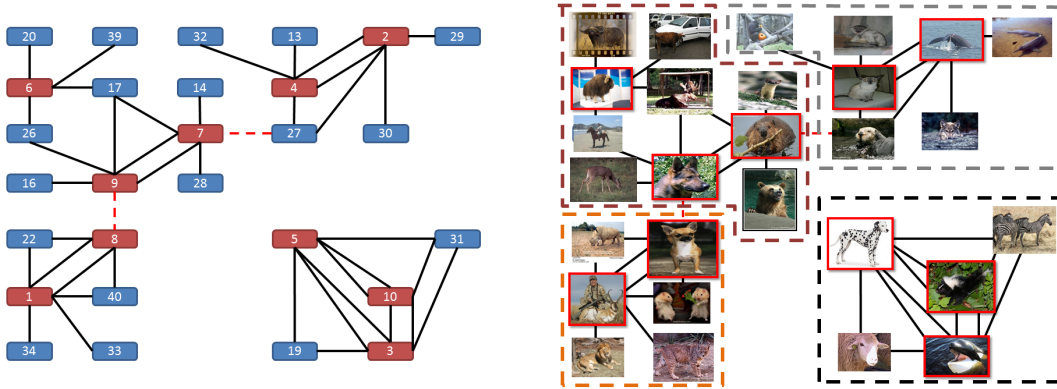


Fig. 6.3 Implicit attribute discovery. Under each simile group, the associated exemplars of each class satisfy a k -nn graph (left). Red vertices indicate *unseen* classes. Our algorithm can cut the weakest edges and cluster the classes with similar implicit attributes (right).

where D is a degree matrix with d_1, \dots, d_C on the diagonal, and each d_c is defined as:

$$d_c = \sum_{c_i=1}^C W_{cc_i}. \quad (6.2)$$

The number of 0s in the eigenvalues of L indicates how many subsets are disconnected. However, in practice, we can decide whether it is necessary to cut those weak connections further by visualising the distribution of remaining non-zero eigenvalues. In Fig. 6.4, we can clearly see that the distribution of the eigenvalues from 40 *seen* classes can be roughly divided into four more groups. Adding on the zero eigenvalue, the optimal number of clusters is 5. Finally, classes are clustered by the k -means algorithm on the first m eigenvectors, where m equals the optimal number of implicit attributes ($m = 5$ in this case).

After graph-cut, each class $c \in \mathbf{Y} \cup \mathbf{Z}$ can be soft-assigned to the discovered implicit attributes according to the original similes \mathbf{NN}_c , i.e. $\mathbf{A}_c = (a_1, \dots, a_m) \in \mathbb{R}^m$. Each dimension indicates the prior probability of each implicit attribute presenting in the class. We repeat such processes for G simile groups.

6.3.4 Grouped Simile Ensemble

The primary purpose of using grouped simile ensemble is to find the most effective attributes for different tasks. Our main idea is to observe the visual data from various semantic aspects. Specifically, we first extract various low-level visual features from the images and concatenate them as base features. We then train embedding functions to map the base features to different simile groups. Such a framework satisfies the spirit of ensemble model

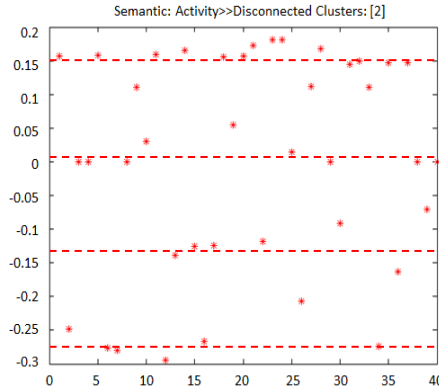


Fig. 6.4 Visualisation of eigenvalues. We demonstrate the example from the simile group of *activity* in the AWA dataset. The k -NN graph of similes has two disconnected subsets (one zero eigenvalue). However, we could find roughly four more layers, which indicates that the optimal value for m is 5.

[34] that a single input can be interpreted with various aspects, *i.e.* simile groups. There are three potential advantages of using ensemble models. 1) The limited training examples now can be utilised multiple times for different simile groups. 2) The difficulty of attribute classification task is lower since the number of implicit attributes in each cluster is much smaller than that of the whole attribute list. Moreover, our pre-process of graph-cut makes the boundaries between implicit attributes more discriminable. 3) the ensemble of base features provides rich representations which make it easier to find discriminative dimensions to satisfy the hypothesis.

The whole ensemble learning task can be defined as a Bayesian probabilistic setting. For each simile group g , we use the discovered implicit attributes as labels to train a hypothesis for supervised multi-label classification. Each hypothesis $h^{(g)}$ embeds the input base visual feature in \mathbf{X} into an implicit attribute space $\mathbf{A}^{(g)}$ satisfy a conditional probability distribution:

$$\mathbb{H}(\mathbf{X}) = \prod_{g=1}^G p(\mathbf{A}^{(g)} | \mathbf{X}, h^{(g)}), \quad (6.3)$$

where the whole GSE model consists of all of the hypotheses in \mathbb{H} , where $\mathbb{H} = \{h^{(1)}, \dots, h^{(G)}\}$, in which each multi-class classifier in each group $h^{(g)}(x)$ possesses a basis. Given a test sample \hat{x} and the training set \mathbf{X} , the problem of predicting the overall implicit attributes of all

simile groups can be expressed as weighted sum over the log ratios all hypotheses:

$$\begin{aligned} p(\hat{\mathbf{A}}|\hat{x}, \mathbb{H}) &= \prod_{g=1}^G h^{(g)}(\hat{x}) p(h^{(g)}|\mathbf{X}) \\ &\propto \frac{1}{G} \sum_{g=1}^G \log h^{(g)}(\hat{x}) p(h^{(g)}|\mathbf{X}), \end{aligned} \quad (6.4)$$

where $\hat{\mathbf{A}} = (\hat{\mathbf{A}}^{(1)}, \dots, \hat{\mathbf{A}}^{(G)})$ is the overall implicit attributes of \hat{x} by concatenating all of the simile groups. During training, \mathbf{A} and \mathbf{X} are in pairs. By taking i.i.d. for Bayes rule we have:

$$p(h^{(g)}|\mathbf{X}) = p(\mathbf{X}|h^{(g)})p(h^{(g)}), \quad (6.5)$$

where $p(h^{(g)})$ is assumed equal to one, the performance of each classifier $p(\mathbf{X}|h^{(g)})$ can be estimated during training. For ZSL tasks, \hat{x} is from unknown classes. The prior training score of $p(\mathbf{X}|h^{(g)})$ may not hold during the test. For an intuitive instance, the *colours* simile group may work better on the training set to distinguish *panda* from *bear*. However, to test with unseen instances *zebra* and *dalmatian*, the *shapes* group is more discriminative. In this paper, we employ the maximum-a-posteriori criteria to make an approximate estimation that can automatically find the most effective simile group for unseen classes. Specifically, we employ LDA [13] to learn discriminative embeddings on the training set so that visual features possessing the same implicit attributes can be projected into a more compact space. Each LDA model $h^{(g)}$ is trained with the g^{th} group of implicit attributes $\mathbf{A}^{(g)}$. We empirically show the advantages of using such embedding in our later experiments. During the test, an unseen instance can be mapped to the embedding hypotheses space by taking the log probability of the maximum likelihood decision rule:

$$\begin{aligned} \hat{\mathbf{A}} &= \arg \max_{\mathbf{A}} \sum_{g=1}^G \log h^{(g)}(\hat{x}) p(\mathbf{X}|h^{(g)}) p(h^{(g)}) \\ &\approx \arg \min_{\mathbf{A}} \sum_{g=1}^G \|h^{(g)}(\hat{x}) - NN_{\mathbf{A}^{(g)}}(h^{(g)}(\hat{x}))\|_F^2, \end{aligned} \quad (6.6)$$

where $NN(\cdot)$ is a nearest neighbour searching from the embedding hypothesis space $\mathbf{A}^{(g)}$ of the g^{th} group, and $\log p(\mathbf{A}|\hat{a}) \propto \sum_{g=1}^G \|h^{(g)}(\hat{x}) - NN_{\mathbf{A}^{(g)}}(h^{(g)}(\hat{x}))\|_F^2$. Intuitively, weights of different simile groups are automatically determined by the Frobenius Norm distances. As a result, the maximum likelihood decision can find the optimal ensemble of implicit attributes of the test instance under each simile group.

6.3.5 Zero-shot Classification

After predicting the implicit attributes $\hat{\mathbf{A}}$, we can make classify a test instance \hat{x} by comparing $\hat{\mathbf{A}}$ to the reference attributes that we have achieved through the graph-cut. As introduced in Section 6.3.3, we have obtained a unique attribute signature \mathbf{A}_j for both *seen* and *unseen* classes, *i.e.* $j \in \mathbf{Y} \cup \mathbf{Z}$. Because $\hat{\mathbf{A}}$ is i.i.d. given its class, the bias towards the *seen* classes can be eliminated. Therefore, we can extend the previous ZSL setting that restricts to test by *unseen* instances. In this paper, our method can classify both *seen* and *unseen* instances at the same time. In order to show the power of our GSE model and the advantages of using implicit attributes, we simply adopt the most straightforward NN classifier:

$$\hat{C} = \arg \min_j \|\hat{\mathbf{A}} - NN(\mathbf{A}_j)\|^2, \quad (6.7)$$

where $\hat{C}, j \in \mathbf{Y} \cup \mathbf{Z}$. Again, if some implicit attributes are incorrectly predicted or annotated, the Frobenius Norm distances can suppress such noises to some extends.

6.4 Experiments and Results

Datasets We evaluate our method on two ZSL benchmark datasets, Animals with Attributes (AwA) [72], and aPascal&aYahoo (aPY) [37]. AwA contains 30,475 images of 50 wild animal classes. In aPY, there are totally 15339 images from more various categories than AwA, including humans, artificial objects, buildings, as well as animals, which makes the recognition task more challenging.

Visual Features In order to compare to as many existing methods as possible, we adopt both low-level features that are provided by the datasets and deep features that are published by [155]. The low-level features include both local and global descriptors, such as SIFT, PHOG, Colour histogram, textual and edge descriptors. Local features are coded by Bag-of-words. We concatenate such low-level features as our base features, on which we perform PCA that results in 9751-dimensional representations. The deep features are extracted by VGG-19 that results in 4096-dimensional representations.

Attributes and Semantic Groups Our GSE does not use the provided explicit attributes in AwA and aPY. On AwA, we adopt the same semantic groups as suggested by [58, 72] for fair comparison. There are nine semantic groups, which are: *colour, texture, shape, part, activity, behaviour, nutrition, and habitat*. For aPY, [58] report that the provided 64 attributes are significantly repeated and redundant. They manually choose 25 of them in their experiments. Such a suggestion also supports the necessity of our idea that using semantic groups. There are five groups: *shape, texture, plant, part, and materials* which are

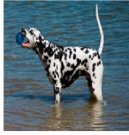

Dalmatian	Colour	Texture	Shape	Part	Activity
	Panda	Giraffe	Bobcat	Zebra	German Shepherd
	Killer Whale	Cow	Leopard	Wolf	Persian Cat
	Zebra	Leopard	Wolf	Lion	Siamese Cat
	Skunk	Skunk	Zebra	Leopard	Collie
	Sheep	Deer	Deer	German Shepherd	Horse
Bicycle	Shape	Texture	Plant	Part	Material
	Motorbike	Motorbike	Motorbike	Motorbike	Motorbike
	Chair	Train	Bus	Person	Boat
	Sofa	TV Monitor	Car	Building	Bus

Fig. 6.5 Examples of images annotated by similes under different groups in AwA (upper) and aPY (lower).

shown in Fig. 6.5. It is noticeable that the *plant* group is unusual and only possessed by the class that is also named *plant*. In the later experiments, we show such an unusual group can be accurately classified.

Simile Annotations We invite three labellers to give annotations for the two data through the process introduced in Section 6.3.1. We accumulate their choices of similes to each target class. We then empirically choose k similes of each target classes, where $k = 5$ and 3 for AwA and aPY respectively. We demonstrate two examples of classes annotated by grouped similes in Fig. 6.5.

6.4.1 Implicit Attribute Discovery

Fig. 6.6 shows some examples of our graph-cut results. We demonstrate the simile groups of *shape* and *part* that shared by the two datasets. Two trends can be seen from the results. Firstly, the clustering tends to agree with the animal taxonomy. For example, in the term of *part*, dogs and wolfs are clustered due to our human visual perception is not isolated from knowledge. The semantic meaning can also affect how we perceive the visual information. The second trend is that we can easily tell many implicit attributes from the cluster of images. For instance, it can be seen that the bulls and goats are clustered. We assume that the implicit attribute is ‘with horns’, although their horns have different styles. In contrast, the aPY dataset is far more challenging. The attributes of natural things, e.g. dog, are barely associated with artificial things, e.g. bikes. Therefore the clusters tend to be more isolated from each other. Consequently, the average size of a cluster tends to be larger than that of

Table 6.2 Compared to the state-of-the-arts using deep features.

Methods	Deep Feature	AwA	aPY
DAP [72]	V	57.23	38.16
SJE [5]	A	61.90	-
ESZSL [116]	V	75.32	24.22
SSE [155]	V	76.33	46.23
JLSE [156]	V	79.12	50.35
Ours	V	78.42	56.38

V: VGG; A: AlexNet; - indicate the published result is missing.

AwA. For example, in Fig. 6.6, seven classes are clustered together in the *shape* group of the aPY dataset, whereas for the AwA dataset, the average cluster size is only 3.57. It is also noticeable that, in the *bicycle* example that is shown in Fig. 6.5, all of the first simile is *motorbike* since this is the only relevant class in the training set. Since the number of classes is small in aPY, such situation does not severely degrade the performance. However, for a large number of unseen classes, we might require the training sources to be more abundant.

6.4.2 Compared to State-of-the-art methods

[t] **Settings** Due to the large variations of published settings that are different in terms of adopted visual features, types of semantics, seen/unseen splits, *etc.* , it is impractical to compare with every possible setting. Therefore, adopt the most common setting, on which the highest published results are reported. Methods under different settings, *e.g.* transductive settings [41, 66, 113], or aided by various semantic informations [3] are not compared. Specifically, the seen/unseen splits is 40/10 for AwA, and 20/12 for aPY. The adopted visual features are extracted by deep models. Our method and most of state-of-the-art methods adopt the VGG-19 features [122] whereas [5] use AlexNet instead. We summarise our comparison in Table 6.2.

Discussion Our method can outperform most of the state-of-the-art methods and the overall recognition rate is only 0.7 % lower than that of [156] on AwA. However, our method achieves significant improvement of 6.03% over [156] on the aPY dataset. We ascribe such performance difference to that the variation of unseen classes of the two datasets is different. For instance, as shown in Fig. 6.5, an *unseen* class of AwA is similar to several *seen* classes, whereas the unseen classes in aPY are often related to only one class. In other words, the boundaries between the implicit attributes in aPY are more discriminative than that of AwA. In contrast, 6.5 adopts explicit attributes which are noisy and therefore cannot share such a priority.

Table 6.3 Compared to baseline methods using low-level features.

Baselines	Attribute	Mapping	AwA	aPY
DAP [72]	A	P	40.5	18.12
DSVA[58]	A+G	E	30.6	19.43
ZSRwUA[57]	A	P	43.0	26.02
ESZSL[116]	A	E	49.3	27.27
DCLA [151]	DA	P	48.3	-
EA + GSE	A+G	E	46.5	25.12
IA + LDA + NN	IA	E	27.4	17.20
IA + Grouping + NN	IA+G	P	44.2	22.82
Ours: IA + GSE	IA+G	E	50.1	30.25

A: Explicit Attributes; G: Attribute Grouping; DA: Data-driven Attributes;
IA: Implicit Attributes; P: Prediction based; E: Embedding based.

6.4.3 Detailed Analysis

Various baseline methods

In order to understand the contribution of each component of our method, we compare to extensive baseline methods and related work using low-level features rather than deep features. For published results, we compare to DAP [72], DSVA [58], ZSRwUA [57], ESZSL [116], and DCLA [151]. We also substitute or remove components in our GSE model so as to show their contributions to the overall performance. Our experiments are summarised in Table. 6.3, using which we can discuss following questions.

Advantages of implicit attributes For the first baseline EA+GSE, we use the same learning framework as our GSE. We only substitute the implicit attributes into conventional explicit attributes. From the comparison between using EA and IA, the performance gains are 4% and 5% on the two datasets, which indicates implicit attributes can adequately fill the visual-semantic gap than explicit attributes. DCLA is data-driven attributes based on visual data that is 8% than DAP, but the performance is 2% lower than ours. More importantly, our implicit has specific semantic meaning, *i.e.* we know which of *seen* classes possess the attributes, whereas DA in DCLA is completely not human-understandable.

Effect of Grouping For the second baseline IA+LDA+NN, we show the effect of using grouped simile. The statistics of all groups are summed up. We then perform graph-cut using non-grouped similes to achieve non-grouped implicit attributes. The model is simply LDA+NN without ensemble. As a result, we observe dramatical performance degradation, 23% on AwA, and 13% on aPY, respectively. The reason is that implicit attributes are only discriminative to class clusters. The classes within the cluster cannot be distinguished, which results in the worst performance.



Fig. 6.6 Partial results of graph-cut class-clustering. Images with in the same colour of frames are from the same cluster.

Visual-semantic mapping approaches Most previous methods adopt the DAP framework that predict each attribute separately. Recent methods are shown improved performance using embedding based framework in [4] that learns all attributes jointly as a whole representation. Our embedding is slightly different from their approach due to the implicit attributes are separated by graph cut. Our purpose is to project the visual data with the same attribute into a compact space rather than multi-label embedding as ESZSL [116]. For the baseline method IA+Grouping+NN, the visual feature is directly mapped to training samples and use the attributes of the nearest neighbour for prediction like IAP[72]. Again, our

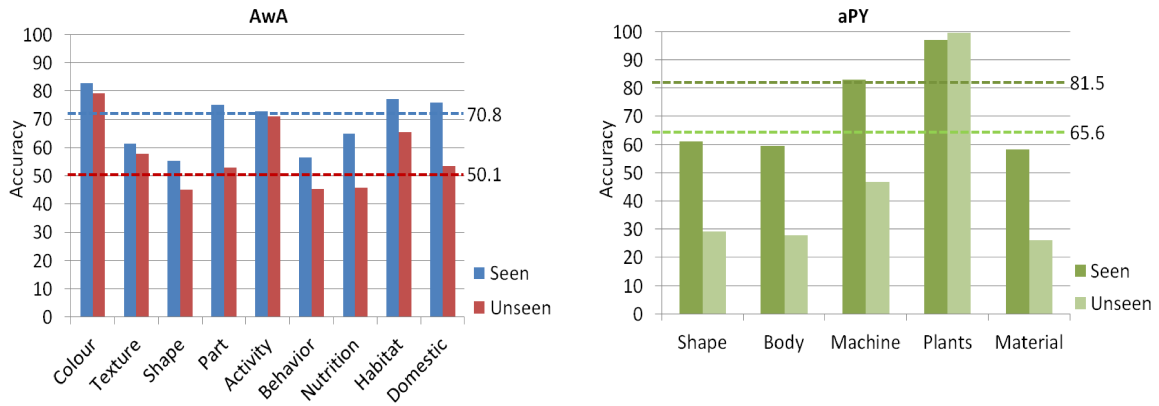


Fig. 6.7 Implicit Attribute Prediction Precision on AWA and aPY. Results are shown by different simile groups.

method significantly outperforms all of the aforementioned baselines.

Efficiency The entire framework is very efficient. Even though the off-line training time is usually not that important, it can determine whether or not the method can be utilised in practical applications. Our work is conducted in Matlab 2014a environment that is installed on a 12-core Linux system with 400G memory. For PCA, it takes 123 seconds and 109 seconds on AWA and aPY datasets, respectively. For LDA, each semantic group requires up to 20 seconds to train each model. Besides these two main training steps, the other procedures are completed within a few seconds. We ascribe the high efficiency to the grouping strategy and the highly compact implicit attributes. Because the learning task is decomposed into grouped subtasks, the computation cost is reduced exponentially.

Implicit attribute prediction

The success of our GSE relies on the premise that the implicit attributes can be reliably predicted. Since our graph-cut algorithm assigns each class to one implicit attribute, during the test, we examine whether the images are mapped to the correct implicit attributes. We test on both *seen* and *unseen* classes to show the performance drop from training to test. From Fig. 6.7, we can see the average performance drop is roughly 20% on both datasets. However, in aPY, only one class use the highest group *plants*. The remaining training-test performance drop is significantly large, which explain the overall ZSL recognition rate is only 30.25% in Fig 6.3. Interestingly, the recognition rate on AWA is the same to the average precision of implicit attribute prediction. Such results manifest our embedding mechanism can reliably make ZSL prediction based on given implicit attributes. The attribute-to-label gap is zero in this case. We assume the visual-semantic error is corrected by our ensemble

Table 6.4 Evaluating GSE on different settings.

Settings	AwA		aPY	
Methods	DAP	Ours	DAP	Ours
$\mathbf{X}_{train} \rightarrow \mathbf{Z}$	50.2	49.7	18.42	30.16
$\mathbf{X}_{train} \rightarrow \mathbf{Y}$	39.8	70.4	49.96	64.32
$\mathbf{X}_{train} \rightarrow \mathbf{Y} + \mathbf{Z}$	12.9	42.5	13.84	24.22

mechanism to some extents.

GSE under different Scenarios

Lastly, we evaluate our GSE under different settings. We mainly concern how is the performance when testing by both *seen* and *unseen* classes. We randomly choose half of the images in each *seen* class for training (denoted by \mathbf{X}_{train}) and the other half for testing (\mathbf{X}_{test}). Firstly, we perform ZSL recognition on the reduced training set. The overall accuracies do not drop down (50.1 to 49.7 and 30.25 to 30.16). The second setting is conventional classification task, (\mathbf{X}_{test}) is also from *seen* classes. We observe significant improvements over the ZSL results. In the last experiment, the test images are from mixture classes of \mathbf{Y} and \mathbf{Z} . The performance loss is not severe, *i.e.* only 7% and 6% recognition rate drop for the two datasets. Such results indicate our method can withstand the training-bias problem in most existing approaches, such as DAP [72].

6.5 Conclusion

In this paper, we proposed a unified framework for ZSL including simile annotating, implicit attribute discovery, and the GSE model for ZSL classification. Our method achieved state-of-the-art results on AwA and significantly outperformed existing methods on aPY. We conclude our work as follows. Firstly, similes are effective to describe complex visual appearance. Grouping makes simile more meaningful and discriminative for ZSL tasks. Secondly, our graph-cut algorithm can reliably capture the implicit attributes from similes and do not suffer from the correlation and training bias problems. Thirdly, our ensemble mechanism can find the most relevant simile groups during the test. As a result, the loss of accuracy from attribute prediction to ZSL recognition is small.

For future work, it is necessary to extend our method on large-scale datasets so as to achieve more class exemplars for similes. Another interesting direction for future investigation is the cross-domain ability of the implicit attributes. Since most of the similes are

visual-based general terms, we do not need to change the attribute list to adapt to different datasets. One could train rich implicit attribute models on a large-scale dataset that can be generalised widely. In this way, the cost of designing attribute list is significantly mitigated.

Chapter 7

Towards Affordable Ontology

7.1 Introduction

Zero-shot learning (ZSL) techniques aim to transfer a learnt model to unseen classes without acquiring new instances. The key problem is how to relate unseen classes to previously trained models using prior human knowledge. Existing ZSL methods leverage structured descriptive models, such as attributes or texts, so as to generalise to novel unseen classes. However, labelling attributes or collecting the semantic descriptions for ever-growing new classes is very expensive. For example, the most popular benchmark, AwA [72], requires the annotator to give 85 attributes for each of 50 classes, let alone instance-level datasets, such as aPY [37] and SUN [105] which contain hundreds of thousands of manual annotations. Such restrictions severely prevent ZSL from being widely applied to many non-attribute scenarios.

The Main Contribution of this chapter focuses on how to spend the minimal cost to apply ZSL on datasets without labelled attributes or texts. We first investigate such a problem on conventional ZSL benchmarks, AwA, and aPY. We then apply the proposed method on one of the most traditional classification dataset, Caltech 101. As a result, the minimum cost for these tasks requires providing only a pair of similes of seen classes for each unseen class. No similes are required for seen classes.

Our key idea is illustrated in Fig. 7.1. To describe an unseen instance, the most straightforward way is to relate it to seen classes. Such expressions are called *similes* which explicitly compare two things by connecting words, *e.g. like, as, as, etc.* Accordingly, we propose a similarity-based representation, which is a direct bridge between low-level visual features and semantic similes without much information loss. Such representations can be achieved by a simple but effective algorithm called *Match Kernel Embedding* (MKE) which only involves visual features and seen class labels in the training set as traditional supervised

Which *unseen class* does each *test image* belong to?

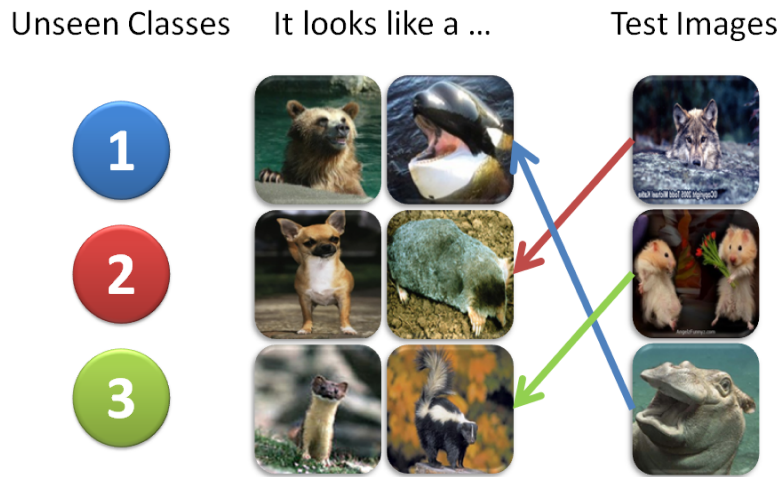


Fig. 7.1 Given some similes as clues, humans can easily make classification for the unseen classes.

settings. Expensive attribute annotations are circumvented.

Furthermore, we find it is infeasible to quantify the simile between each pair of seen and unseen classes. The labelling cost of doing this is not less than that using attributes. Motivated by this problem, we propose our second idea illustrated in Fig. 7.5. Given a pair of similes *bobcat* and *tiger* without seeing the unseen class *leopard*, humans can infer the similarity between *leopard* and *lion* is high. In other words, *given a number of similes, we can infer the similarities between the unseen class and all of the seen classes*. In our empirical study, we use only two similes but achieve state-of-the-art ZSL performance. Such a result significantly reduces the annotation cost. According to the common 40/10 seen/unseen split of AWA, the number of required labels is reduced from 50×85 to only 10×2 .

The third challenge is the difficulty in assigning specific numbers to represent the similarities. Rather, it is easier for humans to provide qualitative similes than saying ‘*a bobcat is 0.8 similar to leopard*’. Therefore, we need to design an approach that can decide values for qualitative similes. A simile quantification process is proposed followed by inferring a class-level prototype using regression models. A test instance then can be directly mapped into the MKE space and find the most similar inferred unseen prototypes to make the ZSL prediction.

The rest of the chapter is organised as follows. In Section 2, we compare our simile-based approach with existing ZSL frameworks. We introduce the key steps in detail in

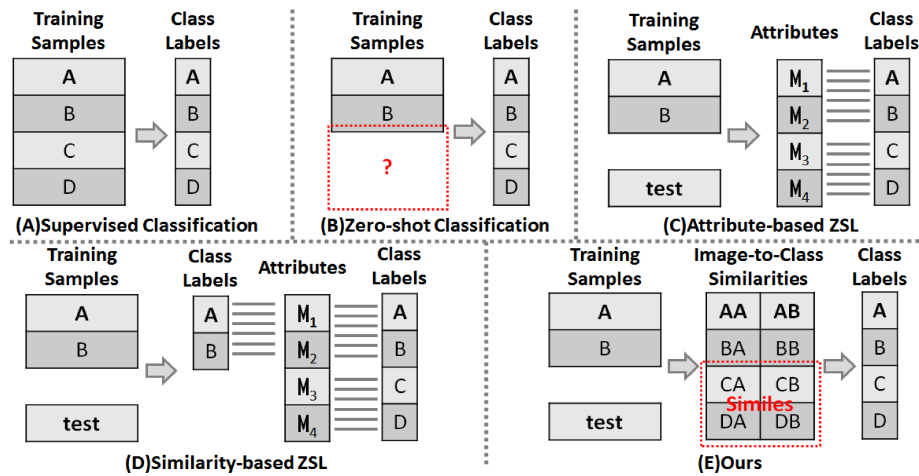


Fig. 7.2 Comparison between ZSL frameworks. The gray bars between attributes and class labels denote the human-defined class-attribute matrices.

section 3. Our experimental results are discussed in Section 4. The final Section 5 concludes our contributions and possible future applications.

7.2 Related Work

Zero-shot Learning Frameworks In conventional supervised classification (Fig. 7.2 (A)), the objective is to learn a mapping from the training images to class labels. Zero-shot learning [72] aims to tackle the problem that test images come from unseen classes without training samples (Fig. 7.2 (B)). The problem has many realistic applications. *Example 1:* we have trained models for class A and B, but the test set may incrementally extend to some new classes C and D that have no available trained models [63]. *Example 2:* it may be difficult to acquire images for some rare animals. How can we use the trained models for A and B to classify C and D? The core issue is how to make the trained model generalise to unseen classes.

Existing frameworks can be roughly divided into two categories. The first category aims to learn an intermediate-level model, such as semantic attributes, using training samples from seen classes [4, 22, 37, 41, 58, 72, 91, 116, 119, 153]. In this way, the learnt representation is interpretable and thus can be shared by unseen classes with human-defined class-attribute associations (Fig. 7.2 (B)). Such attribute-based frameworks aim to tackle the problems like example 2. The assumption is that we have enough prior knowledge to build attribute models on seen classes in order to classify those novel unseen ones. However, adding new attributes will need to re-train the attribute models. The other main category is similarity-based ZSL (Fig. 7.2 (C)) [63, 91, 96, 151, 155]. Instead of training attribute

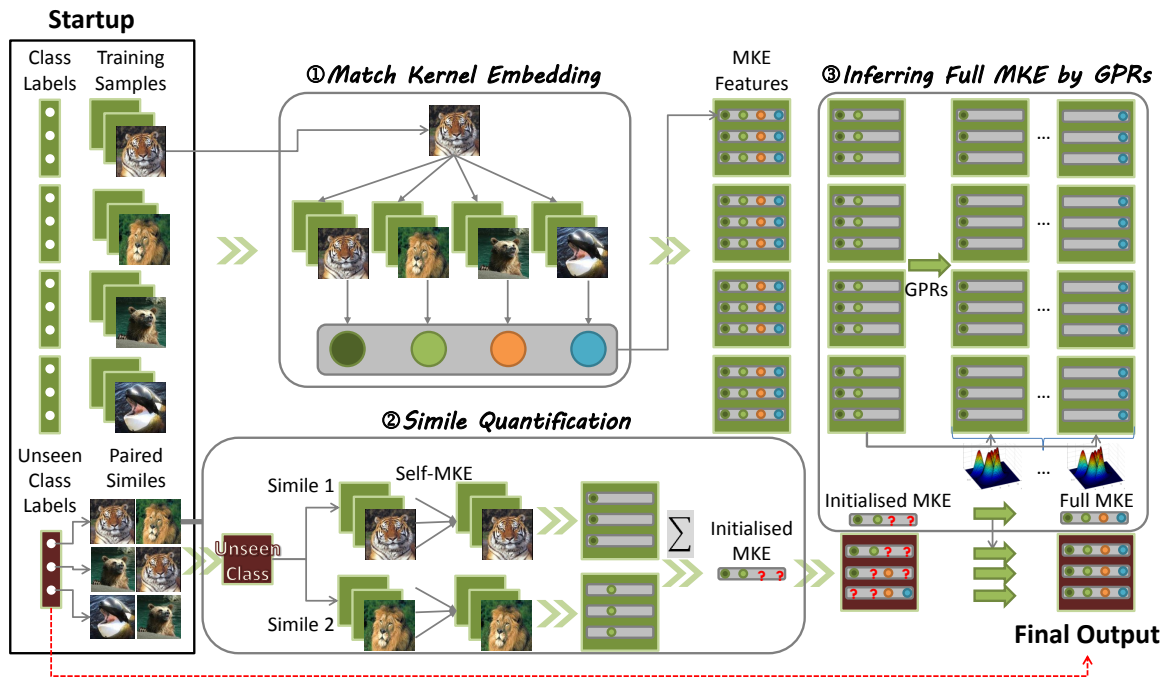


Fig. 7.3 An overview of training stages. Each unseen class gains a class-level prototype by inferring from the paired similes.

models, newly added C and D classes are predicted using the trained model for A and B. The test image is first mapped to seen classes A and B to estimate the similarities which are then mapped to unseen classes through the human-defined class-attribute matrix. Adding new attributes or unseen classes will not need to re-train the learnt model for A and B.

Auxiliary Information The key limitation of using attributes is the high expense that mainly comes from three aspects. Firstly, deciding what attributes to be considered is often ambiguous. For example, in the CUB dataset [131], novice users can hardly decide which attributes are effective for different birds. Expert knowledge is often required. In addition to the first point, the expert who defines the list of attributes may be from other fields, such as a zoologist. The attributes may be effective in semantic or taxonomy but not guaranteed to be directly related to visual appearance, *e.g. domestic* in the AWA dataset. Thirdly, collecting reliable attribute annotations is expensive. For a new task, tens or even hundreds of classes are associated with nearly a hundred attributes, which often require costly on-line services, such as Amazon Turk. Also, the annotator may not be qualified and can suffer from individual bias, which leads to low-quality attribute annotations [57].

In order to circumvent expensive attributes, there are some alternatives proposed in the literature. The most straightforward idea is automatically associate class labels using textual models [8, 36, 76, 94, 101, 109, 115]. However, due to the semantic features are

extracted from resources of Wikipedia or news, the resultant performance is often lower than that of using attributes. Hierarchical class embeddings also provide a promising possibility [48, 114]. Again, constructing hierarchical dataset by ontology engineering could be more expensive than that of using attributes.

Compared to the existing work, our framework (Fig. 7.2 (E)) is similar to the similarity-based ZSL. However, the expensive attribute associations are substitute by image-to-class similarities that are estimated by training images without human interference. The similes we use share the idea of [70] that can effectively express complex visual appearances without be limited by human words or attributes. However, our similes focus on the implicit impression of the whole image without local segmentations, which makes the work easier and more efficient. A very close approach is [151] that pairs each class with other top-5 similar classes for ZSL. However, as argued earlier, such an annotation task for both seen and unseen classes is still burdensome. This is because the number of seen classes is often much larger than that of unseen ones. In contrast, our proposed framework only requires the annotations for unseen classes, while the seen-to-seen similarities are directly estimated using training images.

7.3 Approach

Our training stage is demonstrated in Fig 7.3. Our key idea is to infer the similarities between an unseen class and all seen classes using a few similes, which can form a similarity-based representation as a prototype of the unseen class. In order to achieve this, our first step is *Match Kernel Embedding* (MKE) that converts each training sample into a similarity-based MKE representation. Secondly, due to semantic similes are qualitative descriptions, we need to quantify them into real values in the initialised MKE representation, *i.e.* to specify how similar is an unseen class to the simile of seen class. Using the initialised MKE, we can finally infer the full MKE representation in the third step. During the test, as shown in Fig. 7.6, an image from unseen classes can be mapped into the MKE space and be compared to the inferred unseen MKE prototypes to make the prediction.

In the following, we first introduce the preliminary of the ZSL classification task (Section 3.1), and then explain our key steps in details consecutively (Section 3.2-3.4), and how to achieve ZSL classification during the test (Section 3.5).

7.3.1 Preliminary

Given a training dataset, which is formed by pairs of images and labels from seen classes as most of supervised settings: $(x_1, y_1), \dots, (x_N, y_N) \subseteq \mathbf{X} \times \mathbf{Y}$, where $\mathbf{X} = [x_n] \in \mathbb{R}^{N \times D}$ is a D -dimensional feature space and $y_n \in \{1, \dots, C\}$ consists of C seen classes. ZSL classification aims to predict the labels of test instances to U unseen classes $\mathbf{Z} = [z_u] \in \{C+1, \dots, C+U\}$ which have no training images before the test, *i.e.* $\mathbf{Z} \cap \mathbf{Y} = \emptyset$. Our approach aims to achieve this by using only a few similes using seen classes as clues for each unseen class. In later experiments, we show that only a pair of similes is enough to achieve reliable ZSL classification performance. For simplicity, we denote the paired similes of the u -th unseen class as $\mathbf{S} = [s_u^1, s_u^2] \in \{1, \dots, C\}$.

7.3.2 Match Kernel Embedding

To represent images by similarities, the most straightforward approach is to estimate the likelihood between an image x_n and each image in a seen class in the visual space, $x_c \in \mathbf{X}_c$. Here, we simply adopt the Parzen likelihood estimation to compare a pair of images:

$$p(x_c|x_n) = k(x_c - x_n) = \exp\left(-\frac{1}{2\sigma^2}\|x_c - x_n\|_2^2\right), \quad (7.1)$$

where $k(\cdot)$ is the Parzen match kernel function under a typical Gaussian distribution, which is non-negative and integrates to one; $\|\cdot\|_2$ is the ℓ_2 -norm distance between two vectors. Using the above equation, each training sample can make a contribution to estimate the image-to-class likelihood $p(x_n|c)$. However, due to the visual space is high-dimensional and D -exponentially decreases with the distance, most of the likelihood values are negligible. Moreover, the training set can be noisy. A conclusion using all of the samples may not lead to the best result. Therefore, we use top P nearest points of x_n , *i.e.* $\{x_{NN_c}^1, \dots, x_{NN_c}^P\} \in \mathbf{X}_c$, to make an improved estimation:

$$p(x_n|c) = \frac{1}{P} \sum_{i=1}^P \exp\left(-\frac{1}{2\sigma^2}\|x_{NN_c}^i - x_n\|_2^2\right). \quad (7.2)$$

Our complete MKE representation is achieved by repeating the above equation on all of the seen classes. Each dimension of the MKE vector stands for the likelihood value to the corresponding class. For simplicity, we fix $\sigma = 1$. The MKE is formalised below:

$$f_{MKE}(x_n) = [p(x_n|c=1), \dots, p(x_n|c=C)] = v_n, \quad (7.3)$$

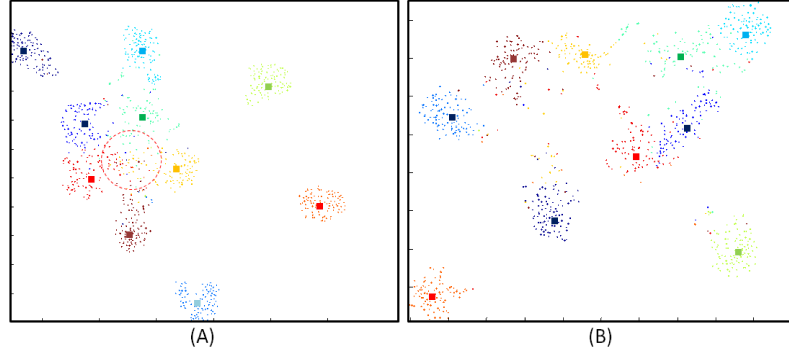


Fig. 7.4 (A) Raw visual feature distribution of the 10 unseen classes in AwA. (B) After MKE, non-discriminative points (red circle in (A)) are separated. Before the test, we aim to infer the centroid of each class as the prototype.

where $v_n \in \mathbf{V}$ denotes a complete MKE vector corresponding to the training sample x_n in the embedding space $\mathbf{V} \in \mathbb{R}^{N \times C}$. Each dimension of \mathbf{V} indicates the likelihood to a seen class.

The proposed MKE representation has two advantages. Firstly, MKE is a direct quantitative representation of similes that can bridge the visual-semantic gap. Moreover, as shown in Fig. 7.4, the embedding space becomes more discriminative after being kernelised, which can significantly benefit the classification performance.

7.3.3 Simile Quantification

For each unseen class z_u , our goal is to achieve a class-level MKE representation $v_u = [p(u|c=1), \dots, p(u|c=C)]$ as a prototype for ZSL classification. However, due to there is no training images of class u at all, we have to infer the above likelihoods using the only known information, *i.e.* the associated similes of two discrete seen class labels. Suppose $s_u^1 = c_1$ and $s_u^2 = c_2$, $c_1, c_2 \in \{1, \dots, C\}$, our first step is to quantify such semantic similes into initialised MKE values, *i.e.* $[s_u^1, s_u^2] \rightarrow [p(\bar{x}_u|c=c_1), p(\bar{x}_u|c=c_2)]$, where \bar{x}_u is the assumed class-level prototype of the unseen class u .

This is a very difficult step since the given semantic similes contain very little information and are often affected by human bias. Fortunately, we can use the training samples in the classes of similes to make an indirect approximate inference. Similar to [72], we assume a deterministic dependence between visual instances and classes, we set $p(\bar{x}_u|x_n) = \log \exp(\|\bar{x}_u - x_n\|_2^2)$. For a simile $s_u = c_1$ we have:

$$p(\bar{x}_u|c=c_1) = p(\bar{x}_u|x_n)p(x_n|c=c_1), \quad (7.4)$$

where $p(x_n|c=c_1)$ is the image-to-class likelihood of sample x_n in the c_1 -dimension of the

MKE representation v_n . In other words, if a training sample x_n is very similar to the unseen class, *i.e.* $p(\bar{x}_u|x_n) = 1$, the objective $p(\bar{x}_u|c = c_1)$ can be reliably estimated by $p(x_n|c = c_1)$. However, such a sample-level simile can hardly be obtained reliably due to human bias, we average all of the self-MKE values in class c_1 as an approximate estimation:

$$p(\bar{x}_u|c = c_1) = \frac{1}{|c_1|} \sum_{i=1}^{|c_1|} p(x_i|c = c_1), \quad (7.5)$$

where $|\cdot|$ is the cardinality of the class c_1 and $x_i \in \mathbf{X}_{c_1}$ is a training sample in class c_1 . We repeat the same process on the second simile $s_u = c_2$ together to achieve initialised MKE of class u : $[p(\bar{x}_u|c = c_1), p(\bar{x}_u|c = c_2)]$.

7.3.4 Inferring Complete Class-level Prototype

Using the initialised MKE, our final goal is to infer a complete MKE representation for the unseen class u as a class-level prototype: $[p(\bar{x}_u|c = c_1), p(\bar{x}_u|c = c_2)] \rightarrow [p(\bar{x}_u|c = 1), \dots, p(\bar{x}_u|c = C)]$. The key idea is to model the correlation between similes, as shown in Fig. 7.5. Assume the unseen class *leopard* is 0.8 and 0.7 similar to *bobcat* and *tiger*, the similarity to the *lion* is probably high due to its high-correlation to *bobcat* and *tiger*.

Given the training set in MKE space \mathbf{V} , we can observe how each sample is similar to the above simile classes c_1 and c_2 . Also, we can observe how such pairs of similes correlate to other seen classes. Our goal is that, for each seen class c , we train a regression model using the MKE values between each sample x_n and c_1 and c_2 as inputs, and the output is the correlations to all of the seen classes. Note that c_1 and c_2 are also included so that the initialised MKE values can be updated and the bias can be mitigated. In order to clearly denote these inter-dimension inferences, we re-write the input simile pairs in MKE values: $\mathbf{Q} = [q_n] = [v_n^{c_1}, v_n^{c_2}] = [p(x_n|c = c_1), p(x_n|c = c_2)] \in \mathbb{R}^{N \times 2}$, and the output value: $v_n^{c_i} = p(x_n|c = c_i)$, $c_i = 1, \dots, C$. For each seen class c_i , the training set is pairwise: $\{(q_n, v_n^{c_i}), n = 1, \dots, N\}$, using which we train a regression model:

$$f_{c_i} := f_{c_i}(q_n) = v_n^{c_i}, \quad (7.6)$$

using which the initialised MKE of unseen class u $\bar{q}_u = [p(\mathbf{X}_{c_1}|\bar{x}_u), p(\mathbf{X}_{c_2}|\bar{x}_u)]$ can be used as input to infer each value in the complete MKE representation: $v_u^{c_i} = f_{c_i}(\bar{q}_u)$.

Target-sensitive Gaussian Process Regression The above regression task f_{c_i} is different from conventional scenarios for two reasons. Firstly, there is no unpredictable test phase. The target value to be predicted has the exact query, *i.e.* the initialised MKE \bar{q}_u . Secondly,

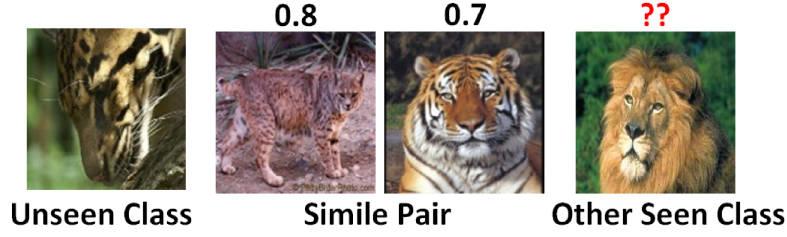


Fig. 7.5 Illustration of the idea to infer complete MKE representation (numbers are only for demonstration purposes).

for each predicted value $v_u^{c_i}$, the whole training set \mathcal{Q} is used. Using all of the n samples is not the best choice since most of them are not relevant to the unseen class u . Therefore, this chapter proposes a Target-Sensitive Gaussian Process Regression (TS-GPR) model to make the prediction introduce as follows.

As most GPR models, we assume Gaussian distribution over the target value $v_n^{c_i}$ given by $v_n^{c_i} = f_{c_i}(q_n) + \varepsilon$, where ε is Gaussian noise with zero mean and variance σ_n^2 . In this way, we can increase the tolerance to the bias from semantic similes. In other words, when we give ‘ u looks like c_1 ’, the implicit meaning is a probability based on prior human knowledge with bias ε . The resultant target values can be described by $f_{c_i} \sim \mathbf{N}(\mu(\mathcal{Q}), \mathbf{K}(\mathcal{Q}, \mathcal{Q}) + \sigma_n^2 \mathbf{I})$, where $\mu(\cdot)$ is the mean operation and the covariance matrix is achieved by $k(q, q') = \exp(-\frac{1}{2\sigma^2} \|q - q'\|_2^2), \forall q, q' \in \mathcal{Q}$ that is the same match kernel function which we use in the Eq. 7.1 which is well-known as a positive semi-definite matrix. The identity matrix \mathbf{I} with hyper-parameter σ_n controls the noise. The advantage is that using such match kernels is parameter-free and consistent to our previous measurement.

Hereafter, the joint distribution of the observed MKE values in \mathcal{Q} and the inferred MKE value $\bar{q}_u = [p(\mathbf{X}_{c_1} | \bar{x}_u), p(\mathbf{X}_{c_2} | \bar{x}_u)]$, the corresponding Gaussian process regression is:

$$\begin{bmatrix} f_{c_i} \\ f_{c_i}(\bar{q}_u) \end{bmatrix} \sim \mathbf{N} \left(\mathbf{0}, \begin{bmatrix} \mathbf{K}(\mathcal{Q}, \mathcal{Q}) + \sigma_n^2 \mathbf{I} & \mathbf{k}(\mathcal{Q}, \bar{q}_u) \\ \mathbf{k}(\bar{q}_u, \mathcal{Q}) & k(\bar{q}_u, \bar{q}_u) \end{bmatrix} \right). \quad (7.7)$$

The conditional distribution yields the predicted mean value that is used as the inferred MKE value:

$$v_u^{c_i} = f_{c_i}(\bar{q}_u) = \mathbf{k}(\mathcal{Q}, \bar{q}_u)^\top (\mathbf{K}(\mathcal{Q}, \mathcal{Q}) + \sigma_n^2 \mathbf{I})^{-1} f_{c_i}, \quad (7.8)$$

where the hyper-parameter σ_n is optimised by maximising the log marginal likelihood using Quasi-Newton methods. Note that the computational cost of the inverse matrix $(\mathbf{K}(\mathcal{Q}, \mathcal{Q}) + \sigma_n^2 \mathbf{I})^{-1}$ is $\mathcal{O}(n^3)$. Due to we only need to make the prediction once for $v_u^{c_i}$, it is not necessary to consider all of the n samples. Instead, we propose to use R closest points of \bar{q}_u ($R \ll N$)

to build R local models:

$$\begin{bmatrix} f_{c_i}(q_r) \\ f_{c_i}(\bar{q}_u) \end{bmatrix} \sim \mathbf{N} \left(\mathbf{0}, \begin{bmatrix} 1 + \sigma_n^2 & k(q_r, \bar{q}_u) \\ k(\bar{q}_u, q_u) & 1 \end{bmatrix} \right), \quad (7.9)$$

where q_r is one of the R closest points of \bar{q}_u . The prediction for a mean value $v_u^{c_i}$ is performed using a weighted sum of the R models: $f_{c_i}(\bar{q}_u) = \sum_{r=1}^R f_{c_i}(\bar{q}_u)_r p(k|\bar{q}_u)$. Using the Bayesian theorem we have $p(k|\bar{q}_u) = p(k, \bar{q}_u) / \sum_{r=1}^R p(k, \bar{q}_u) = k(q_r, \bar{q}_u) / \sum_{r=1}^R k(q_r, \bar{q}_u)$. Therefore, the overall prediction is:

$$v_u^{c_i} = f_{c_i}(\bar{q}_u) = \frac{\sum_{r=1}^R k(q_r, \bar{q}_u)^2 (1 + \sigma_n^2) f_{c_i}(q_r)}{\sum_{r=1}^R k(q_r, \bar{q}_u)}. \quad (7.10)$$

The model can be interpreted as R Gaussian processes weighted by the kernelised distances. We empirically find that $R = 5 \sim 15$ can give the best results, which is much more efficient than using all of the N training samples. Therefore, the proposed method can be efficiently repeated to infer each c_i -th value of the completed MKE for each unseen class u . The overall complexity is $C \times U \times \mathcal{O}(R^3)$. Our algorithm at the training stage is summarised in Algorithm 4.

Algorithm 4: Unseen MKE Prototype Inference

Input: Training set $\{\mathbf{X}, \mathbf{Y}\}$, Hyper-parameters P, R , Unseen labels with similes $\{\mathbf{Z}, \mathbf{S}\}$,

Output: Unseen MKE prototypes: $\mathbf{V}_U = [v_u^{c_i}] \in \mathbb{R}^{U \times C}$

- 1 Convert \mathbf{X} into MKE space \mathbf{V} using Eq. 7.3;
 - 2 **Forall** $u \in \{1, \dots, U\}$
 - 3 Initialise MKE $\bar{q}_u = [p(\bar{x}_u|c = c_1), p(\bar{x}_u|c = c_2)]$ using Eq. 7.5;
 - 4 **Forall** $c_i \in \{1, \dots, C\}$
 - 5 Extract training observations: $\mathbf{Q} = [q_n] = [v_n^{c_1}, v_n^{c_2}]$;
 - 6 Find R nearest neighbours from \mathbf{Q} : $\{NN_1(\bar{q}_u), \dots, \{NN_R(\bar{q}_u)\}$;
 - 7 Predict $v_u^{c_i}$ using Eq. 7.10;
 - 8 **Return** \mathbf{V}_U
-

7.3.5 Zero-shot Classification

Once we obtain the inferred MKE prototype of each unseen class, the zero-shot classification test can be carried out efficiently. As illustrated in Fig. 7.6, the MKE prototypes and unseen class labels are in pairs: $(v_u, z_u) \in \mathbf{V}_U \times \mathbf{Z}$. The test image \hat{x} is first mapped into the MKE space \hat{v} by computing its similarities to all of seen classes using Eq. 7.3. The

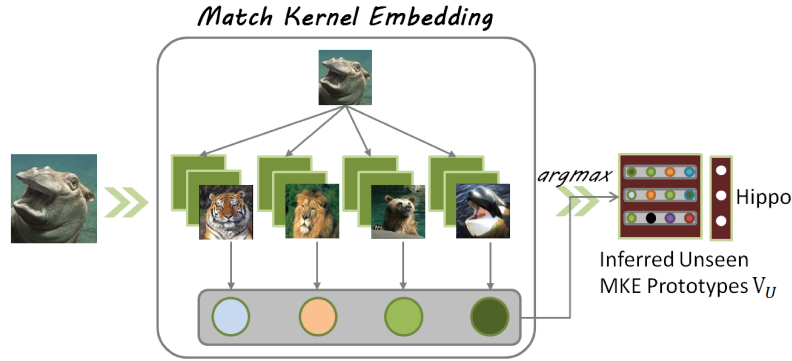


Fig. 7.6 Illustration of test phase: a test image can be converted into the MKE space and compared to the inferred MKE prototypes of unseen classes to make a prediction.

Table 7.1 Dataset statistics

Datasets	#Images	Seen/Unseen Splits	#Attributes	#Similes
AwA	30,475	40/10	50*85	10*5
aPY	14,339	20/12	32*64	12*5
Caltech-101	9,144	50/51 and 51/50	-	101*5

zero-shot prediction is then achieved by comparing to the prototype list and find the one with the highest match score:

$$\hat{u} = \arg \max_u \|\hat{v} - v_u\|_2^2. \quad (7.11)$$

The above equation denotes the procedure of conventional ZSL settings. It is easy to extend to generalised ZSL (GZSL) tasks as well. Instead of considering only unseen prototypes V_U , we can consider both V_U and the training set in the MKE space V , which assume the test image can come from both seen and unseen classes, *i.e.* $\hat{u} \in \{1, \dots, C, C+1, \dots, C+U\}$.

7.4 Experiments

Datasets Our method is evaluated on two of the most popular ZSL datasets: **Animals with Attributes** (AwA) [72] and Attribute Pascal and Yahoo (aPY) [37] so as to compare to

Table 7.2 Simile annotation evaluation using hit rate.

Datasets	S1	S2	S3	S4	S5	Overall
AwA	1.00	1.00	0.70	0.60	0.30	0.72
aPY	1.00	0.83	0.67	0.42	0.25	0.63
Caltech-101	0.97	0.81	0.60	0.27	0.19	0.57

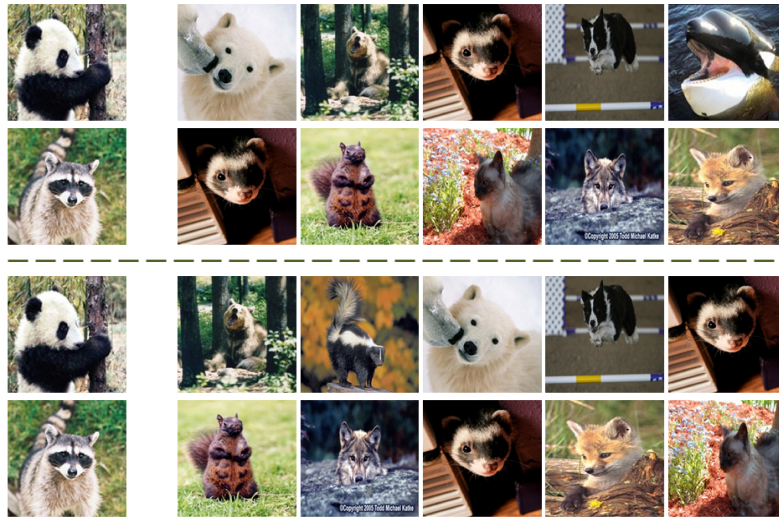


Fig. 7.7 Examples of top-5 similes from human annotations (upper) versus top-5 classes with largest MKE values (lower).

previous ZSL methods. In order to show the feasibility of applying our method on non-attribute datasets, we provide simile annotations for one of the most classic datasets for object recognition: **Caltech 101** [78]. Key dataset statistics are summarised in Fig. 7.1, in which we can see the required number of simile annotations is significantly smaller than that of attributes.

Evaluation Criterion Since existing ZSL methods vary in adopted visual features, auxiliary information, frameworks *etc.*, we compare to both published results and implementations using alternative ZSL techniques in our framework. The key criterion is the single label image classification accuracy which measures whether the top-1 prediction is the correct label. We use the most widely adopted 40/10 and 20/12 seen/unseen splits on AWA and aPY. In order to compare to previous results on the Caltech 101 dataset, we alternate the seen and unseen classes so that all of the 101 categories are evaluated. We average the accuracies as the overall results.

Implementation Details The visual features used on the AWA and aPY datasets are extracted by the VGG-19 model and released by [155]. In order to compare to the previous results on Caltech 101 using conventional features, we extract SIFT, PHOG, LBP, and colour histogram as [72] and aggregate each type of local features using 500-D VLAD [10] and concatenate all types of involved features into a rich representation, on which we perform PCA that results in a 9751-dimensional feature space. For the hyper-parameter of MKE P , we empirically fix $P = 10$ for all of the experiments. We use 5-fold cross-validation to obtain R . Specifically, the seen classes are divided into five groups, four of which are used for training while the other one is used for validation. Classes with top-2 MKE scores are

Table 7.3 Main comparison with the state-of-the-art results.

Methods	Feature	Prior	AwA	aPY
DAP [72]	VGG	A	57.23	38.16
SJE [5]	Alex	Comb	73.90	-
ESZSL [116]	VGG	A	75.32	24.22
SSE [155]	VGG	T	76.33	46.23
JLSE [156]	VGG	T	79.12	50.35
CAAP [8]	GLN	Wiki	67.5	37.0
Ours	VGG	Si	82.28	55.62

A: Attributes; Comb: Combination of Output Embeddings;
T: Transductive Information; Si: Similes

Table 7.4 Compared to supervised results (%) on Caltech 101.

Method	15 images/ seen Class	30 images/ seen class
Grauman&Darrell <i>et al.</i> [50]	49.5	58.2
Tuytelaars <i>et al.</i> [127]	61.3	69.6
Boiman <i>et al.</i> [15]	65.0	70.4
Vedaldi <i>et al.</i> [128]	66.3	-
Ours	67.8	71.2

used as similes of the validating classes.

7.4.1 Simile Annotations

We adopted a straightforward strategy to collect simile annotations, which is similar to [151]. Our similes were annotated by eight students, who were in computer vision field but not aware of the technique details of this work. During the annotation task, they were encouraged to think about the similarities from visual aspects rather than semantics. For each unseen class, the users intuitively chose top-5 similar seen classes as similes. Finally, we concluded the provided similes for each unseen class and selected five classes with highest votes for our experiments.

Human versus Machine One of the vital issues is how accurate the human similes can match the visual similarities estimated by machines. We evaluate the quality of our collected similes by hit rates, *i.e.* whether the annotated simile is one of the top-similar seen classes in the feature space. For example, in Fig. 7.7, we show two of the unseen classes, *Panda* and *Raccoon*, and their human-annotated five similes (upper half). For machine views, we use the mean vector of the extracted visual features and convert it into MKE space by estimating its similarities to seen classes using Eq. 7.3. A hit is counted if the human-annotated simile can be found in the five classes with the top-5 highest MKE values, *e.g.* the first simile (*S1*) of *Panda*, *Polar Bear*, is found in the third position of the top-5 similar seen classes. In Table 7.2 we can see most of the first two similes can accurately express the machine-

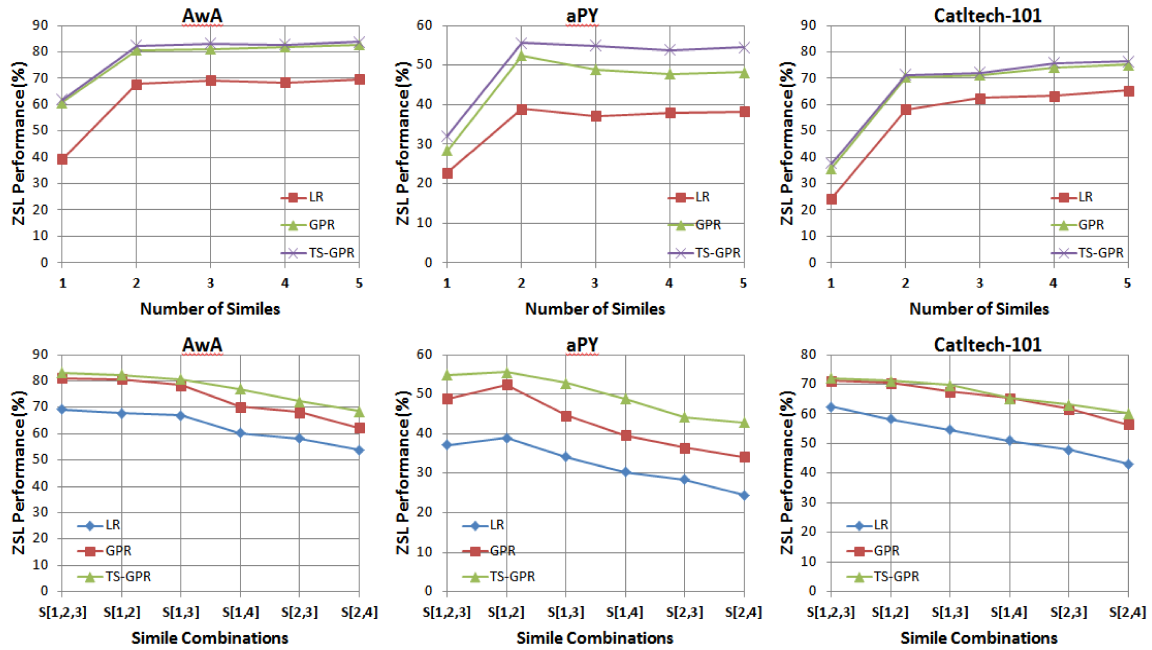


Fig. 7.8 Simile error tolerance: ZSL performance *w.r.t.* different combinations (upper). Amount of supervision: ZSL performance *w.r.t.* different numbers of similes (Lower). Best viewed in colours.

visual similarities. The error of the simile order does not degrade the performance too much because the top-5 classes are often highly correlated and result in close MKE values. Details are discussed in later experiments.

7.4.2 Compared to Published Results

For all of the three datasets, our compared results are achieved using **only top-2** similes for each unseen class. As shown in Table 7.3, our method significantly outperforms existing state-of-the-art ZSL methods. Among various prior auxiliary information, methods

Table 7.5 Averaged inference time (s) for each unseen class .

	# Similes	1	2	3	4	5
AwA	LR	0.83	0.92	1.31	1.75	1.89
	GPR	863.16	1206.48	1378.5	1463.8	1675.2
	TS-GPR	0.74	0.82	0.99	1.48	1.78
aPY	LR	0.37	0.41	0.47	0.58	0.65
	GPR	289.32	308.76	315.8	334.74	365.42
	TS-GPR	0.41	0.45	0.52	0.64	0.71
Caltech-101	LR	0.77	0.82	0.94	1.05	1.15
	GPR	147.72	157.83	166.65	178.28	195.47
	TS-GPR	0.67	0.79	0.97	1.14	1.35

Table 7.6 Compared to state-of-the-art methods on ZSL and GZSL scenarios. Performances (%) under different Auxiliary Information (AI) are compared, in terms of MKE and Attributes.

Scenario	Animals with Attributes (AwA)				Attribute Pascal and Yahoo (aPY)				Caltech-101	
	ZSL		GZSL		ZSL		GZSL		ZSL	GZSL
AI	MKE	Attributes	MKE	Attributes	MKE	Attributes	MKE	Attributes	MKE	MKE
DAP[72]	50.92	45.88	2.91	0.92	37.92	34.15	6.87	5.22	30.6	2.2
SJE[5]	74.81	66.25	16.80	13.93	38.24	33.82	9.62	4.84	36.3	4.9
ESZSL[116]	78.34	59.72	12.25	7.85	44.66	39.12	7.84	3.24	58.9	10.3
LatEm[141]	69.25	55.85	14.77	8.86	40.97	35.77	6.85	1.27	47.9	8.5
Ours	82.28	73.88	19.92	6.99	55.62	40.28	11.27	4.43	71.2	15.3

using attributes demonstrate favourable performance compared to the automatic models, such as Word2Vec [5] or skip-gram [8]. Our simile-based ZSL also outperforms that using transductive information which assumes unsupervised unseen data can be obtained during training. We ascribe our success to that our proposed MKE representation can smoothly express semantic similes without much information loss.

Caltech 101 Revisit Our comparison on Caltech 101 is summarised in Table 7.4. One of the reasons to use Caltech 101 is that it has been widely adopted by many classical kernel methods [15, 50, 127]. We follow the conventional settings that use 15 or 30 images in each class for training. Our method under ZSL scenario exceeds the performances of most of the previous kernel methods under supervised scenarios. Although these results may not be the state-of-the-art now, the experiment can still provide meaningful comparisons, in terms of the low-level features, kernel techniques. Such results indicate our method can help many existing systems find a feasible solution for ZSL problems without providing extra attributes or other forms of expensive auxiliary information.

7.4.3 Detailed Analysis

To understand the success of our method, we study the impacts of the following key aspects by extensive experiments.

Amount of Supervision In the upper session of Fig. 7.8, we demonstrate our ZSL performances *w.r.t.* different number of similes. For all of the three datasets, using top-2 similes can achieve relatively stable performance. Different from the expectation, adding more similes as clues can even harm the performance on aPY. This is because there is a large variety of classes while the number of seen class is small. More similes of irrelevant classes cannot benefit the results. Similar evidence can be found in Table 7.2, where the hit rates after the third simile decrease severely. In practice, it is also easier for human annotators to give fewer similes. Therefore, the experiments suggest that using two or three similes for each unseen class is enough for these small datasets. Our rest experiments are carried out using

top-2 similes.

Simile Error Tolerance As discussed earlier, the first two similes can achieve high hit rates, but can hardly guarantee the correct orders. The raised question is: how much performance degradation is caused by giving incorrect similes? In the lower session of Fig. 7.8, we show how our ZSL performance varies with different combinations of similes, which assumes less similar classes are used as top similes by mistake. It can be seen that alternating the first three similes will not cause a severe performance drop. The rationale behind this can be explained by Eq. 7.4. Two seen classes that are both similar to the same unseen class will result in close unseen-to-seen distances. Therefore, such degree of simile errors can be tolerated.

Efficiency From the above experiments, our proposed TS-GPR and conventional GPR achieve similar results, which are both remarkably higher than that of using linear regression (LR). However, it is widely acknowledged that kernel methods, such as GPR, have very high computational cost. In our TS-GPR model, we only consider the points close to the predicting one, which subversively reduced the inference time. As shown in Table 7.5, the computational cost of conventional GPR model using the whole training data is nearly a thousand times higher than the proposed TS-GPR model while the resultant ZSL performance is worse than ours.

Visual MKE versus Semantic MKE There are two usages of the proposed MKE representations. The first is to use MKE as a kernelised visual feature vector, which can improve the robustness of the representation. In order to understand the contribution to ZSL performance, we evaluate our MKE using the recent novel evaluation metric [22] that is proposed to estimate the upper bound of the expected performance empirically. In our case, the ZSL performance using real unseen prototype can be viewed as the upper bound of that using perfectly inferred prototypes. We use the mean of the original visual features in each unseen class as a prototype and compare to that using the mean vector of MKE representations. We then separately use test instances in raw and MKE feature spaces to find their nearest neighbours from the corresponding prototype lists as predictions. Table 7.7 shows the accuracies using MKE representations on both ZSL and Generalised ZSL (GZSL) tasks are significantly higher than that using raw features. Furthermore, we use MKE as visual features and train SVM models for ZSL using conventional attributes as DAP, which achieves 73.88 and 40.28 on AwA and aPY, respectively. However, our MKE fails to train DAP for GZSL tasks.

MKE can also be viewed as a real-valued attribute vector. Each dimension corresponds to a seen class and the value indicates the strength of that attribute being present in the unseen class. We implement state-of-the-art methods using MKE as attributes and com-

Table 7.7 Comparing the performance upper bounds (%) of ZSL and GZSL using MKE and raw features.

Scenario	AwA		aPY		Caltech-101	
	ZSL	GZSL	ZSL	GZSL	ZSL	GZSL
Raw	92.33	19.87	80.64	12.85	67.8	26.4
MKE	96.83	29.62	91.88	19.26	85.2	35.8

pare the results of using conventional attributes. In Table 7.6, we can observe that using MKE as semantic representations can also benefit the compared ZSL methods (MKE versus Attributes).

Generalised Zero-shot Learning In ZSL, the test instance is assumed from unseen classes only. Generalised ZSL (GZSL) breaks this unrealistic assumption and extends the potential label space to both seen and unseen classes. The results are summarised in Table 7.6 and great performance gap between ZSL and GZSL can be observed. One of the reasons is that the label space size is often double or triple of that in ZSL scenarios. Another reason is that there is a trend making unseen instances be classified towards seen classes [72]. For both ZSL and GZSL, our method outperforms all of the compared methods using MKE representations.

7.5 Conclusions

This chapter has proposed to address ZSL problems using a few similes of seen classes. MKE has been shown as a promising visual representation with high robustness. It can also benefit existing ZSL methods when using it as an attribute vector. Most importantly, MKE can directly bridge the gap between visual features and semantic similes. Using two similes for each unseen class, reliable unseen class prototypes can be inferred by the proposed efficient TS-GPR models, which have led to the state-of-the-art performance on both ZSL and GZSL tasks. The proposed method has significantly reduced the annotation cost and has made it feasible to apply ZSL on non-attribute tasks, *e.g.* Caltech 101. Future work includes widely applying ZSL using cheap similes. One of the challenges is how to apply similes for large-scaled tasks.

Chapter 8

Conclusion and Future Work

This thesis thoroughly studied the problem of Zero-shot Image Classification. As the demand of real intelligent system is increasing, ZIC has become an urgent and inevitable issue, especially for the explosive big-data era. Beyond image classification, Zero-shot Learning was also introduced as a high-level life-long learning model, as shown in Fig. 1.3. Motivated by these ambition, the thesis concluded four fundamental technologies. Intelligent visual system and machine learning were viewed as solid modelling problems while knowledge representation and ontological engineering focused more on understanding and organising human knowledge so as to explicitly direct the goal of machine learning. Chapter 3 to Chapter 5 mainly addressed the modelling issues and converted the ZIC problem into conventional image classification. Chapter 6 and 7 extended the knowledge representation from attributes to semantic similes, based on which a new ontology was proposed to remarkably boost the performance. Specific highlights are summarised as follows.

8.1 Learning and Data Synthesis

Chapter 3 concluded that the visual-semantic ambiguity is a common issue in ZSL tasks. Experimental results supported that ambiguity removal can significantly benefit the recognition performance. The proposed VSAR was a unified framework that can incorporate various semantic inputs. The regularisation term of the spectral graph was essential to mitigate the heterogeneous problem between visual and semantic spaces, through which different views of information got aligned in the intermediate embedding space. From the experiments, we can see that directly embedding using regression-based models can lead to low zero-shot recognition rates. Therefore, in chapter 4, we further push the embedding space backwards to the input visual feature space. we proposed a novel algorithm that can infer visual data for unseen classes using semantic attributes. The attributes were regarded as a full represen-

tation and embedded into the visual feature space. Using inferred visual features, we could convert the ZSL problem into conventional supervised classification and employ powerful classifiers for fine-grained open ZSL. On both standard and open ZSL scenarios, remarkable benefits were manifested by making classification using inferred visual features. The success of our method owed to the orthogonal embedding space that can jointly compromise the structural differences between visual and attribute spaces and remove the redundant correlations simultaneously. In chapter 5, we concerned three problems of visual data synthesis, in terms of imbalanced variances, over-fitting, and indiscrimination. The graph and orthogonal regularisation constraints are combined, which aims to infer higher-quality unseen visual features. The structure-preserving graph regularisation and the orthogonal embedding for redundancy removal are unified in a learning objective. Furthermore, the orthogonal embedding problem is reconsidered as information diffusion, which could make the synthesised data more discriminative. Our approach outperformed the state-of-the-art methods on all of the four benchmark datasets. Moreover, we challenged the GZSL on the large-scaled ImageNet. Also, attributes were substituted by automatic Word2Vec. For all of the scenarios, the proposed method achieved significant performance improvement over the compared state-of-the-art approaches.

8.2 Simile Ontology

In Chapter 6, we proposed a unified framework for ZSL including simile annotating, implicit attribute discovery, and the GSE model for ZSL classification. Our method achieved comparable results to state-of-the-art methods on the AWA dataset and significantly outperformed existing methods on aPY. We concluded our work as follows. Firstly, similes were effective to describe complex visual appearance. Grouping made similes more meaningful and discriminative for ZSL tasks. Secondly, our graph-cut algorithm could reliably capture the implicit attributes from similes and do not suffer from the correlation and training bias problems. Thirdly, our ensemble mechanism could find the most relevant simile groups during the test. As a result, the loss of accuracy from attribute prediction to ZSL recognition was reduced. However, the annotation cost was still quite high due to both training and test samples are required to be annotated by different groups of similes. To further improve, chapter 7 proposed to address ZSL problems using a few similes of seen classes. MKE has been shown as a promising visual representation with high robustness. It could also benefit existing ZSL methods when using it as an attribute vector. Most importantly, MKE could directly bridge the gap between visual features and semantic similes. Using two similes for each unseen class, reliable unseen class prototypes could be inferred by the proposed

efficient TS-GPR models, which have led to the state-of-the-art performance on both ZSL and GZSL tasks. The proposed method has significantly reduced the annotation cost and has made it feasible to apply ZSL on non-attribute tasks, Caltech 101. Future work included widely applying ZSL using cheap similes. One of the challenges was how to apply similes for large-scaled tasks.

8.3 Future Research Interests

For future work, a worthy attempt is to synthesise instance-level features so that the SVM-based framework can be widely applied. For another, our qualitative experiments gave positive results since we have shown the synthesised features are close to the real features in the same class. In the future, the synthesised data can be leveraged for more applications such as image retrieval or unseen image reconstruction. Also, how to address the inverted ZSL with a larger number of test classes requires further investigation.

In Chapter 5, some initial experiments have been conducted on the large-scaled ImageNet under the GZSL scenario. The performance was still far from the expectation for realistic applications. Intuitively, it might be an ill-posed problem to train a model on small source domain and test it on a much larger domain. It was like to make a pupil to build a rocket. A more realistic alternative could be to provide the minimal supervision for the source domain and let the machine freely explore the unsupervised test domain in an incremental semi-supervised approach. Different to conventional semi-supervised learning, the learning objective is not to learn the mapping from images to labels. Rather, the machine should learn the underlying rules that make things to be classified from human-knowledge view. New unlabelled data could gradually make the machine to build systematic connections to previously learnt concepts following the learnt rules. In this way, it could be possible to generate machine-view ontology that can be matched to large-scaled human -view ontology in the future.

The promising results of simile-based model supported the possibility of directly connecting visual concepts for inference tasks, such as ZIC. However, most of the existing work, such as ImageNet, is still based on semantic taxonomy. An interesting future direction could be how to leverage complex visual ontology, *e.g.* VisualGenome, to achieve human-level inference.

All of the introduced frameworks have specific criteria, *e.g.* regression loss. In the future, for a more complex problem, we may have no idea about what criterion should the machine to follow. To this end, reinforcement learning [68] is proposed to just give feedback about the result, *i.e.* whether good or bad. The machine can freely create any criterion

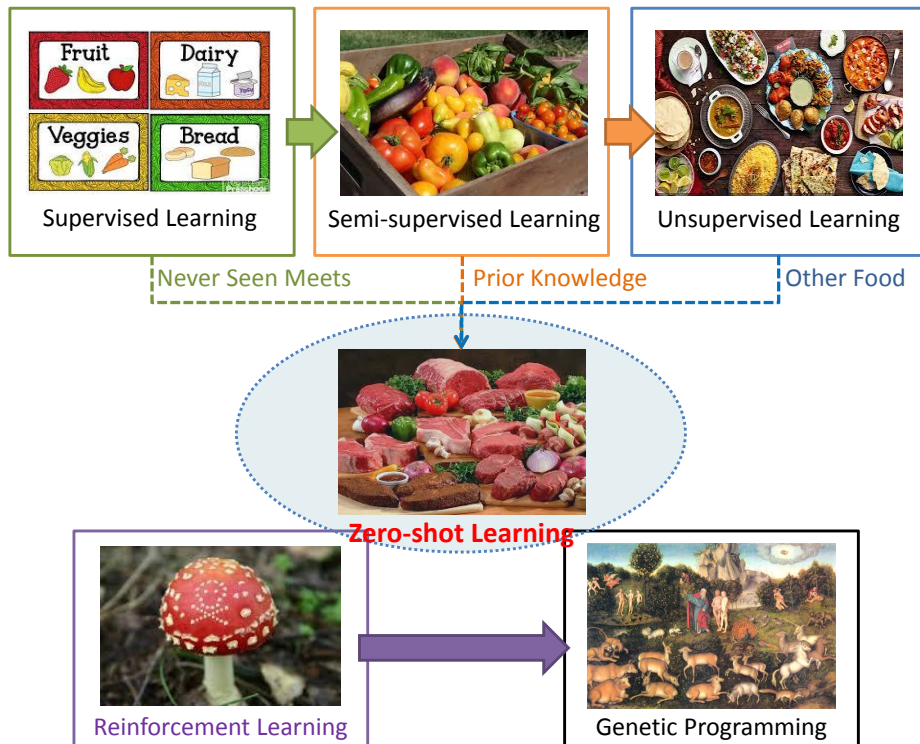


Fig. 8.1 Zero-shot Learning and other learning framework.

by itself at this stage, *e.g.* whether to include the toxic mushroom into food categories. Such mechanism enables machines to discover rules that can exceed human knowledge, *e.g.* the AlphaGo utilise reinforcement learning to play better than human players. Finally, if it is uncertain about what is right or wrong for human beings, Genetic Programming [11] could be the ultimate solution. Machines are competing under the natural rules, and the required judgement would have exceeded the intelligence of human beings.

In short, this thesis has summarised the feasibility of making ZIC on a closed small set of unseen classes. Larger open ZIC requires further development of both low-level learning technologies from the computer vision community and high-level ontological engineering from the deeper comprehension of the artificial intelligence.

References

- [1] Afridi, M. J., Ross, A., and Shapiro, E. M. (2017). On automated source selection for transfer learning in convolutional neural networks. *Pattern Recognition*.
- [2] Ahonen, T., Hadid, A., and Pietikäinen, M. (2004). Face recognition with local binary patterns. In *ECCV*.
- [3] Akata, Z., Malinowski, M., Fritz, M., and Schiele, B. (2016). Multi-cue zero-shot learning with strong supervision. In *CVPR*.
- [4] Akata, Z., Perronnin, F., Harchaoui, Z., and Schmid, C. (2013). Label-embedding for attribute-based classification. In *CVPR*.
- [5] Akata, Z., Reed, S., Walter, D., Lee, H., and Schiele, B. (2015). Evaluation of output embeddings for fine-grained image classification. In *CVPR*.
- [6] Al-Halah, Z., Gehrig, T., and Stiefelhagen, R. (2014). Learning semantic attributes via a common latent space. In *VISAPP*.
- [7] Al-Halah, Z. and Stiefelhagen, R. (2015). How to transfer? zero-shot object recognition via hierarchical transfer of semantic attributes. In *WACV*.
- [8] Al-Halah, Z., Tapaswi, M., and Stiefelhagen, R. (2016). Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning. In *CVPR*.
- [9] Antol, S., Zitnick, C. L., and Parikh, D. (2014). Zero-shot learning via visual abstraction. In *European conference on computer vision*, pages 401–416. Springer.
- [10] Arandjelovic, R. and Zisserman, A. (2013). All about vlad. In *CVPR*.
- [11] Banzhaf, W., Nordin, P., Keller, R. E., and Francone, F. D. (1998). *Genetic programming: an introduction*, volume 1. Morgan Kaufmann San Francisco.
- [12] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359.
- [13] Belhumeur, P. N., Hespanha, J. P., and Kriegman, D. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720.
- [14] Berg, T. L., Berg, A. C., and Shih, J. (2010). Automatic attribute discovery and characterization from noisy web data. In *ECCV*.

- [15] Boiman, O., Shechtman, E., and Irani, M. (2008). In defense of nearest-neighbor based image classification. In *CVPR*.
- [16] Bosch, A., Zisserman, A., and Munoz, X. (2007). Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 401–408. ACM.
- [17] Bucher, M., Herbin, S., and Jurie, F. (2016). Improving semantic embedding consistency by metric learning for zero-shot classification. In *ECCV*.
- [18] Caba Heilbron, F., Escorcia, V., Ghanem, B., and Carlos Niebles, J. (2015). Activi-tynet: A large-scale video benchmark for human activity understanding. In *CVPR*.
- [19] Cai, D., He, X., Han, J., and Huang, T. S. (2011). Graph regularized nonnegative matrix factorization for data representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1548–1560.
- [20] Changpinyo, S., Chao, W.-L., Gong, B., and Sha, F. (2016a). Synthesized classifiers for zero-shot learning. In *CVPR*.
- [21] Changpinyo, S., Chao, W.-L., and Sha, F. (2016b). Predicting visual exemplars of unseen classes for zero-shot learning. *arXiv preprint arXiv:1605.08151*.
- [22] Chao, W.-L., Changpinyo, S., Gong, B., and Sha, F. (2016). An empirical study and analysis of generalized zero-shot learning for object recognition in the wild.
- [23] Chen, C.-Y., Jayaraman, D., Sha, F., and Grauman, K. (2017). Divide, share, and conquer: Multi-task attribute learning with selective sharing. In *Visual Attributes*, pages 49–85. Springer.
- [24] Cheng, H.-T., Griss, M., Davis, P., Li, J., and You, D. (2013). Towards zero-shot learning for human activity recognition using semantic attribute sequence model. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM.
- [25] Cheng, Y., Qiao, X., Wang, X., and Yu, Q. (2017). Random forest classifier for zero-shot learning based on relative attribute. *IEEE Transactions on Neural Networks and Learning Systems*.
- [26] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.
- [27] Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27.
- [28] Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague.
- [29] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR*.

- [30] Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O. (2007). Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1052–1067.
- [31] Deng, J., Ding, N., Jia, Y., Frome, A., Murphy, K., Bengio, S., Li, Y., Neven, H., and Adam, H. (2014). Large-scale object classification using label relation graphs. In *ECCV*.
- [32] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *CVPR*.
- [33] Devin, C., Gupta, A., Darrell, T., Abbeel, P., and Levine, S. (2017). Learning modular neural network policies for multi-task and multi-robot transfer. In *ICRA*.
- [34] Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer.
- [35] Dollár, P., Rabaud, V., Cottrell, G., and Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 65–72. IEEE.
- [36] Elhoseiny, M., Saleh, B., and Elgammal, A. (2013). Write a classifier: Zero-shot learning using purely textual descriptions. In *CVPR*.
- [37] Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. (2009). Describing objects by their attributes. In *CVPR*.
- [38] Felzenszwalb, P., McAllester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *CVPR*.
- [39] Ferrari, V. and Zisserman, A. (2007). Learning visual attributes. In *NIPS*.
- [40] Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Mikolov, T., et al. (2013). Devise: A deep visual-semantic embedding model. In *NIPS*.
- [41] Fu, Y., Hospedales, T. M., Xiang, T., Fu, Z., and Gong, S. (2014a). Transductive multi-view embedding for zero-shot recognition and annotation. In *ECCV*.
- [42] Fu, Y., Hospedales, T. M., Xiang, T., and Gong, S. (2014b). Learning multimodal latent attributes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(2):303–316.
- [43] Fu, Y., Hospedales, T. M., Xiang, T., and Gong, S. (2015a). Transductive multi-view zero-shot learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(11):2332–2345.
- [44] Fu, Z., Xiang, T., Kodirov, E., and Gong, S. (2015b). Zero-shot object recognition by semantic manifold distance. In *CVPR*.
- [45] Gan, C., Lin, M., Yang, Y., de Melo, G., and Hauptmann, A. G. (2016a). Concepts not alone: Exploring pairwise relationships for zero-shot video activity recognition. In *AAAI*.

- [46] Gan, C., Lin, M., Yang, Y., Zhuang, Y., and Hauptmann, A. G. (2015). Exploring semantic inter-class relationships (sir) for zero-shot action recognition. In *AAAI*.
- [47] Gan, C., Sun, C., and Nevatia, R. (2017). Deck: Discovering event composition knowledge from web images for zero-shot event detection and recounting in videos. In *AAAI*.
- [48] Gan, C., Yang, T., and Gong, B. (2016b). Learning attributes equals multi-source domain generalization. *CVPR*.
- [49] Gong, Y. and Lazebnik, S. (2011). Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*.
- [50] Grauman, K. and Darrell, T. (2005). The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*.
- [51] Guo, Y., Ding, G., Han, J., and Gao, Y. (2017a). Zero-shot learning with transferred samples. *IEEE Transactions on Image Processing*.
- [52] Guo, Y., Ding, G., Han, J., and Gao, Y. (2017b). Zero-shot recognition via direct classifier learning with transferred samples and pseudo labels. In *AAAI*.
- [53] Habibian, A., Mensink, T., and Snoek, C. G. (2014). Videostory: A new multimedia embedding for few-example recognition and translation of events. In *ACMMM*.
- [54] Hinton, G., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.
- [55] Huang, S., Elhoseiny, M., Elgammal, A., and Yang, D. (2015). Learning hypergraph-regularized attribute predictors. In *CVPR*.
- [56] Isard, M. and Blake, A. (1998). Condensation—conditional density propagation for visual tracking. *International journal of computer vision*, 29(1):5–28.
- [57] Jayaraman, D. and Grauman, K. (2014). Zero-shot recognition with unreliable attributes. In *NIPS*.
- [58] Jayaraman, D., Sha, F., and Grauman, K. (2014). Decorrelating semantic visual attributes by resisting the urge to share. In *CVPR*.
- [59] Jégou, H., Douze, M., Schmid, C., and Pérez, P. (2010). Aggregating local descriptors into a compact image representation. In *CVPR*.
- [60] Jetley, S., Romera-Paredes, B., Jayasumana, S., and Torr, P. (2015). Prototypical priors: From improving classification to zero-shot learning. *arXiv preprint arXiv:1512.01192*.
- [61] Jia, X., Khandelwal, A., Nayak, G., Gerber, J., Carlson, K., West, P., and Kumar, V. (2017). Incremental dual-memory lstm in land cover prediction. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

- [62] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*.
- [63] Kankuekul, P., Kawewong, A., Tangruamsub, S., and Hasegawa, O. (2012). Online incremental attribute-based zero-shot learning. In *CVPR*.
- [64] Karessli, N., Akata, Z., Schiele, B., and Bulling, A. (2017). Gaze embeddings for zero-shot image classification. In *CVPR*.
- [65] Klaser, A., Marszałek, M., and Schmid, C. (2008). A spatio-temporal descriptor based on 3d-gradients. In *BMVC*.
- [66] Kodirov, E., Xiang, T., Fu, Z., and Gong, S. (2015). Unsupervised domain adaptation for zero-shot learning. In *ICCV*.
- [67] Kong, W. and Li, W.-J. (2012). Isotropic hashing. In *NIPS*.
- [68] Kotsiantis, S. B., Zaharakis, I. D., and Pintelas, P. E. (2006). Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26(3):159–190.
- [69] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., Bernstein, M., and Fei-Fei, L. (2016). Visual genome: Connecting language and vision using crowdsourced dense image annotations.
- [70] Kumar, N., Berg, A. C., Belhumeur, P. N., and Nayar, S. K. (2009). Attribute and simile classifiers for face verification. In *ICCV*.
- [71] Kumar, N., Berg, A. C., Belhumeur, P. N., and Nayar, S. K. (2011). Describable visual attributes for face verification and image search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(10):1962–1977.
- [72] Lampert, C. H., Nickisch, H., and Harmeling, S. (2009). Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*.
- [73] Lampert, C. H., Nickisch, H., and Harmeling, S. (2014). Attribute-based classification for zero-shot visual object categorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(3):453–465.
- [74] Larochelle, H., Erhan, D., and Bengio, Y. (2008). Zero-data learning of new tasks. In *AAAI*.
- [75] LeCun, Y. and Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361:310.
- [76] Lei Ba, J., Swersky, K., Fidler, S., et al. (2015). Predicting deep zero-shot convolutional neural networks using textual descriptions. In *ICCV*.
- [77] Li, A., Lu, Z., Wang, L., Xiang, T., and Wen, J.-R. (2017). Zero-shot scene classification for high spatial resolution remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*.

- [78] Li, F.-F., Rob, F., and Pietro, P. (2006). One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611.
- [79] Li, X. and Guo, Y. (2015). Max-margin zero-shot learning for multi-class classification. In *AISTATS*.
- [80] Li, X., Guo, Y., and Schuurmans, D. (2015). Semi-supervised zero-shot classification with label representation learning. In *ICCV*.
- [81] Liang, K., Chang, H., Shan, S., and Chen, X. (2015). A unified multiplicative framework for attribute learning. In *ICCV*.
- [82] Liu, C., Shang, Z., and Tang, Y. Y. (2017). Zero-shot learning with fuzzy attribute. In *CYBCON*.
- [83] Liu, W., Wang, J., Kumar, S., and Chang, S.-F. (2011). Hashing with graphs. In *ICML*.
- [84] Long, Y., Liu, L., and Shao, L. (2016a). Attribute embedding with visual-semantic ambiguity removal for zero-shot learning. In *BMVC*.
- [85] Long, Y., Liu, L., and Shao, L. (2017). Towards fine-grained open zero-shot learning: Inferring unseen visual features from attributes. In *WACV*.
- [86] Long, Y. and Shao, L. (2017). Describing unseen classes by exemplars: Zero-shot learning using grouped simile ensemble. In *WACV*.
- [87] Long, Y., Zhu, F., and Shao, L. (2016b). Recognising occluded multi-view actions using local nearest neighbour embedding. *Computer Vision and Image Understanding*, 144:36–45.
- [88] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *ICCV*.
- [89] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.
- [90] Madapana, N. and Wachs, J. P. (2017). A semantical & analytical approach for zero shot gesture learning. In *FG*.
- [91] Mahajan, D., Sellamanickam, S., and Nair, V. (2011). A joint learning framework for attribute models and object descriptions. In *ICCV*.
- [92] Manjunath, B. S., Ohm, J.-R., Vasudevan, V. V., and Yamada, A. (2001). Color and texture descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):703–715.
- [93] Mattivi, R. and Shao, L. (2009). Human action recognition using lbp-top as sparse spatio-temporal feature descriptor. In *Computer Analysis of Images and Patterns*, pages 740–747. Springer.
- [94] Mensink, T., Gavves, E., and Snoek, C. (2014). Costa: Co-occurrence statistics for zero-shot classification. In *CVPR*.

- [95] Mensink, T., Verbeek, J., Perronnin, F., and Csurka, G. (2012). Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *ECCV*.
- [96] Mensink, T., Verbeek, J., Perronnin, F., and Csurka, G. (2013). Distance-based image classification: Generalizing to new classes at near-zero cost. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2624–2637.
- [97] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *NIPS*.
- [98] Mitchell, T. M., Shinkareva, S. V., Carlson, A., Chang, K.-M., Malave, V. L., Mason, R. A., and Just, M. A. (2008). Predicting human brain activity associated with the meanings of nouns. *science*, 320(5880):1191–1195.
- [99] Mukherjee, T. and Hospedales, T. M. (2016). Gaussian visual-linguistic embedding for zero-shot recognition. In *EMNLP*.
- [100] Naphade, M., Smith, J. R., Tesic, J., Chang, S.-F., Hsu, W., Kennedy, L., Hauptmann, A., and Curtis, J. (2006). Large-scale concept ontology for multimedia. *MultiMedia, IEEE*, 13(3):86–91.
- [101] Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G. S., and Dean, J. (2014). Zero-shot learning by convex combination of semantic embeddings. In *ICLR*.
- [102] Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175.
- [103] Palatucci, M., Pomerleau, D., Hinton, G. E., and Mitchell, T. M. (2009). Zero-shot learning with semantic output codes. In *NIPS*.
- [104] Parikh, D. and Grauman, K. (2011). Relative attributes. In *ICCV*.
- [105] Patterson, G., Xu, C., Su, H., and Hays, J. (2014). The sun attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108(1-2):59–81.
- [106] Peng, P., Tian, Y., Xiang, T., Wang, Y., Pontil, M., and Huang, T. (2017). Joint semantic and latent attribute modelling for cross-class transfer learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- [107] Qi, G.-J., Liu, W., Aggarwal, C., and Huang, T. (2017). Joint intermodal and intramodal label transfers for extremely rare or unseen classes. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1360–1373.
- [108] Qiao, R., Liu, L., Shen, C., and Hengel, A. v. d. (2017). Visually aligned word embeddings for improving zero-shot learning. *BMVC*.
- [109] Qiao, R., Liu, L., Shen, C., and van den Hengel, A. (2016). Less is more: Zero-shot learning from online textual documents with noise suppression. In *CVPR*.

- [110] Qin, J., Liu, L., Shao, L., Shen, F., Ni, B., Chen, J., and Wang, Y. (2017). Zero-shot action recognition with error-correcting output codes. In *CVPR*.
- [111] Qin, J., Wang, Y., Liu, L., Chen, J., and Shao, L. (2016). Beyond semantic attributes: Discrete latent attributes learning for zero-shot recognition. *IEEE Signal Processing Letters*, 23(11):1667–1671.
- [112] Reed, S., Akata, Z., Lee, H., and Schiele, B. (2016). Learning deep representations of fine-grained visual descriptions. In *CVPR*.
- [113] Rohrbach, M., Ebert, S., and Schiele, B. (2013). Transfer learning in a transductive setting. In *NIPS*.
- [114] Rohrbach, M., Stark, M., and Schiele, B. (2011). Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *CVPR*.
- [115] Rohrbach, M., Stark, M., Szarvas, G., Gurevych, I., and Schiele, B. (2010). What helps where—and why? semantic relatedness for knowledge transfer. In *CVPR*.
- [116] Romera-Paredes, B. and Torr, P. (2015). An embarrassingly simple approach to zero-shot learning. In *ICML*.
- [117] Russakovsky, O. and Fei-Fei, L. (2012). Attribute learning in large-scale datasets. In *Trends and Topics in Computer Vision*, pages 1–14. Springer.
- [118] Shao, L., Liu, L., and Yu, M. (2016). Kernelized multiview projection for robust action recognition. *International Journal of Computer Vision*, 118(2):115–129.
- [119] Sharmanska, V., Quadrianto, N., and Lampert, C. H. (2012). Augmented attribute representations. In *ECCV*.
- [120] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905.
- [121] Siddiquie, B., Feris, R. S., and Davis, L. S. (2011). Image ranking and retrieval based on multi-attribute queries. In *CVPR*.
- [122] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint:1409.1556*.
- [123] Smeulders, A. W., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000). Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(12):1349–1380.
- [124] Socher, R., Ganjoo, M., Manning, C. D., and Ng, A. (2013). Zero-shot learning through cross-modal transfer. In *NIPS*.
- [125] Tian, T., Chen, N., and Zhu, J. (2017). Learning attributes from the crowdsourced relative labels. In *AAAI*.
- [126] Turk, M. A. and Pentland, A. P. (1991). Face recognition using eigenfaces. In *CVPR*.

- [127] Tuytelaars, T., Fritz, M., Saenko, K., and Darrell, T. (2011). The nbnn kernel. In *ICCV*.
- [128] Vedaldi, A., Gulshan, V., Varma, M., and Zisserman, A. (2009). Multiple kernels for object detection. In *ICCV*.
- [129] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408.
- [130] Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.
- [131] Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. (2011). The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.
- [132] Wang, H., Klaser, A., Schmid, C., and Liu, C.-L. (2011). Action recognition by dense trajectories. In *CVPR*.
- [133] Wang, J., Kumar, S., and Chang, S.-F. (2010). Semi-supervised hashing for scalable image retrieval. In *CVPR*.
- [134] Wang, P., Liu, L., Shen, C., Huang, Z., and Hengel, H. (2017a). Multi-attention network for one shot learning. In *CVPR*.
- [135] Wang, X. and Ji, Q. (2013). A unified probabilistic approach modeling relationships between attributes and objects. In *ICCV*.
- [136] Wang, Y., Kwok, J. T., Yao, Q., and Ni, L. M. (2017b). Zero-shot learning with a partial set of observed attributes. In *IJCNN*.
- [137] Wen, Z. and Yin, W. (2013). A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434.
- [138] Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., and Ma, Y. (2009). Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227.
- [139] Wright, S. and Nocedal, J. (1999). Numerical optimization. *Springer Science*, 35:67–68.
- [140] Wu, S., Bondugula, S., Luisier, F., Zhuang, X., and Natarajan, P. (2014). Zero-shot event detection using multi-modal fusion of weakly supervised concepts. In *CVPR*.
- [141] Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., and Schiele, B. (2016). Latent embeddings for zero-shot classification. In *CVPR*.
- [142] Xian, Y., Schiele, B., and Akata, Z. (2017). Zero-shot learning-the good, the bad and the ugly. *CVPR*.

- [143] Xie, S. and Philip, S. Y. (2017). Active zero-shot learning: a novel approach to extreme multi-labeled classification. *International Journal of Data Science and Analytics*, 3(3):151–160.
- [144] Xu, B., Bu, J., Lin, Y., Chen, C., He, X., and Cai, D. (2013). Harmonious hashing. In *IJCAI*.
- [145] Xu, X., Hospedales, T., and Gong, S. (2017a). Transductive zero-shot action recognition by word-vector embedding. *International Journal of Computer Vision*, pages 1–25.
- [146] Xu, X., Hospedales, T. M., and Gong, S. (2016). Multi-task zero-shot action recognition with prioritised data augmentation. In *European Conference on Computer Vision*, pages 343–359. Springer.
- [147] Xu, X., Shen, F., Yang, Y., Shao, J., and Huang, Z. (2017b). Transductive visual-semantic embedding for zero-shot learning. In *ICMR*.
- [148] Xu, X., Shen, F., Yang, Y., Zhang, D., Shen, H. T., and Song, J. (2017c). Matrix tri-factorization with manifold regularizations for zero-shot learning. In *CVPR*.
- [149] Yang, Y. and Hospedales, T. M. (2015). A unified perspective on multi-domain and multi-task learning. *ICLR*.
- [150] Yann, N. D., Gokhan, T., Dilek, H.-T., and Heck, L. (2014). Zero-shot learning for semantic utterance classification. In *ICLR*.
- [151] Yu, F., Cao, L., Feris, R., Smith, J., and Chang, S.-F. (2013). Designing category-level attributes for discriminative visual recognition. In *CVPR*.
- [152] Yu, M., Liu, L., and Shao, L. (2016). Structure-preserving binary representations for rgb-d action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 38(8):1651–1664.
- [153] Yu, X. and Aloimonos, Y. (2010). Attribute-based transfer learning for object categorization with zero/one training example. In *ECCV*.
- [154] Zhang, Y. C., Li, Y., and Rehg, J. M. (2017). First-person action decomposition and zero-shot learning. In *WACV*.
- [155] Zhang, Z. and Saligrama, V. (2015). Zero-shot learning via semantic similarity embedding. In *ICCV*.
- [156] Zhang, Z. and Saligrama, V. (2016a). Zero-shot learning via joint latent similarity embedding. In *CVPR*.
- [157] Zhang, Z. and Saligrama, V. (2016b). Zero-shot recognition via structured prediction. In *ECCV*.
- [158] Zhao, Z. and Liu, H. (2007). Spectral feature selection for supervised and unsupervised learning. In *ICML*.
- [159] Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2004). Learning with local and global consistency. In *NIPS*.

-
- [160] Zhou, D. and Burges, C. J. (2007). Spectral clustering and transductive learning with multiple views. In *ICML*.

Appendix A

Glossary

- **ZIC:** Zero-shot Image Classification, particularly refers to classify images from unseen classes that have no training data before the test.
- **ZSL:** Zero-Shot Learning, is a general term for classifying unseen classes, which could be images, actions, etc.
- **G-ZSL:** Generalised ZSL, assumes the test image may come from both seen and unseen classes. In contrast, conventional ZSL only considers unseen classes.
- **T-ZSL:** Transductive ZSL, assumes images of unseen images are available during training but in an unlabelled condition.
- **VSAR:** Visual-Semantic Ambiguity Removal, refers to the inconsistent structure between visual and semantic spaces.
- **OSVE:** Orthogonal Semantic-Visual Embedding, is an embedding algorithm that can map semantic representations into visual feature space while removing the redundant correlations in an intermediate orthogonal space.
- **UVDS:** Unseen Visual Data Synthesis, is an effective algorithm for synthesising unseen visual features from given attributes. The consistency is held by graph regularisation and the feature variety is increased by an orthogonal rotation.
- **MKE:** Match Kernel Embedding, is a high-level representation of images. Each dimension stands for the similarity to a seen class.
- **SIFT:** Scale-Invariant Feature Transform, firstly detects key points and then encode it into histograms with several different scales.

- SURF: Speed Up Robust Features, is another faster local visual descriptor.
- DoG: Difference of Gaussian, smooths images by convolution with Gaussian kernels and detects key points or edges.
- HOG: Histogram of Gradient, is an effective descriptor for edges and local shaps.
- PHOG: Pyramid Histogram of Gradient, extracts HOG features from different scales and regions of an image like a pyramid. The extracted features are concatenated as a global representation.
- LBP: Local Binary Pattern, thresholds local pixels into binary codes which are encoded by a histogram.
- PCA: Principal Component Analysis, is an unsupervised dimension-reduction method which maximise the variances in the reduced dimensions.
- LDA: Local Discriminant Analysis, is a supervised approach that can reduce the representation to the total number of classes minus one.
- SVM: Support Vector Machine, is a supervised classifier that finds soft-margin for binary classification.
- CNN: Constitutional Neural Network, normally consists of constitutional layers and pooling layers for feature extraction and loss fitting.
- DBN: Deep Belief Network, firstly pre-trains the model by Restricted Boltzmann Machine and then fine-tunes the model as conventional deep neural network.
- SAE: Stacked Auto-Encoder, is normally used for unsupervised deep learning. The objective is to reconstruct the input.
- DAP/IAP: Direct/Indirect Attribute Prediction, are two frameworks for ZSL.
- fMRI: functional Magnetic Resonance Imaging.
- AwA: Animal with Attributes, a famous benchmark for ZSL.
- aPY: attribute Pascal/attribute Yahoo, provides instance-level attributes for ZSL.
- SUN: SUN database for fine-grained ZSL scene classification.
- CUB: Caltech-UCSD Birds, for fine-grained ZSL bird classification.

Appendix B

Notation

This thesis tried to keep the mathematical notation consistent through the context. However, due to the focus on each chapter was quiet different, special notations were claimed within the chapters. The following explains the general notation of a ZSL problem.

For ZSL, the training set contains visual features, attributes, and seen class labels that are in 3-tuples: $(\mathbf{x}_1, \mathbf{a}_1, y_1), \dots, (\mathbf{x}_N, \mathbf{a}_N, y_N) \subseteq \mathbf{X}_s \times \mathbf{A}_s \times \mathbf{Y}_s$, where N is the number of training samples; $\mathbf{X}_s = [\mathbf{x}_{nd}] \in \mathbb{R}^{N \times D}$ is a D -dimensional feature space; $\mathbf{A}_s = [\mathbf{a}_{nm}] \in \mathbb{R}^{N \times M}$ is an M -dimensional attribute space; and $y_n \in \{1, \dots, C\}$ consists of C discrete class labels. During the test, the given attributes can be either *category-level* or *instance-level*. Given \hat{N} pairs of unseen instances with semantic attributes from \hat{C} discrete categories: $(\hat{\mathbf{a}}_1, \hat{y}_1), \dots, (\hat{\mathbf{a}}_{\hat{N}}, \hat{y}_{\hat{N}}) \subseteq \mathbf{A}_u \times \mathbf{Y}_u$, where $\mathbf{Y}_u \cap \mathbf{Y}_s = \emptyset$, $\mathbf{A}_u = [\mathbf{a}_{\hat{n}m}] \in \mathbb{R}^{\hat{N} \times M}$, the goal of zero-shot learning is to learn a classifier, $f: \mathbf{X}_u \rightarrow \mathbf{Y}_u$, where the samples in \mathbf{X}_u are completely unavailable during training. We use *Bold* typeface to indicate a space. Subscripts s and u refer to ‘seen’ and ‘unseen’. *hat* denotes the variables that are related to ‘test’ samples.

