Deep Learning Based Multimodal Retrieval

Jianan Chen

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

THÈSE DE DOCTORAT DE

L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *Signal, Image, Vision*

Par
Jianan CHEN

Deep Learning Based Multimodal Retrieval

Thèse présentée et soutenue à Rennes, le 17/10/2023
Unité de recherche : IETR
Thèse N° : 23ISAR 29 / D23 - 29

Rapporteurs avant soutenance :

Adrian MUNTEANU — Professeur des Universités, Vrije Universiteit Brussel (VUB)
Giuseppe VALENZISE — Chargé de Recherche, CNRS-CentraleSupelec-Université Paris-Saclay

Composition du Jury :

Président :	Didier COQUIN	Professeur des Universités, Université de Savoie Mont Blanc
Examinateurs :	Adrian MUNTEANU	Professeur des Universités, Vrije Universiteit Brussel (VUB)
	Giuseppe VALENZISE	Chargé de Recherche, CNRS-CentraleSupelec-Université Paris-Saclay
	Ewa KIJAK	Maître de conférences, Université de Rennes
Dir. de thèse :	Kidiyo KPALMA	Professeur des Universités, INSA Rennes
	Lu ZHANG	Maître de conférences, INSA Rennes

Acknowledgement

First and foremost, I would like to express my gratitude to Professor KPALMA Kidiyo and Professor ZHANG Lu for accepting me as their Ph.D. student, which has provided me with the opportunity to delve deeper into the realm of scientific research. Their patient guidance and exceptional academic expertise have allowed me to acquire invaluable knowledge that will benefit me throughout my lifetime.

I would like to thank Professor BAI Cong for his numerous clarifications, assistance, and guidance, both before and after my arrival in France.

I am grateful to Dr. WANG Qiong for her help and collaboration in our research endeavors.

I would like to express my appreciation to the teachers and fellow students of the CSC/UT-INSA program.

I am indebted to my parents for their unconditional love and support.

Last but not least, I want to thank my wife, Ding Anan, for her companionship and supervision. Her trust serves as my fuel, and her comfort is my refuge. I would also like to express my gratitude to my nine-month-old daughter, whose every smile is worth ten cups of instant coffee.

It is truly remarkable to witness the rapid development of the multimodal domain, especially during my Ph.D. student research. Each day brings forth new and exciting multimodal algorithms. As I draft this manuscript, I come across numerous emerging and game-changing multimodal works.

However, multimodal encounters several challenges, including uninterpretability, substantial computational resource demands, and the risk of spurious propagation. These issues need to be addressed in order to enhance the reliability and efficiency of multimodal systems. Naturally, every emerging discipline brings forth a multitude of challenges. At the end, I would like to present two quotes from books written on the time of rapid development era of computer graphics [1].

> The bad news is that we have still a long way to go.
> The good news is that we have still a long way to go.

RÉSUMÉ ÉTENDU

Introduction

Alors que le XXIe siècle a été le témoin de progrès remarquables dans la révolution de l'information, en particulier dans le domaine de l'intelligence artificielle (IA), les algorithmes informatiques ont atteint un stade sans précédent de développement. L'objectif ultime de ces algorithmes est d'améliorer l'efficacité du travail et les conditions de la vie humaine. Parmi les objectifs cruciaux poursuivis par de nombreux informaticiens figure la réduction des coûts de communication entre les humains et les ordinateurs.

Un des facteurs principaux contribuant au coût de la communication est la disparité dans le traitement de l'information entre les êtres humains et les ordinateurs. En tant qu'êtres humains, nous nous appuyons sur plusieurs sens pour traiter l'information. Notre vision nous permet de percevoir les images et les vidéos, notre ouïe nous permet de percevoir les sons et les autres bruits, et notre sens du toucher nous permet de ressentir les vibrations, les variations de température et d'autres sensations physiques. Surtout, nous possédons le langage comme moyen d'exprimer nos pensées. En revanche, les ordinateurs s'appuient principalement sur des états binaires représentés par des niveaux de tension haut ou bas pour transmettre l'information. Différents formats de données binaires ont été développés pour faciliter la communication entre les humains et les ordinateurs. Cependant, ces formats présentent souvent des différences significatives dans la représentation des mêmes concepts humains. L'entropie de l'information des données d'image et de texte représentant le concept de "un chien" dans un ordinateur peut différer de manière significative.

Les ordinateurs stockent des données sous diverses formes, telles que du texte, des images, du son, des vidéos et autres. Alors que les êtres humains possèdent la capacité cognitive innée de comprendre facilement le sens véhiculé par ces différentes modalités, les ordinateurs sont confrontés à un défi majeur connu sous le nom d'écart ou fossé de modalités. Pour combler cet fossé, il est nécessaire de développer des modèles multimodaux qui exploitent les techniques d'apprentissage automatique pour extraire des informations sémantiques à partir de données hétérogènes. En d'autres termes, les algorithmes multi-

modaux visent à réduire cette entropie de l'information des données multimodales.

L'apprentissage automatique est une sous-discipline de l'IA qui vise à atteindre une capacité de généralisation grâce à l'apprentissage des données. Le processus d'apprentissage implique la sélection de la structure et des algorithmes du modèle appropriés, ainsi que l'utilisation d'une grande quantité de données d'apprentissage pour régler et optimiser les paramètres du modèle. Différents types d'apprentissage automatique existent, notamment l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement. Indéniablement, l'apprentissage automatique englobe une large gamme de méthodologies, et notre attention se porte sur l'apprentissage profond. Comparés à d'autres types d'apprentissage automatique, les algorithmes multimodaux basés sur l'apprentissage profond leur capacité à traiter des ensembles de données plus importants, une plus grande variété de types de données, et possèdent des applications plus polyvalentes.

Plus précisément, nous utilisons des approches d'apprentissage profond pour traiter deux tâches multimodales : la recherche d'images et de textes (ITR) et la segmentation d'expressions référentielles (RES). Les images et le textes représentent les formes principales de stockage des données dans le monde numérique. Cependant, le fossé de modalité significatif entre eux présente un défi considérable pour établir des corrélations. Le principal objectif de la recherche d'images et de textes est de combler fossé. Dans le cas de la tâche de segmentation d'expressions référentielles, l'entrée consiste toujours en texte et en images. Cela signifie que la recherche d'images et de texte et la segmentation d'expressions référentielles peuvent partager des techniques de fusion multimodale et d'alignement inter-modal pour atteindre les objectifs de réduction du fossé sémantique entre les images et le texte. Cependant, au lieu de considérer l'ensemble de l'image comme l'objet cible pour la correspondance, cette tâche se concentre sur les objets à l'intérieur de l'image. Elle génère une carte prédite au niveau des pixels qui correspond aux objets inférés. La FIGURE 1 illustre les liens et les différences entre la recherche d'images et de textes et la segmentation d'expressions référentielles.

La recherche d'images et de textes trouve également de nombreuses applications dans le domaine médical. Bien que partage la même architecture multimodale pour l'extraction des caractéristiques que la recherche d'images et de textes, elle diffère en termes de représentation des caractéristiques et de fonction de perte. Il est important de noter que la recherche d'images et de textes fait face à un fossé de modalités plus important par rapport à la segmentation d'expressions référentielles (RES), principalement en raison de l'entropie de l'information variable des données entre les différentes modalités.

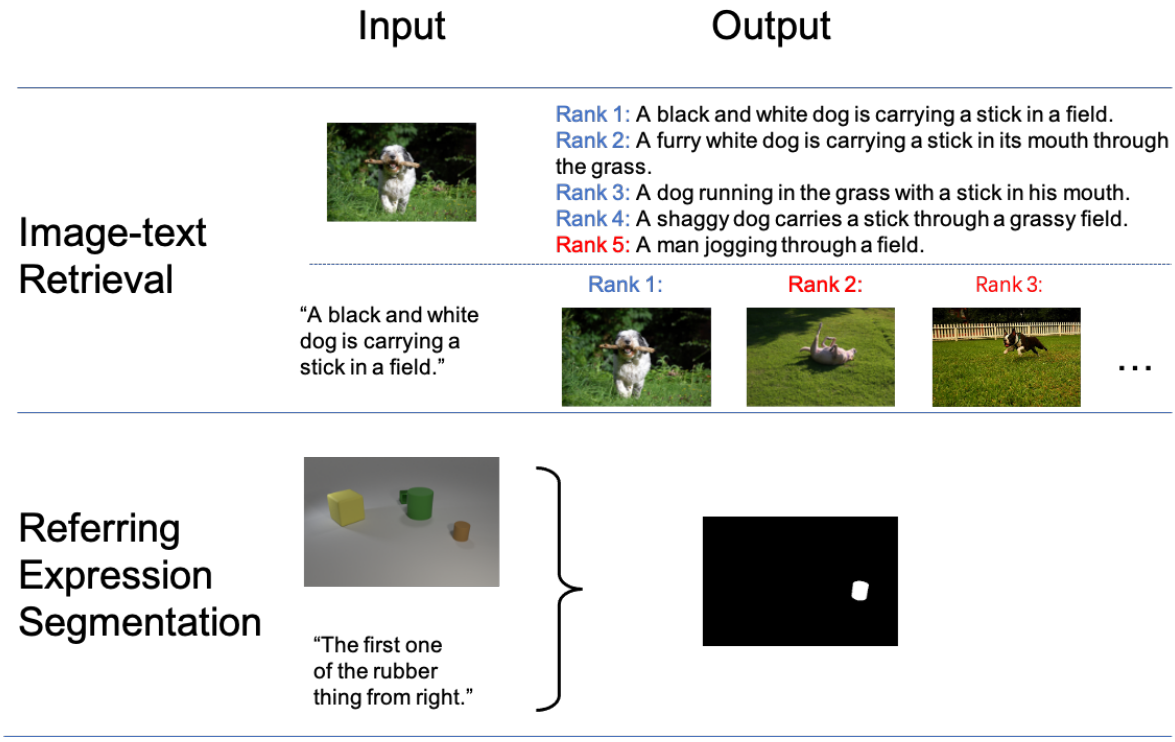


FIGURE 1 – Illustration des tâches de recherche d'images et de textes et de segmentation d'expressions référentielles. Dans la deuxième colonne, les deux tâches sont des paires image-texte en tant que données d'entrée. Dans la troisième colonne, la sortie de la recherche d'images par texte est le classement par pertinence, tandis que la sortie de la segmentation d'expressions référentielles est le masque segmenté des objets référents dans les images.

Une discussion plus approfondie sur les différences de tâches est présentée au début de la Partie III.

La recherche d'images et de textes ainsi que la segmentation d'expressions référentielles représentent des tâches fondamentales dans la recherche multimodale. Ces algorithmes possèdent un potentiel important et une utilité indéniable pour traiter des données hétérogènes et aborder des problèmes multimodaux complexes. La FIGURE 2 illustre l'application de la recherche d'images et de textes pour la recherche de formules moléculaires chimiques. Les formules moléculaires englobent de nombreuses variations structurelles, et certaines structures moléculaires posent des défis en termes d'identification visuelle.

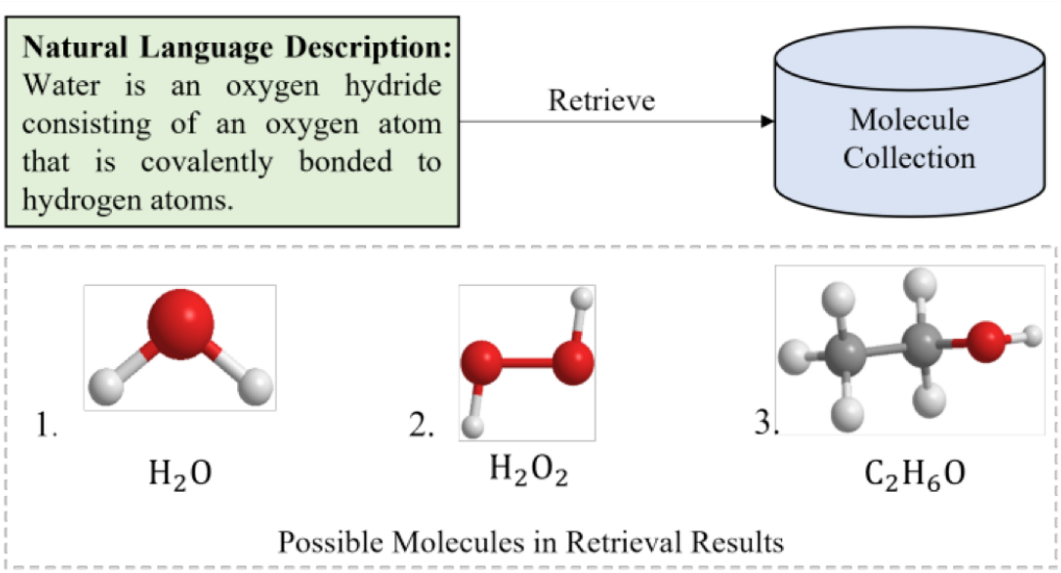


FIGURE 2 – Exemple de modèle de recherche d'images et de textes pour récupérer la structure des molécules [2].

En exploitant la recherche d'images et de textes, l'utilisation de descriptions textuelles devient possible, ce qui permet de réduire les complexités de recherche. L'analyse des images radiographiques consomme souvent une partie importante du temps d'un radiologue expérimenté. Cependant, grâce à la recherche multimodale d'images et de textes, la génération automatisée de rapports médicaux devient réalisable, ce qui entraîne une réduction substantielle de la charge de travail des radiologues, voir FIGURE 3.

La FIGURE 4 montre l'exemple de la segmentation multimodale croisée pour une vue de conduite. Dans cet exemple particulier, l'utilisation combinée des données LiDAR 3D et des images RVB facilite la prédiction simultanée des instances d'objets dans le champ de vision donné, ainsi que l'estimation précise du mouvement des objets dans une

Indication: ASTHMA

Findings: Lungs are clear . no pleural effusions or pneumothoraces. heart and mediastinum are stable with normal sized heart . degenerative changes in the thoracic spine

Impression: Clear Lungs

Baseline: The heart is normal in size . the lungs are clear . no pleural effusion or pneumothorax . no pleural effusion or pneumothorax . no pneumothorax .

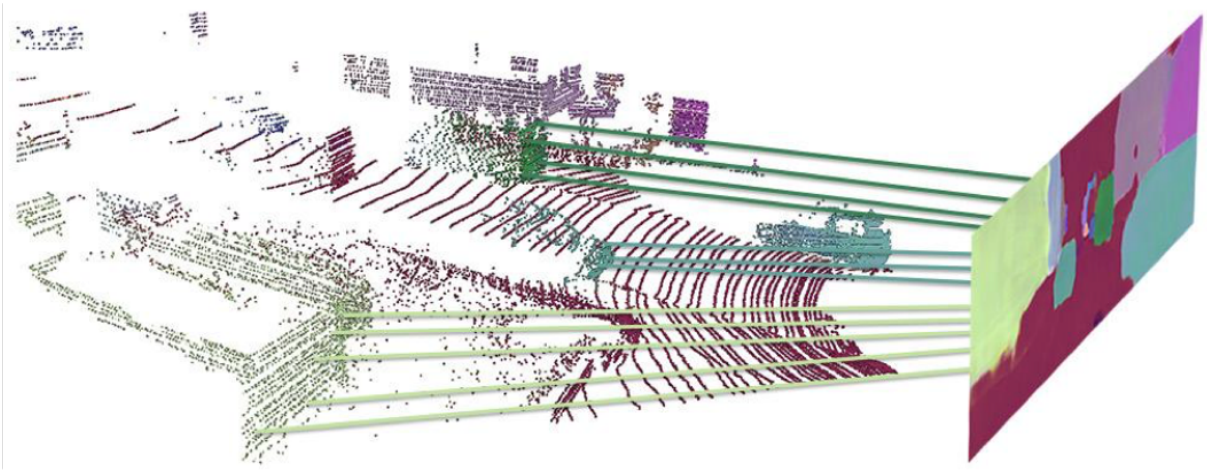
Relation-paraNet: The heart is normal in size and contour . no focal airspace consolidation . no pneumothorax or pleural effusion . degenerative changes in the thoracic spine

FIGURE 3 – Exemple du modèle de recherche d'image et de textes pour les rapports d'images médicales [3]. Trois résultats de rapports médicaux provenant du radiologiste, de la référence de base et de Relation-paraNet à partir de l'image radiographique thoracique requise à gauche.

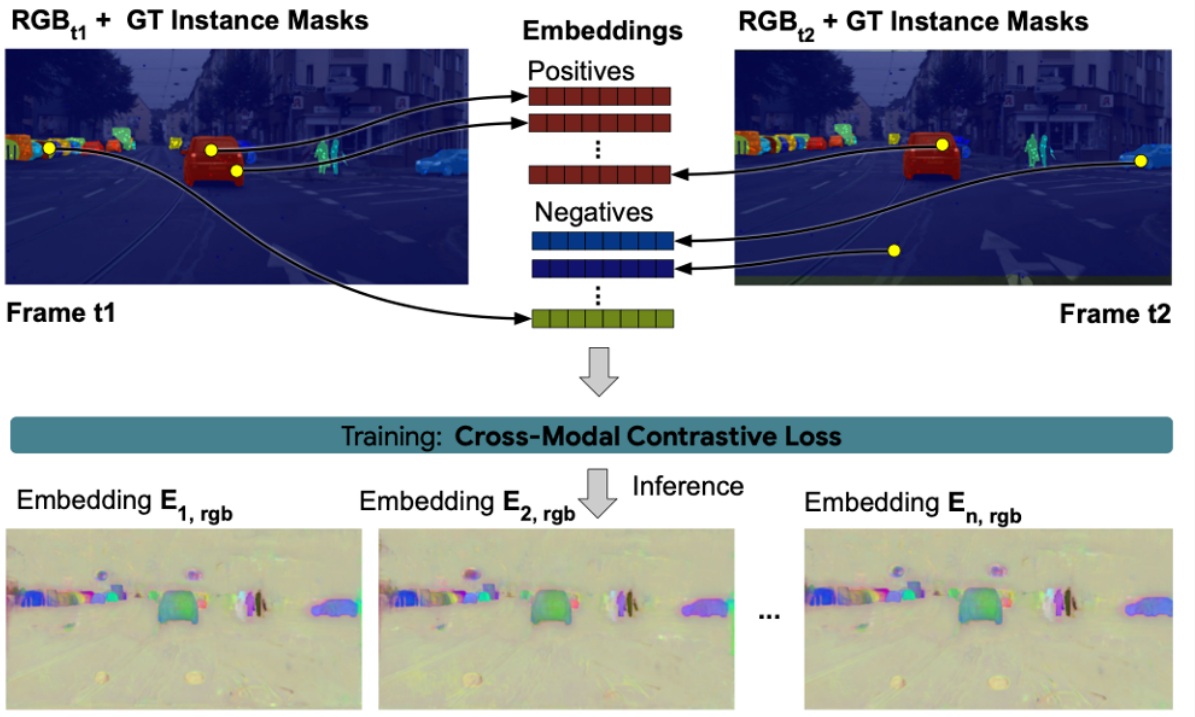
séquence temporelle. Ce modèle multimodal joue un rôle crucial dans l'amélioration des capacités analytiques des systèmes de conduite automobile, permettant ainsi des progrès significatifs vers l'atteinte de niveaux avancés de conduite autonome, tels que le niveau 3, niveau 4 et même le niveau 5.

La FIGURE 5 donne un exemple de segmentation par expression référentielle dans le domaine de l'imagerie médicale. RES s'avère être un outil précieux pour discerner avec précision les zones à l'intérieur des images médicales qui présentent une ressemblance avec d'autres régions. Cette capacité réduit considérablement les erreurs d'identification, améliorant ainsi la précision et la fiabilité globales du processus d'analyse.

Bien que les algorithmes multimodaux aient une large gamme d'applications et aient été entraînés avec des données spécialisées, ils présentent encore certaines limites. Au niveau de la tâche, l'un des problèmes est la difficulté de transférer les modèles entraînés vers des tâches alternatives et la demande élevée en énergie pour l'entraînement en raison du grand nombre de paramètres impliqués. Au niveau de l'algorithme, il existe des défis tels que le calcul exact des distances entre les caractéristiques dans la représentation multimodale des caractéristiques, l'ajustement chronophage de la distribution des caractéristiques, le manque de bases de données multimodales permettant d'obtenir des informations plus



(a) Nuages de points LiDAR 3D et image de segmentation 2D correspondante.



(b) Les données multimodales provenant de capteurs multiples sont utilisées pour entraîner la prédiction de la segmentation de conduite grâce à une fonction perte contrastive inter-modale.

FIGURE 4 – Exemple de segmentation inter-modale pour la vue de conduite [4].

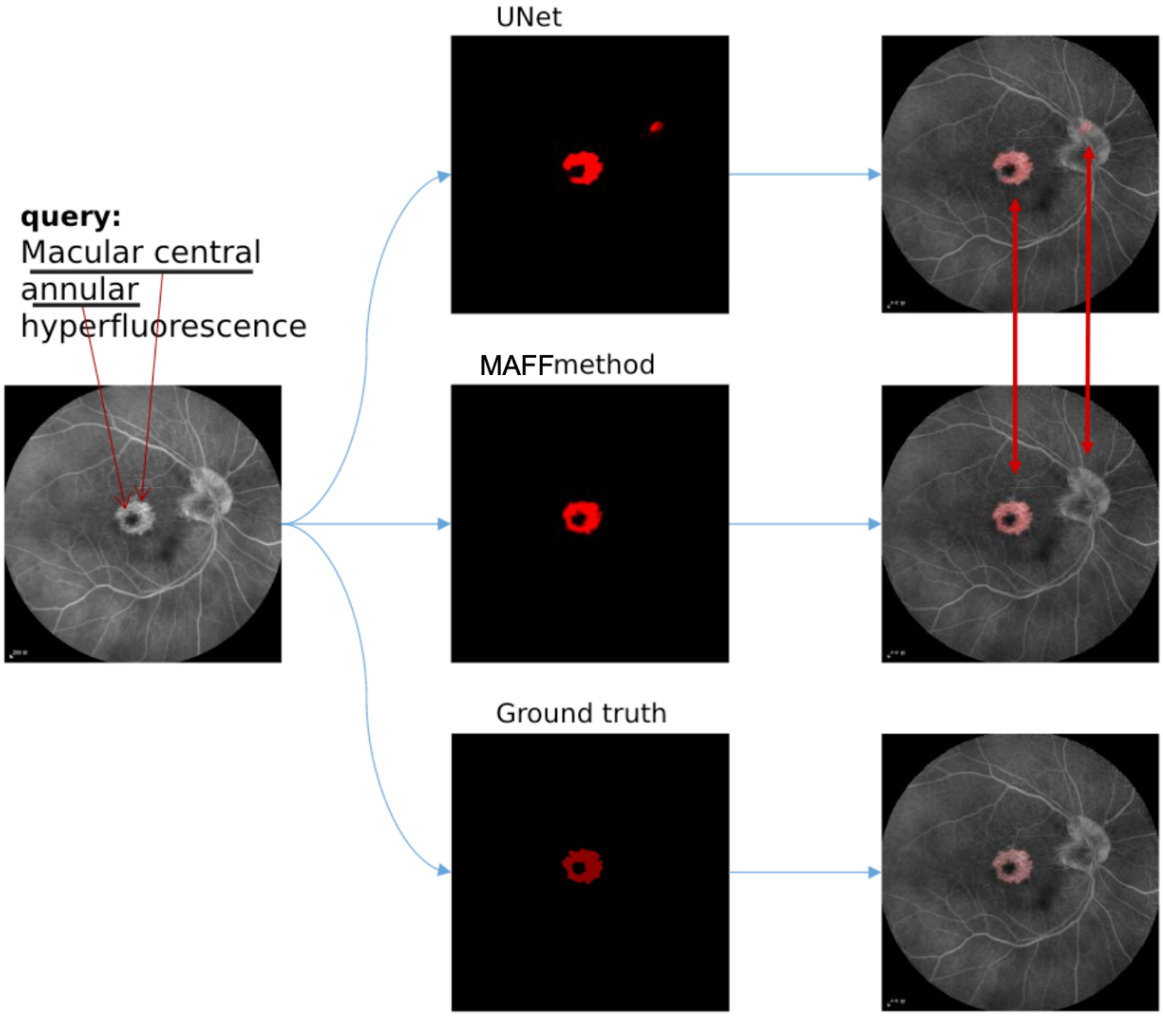


FIGURE 5 – Exemple de segmentation d'expression de référence pour une image médicale [5].

détaillées et le manque d'interprétabilité dans les modèles d'apprentissage profond.

Cette thèse se concentre principalement sur la résolution des problèmes au niveau algorithmique. Elle propose des améliorations de la fonction de perte des algorithmes de données multimodaux, la construction d'une nouvelle base de données multimodale, la proposition d'une nouvelle métrique pour mesurer la robustesse multi-view de la SOTA, et l'utilisation de techniques de visualisation pour analyser les capacités de compréhension sémantique de haut niveau des modèles multimodaux. Dans l'ensemble, les contributions visent à améliorer l'efficacité et l'interprétabilité des algorithmes multimodaux, améliorant ainsi leurs capacités de prise de décision dans des domaines spécialisés.

La structure de la thèse

Cette thèse se compose de trois parties :

La Partie I offre une perspective d'ensemble sur les approches de pointe dans les domaines multimodaux. Nous passons en revue les défis et les avancées en vision par ordinateur (CV, au chapitre 1) et en traitement du langage naturel (NLP, au chapitre 2), qui sont des domaines critiques dans le domaine multimodal. De plus, nous discutons des approches spécifiques visant à combler le fossé entre les modalités, en mettant l'accent sur l'intégration multimodale au chapitre 3.

La Partie II se concentre sur les tâches de recherche d'images et de textes multimodales. Dans le chapitre 4, nous catégorisons et comparons la structure des modèles SOTA actuels pour la recherche d'images et de textes. Au chapitre 5, nous proposons une nouvelle fonction de perte qui permet l'approche "l'apprentissage par paires", *i.e.*, la perte Intra-Modal Constraint (IMC). Grâce à une validation expérimentale sur des ensembles de données multimodaux disponibles publiquement, nous démontrons l'efficacité de notre fonction de perte IMC proposée pour améliorer les performances de la recherche d'images et de textes.

La Partie III explore la tâche de la segmentation des expressions référentielles (RES). Pour faciliter notre étude, nous utilisons une approche générative automatique pour construire une nouvelle base de données multimodale. Cette base de données offre plusieurs points de vue pour évaluer la stabilité des modèles SOTA pour la RES, évaluant ainsi leur capacité à comprendre l'information sémantique de haut niveau. Nous analysons et catégorisons les structures des modèles des approches SOTA, en mettant l'accent sur le rôle important des modèles d'attention dans la facilitation de l'alignement intermodal.

Pour mesurer la stabilité multi-vues des modèles multimodaux, nous établissons un banc d'essai multi-vues et des métriques. De plus, nous présentons les résultats des expériences comparatives afin d'améliorer la compréhension des modèles multimodaux.

Cette organisation du manuscrit de la thèse est représentée graphiquement sur la FIGURE 6 :

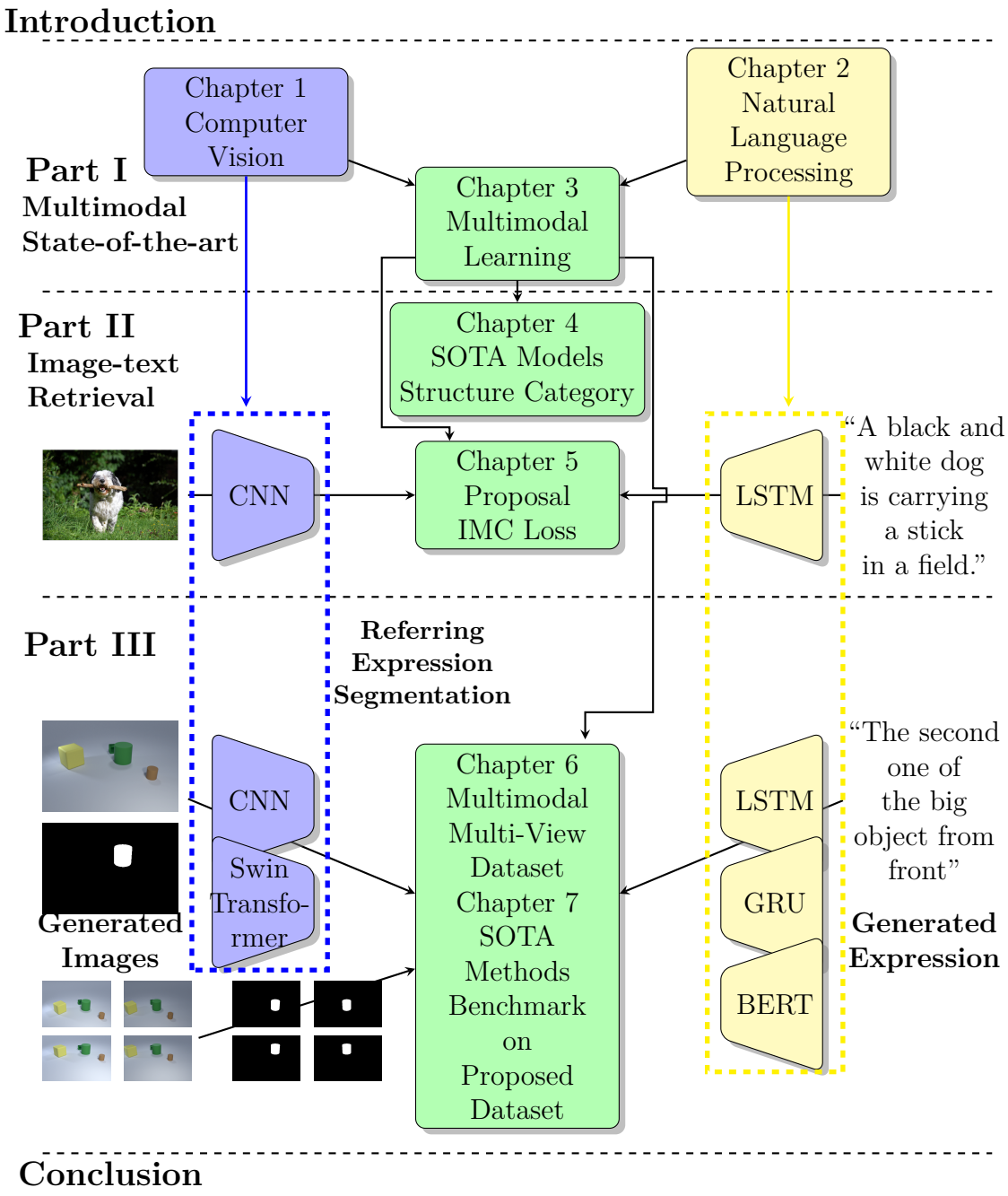


FIGURE 6 – Organisation du manuscrit de thèse. La figure est divisée en trois parties principales de haut en bas, avec les chapitres correspondants à chaque partie. La colonne de gauche de l'image représente les caractéristiques visuelles, la colonne de droite représente les caractéristiques textuelles, et les méthodes de fusion multimodale sont au milieu. Les flèches indiquent les relations entre les différents composants.

Table of Contents

INTRODUCTION

Context

While the 21st century has witnessed remarkable advancements in the information revolution, particularly in the field of computer artificial intelligence (AI), computer algorithms have reached an unparalleled stage of rapid development. The ultimate objective of these algorithms is to enhance the efficiency of human work and life. Among the critical goals numerous computer scientists pursue is reducing communication costs between humans and computers.

One primary factor contributing to the cost of communication is the disparity in information processing between people and computers. As humans, we rely on multiple senses for processing information. Our vision enables us to perceive image and video, our hearing allows us to perceive audio and other sounds, and our sense of touch enables us to feel vibrations, temperature variations, and other physical sensations. Most importantly, we possess language as a means to express our thoughts. In contrast, computers primarily rely on binary states represented by high and low voltage levels to convey information. Various binary data formats have been developed to facilitate communication between humans and computers. However, these formats often present significant differences in representing the same human conceptions. The information entropy of picture data and text data representing the concept of "a dog" in a computer can significantly differ. In other words, multimodal algorithms aim to reduce this information entropy of multimodal data.

Diverse forms of data, such as text, image, audio, video, and others, are stored in computers. While humans possess the innate cognitive ability to comprehend the meaning conveyed by these different modalities easily, computers face a significant challenge known as the multimodality gap. Bridging this gap requires developing multimodal models that leverage machine learning techniques to extract semantic information from heterogeneous data.

Machine learning is a subfield of AI that aims to achieve generalization capability through data training. The training process involves selecting the appropriate model

structure and algorithms, as well as utilizing a large quantity of training data to tune and optimize model parameters. Different types of machine learning exist, including supervised learning, unsupervised learning, and reinforcement learning. Undoubtedly, machine learning encompasses a wide range of methodologies, and our focus lies within the realm of deep learning. Compared to other types of machine learning, multimodal algorithms based on deep learning demonstrate the ability to handle larger datasets, a wider range of data types, and possess more versatile applications.

Specifically, we employ deep learning approaches to address two multimodal tasks: image-text retrieval (ITR) and referring expression segmentation (RES). Images and texts represent the predominant forms of data storage in the digital realm. However, the significant modality gap between them presents a substantial challenge in establishing correlations. Bridging this gap is the primary objective of image-text retrieval. In the case of the referring expression segmentation task, the input still consists of texts and images. It means that, the image-text retrieval and the referring expression segmentation tasks could share the multimodal fusion and cross-modal alignments techniques to achieve the goals of reduce the image and text semantic gap. However, instead of considering the entire image as the target object for matching, this task focuses on the objects within the image. It generates a pixel-wise predicted map that corresponds to the inferred objects. Figure 1 illustrates the connections and difference of image-text retrieval and the referring expression segmentation tasks.

Image-text retrieval also finds extensive applications in the medical field. While it shares the same multimodal architecture for feature extraction as image-text retrieval, it differs in terms of feature representation and loss function. Notably, RES faces a higher multimodalities gap compared to image-text retrieval, primarily due to the varying information entropy of data across different modalities. A more comprehensive discussion on the task differences is presented in the beginning of Part III.

Image-text retrieval and referring expression segmentation represent fundamental tasks in multimodal research. These algorithms possess significant potential and utility in handling heterogeneous data and addressing complex multimodal problems. Figure 2 illustrates the application of image-text retrieval to search for chemical molecular formulas. Molecular formulas encompass numerous structural variations, and certain molecular structures pose challenges in visual identification.

By employing image-text retrieval, the utilization of textual descriptions becomes possible, thereby mitigating retrieval complexities. Analyzing X-ray images often consumes

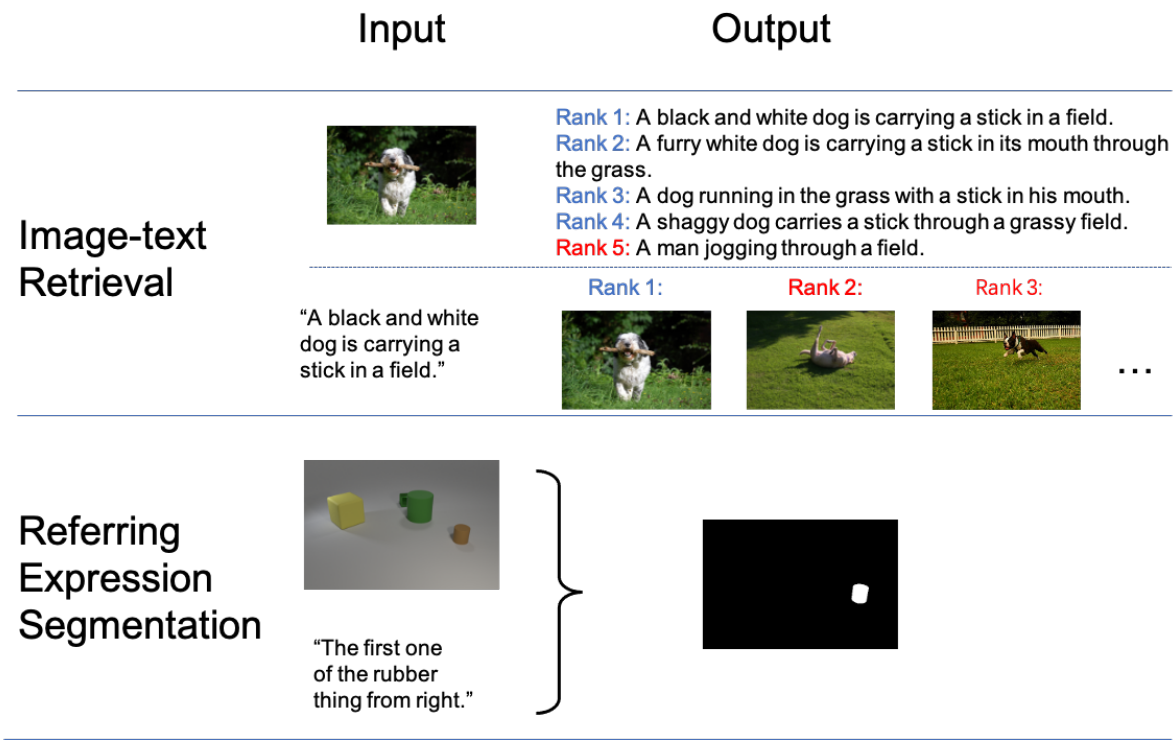


Figure 1 – Illustration of image-text retrieval and referring expression segmentation tasks. In the second column, both tasks have image-text pairs as input data. In the third column, the output of the image-text retrieval is the relevance ranking, while the referring expression segmentation output is the segmented mask map of the referring objects in the images.

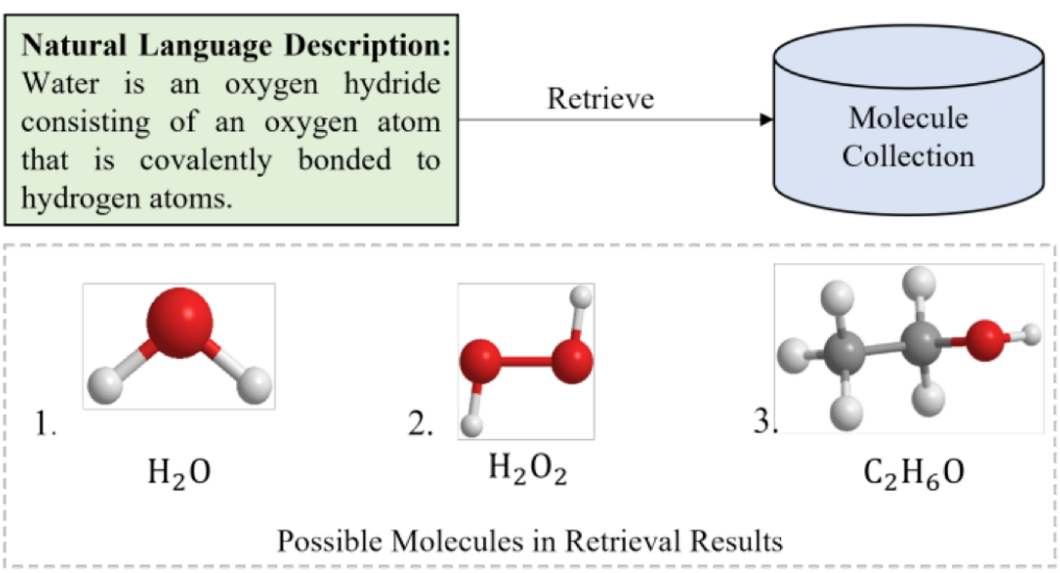


Figure 2 – Example of image-text retrieval model to retrieve molecules structure [2].

a significant portion of an experienced radiologist's time. However, through multimodal image-text retrieval, automated generation of medical reports becomes feasible, resulting in a substantial reduction in radiologist' workload, see Figure 3.

Figure 4 shows the example of cross-modal segmentation for drive view. In this particular example, the combined utilization of 3D LiDAR data and RGB images facilitates the concurrent prediction of object instances within the given field of view, as well as the accurate estimation of object motion in a temporal sequence. This multimodal model plays a crucial role in enhancing the analytical capabilities of car driving systems, thereby enabling significant progress towards achieving advanced levels of autonomous driving, *e.g.*, L3, L4, and even top L5.

Figure 5 gives an example of referring expression segmentation in medical image domian. RES prove to be a valuable tool in accurately discerning areas within medical images that bear resemblance to other regions. This capability significantly reduces the occurrence of false identifications, enhancing the overall precision and reliability of the analysis process.

Although multimodal algorithms have a wide range of applications and have been trained with specialized data, they still have certain shortcomings. One of the problems at the task level is the difficulty in transferring the trained models to alternative tasks and the high energy demand for training due to the large number of parameters involved. At the algorithm level, there are challenges such as inaccurate and time-consuming calculated

Indication: ASTHMA

Findings: Lungs are clear . no pleural effusions or pneumothoraces. heart and mediastinum are stable with normal sized heart . degenerative changes in the thoracic spine

Impression: Clear Lungs

Baseline: The heart is normal in size . the lungs are clear . no pleural effusion or pneumothorax . no pleural effusion or pneumothorax . no pneumothorax .

Relation-paraNet: The heart is normal in size and contour . no focal airspace consolidation . no pneumothorax or pleural effusion . degenerative changes in the thoracic spine

Figure 3 – Example of image-text retrieval model for medical image report [3]. Three medical report results of radiologist, baseline, and Relation-paraNet from the query chest X-Ray image on left.

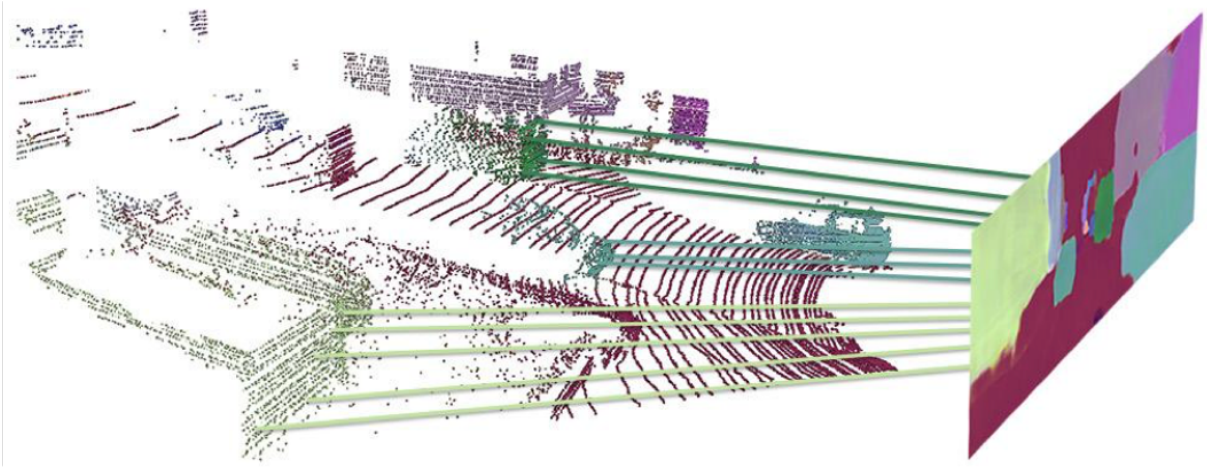
feature distances in multimodal feature representation, inadequate multimodal databases for obtaining finer-grained information, and the lack of interpretability in black box deep learning models.

This thesis primarily focuses on addressing the issues of inaccurate and time-consuming calculating multimodal features in common latent space at the algorithmic level. It proposes improvements to the loss function of multimodal data algorithms, constructing a new multimodal database, proposing a new metric to measure the SOTA multi-view robustness, and using visualization techniques to analyze multimodal models' high-level semantic understanding capabilities. Overall, these proposed modifications aim to enhance the effectiveness and interpretability of multimodal algorithms, thereby improving their decision-making capabilities in specialized domains.

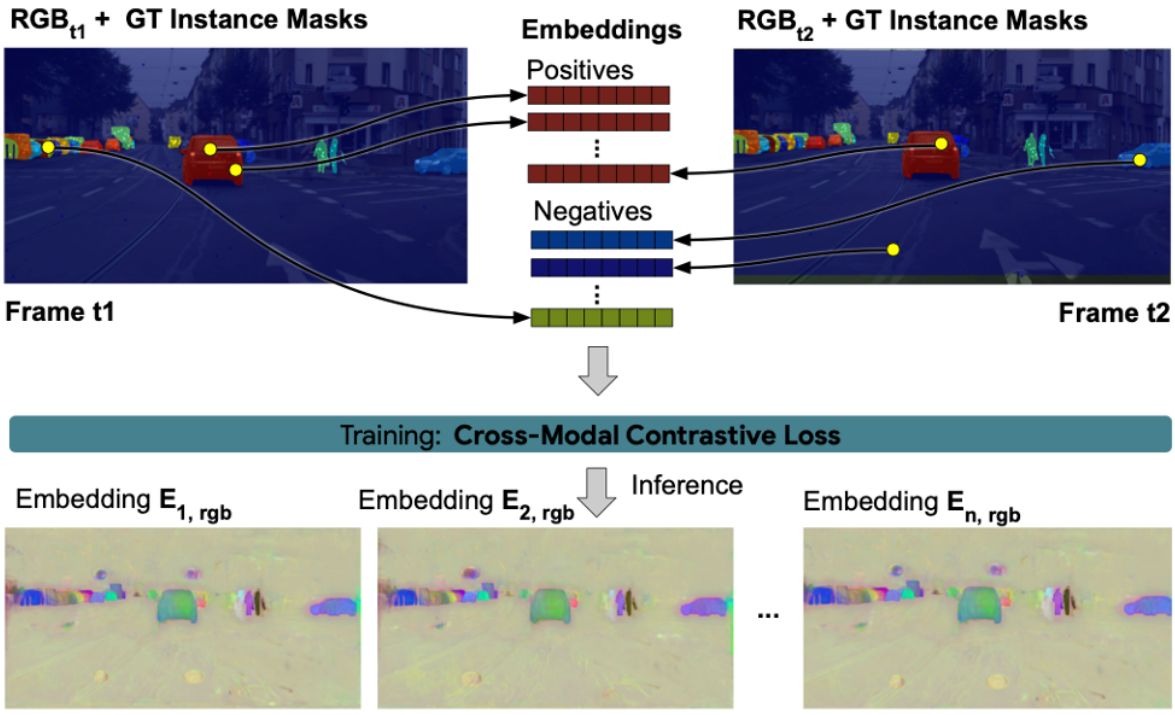
Structure of the thesis

This thesis consists of three parts:

Part I provides a high-bird perspective on the state-of-the-art (SOTA) approaches in multimodal domains. We review the challenges and breakthroughs in computer vision (CV, in chapter 1) and natural language processing (NLP, in chapter 2), which are critical



(a) 3D LiDAR point clouds and corresponding 2D segmentation image.



(b) Multimodal data from multiple sensors are trained to predict the drive segmentation via cross-modal contrastive loss.

Figure 4 – Example of cross-modal segmentation for drive view [4].

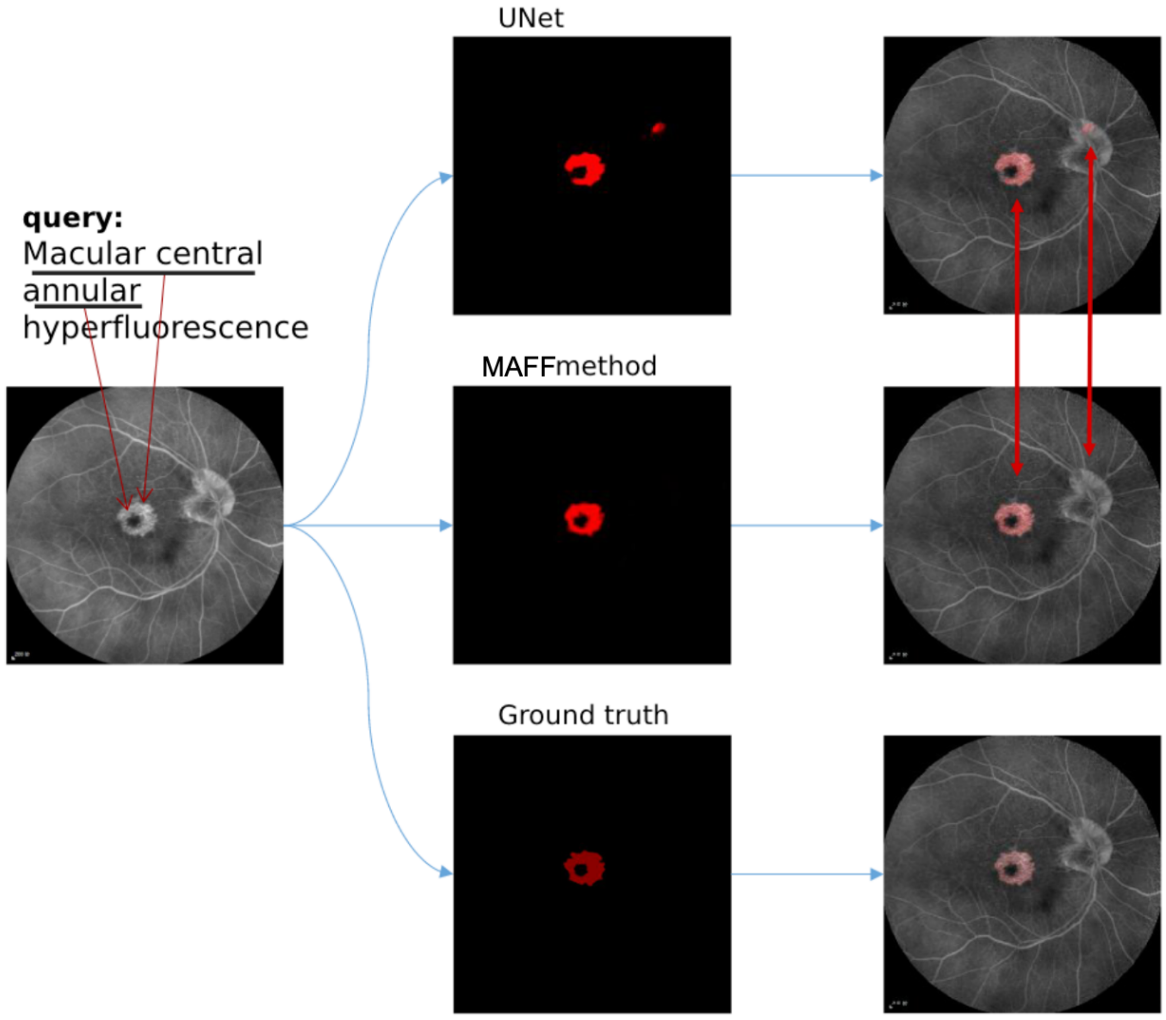


Figure 5 – Example of referring expression segmentation for medical image [5].

domains within multimodality. Additionally, we discuss specific approaches to bridge the gap between multimodalities, focusing on multimodal integration in chapter 3.

Part II focuses on multimodal image-text retrieval tasks. In the first chapter 4, we categorize and compare the present SOTA models' structure of image-text retrieval. In chapter 5, we propose a novel loss function that enables the "pairwise learning" approach, *i.e.*, IMC loss. Through experimental validation on publicly available multimodal datasets, we demonstrate the effectiveness of our proposed IMC loss function in improving the performance of image-text retrieval.

Part III explores the task of referring expression segmentation (RES). To facilitate our study, we employ an automatic generative approach to construct a new multimodal database. This database provides multiple views to assess the stability of SOTA models for RES, thus evaluating their capability to comprehend high-level semantic information. We analyze and categorize the model structures of SOTA approaches, emphasizing the significant role of attention models in facilitating cross-modal alignment. To measure the multi-view stability of multimodal models, we establish a multi-view benchmark and metrics. Furthermore, we visualize the results of comparative experiments to enhance the interpretability of multimodal models.

This thesis roadmap can be seen in Figure 6:

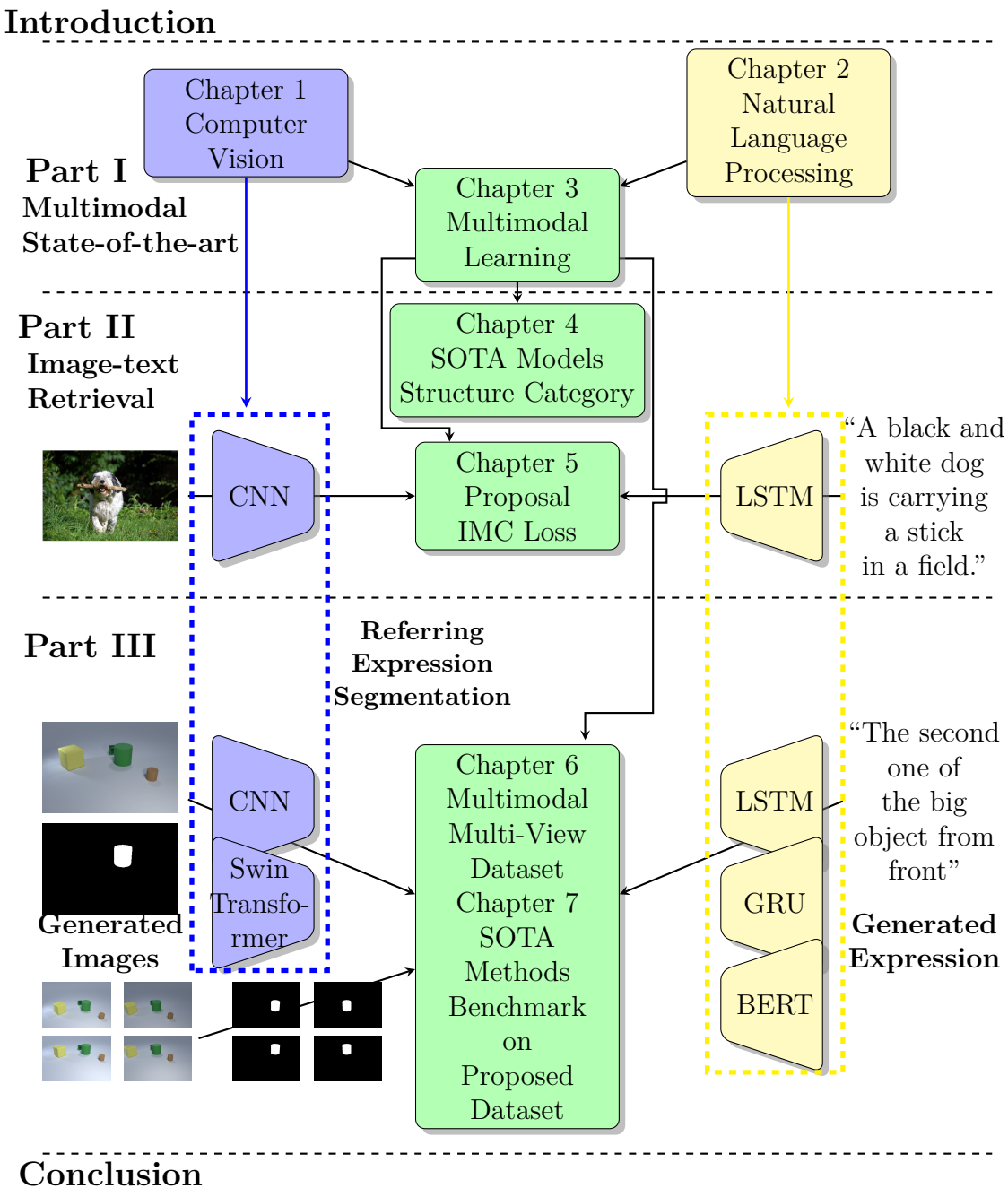


Figure 6 – Roadmap of this thesis. The figure is divided into three main parts from top to bottom, each part's corresponding chapters. The left column of the image represents visual features, the right column represents text features, and the multimodal fusion methods are in the middle. Arrows indicate components that contain relationships.

Part I

Multimodal State-of-the-Art

INTRODUCTION OF PART I

This Part I serves as a foundational part on multimodal algorithms based on deep learning. We begin with an overview of multimodal problems before proceeding to divide the problem into domain-specific subbranches. Examining the technological bottlenecks and breakthroughs encountered during the evolution of each of these domain branches, we make generalization of these key method points. By laying the foundation for these fundamental technologies, the various branches will eventually converge on two key areas: multimodal fusion (*i.e.*, feature representation) and cross-modal alignment (*i.e.*, multimodal loss function). This Part I will provide readers with a comprehensive understanding of multimodal models.

Multimodal Framework

Cross-modal information retrieval algorithms play a fundamental role in multimodal model learning. The core of a cross-modal information retrieval algorithm rely on representing multimodal features in a shared latent space and calculating the distance of information based on relevance. The algorithmic process involves several steps, including multimodal data preprocessing, feature extraction, feature fusion, feature representation, and loss function calculation.

In Figure 7, the steps of multimodal algorithm could be categorized by functional scope with different colors. Thus, the blue branch represents one kind of modality data in multimodal database, while the yellow branch deals with another kind of modality. The green parts stands for features fusion and latent space, which constructs a common space with different modality features. This latent space should not only accommodate distinct features but also project feature representations with different semantics to various locations based on task objectives. From a cognitive standpoint, the further to the left a figure is, the closer it is to low-level digital data, and the further to the right it is to high-level perception. In addition, this framework suggests that multimodal models should not only combine heterogeneous modal feature data but also bridge the semantic gap between low-level data and high-level cognition.

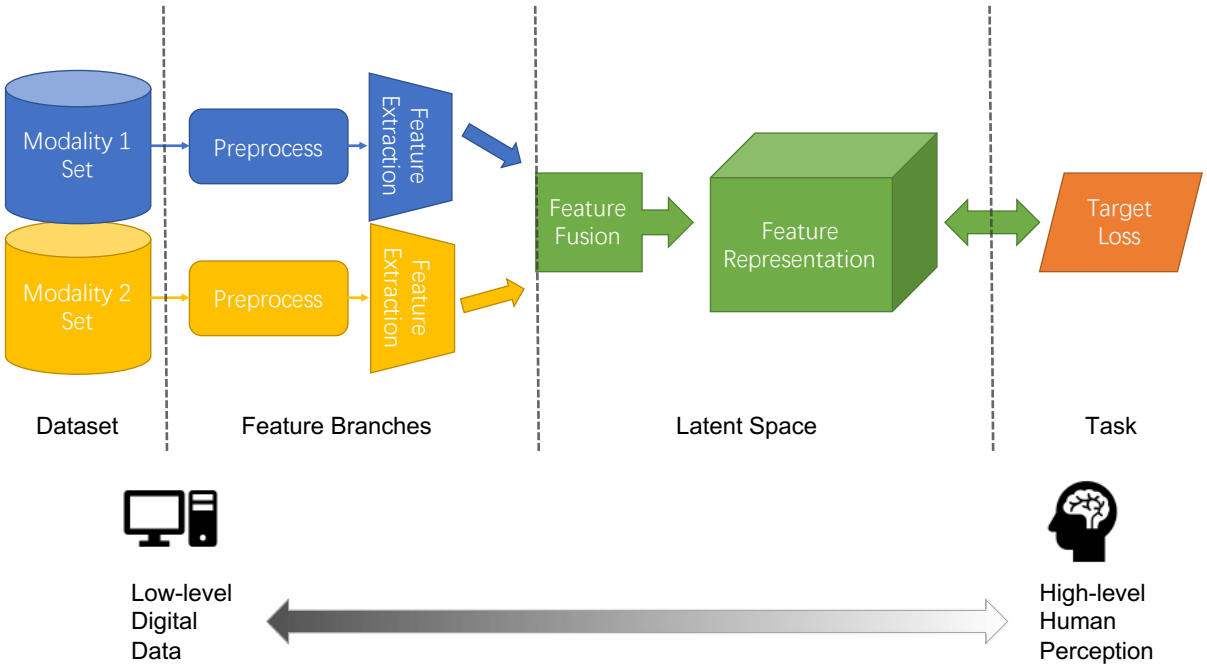


Figure 7 – A framework of multimodal algorithm.

Different modalities of data have varying levels of information entropy density, which affects the processing phases of multimodal information. For example, in computer hard disks, information is typically stored as bits, resulting in low information entropy density for all modalities. In contrast, the human brain processes information as simple cognitive concepts, and it is generally believed to have high information entropy density. In terms of information theory, multimodal processing is a process that rapidly reduces information entropy, especially in the final stage of multimodal processing. Multimodal learning is the desire to complete this entropy reduction process spontaneously by algorithms with minimal human intervention.

There are different sophisticated machine learning algorithms for processing different information structured data. Before the fusion phase, each branch closely corresponds to that of the unimodality task, and the most of multimodal algorithms employ domain-specific components from every single modal domain, *e.g.*, CNNs and RNNs. Unlike the unimodal model that can map information end-to-end, the multimodal latent space is closer to the semantic sense field, and this special property will be reflected in later parts of the chapters. In general, multimodal models use a combination of sophisticated unimodal approaches to deal with low-level features in different modalities, and then focus

on high-level semantic information using a special approach in the latent space.

In the rest of Part I, we present the algorithmic development in the three concerned subbranches: Computer Vision (CV), Natural Language Processing (NLP), and Multimodality Learning (MML), respectively.

Chapter 1

Computer Vision

Computer Vision (CV) concentrates on replicating portions of the complexity of our visual system and enabling computers recognize and analyze objects in images and videos similarly to humans. In the 1960s, Artificial Intelligence (AI) specialists believed that making computers see was comparable to a college student's summer project. Sixty years later, the problem is still not entirely resolved. The discipline of computer vision has evolved into a distinct field with strong connections to mathematics and computer science and weaker ties to physics, perceptual psychology, and neurology. Over the past few decades, computer vision technology has evolved rapidly. CV has significantly altered the way people live and work, especially since the rise of smartphones and online social media.

Computer scientists package the most extensive and urgent visual image processing requirements into a variety of CV tasks. For example, from traditional image classification, object detection, instance segmentation, to current image caption, image generation, style migration, and other novel CV tasks. Those CV tasks, in turn, have made major contributions to the advancement of computer science, particularly in Machine Learning (ML).

The field of machine learning encompasses a wide range of areas, and there are various ways to classify it from different perspectives. Depending on the data and task requirements, machine learning can be classified as supervised learning, unsupervised learning, self-supervised learning, reinforcement learning, and other categories. Based on the features of algorithms used, machine learning can also be categorized as handcrafted algorithms and neural network algorithms, among others. For instance, Janiesch et al.[6] divides ML into shallow machine learning and deep learning categories based on the scale of the algorithm hierarchy, see figure 1.1. Typically, shallow machine learning involves selecting components or parameters by handcraft. However, deep learning typically employs neural networks with a large number of neurons and backpropagation algorithms to determine parameter values.

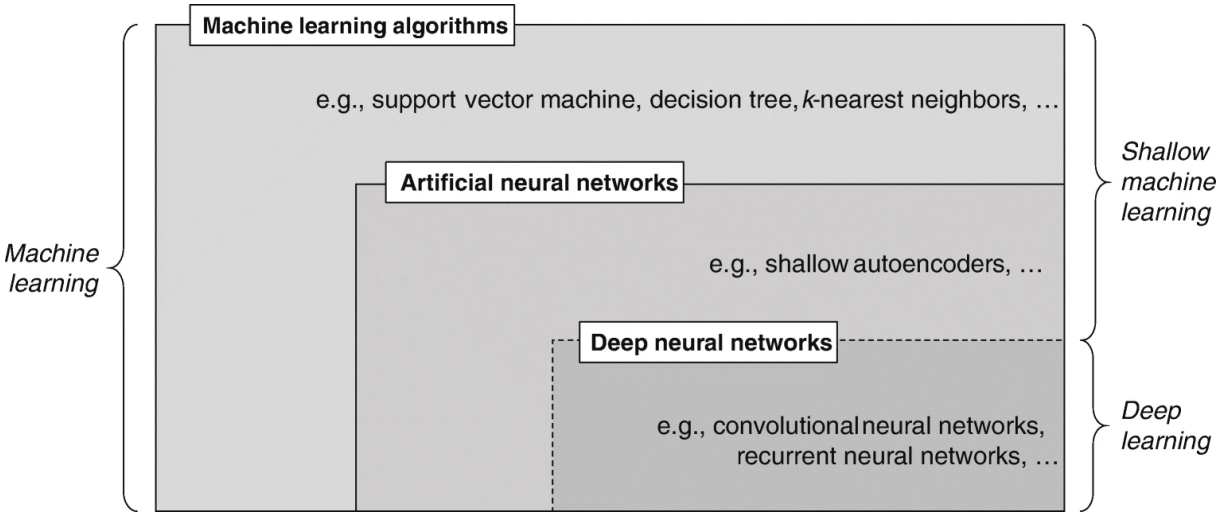


Figure 1.1 – Machine Learning (ML) approaches type [6].

1.1 Shallow Machine Learning in CV

In this section, our focus centers on shallow machine learning algorithms employed in computer vision machine learning. Within shallow machine learning, we emphasize two fundamental algorithms: Support Vector Machines (SVM) for supervised learning and k-means for unsupervised learning. These foundational algorithms serve as building blocks for the subsequent advancements in deep learning and multimodality. In fact, certain algorithms, such as the hinge loss in SVM and the bag-of-words model associated with k-means, are borrowed and further developed within the realm of deep learning and multimodal algorithms.

1.1.1 Support Vector Machine

Support vector machines (SVM [7]) are supervised learning models with associated learning algorithms that perform classification and regression analysis in machine learning. An SVM aims to construct a discriminatory hyperplane between data points of different classes. The input data is frequently projected into a higher-dimensional feature space for improved separation. A hyperplane is defined as a collection of points whose dot product with a vector in that space is constant. Such a set of vectors is an orthogonal (and hence

minimum) set of vectors that defines a hyperplane. SVM can be further divided into linear SVM and nonlinear SVM.

Linear SVM

If in an n-dimensional Euclidean space, $\mathbf{D}_1$ and $\mathbf{D}_2$ are two point sets, $\mathbf{D}_1$ and $\mathbf{D}_2$ are linearly separable if there exist n-dimensional vectors $\mathbf{w}$ and real numbers $\mathbf{b}$ such that all points $\mathbf{x}_i$ in $\mathbf{D}_1$ have $\mathbf{w}^\top \mathbf{x}_i + \mathbf{b} > 0$ and all points $\mathbf{x}_j$ in $\mathbf{D}_2$ have $\mathbf{w}^\top \mathbf{x}_j + \mathbf{b} < 0$. Based on this definition, one can derive 2 concepts: hard-margin and soft-margin.

Hard-margin: If the training data is linearly separable, we can choose two parallel hyperplanes that maximally separate the two data classes. The region circumscribed by these two hyperplanes is referred to as the "margin", and the maximum-margin hyperplane is the hyperplane that lies midway between them. With a normalized or standardized dataset, these hyperplanes can be described by the equations

$$\begin{aligned} \mathbf{w}^\top \mathbf{x}_i + \mathbf{b} &= 1, \\ \mathbf{w}^\top \mathbf{x}_j + \mathbf{b} &= -1. \end{aligned} \tag{1.1}$$

Soft-margin: The hinge loss [8] function is useful for extending SVM to situations where data are not linearly separable

$$\begin{aligned} &\max\left(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i - \mathbf{b})\right), \quad y_i = 1, \\ &\max\left(0, 1 - y_j(\mathbf{w}^\top \mathbf{x}_j - \mathbf{b})\right), \quad y_j = -1. \end{aligned} \tag{1.2}$$

The objective of optimization is therefore to minimize

$$\begin{aligned} &\lambda\|\mathbf{w}\|^2 + \left[\frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i - \mathbf{b})\right)\right], \quad y_i = 1, \\ &\lambda\|\mathbf{w}\|^2 + \left[\frac{1}{n}\sum_{j=1}^{n} \max\left(0, 1 - y_j(\mathbf{w}^\top \mathbf{x}_j - \mathbf{b})\right)\right], \quad y_j = -1, \end{aligned} \tag{1.3}$$

where the parameter λ determines the balance between increasing the margin size and ensuring $\mathbf{x}_i$ and $\mathbf{x}_j$ lie on the appropriate side of the margin.

Nonlinear SVM

However, a method for creating nonlinear SVM classifiers was proposed by applying the kernel trick to maximum-margin hyperplanes [9]. Figure 1.2 shows an example of a nonlinear SVM kernel classifier.

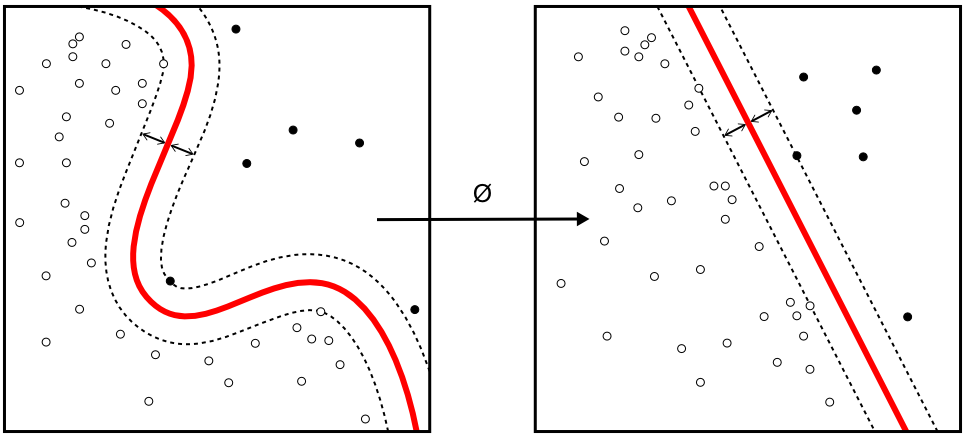


Figure 1.2 – SVM kernel machine. Souce from online [2]. The kernel function ∅ in the figure, computes a low-dimensional nonlinear separable function (left) into a high-dimensional linear separable function (right)

1.1.2 K-means

K-means clustering is an unsupervised method of vector quantization, originally from signal processing, that seeks to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), which serves as the cluster's prototype. It begins by randomly selecting K points from the data as initial centroids, and then assigns each data point to the closest centroid according to the Euclidean distance. After the initial allocation, the centroids are recalculated using the mean of the data points assigned to each cluster. This process of assigning points to the nearest centroid and recalculating the centroids is repeated until convergence is achieved. The end result is K clusters, each with a centroid corresponding to its mean value. Figure 1.3 shows a diagram of k-means clustering for different numbers of K.

2. `https://commons.wikimedia.org/wiki/File:Kernel_Machine.png`

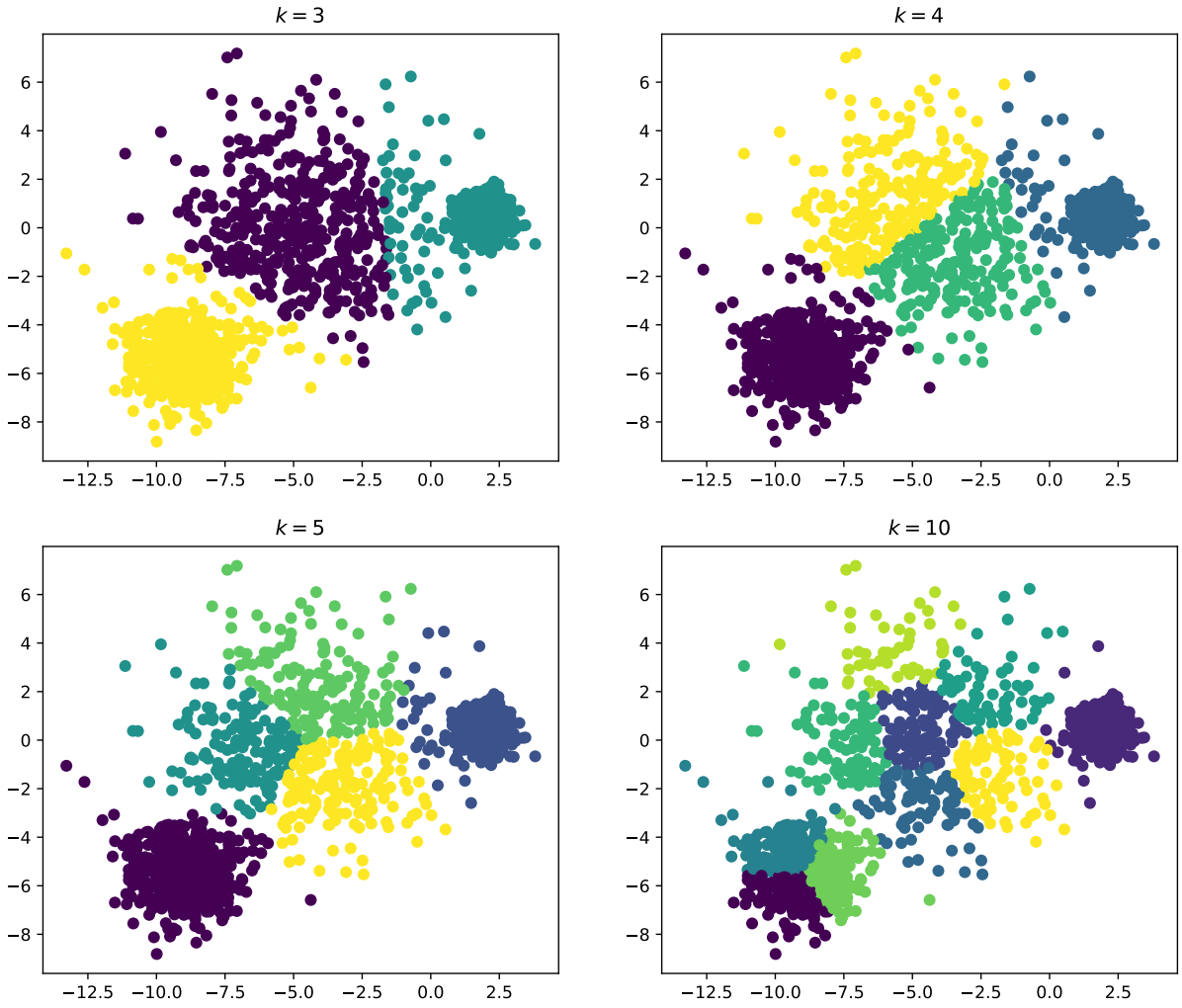


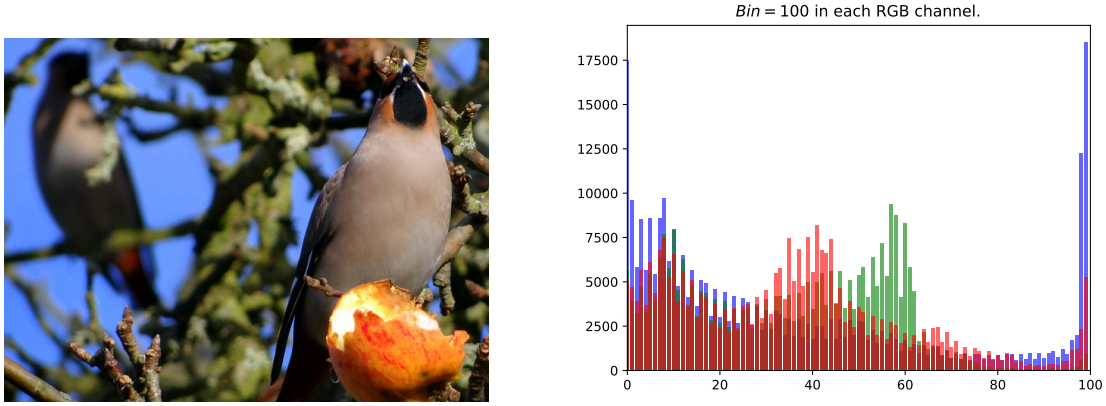
Figure 1.3 – K-means with $k = 3$, $k = 4$, $k = 5$ and $k = 10$.

1.1.3 Visual Features

Both supervised learning SVM and unsupervised learning k-means constitute the fundamental machine learning algorithms. For the vision tasks, these algorithms typically combine various visual features or visual feature descriptors. In this section we list some common visual features (*e.g.*, color, texture, shape feature) for handcrafted machine learning.

Color feature

Human vision is very sensitive to the perception of color. For example, in figure 1.4(a), we can quickly recognize "red fruit", "birds with gray and yellow feathers", "green leaves", "blue sky". However, the data format of an image in a computer is often represented only as a numerical value. Counting the number of pixels with the same color value within a color space can reveal information about the objects contained in an image. Figure 1.4(b) shows the RGB histograms for figure 1.4(a). We can see that a very large number of pixels



(a) Oiginal Image (MSCOCO) (b) Color Histogram

Figure 1.4 – Visual feature of color histogram in RGB space.

have very high blue values, which most likely indicates that the image contains elements of "blue sky". As the example demonstrates, color histogram features only reflect data information and do not reflect object shape or texture.

Color features, such as color sets [10], color moments [11], color coherence vector [12], color correlogram [13], and other color feautres., attempt to mitigate for the shortcomings of the color histogram feature. RGB is the most useful color space for digital image storage

formats. In addition to RGB, there are color spaces such as HSV and YCbCr that can be used to extract color features, as well as a gray channel for gray images.

Texture feature

Complex and variable textures are also crucial identifiers of distinct objects for humans. For instance, the thoughts "tiger" and "leopard" will generate the respective patterned images in the brain. Texture features, in contrast to color features, tend to concentrate more on the image's local information and the object's detailed information. Typically, texture features describe digital image with visual feature descriptors. In other words, these descriptors convert the pixel information of a digital image into a vector or descriptor that can be used to identify and match identical features across multiple images. They exhibit invariance and repeatability across multiple images, making them useful for a variety of computer vision applications, including object recognition, tracking, and 3D reconstruction, among others.

In contrast to color features, which can be pixel-wise calculated directly, texture features must respond in some manner to neighboring pixels. Image convolution operations are a very effective way to reflect the relationship between the central pixel and the surrounding pixels. Assuming an image $\ell(x, y)$ with resolution of $x \times y$ pixels and a convolution kernel $\hbar(u, v)$ with resolution of $u \times v$, we can define the convolution equation as

$$\ell * \hbar = \sum_{(x-u,y-v)\in\ell,(u,v)\in\hbar} \ell(x-u, y-v) \cdot \hbar(u, v). \tag{1.4}$$

Figure 1.5 depicts the fundamental convolution operation, where the kernel size is 3×3. In actual image convolution, the convolution kernel size can be adjusted based on the situation, and each kernel weight can be assigned to a distinct value as necessary. The input image consists of two-dimensional pixels, so the convolution kernel employs a sliding window mechanism to compute in the next receptive field. In addition, convolution kernel is also known as filter or operator in various contexts.

SIFT (Scale-Invariant Feature Transform [14]), SURF (Speeded-Up Robust Features [15]), HOG (Histogram of Oriented Gradients [16]), ORB (Oriented FAST and Rotated BRIEF [17]), and others are common computer visual feature descriptors. These descriptors have various properties, such as scale-invariance, rotation-invariance, and computational efficiency, and are suited for a variety of application scenarios. Figure 1.6 shows the texture features extracted by different descriptors in the same scene. All descriptors

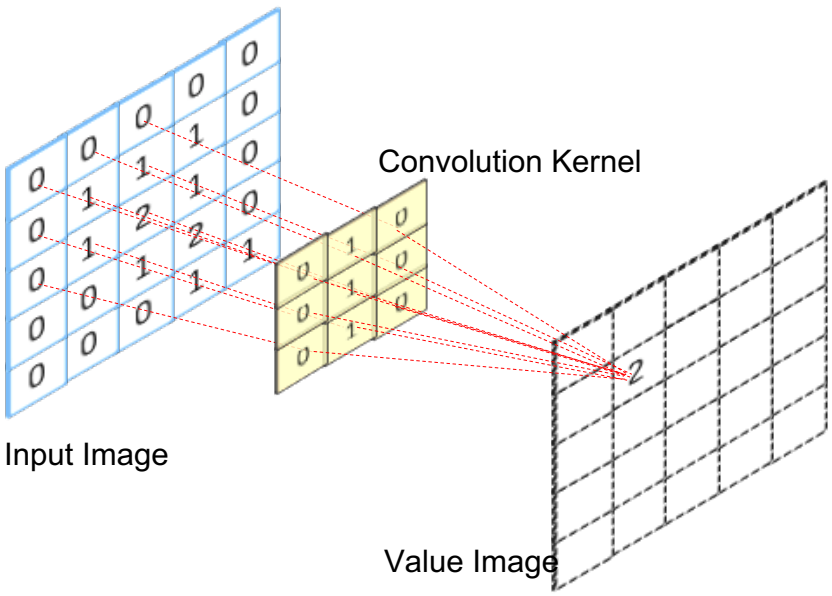


Figure 1.5 – Image convolution illustration.

detect more feature points for image regions with complex and highly variable texture. These regions frequently contain more information and the distinguishing characteristics that set them apart.

There are also texture features in the frequency domain, such as Fourier transform or wavelet transformed frequency features [18], which are particularly valuable for medical images. Texture feature have a global property, *i.e.*, global image detection, but place a greater emphasis on regions with significant texture variations.

Shape feature

As its name implies, it is a shape-based visual feature that can be used to detect objects of interest in images more effectively. As with the textural features mentioned above, human consciousness readily associates objects with their corresponding visual field projection shapes, such as “circle” when the word “basketball” is noted or “parallelogram” when the word “bus” is noted. Numerous algorithms begin with this aspect and search digital data for information pertaining to the geometry of the object. Figure 1.7 illustrates many kinds of shape features based on input grayscale image 1.4(a).

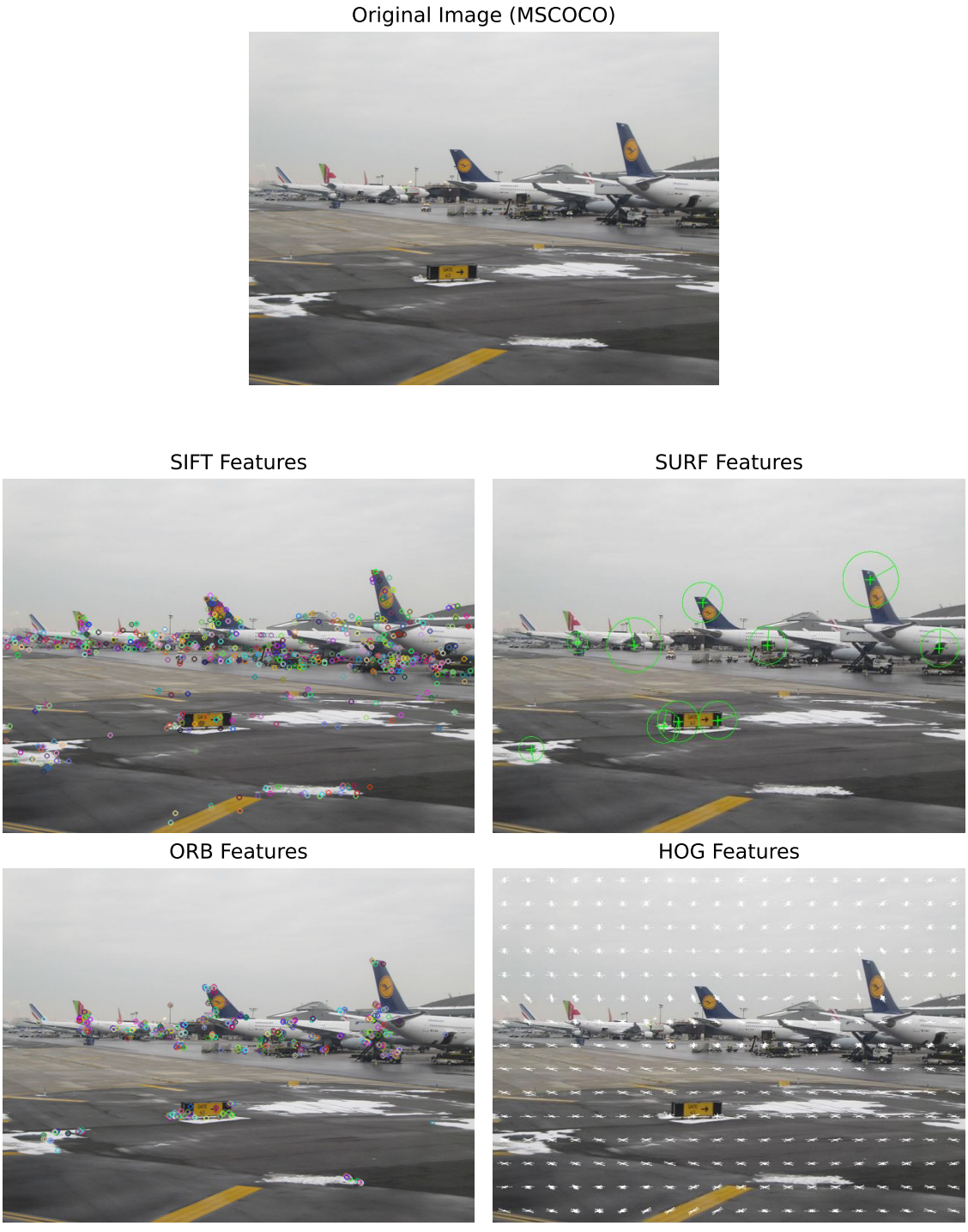


Figure 1.6 – Examples of SIFT, SURF, ORB, and HOG descriptors.

Edge detection: Edges are areas in an image where the pixel intensity abruptly changes, typically denoting the boundaries of objects. Calculating the gradient of each pixel within an image, edge detectors identify edges. Pixels with significant gradient values are typically located along boundaries. *E.g.*, Canny detector [19] and Sobel detector [20].

Skeletonization: While preserving their topology, skeletonization algorithms are used to reduce binary objects to their skeletal structure. This is accomplished by removing pixels from the object's boundary until only its skeleton remains. *E.g.*, Zhange-Suen Thinning [21] and GuoHall Thinning [22].

Contour extraction: Using contour extraction algorithms, the contours of objects in an image can be extracted. Contour extraction aims to extract the outer boundaries of objects or shapes in an image, typically representing the shape of the object. *E.g.* Suzuki's algorithm [23].

1.1.4 Shallow Autoencoder in CV

In most vision algorithms, visual features in 1.1.3 do not tend to appear alone, but execute the task together with algorithms such as SVMs mentioned in 1.1.1 or k-means mentioned in 1.1.2. Suppose we have a set of n-dimensional visual feature $\mathbf{x}$ from input images $\{I\}$, and use the SVM formula $f(\cdot)$ same as the previous without using any kernel trick for classification task (*i.e.*, when the margin is infinitely tiny, the linear SVM can be thought of as a linear classification function [24]), which can be defined as

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + \mathbf{b}, \tag{1.5}$$

where $\mathbf{w}$ represents n-dimensional of weights, $\mathbf{b}$ represents bias. Figure 1.8(a) depicts a binary classifier that can distinguish between "red fruit" and "green leaves" for a single attribute. However, if we also wish to classify the fruit and the leaf based on their texture, we can add a second binary classifier to manage this feature classification. We can express this in two of Equation 1.5, by

$$f_1(\mathbf{x}_i) = \mathbf{w}_1^\top \mathbf{x}_i + \mathbf{b}_1, \tag{1.6}$$

$$f_2(\mathbf{x}_j) = \mathbf{w}_2^\top \mathbf{x}_j + \mathbf{b}_2, \tag{1.7}$$

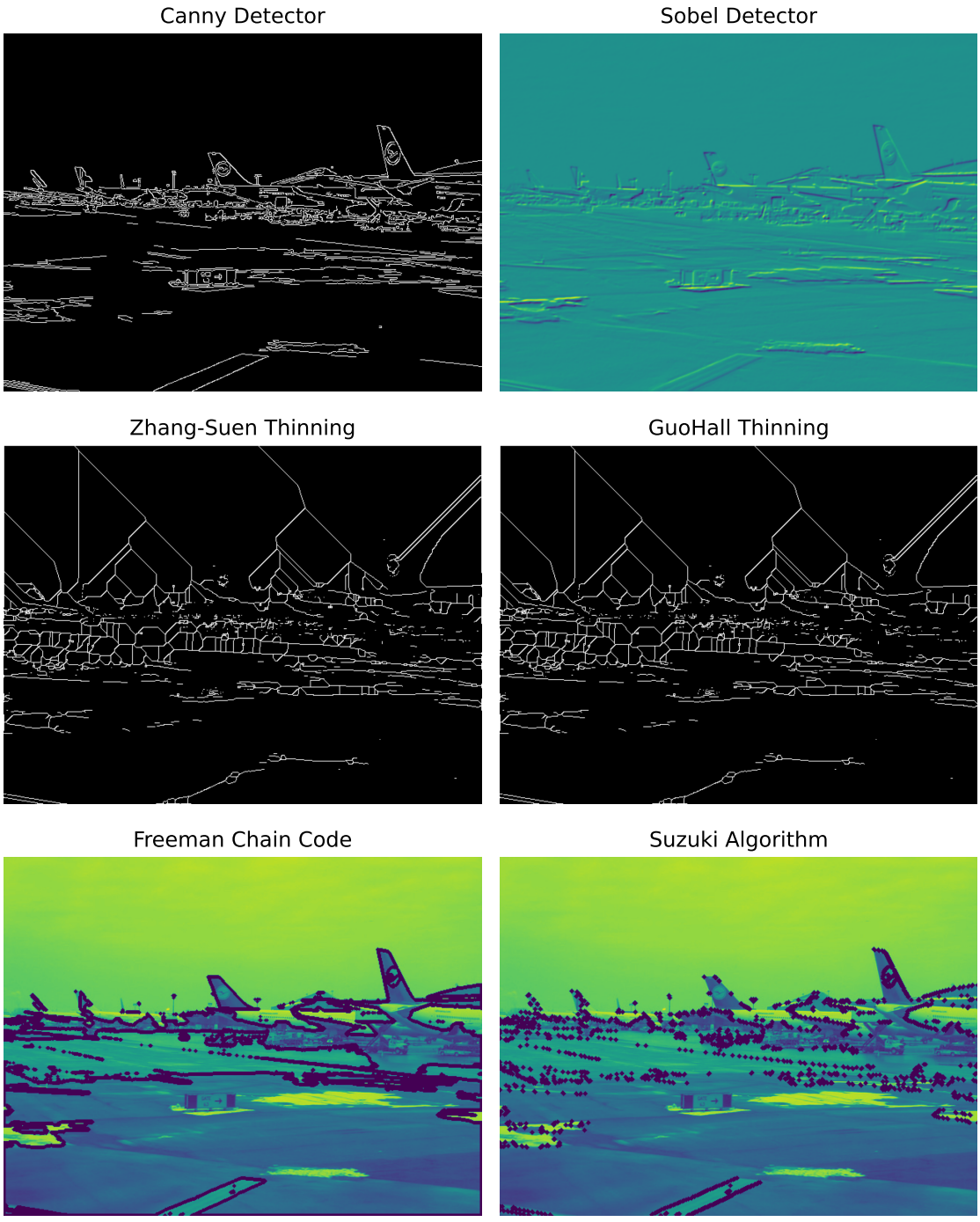


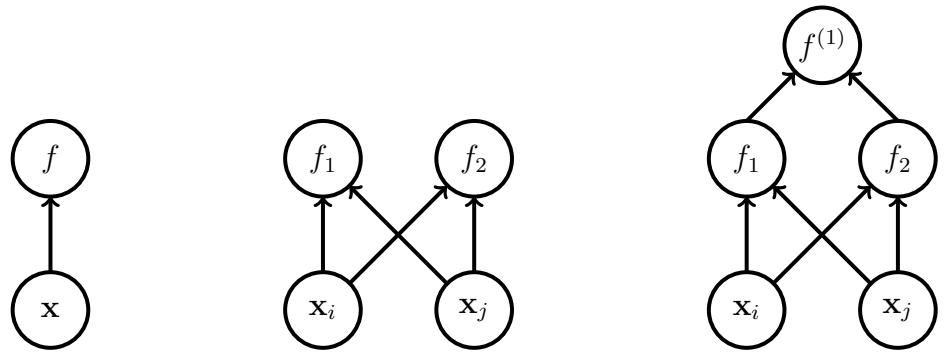
Figure 1.7 – Examples of shape features algorithms.

where $\mathbf{x}_i, \mathbf{x}_j$ are features extracted from input images $\{I\}$. Thus, "fruit" and "leaves" can be distinguished by two dimensions (*i.e.*, the results of f_1 and f_2). Obviously, we can also use a third binary classifier $f^{(1)}$ connected after the first two to classify the outputs of f_1 and f_2 further, which are described by

$$\begin{aligned} f^{(1)}(f_1(\mathbf{x}_i)) =& {\mathbf{w}^{(1)}}^\top f_1(\mathbf{x}_i) + \mathbf{b}^{(1)} \\ =& {\mathbf{w}^{(1)}}^\top (\mathbf{w}_1^\top \mathbf{x}_i + \mathbf{b}_1) + \mathbf{b}^{(1)}, \end{aligned} \tag{1.8}$$

$$\begin{aligned} f^{(1)}(f_2(\mathbf{x}_j)) =& {\mathbf{w}^{(1)}}^\top f_2(\mathbf{x}_j) + \mathbf{b}^{(1)}, \\ =& {\mathbf{w}^{(1)}}^\top (\mathbf{w}_2^\top \mathbf{x}_j + \mathbf{b}_2) + \mathbf{b}^{(1)}. \end{aligned} \tag{1.9}$$

Figures 1.8 depict the equations aforementioned using diagrams that clearly illustrate the structural differences resulting from various combinations of linear classifiers. These fundamental structures often serve as building blocks for more complex structures.



(a) Eq.(1.5) (b) Eq.(1.6) and (1.7) (c) Eq.(1.8) and (1.9)

Figure 1.8 – (a)(b)(c) represent single classifier, two classifiers and two-level classifier.

In addition to supervised learning algorithms like SVM, shallow autoencoding identifiers can also utilize unsupervised learning techniques. For example, in a scenario where n-dimensional visual features $\mathbf{x}_i$ are clustered into k categories using a technique like k-means (mentioned in 1.1.2), the n-dimensional features extracted from test or query images can be compared with each of the clustered categories to determine the minimum distribution distance and subsequently assign categories.

Since a shallow autoencoder can extract visual features from multiple aspects and attributes, it significantly improves prediction accuracy compared to single-feature meth-

ods. However, the shallow autoencoder approach requires empirical selection of suitable features, the number of classifiers or clustering centers, and other parameters. Additionally, while visual features are a well-established success, their consistency across different datasets can vary. To achieve comparable performance on other datasets, several parameters or structures may require modification. Despite its lack of generality, the shallow autoencoder approach provides a stable and relatively dependable solution for vision tasks, particularly in settings with limited computational resources.

1.2 Deep Learning in CV

Before delving into the concept of deep learning, it's important to review the history of artificial neural networks (ANN). The origins of this mathematical model date back to the 1960s and were inspired by the transmission of signals in the human nervous system. In this system, visual signals are transmitted through a chain of neurons that activate successive nerves until they reach the brain. This can be thought of as a nested function, where the output of the previous function serves as the input to the next one. In a neural network, the first layer has n neurons, and the second layer has m neurons. As a result, the computational workload of these two layers scales at least as $O(n \times m)$, *i.e.*, feed-forward propagation only considered. Consequently, increasing either the number of neurons in each layer or the number of layers results in an exponential increase in the computational complexity. This strategy was not viable in decades where computing resources are limited. Because of this, shallow neural networks have received less attention than handcrafted features algorithms such as SVM and k-means.

1.2.1 Fully Connected Neural Network

As computer hardware performance improves, artificial neural networks are once again attracting the attention of vision algorithm researchers. Multi-layer perceptrons (MLPs) are built by combining basic components (*i.e.*, the structure of figure 1.8(c)), and can be further enhanced by incorporating additional activation functions to create more flexible and versatile configurations. In MLP, the first layer is called the input layer, which recevies input data; the last layer is called the output layer; the layers between input layer and output layer are middle layers, which are also called hidden layers. The neurons of hidden layers are connected to one another, which means that the nodes in the graph structure

are all connected. Therefore, each layer is referred to as a fully connected layer (FC-layer). Figure 1.9 showns this structure of FCNN, which has 16 neurons in input layer, 12 and 10 neurons in the first and second hidden layers respectively, and one neuron as output layer.

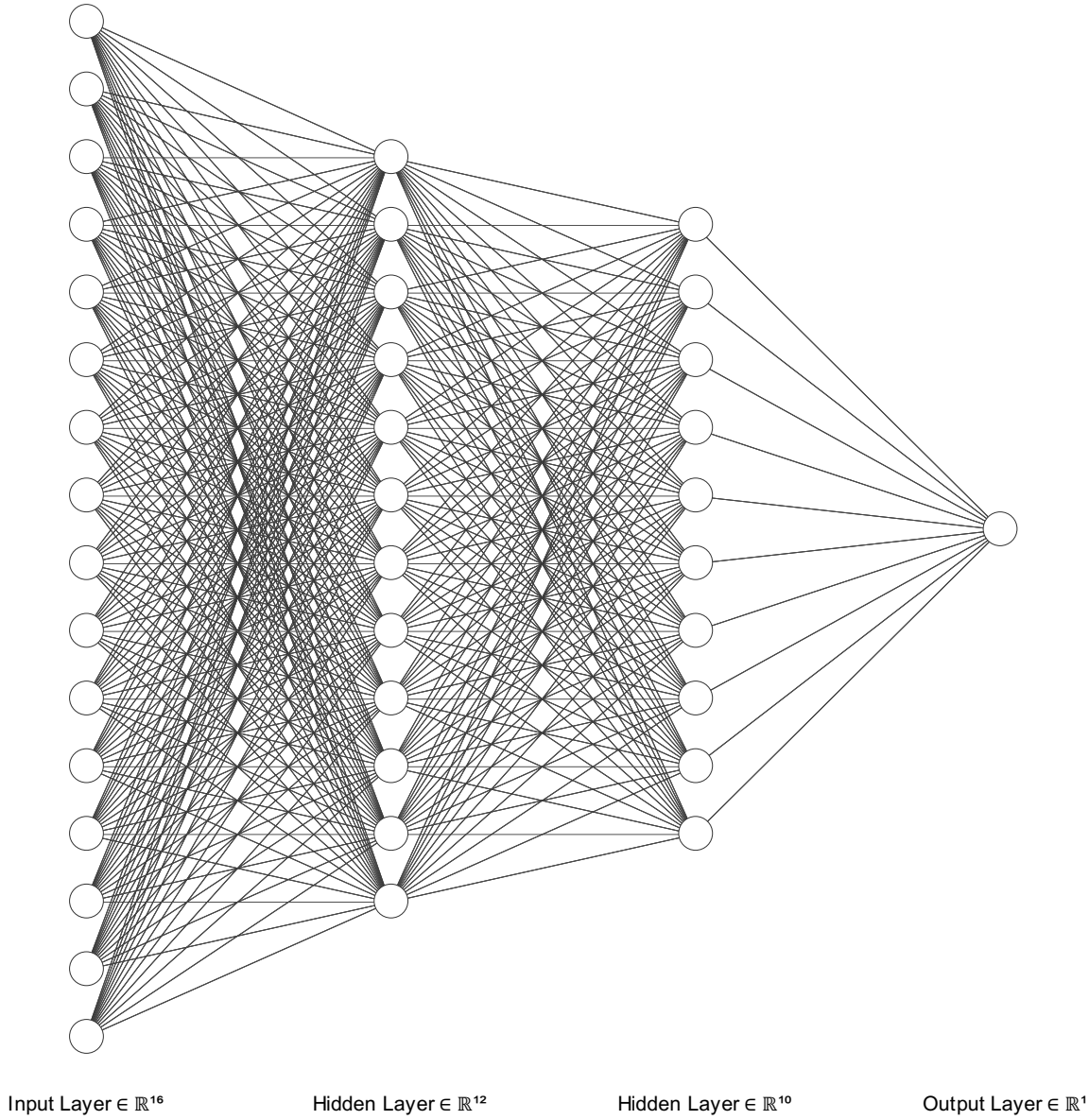


Figure 1.9 – Example of fully connected neural network. (Figure is made online [3]).

3. `http://alexlenail.me/NN-SVG/`, visited 22/04/2023

Activation function

In previous sections, we used linear functions as classifiers to quickly understand the concept of neural networks. However, in reality, data is sometimes nonlinear distributional, which requires the use of nonlinear functions to map the input data to the output values [25]. These nonlinear functions are called activation functions, which determine whether or not the output data should be passed to the next layer of neurons and how much of it will be passed. Activation functions are crucial for learning complex patterns and relationships in the data, which linear functions cannot capture. The most commonly used activation functions are `sigmoid`, `ReLU` (rectified linear unit), and `tanh`. `sigmoid` can be described by

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{1.10}$$

`ReLU` can be depicted by

$$\texttt{ReLU}(x) = \max(0, x). \tag{1.11}$$

`tanh` activation funciton is

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \tag{1.12}$$

Figure 1.10 shows these three kinds of activation functions. The sigmoid activation func-

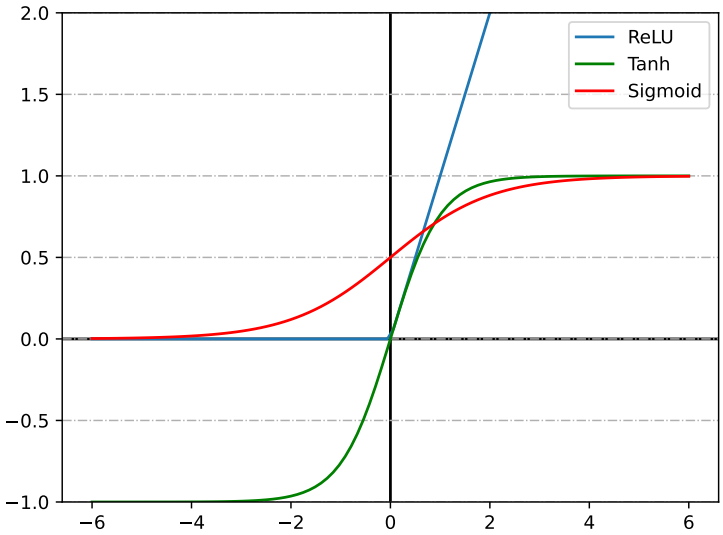


Figure 1.10 – The Sigmoid (1.10), `ReLU` (1.11) and tanh (1.12) activation function.

tion is a popular choice because of its easy-to-calculate derivative, which is simply the function itself multiplied by $1 -$ the `function`, (*i.e.*, $\sigma'(x) = \sigma(x) \times (1 - \sigma(x))$). The sigmoid function can map any input value to a value between 0 and 1, making it useful for problems that require probabilistic outputs. However, one limitation of the sigmoid function is that it becomes saturated for very large or small input values, leading to issues with gradient explosion or vanishing during backpropagation after multiple iterations. These problems can make it challenging for the neural network to effectively learn from the data. In contrast, the ReLU (Rectified Linear Unit) activation function overcomes this limitation by being a simple linear function that returns the input value if it is positive, and 0 for otherwise. This results in faster and more efficient learning, especially for deep neural networks.

Back forward propagation

The back forward propagation algorithm is a powerful method for training artificial neural networks [26]. It enables a deep neural network to update the weights of each node in the network based on the error between the predicted output and the true, which enhance the network to make more accurate predictions. Back forward propagation involves two key mathematical tools: the chain rule and gradient descent. The chain rule allows the error to be propagated backwards through the network, updating the weights of each node based on their contribution to the error. Gradient descent is a method to optimize the weights by iteratively adjusting them in the direction of steepest descent of the loss function. Together, these tools enable the network to find the weights that minimize the difference between the outputs and the true, resulting in a trained network that can make accurate predictions on new data.

We now briefly illustrate the principle of back forward propagation with the example in Figure 1.8(c). Assuming that the loss function $L(x, y)$ is the object to gradient descent algorithm, here represented by:

$$L(x, y) = \sum_{n=1}^{2} (f^{(1)} - y_n)^2. \tag{1.13}$$

The $f^{(1)}$ linear function is also replaced with a sigmoid activation function, to give

$$f^{(1)} = \frac{1}{1 + e^{-f_n(\mathbf{x})}}, \quad n = \{1, 2\}. \tag{1.14}$$

Merge Equation 1.6 and 1.7 to become

$$f_n(\mathbf{x}) = \mathbf{w}_n\mathbf{x} + \mathbf{b}_n, \quad n = \{1, 2\}. \tag{1.15}$$

First, we observe that L is only dependent on the weight $\mathbf{w}_n$ via the summed input f_n to activation unit $f^{(1)}$. Therefore, we can apply the chain rule to partial derivatives to arrive at

$$\frac{\partial L}{\partial \mathbf{x}_1} = \frac{\partial L}{\partial f^{(1)}} \frac{\partial f^{(1)}}{\partial \mathbf{x}_1}. \tag{1.16}$$

Setting the partial derivatives of Equation 1.13 and 1.14, we obtain:

$$\frac{\partial L}{\partial f^{(1)}} = 2 \sum_{n=1}^{2} (f^{(1)} - y_n), \tag{1.17}$$

and

$$\frac{\partial f^{(1)}}{\partial \mathbf{x}_1} = \frac{\partial f^{(1)}}{\partial f_n} \frac{\partial f_n}{\partial \mathbf{x}_1} = f^{(1)}(1 - f^{(1)})\mathbf{w}_1. \tag{1.18}$$

Bringing Equation 1.17 and 1.18 to 1.16, then

$$\frac{\partial L}{\partial \mathbf{x}_1} = 2 \sum_{n=1}^{2} (f^{(1)} - y_n) f^{(1)}(1 - f^{(1)})\mathbf{w}_1. \tag{1.19}$$

The final Equation 1.19 tells us that once the output of the feed-forward propagation is determined, we can know the partial derivatives that need to be updated for each weight parameter by the chain rule. Obviously, in each iteration of back-propagation, $\mathbf{w}_1$ does not update the entire partial $\frac{\partial L}{\partial \mathbf{w}_1}$, but rather tends to multiply by a parameter η to control the descent speed, which can be presented by

$$\mathbf{w}_1^{\tau+1} = \mathbf{w}_1^{\tau} - \eta \frac{\partial L}{\partial \mathbf{w}_1^{\tau}}, \tag{1.20}$$

where τ indicates an iteration, parameter η is known as learning rate.

1.2.2 Convolutional Neural Network

The convolutional neural network (CNN) combines a fully connected neural network (FCNN) and image convolution (*i.e.*, in Figure 1.5). Through back-propagation, the convolution kernel can automatically learn the kernel values instead of manually setting the parameter values earlier via experience, as is the case with shallow machine learning

methods (1.1). The classical networks of CNN are LeNet, AlexNet, ResNet, DenseNet.

LeNet

LeNet is a convolutional neural network proposed by LeCun et al. in 1989 [27] for the purpose of recognizing handwritten digits in the U.S. mail system. The input is a grayscale image, and the output is classified into ten categories ranging from "0" to "9". In addition to the convolutional and fully connected layers, LeNet includes a pooling layer, a technique for subsampling in a patch square that drastically reduces the number of network parameters. In general, there are two modes of pooling layers: average pooling, *i.e.*, taking the average value within a patch, and max pooling, *i.e.*, taking the maximum value within a patch. Figure 1.11 shows the structure of an improved version of LeNet, also known as LeNet-5 [28]. In LeNet-5, the input is a 32×32 pixels image, followed by two convolutional layers with 5×5 convolution kernel size, two max pooling layers with 2×2 patch size, two FC layers as hidden layers with 120 units and 84 units, and one output layer with 10 units. LeNet demonstrates the practical utility of image convolution in conjunction with multilayer perceptrons for image classification tasks.

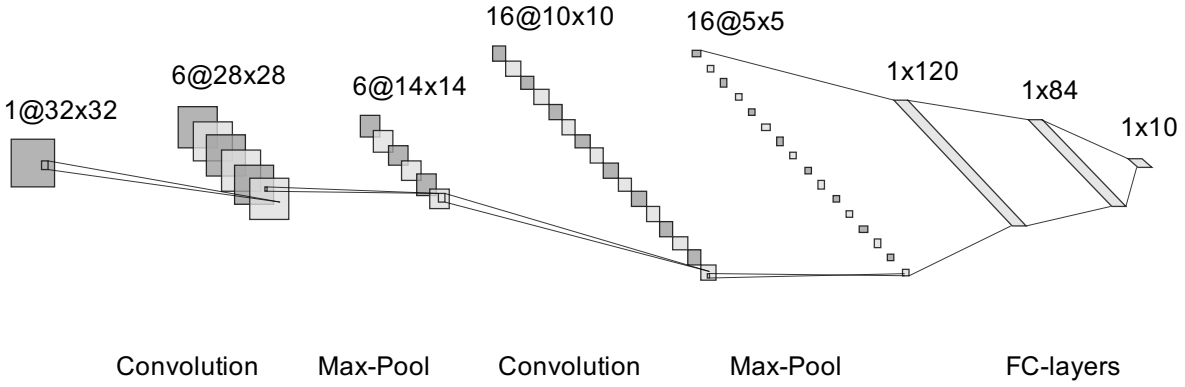


Figure 1.11 – The structure of LeNet-5. (Figure is made online. [4])

AlexNet

Alex et al. [29] proposed AlexNet at the ImageNet Large Scale Visual Recognition Challenge [30], where it significantly enhanced the performance of the image classification

4. `http://alexlenail.me/NN-SVG/`, visited 22/04/2023

task and notably separated itself from the other competing groups. AlexNet increases the number of layers of convolutional kernels, which allows one convolutional kernel to process multiple image channels simultaneously. In addition, AlexNet deepens the depth of convolutional layers, which can be called a deep convolutional network. To solve the gradient vanishing problem of multiple iterations of the sigmoid activation function, AlexNet uses ReLU (1.11) as an alternative. AlexNet also uses dropout trick to control the complexity of the model in the fully connected layer to avoid overfitting. Image augmentation, such as flipping, cropping and color change, is introduced in AlexNet to further expand the dataset to mitigate overfitting. Alexnet's emergence can be viewed as a tuning point in the computer vision domain, which proposes many of the extremely practicable deep convolutional innovation points. Figure 1.12 shows the structure of AlexNet. In this figure, there are 5 convolutional layers and 3 fully connected layers, but 3 max pooling layers are not marked, where blue and red indicate the convolutional kernels and the concatenation with the next layer, respectively. Note that the size of the first layer convolution kernel is 11×11, the size of the second convolution kernel is 5×5, and the size of the last three layers convolution kernels is 3×3.

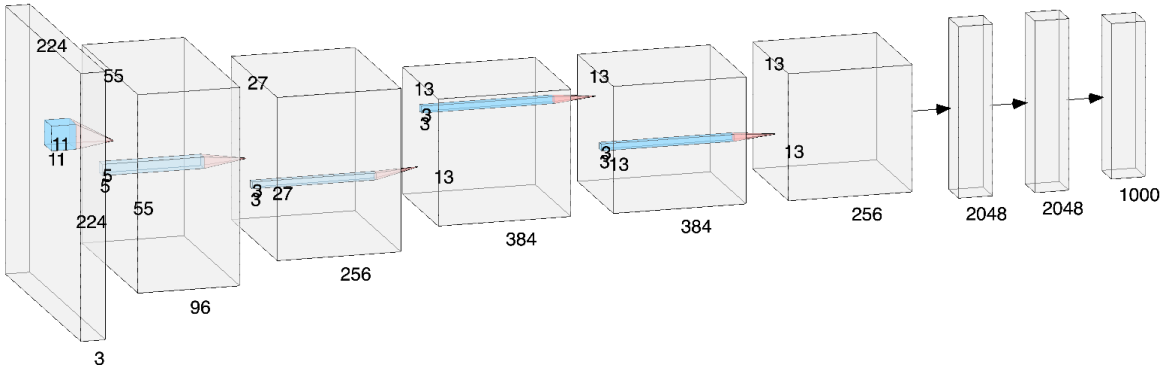


Figure 1.12 – The structure of AlexNet. (Figure is made online. [5])

VGG-Net

In 2014, Oxford University proposed VGG-Net, a deep convolutional neural network with smaller convolution kernels and more layers than AlexNet [31]. While AlexNet's first and second convolutional layers use 11×11 and 5×5 convolution kernels to obtain a larger

5. `http://alexlenail.me/NN-SVG/`, visited 22/04/2023

receptive field on the image. VGG-Net uses smaller convolution kernels with a deeper network to increase parameter efficiency. They found that two convolution kernels of size 3×3 can replace one convolution kernel of size 5×5 under the condition that the kernel values can be updated. The two small-size kernels have parameters of $2 \times (3 \times 3) \times C = 18C$, ($C$ indicats the number of channels) and one large-size parameter of $(5 \times 5) \times C = 25C$. And their receptive fields are the same size, as shown in Figure 1.13. Similarly, three 3×3 kernels can be used instead of 7×7, in which case their parameters are $3 \times (3 \times 3) \times C = 27C$ and $(7 \times 7) \times C = 49C$, respectively. Thus, this greatly reduces the deep convolutional network parameters and computational load. The VGG-Net structure configurations is shown in Table 1.1. Different columns show different depths of the network structure. VGG-Net demonstrates superior performance and is frequently used for tasks such as image feature extraction, object detection, and segmentation masks generation.

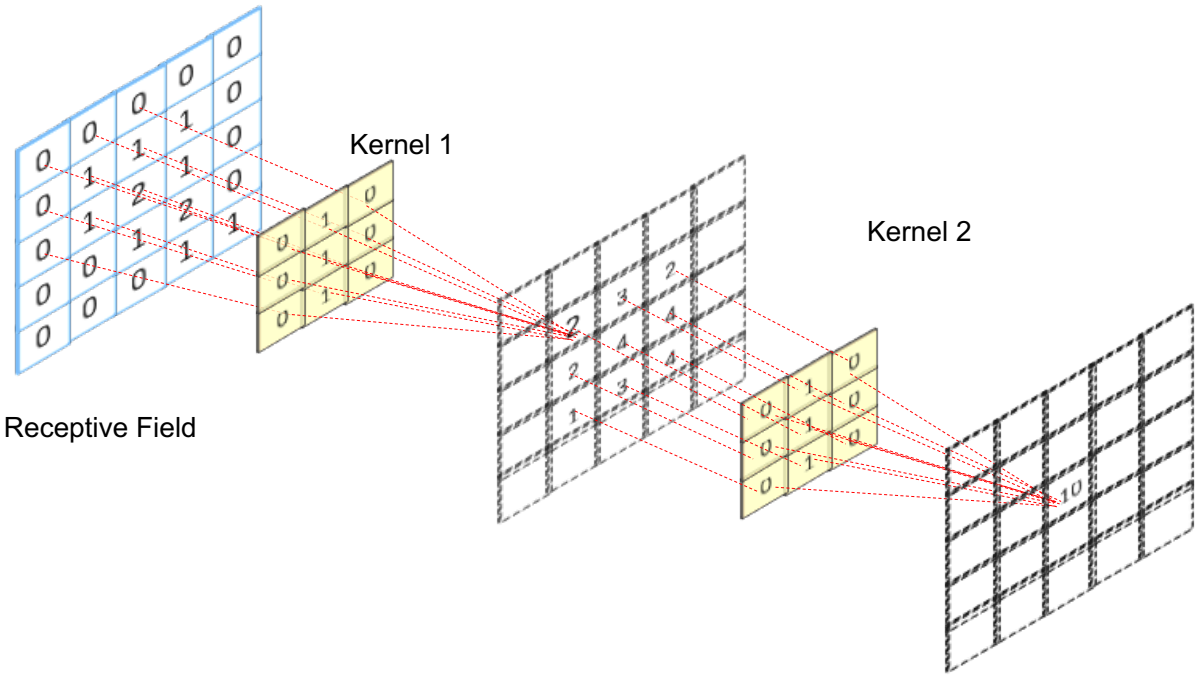


Figure 1.13 – Two 3×3 kernels' receptive field equal to one 5×5 kernel.

ResNet

Kaiming et al. introduced ResNets (residual networks) in 2015 [32]. Their ILSVRC2015 score of 3.6% is even lower than the human error rate of 5% in classification task. That is

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 1.1 – VGG network configurations [31].

the first time an AI has outperformed the average human in a large-scale data vision task. The proposed residual network enables the training of deeper networks. We know from

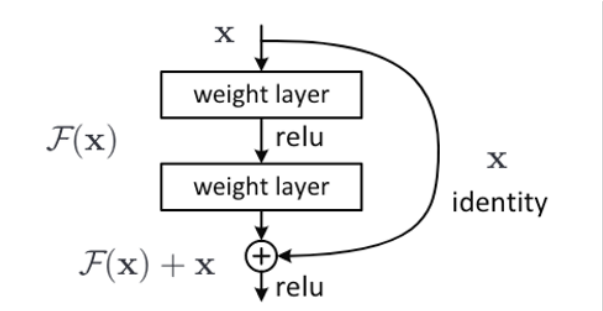


Figure 1.14 – A residual learning block [32].

previous networks that deeper network structures often produce better results. However, as the number of layers in the network increases, degradation can occur, making it more difficult for deeper networks to be trained (to converge). One possible reason is that multiple nonlinear activation functions can obscure identity features. Therefore, ResNet proposed residual learning. In a residual block, the input value of the previous layer is directly passed over two (or possibly more) weight layers and an activation function and added as a bias to the output. A residual block diagram shows in Figure 1.14. Building the block can be presented by

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{\mathbf{W}_i\}) + \mathbf{x}, \tag{1.21}$$

where $\mathbf{y}$ indicates output for under-layer, $\mathbf{x}$ indicates input from up-layer, $\mathbf{W}_i$ indicates the weights in the weight layer. This operation of skip weight layers is called a shortcut. Additionally, ResNet uses batch normalization (BN) layer to control batch data rather than dropout operations. Thus, more identical features could map into deeper place in deep neural network. In their experiments, ResNet expanded the network to more than 1000 layers and obtained positive results. Figure 1.15 gives a visualized network structure comparison of ResNet and VGG-Net.

DenseNet

Inspired by ResNet, DenseNet [33] uses a more extensive range of residual connections, which provide a direct pathway to connect shallow layers. Between each dense block, a

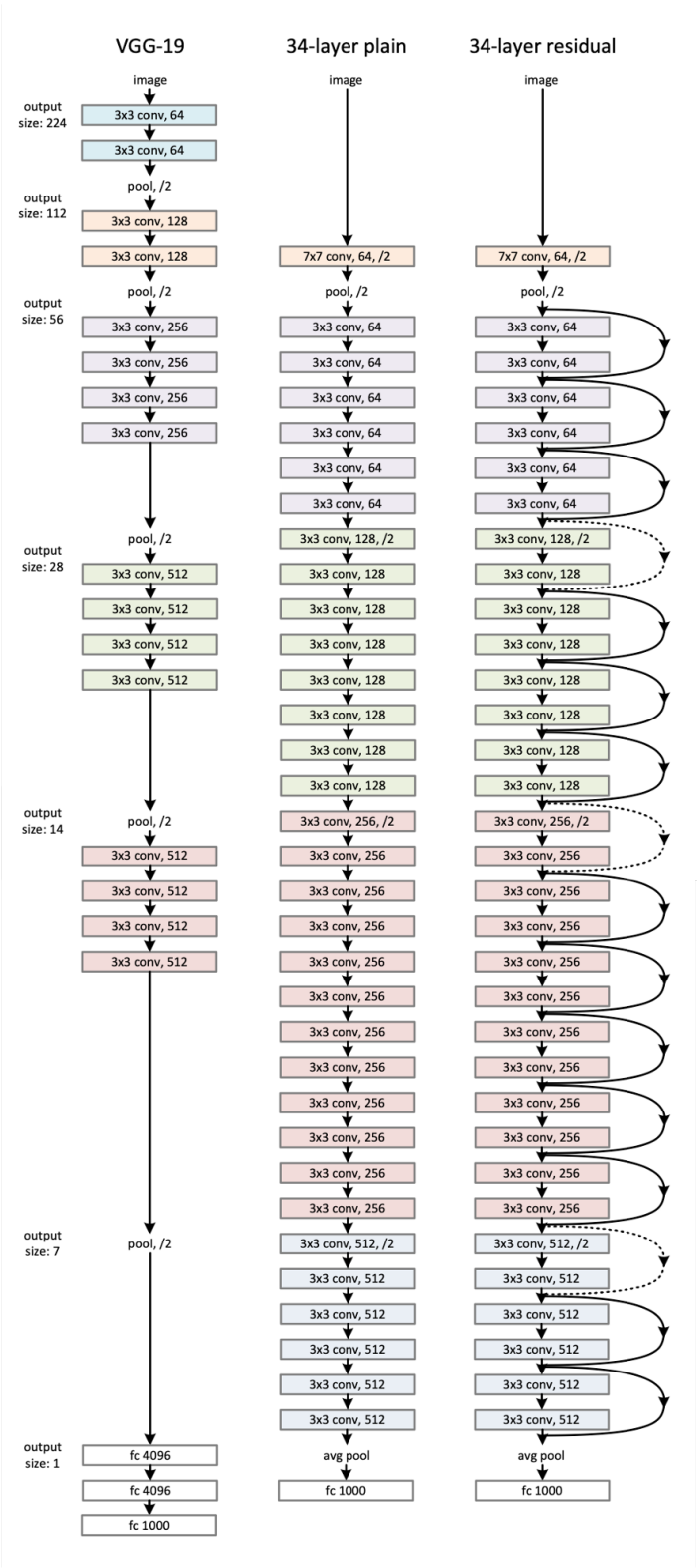


Figure 1.15 – ResNet's structure vs. VGG-Net's structure [32].

transition layer has been use to reduce the size of feature maps, which consists of one 1×1 convolutional and one 2×2 average pool with stride of 2. This results in similar benefits to ResNet with fewer parameters. Although DenseNet was initially proposed for object recognition tasks, this approach of being friendly to shallow layers was quickly adopted by other deep networks with more demanding pixel-level tasks such as semantic segmentation. Figure 1.16 shows the main idea of DeseNet.

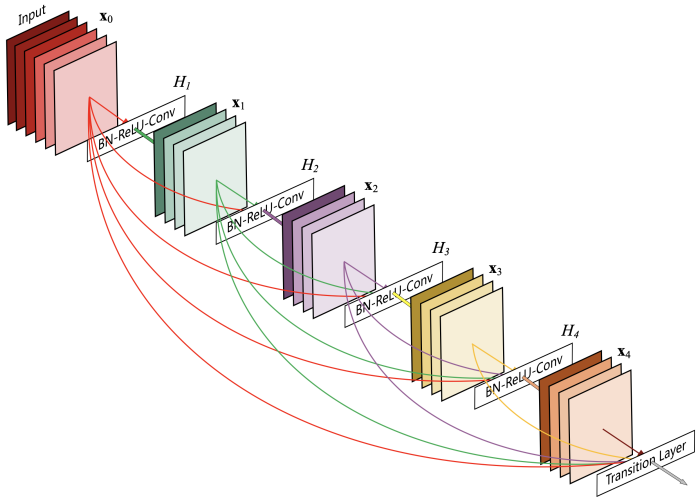


Figure 1.16 – The features connections between each DenseNet's dense blocks [32].

Since the proposals of ResNet and DenseNet, convolutional neural networks (CNNs) have dominated various vision tasks. Some CNNs are used as backbone networks to directly derive visual feature maps output from hidden layers, which serve as a basis for further investigation. Others use CNNs as modules that complement other network components. CNNs appear in both forms in any field involving vision. Despite their success, CNNs have some limitations. First, they have a large number of parameters and require substantial amounts of training data, making both data collection and training computation resource-intensive. Moreover, the feature space of the hidden layers in CNNs can be challenging to interpret, unlike handcrafted features (e.g, in 1.1.3) that are more easily understandable by human. As a result, the transfer learning features from middle layers of CNNs are often only interpretable by CNN-like models.

In contrast, the proposals of Transformers [34], the renowned concept from natural language processing, is beginning to challenge the foundation of convolutional neural networks in the field of vision. Around this two years, two clouds loom over computer vision:

the limitations of convolutional neural networks and the emergence of Transformers.

1.2.3 Transformer-based Vision Model

Transformer is a novel neural network architecture for natural language processing (NLP) applications proposed by Vaswani et al. in 2017 [34]. Transformer effectively addresses the inefficient processing of extended sequences and the difficulty of capturing global information by employing a self-attention mechanism as a subsitute for RNNs and CNNs, and has achieved remarkable results in the field of NLP. In the following section 2.2.2, we will go deeper into the specifics of Transformer. Currently, the Transformer can briefly be considered as "a multi-category classifier"; instead of outputting each category's probability values, this classifier outputs the dependencies between each category. This section highlights the Transformer-based vision model, which could be divided into two categories. One uses Transformer to complement traditional visual models like CNNs. The other is to view the Transformer-based model as a backbone to extract visual features instead of the traditional visual models.

Transformer-based model to complement CNNs

DETR [35] connects a transformer encoder and decoder after a CNN backbone to solve object detection tasks. The extracted visual features from the CNN are fed into a feed-forward network (FFN) to obtain the predicted classes and bounding boxes. Those images features are embedding with positional encoding. In the middle of the transformer encoder and decoder, feature vectors learn the attention between each other component in their selves, *i.e.*, self-attention mechanism. In the decoder phase, features have been calculated the attention relationship with the queries in order to output predictions. The transformer encoder and decoder are illustrated with pipeline of DETR structure in Figure 1.17.

Transformer-based model as backbone for vision task

When transformers have been effectively introduced to CV from NLP, researchers wonder whether a transformer-based model could replace CNN's backbone and learn self-attention directly from the input image. Several research efforts are comparing the visual features of the transformer-based model with the CNN model.

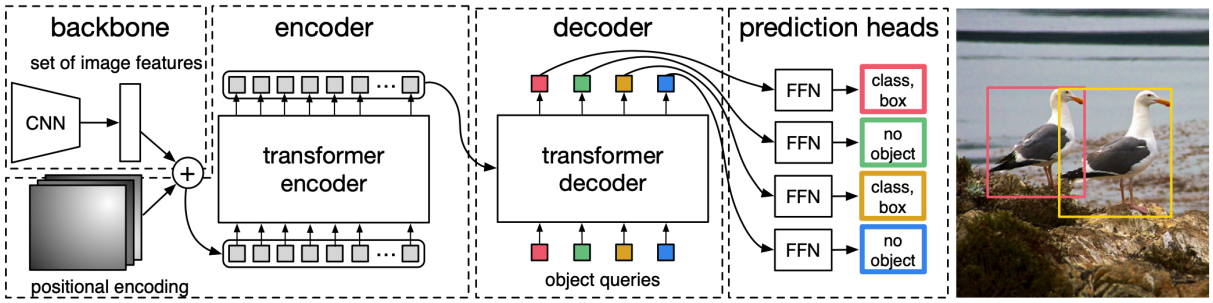


Figure 1.17 – The pipeline of DETR [35].

ViT: Following the original Transformers architecture, the Google Research Brain team aimed to develop simple, universally scalable architectures. In 2021, they proposed the Visual Transformer [36] (ViT), which divides an image into 16×16 patches and linearly embeds them within the transformer encoder as 16-by-16 words. Unlike CNNs that use convolutional kernels to advance the two-dimensional features of an image, ViT flattens separated small image patches directly as if they were one-dimensional information for subsequent operations. Although this diminishes the flat visual information, attentions are enhanced between each small patch. Compared with CNN-like models, transformer-based models focus more on the connections between high-level semantic features. The ViT only uses the Transform encoder module, which is followed by a MLP classifier. The overview of ViT model is shown in Figure 1.18

Swin Transformer: Although ViT performs well and even outperforms some CNNs on classification tasks, it falls short in image tasks that rely more on pixel-level annotation, such as image segmentation. This is presumably due to the lack of storage for two-dimensional planar information. Swin Transformer [37], on the other hand, proposes to address this deficiency by preserving two-dimensional positional information while transforming input data into sequential information using the concept of hierarchy. First, Swin Transformer adopts different size of patch in different hierarchical layer. The more higher layers' patch size is bigger, which allows more region information, see Figure 1.19(a) Then, Swin Transformer uses shifted window to shuffle the order of local encoder, when the features through one layer to next layer, which have been shown in Figure 1.19(b). There are some other detailed changes, but the model architecture of Swin Transformer is basically similar to ViT, as can be seen in the Figure 1.19(c). With these improved refinements, Swin Transformer is able to outperform CNNs backbone model for pixel-level tasks (*i.e.*,

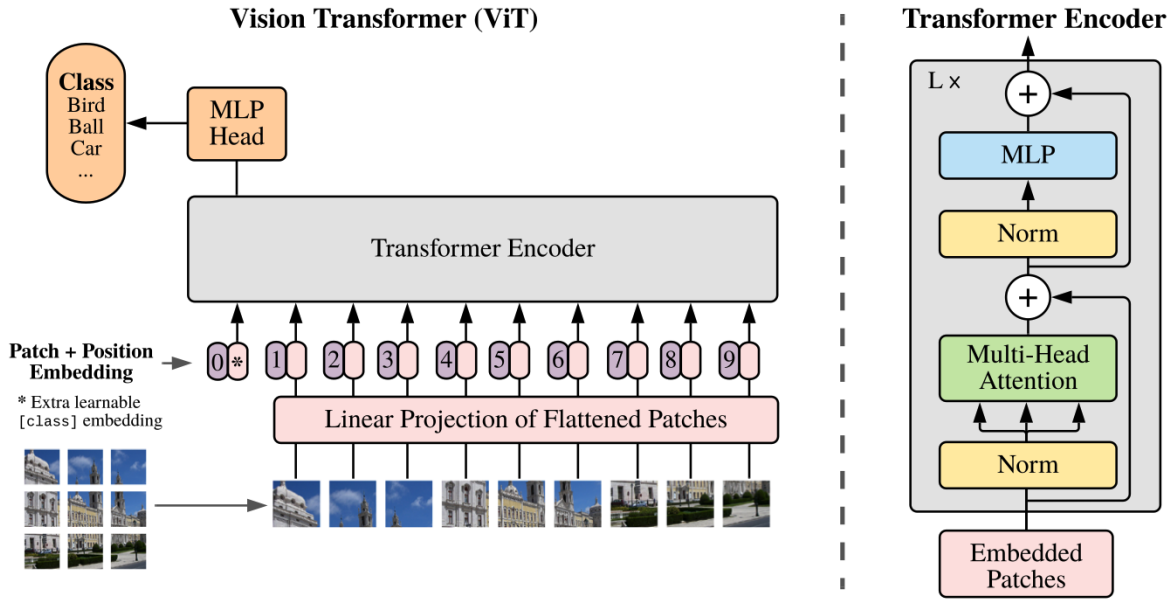
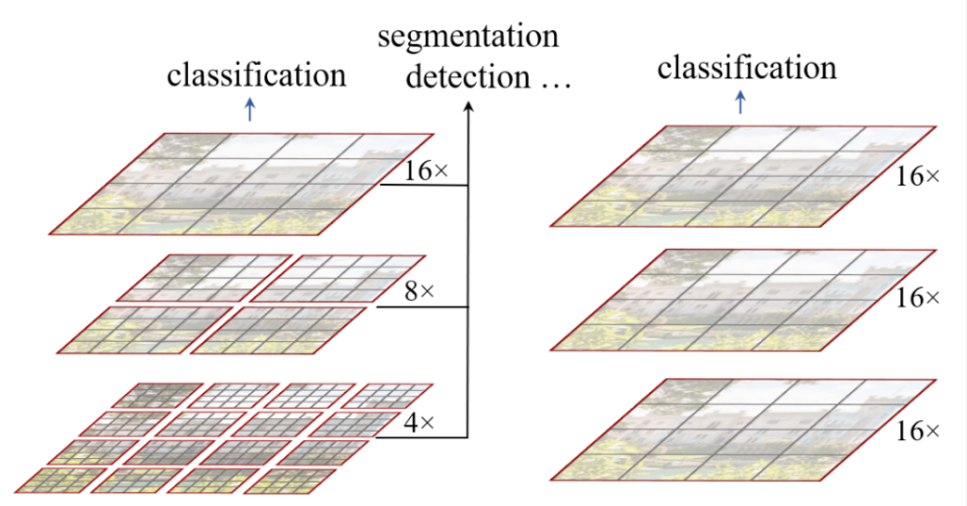
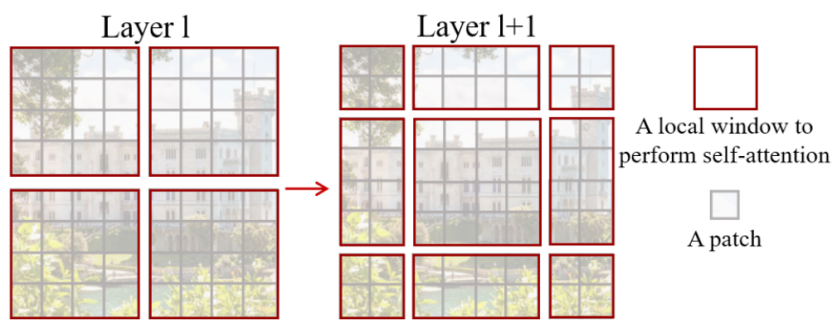


Figure 1.18 – Overview of ViT model [36].

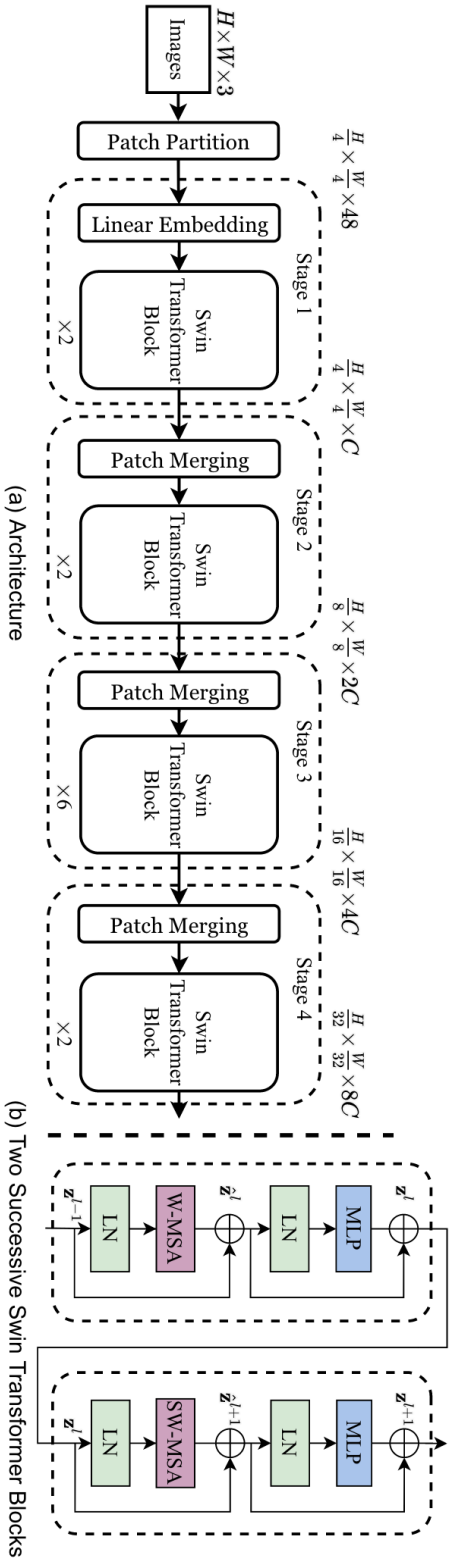
image segmentation).



(a) Hierarchical layers



(b) shifted window



(c) Swin model overview

Figure 1.19 – Swin concept illustration diagram and architecture [37].

Chapter 2

Natural Language Processing

Similar to computer vision (CV), Natural Language Processing (NLP) has been bring forward around for roughly fifty years. This subfield of computer science and artificial intelligence focuses on the interaction between computers and human language. In contrast to CV, which emphasizes visual information, NLP emphasizes natural language comprehension. Significant progress has been made in machine learning techniques and strategies to develop algorithms and models that enable computers to comprehend, interpret, and generate natural language. NLP encompasses a vast array of tasks, including text classification, sentiment analysis, machine translation, context information extraction, and question answering system. This chapter introduces some breakthroughs and the problems they solved in the NLP domain. As with the vision domain, NLP algorithms are introduced in accordance with classification of Machine Learning (ML) in Figure 1.1.

2.1 Shallow Machine Learning in NLP

The first issue in natural language processing is how to represent human language in a computer in a uniform manner. In contrast to vision, where the intensity of a visual signal can be independently represented by the value of a pixel, words in language cannot be represented directly in a computer and can only be recorded as a character transformation code for textual information. Because there are thousands of human languages, a massive dictionary would be required if each word were treated separately. According to the Oxford English Dictionary, there are approximately 170,000 words in the vocabulary in the English language, or 220,000 if obsolete words are included. Global estimates of the quantity of human languages range from 5,000 to 7,000 [1]. With such a large "world of words", we need efficient methods to represent words.

1. https://en.wikipedia.org/wiki/Language

2.1.1 Word Embedding

In a computer, text is not divided into words but stored in characters, one by one. Figure 2.1 give an example text "Who is John Doe?", and its stored code by characters. Those who understand English can comprehend the meaning of the text "Who is John Doe?" that follows, whereas a computer interprets it as an ordered series of 0-1s.

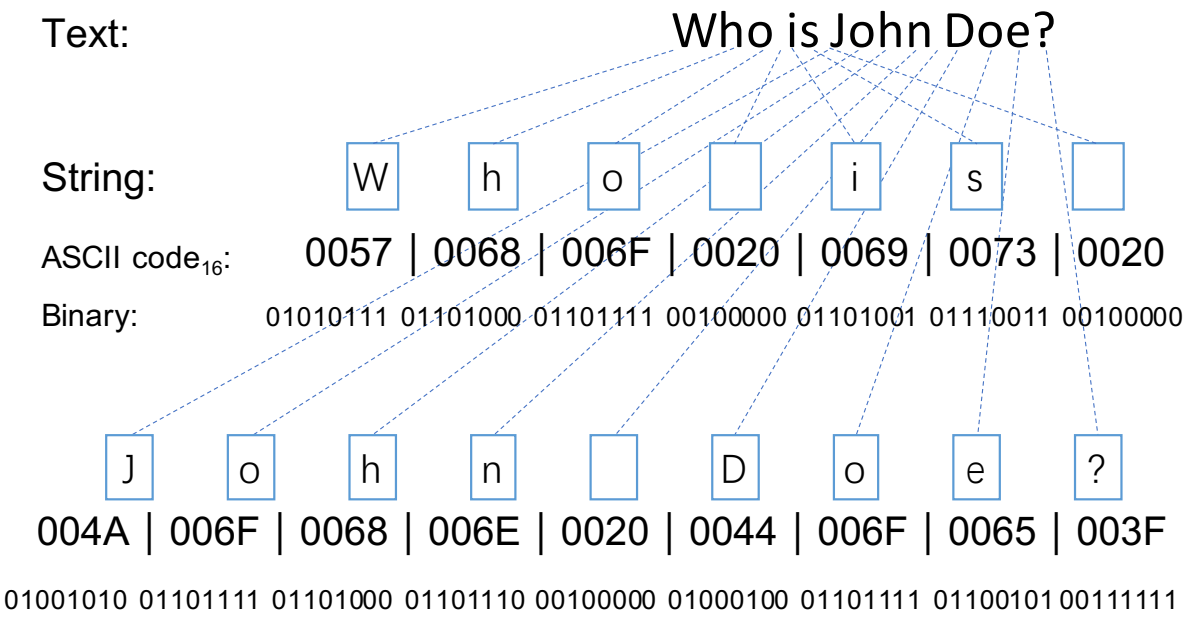


Figure 2.1 – Stored code of "Who is John Doe?".

Word embedding is a natural language processing technique that converts text words (stored code for computer) into vector representations. It is based on the distributional assumption that similar words tend to occur in similar context positions and frequencies. Unlike a document stored as individual characters on a computer, word embedding treats each word as a distinct unit in natural language.

One-hot encoding

The one-hot method is a simple approach to representing words as binary numbers by creating an n-dimensional vocabulary of 0-1s to index each occurrence of N words in a document. Figure 2.2 shows the one-hot corresponding of a ten words example document. However, this method has significant drawbacks. For example, a separate set of index tables must be created for each file, which then needs to be merged and aligned across

files' vocabulary. Additionally, creating a vocabulary index tables containing all the words in a language corpus would lead to a huge amount of redundancy, as the total number of words would be very large. Finally, the one-hot method does not capture any semantic relationships between words.

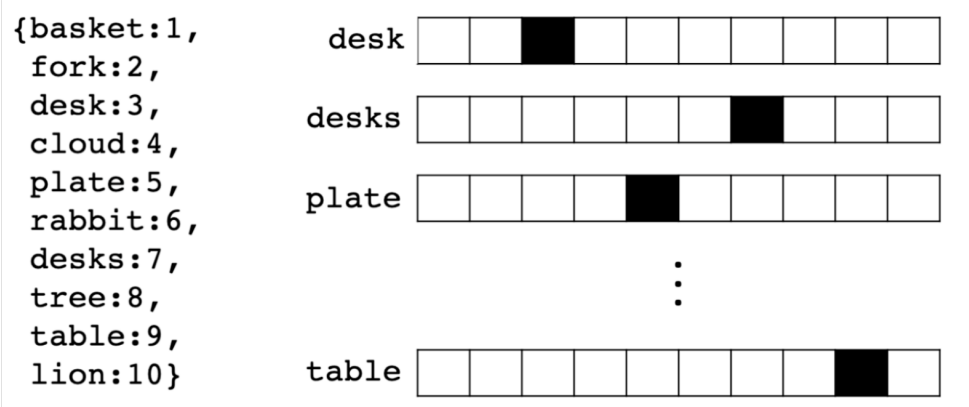


Figure 2.2 – The one-hot representation for a ten words document [38].

Word to vector

If we use a continuous value instead of the 0-1 binary, we can represent different words in distinct directions within a plane or a space. This reduces the dimension of the index table and also computes the distance between vectors. A space that projects words with similar semantic properties to similar distances is referred to as a latent semantic space. Figure 2.3 depicts the example from Figure 2.2 in a three-dimensional latent semantic space.

N-gram language model: If we want to achieve this result, we also need the N-gram language model, which was proposed by Bengio et al.[39]. A language model is a probability distribution over sequences of words [40]. Words in a language are strongly related to the words that come before them. Using a language model, we can infer the probability of the next word based on the preceding words and their order. In an N-gram statistical model, the i^{th} word's conditional probability $P(w_1, ..., w_t)$ were supposed by given all of the preceding words' probabilities $(w_1, ..., w_{i-1})$, which can be depicted by a

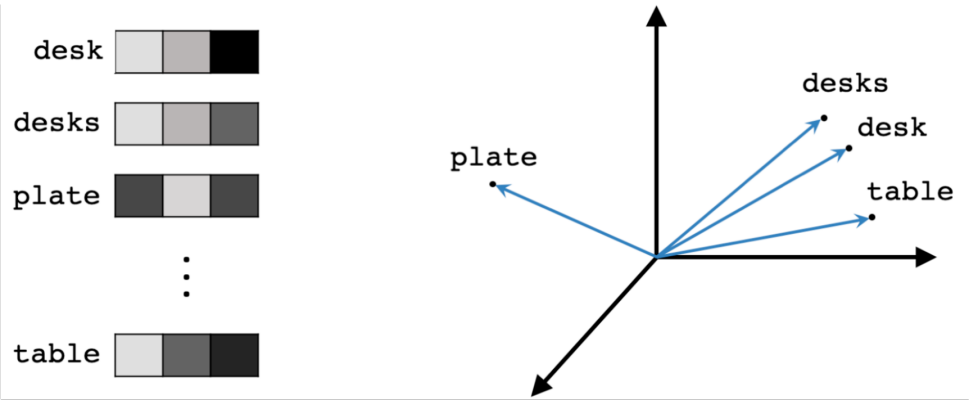


Figure 2.3 – Smaller distances between each vectors which have the similar semantic attributes [38].

Markov chain [41] in equation

$$\hat{P}(w_1, ..., w_i) = \prod_{t=1}^{i} \hat{P}(w_t | w_1, w_2, ..., w_{t-1}). \tag{2.1}$$

When the conditional probability of a word in a language model is highly dependent on the N preceding words' observing probability, the probability model can be approximated as a conditionally independent probability model, which can be described by

$$\prod_{t=1}^{i} \hat{P}(w_i | w_1, w_2, ..., w_{t-1}) \approx \prod_{t=n}^{i} \hat{P}(w_t | w_{t-n+1}, w_2, ..., w_{t-1}). \tag{2.2}$$

Bengio et al.[39] train a shallow neural network to predict i^{th} word, which has shown in Figure 2.4.

Using this N-gram language model, we can map word probabilities directly to the latent space based on the weight matrix C that has been learned.

2.1.2 Information Retrieval

Information retrieval (IR) aims to find relevant information based on a query using text. Sometimes, the desired documents are buried in a large collection of resources. Text retrieval involves matching the features of the documents to the query. Commonly used information retrieval techniques include Bag-of-Words and Tf-idf.

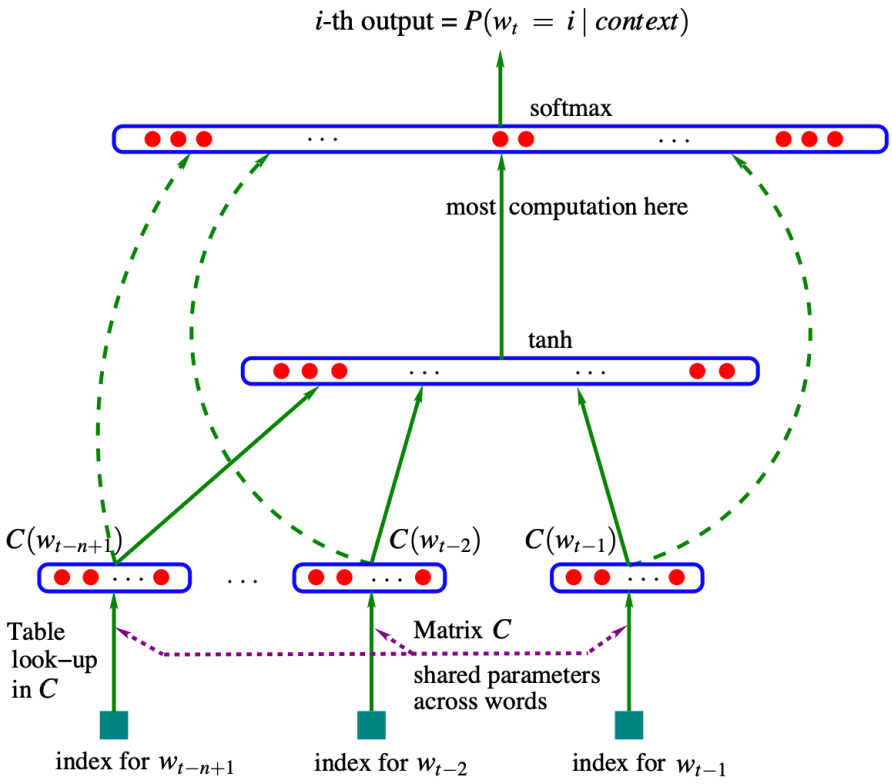


Figure 2.4 – N-gram with a shallow neural network architecture: $\prod_{t=n}^{i} \hat{P}(w_t|w_{t-n+1}, w_2, ..., w_{t-1}) = g(i, C(w_{t-n+1}, ..., C(w_{t-1})))$, where g is the neural network, $C(i)$ is the i^{th} word feature vector [39].

Bag-of-Words model

The Bag-of-Words (BoW) model counts the occurrence of each word in a document or the cluster of features embedded in the latent space. Regardless of their order, BoW only considers the frequency of each word. Sometimes, BoW is also based on N-gram, which groups N words into a single bag. The BoW model has also been used for computer vision (CV), which counts the handcrafted features directly or combination with visual features clusters as a new features for image retrieval. Table 2.1 gives the BoW model for example document 1.

Example document 1:

Language consists of grammar and vocabulary. Grammar is language's structural constraints.

word	count
language	2
is	1
consist	1
of	1
grammar	2
and	1
vocabulary	1
structural	1
constraint	1

Table 2.1 – BoW of example document.

Tf-idf

Although BoW provides a representation of document features based on word frequency or groups of words, it does not take into account the length of the document or the number of documents in the corpus. Tf-idf, which stands for term frequency-inverse document frequency, reflects how important a word is to a document in the corpus. Tf-idf consists of two parts: tf and idf. The term frequency is represented by

$$\texttt{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}, \tag{2.3}$$

where $f_{t,d}$ is the count of term t in document d. The inverse document frequency is represent by

$$\texttt{idf}(t, N) = -\log \frac{n_t}{N} \tag{2.4}$$

where N is the number of total documents in the corpus, n_t is the number of documents containing term t. Here is an example for tf-idf. Suppose the example document 1 and document 2 construct a corpus $N = 2$.

Example document 2:

A vocabulary is the sets of words in a language.

We can calculate the tf-idf of word 'grammar' in document 1 and document 2, respectively.

$$\begin{aligned} \texttt{tf-idf}(\text{'grammar'}, d_1) &= \texttt{tf}(\text{'grammar'}, d_1) \times \texttt{idf}(\text{'grammar'}, N) \\ &= \frac{2}{11} \times \log(\frac{2}{1}) \\ &\approx 0.167 \times 0.301 \\ &= 0.050267 \end{aligned} \tag{2.5}$$

$$\begin{aligned} \texttt{tf-idf}(\text{'grammar'}, d_2) &= \texttt{tf}(\text{'grammar'}, d_2) \times \texttt{idf}(\text{'grammar'}, N) \\ &= \frac{0}{10} \times \log(\frac{2}{1}) \\ &= 0 \end{aligned} \tag{2.6}$$

Equation 2.6 assumes that a term has meaning only when it appears in document 2. Nonetheless, in reality, a more accurate idf can be obtained by using $1 + n_t$ as the denominator, where n_t is the number of documents in the corpus that contain the term. Tf-idf is commonly used by search engines to calculate page weights due to its simplicity in computation and updating. However, it has a disadvantage in that, for very long documents, important word frequencies can be overshadowed by less specific words.

2.2 Deep Learning in NLP

The early approach to machine translation assumed that it was a simple task of corresponding nouns in different languages and applying grammatical rules. However, it soon became clear that this was far from reality. In the 1960s and 1970s, IBM attempted to build a comprehensive system of English grammar rules, but it failed because the grammar rules could not account for all linguistic phenomena. There are terms with multiple meanings. "bank" typically refers to a financial institution (such as "BNP Paribas"), whereas "water bank" refers to a piece of land along the side of a river or lake. As a result, the translation system was not as accurate as it could have been. Machine translation is also known as one of the most important tasks driving progress in the entire NLP field.

Machine translation (MT) can basically be divided into rule-based translation, statistical-based translation and neural network translation. As mentioned earlier, rule-based machine translation has its own limitations. In contrast, statistical-based translation can be seen as a precursor to neural networks, since neural network models are also statistical probability-based models. The breakthrough came in 2014, when Cho et al. [42] proposed the RNN (recurrent neural network) Encoder-Decoder, which achieved excellent translation results and garnered significant attention in the field of NLP.

2.2.1 Recurrent Neural Network

The proposal of the recurrent neural network (RNN) model can be traced back to the 1980s. In 1986, RNN[26] were proposed with the backpropagation algorithm in the same paper for handwriting recognition, which was applied to tasks such as speech recognition and time series prediction. This paper demonstrated the ability of RNNs to deal with sequential data, but they were not widely used at the time due to the difficulty of training them. RNN model can be described by

$$O_t = \sigma(W \times x_t + U \times h_{t-1} + b_o), \tag{2.7}$$

where $\sigma()$ is the activation function (*e.g.*, sigmoid 1.10), x_t is t^{th} term in the input sequences, h_{t-1} is the hidden state from previous step $(t-1)$, W and U are the weight matrices for learning, b_o is the bias. Through training, an RNN associates a word x_t at a given time t with the input at the previous time $(t-1)$ using its hidden state h_{t-1}. The network then generates an output based on this joint processing, and the current state is

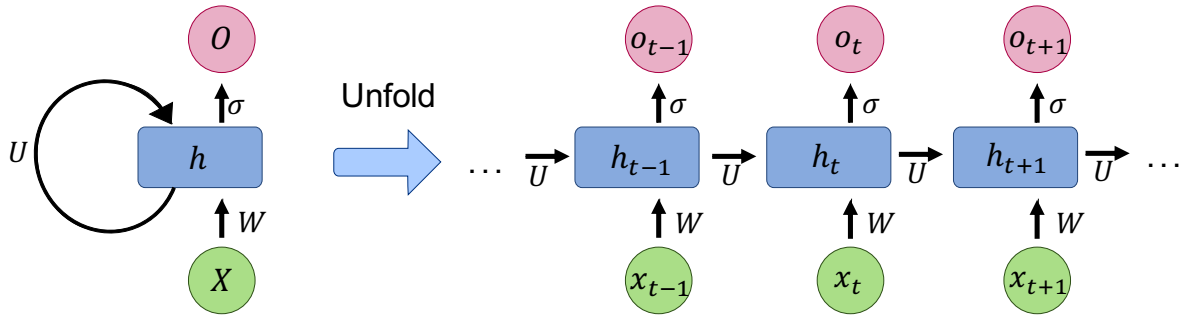


Figure 2.5 – RNN unit unflod, *i.e.*, Equation 2.7 illustration. Figure is modified from online [3].

transferred to the next time step $(t+1)$ using a state transition matrix U. Thus, RNN model functions as artificial neural networks that have the ability to remember temporal information. The Equation 2.7 is shown in Figure 2.5. While RNNs offer memory storage capabilities, training these networks becomes increasingly difficult as the sequence length increases. This is because multiple iterations of nonlinear activation functions can result in gradient explosion or gradient disappearance issues when applying the chain rule to partial derivatives of the learning weights.

Long Short-Term Memory

The Long Short-Term Memory [43] (LSTM) network addresses the limitations of RNNs by incorporating control components that restrict the input and output of hidden states. These components include the forget gate, input gate, and output gate. The forget gate is responsible for filtering the influence of the previous state, while the input gate regulates the increase rate using two types of activation functions: the sigmoid, (*i.e.*, σ (1.10)) and the tanh (1.12). The output gate determines the size of the output passed to the next state. This process can be described using a set of equations:

$$\begin{aligned} f_t &= \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \\ i_t &= \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \\ p_t &= \sigma\left(W_p x_t + U_p h_{t-1} + b_p\right) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \\ h_t &= p_t \circ \tanh\left(c_t\right) \end{aligned} \tag{2.8}$$

3. `https://upload.wikimedia.org/wikipedia/commons/b/b5/Recurrent_neural_network_unfold.svg`

where the $\circ$ is the element-wise product, f_t is forget gate activation vector, i_t is input gate vector, and p_t is output gate vector, and σ is the sigmoid function. Figure 2.6 show those Equations 2.8. LSTM can avoid the issue of gradient explosion or gradient vanishing

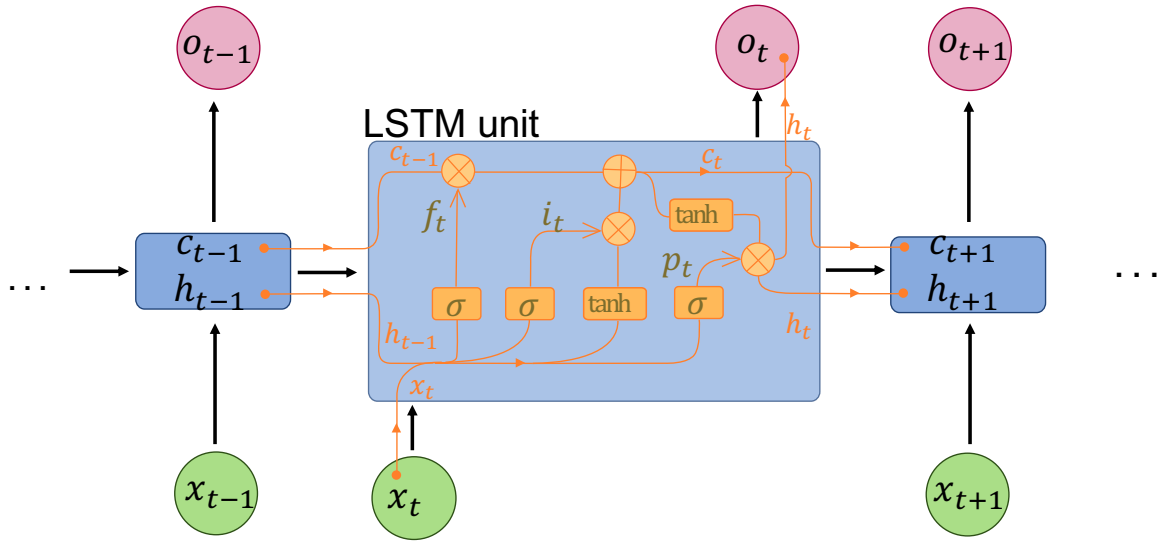


Figure 2.6 – Details of a LSTM unit corresponding to Equations 2.8. Figure is modified from online [5].

during the training backpropagation even input with a long length sequence via controling those three gates. However, the disadvantage is that the calculation is complex and also reduces the learning efficiency of each unit.

Gated Recurrent Unit

Gated Recurrent Unit (GRU[42]) is proposed to minimize the computation of LSTM and enhance the efficiency of original version of RNN. GRU just use reset gate to control the rate of input from previous state $(t-1)$, and output to the next state $(t+1)$, while an update to determine the rate update. GRU unit can be depicted in a set of equations:

$$\begin{aligned} z_t &= \sigma\left(W_z x_t + U_z h_{t-1} + b_z\right) \\ r_t &= \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right) \\ \tilde{h}_t &= \tanh\left(W_h x_t + U_h\left(r_t \circ h_{t-1}\right) + b_h\right) \\ h_t &= (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t \end{aligned} \tag{2.9}$$

5. https://commons.wikimedia.org/wiki/File:Long_Short-Term_Memory.svg

where the $\circ$ is the element-wise product, z_t is update gate's vector, r_t is the reset gate's vector, and σ is the sigmoid activation function. Figure 2.7 shows the internal structure of a GRU unit. Compared to LSTM, the GRU reduces the number of control gates by one and

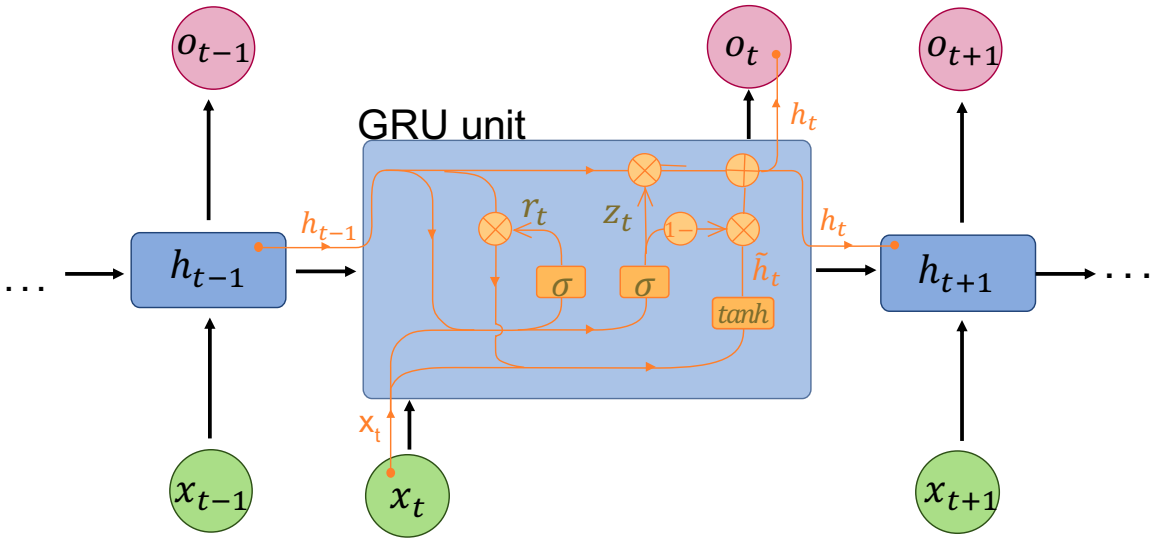


Figure 2.7 – Zoom in a unit of GRU. Corresponding to Equations 2.9. Figure is modified from online [7].

uses only six weight matrices, while achieving similar results as LSTM which requires eight weight matrices. This reduction in parameters greatly improves computational efficiency but reduces the computational load required for training and inference.

Cho et al. [42] use GRU to do machine translation, which is divided into an encoder network and a decoder network. Figure 2.8 shows the structure of this Encode-Decoder. The encoder part embeds sentences from one language X, the decoder calculates the pair sentences in corresponding language Y. C indicates the transfer weights from encoder to decoder.

One of the roles of RNNs is to embed words in the language into vector space. This embedding process can greatly help the algorithm understand natural language, *e.g.*, Word2vec.

Word2vec

Word2vec is a word embedding model see section (2.1.1) that can generate word vectors in the latent space based on RNN model [44]. Word2vec contains two different embedding

7. `https://commons.wikimedia.org/wiki/File:Gated_Recurrent_Unit.svg`

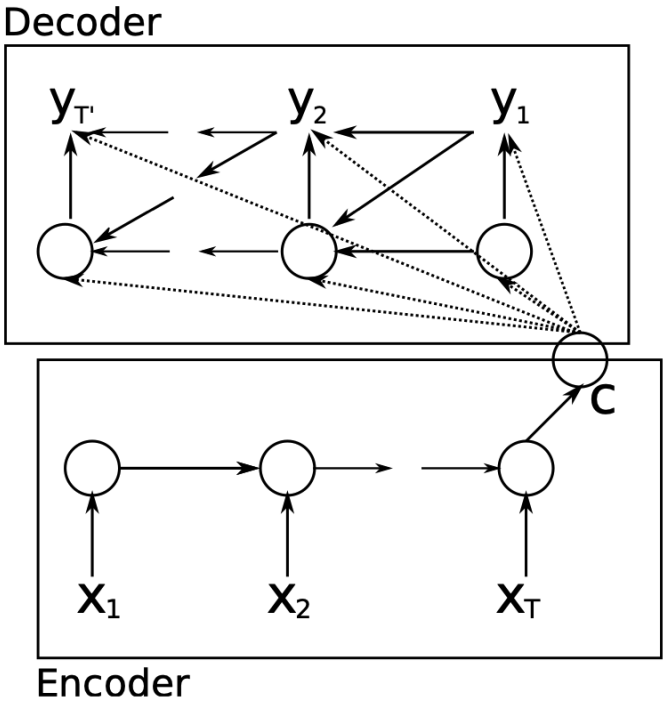


Figure 2.8 – The structure of RNN Encoder-decoder [42].

models: the Continuous Bag-of-Word (CBOW) and the Skip-gram. The CBOW model predicts the target word based on the context word, whereas the Skip-gram model predicts the context word based on the target word, which can be understood more clearly in Figure 2.9. Theoretically, the relationship is defined by subtracting and adding two groups of same concept word vectors. For example, `Paris` − `France` = `Rome` − `Italy`. Figure 2.10 showns the visualization of word2vec embedding vectors of first 2-dimensions.

RNN models are rapidly gaining popularity in various areas of NLP. It has been discovered that combining two RNNs can result in a double-linked LSTM or GRU, which has shown to produce good results. Even more impressive is the use of both forward and reverse inputs on an ordered sequence, such as a sentence, which provides a significant performance boost. These bi-directional LSTMs or bi-directional GRUs are referred to as Bi-LSTMs and Bi-GRUs.

RNN models, including LSTM and GRU, offer memory functions and provide flexibility to adapt to various tasks. These tasks may include machine translation, text feature extraction, relationship extraction, information retrieval, summary abstraction, chat system, and others that require time series, such as continuous handwriting recognition,

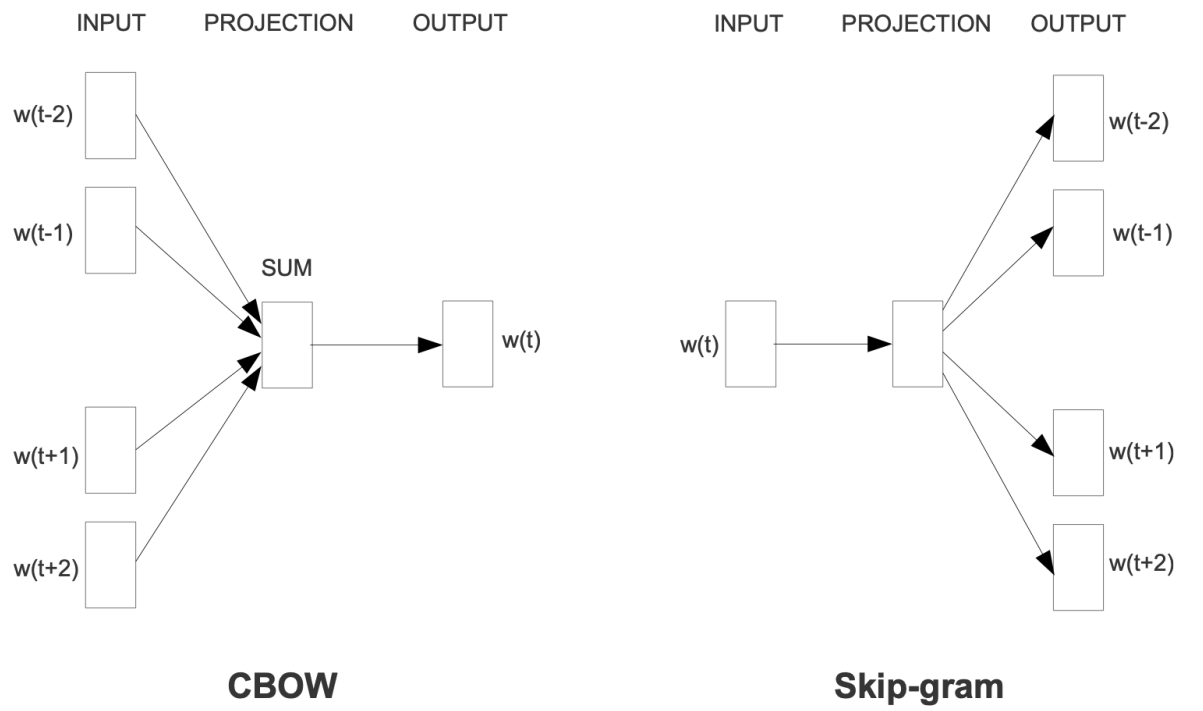


Figure 2.9 – Illustration for CBOW model and Skip-gram model [44].

speech recognition, and video prediction. However, RNNs have their drawbacks. Firstly, they are computationally intensive and often require multiple iterations of training. Additionally, because each hidden state depends on the previous state for computation, it is like having multiple serially connected computational units that cannot perform parallel data processing.

2.2.2 Transformer-Based Model

The Transformer is a machine translation model proposed by Vaswani et al. in 2017[34]. It utilizes the multi-head attention mechanism to capture the intrinsic relationship between embedding vectors, which is similar to the high-level semantics in natural language. Transformer-based models have demonstrated superior performance compared to RNN models on many NLP tasks. As a result, they have become a cornerstone in the field of NLP.

According to the development process of transform, we introduce the attention mechanism of transform, the basic framework, and the extension application of transform in the following sections.

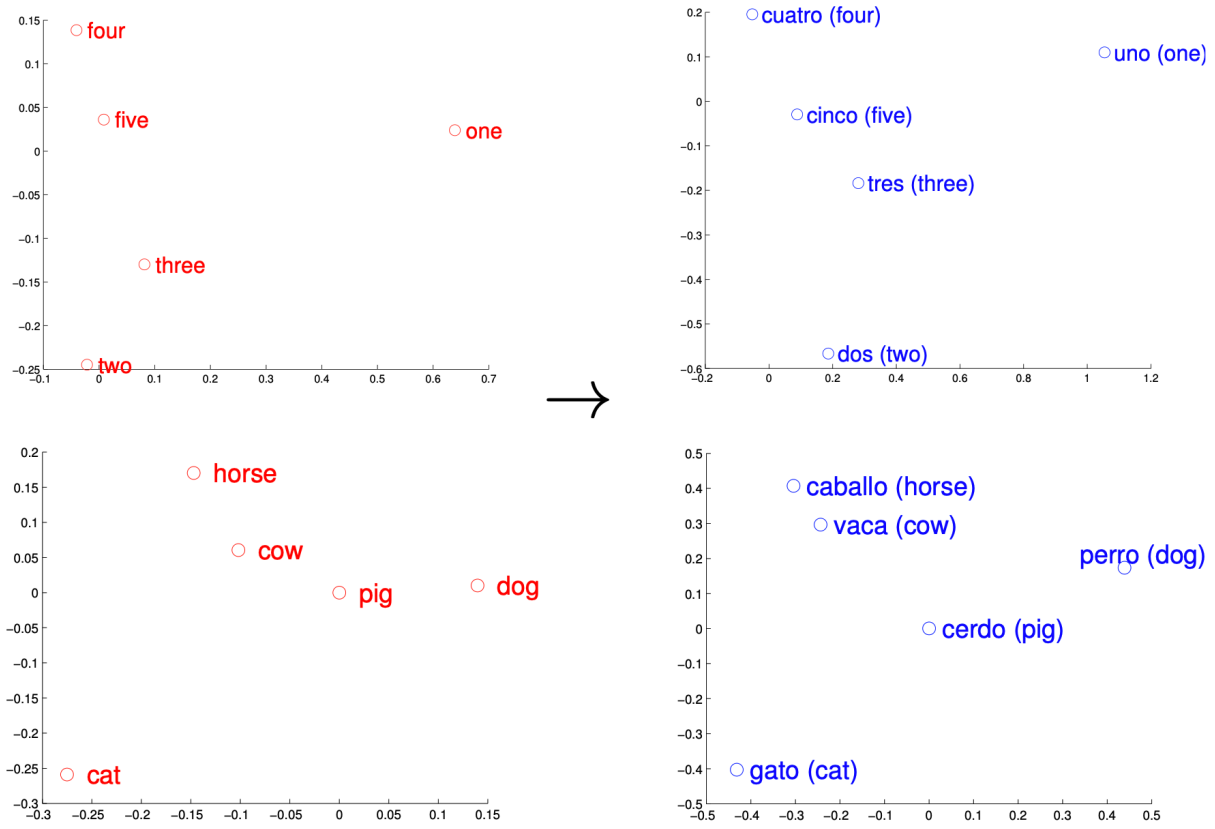


Figure 2.10 – Word2vec embedded vectors in visualization. The points in left column represent English words, the points in right column represent Spanish words. The same concepts have similar geometric arrangements in both spaces [45].

Attention Mechanism

Bahdanau et al. [46] propose an attention mechanism to overcome the issue of RNN Encoder-decoder model see section (2.8), that is fixed-length vector problem. The length of decoder must be equal to length of encoder. Each word has a corresponding unique probability distribution with input order. With the attention mechanism, information can be dispersed throughout the sequence of annotations, from which the decoder can focus on retrieved data. This attention mechanism adds attention score $\mathbf{s}$ between RNNs' hidden layers and outputs, as follow:

$$\mathbf{s} = (h_e h_d)\,, \tag{2.10}$$

where h_e, h_d indicate hidden states of RNN encoder and decoder, respectively. This score could get the attention distribution α, and scaled by `softmax` function:

$$\alpha = \texttt{softmax}(\mathbf{s}), \tag{2.11}$$

that means,

$$\alpha_t = \frac{e^{\mathbf{s}_t}}{\sum_{t=1}^{N} e^{\mathbf{s}_t}}, \tag{2.12}$$

where N is the total number of input words. Then, this attention distribution could be seen as an attention weight a for the encoder hidden states:

$$\mathbf{a} = \sum_{t=1}^{N} \alpha_t h_e. \tag{2.13}$$

Finally, the RNN output with attention mechanism will be:

$$\mathbf{o} = \sigma(\mathbf{a} h_d), \tag{2.14}$$

where σ represents the activation function, *e.g.*, tanh see (Eq. 1.12). Figure 2.11 shows the visualization results of attention mechanism for translation from English to French.

Transformer

In the Transformer architecture, Vaswani et al. [34] found that it is beneficial for the encoder to learn the attention patterns between its own internal components before transferring them to the decoder. The question is "how to learn attention of oneself". In the Transformer architecture, three weight matrices are used to transform the input

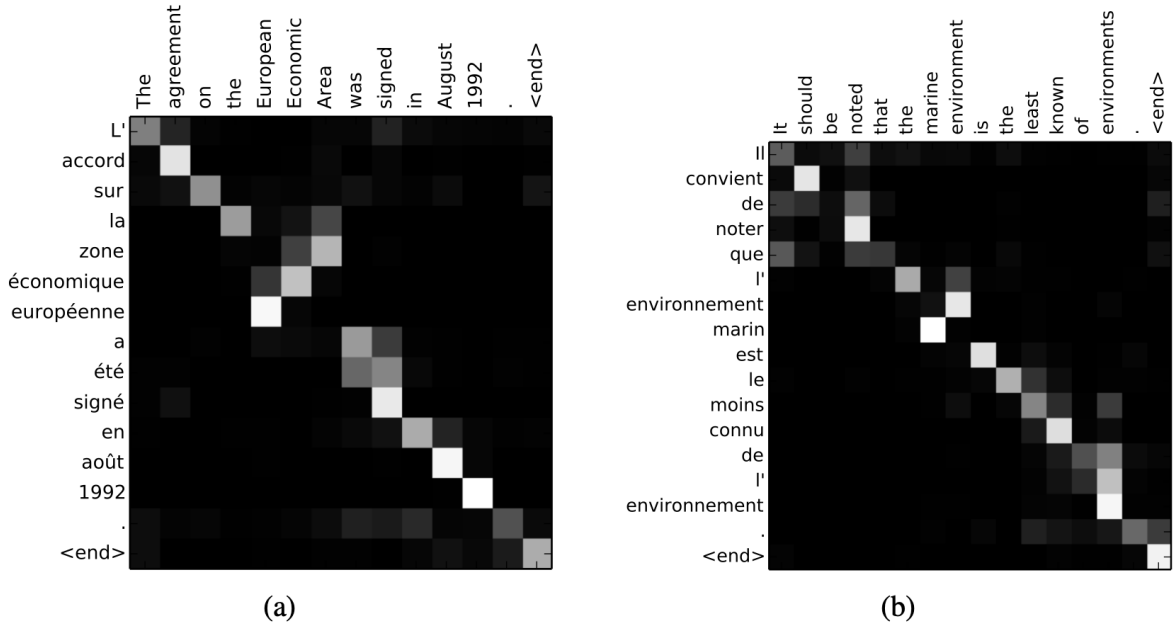
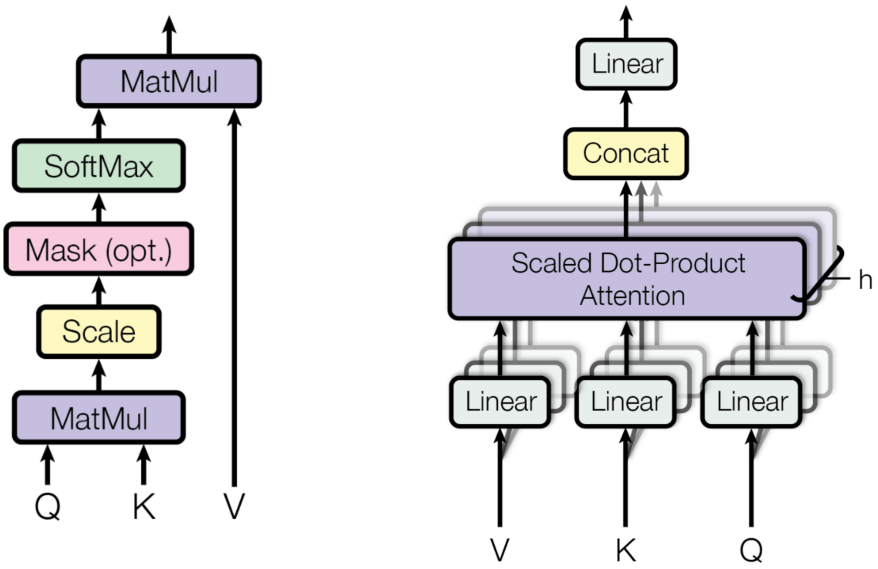


Figure 2.11 – Attention distribution matrices of English-French alignments. The more white patch means the higher attention probability [46]. In column (a), with the attention mechanism, this RNN model translates phrase of "European Economic Area" into reverse order in French "zone économique européene". In column (b), the phrase "marine environment" also translates correctly with the reverse order phrase of French "l'environnment marin".

vector into queries (Q), keys (K), and values (V) matrices, respectively. The Q and K matrices are then multiplied to obtain the attention distribution using `softmax` function, which is applied to the scaled dot-product of Q and K. Finally, the attention distribution is used to weight the V matrix and compute the self-attention representation of the input vector. The self-attention distribution can be represented in equation:

$$\text{Attention}(Q, K, V) = \texttt{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{2.15}$$

where d_k indicates the dimension of K (keys) matrix. The self-attention mechanism can be viewed as a lookup table that matches input queries with keys and combines their associated values, similar to the "bag-of-words" model. The self-attention Equation 2.15



(a) Scaled self-attention (Eq.2.15) (b) Multi-head structure

Figure 2.12 – The visualization of multi-head self-attention [34].

is shown in Figure 2.12(a) as well.

A single self-attention layer captures attention from one perspective. By stacking multiple self-attention layers, the encoder can find attentions from multiple perspectives. Multi-head attention stacks many self-attention layers, and concatenates their output, as shown in Figure 2.12(b), to form the encoder part of the Transformer. In the decoder

part, a masked operation randomly masks one word to increase the attention of the other words to each. The Transformer solves the problem of sequential computations in RNNs, meaning that its multi-head self-attention mechanism can be computed in parallel, and the results can be concatenated together. This greatly improves the speed of sequence processing tasks.

BERT

Devlin et al. [47] introduced the BERT (Bidirectional Encoder Representations from Transformers) language model in 2018. BERT, in contrast to Transformer, employs masked language modeling by arbitrarily masking out words in a sentence and training the model to predict the absent words based on context. BERT employs a bidirectional strategy in which both preceding and subsequent terms are considered. BERT is designed specifically for natural language understanding (NLU) tasks, such as question answering system and sentiment analysis. Whereas the original Transformer is intended for broad sequence-to-sequence tasks. BERT is typically fine-tuned for a subsequent task using a reduced labeled dataset, allowing it to adapt to the tasks in other domain. Figure 2.14 shows the fine-tuning BERT for multiply NLP tasks.

MNLI [48] is a dataset for evaluating natural language inference models, NER [49] is a task for recognizing and classifying named entities in text, and SQuAD [50] is a dataset for evaluating reading comprehension models.

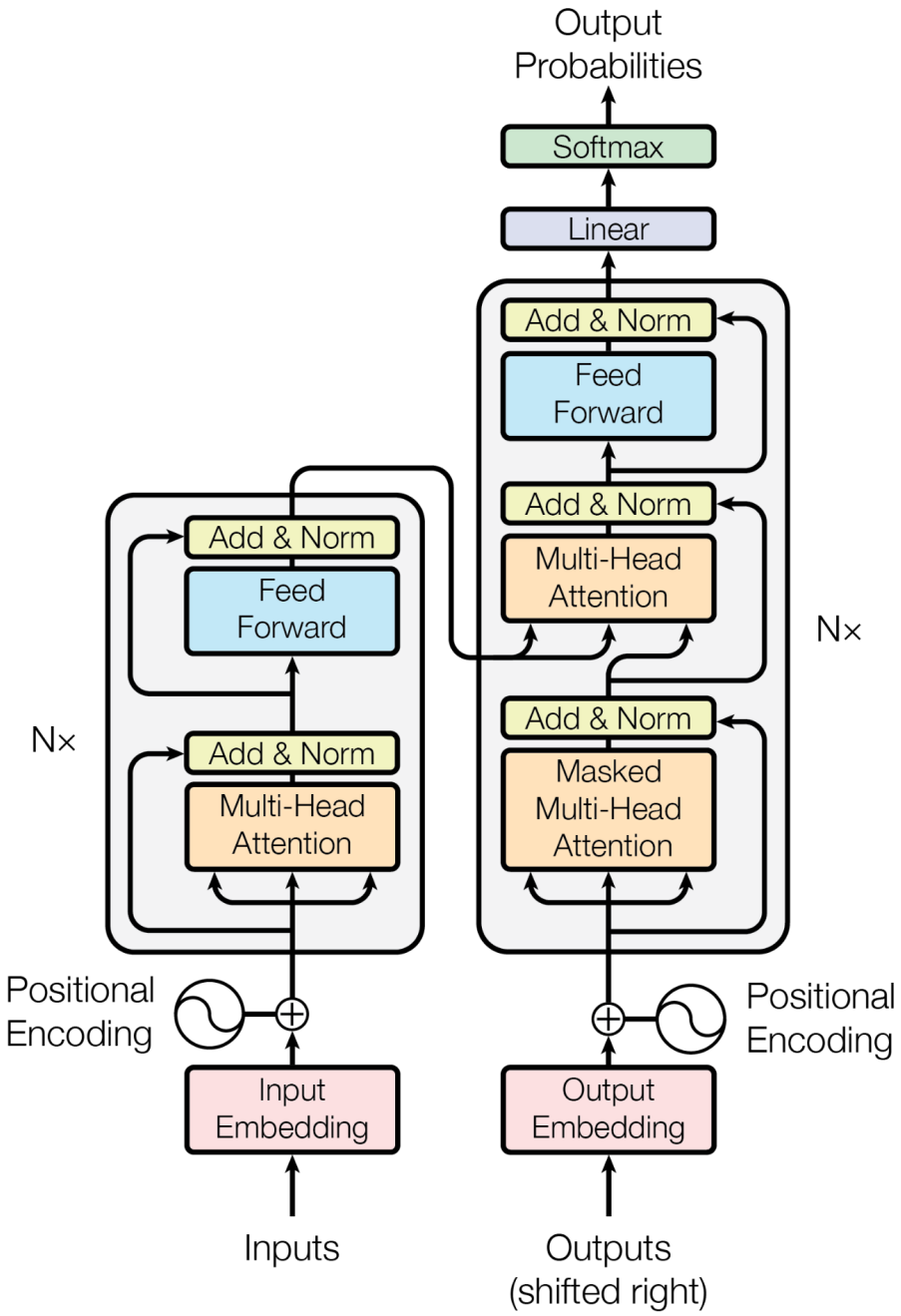


Figure 2.13 – The Transformer architecture [34].

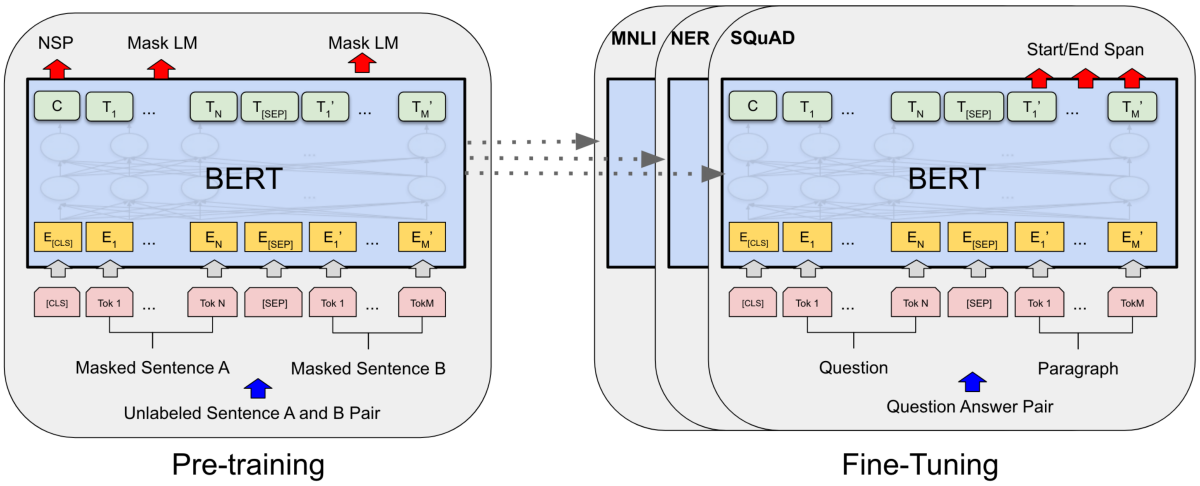


Figure 2.14 – Over view of pre-train and fine-tune of BERT [47].

Chapter 3

Multimodal Learning

In the previous two chapters, we learned about feature extraction methods for computer vision (CV) and natural language processing (NLP). In this chapter we focus on how they are projected into the shared latent space and computed in this space. The projection and computation correspond to the fusion phase and alignment phases, respectively.

3.1 Multimodal Fusion

As the Figure 7 demonstrates, in general, multimodal algorithms process different modalities data with different branches. Without considering to bridge the multimodality semantic gaps, we can choose a fixed dimension and project the obtained heterogeneous features onto the space of this dimension.

The question is "what is multimodality semantic gap?" The **multimodality semantic gap** is the difference of heterogeneous data with a same concept. For example, for a concept of "cat", visual modality is represented by an image, while textual modality is represented by word. We can quantify this multimodal semantic gap by a *concept entropy*. Assuming this "cat" image is a 32×32 256-bits pixels image, then the information entropy of the visual modality can be represented by $\mathtt{E}(i)$, while this "cat" word is represented by an index of a dictionary with 1000 vocabulary words, then the information entropy of the textual modality can be represented by $\mathtt{E}(t)$. According to the Shannon information entropy formula:

$$\mathtt{E}(i) = -\log_2 \tfrac{1}{256 \times 32 \times 32} = 13 \tag{3.1}$$

$$\mathtt{E}(t) = -\log_2 \tfrac{1}{1000} = 9.03 \tag{3.2}$$

We recognize that the different information entropies of the unified concept are significant cause of the multimodality semantic gap. A low-dimensional fusion space can be utilized to reduce information redundancy and computational complexity when dealing

with multimodal tasks such as classification or retrieval that require relatively low information entropy. To completely leverage the features, tasks such as recognition and segmentation, which have high information entropy requirements, require a high-dimensional fusion space. Mapping the features to a high-dimensional fusion space enables a more comprehensive representation of the relationships between various modalities, which enables accurate recognition and segmentation. Nonetheless, this method incurs increased computational complexity, which can be a significant obstacle for large-scale applications. Consequently, the selection of the proper fusion space is a crucial aspect of the design of efficient multimodal algorithms.

3.1.1 Dual Projection

In most double-branch multimodal models, *i.e.*, one branch for vision, one branch for text, a dual projection is used. As the Figure 3.1 shows, CMPM [51] uses a CNN to extract visual features, a Bi-LSTM to extract text feature, then project them onto a latent space. Both image feature and text feature have been reduced to the same size. It is worth noting that the CMPM model uses the mini-batch size as the dimensionality of the learning vector, *i.e.*, the latent space dimension. Table 3.1 shows the results of different size of latent space dimensions. 32-dimension is better than others. This means that the number of space dimension is not necessarily the higher the better for retrieval tasks.

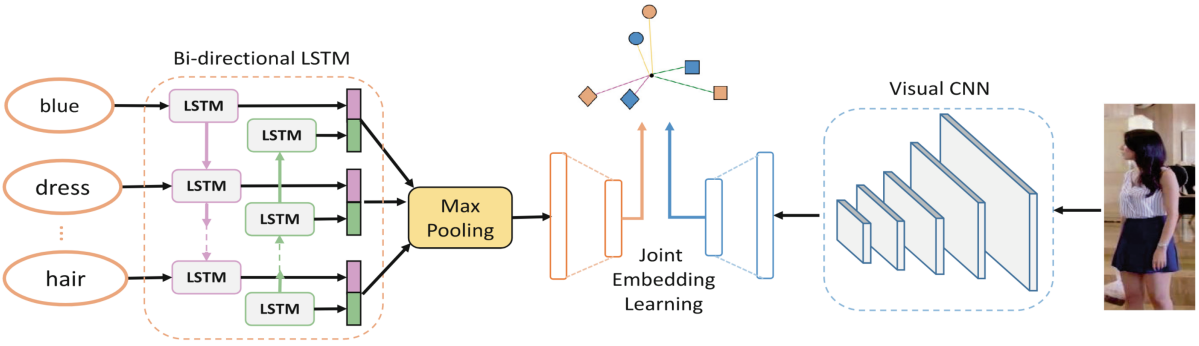


Figure 3.1 – A typically double-branch cross-modal retrieval structure [51].

Matching Loss	Text-to-Image				Image-to-Text			
	16	32	64	128	16	32	64	128
$KL(q_i \text{E} p_i)$	42.58	43.81	41.89	36.06	41.87	38.81	22.35	19.97
CMPM	42.28	43.42	44.02	42.43	51.95	52.09	51.98	48.67

Table 3.1 – The latent space dimensions vs the performance in retrieval task [51].

3.1.2 Encoder-Decoder

Andrej and Li [52] propose a multimodal generative model, which uses CNN as encoder. Then through bottleneck (*i.e.*, the place where connects encoder and decoder) transfer into a multimodal RNN. This RNN are not training for matching features but for predict next word of generation. Figure 3.2 shows, the image features are passed to the

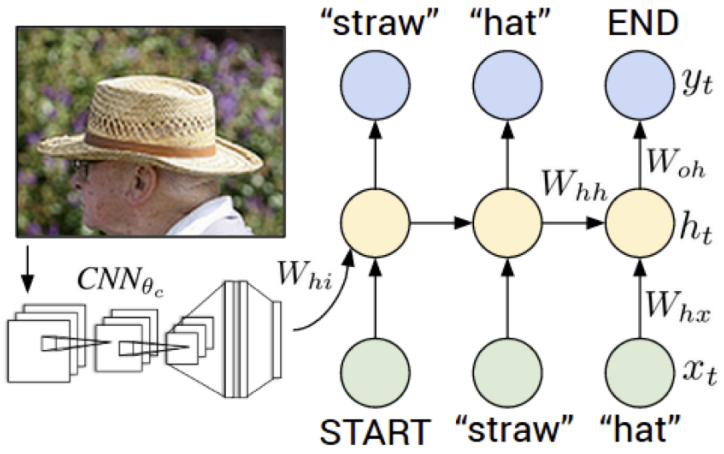


Figure 3.2 – A multimodal generative model architecture [52].

RNN in one direction, while the dual projection structure in Figure 3.1 allows the results to be calculated in both directions.

3.2 Multimodal Alignment

Multimodal alignment in cross-modal domain is the process of reducing the distance between the feature distributions of two or more modalities in a shared latent space. By aligning the features across modalities, we can better capture their underlying relationships and enhance the precision and efficiency of cross-modal model. Various techniques, such as canonical correlation analysis (CCA), adversarial training, and metric learning,

can be used to accomplish multimodal alignment. The choice of a method is determined by the particular features of the modalities and the level of alignment desired.

3.2.1 Contrastive Learning

Due to the fact that double-branch structure parameters are not fixed (in training stage), reducing the distance between samples and labels in the latent space is not directly computable in multimodal data. If the loss function of samples and lables in two directions is minimized directly, features collapse may occur.

Contrastive learning has become a common learning strategy in multimodal model to address this issue. This method divides the data into positive and negative sample pairs and attempts to maintain or increase the distance between negative sample pairs while decreasing the distance between positive sample pairs. Contrastive learning effectively balances the optimization of similarity and dissimilarity measures, thereby preventing feature collapse and enhancing multimodal model performance.

Although [53] proposed an early version of the contrastive learning approach in 2006, it was not until the introduction of FaceNet [54] in 2015 that it became widely adopted in the field of computer vision and machine learning. FaceNet proved the efficacy of contrastive learning in learning high-quality feature representations for face recognition tasks and paved the way for its implementation in other domains, such as multimodal information retrieval. FaceNet using anchor vector $\mathbf{a}$ indicates the sample, positive vector $\mathbf{p}$ indicates the paired label, negative vector $\mathbf{n}$ indicates the non-paired label. The **Triplet Loss** can be described:

$$\texttt{Triplet Loss} = \sum_{i}^{N} \left[Ł_2 \left(\mathbf{a} - \mathbf{p} \right) - Ł_2 \left(\mathbf{a} - \mathbf{n} \right) + \alpha \right]_+ , \tag{3.3}$$

where $Ł_2$ represents $Ł_2$-distance, α represents the margin that enforce at the distinction between negative pair and positive pair. Through triplet loss learning, the distance between the anchor and the positive sample is reduced to one margin away from the distance between the anchor and the negative sample. Figure 3.3 shows the learning process of Triplet Loss.

The essence of the basic idea of triplet loss did not change remarkably in the subsequent contrastive learning methods. In addition to matching, triplet loss is also widely used for retrieval and prediction tasks.

Triplet loss, a popular variant of contrastive learning, was not immediately applied to multimodal domain. This is primarily due to multimodality involving two sets of feature

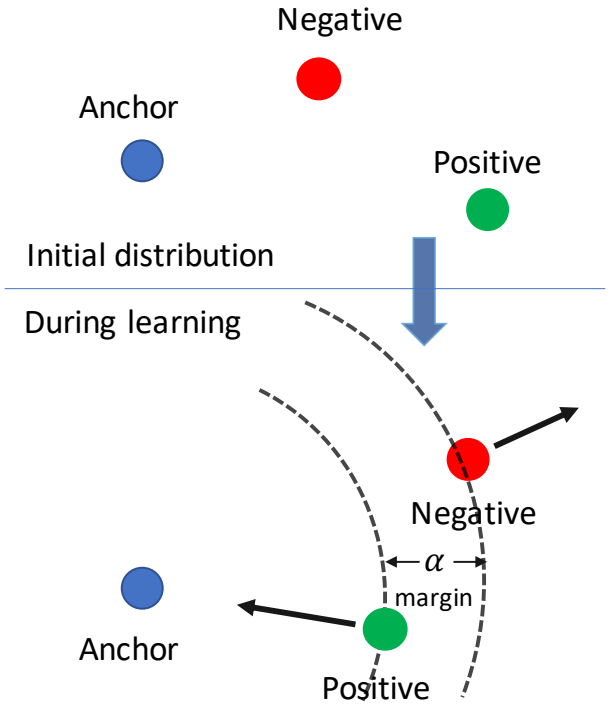


Figure 3.3 – Triplet Loss illustration. During the learning process, the positive samples are drawn apart and the negative samples are pushed apart until they are separated by a margin. (Eq. 3.3)

samples in the same space. Using one sample as a negative example may result in it being a positive sample for the other set of feature samples. As shown in Figure 3.3, the negative sample is pushed away from the anchor direction and is likely to be far from the positive sample of the other set of feature samples. This issue, known as the "modality mismatch" problem, can compromise the performance of the multimodal model.

VSE++

VSE++ [55] improves triplet loss with a double direction Max of Hinges loss (the hinge loss Cf. 1.1). The Max of Hinges loss only pushes the hardest negative pair in one direction, and only apply this function on the max distance in a mini-batch. The Max of Hinges can be depicted by:

$$\begin{aligned} \texttt{Max of Hinges Loss} = \quad & \max_{\mathbf{t}'} \left[\alpha + \mathbf{i} \times \mathbf{t}' - \mathbf{i} \times \mathbf{t}\right]_+ \\ & + \max_{\mathbf{i}'} \left[\alpha + \mathbf{i}' \times \mathbf{t} - \mathbf{i} \times \mathbf{t}\right]_+ , \end{aligned} \tag{3.4}$$

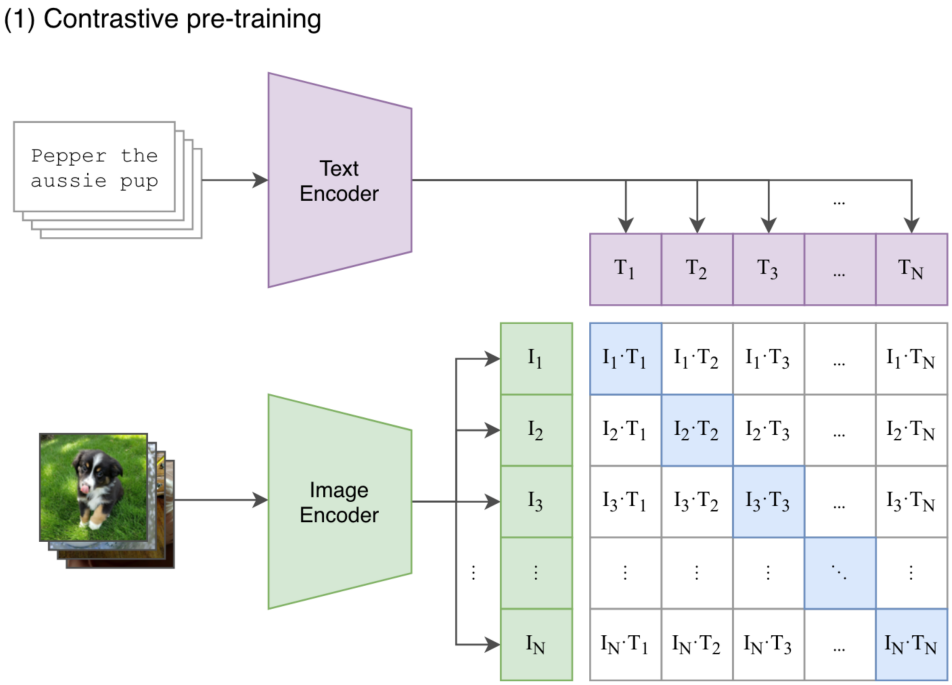
where $\mathbf{t}'$ and $\mathbf{i}'$ refer to

$$\begin{aligned} \mathbf{t}' &= \arg\max_{\mathbf{d} \neq \mathbf{t}} \mathbf{i} \times \mathbf{d}, \\ \mathbf{i}' &= \arg\max_{\mathbf{j} \neq \mathbf{i}} \mathbf{j} \times \mathbf{t}, \end{aligned}$$

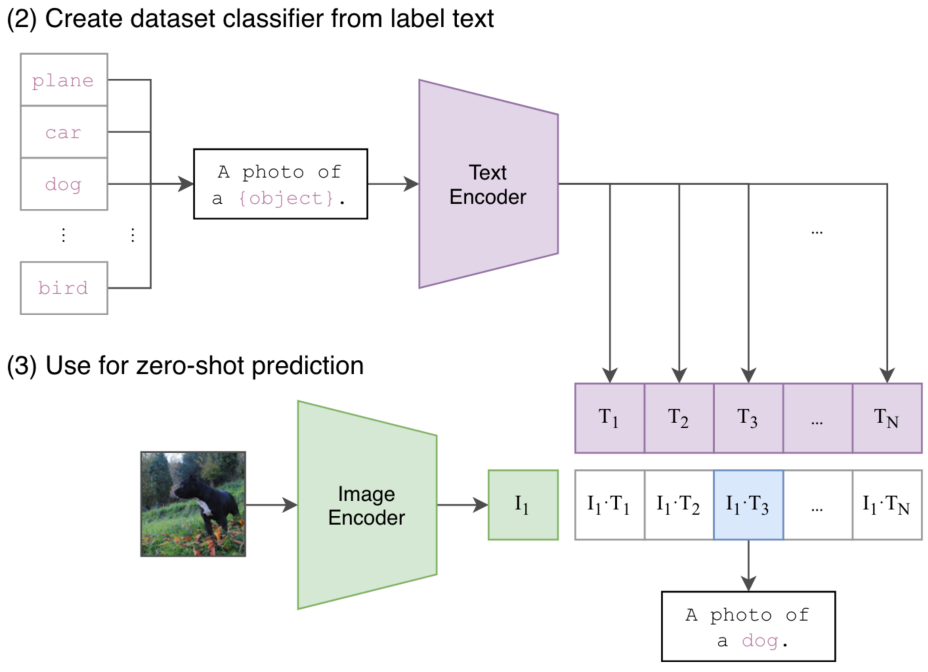
where $\mathbf{i}$ and $\mathbf{t}$ indicate the image and text feature vectors, respectively.

CLIP

CLIP [56], abbreviation for Contrastive Language-Image Pre-training, is the first of the large-scale contrastive cross-modal models to be published. In theory, CLIP is capable of evaluating the semantic similarity between the image and its caption. CLIP computes the dot product of image features and text features. The diagonal elements of the resulting matrix represent positive samples, while the off-diagonal elements correspond to negative samples. In CLIP, negative samples are no longer limited to a single pair, but rather comprise all samples in the entire mini-batch, except for the positive samples. That is, the anchor and other samples in the mini-batch serve as negative samples. By employing this large-scale contrastive learning approach, CLIP has achieved state-of-the-art results in various tasks. Moreover, this approach can be applied to zero-shot learning. Figure 3.4 illustrates the effectiveness of this approach.



(a) Image and text features product. The diagonal elements are positive, otherwise.



(b) Illustration for zero-shot image caption task.

Figure 3.4 – The visualization CLIP pre-training and zero-shoot prediction [56].

3.2.2 Attention Mechanism

In the previous chapter, we introduced several state-of-the-art models, such as ViT [36] and Swin Transformer [37] in the computer vision chapter, Transformer [34] and BERT [47] in the natural language processing subsection 2.2.2, and CLIP [56] in the previous subsection 3.2.1, which all share a common feature, *i.e.*, the self-attention mechanism. Although the mainstream attention mechanism originates from the NLP domain, it has been increasingly used in cross-modal and visual models. The essence of self-attention mechanism is to compute the relationship between each element in the input vector. This is a significant departure from traditional saliency in computer vision, which is a pixel-wise gradient difference detection. Different from the text input, which takes a token (*e.g.*, a word) as the basic unit, an image has tens of thousands of relatively independent pixels. How to choose the number of image feature vector dimensions to align with the text feature vector, (*i.e.*, to bridge the multimodality gap), is an important issue for multimodal models. Figure 3.5 shows the types of the attention model.

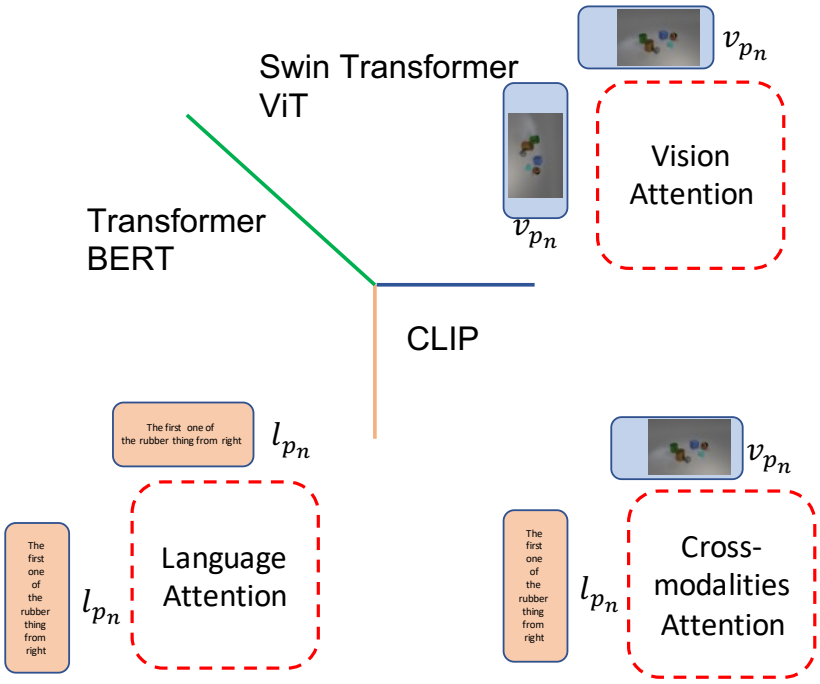


Figure 3.5 – Attention in CV, in NLP and cross-modal. v_{p_n} indicates visual feature vector, l_{p_n} indicates the linguistic feature vector.

In the section 7.2.4, we will further analyze the attention mechanism based on specific tasks. This mechanism has been shown to outperform traditional models like RNNs and

CNNs on several tasks, and has emerged as one of the most promising models in recent years. However, further comprehensive testing is required to fully evaluate the potential of self-attention models.

Conclusion of Part I

In this Part I, we sort out the bottlenecks and breakthroughs in the evolution of computer vision, natural language processing, and multimodality learning, respectively. An overview of the evolution of machine learning techniques in multimodal research, starting from early manual feature extraction methods, progressing through the emergence of automated perceptrons, and culminating in the dominance of deep learning models like CNNs and RNNs. Furthermore, we also highlight the recent emergence of Transformer-based models and their impact on the field, with attention models increasingly replacing CNNs and RNNs in many tasks. The field of machine learning has been the driving force behind technological advancements in various fields, including multimodality. Despite the complexity of multimodal tasks, they are ultimately based on machine learning algorithms. Thus, a solid understanding of machine learning fundamentals can enhance our comprehension of multimodality. In Figure 7, the fundamentals of machine learning are located on the left side of the human-machine relationship map, which is closer to the machine side.

In subsequent chapters, we will go deeper into the specifics of two different kind of multimodal tasks.

Part II

Image-text Retrieval

Introduction of Part II

In the previous part I, we have introduced the framework of multimodal algorithm and the state-of-the-art of multimodal tasks. In this part, we will focus on one specific multimodal task: cross-modal retrieval, particularly in image-text retrieval. According to the multimodal framework in Figure 7, image-text retrieval falls to the right side, closer to the human perception. The main objective of this task is to find high-level semantic information in the latent space that captures the multimodal features. The retrieval task is a fundamental task in both the computer vision (CV) and natural language processing (NLP) domains. From web search to search engines, from product matching to recommendation systems, retrieval algorithms are the core algorithms involved.

Earlier retrieval algorithms used word or character labels to compare data, but this approach was limited by the lack of contextual information. Image search algorithms, such as CBIR (content-based image retrieval), improved on this by allowing image-based queries and using keypoints matching. However, CBIR still requires a similar photo to the query information.

Multimodal retrieval addresses this limitation by enabling retrieval in different modalities, such as images or text. It transforms the input into high-level semantic information, which provides a more robust and accurate way to find relevant data. Unlike tag matching, multimodal retrieval focuses on semantic description of the query information, which can apply to both images and text. It allows for multiple statement descriptions of the same image and multiple related images for the same description.

Figure 8 shows an example of cross-modal image-text retrieval. If query by image, the ranked relevant texts will be given, and vice versa.

In chapter 4, we review the state-of-art of image-text retrieval, category the cross-modal retrieval model by structure and compare the different categories model. In chapter 5, we propose a new loss function to enhance the multimodalities features representations in the latent space and further improve the performance on the public image-text database.

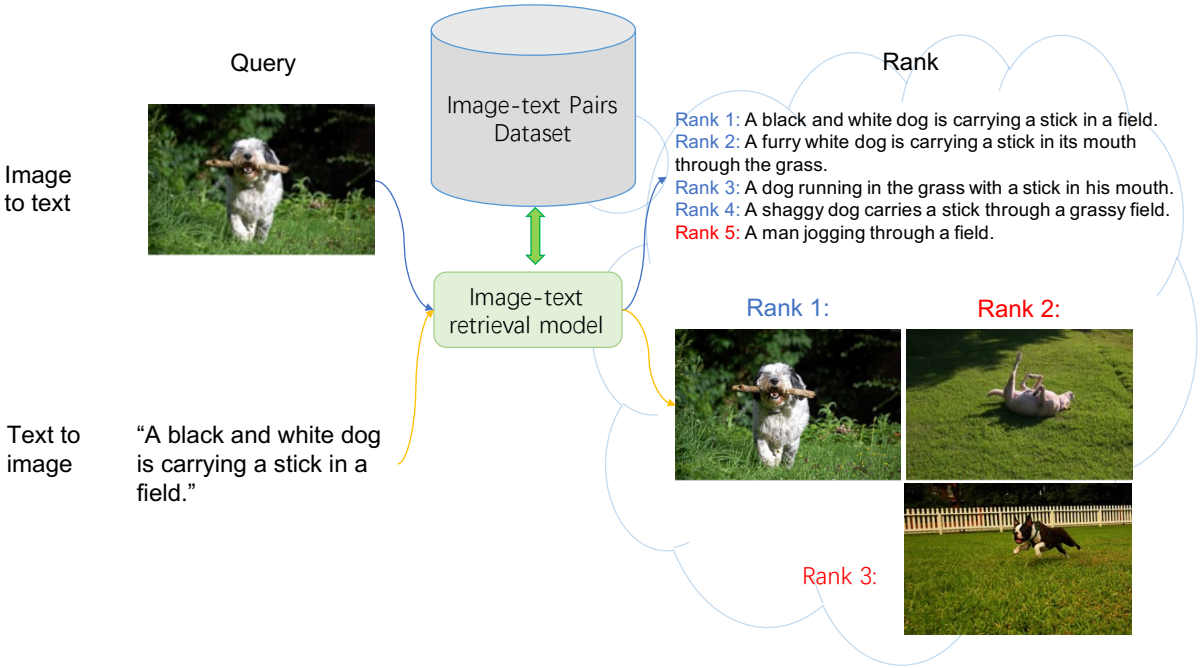


Figure 8 – Example of image-text retrieval task. Image-to-text is query by image. First row shows five sentences feedback, but the rank 5 is not correct. While text-to-image is query by text, the relevant images are displayed. In this example, although only rank 1 image is ground truth, rank 2 and rank 3 are closely relevant to the description of the query text.

Chapter 4

Image-text Retrieval Models Classification

4.1 Introduction

Over the last decade, cross-modal retrieval has made significant progress. The goal of cross-modal retrieval is to retrieve relevant information across heterogeneous modalities. It is widely used in many fields, such as visual questioning and answering [57], image or video caption [58], [59], phrase localization [60], knowledge transfer [61] and text-to-image generation [62]–[64]. Benefiting from Content-Based Image Retrieval (CBIR) and Natural Language Processing (NLP) techniques, the computer could almost bridge the semantic gap between high-level human perception and low-level features in single mode. As deep learning achieves remarkable results in both vision and language domain, researchers begin to explore the semantic gap between image and text. In bidirectional image-text cross-modal retrieval, taking image as the query to retrieve relevant information in text data is called image-to-text retrieval, and vice versa.

The first cross-modal retrieval review is written by Liu et al. [65], which focuses on summarizing traditional methods. The overviews of multimedia information retrieval in the papers [66], [67] are not only for image-text but also for video and audio modalities. The most recent cross-modal overview [68] focuses on music and sound data retrieval. In this chapter, we focus on cross-modal retrieval methods based on deep-learning, only for image-text context, and proposed in the last two years as some new methods based on deep learning have been proposed, which significantly improve the performance. We give an analysis of this relatively narrow topic in the image-text cross-modal retrieval domain and propose to classify these algorithms into four categories according to their embedding methods:

1. pairwise learning embedding methods;

2. adversarial learning methods;
3. interaction learning methods;
4. attributes learning methods.

The rest of the chapter is organized as follows: we classify the most recent image-text cross-modal retrieval algorithms by their embedding methods and highlight their pros and cons in section 4.2; we show the performance comparison results of the representative algorithms in each category using two most popular datasets in this domain (Flickr30K and MSCOCO) in section 4.3; then the chapter concludes with the recent image-text retrieval works and gives some perspectives in section 4.4.

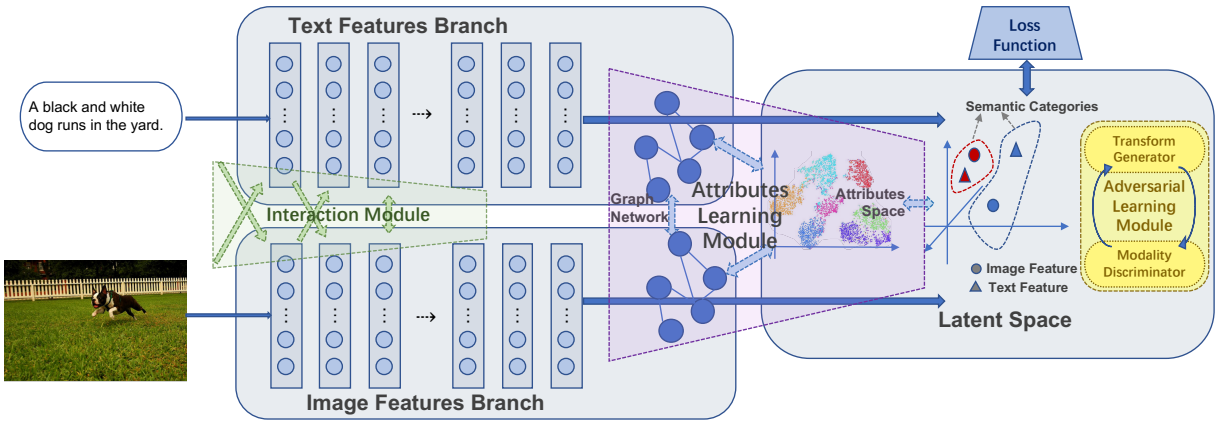


Figure 4.1 – Generic architecture of Image-Text cross-modal retrieval methods. The blue regions are the basic three-parts structure and denotes the **pairwise learning** methods. With an additional dotted green region, it is the structure of the **interaction learning** methods. With the dotted yellow square in "Latent Space", it indicates the **adversarial learning** methods. With an additional dotted violet region, it becomes the structure of the **attributes learning** methods.

4.2 Image-Text Retrieval Methods Classification

In general, the cross-modal retrieval architecture be divided into three parts: "Image Features Branch", "Text Features Branch" and "Latent Space", as shown in Figure 4.1. The first two branches extract image features and text features, separately. Recurrent Neural Network (RNN [26]) or Long Short-Term Memory (LSTM [43]) is used to extract stylistic features from texts, while Convolutional Neural Network (CNN [27]) is used to extract image features. Then the "Latent Space" part projects the features corresponding

to different modalities to one common space, and measure the similarity between the projected text features and the projected image features.

Here, the methods following the above architecture are classified as the *pairwise learning*, (cf. Figure 4.2). If adversarial machine learning methods are adopted in the "Latent Space" part, we classify these methods into the *adversarial learning*, (cf. yellow region in Figure 4.3). If there are some interaction flows between the "Image Features Branch" and the "Text Features Branch" in addition to the general architecture, we classify these methods into the *interaction learning*, (cf. green region in Figure4.4). If high-level semantic attributes are exploited, instead of the direct use of the basic image features and text features, we classify these methods into the *attributes learning*,(cf. violet region in Fig. 4.5). In the following, we detail these four types of methods.

4.2.1 Pairwise Learning Methods

Pairwise learning methods attempt to find a cross-modal loss function that can calculate the distance between corresponding feature pairs directly in a common space. By learning this loss function, the distance between associated images and texts reduces, and the distance between independent samples increases. There are some different forms of pairwise learning, but all of them represent two different features in the same common space directly, (see Figure 4.2). Pairwise learning methods differ in the factors of the loss function, such as corresponding label relation, feature space, similarity measure evaluation.

Zhang and Lu introduce a new matching loss called Cross-Modal Projection Matching (CMPM [69]). The idea behind this is to increase the correlation of matching pairs and to reduce it for unmatching pairs by minimizing the Kullback-Leibler divergence between the probability of matching image features to text features and the normalized matching probability. All the positive and negative samples are thus considered in the CMPM. The disadvantage may stem from the absence of inherent word associations within the text, as bidirectional LSTM primarily integrates word sequence information without adequate semantic contextual logic. After CMPM, Deep Pairwise Ranking model with multi-label information for Cross-Modal retrieval (DPRCM [70]) is proposed, which employs a bi-triplet loss to reduce the distance between positive samples and increase the distance between negative independent samples. DPRCM also combines cross-entropy loss with bi-triplet loss in their retrieval network so that multi-label information can be learned in common space under supervision. DPRCM extracts image features and text features

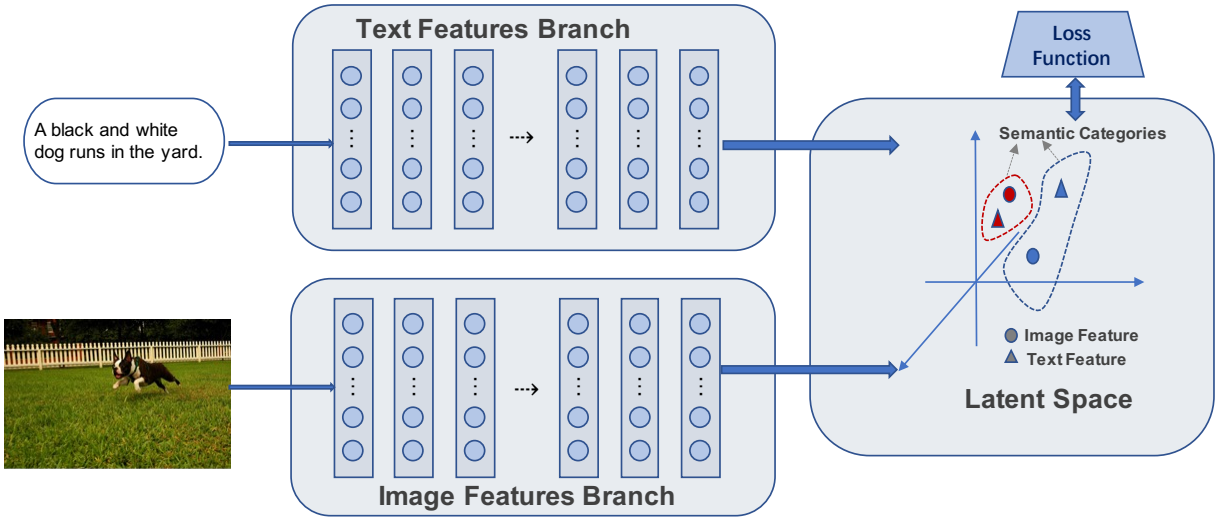


Figure 4.2 – Pairwise learning structure.

only by two-layer neural networks separately, which is a more straightforward way than other cross-modal retrieval methods. Unlike DPRCM, Deep Supervised Cross-Modal Retrieval (DSCMR [71]) uses fully connected layers to build common representation space. A linear classifier is used to predict the category of each sample in the common representation space. Simultaneously, another discrimination loss is minimized in label space. Both DPRCM and DSCMR belong to supervised learning methods. They use label information to enhance the learning progress when they deal with multi-modal pairs. Finally, Liu et al. propose neighbor-aware network (NAN [72]), which calculates the neighbor-aware ranking loss in common semantic space under the influence of the intra-attention module. The neighbor-aware ranking loss can be divide into inter-modal and intra-modal parts. The inter-modal neighbor-aware ranking loss emphasizes semantic relations within a single modality, while the intra-modal neighbor-aware ranking loss focuses on semantic relations between different heterogeneous modalities. NAN adds an attention module to re-weight feature map since different semantics are distinguished in intra-modal and inter-modal neighbor-aware networks. As attention features map could be associated with the semantic relation during the neighbor-aware ranking loss learning, the intra-attention module plays an important role in image-text matching.

Unlike CMPM, DPRCM and DSCMR rely more heavily upon label distance information. There are some other pairwise loss functions belonging to supervised learning, such as kNN-margin loss [73], triplet loss [55]. The key point of pairwise learning is to design an efficient loss function that could reduce features distance of the same semantic category

in common space.

4.2.2 Adversarial Learning Methods

Adversarial learning methods are enlightened by Generative Adversarial Nets (GAN [74]). Wang et al. [75] introduced adversarial learning firstly into cross-modal retrieval domain. In latent space, a two-player minimax value game has been played between a discriminator and a generator in adversarial network learning. The expectation value $V_{D,G}$ is defined as:

$$V_{D,G} = \mathop{\mathbb{E}}_{I_i \sim I} [\log D((I_i))] + \mathop{\mathbb{E}}_{T_i \sim T} [\log (1 - D(G(T_i)))] \tag{4.1}$$

where I and T indicate the image and text modalities. Adversarial learning method uses minimax game played between generator and discriminator to bridge image and text features. Figure 4.3 shows how the adversarial function works on the cross-modal retrieval task.

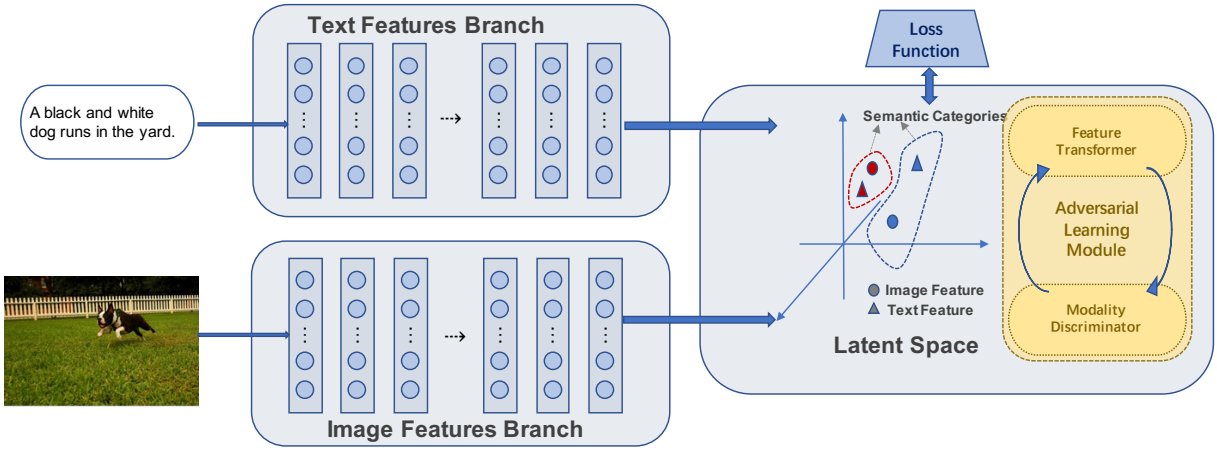


Figure 4.3 – Adversarial learning structure. The yellow area represents the module on which the adversarial function acts.

After that, Sarafianos et al. propose Text-Image Modality Adversarial Matching (*i.e.*, TIMAM [76]), which adopts an Adversarial Representation Learning (ARL) framework to learn modality-invariant representations for more effective image-text matching. In the ARL framework, a two-layer fully-connected network adversarial discriminator is optimized in the common space. The better discriminator pain, the better cross-modal retrieval gain. TIMAM also adds Bidirectional Encoder Representations from Transformers (BERT [47]) in front of LSTM branch to optimize text features. At the same time, Liu et

al. propose a new deep adversarial graph attention convolution network (A-GANet [77]). A-GANet extracts image features not only from the CNN branch but also from a graph attention convolution layer based on a visual scene graph. The visual scene graph carries information about object regions and relationships according to human visual perception characteristics. High-level structured semantic visual features are learned from this designed graph attention convolution layers. Particular joint embedding layers connect the image and text features through the adversarial learning module. Furthermore, Wang et al. [78] and Zhu et al. [79] use adversarial learning in food images and recipes matching.

Adversarial learning methods have not been around for a long time in the field of cross-modal retrieval. It has also been used in other areas such as image synthesis and style transfer that require more inference.

4.2.3 Interaction Learning Methods

In this section, we define interaction learning methods as those having a large amount of information transfer between the two branches before the image and text features enter the common space. Figure 4.4 shows the interaction learning architecture.

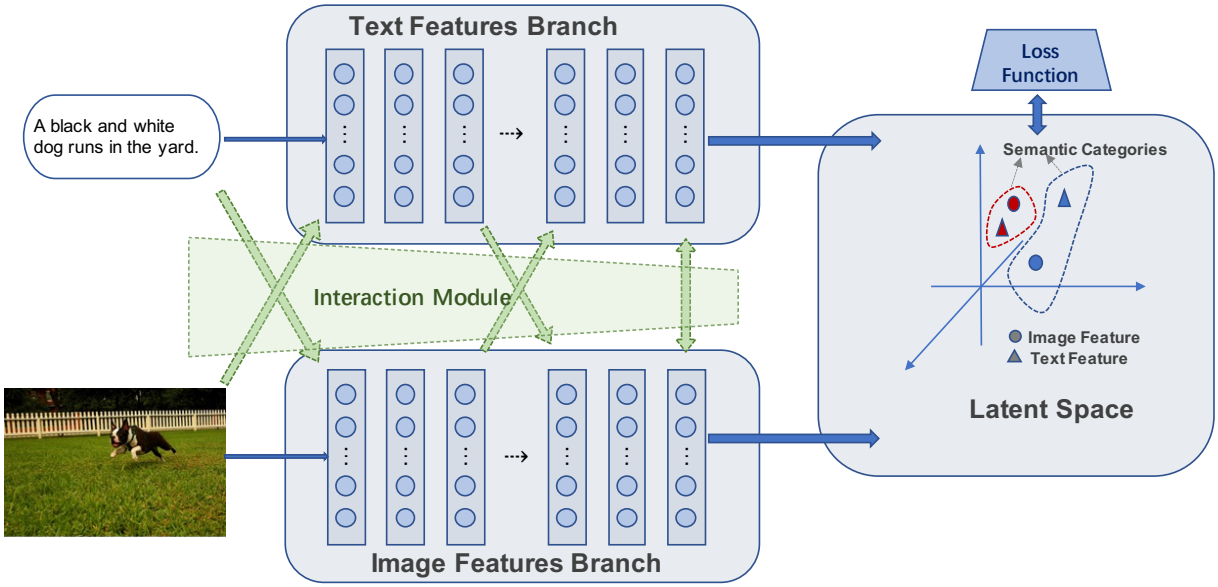


Figure 4.4 – Interaction learning structure. The green area represents the module where the interaction pathways interact on.

Lou et al. propose a Multitask learning approach for Cross-modal Image-Text Retrieval (MCITR [80]) to take into account the common features extracted from image-

text cross-modal data. MCITR employs relation enhanced correspondence cross-modal autoencoder [81] to correlate the hidden representations, before text and image are projected into an embedding space. Simultaneously, Cross-Modal Adaptive Message Passing (CAMP [78]) adopts a cross-modal message-passing aggregation at the beginning of the network. CAMP explores the interactions between images and text before calculation in common space. Other methods add attention module to transfer the information between image and text branch, such as Wang et al. [82] and Wu et al.[83].

Due to the information transfer between image and text branches in the initial and low-level processing, more corresponding connections could be represented in latent space. Nevertheless, this kind of algorithms is more complicated, and the amount of calculation increases exponentially.

4.2.4 Attributes Learning Methods

In deep learning, a vast number of parameters are trained by large-scale calculations to obtain excellent results, which means that a massive amount of data is needed for training deep neural networks. However, human beings can learn the properties of things from a few examples. Attributes learning imitates human thought processes and learns the characteristics of objects. The essence of attributes learning is "learning to learn". There are also some attempts to apply attributes learning in cross-modal retrieval. The architecture of typical attributes learning is shown in Figure 4.5.

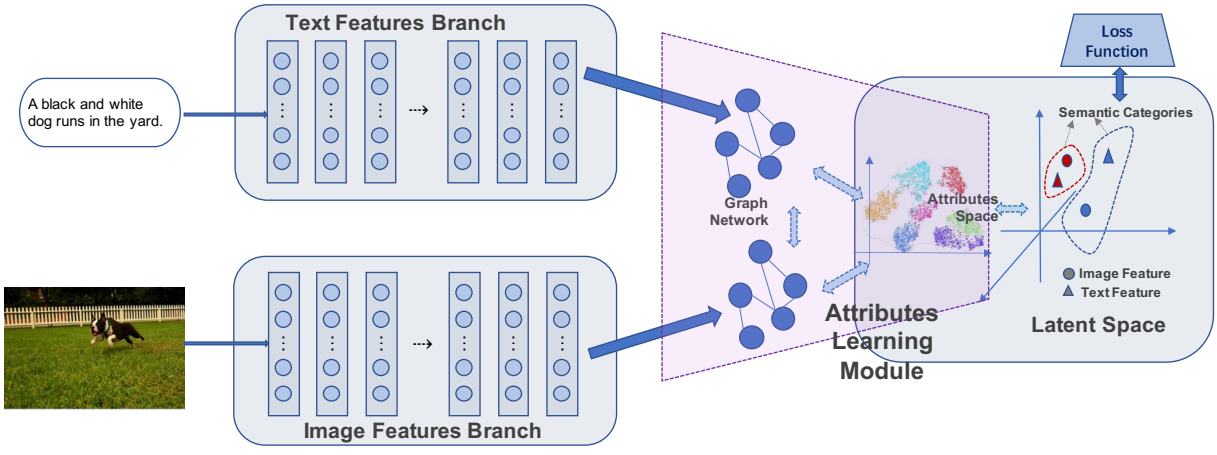


Figure 4.5 – attributes learning structure. The violet area represents the module which contains attributes learning components, such as graph network.

Ji et al. propose an Attribute-Guided Network (AgNet [84]) for cross-modal retrieval,

which combines with zero-shot learning and hashing retrieval. Objective functions are designed to transform image and text feature vectors into object attribute vectors in attributes space. Then a three-layer neural network transforms attribute vectors to hash codes. Without the supervision information, instances cluster themselves in attribute space. Hamming distance is selected to calculate the similarity between different modalities. Although hash coding is an efficient representation, we cannot determine whether there is a linear relationship between the hash code length and the number of attributes. From then on, Huang and Wang propose Aligned Cross-Modal Memory (ACMM [85]) for few-shot image and sentence matching. ACMM includes two key steps: aligned memory controller network and memory read and update. A cross-modal graph convolutional network based on aligned memory controller network aims to generate modality-specific interface vectors to connect with shared memory item. Memory read and update module aims to score the similarity between sentence and image semantically and to update the memory when few-shot content learning is used.

Attributes learning can learn the characteristics of objects from a few examples. Comparing with direct pairwise learning by the deep neural network, the attributes learning owns stronger cross-modal correlation features extraction. Thus attributes learning methods show the highest performances in Table 4.1 and Table 4.2.

4.3 Databases and Evaluation

There are many established databases in the cross-modal retrieval field, especially for image-text retrieval tasks. For example, CUHK-PEDES dataset [91] focuses on pedestrians on the road; Wikipedia dataset [92] has more text information which could mine NLP capabilities; Recipes1M dataset [93], [94] owns large-scale food images and recipes, etc. Since various image-text retrieval methods have reported their performances on the two most common databases Flickr30K and MSCOCO, we sum them up here for comparison.

4.3.1 Databases

Flickr30K [95] is a standard dataset for image-text retrieval, containing more than 31K images and 155K sentences in total, each image has five corresponding sentences. Flickr30K has 44,518 categories in total. It is usually split into 29K images for training, 1K images for validation and 1K images for test. The performance are shown on Table

Method		Flickr30K						
		Text-to-Image			Image-to-Text			mR
		R@1	R@5	R@10	R@1	R@5	R@10	
Pairwise Learning	kNN-margin [73]	36.0	64.4	72.5	26.7	54.3	65.7	53.3
	CMPM [69]	48.3	75.6	84.5	35.7	63.6	74.1	63.6
	CMPM+CMPC [69]	49.6	76.8	86.1	37.3	65.7	75.5	65.2
	VSE++ [55]	52.9	80.5	87.2	39.6	70.1	79.5	68.3
	NAN [72]	55.1	80.3	89.6	39.4	68.8	79.9	68.9
Adversarial Learning	A-GANet [77]	-	-	-	39.5	69.9	80.9	-
	TIMAM [76]	53.1	78.8	87.6	42.6	71.6	81.9	69.3
Interaction Learning	MFM* [86]	50.2	78.1	86.7	38.2	70.1	80.2	67.3
	SCAN* [87]	67.4	90.3	**95.8**	48.6	77.7	85.2	77.5
	MTFN-RR [88]	65.3	88.3	93.3	**52.0**	**80.1**	86.1	77.5
	BFAN* [89]	68.1	91.3	-	50.8	78.4	-	-
	CAMP [78]	68.1	89.7	95.2	51.5	77.1	85.3	77.8
	PFAN* [82]	**70.0**	**91.8**	95.0	50.4	78.7	86.1	78.7
	SAEM [83]	69.1	91.0	95.1	**52.4**	**81.1**	**88.1**	**79.5**
Attributes Learning	GVSE* [90]	68.5	90.9	95.5	50.6	**79.8**	**87.6**	78.8
	ACMM [85]	**80.0**	**95.5**	**98.2**	50.2	76.8	84.7	**80.9**
	ACMM* [85]	**85.2**	**96.7**	**98.4**	**53.8**	**79.8**	**86.8**	**83.5**

Table 4.1 – The performance of state-of-the-art methods on Flickr30K. Red, green, and blue represent the best, second, and third performance respectively. * indicates ensemble methods.

4.1.

MSCOCO [96] contains 123,287 images and each one is described by five sentences. It has 91 objects categories. Generally experiments use 5K images for validation, 1K or 5K images for test. Table 4.2 shows performances on 5 folds of 1K test images as the most commonly setting.

4.3.2 Evaluation

We collect the state-of-the-art approaches *Recall@K* results shown on papers, which measures the number of correct items found among the top K retrieval results. For convenience, we also give a general evaluation indicator mR, which means the mean of *Recall@K*. For all the algorithms, we show the best performance in the database. However, these performances maybe got from module ensemble method, we use * indicates that in Table 4.1 and Table 4.2. The best results of all retrieval methods are in red, second in green, and third in blue.

Method		MSCOCO (1K test images)						
		Text-to-Image			Image-to-Text			mR
		R@1	R@5	R@10	R@1	R@5	R@10	
Pairwise Learning	kNN-margin [73]	65.4	91.9	97.1	49.6	82.7	91.2	79.7
	CMPM [69]	56.1	86.3	92.9	44.6	78.7	89.0	74.6
	CMPM+CMPC [69]	52.9	83.8	92.1	41.3	74.6	85.9	71.8
	VSE++ [55]	64.6	90.0	95.7	52.0	84.3	92.0	79.7
	NAN [72]	61.3	87.9	95.4	47.0	80.8	90.1	77.1
Adversarial Learning	A-GANet [77]	-	-	-	-	-	-	-
	TIMAM [76]	-	-	-	-	-	-	-
Interaction Learning	MFM* [86]	58.9	86.3	92.4	47.7	80.1	90.9	76.2
	SCAN* [87]	72.7	94.8	98.4	58.8	88.4	94.8	84.7
	MTFN-RR [88]	74.3	94.9	97.9	60.1	**89.1**	**95.0**	85.2
	BFAN* [89]	74.9	95.2	-	59.4	88.4	-	-
	CAMP [78]	72.3	94.8	98.3	58.5	87.9	95.0	84.5
	PFAN* [82]	**76.5**	**96.3**	**99.0**	**61.6**	**89.6**	**95.2**	**86.4**
	SAEM [83]	71.2	94.1	97.7	57.8	88.6	94.9	84.1
Attributes Learning	GVSE* [90]	72.2	94.1	98.1	**60.5**	**89.4**	**95.8**	85.0
	ACMM [85]	**81.9**	**98.0**	**99.3**	58.2	87.3	93.9	**86.4**
	ACMM* [85]	**84.1**	**97.8**	**99.4**	**60.7**	88.7	94.9	**87.6**

Table 4.2 – The performance of state-of-the-art methods on MSCOCO (1K). Red, green, and blue represent the best, second, and third performance respectively. * indicates ensemble methods.

4.3.3 Discussion

The comparison of the results from these two databases can only show part of the performances of the algorithms. Comparing the results, we can see that some algorithms have better results in retrieving text from images, and others are better in retrieving images from text. If a method gets better results in image-to-text retrieval direction than text-to-image direction, it means that the image feature representations in latent space are more precise, and vice versa. But no algorithm can achieve the best results in both directions currently. In other words, no algorithm gets the best balance point between two directions. Moreover, the same method performs differently on different databases, which may be caused by the number of object categories in the data. As we know, the number of object categories in Flickr30k is hundred times that of MSCOCO. More categories means higher learning costs, which is a challenge for retrieval algorithms. Some categories have a large sample size, but some have a small sample size. A large number of samples may cause overfitting, while a small number is not enough to train appropriate network parameters. For this reason, attention models and attributes learning methods are used to reduce the impact of small number of samples on retrieval results. Therefore, the interaction learning methods and attributes learning methods achieve as high performance on the Flick30K as on the MSCOCO database.

4.4 Conclusion

Deep learning based methods in image-text cross-modal retrieval have achieved significant progress in recent years. Pairwise learning proposes two-branch architecture; adversarial learning and interaction learning methods are based on it; attributes learning may become a popular trend for cross-modal retrieval tasks due to the usage of relations between attributes and semantics. In the future, more image-text free pairs databases for cross-modal retrieval should be explored. More evaluation metrics should be used to compare different methods.

This chapter provides an overview of text-image pair matching for cross-modal retrieval. We categorize existing text-image retrieval algorithms into four types based on the structural design of the network. We summarize each type's advantages and disadvantages of the included methods. Furthermore, we gather and organize the performance results of these methods on two popular databases and classify them according to our proposed structure classification. In the final part, we discuss the features and limitations

of text-image retrieval as a type of cross-modal task and outline potential directions for future research.

This part work have published in:

— J. Chen, L. Zhang, C. Bai, and K. Kpalma, « Review of recent deep learningbased methods for image-text retrieval », in 2020 IEEE Conference on MultimediaInformation Processing and Retrieval (MIPR), 2020, pp. 167-172. doi:doi: 10.1109/MIPR49039.2020.00042. (Cf. [97])

Chapter 5

IMC Loss: A Proposed Loss Function for Image-Text Retrieval

5.1 Introduction

In recent years, cross-modal retrieval has attracted a lot of attention in both computer vision and natural language processing domains. Image-text retrieval is a task that aims to find the most relative semantic pairs in heterogeneous modalities. It can generally be seen as two-direction queries: one is the query by image feature to retrieve the text information by the relevant rank and vice versa. With advances in deep neural network technology, the bottleneck in image-text retrieval has shifted from feature extraction from different modalities to embedding representational loss function learning.

Many loss functions have been proposed in the text-image retrieval domain. Most recent approaches use a hinge-based triplet ranking loss [52], [55], also referred to as Sum of Hinges (SH) loss, to reduce the retrieval distance in both directions. Faghri et al. further proposed a Max of Hinges (MH) loss based on the SH loss, by emphasizing hard negatives for training, which achieves better performances than the SH loss [55]. The existing loss functions deal well with heterogeneous modalities pairs. However, few losses consider the effect of homogeneous modality pair distances. To address this issue, we propose a novel Intra-Modal Constraint (IMC) loss to reduce the violation between negative pairs within the same modality.

In our previous survey work [97], the deep learning based image-text retrieval architectures are divided into four categories: 1) "pairwise embeddings learning" which uses one branch to generate the image features and another one to generate the text image features, *e.g.*, papers [55], [69]; 2) "adversarial learning" which introduces the Generative Adversarial Nets (GAN [74]) in the latent space; 3) "interaction learning" which applies information transfer between the image and text branch; 4) "attribute learning" which involves high-level semantic intrinsic attributes through attention mechanisms or graph

neural networks. Because of the conventionality and simplicity of the "pairwise embeddings learning", we choose to validate our proposed IMC loss on this type of architecture.

Experimental results on two commonly used image-text retrieval datasets show that a typical "pairwise embeddings learning" architecture combined with our IMC loss can achieve higher performance, compared to state-of-the-art methods. The ablation study also shows the improvements brought by the IMC loss, compared to the MH loss.

Our main contributions are as follows:

1. We propose an Intra-Modal Constraint loss to reduce the violation of negative pairs in the homogeneous modality.
2. We develop a "pairwise embeddings learning" network combined with our loss function for image-text retrieval.
3. We test our methods on the two most popular image-text retrieval datasets and evaluate the influence of different similarity distances in the IMC loss.

5.2 The Proposed Method

In this section, we first present the architecture in section 5.2.1. Then we introduce our designed loss function for image-text retrieval in section 5.2.2. Experiments in section 5.3 result the proposal and section 5.4 concludes the chapter.

5.2.1 Architecture

We use a two-branch structure network to extract the image and text features, then project heterogeneous modalities features into a common embedding space and learn the representation by intra-modal constraint loss, as shown in Figure 5.1, *i.e.*, a typical "pairwise embeddings learning" architecture. Here, the image encoder is the pre-trained ResNet152 [32], the same as most previous works. Then ResNet152 is followed by an additional fully connected (FC) layer to get the image feature vector i_n (where n is the image index) with the same dimension d ($d = 1024$) as the text feature vector t_n. In the text encoder, we firstly get word representation via the pre-trained GloVe [98]; then employ a Bi-direction Long Short-Term Memory (Bi-LSTM [43]) to obtain the final text feature vector t_n. Finally, the image feature vector i_n and the text feature vector t_n are embedded into a common space.

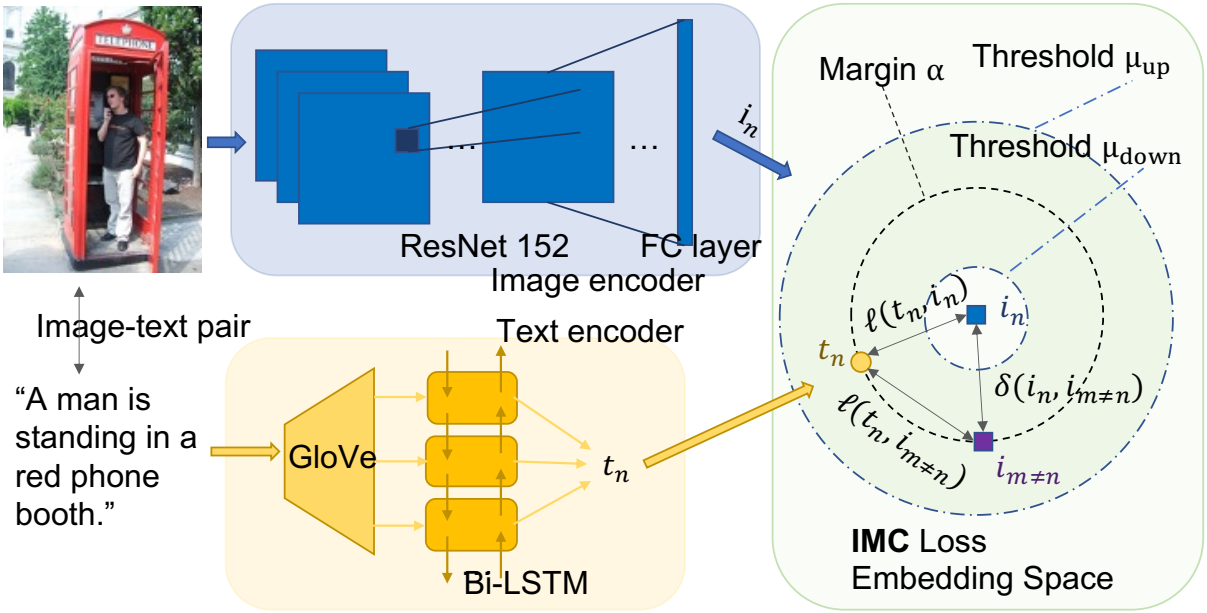


Figure 5.1 – The framework of Intra-Modal Constraint network. Two-branch network encoders extract the image and text features on the left side. The embedding space for features projection is on the right side: the intra-modal constraint loss reduces the distances of inter-modal pairwise feature representations (*i.e.*, yellow circle and blue square) and increases the intra-modal non-pair distance (*i.e.*, the blue square and purple square) simultaneously. More details will be explained in section 5.2.2.

Hereafter, we refer to (i_n, t_n) as a positive image-text features pair coming from the n-th image and its relevant text; and refer to $(i_n, t_{m\neq n})$ as a negative pair from the n-th image and the m-th text which is non-relevant to the n-th image.

To train the FC layer in the image encoder and the Bi-LSTM, we need to design a loss function, the minimization of which should lead to the minimization of the distance between the positive pairs and the maximization of the distance between the negative pairs in the embedding space at the same time. In the following section, we'll firstly recall two state-of-the-art loss functions (SH loss and MH loss), and then introduce our proposed Intra-Modal Constraint (IMC) loss.

5.2.2 Intra-Modal Constraint Loss

Suppose that we have N images in the training set, the total training image and text feature vectors can then be denoted as $I = \{i_n\}_{n=1}^{N}$ and $T = \{t_n\}_{n=1}^{N}$, respectively. The most commonly used SH loss [52], [72] aims to minimize the cumulative loss over training data:

$$\begin{aligned} Ł_{\texttt{SH}}(I,T) = &\sum_{n,m\in N} [\alpha - \ell(i_n, t_n) + \ell(i_n, t_{m\neq n})]_+ + \\ &\sum_{n,m\in N} [\alpha - \ell(t_n, i_n) + \ell(t_n, i_{m\neq n})]_+, \end{aligned} \tag{5.1}$$

where α is a margin parameter, $[x]_+ \equiv max(x, 0)$ and $\ell(x, y)$ is some distance function between two vectors x and y.

The SH loss counts the distance of every negative pair with a larger distance than that of the positive pair in the margin. Faghri et al. [55] showed that in case multiple negatives with small violations combine to dominate this loss, local minima may be created in the SH loss. Thus they proposed the MH loss, which solves this problem by focusing on the hardest negative:

$$\begin{aligned} Ł_{\texttt{MH}}(I,T) = &\max_{n,m\in N} [\alpha + \ell(i_n, t_{m\neq n}) - \ell(i_n, t_n)]_+ + \\ &\max_{n,m\in N} [\alpha + \ell(t_n, i_{m\neq n}) - \ell(t_n, i_n)]_+. \end{aligned} \tag{5.2}$$

They also demonstrated empirically that the MH loss performs better than the SH Loss [55].

However, both SH and MH losses focus on heterogeneous modalities pairs but ignore the negative pairs in the homogeneous modality. To retrieve more relevant pairs of image

and text, we propose a novel loss function for reducing the intra-modal pairwise effect, named Intra-Modal Constraint (IMC) loss. Our IMC loss combines the MH loss with two Intra-Modal Constraint (IMC) terms:

$$\begin{aligned} Ł_{\texttt{IMC}}(I,T) = & \max_{n,m\in N}[\alpha + \ell(i_n, t_{m\neq n}) - \ell(i_n, t_n)]_+ + \\ & \max_{n,m\in N}[\alpha + \ell(t_n, i_{m\neq n}) - \ell(t_n, i_n)]_+ + \\ & \texttt{IMC}(I) + \texttt{IMC}(T), \end{aligned} \tag{5.3}$$

where $\texttt{IMC}(I)$ and $\texttt{IMC}(T)$ are the image-modal constraint and text-modal constraint, respectively. They are defined in the same way as follows:

$$\texttt{IMC}(V) = \lambda \sum_{n,m\in N} \begin{cases} 0, & \delta(v_n, v_m) \leq \mu_{\text{down}} \\ \delta(v_n, v_m), & (m \neq n) \\ 0, & \mu_{\text{up}} \leq \delta(t_n, v_m) \end{cases} \tag{5.4}$$

where λ is a weight parameter to balance the influence of IMC terms, $\delta(x, y)$ is a similarity function between vectors x and y, and μ_{up} and μ_{down} are two thresholds defining boundaries.

On the right side of Figure 5.1, we illustrate the principle of the IMC loss, which considers both positive/negative and inter-/intra-modal pairs. Note that we only calculate the similarity distance within the boundaries.

In general, the similarity function δ can be the same as ℓ which is normally a cosine distance function in the literature:

$$\delta_{\cos}(v_n, v_m) = 1 - \cos(v_n, v_m) = 1 - \frac{\sum_{k=1}^{d} v_{nk} \cdot v_{mk}}{\sqrt{\sum {v_{nk}}^2}\sqrt{\sum {v_{mk}}^2}}, \quad (m \neq n) \tag{5.5}$$

where $(v_n, v_{m\neq n})$ are negative pairs in the same modality, d is the vector dimension. The similarity function δ can, of course, use other similarity metrics, *e.g.*, Mean Squared Displacement (MSD):

$$\delta_{\text{msd}}(v_n, v_m) = \text{msd}(v_n, v_m) = \sum_{k=1}^{d} (v_{nk} - v_{mk})^2, \quad (m \neq n) \tag{5.6}$$

Manhattan distance (L1):

$$\delta_{l1}(v_n, v_m) = L1(v_n, v_m) = \sum_{k=1}^{d} |v_{nk} - v_{mk}|, \quad (m \neq n) \tag{5.7}$$

or Euclidean distance (L2):

$$\delta_{l2}(v_n, v_m) = L2(v_n, v_m) = \sum_{k=1}^{d} \sqrt{(v_{nk} - v_{mk})^2}, \quad (m \neq n). \tag{5.8}$$

The influence of these different similarity functions on the final results will be shown in section 5.3.4.

5.3 Experiments

In this section, we evaluate our approach on two popular public datasets, discuss the results, and analyze the contributions of the IMC loss compared to the MH loss.

5.3.1 Datasets

We use the same database as section 4.3.1, MSCOCO and Flickr30K.

MSCOCO [96] consists of 128K images and each one is described by five sentences. MSCOCO is split into 82,783 training images, 5000 validation images and 5000 test images [52]. We also use the rest of 30,504 images in original validation set of MSCOCO as training images which gives totally 113,287 images in our training set following the previous work [55]. We report the results both on 5K test images and the average over 5 folds of 1K test images.

Flickr30K [95] is a standard dataset for image-text retrieval, including 30,000 images. It is split into 29K training images, 1K validation images and 1K test images [52].

5.3.2 Settings and performance metrics

Implementation details: The model is implemented in PyTorch with a NVIDIA 2080Ti GPU. We resize and crop the input images in the same way as Faghti et al. [55] The Bi-LSTM is initialized with Xavier init [99] and uses the dropout with a probability of 0.5 to avoid overfitting. We train image and text encoders using Adam [100] optimizer, set

the mini-batch size to 128 and the learning rate to 0.0002 with decay every 15 epochs. The model is trained for 30 epochs. The distance function ℓ used in Equation 5.3 is the cosine distance function. Following [55], we set the margin α in Equation 5.3 to 0.2 in all experiments. The thresholds μ_{down} and μ_{up} in Equation 5.4 are empirically set to 0.05 and 0.5, respectively.

Evaluation Metrics: We evaluate our experimental results by R@K and R-sum metrics. R@K is the abbreviation of Recall at K, the proportion of correct matches in the top $K = [1, 5, 10]$ of retrieving rank. R-sum is defined as:

$$\text{R-sum} = \overbrace{\text{R@1+R@5+R@10}}^{\text{Image-query-Text}} + \overbrace{\text{R@1+R@5+R@10}}^{\text{Text-query-Image}}. \tag{5.9}$$

		MSCOCO 1K test images						
		Image-query-Text			**Text-query-Image**			
Method	**Encoder Backbone**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R-sum**
GMM-FV [101]	VGG, GMM+HGLMM	39.4	67.9	80.9	25.1	59.8	76.6	349.7
DVSA [52]	RCNN, Bi-RNN	38.4	69.9	80.5	27.4	60.2	74.8	351.2
VQA-A [102]	VGG, GRU	50.5	80.1	89.7	37.0	70.9	82.9	411.1
TOP-k Ranking [103]	VGG, MLP	47.8	80.7	87.9	38.1	77.8	87.1	419.4
CMPM [69]	ResNet, Bi-LSTM	56.1	86.3	92.9	44.6	78.8	89.0	447.7
NAR [72]	ResNet, HGLMM	61.3	87.9	95.4	47.0	80.8	90.1	462.5
DPC [104]	ResNet, TextCNN	**65.6**	89.8	95.5	47.1	79.9	90.0	467.9
VSE++ [55]	ResNet, GRU	64.6	90.0	95.7	52.0	84.3	92.0	478.6
IMC(ours)	ResNet, Bi-LSTM	65.3	**90.8**	**96.4**	**53.9**	**86.0**	**93.6**	**486.0**

Table 5.1 – Comparison results with the state-of-the-art methods on MSCOCO [96] 1K dataset. R@1, 5, 10 of two direction queries are listed and sorted by R-sum of 1K test. The bests are in bold. We collect the state-of-the-art results from their papers. The second column gives each method's backbone networks of the image and text feature encoder, respectively.

5.3.3 Experimental Results

Table 5.1, Table 5.2 and Table 5.3 show the results of our approach on the MSCOCO [96] dataset and Flickr30K [95] dataset, respectively. Here, the used similarity function is L1 distance as defined in Equation 5.7 and the parameter λ is set to 1.

We can see that our approach (a typical "pairwise embeddings learning" architecture combined with IMC loss) achieves the highest performance in most cases on MSCOCO,

		MSCOCO 5K test images						
		Image-query-Text			Text-query-Image			
Method	**Encoder Backbone**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R-sum**
GMM-FV [101]	VGG, GMM+HGLMM	17.3	39.0	50.2	10.8	28.3	40.1	185.7
DVSA [52]	RCNN, Bi-RNN	16.5	39.2	52.0	10.7	29.6	42.2	190.2
VQA-A [102]	VGG, GRU	23.5	50.7	63.6	16.7	40.5	53.8	248.8
CMPM [69]	ResNet, Bi-LSTM	31.1	60.7	73.9	22.9	50.2	63.8	302.6
DPC [104]	ResNet, TextCNN	41.2	70.5	81.1	25.3	53.4	66.4	337.9
VSE++ [55]	ResNet, GRU	**41.3**	71.1	81.2	30.3	59.4	72.3	355.6
IMC(ours)	ResNet, Bi-LSTM	41.1	**71.5**	**81.9**	**30.6**	**61.7**	**74.1**	**360.9**

Table 5.2 – Comparison results with the state-of-the-art methods on MSCOCO [96] 5K dataset. R@1, 5, 10 of two direction queries are listed and ordered by R-sum of 1K test. The bests are in bold. We collect the state-of-the-art results from their papers. The second column gives each method's backbone networks of the image and text feature encoder.

except that DPC has the highest R@1 value on MSCOCO 1K test images and the VSE++ has the highest R@1 value on MSCOCO 5K test image for the Image-query-Text task. Our approach also achieves the highest performance in terms of all the metrics for both the Image-query-Text and the Text-query-Image tasks on Flickr30K.

5.3.4 Ablation study

In Table 5.4, the ablation study on the Flickr30K dataset clearly shows the improvement brought by our IMC loss. When $\lambda = 0$, the $\texttt{IMC}(I)$ and $\texttt{IMC}(T)$ in Equation 5.3 are equal to zero, thus it indicates that the MH loss is used. With $\lambda = 1$, we show the results using different similarity functions (δ in Equation 5.4). We can see that whatever the similarity function is used, the IMC loss achieves better results than the MH loss in general.

From Table 5.4, we can also observe the influences of different similarity distance on the results and conclude that using the L1 distance (δ_{l1}) can achieve the best performances in general, especially in terms of R@1 and R-sum.

The influence of the weight parameter λ

We vary λ to evaluate the influence of this weight parameter in Equation 5.4. Figure 5.2 shows part of the results using δ_{l1} on Flickr30K. With the increase of λ, intra-modal pairs gain more emphasis. The experimental results peak at $\lambda = 1$ and then decline. This may

Figure 5.2 – Experimental results with different values of λ on Flickr30K.

	Flickr30K						
	Image-query-Text			Text-query-Image			
Method	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R-sum**
DVSA[52]	22.2	48.2	61.4	15.2	37.7	50.5	235.2
VQA-A [102]	33.9	62.5	74.5	24.9	52.6	64.8	313.2
GMM-FV [101]	35.0	62.0	73.8	25.0	52.7	66.0	314.5
kNN-margin [105]	36.0	64.4	72.5	26.7	54.3	65.7	319.6
TOP-k Ranking [103]	41.3	70.3	79.8	33.1	61.5	72.9	358.9
BSSAN [106]	44.6	74.9	84.3	33.2	62.6	72.9	372.5
MDM [73]	44.9	75.4	84.4	34.4	67.0	77.7	383.8
CMPM+CMPC [69]	49.6	76.8	86.1	37.3	65.7	75.5	391.0
VSE++ [55]	52.9	80.5	87.2	39.6	70.1	79.5	409.8
NAR [72]	55.1	80.3	89.6	39.4	68.8	79.9	413.1
DPC [104]	55.6	81.9	89.5	39.1	69.2	80.9	416.2
SCO [107]	55.5	82.0	89.3	41.1	70.5	80.1	418.5
CVSE++ [108]	56.6	82.5	90.2	42.4	71.6	80.8	424.1
CMKA [109]	55.7	82.9	90.0	45.0	73.4	82.7	429.7
IMC(ours)	**58.5**	**85.0**	**91.2**	**45.4**	**74.8**	**83.0**	**437.9**

Table 5.3 – Comparison results on Flickr30K

	Flickr30K						
	Image-query-Text			Text-query-Image			
IMC	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R-sum**
$\lambda = 0$ (*i.e.*, MH loss)	57.1	83.7	90.9	44.5	74.5	83.2	433.9
$\lambda = 1, \delta_{\text{msd}}$ (Eq.5.6)	56.7	83.5	**91.4**	44.9	**75.5**	83.2	435.2
$\lambda = 1, \delta_{\cos}$ (Eq.5.5)	57.4	84.1	90.9	44.9	75.1	83.3	435.9
$\lambda = 1, \delta_{l2}$ (Eq.5.8)	58.0	84.2	90.5	44.7	75.1	**83.4**	435.9
$\lambda = 1, \delta_{l1}$ (Eq.5.7)	**58.5**	**85.0**	91.2	**45.4**	74.8	83.0	**437.9**

Table 5.4 – Influence of different similarity distances in IMC

indicate that intra-modal and inter-modal pairs in the IMC loss are equally important.

5.4 Conclusion

In this chapter, we have proposed a new loss (IMC loss) adapted for image-text retrieval and demonstrate its effectiveness using a two-branch "pairwise embeddings learning" network on two popular datasets. Our network outperforms tested state-of-the-art

image-text retrieval methods and our IMC loss can improve the network's performances, compared to the MH loss. Without loss of generality, the IMC loss can also be used with other three categories of network architecture [97]. It may also improve their performances, which will be verified in our future work.

This part work have published in:

— J. Chen, L. Zhang, Q. Wang, C. Bai, and K. Kpalma, « Intra-modal constraintloss for image-text retrieval », in 2022 IEEE International Conference on ImageProcessing (ICIP), 2022, pp. 4023-4027. doi: 10.1109/ICIP46576.2022.9897195. (cf. [110])

Conclusion of Part II

Cross-modal retrieval has drawn much attention in both computer vision (CV) and natural language processing (NLP) domains. With the development of convolutional and recurrent neural networks, the bottleneck of retrieval across image-text modalities is no longer the extraction of image and text features but an efficient loss function learning in embedding space. Many loss functions try to closer pairwise multimodalities features in the latent space.

In this part, we start by reviewing the current state-of-the-art in image-text retrieval. We categorize the mainstream image-text retrieval models into four types:

— pairwise learning embedding methods;
— adversarial learning methods;
— interaction learning methods;
— attributes learning methods.

Then, we compare the strengths and weaknesses of each category with the public result on the multimodal database.

We then introduce a new loss function, *i.e.*, IMC Loss, that enhances the feature representation of image-text pairs in the cross-modal latent space. In our experiments, we proposed IMC on a two-branch "pairwise learning embedding methods" network, which further improves the effectiveness of multimodal models on open data. Experimental results show that our approach outperforms state-of-the-art bi-directional image-text retrieval methods on Flickr30K and Microsoft COCO datasets. Our code is publicly available [1]. In future work, we will try to expand IMC to other categories that might bring enhancements.

1. `https://github.com/CanonChen/IMC`

Part III

A Generative Multimodal Database and Multi-View Benchmark for Referring Expression Segmentation

Introduction of Part III

Introduction of Referring Expression Segmentation

In recent years, multimodal tasks have attracted most of the attention in the field of artificial intelligence (AI). It covers knowledge related to computer vision (CV), natural language processing (NLP), and is a necessary path to general artificial intelligence (GAI). An increasing number of multimodal algorithms try to bridge the multimodalities gap. However, bridging the gap between modalities in multimodal learning is not a simple task. Language information is usually presented as a sequential feature, while visual information is represented as a planar feature. The first challenge is to extract features from both modalities. The second challenge is to align the huge gap in information entropy for the same semantic concept. For instance, in text, the word "dog" can represent a dog, whereas in an image, this "dog" may consist of tens of thousands of independent pixels. Referring expression segmentation is one of the most challenging tasks in the multimodal domain due to the large difference in information entropy between modalities.

Referring Expression Segmentation (RES) is a pixel-wise binary labeling problem aiming to separate the target object from others and the background region of an image, which requires multiple skills, including visual perception, text comprehension, and visual-linguistic cross-modal reasoning. Although referring segmentation is performed on one object at a time, similar to instance segmentation, it is closer to semantic segmentation in terms of its output, which consists of undifferentiated masks with only one value per pixel. There are few of publicly available databases for assessing RES algorithms, *e.g.*, [111]–[113]. However, most of their annotated data are simple combinations of category labels, which are far from natural language. Most of current pre-trained language models are trained on corpora containing complete sentence structures such as subject-verb-object. In general, complete sentences contain richer semantic information than short tags. But, longer and more complex sentences mean that it is more difficult to extract features. The proposed dataset attempts to consider the trade-off between the richness of textual information and the difficulty of features extraction.

One other difficulty of referring expression segmentation task is the lack of supervised

learning data. The most important factors of supervised learning are labels and ground truth. But both description referring expression and the correspondence pixel-wise ground truth for semantic segmentation are expensive and time-consuming by manual. If we could find a way to automatically generate images and the corresponding descriptive text of the objects in the images, it would greatly increase the number of supervised samples for referring expression segmentation and reduce the cost. With the evolution of computer graphics, one can deploy simulation scenarios automatically through 3D software, *e.g.*, Blender [114]. Then we get the generated multi-view images and pixel-level mask ground truth by the calculated distance between the objects in the scene and the digital camera, light reflection, and other conditions. In addition, one can also use Blender's python API (application programming interface) to generate expression text based on referring objects in the scenes.

After selecting a multimodal task and creating the dataset, the question remains whether the model can truly understand high-level semantic information or if it merely overfits to the ground truth. To test the model's semantic understanding, it is necessary to evaluate its performance from multiple perspectives beyond those used during training. Existing benchmarks and datasets do not provide enough information to distinguish between high-level semantic understanding and low-level information extraction.

To address these issues, our database provides ground truth from multi-views images, which enables us to observe differences between different views of the multimodal model under the same semantic information. We propose a new metric to capture these differences, which can be used to study the interpretability of multimodal learning. The state-of-the-art (SOTA) models have been evaluated on our multimodal multi-view dataset, and a new benchmark is established for comparison. During our comparison experiments, we observed that models incorporating attention mechanisms tended to achieve superior performance. As a result, we conducted a detailed analysis of the role played by the attention module in multimodal tasks. Figure 9 shows our database framework.

In this Part, our main contributions can be summed up as follows:

1. Construction of a new database for referring expression segmentation, consisting of automatically generating multimodal multi-view pixel-wise masked image and fine-grained semantic referring expression.
2. We analyze and categorize RES SOTA models and emphasizing the significant role of attention module in facilitating cross-modal alignment.
3. A new metric and benchmark for evaluating the multi-view robustness of the SOTA

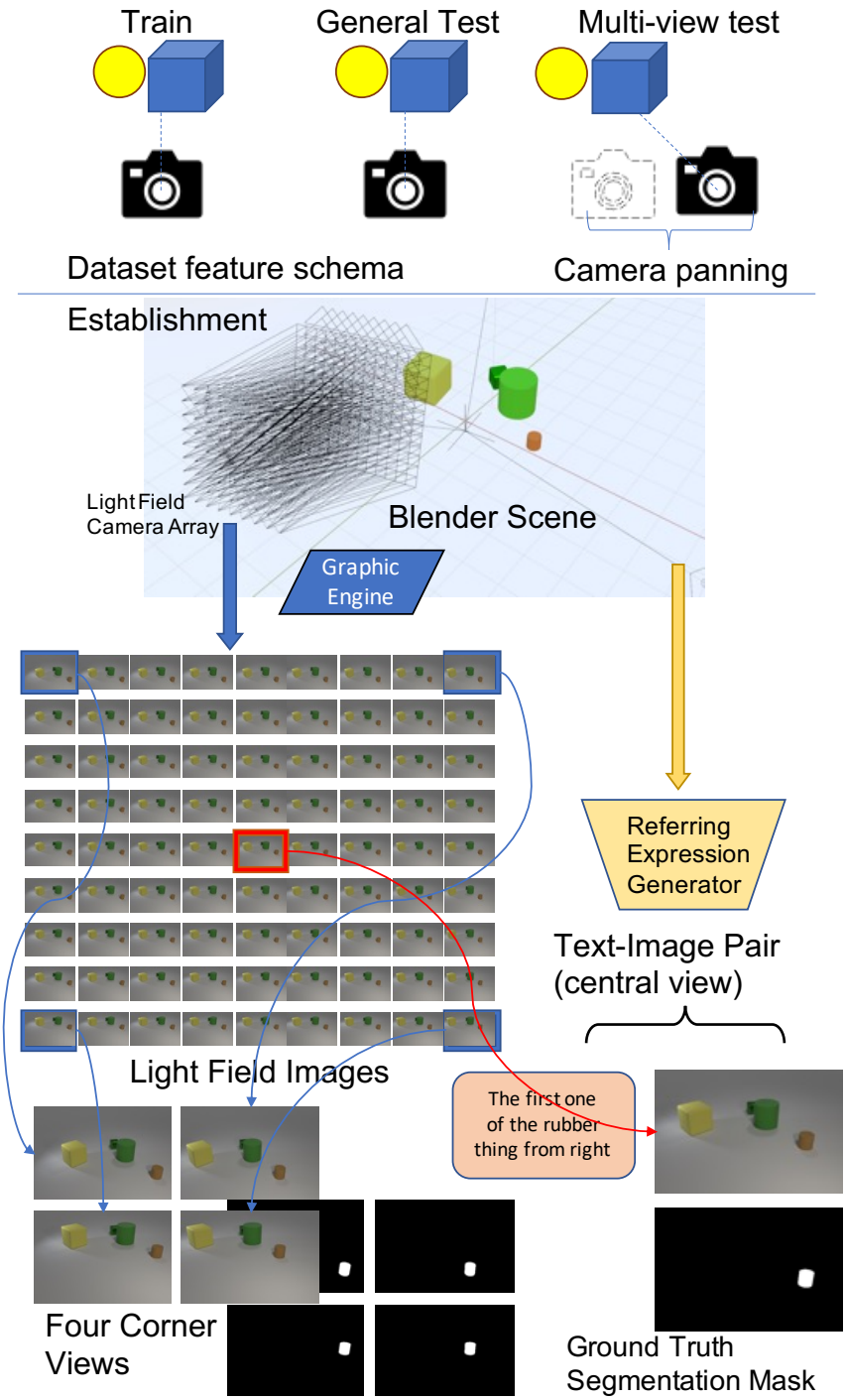


Figure 9 – CLEVR-Remv is a new multimodal multi-view database for RES. The dataset feature schema on the top, that represents our multi-view image for test generating via camera panning. Below that is an example of generating light field image array, text-image pair, and multi-views images with ground truth masks of the four corners from a scene via Blender.

models. Thus evaluating their capability to comprehend high-level semantic information. Furthermore, we visualize the results of comparative experiments to enhance the interpretability of multimodal models.

The rest of the Part III is as follows: in Section 6.2, the work is analyzed in terms of attention mechanism, referring expression, and multi-view segmentation. While section 6.3 explains our multimodal multi-view database creation pipeline. In section 7, we classify existing model structures and attention modules. Finally, in section 7.3, we explain the benchmarks and multi-view robustness metrics we designed for evaluating model performance.

Introduction of Relationships Between RES and Image-Text Retrieval

In this part, we explore another multimodal task known as referring expression segmentation (RES). Although it shares similarities with image-text retrieval in terms of multimodal architectures and the multimodalities feature extraction, there are some key differences.

Firstly, while image-text retrieval requires multimodal feature alignment and sorting, RES involves logical inference, which represents the demand of features representation in the latent space. In image-text retrieval, it is essential for multimodal embedded features vectors to represent a uniform distribution within the latent space. The level of correspondence between query vectors and candidate vectors can be quantified using distance measurements. In RES, the multiomodal model strives to establish a correlation between feature vectors and high-level semantic information, thereby enabling reasoning processes that yield the intended referent results. Furthermore, in image-text retrieval, a distance function is used as the employed loss function, whereas RES utilizes the Intersection over Union (IoU) metrics as its corresponding loss function. Additionally, while image-text retrieval retrieves images and text as global features, RES requires fine-grained labeling of image pixels (*i.e.*, pixel-wise mask) and more detailed semantic understanding of the corresponding text pair. Different combinations of subject and object entities can produce many variations in the text, making RES a more challenging task than image-text retrieval at the same scale of data.

In summary, RES has a higher information content and larger modality gap between

different multimodalities data, making it a more complex and challenging task than image-text retrieval.

Chapter 6

A Generative Multimodal and Multi-View Dataset for Referring Expression Segmentation

6.1 Introduction

With advancements in computer graphics, we have achieved the capability to simulate dynamic changes in lighting within large-scale real scenes using 3D software. The software employs a physics engine to simulate the behavior of all light sources present in the scene. This simulation includes the intricate interplay of light, encompassing phenomena such as reflection, scattering, and refraction as light interacts with the surfaces of objects. As depicted in Figure 6.1, the illumination from the ceiling lamp radiates onto the adjacent walls, while the light emitted by the walls illuminates the nearby rectangular surface. This rectangular surface exhibits a phenomenon known as color reflection, where it reflects red and green light separately. Subsequently, these reflected colors are captured by the camera's viewpoint.

According to Figure 6.2, we can obtain the light diffusion equation as shown

$$I_{diff} = K_d \, I \, \cos\theta \tag{6.1}$$

where I is the incoming light and I_{diff} is the reflected light, K_d is the coefficient of diffusion of object surface, θ is the angle of incoming light. The surface of objects of different materials will have different diffusion coefficients K_d. For instance, the Phong reflection model is often employed to accurately represent how light behaves when it interacts with different materials. A shiny material object's reflection could be denoted

1. https://en.wikipedia.org/wiki/File:Cornellbox_pathtracing_irradiancecaching.png

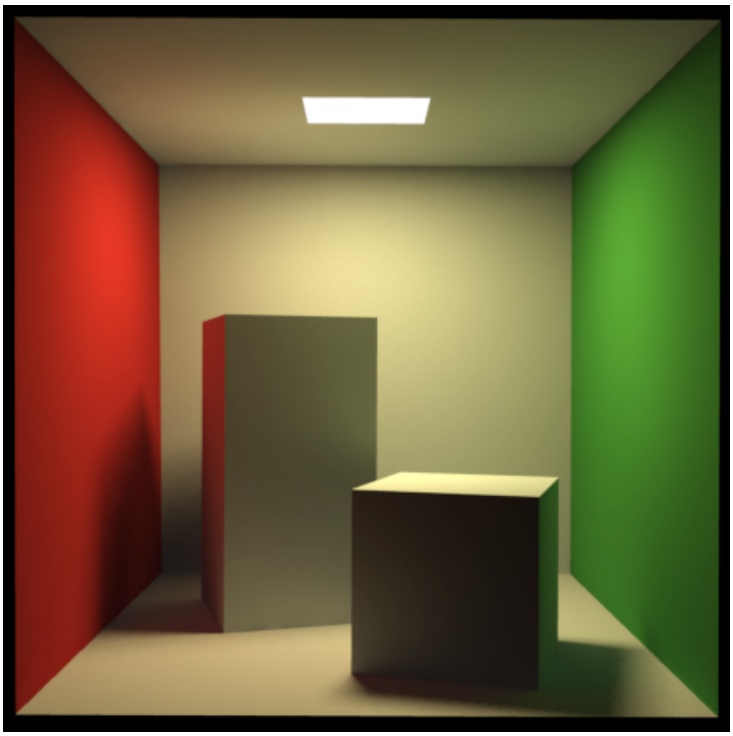


Figure 6.1 – An indirect diffuse scattering simulated picture. [1]

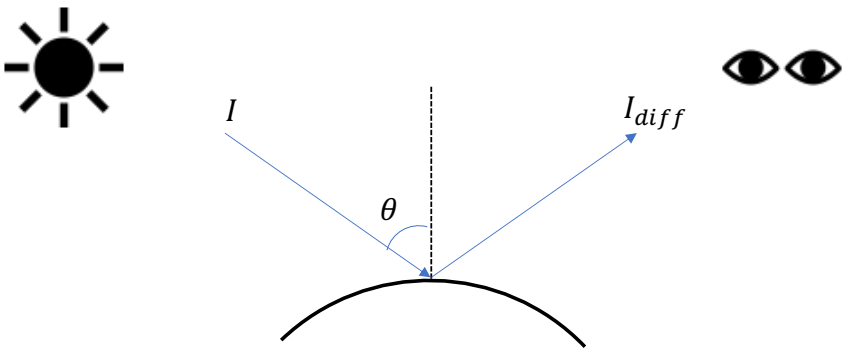


Figure 6.2 – Illustration of light diffusion.

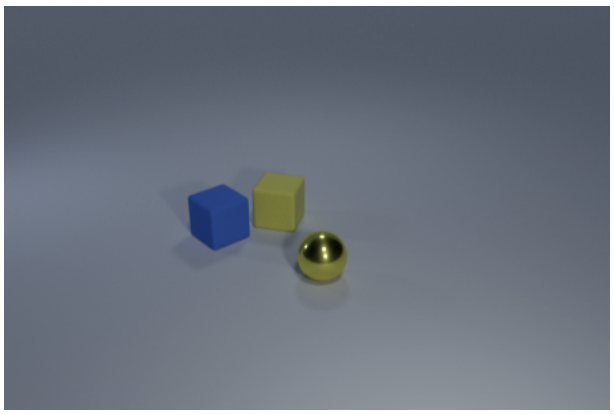


Figure 6.3 – Example rendering image of our dataset.

as:

$$I_{diff} = K_a I_a + \sum_{I \in lights} (K_d I_d \cos\theta_{I_d} + K_s I_s \cos^n \theta_{I_s}) \tag{6.2}$$

where I_a represents the ambient reflection, I_d represents the diffuse reflection, I_s represents the specular reflection, n represent the shininess constant for this material, K represent weights. By considering factors like the surface normals, light source positions, and viewing angles, the Phong reflection model enhances the visual realism by accurately simulating the reflection of light on object surfaces.

In Figure 6.3, the cube exhibits characteristics that align with the diffraction Equation 6.1, indicating that the interaction of light with roughness surface follows diffusion principles. On the other hand, the sphere displays properties that are more closely associated with the Phong reflection, *i.e.*, Equation 6.2, suggesting that the behavior of light interacting with metal surface follows scattering principles.

Based on the reflected light observed in the figure, we can make deductions about the surface finish of the objects. The specific characteristics of the reflected light, such as its intensity, direction, and color, can provide insights into the surface texture, roughness, and material properties of the objects. By analyzing these reflections, we can gather information about the surface finish and potentially infer details about the objects' physical properties and composition.

Overall, the continuous development of computer graphics enables us to achieve remarkable fidelity in simulating lighting conditions, contributing to highly immersive virtual environments and lifelike visual effects in diverse applications. In theory, given ample computational resources, it is possible to calculate subtle variations in lighting that are

imperceptible to the human eye in certain scenarios. Certainly, in the database we constructed, we exclusively utilized templates that incorporated the diffusion coefficient and did not incorporate the modeling of the new reflection equation. The focus was on capturing the diffuse behavior of the materials rather than considering the specific characteristics of the reflection equation.

6.2 Related Work

Since this work belongs to a pioneering field of multi-view fine-grained semantic referring expression segmentation, so far we have no fully consistent related work. Therefore, we review the latest developments in related research from two perspectives: referring expression dataset, and multi-view segmentation dataset.

6.2.1 Referring expression dataset

There are three major popular referring expression datasets: RefCOCOg [111], RefCOCO [112] and RefCOCO+ [112]. All of them are manully annotated on MSCOCO [96] collection. Although manually labeled databases are more consistent with human perception, they are expensive and subject to subjectivity. There are also objective disadvantages, such as the restriction of categories by MSCOCO and the short average length of annotated expressions (*i.e.*, RefCOCOg annotation average length of expressions is 8.43 words, RefCOCO is 3.61, and RefCOCO+ is 3.53). UNC [113], UNC+ [113] and RefCOCOg [111] employ the methods for generating referring expressions in the MSCOCO dataset based on image caption-like tricks. CLEVR [115] proposes a diagnostic database for generating visual question-and-answer systems and evaluating the reasoning abilities of various components, including CNN, LSTM (Long short-term memory [43]), Bag-of-words and spatial attention. CLEVR-Ref+ [116] converts CLEVR from a diagnostic dataset into referring expression dataset. The initial questions are converted into referring expressions, while the initial answers are converted into semantic segmentation masks. In addition to CLEVR-Ref+, there are a number of other CLEVR-based generative segmentation databases. ClevrTex [117] mainly adds many texture features templates into the dataset, which enforces the distinction of the objects surface. CLEVR-X [118] builds a visual reasoning dataset for the explanations derived from natural language. Cops-Ref [119] generates referring expressions from natural image for comprehension, which is inspired by CLEVR's

expression generation.

6.2.2 Multi-view segmentation dataset

The CLEVR-like database reduces the cost of annotation for large-scale segmentation and enables more precise segmentation of the same object from various viewpoints. MVMO [120] captures segmented images of a scene from 25 cameras evenly distributed in the upper hemisphere and covering the area above the scene. The light field camera array generates a partial database image for UrbanLF [121], which is then used to segment the central view.

Our database follows the abovementioned databases, *i.e.*, it is based on CLEVR-Ref+, and utilizes a light field camera array to obtain a multimodal generative database of multi-views, for comparing and evaluating multiple 2D referring expression segmentation SOTA models with and without attention mechanisms. Compared to other databases, our database has more perspectives on images, and in addition, we use sentences instead of the expression phrase, which requires a higher understanding of scene semantics from multimodal models. In other words, our database contains information closer to high-level semantics regarding images, text, and cross-modality.

6.3 Database Construction

This section depicts our database construction from three process pipelines: scene layout, images rendering, and expression generation.

6.3.1 Scene Layout

Following the construction method of the CLEVR-Ref+ dataset, we use the python scripts to layout our CLEVR-based **R**eferring **E**xpression **M**ulti-**V**iews dataset (CLEVR-REMV). Through Blender API, one can adjust the number of items in each scene, resize the frame, and set the cameras. From three up to ten objects are chosen randomly and placed on an initial empty plane for each scene. All of these objects are described by templates that include their inherent attributes, such as size, material, shape, and color. Furthermore, light reflects differently on the surface of objects made of different materials. The scripts also regulate the external state of objects, such as their position and rotation. Each scene has three different angles of light to provide illumination. One can use S_n to

denote a certain script scene containing objects from 2 to 10 at most. We generated a total of 10,000 scenes ($n \in N = 10,000$), wihch is divided into 7,000 in the training set, 1,500 in the validation set, and 1,500 in the test set.

As an extension of the dataset, we primarily adhere to the CLEVR and CLEVR-Ref+ rules for scene layout, with the most notable change being our camera system. We use an array of light field cameras instead of a single camera. In the CLEVR and CLEVR-Ref+, a scene corresponds to the production of a single image. While in our CLEVR-REMV dataset, a single scene corresponds to a multiple-image group (*i.e.*, 81 sub-apperture images), which are generated by calculating various light angles. In other words, this collection of multiple images is generated by an light field camera system. Each image is the result of a single camera's view. In addition, the distribution of camera positions in the array is correlated, such that the images in the multiple-image group are interrelated from multiple perspectives, *i.e.*, these multiple images could be considered as multi-view images of one light field scene.

6.3.2 Rendering Images

After identifying all the objects in the scene, we employ the MANet [122] 4D light fields camera system in our scene. The light field camera system exhibits a strong relationship between the central view and the other views, which can be defined as follows:

$$\begin{aligned} LF(x, y, u, v) = LF\Big(&x + (u_{\frac{1}{2}} - u)d(x, y), \\ &y + (v_{\frac{1}{2}} - v)d(x, y), \\ &u_{\frac{1}{2}},\ v_{\frac{1}{2}}\Big), \quad u \in [1, 9], \quad v \in [1, 9], \end{aligned} \tag{6.3}$$

where (u, v) represents the camera plane coordinates, and (x, y) is the image plane coordinates, $(u_{\frac{1}{2}}, v_{\frac{1}{2}})$ indicates the central view coordinates, $d(x, y)$ is the shifted distance of the image (x, y). We arranged a matrix of $9 \times 9 = 81$ (*i.e.*, $u \times v = 81$) cameras in each scene to capture the images. All light field cameras are positioned on one side of the scene and capture the scene from parallel angles. In building the 3D scene, we assume that each camera is represented by a visual frustum, then each visual frustum has the same size, shape, and orientation angle. All the visual frustum sections and focal points are coplanar with each other respectively. At the same time, we set the focal points at 2.5cm to make sure each object could be covered by light field camera.

After layout of one scene, light field camera ($C_{u,v}$) renders the corresponding light field

image ($I_{u,v}$) via Blender engine, which can be denote in Figure 6.4

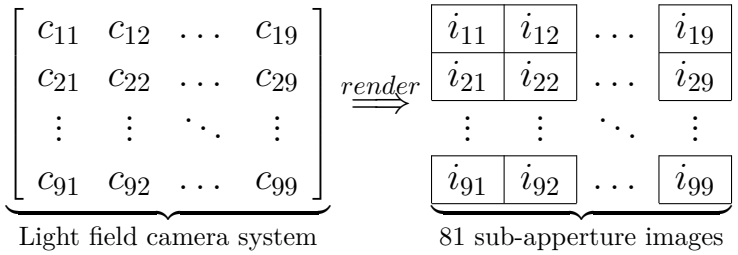


Figure 6.4 – Sub-apperture images rendering.

The size of each light field image is 320×480, so that our database has a total size of over 150GB.

In addition to rendering the light field images, we also record object annotations and ground truth for semantic segmentation based on different camera views.

6.3.3 Expression Generation

Expression generation follows the same methodology as CLEVR-Ref+. CLEVR has provided templates for generating questions and answers according to the scene layout. CLEVR-Ref+ transforms the question into a reference to an object, and maps the answer to the referred object. But we make some further changes for fine-grained semantic. First, we filter the response of multi-object, so only remains a single-object refering. Secondly, we eliminate referential relations that occur twice or more (2-relation referential) because such relations are better suited for visual question-answering tasks and are rarely encountered in ordinary natural language scenarios. Finally, as our camera system operates based on light field capability, movement of the camera coordinates can result in particular objects being completely occluded or disappearing from the viewfinder. Thus, we remove those missing pairs ($< 0.1\%$). We denote the pair of the referred object and the referring expression as $P_r = \{O_j, E_k\}$, where r, j, k represent the index of the referring Pairs, Object and Expressions in one scene, respectively. Each scene contains 7 or 8 pairs of object and expression. We can obtain the proportional relationship of the quantities in the database:

$$s_n = 9 \times 9 \; c_{u,v} = 81 \; i_{u,v} \approx 8 \; P_r$$

The train/validation/test sets have 55,966/11,992/11,989 pairs of objects and referring expressions, respectively.

6.4 Statistical Analysis

Within this section, we undertake a comprehensive examination of several distributions. Firstly, we analyze the distribution of the number of objects in the image-text pairs. Subsequently, we investigate the distribution of word lengths in the text across the training set, validation set, and test set. To facilitate comprehension and enable effective comparison, we present these distributions in a visual format.

Our training set consists of a total of 55,966 image-text pairs. On average, each image in the training set contains approximately 6.529 objects. Furthermore, the average length of the text pairs accompanying these images is 8.0175, as illustrated in Figure 6.5.

Our validation set consists of a total of 11,989 image-text pairs. The validation set's images typically have 6.528 objects in each one. Additionally, as computed from Figure 6.6, the average length of the text pairs that go with these images is 8.0174.

Our testing set consists of 11,994 image-text pairs in total. Each image in the training set comprises roughly 6,495 objects on average. In addition, as can be computed from Figure 6.7, the average length of the text pairs accompanying these images is 8,0420 characters.

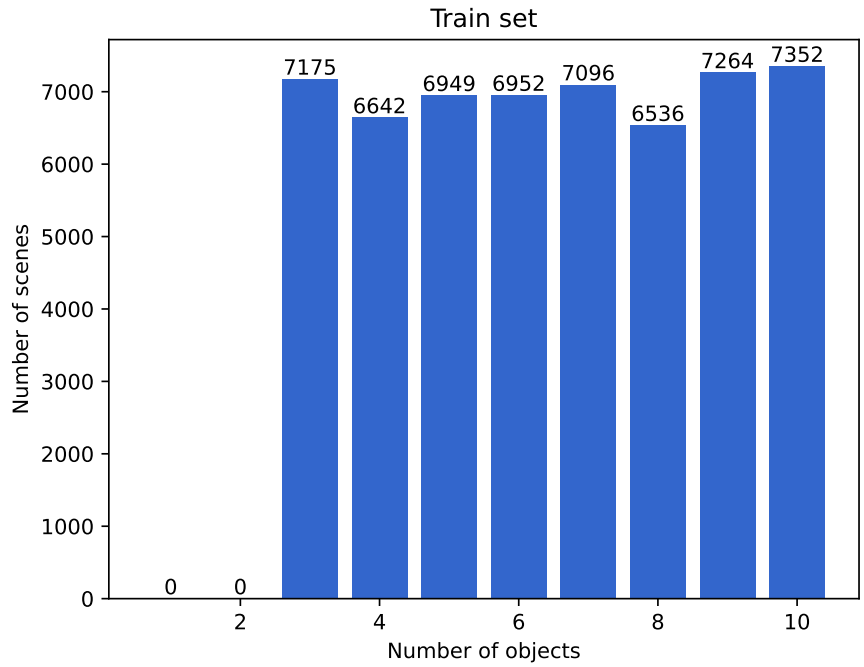
Within our dataset, each scene comprises a grid of 81 views arranged in a 9×9 formation. Among these views, 80 are multi-view images, excluding the central view. It is worth noting that if all objects within a specific view are occluded, the corresponding text pairs associated with that view are subsequently removed.

6.5 Conclusion

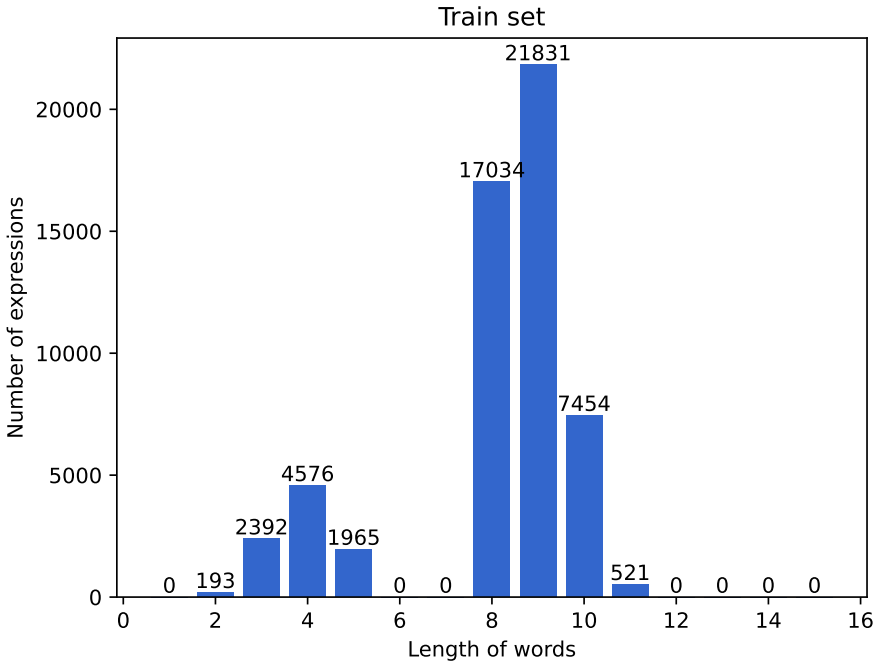
We have constructed a new database for referring expression segmentation (RES) task, consisting of automatically generated multimodal multi-view data. To better understand this RES dataset, we analyze the statistical information.

By utilizing templates with predefined diffusion coefficients, we are able to simulate and represent the diffusing effects of different materials accurately. Additionally, the reflection equation, which encompasses the intricate details of light reflection, is not explicitly modeled in this particular database. In other words, our dataset provides the necessary information to make simple judgments regarding light interactions and scattering effects based on templates.

It's important to note that the exclusion of the new reflection equation from the

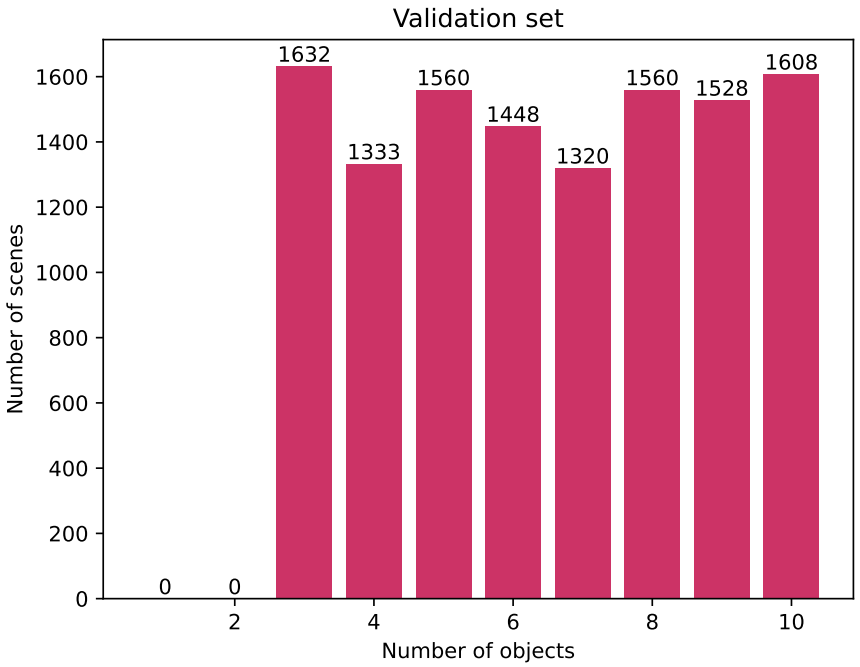


(a)

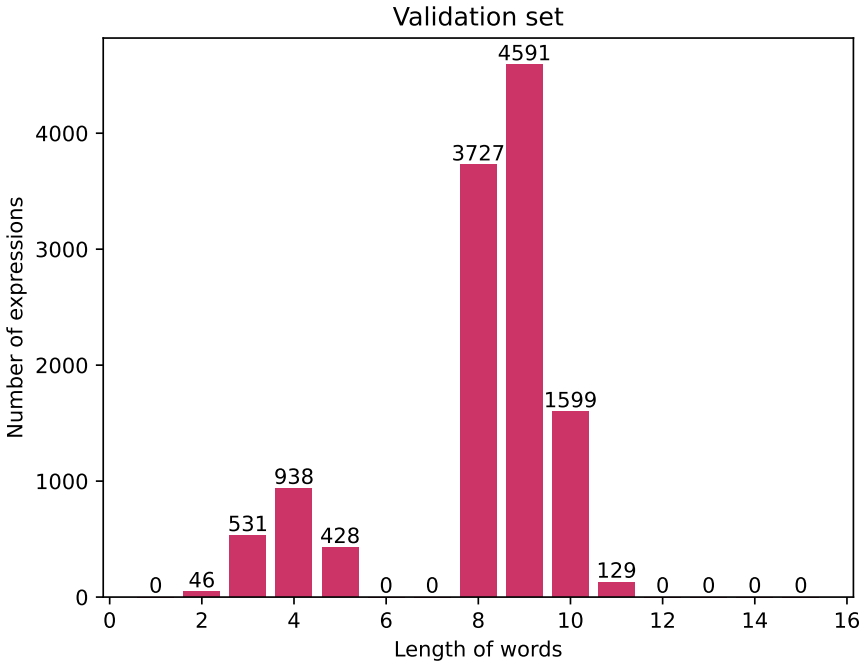


(b)

Figure 6.5 – Train set image-text pairs statistic.

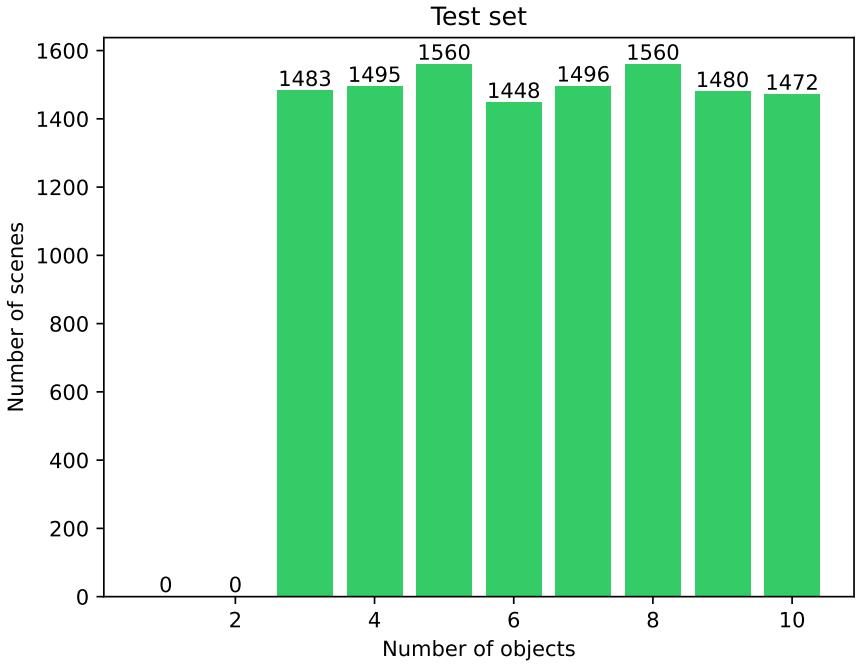


(a)

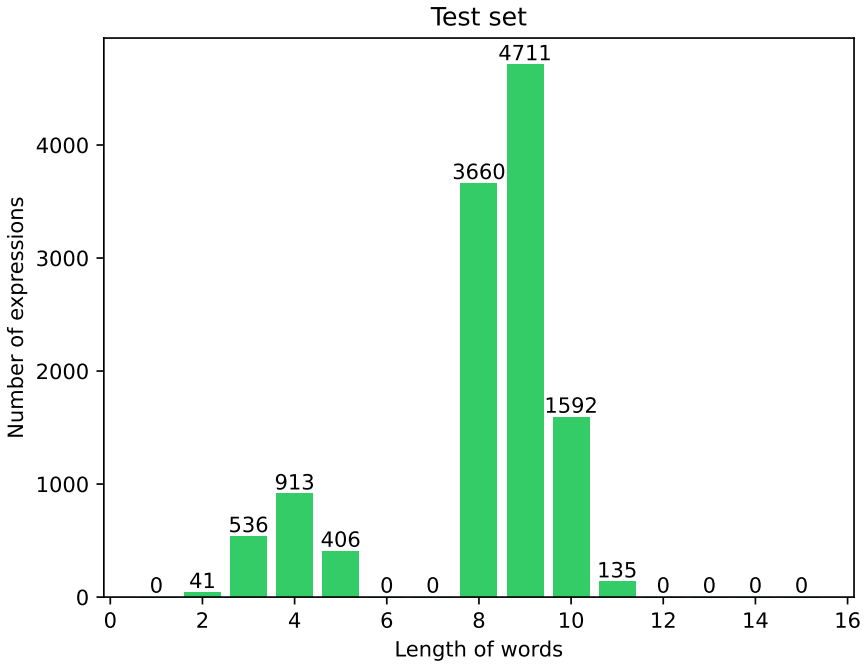


(b)

Figure 6.6 – Validation set image-text pairs statistic.



(a)



(b)

Figure 6.7 – Test set image-text pairs statistic.

database may limit the ability to analyze or infer the specific surface finish characteristics based on the reflected light observed in the figure. If a more comprehensive understanding of the reflection equation is desired, it would be beneficial to incorporate and model it in future iterations of the database or simulations.

Chapter 7

RES Methods Benchmark on the Proposed Dataset

7.1 Introduction

As mentioned in the introduction of Part III, our main objective in this section is to delve into the state-of-the-art models of the Referring Expression Segemtation (RES) algorithm. We conduct an extensive analysis of the cross-modal attention mechanism employed within the model. Additionally, we establish a benchmark experiments and develop multi-view metrics specifically designed to assess the multi-view robustness of the RES model. Finally, we show the experimental comparison results of SOTA models by visualization.

7.1.1 Attention Mechanism Model

The attention mechanism is initially introduced in sequence-to-sequence modeling as a translation task [46]. The Transformer [34] boosts attention mechanism based on self-attention that can simultaneously calculate the relevance between each sequence component instead of waiting for the preceding output token by token, as is the case with RNN. ViT [36] is the first model to adopt the Transformer-based attention model to the vision domain by dividing two-dimensional images into non-overlapping 16×16 patches and computing self-attention between the patches. While ViT addresses the challenge of varying image resolution compared to the textual environment, it does not account for changes in the scale of the same semantic object. In contrast, Swin Transformer [37] tackles this issue with a hierarchical transformation. Due to the existence of attention mechanisms in both the NLP and visual domains, the evolution of cross-modal attention mechanisms is inevitable. CLIP [56] combines image and text pre-training pairs with attention-like contrastive loss, resulting in superior performance on the zero-shot task.

We collect the state-of-the-art referring expression segmentation and methods with released source code as far we know. In 7.2, a concise summary and comparison of each model structure is provided. In 7.2.4, we divide the attention model into 3 categories and show the corresponding categories by the attention module in SOTA.

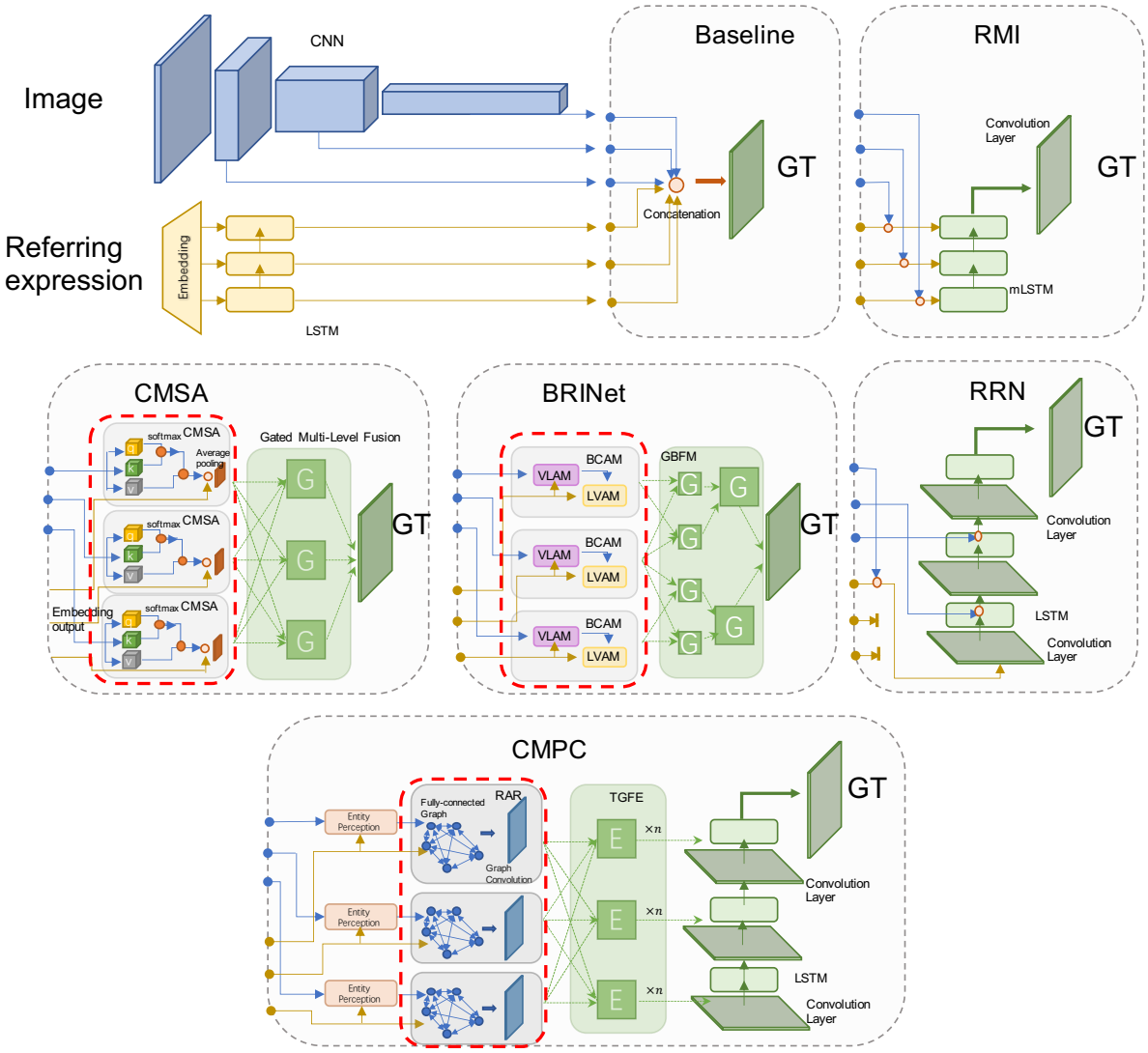


Figure 7.1 – High-level architecture summary of RMI [123], RRN [124], BRINet [125], CMSA [126], CMPC [127] models. The vision parts are colored blue, the language parts are yellow, the cross-modal parts are green. The attention module have been boxed with a red dash line, wihch corresponds to the fourth column in Table 7.1. Figure 7.2-7.6 are the same. GT is short for Ground Truth.

7.2 SOTA Models Structure Classification and Attention Analysis

First, the RES methods are grouped into three categories according to the model structure, which are posterior fusion, anterior fusion and multiple fusion. The posterior fusion methods first extract image and text features with backbone neural networks and then project them onto a latent space. While the anterior fusion methods usually use a multimodal attention module after the fusion components to reduce the distance of corresponding representations in the latent space. The multiple fusion method has multiple interactions along with the image and text feature operations. We have drawn a schematic diagram of the algorithmic framework for each method, see Figs. 7.1-7.6. By visually comparing the algorithms of each category, it is possible to determine their similarities and differences.

7.2.1 Posterior Fusion Methods

In this group, the extracted multimodalities features are concatenated or learned by attention module firstly, then fused with one convolutional layer or other fusion module, see Figure 7.1 and 7.2.

RMI

Liu et al. propose RMI [123], which has a multimodal LSTM model to capture the progression of multimodal information across time. They use LSTM to encode referring expression with a word embedding method, and use the CNN layers to extract image features, which have been trained in Deeplab [128]. At the same time, an 8-dimensions preprocessed spatial coordinate are concatenated after the image features, which follows the method in [129]. Then both image and text features are concatenated and retrained by multimodal LSTM. Specifically, multimodal LSTM that distributes weights according to the time step and spatial position. The high-level architecture of RMI is shown in Figure 7.1.

In a word, the multimodal LSTM ignore lower parts of the input representation but forces the use of high level word-visual interaction and generates multimodal feature with recurrent progression.

RRN

Recurrent Refinement Network (RRN [124]) has been proposed as a model that takes pyramidal features as input to refine the segmentation mask progressively. Like RMI, RRN uses CNN and LSTM to extract image and text features. Then, image feature and text feature are concatenated with 8-dimensions spatial coordinates. Different from RMI, a pyramid structure is used in RRN, which aims to aggregate hierarchical information. This pyramid structure contains 4 multi-scale semantic feature maps drawn from segmented images. Between each semantic feature maps, there is a convolution LSTM layer connected with upper and lower maps. This convolution LSTM also combines with corresponding CNN layers. Different scale feature maps are connected through a top-down pathway within CNN layers. In this pathway, hierarchical information could been captured sufficiently.

LSTM and CNN layers cross-passing merged information produces excellent results on the RES task, even though RRN does not use any attention module in the network structure. This demonstrates that the RNN and CNN layers have a good fundamental feature extraction ability and can accomplish excellent results through network structure design. Of course, this also shows the challenge of attention mechanism to replace RNN and CNN in textual and visual domains.

CMSA

Ye et al. propose the CMSA [126] model, that consists of three parts: multimodal features, cross-modal self-attention (CMSA), and a multi-level fusion gate. Multimodal features are built from the image feature, the spatial coordinate feature, and the language feature for each expression. Then each level multimodal feature is put into a cross-modal self-attention module to build long-range dependencies across separate expressions and spatial areas. At last, the multi-level fusion gate module fuses the features from different levels to yield the final segmentation mask.

CMSA introduces the self-attention mechanism in the cross-modal domain. Different modalities' features are computed and their attention scores are combined using a control gate. Notably, the fundamental models for extracting sequence features, such as RNN or LSTM, are not used in the CMSA model but instead directly attached to word embeddings and CNN features. This is equivalent to word-for-word handling of the expression's sequence as two-dimensional planar information.

BRINet

BRINet [125] has the same structure as the former methods, which use CNN and LSTM to extract image and text features. Unlike the formers, those features are fed into the bi-directional cross-modal attention module (BCAM). BCAM has two parts: one is vision-guide linguistic attention, and the other one is language-guide visual attention. Vision-guided linguistic attention module (VLAM) aims to calculate the association between the text context and each image region. Language-guided visual attention module (LVAM) computes region relationships in contextual language. Finally, BRINet uses a Gated Bi-directional Fusion Module (GBFM) to mix different levels of features and then yields the final segmentation mask.

BRINet employs both directions of attention to obtain multimodal information and enhances the number of control gate structures, but additional tests are required to determine whether this architecture can function as a multi-head attention mechanism.

CMPC

Huang et al. propose the CMPC [127], which also uses the CNN component to extract 3-level image features and LSTM to extract text features. Then 3-level image and text features are fed into Entity-Perception (EP) and Relation-Aware Reasoning (RAR) modules. These two modules aim to emphasize the referent in referring expression with EP and unite the referent entity and spatial image region via RAR. Afterwards RAR module, multi-level features exchange the hierarchical information in Text-Guided Feature Exchange (TGFE) module. Finally, a ConvLSTM component connects after the TGFE module to generate the segmentation mask.

Unlike previous approaches, the CMPC focuses more on semantic information in natural language processing. The CMPC module categorizes referring expressions into entity phrases, attribute phrases, and relation phrases. The visual information is bilinearly fused with entity phrases and attribute phrases in the EP stage. In the RAR stage, the graph convolution maps previous information and calculates relationships with related phrases. Different from LSTM, wihch focuses on sequential information features, the convolutional graph network focuses more on the relational information between multiple entities.

IEP-Ref

IEP-Ref [116] is distinct from previous algorithms, employing a two-layer LSTM to extract text features and a 1-level CNN to derive image features. The fusion stage can be subdivided into modules, and the parameters of various modules can be trained multiple times to increase their effectiveness. Finally, IEP-Ref outputs the predicted features using a convolutional layer of the same size.

IEP-Ref employs a multi-step training strategy to train the model and obtains excellent results with a simple model structure. It is important to note that IEP-Ref's attention mechanism focuses on spatial attention, which does not directly calculate the attention to image and text features but serves as a training aid.

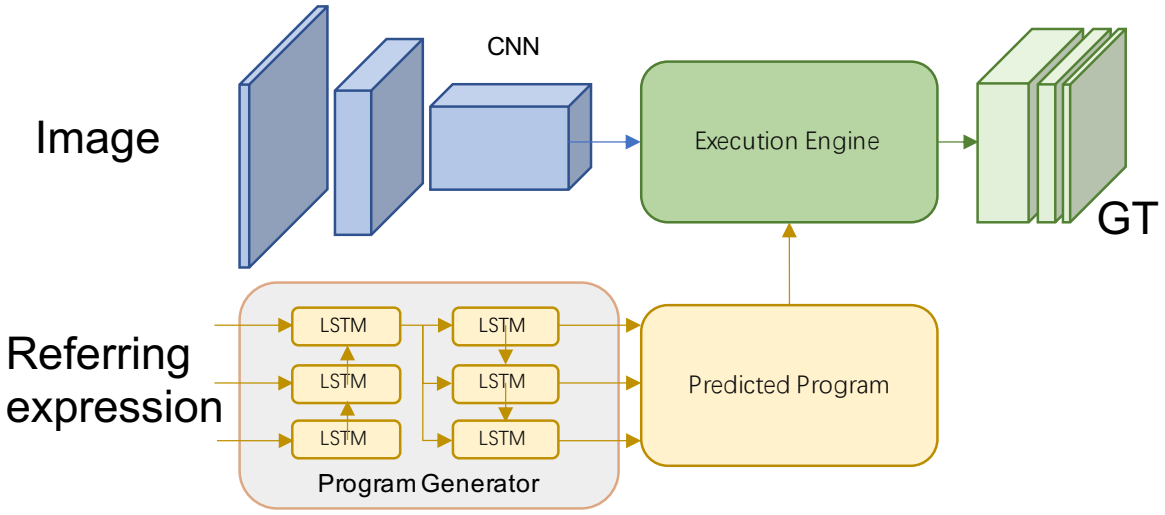


Figure 7.2 – IEP-Ref model's high-level architecture summary.

7.2.2 Anterior Fusion Methods

In this group, the structures of models are more complex. However a typical attention module has been set after fusion module, such as the energy function or transformer-based encoder-decoder, which can be seen in Figure 7.3, 7.4 and 7.5.

MCN

In MCN [130], it uses CNN to extract image features and Bi-GRU [131] to extract language features. Then fuse these two kinds of features are fused together with a multi-scale Multimodal Fusion module. These multimodal features are fed to the REC and

RES branches, respectively. The different branches adjust the attention weights during training. Meanwhile, the two branches further reinforce each other through Consistency Energy Maximization (CEM) module.

MCN proposes an unusual network architecture that can simultaneously solve the tasks of REC and RES. The joint learning of features in a feature transformation space represents cross-modal information.

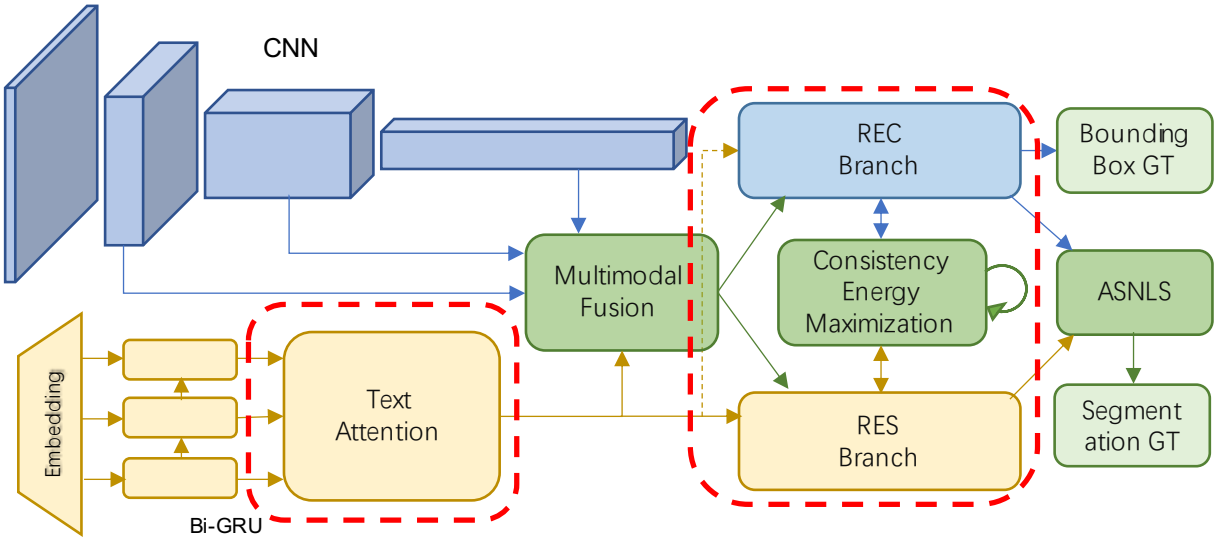


Figure 7.3 – MCN model's high-level architecture summary. The red dash line boxes are the areas where the attentions are in effect.

VLT

In the beginning of VLT [132], the input image and language expression are mapped into the feature space. Secondly, linguistic and visual features are combined by the QGM to generate a set of language query vectors. Simultaneously, vision features are supplied to the Transformer-based encoder to produce a set of memory features. The Query Balance Module then selects the responses from the decoder based on the query vectors acquired from the Query Generation Module (QGM). Finally, the network outputs a mask corresponding to the target object.

For attention operations on input features, VLT uses a comprehensive but shallow transformer layers. The shallow transformer encoder and decoder networks each consist of two layers. Each layer is comprised of either one (encoder) or two (decoder) multi-head attention modules and one feed-forward network, as demonstrated on Figure 7.4.

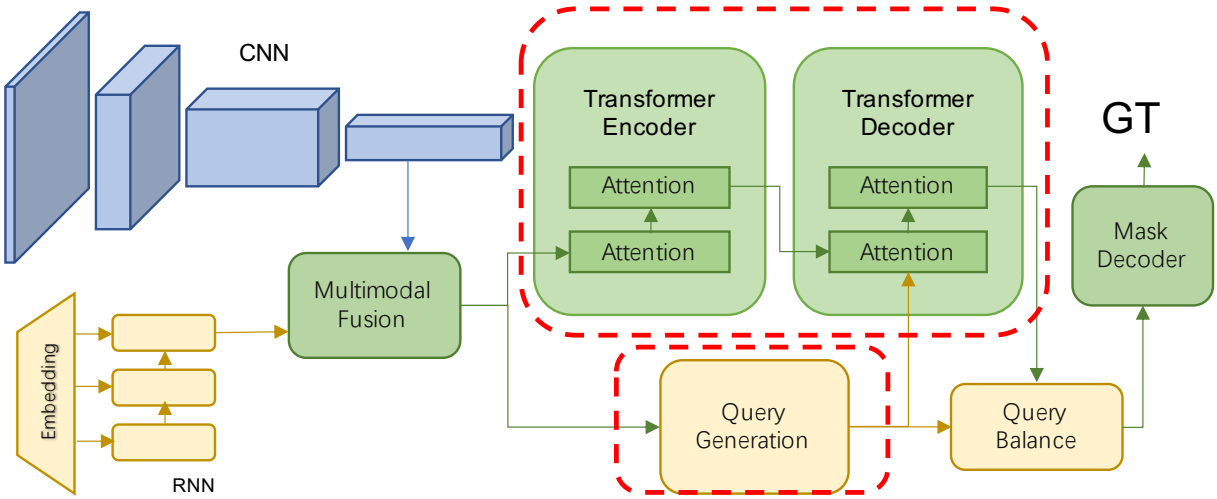


Figure 7.4 – VLT model's high-level architecture summary. The red dash line box indicates the part where the attention works on.

MDETR

MDETR [133] is an end-to-end multi-modal understanding system based on DETR, which is a detection model composed of a CNN followed by a Transformer-based Encoder-Decoder. MDETR uses pre-trained transformer model (*e.g.*, RoBERT [134]) as text encoding, and CNN as image feature extraction. Then MDETR utilizes linear layers as a fusion part, which projects and concatenates text and image features. This fused feature is transmitted to a Transformer-based Cross-Encoder and a Decoder, *i.e.*, DETR [135] .

Since MDETR uses a compound loss function, the transformer output sequence contains several components of the predictions, such as loss for referring expression comprehension, loss for referring expression segmentation, and loss for visual question answering. This means that the output sequence contains different components of feature tokens. We should split the segmentation token from the output if we need the segmentation output sequence.

7.2.3 Multiple Fusions Methods

LAVT

Language-Aware Vision Transformer (LAVT [136]) makes use of a hierarchical vision Transformer to embed language and vision information together in order to promote cross-modal mapping. LAVT uses BERT to embed language expression into word vectors. Those

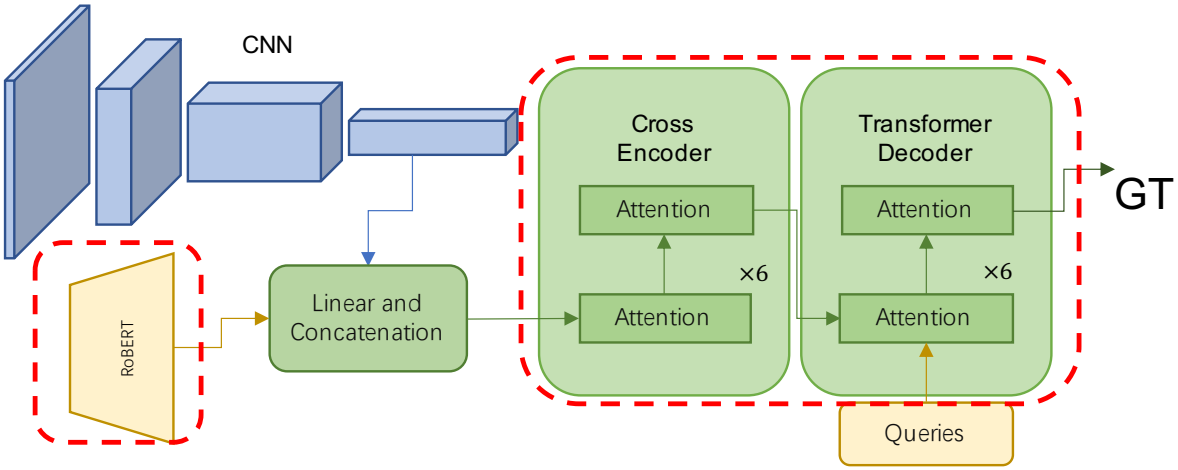


Figure 7.5 – MDETR model's high-level architecture summary. The red dash line boxes are the areas where the attention modules are in effect. "×6" represents 6 same attention layers stacked together.

word vectors combine with vision provided by four stages of Swin Transformer encoding layers. Within each step, the multi-modal features are fused by pixel-word attention module (PWAM), which is intended to correlate language meanings with visual cues densely. And the gating unit is the language gate (LG), a particular unit designed by LAVT to regulate the flow of linguistic data down the language pathway (LP).

LAVT is the only model that employs the attention modules in all components, which includes the underlying features and the high-level semantic information. LAVT's best performance on our cross-modal dataset indicates that attentional mechanism modules can supplant CNN and RNN backbones in the RES task.

7.2.4 Attention Module Category

Although the attention mechanism originates from the NLP domain, it has been increasingly used in cross-modal and visual models, such as CLIP and ViT, which both employ a "Transformer-style" self-attention mechanism. The essence of self-attention mechanism is to compute the relationship between each element in the input vector. This is a significant distinguish from traditional saliency in computer vision, which is a pixel-wise gradient difference detection. Different from the text input, which takes a token (*e.g.*, a word) as the basic unit, an image has tens of thousands of relatively independent pixels. How to choose the number of image feature vector dimensions to align with the text feature vector, (*i.e.*, to bridge the multimodality gap), is an important issue for multimodal

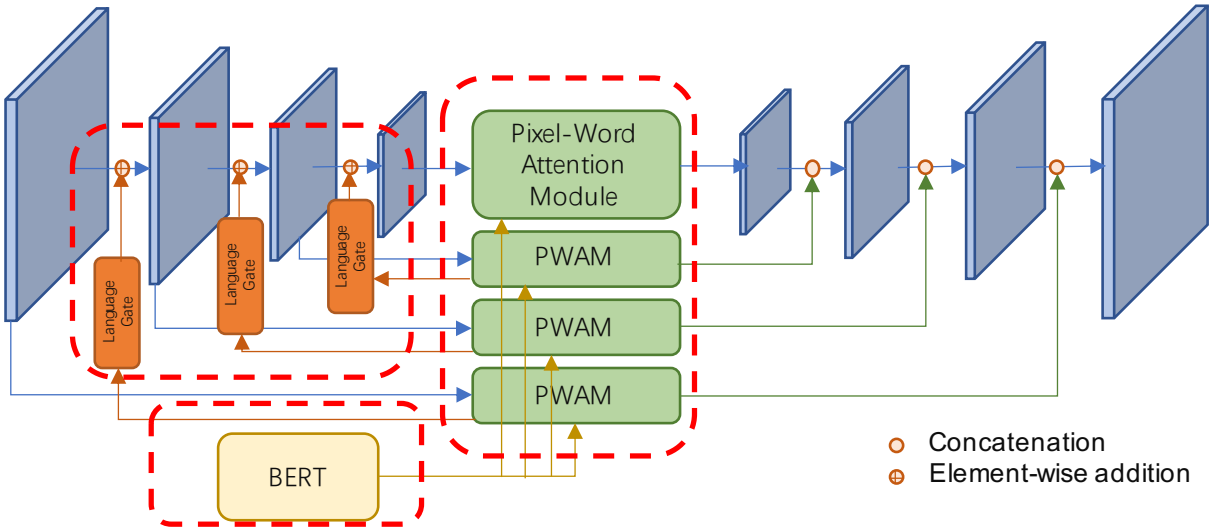


Figure 7.6 – LAVT model's high-level architecture summary. The red dash line boxes are the attention modules work places.

models. Many models use attention mechanisms to bridge the gap between heterogeneous modalities, while others use them to extract features from homogeneous data. However, in this chapter, we categorize attention modules into three types based exclusively on the type of input data: cross-modal attention module, language attention module, and vision attention module. Those three types of attention module adopted by the SOTA models can been seen in Figure 7.7 All the RES models employ the cross-modalities attention module. The Anterior fusion models also use the language attention module to extract expressions features. Only multiple fusion, *i.e.*, LAVT model, covers all the three types. The performance of RES models with the attention types can been seen in Table 7.1.

7.3 Multi-View Benchmark and Robustness Metrics

Our goal is to design a metric that can accurately assess the performance of the multimodal model and highlight the role of the cross-modalities module, particularly in high-level semantic tasks. To achieve this, the model must also be capable of handling input changes beyond simple input image augmentation like cropping or rotation. These changes may include variations in the object's position relative to the frame's plane or in the angle of occlusion that cause distortions. To simulate these changes, we selected quantifiable light field (LF) images as the comparison input. All models are trained on single-view data (central view) and tested across multiple views (four corners). The ratio of

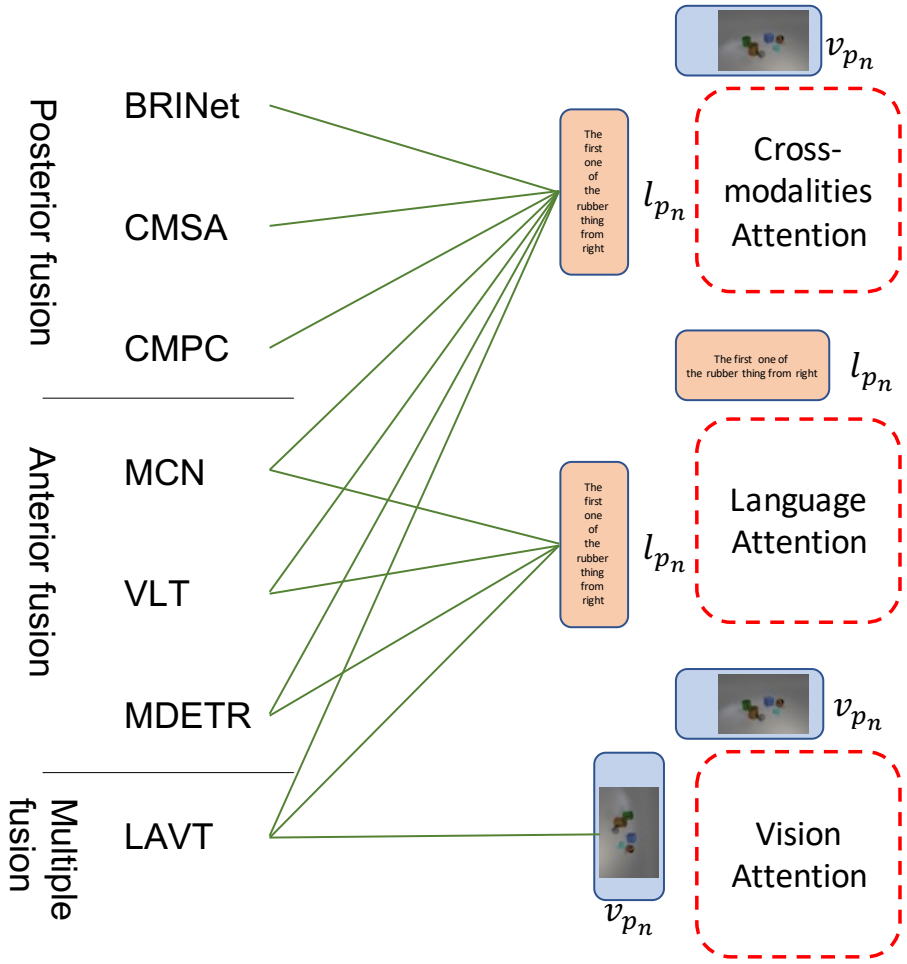


Figure 7.7 – On left side of the figure are the models, on right side are the types of attention modules. The green line between them represents employing ship. Figures 7.1-7.6 shown the corresponding attention module effect area.

the resulting distributions serves as an indicator to evaluate the model's ability to handle input variations and the contribution of the attention module in high-level semantic tasks.

In this section, we first present the metrics and rationale of our design in subsection 7.3.1, followed by benchmark experiments design in subsection 7.3.2, SOTA results analysis on the central view of our CLEVR-Remv database in subsection 7.3.3, and a summary of the role of the attention module in multi-views according to our metrics in subsection 7.3.4.

7.3.1 MVR Metrics

In contrast to other RES databases, we build scenes using a a multiview light field camera system rather than a single camera. This implies that we build a scene using a computer simulation of light reflection as the generated images from multiple viewpoints. Based on the properties of our database, RES algorithms could be measured the abilities of viewpoint panning robustness. This robustness of the model under multi-view RES can directly reflect an enhanced semantic understanding in cross-modal high-level latent space. Therefore, we propose a metric for measuring the Multi-View Robustness (MVR) of panning, which take the shifted distance of light field cameras and the scale of central view into consideration. First, if $Q_{u,v}$ represents the Intersection over Union (IoU) of multi-view and central view in the test, and $Q_{u_{\frac{1}{2}},v_{\frac{1}{2}}}$ represents the ratio of IoU of the central view in the test to itself, we can define it as follows:

$$Q_{u,v} = \frac{\texttt{IoU}_{model}\left(P_r \in \{I_{u,v}|S_{test}\};\boldsymbol{\theta}\right)}{\texttt{IoU}_{model}\left(P_r \in \{I_{u_{\frac{1}{2}}v_{\frac{1}{2}}}|S_{test}\};\boldsymbol{\theta}\right)}, \tag{7.1}$$

where $P_r \in \{I_{u,v}|S_{test}\}$ indicates the pairs of objects and refering expressions in test set, $\boldsymbol{\theta}$ represents the $model()$ parameters of maximum likelihood training stage, which is defined as:

$$\boldsymbol{\theta} = \text{argmax } model(P_r), \quad P_r \in \{I_{u_{\frac{1}{2}},v_{\frac{1}{2}}}|S_{train}\},$$

where $\{I_{u_{\frac{1}{2}}v_{\frac{1}{2}}}|S_{train}\}$ means the center view images in training set. While the model testing with trained parameters $\boldsymbol{\theta}$ is considered as:

$$\texttt{IoU}_{model} = model(P_r;\boldsymbol{\theta}), \quad P_r \in \{I_{u,v}|S_{test}\}.$$

The different test sets of viewpoint images are set by different u, v parameters. Bringing $Q_{u_{\frac{1}{2}},v_{\frac{1}{2}}}$ and $Q_{u,v}$ into the Kullback-Leibler Divergence (KLD) formula:

$$\begin{aligned} \texttt{KLD}_{multi-view} &= \sum_{u,v} Q_{u,v} \log \frac{Q_{u,v}}{Q_{u_{\frac{1}{2}},v_{\frac{1}{2}}}} \\ &= \sum_{u,v} Q_{u,v} \log Q_{u,v} \end{aligned} \tag{7.2}$$

Because these KLD values are negative, a negative sign is added in front of it to ensure non-negativity of the evaluation. Finally, this value is divided by the ratio of camera shifted distance to obtain the metric value. MVR can be described by equation:

$$\texttt{MVR}_{model} = -\sum_{u,v} \frac{Q_{u,v} \log Q_{u,v}}{\text{Dist}(C_{u,v}, C_{u_{\frac{1}{2}}v_{\frac{1}{2}}})}, \tag{7.3}$$

where $\text{Dist}(C_{u,v}, C_{u_{\frac{1}{2}},v_{\frac{1}{2}}})$ indicates the distance between the multi-view camera position and the center camera position. Here we use the Manhattan distance:

$$\text{Dist}(C_{u,v}, C_{u_{\frac{1}{2}},v_{\frac{1}{2}}}) = |u - u_{\frac{1}{2}}| + |v - v_{\frac{1}{2}}|. \tag{7.4}$$

According to equation 7.1, Q is the ratio of multi-view IoU and the central view IoU. Thus, the first step is to design an experiment to evaluate the SOTA *model* central view IoU, which can be define as:

$$\texttt{IoU}_{model} = model(P_r; \boldsymbol{\theta}), \quad P_r \in \{I_{u_{\frac{1}{2}},v_{\frac{1}{2}}} | S_{test}\}.$$

7.3.2 Benchmark Experiment Design

In order to compare SOTA methods fairly and reasonably, we have established three algorithm-level principles, which are the *maximum consistency* principle, the *minimum modification* principle and the *priority of consequences over speed* principle. Each algorithm is adapted to these three principles so that they can jointly compare the pros and cons of the models.

Firstly, we strive for maximum consistency principle across all models to trade-off the model performance and the computational capability. Although there are many different evaluation metrics for RES algorithms, we use the most general metrics, *i.e.*, Precision@K, Mean IoU, and Overall IoU, to give the results. The IoU metrics mean Intersection-

Table 7.1 – 2D referring expression base semantic segemtation SOTAs on our database. Different categories are ranked separately by Overall IoU (OIoU). MIoU means Mean IoU. $P@K$ represents precision@K value in %. V L C indicates whether the Visual, linguistic, and Cross-modal fusion portions of the model use attention mechanisms. * means the sum of multi-stage training iterations. Contrastive represents using Contrastive Loss in the model loss function. The first best performing OIoU model is in red text, the second is blue, the third is green.

		Backbone	V L C	iter/epoch	Option	P@0.5	P@0.6	P@0.7	P@0.8	P@0.9	MIoU	**OIoU**
Posterior fusion	CNN-LSTM (baseline)	DeepLab101 LSTM	- - -	800000	w/ DensCRF	28.84	22.67	11.72	03.55	00.03	26.21	**36.06**
					w/o DensCRF	36.65	29.06	20.85	11.26	01.41	34.78	**42.90**
	RMI [123] (ICCV2017)	DeepLab101 LSTM	- - -	800000	w/ DensCRF	35.47	28.81	18.34	08.85	00.68	32.00	**41.47**
					w/o DensCRF	42.92	35.85	26.60	15.76	03.59	40.16	**46.93**
	BRINet [125] (CVPR2020)	DeepLab101 LSTM	- - ✓	800000	w/ DensCRF	71.69	66.54	56.59	38.09	13.71	60.17	**68.96**
					w/o DensCRF	79.74	76.13	69.64	55.55	25.19	68.66	**72.75**
	IEP-Ref [126] (CVPR2020)	DeepLab101 LSTM	- - -	800000*		85.60	83.52	81.47	78.25	67.61	80.33	**80.91**
	CMSA [124] (CVPR2019)	DeepLab101 -	- - ✓	800000	w/ DensCRF	85.88	83.12	78.11	67.88	35.28	75.72	**82.89**
					w/o DensCRF	91.85	90.06	87.93	83.58	68.88	85.25	**87.65**
	RRN [123] (CVPR2018)	DeepLab101 LSTM	- - -	800000	w/ DensCRF	89.74	87.09	81.72	69.32	34.79	78.36	**84.76**
					w/o DensCRF	93.43	91.81	89.14	83.58	63.74	85.64	88.23
	CMPC [127] (CVPR2020)	LSTM	- - ✓	800000	w/ DensCRF	95.76	94.59	92.09	82.65	45.24	84.79	**89.54**
					w/o DensCRF	98.02	97.62	96.92	95.03	80.90	91.39	93.01
Anterior fusion	MCN [130] (CVPR2020)	Darknet-53 Bi-GRU	- ✓✓	40(epochs)		60.58	51.18	37.74	19.42	00.28	49.43	**49.62**
	VLT [132] (ICCV2021)	Darknet-56 Bi-GRU	- ✓✓	40(epochs)		57.82	56.24	54.51	52.08	42.05	49.10	**54.06**
	MDETR [133] (ICCV2021)	Resnet101 RoBERTa	- ✓✓	40(epochs)	w/o Contrastive	67.61	67.31	66.43	64.20	44.07	50.87	**61.26**
					w/ Contrastive	72.24	71.20	69.74	64.90	34.55	56.01	**64.21**
Multiple fusion	LAVT [136] (CVPR2022)	Swin Tran. BERT-base	✓✓✓	40(epochs)		95.81	95.48	94.85	94.31	93.08	94.08	94.72

over-Union, which indicates a ratio of the prediction area and the ground truth. The Precision@K represents the ratio of images in test, which images' IoU are bigger than the threshold K, where $K \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$. We used two parameters, the number of $iterations = 800000$ and $epoch = 40$, to unify the model's training. 800,000 iterations can fully fit training for posterior models, which could be demonstrated on Figure 7.13.

Secondly, we implement the SOTAs with minimum modification principle, which means that we expect to keep the models as original as possible. We minimize the change of parameters or code of the original model so that those models can run in a unified environment. But original models are deployed on many different platforms, so some changes have been necessary for implementation on our dataset. For example, if we had to choose between reducing the image resolution or reducing the batch size to feed the GPU memory of the experimental platform (NVIDIA RTX2080Ti), we would reduce the batch size first because it is clear that the image resolution has a greater impact on the model. It is essential to point out that MCN and MDETR are multi-task models. And their loss functions are compound loss functions. They use weighting coefficients to balance the RES loss and other task losses. Without specific segmentation coefficients given in their papers, we use the original coefficients for training/testing.

Thirdly, the principle of priority of consequences over speed. Multimodal models are complex and extensive. Efficiency is no longer a significant evaluation metric compared to other machine learning algorithms, especially in the training phase. Good consequences are often the first goal pursued by large models. Many multimodal models using pre-trained backbone extract image and text features and then fine-tune in the fusion stage. So it is unfair to compare time consumption with different pre-trained architecture. Different data input formats also affect the training rate. Some algorithms will store the data alongside the processing data, while others read it while training. Finally, some algorithms are implemented using static graphs (TensorFlow libraries [137]) and others using dynamic graphs (PyTorch libraries [138]). Moreover, static graphs run several times faster than dynamic graphs.

Based on the above three principles, we design comparison experiments to train/test SOTAs.

7.3.3 Central View Results Analysis

The RES methods are fine-tuned on our test set central views to obtain the experimental results in Table 7.1. In this table, RES methods are divided into three categories

based on the models' structure and ranked separately according to Overall IoU. The table provides details on the various models' underlying structures as well as some optional modules, such as whether to use DensCRF [139] or Contrastive Loss [56] for comparison. One of the possible reasons for DensCRF not working is that it performs better for complex and irregular projections, while CLEVR's referring object has a very regular, *i.e.*, geometric image projection. MDETR with contrastive loss significantly improves the cross-modal model as said in the original paper [133]. Although the original paper also cannot explain the reason why the performance is increased by contrastive loss, our benchmark confirms this conclusion. Figure 7.8 gives examples of prediction masks of SOTA models on our CLEVR-Remv's test dataset, which allows us to evaluate the improvement brought by the attention module in a more intuitive way of visualization.

First, considering the posterior fusion methods, we refer to the first seven models in Table 7.1. Comparing the model structure in Figure 7.1, we can conclude that the more dependent on attention structure tends to achieve better segmentation performance. RMI adds a multimodal LSTM module compared to baseline model. BRINet and CMSA both employ attention components to improve their performance. RRN chose mixed LSTM and convolution layers to connect hierarchical features. Although RRN does not utilize the attention mechanism to obtain excellent results, it cannot compete with CMPC, which does. Comparing the CMPC and RRN model structures, we observe that both contain the same components at the end of the fusion phase, but CMPC adds the relation-aware and exchanging modules in front. The relation-aware module employs the graph convolution module to map the relationship between text and image based on the entity relations. RMI, BRINet, IEP-Ref, CMSA, RRN, CMPC were 9.4%, 69.6%, 88.6%, 104%, 106%, 117% higher than baseline, respectively. In addition, we provide the model's results on the validation set along with the number of training iterations in Figure 7.13. Over 800,000 iterations, the baseline and RMI are climbing slowly. BRInet and RRN rise the most in the first 400,000 iterations and then decline slightly. CMSA and RNN are almost similar in outcome, with some samples better than RRN and some worse. CMPC maintains a substantial lead over the other posterior fusion models throughout the entire iteration.

Second, MDETR model significantly outperforms MCN and VLT in the anterior fusion models in segmentation task. However, it should be noted again that both MCN and MDETR are multi-task models, and they both use compound loss functions, meaning that the effect of a single task in these algorithms tends to be correlated with the component coefficients, but we did not change the correlation coefficients for the sake of fairness

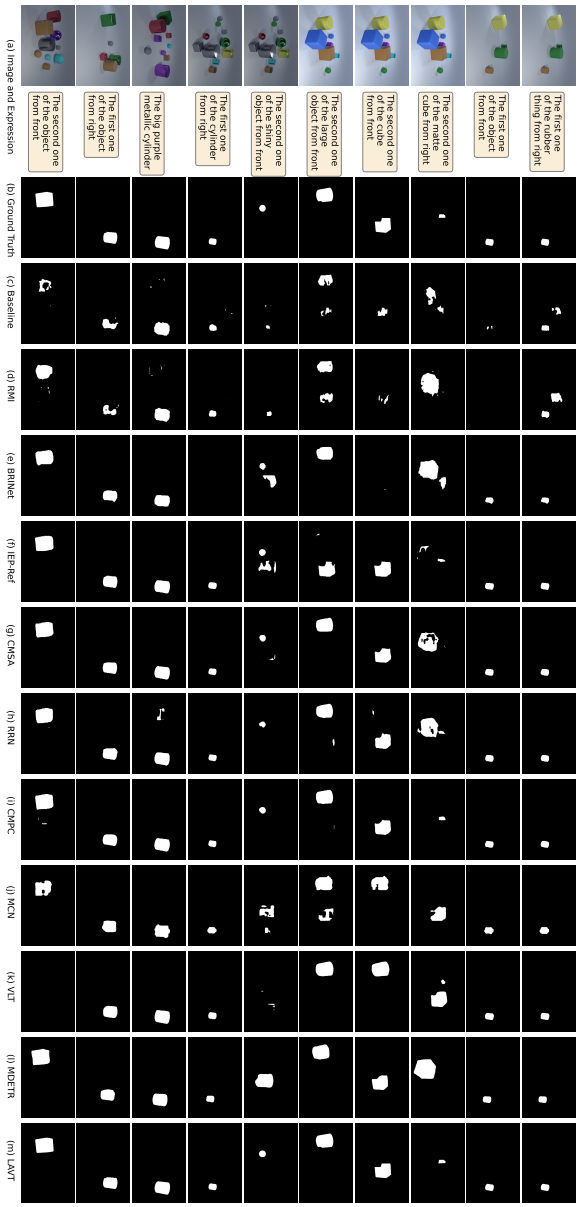


Figure 7.8 – Examples of CLEVR-Remv's central view test. Due to limited space, enlarged images will be displayed on subsequent pages. From the third column to the last column are the prediction of baseline, RMI, BRINet, IEP-Ref, CMSA, RRN, CMPC, MCN, VLT, MDETR , LAVT and ground truth. The three purple prediction masks columns on the right are AF models. They are RES derived from the calculation of the composite loss function (*i.e.*, low weight in the loss function). The backbone component partially replaces the BRInet model in the third column, whose original model is no longer operational. The first and second rows compare various expressions of the same object. Rows 3, 4, and 5 compare occluded objects in the same scene to varying degrees. Rows 6 and 7 are object comparisons within the same scene (with 10 objects). Other examples can be found in the last three rows.

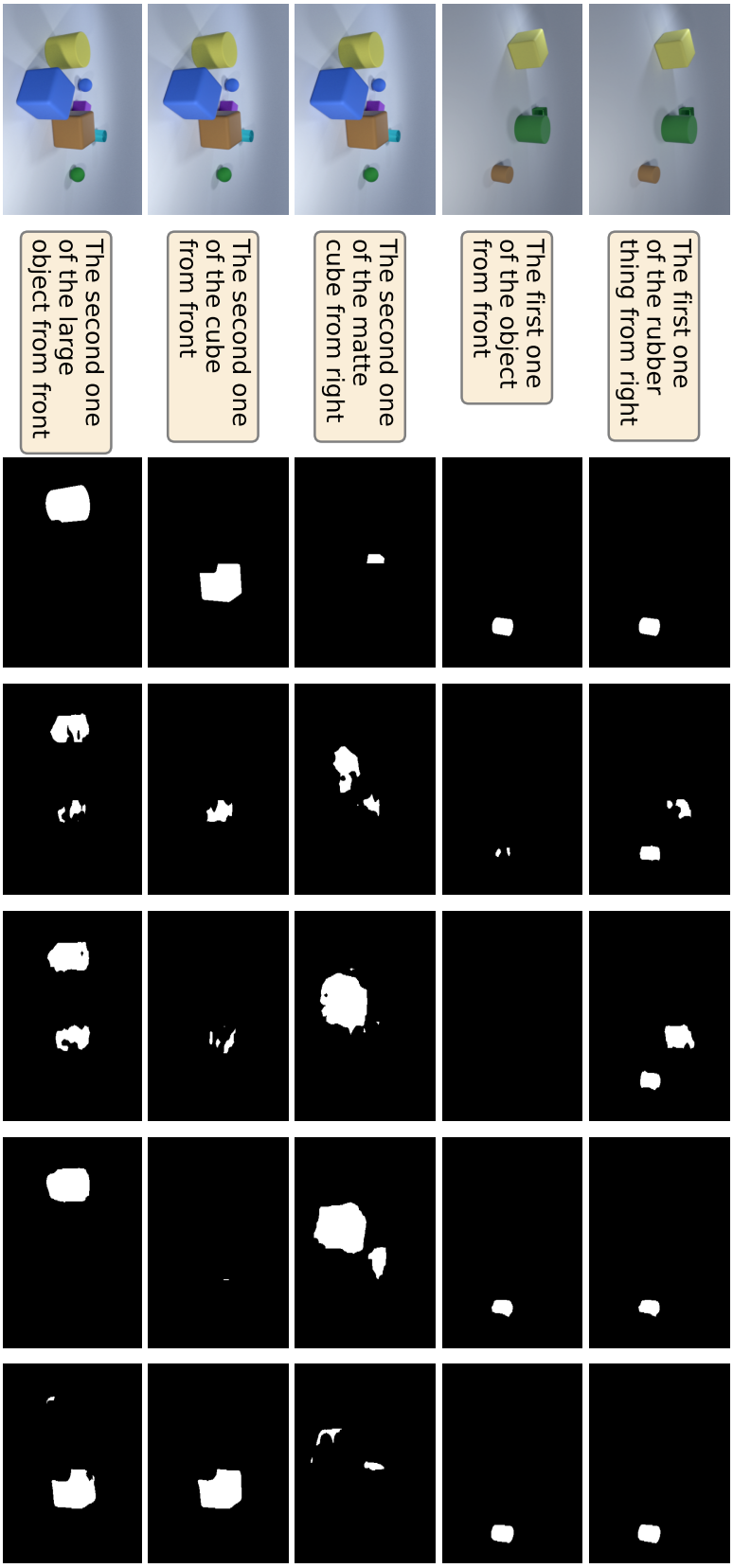


Figure 7.9 – Enlarged Figure 7.8 part A.

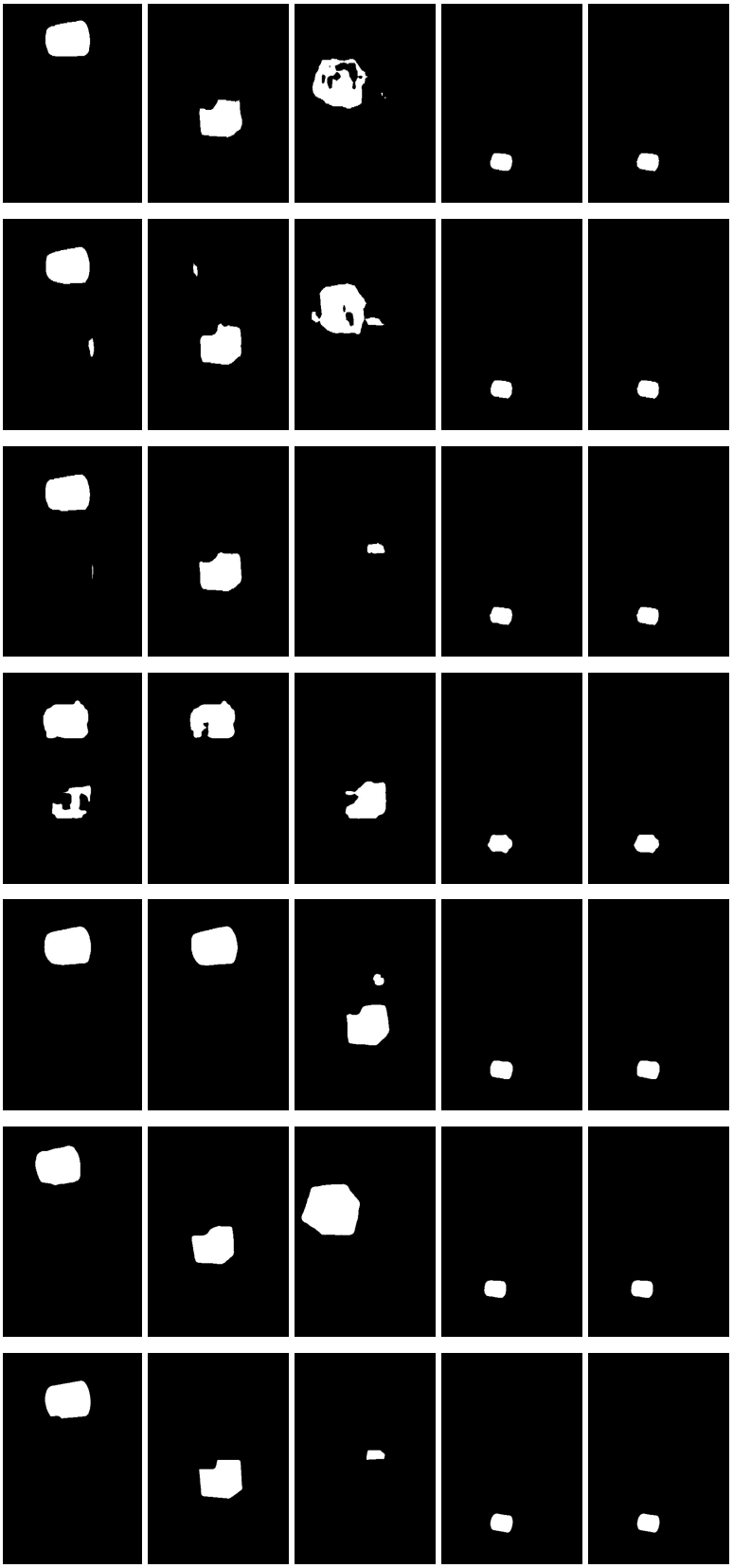


Figure 7.10 – Enlarged Figure 7.8 part B.

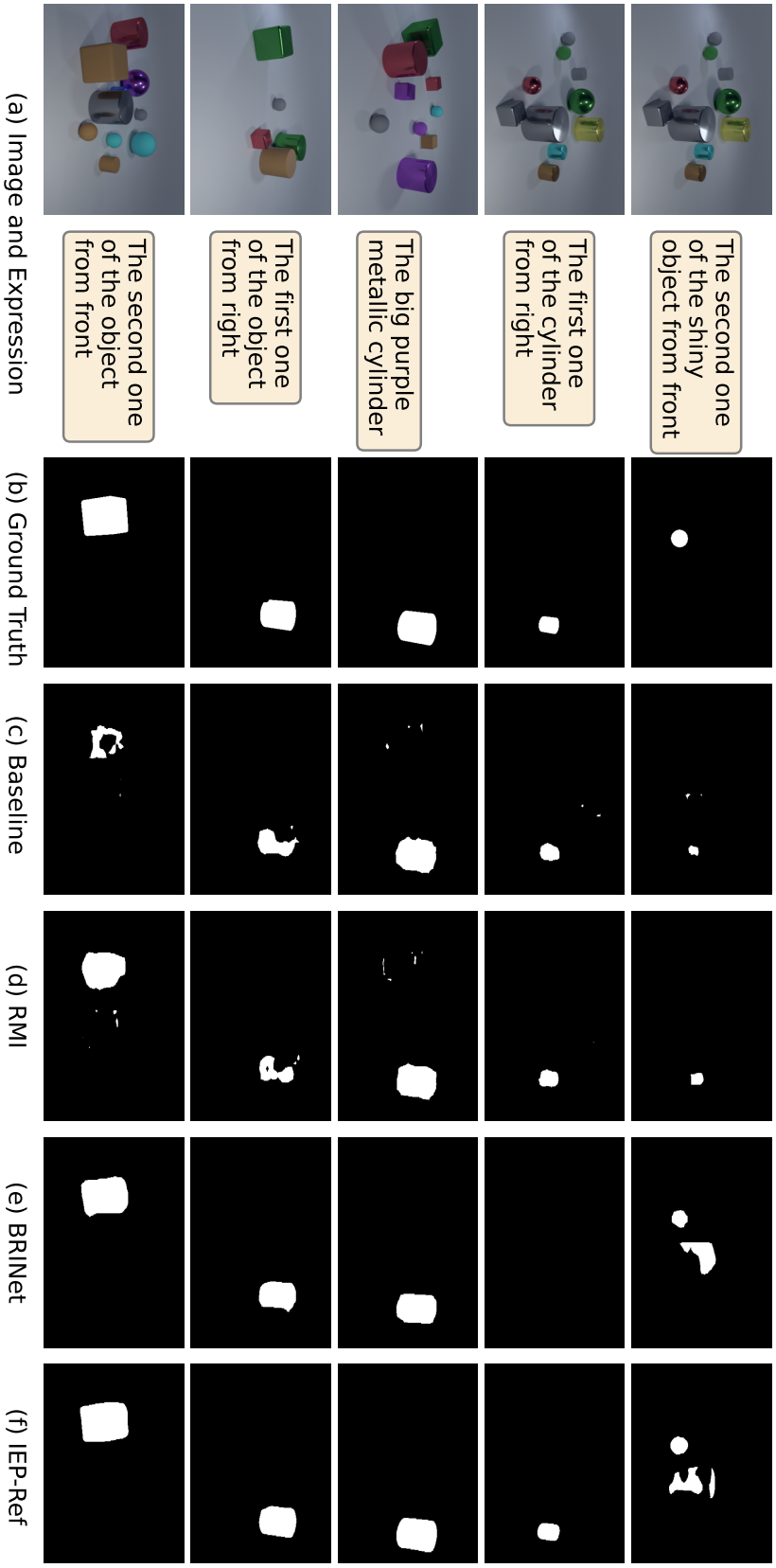


Figure 7.11 – Enlarged Figure 7.8 part C.

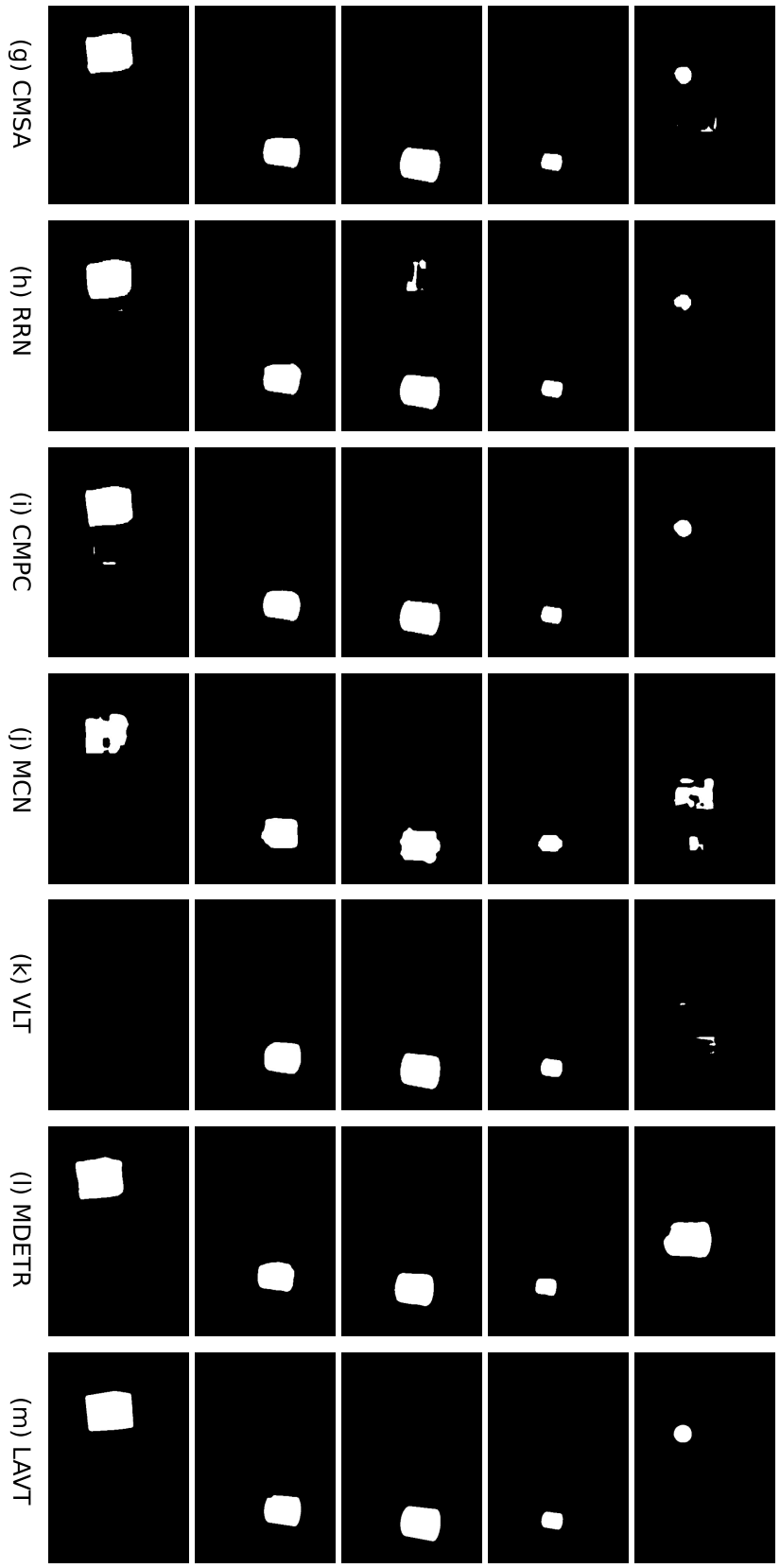


Figure 7.12 – Enlarged Figure 7.8 part D.

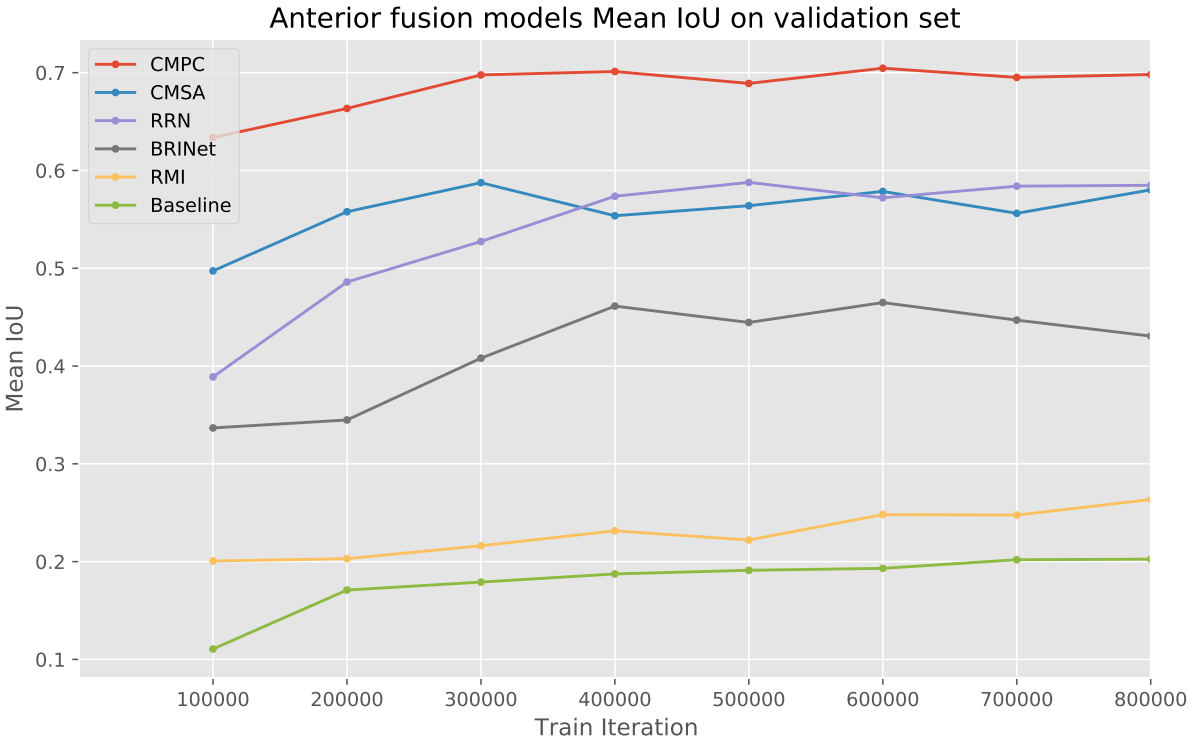


Figure 7.13 – Alongside training iteration, anterior fusion models performance.

(because the recommended coefficient thresholds were not found in the paper). It is also worth noting that MDETR, which employs the attention mechanism in both the language feature and fusion phases, is substantially superior to MCN and VLT, which employ the attention mechanism only in the fusion phase. We have also given the results of MCN and MDETR Bounding Box (BBox) recognition performance in table 7.2.

Table 7.2 – Acc@0.5 represents the Accuracy of BBox IoU bigger than threshold=0.5

	Acc@0.5↑
MCN	0.741430
MDETR	0.755053

Third, we will delve into a discussion of the latest algorithms for multiple fusion structures. The multiple fusion structure method performs the best in this table, *i.e.*, LAVT. Figure 7.6 shows the complex and refined fusion structure of LAVT, which uses multiple branching interactions to combine visual and textual features before and behind the cross-modal attention module. This structure increases the information exchange between different modal features. LAVT is the only model with attention modules for all text, image feature extraction and cross-modal representation. Comparing all models, LAVT achieves the highest performance for RES.

Figure 7.14 gives a plot of all numbers of models' parameters (including trainable or fixed parameters) versus Overall IoU. The comparison of the number of parameters in each model can be compared via the area of the circles in the figure. Overall, the experimental effect is highly correlated with the number of parameters, *i.e.*, the greater the number of parameters, the greater the experimental effect. In terms of text feature extraction, image feature extraction, and cross-modal latent space representation on our database, attention mechanism model works better than RNN or CNN model, respectively.

7.3.4 Multi-View Study

Theoretically, although we should test all 81 cameras, limited by resources, we currently choose the four corners (the farthest distance of camera panning) for comparison experiments, where $I_{u,v} \in \{I_{11}, I_{19}, I_{91}, I_{99}\}$. Figure 7.15 gives three examples of our CLEVR-Remv's multi-view test. Comparing segmentation predictions from various viewpoints reflects the model's ability to extract semantic information from text, images, or cross-modalities. The visualized comparison demonstrates that the attention module can

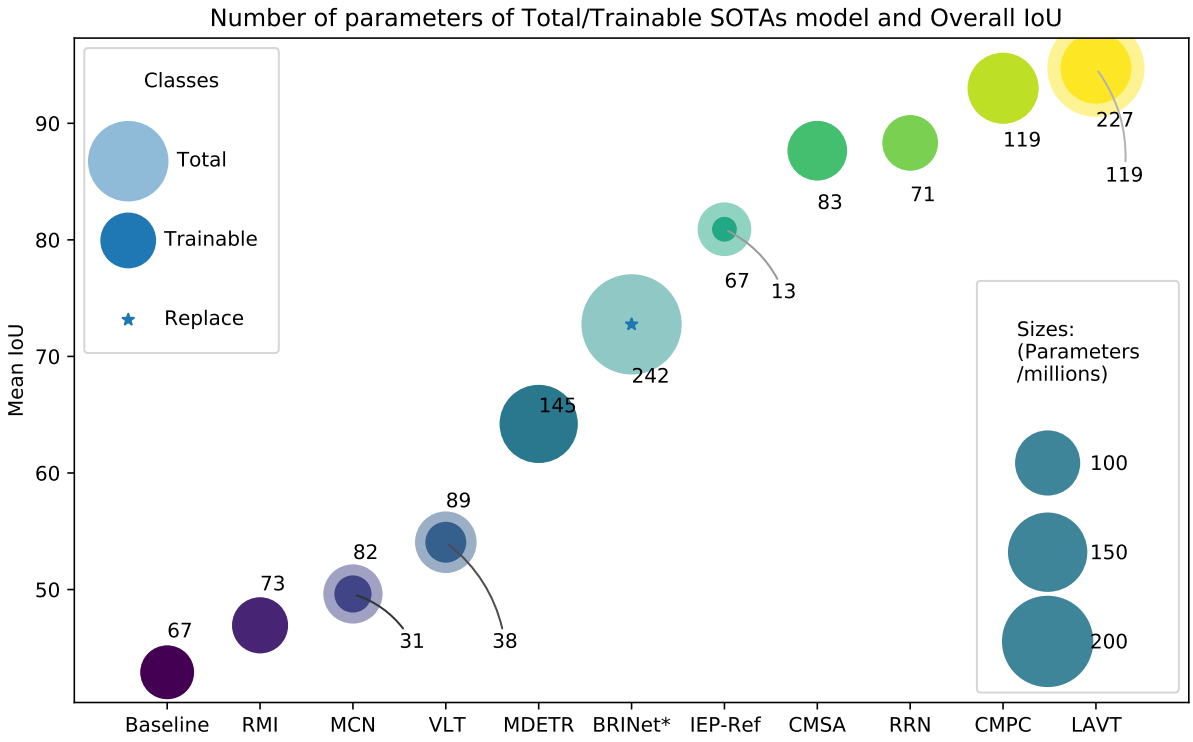
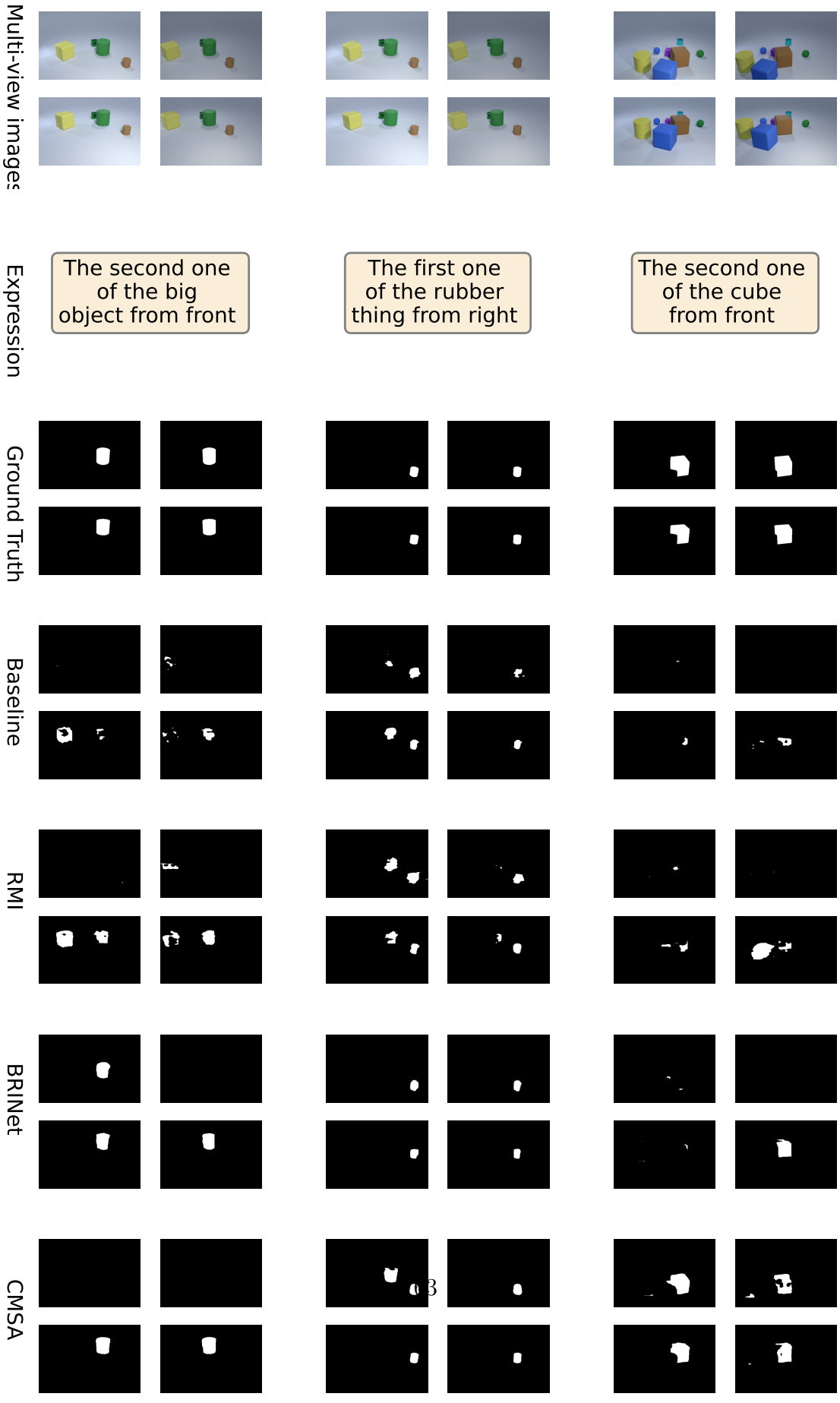


Figure 7.14 – Models performance versus the number of parameters. The publicly released BRINet source code does not work, so we have substituted it with ResNet [32] components. The BRINet* is a replacement for the original BRINet.

significantly improve the RES model's semantic information understanding.

Figure 7.16 and Table 7.3 present the Mean IoU and Overall IoU performance outcomes from the multi-views for the Posterior fusion and Multiple fusion models via visualization and statistical value, respectively. The typical effect of the four corners is indicated by the average value. In addition, the standard deviation represents the extent of the difference between the results of the four viewpoints. By comparing the performance of the central view, the algorithm with better results in the central view will also have better results in the four corners. However, the CMSA model with one attention mechanism outperforms the RRN model, which is without any attention mechanism. LAVT produces the most outstanding results in terms of both the average and standard deviation, increasing the distance between it and competing models. Nevertheless, the average and standard deviation of the four corners does not reflect the model's multi-view shifting robustness. On the one hand, the average does not reflect the magnitude of the drop in the shifted outcome from the center view. On the other hand, the standard deviation does not consider the


Multi-view images
Expression
The second one of the big object from front
The first one of the rubber thing from right
The second one of the cube from front
Ground Truth
Baseline
RMI
BRINet
CMSA

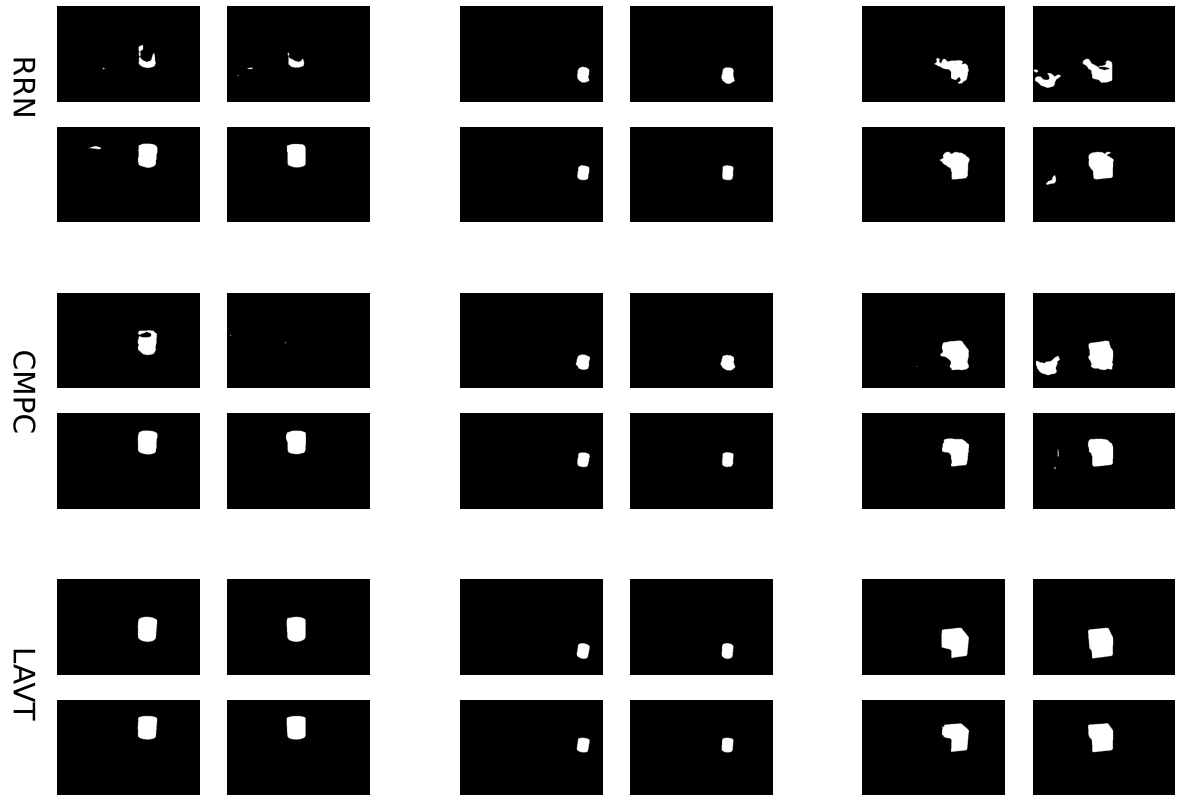


Figure 7.15 – Examples of the our CLEVR-Remv's multi-view test. Four prediction images as a groupe. From the third row of groups to the last row of groups are the prediction of baseline, RMI, BRINet, CMSA, RRN, CMPC, LAVT and ground truth. It is worth noting that the first and second columns of groups are for different objects with corresponding expressions in the same scene, which primarily evaluated the model's ability to process text. The third column of groups is an occluded object, and the shape of the occluded object differs between multi-views. If the models perform well, it indicates that they have a solid ability to extract semantic information, which is the panning capability of models we strive to measure with our proposed MVR (*i.e.*, the Equation 7.3) metric.

size of central view scales.

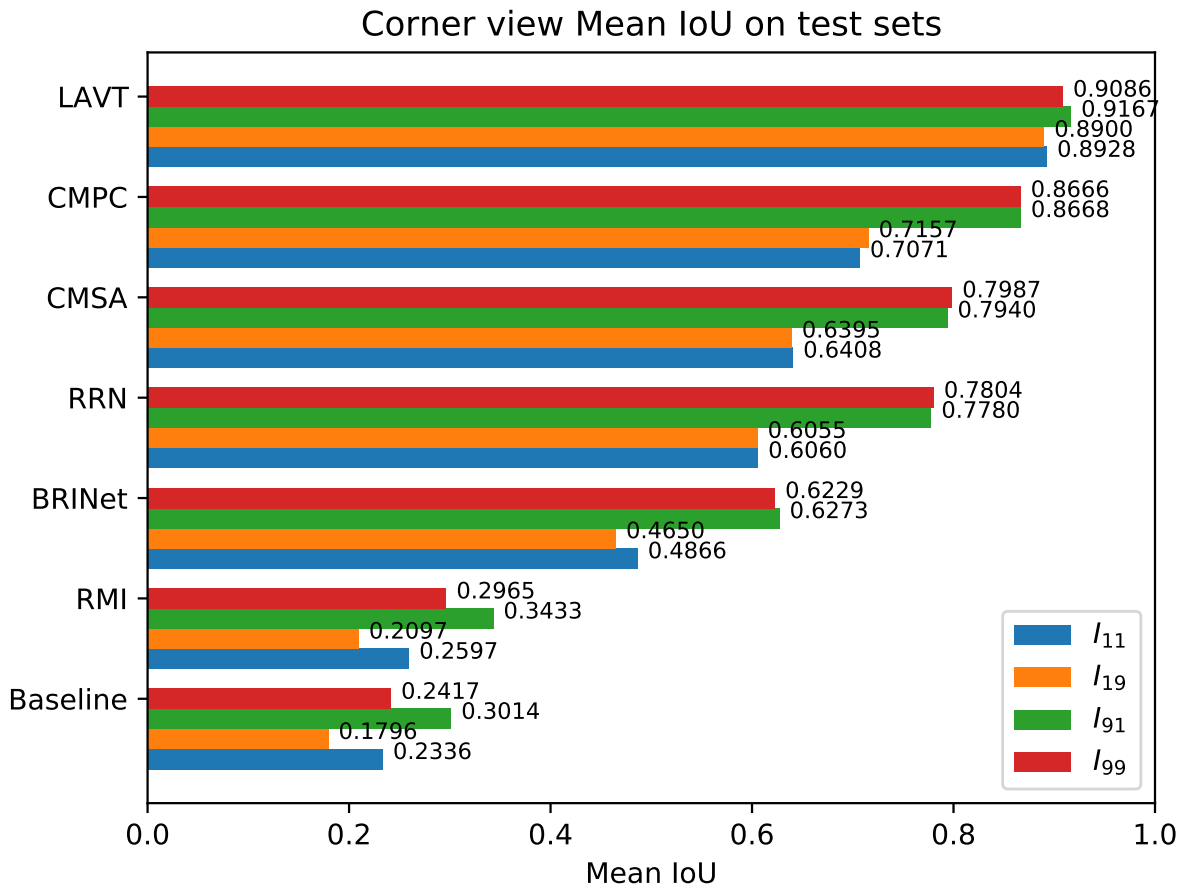


Figure 7.16 – SOTAs RES **Mean IoU** at four light field corner viewpoints.

Table 7.4 displays the MVR metric measurements for each model in our database, while the second column counts the number of portions using the attention mechanism for each model (the division of model portions is in subsection 7.3.2). The LAVT model engaging three attention mechanisms yielded the most significant results, followed by the models employing one attention mechanism and then the models employing none. Although the RRN model using only LSTM and CNN layers achieves better RES results in the central view and the four panning views, the ranking drops below the model using the attention mechanism regarding the robustness of multi-views when the self-scale is considered.

In conclusion, due to the multi-view characteristics of our multimodal database, we design multi-view experiments to compare RES algorithms. In addition, we propose a

Table 7.3 – Four corner viewpoint RES Overall IoU with the results average and Standard Deviation.

	I_{11}	I_{19}	I_{91}	I_{99}	**Average**	**SD**
baseline	0.2873	0.2246	0.3781	0.3171	**0.3017**	**0.0639**
RMI	0.2928	0.2385	0.4069	0.3593	**0.3244**	**0.0740**
BRINet	0.4949	0.4805	0.6579	0.6620	**0.5739**	**0.0996**
RRN	0.5944	0.5933	0.8104	0.8155	**0.7034**	**0.1265**
CMSA	0.6184	0.6169	0.8166	0.8305	**0.7206**	**0.1190**
CMPC	0.6644	0.6619	0.8851	0.8858	**0.7743**	**0.1283**
LAVT	0.8755	0.8945	0.9204	0.9270	**0.9043**	**0.0238**

Table 7.4 – MVR of models. Column **A**-Num counts how many attention mechanisms are used in the textual, visual, or cross-modal fusion portions.

	A-Num	**MVR** $\downarrow$
RMI	0	7.8259%
RRN	0	7.5840%
BRINet	1	7.0202%
CMSA	1	6.6705%
CMPC	1	6.5862%
LAVT	**3**	**2.1996%**

new metric for assessing the robustness of models with various attention mechanisms. The experimental results demonstrate that utilizing the attention module can improve the RES of multi-views, while the attention mechanism benefits the model in achieving better MVR.

7.4 Conclusion

In this chapter, we propose a benchmark and MVR metric to evaluate the stability of models in understanding high-level semantic information. The MVR metrics will help us assess the effectiveness of different models in handling the complexities of RES, such as the pixel-wise labeling of image pixels and the fine-grained semantic text.

Comparing the correlation between experimental outcomes and the number of attention modules employed, one can make the following observations:

Observation 1 The RNN and CNN models can be replaced favourably by the attention module in the low-level feature extraction stage.

Observation 2 Using the attention module in the fusion stage can enhance the MVR metric of the model, suggesting that the attention module may improve the comprehension of high-level semantics in multimodal latent space.

Taking both observations together leads to: Attention model improves the comprehension of multimodal model from low-level to high-level semantic information, and it also enhances the capability to bridge the multimodalities gap. In a word, "attention is all you need even in multimodal RES tasks".

Overall, our work provides a valuable resource for researchers working on referring expression segmentation, as well as a standardized benchmark for evaluating the performance of different models.

Conclusion of Part III

Due to their potential to bridge the gap between different modalities and levels of semantic information, multimodal tasks have been increasingly investigated in terms of interpretability in recent years. Referring expression segmentation (RES) is one of the multimodal tasks with the largest information modality gap, as it requires aligning sequential language information with planar visual information.

We use graphical tools to automatically generate pixel-wise masked images and their corresponding fine-grained text expressions. Our database provides a multi-view perspective to evaluate the high-level semantic comprehension capabilities of multimodal models. We developed a novel metric and benchmark to evaluate the multi-view effectiveness of state-of-the-art models in the RES task, and we also analyzed and summarized the role of the attention module in these models.

Different from the image-text retrieval tasks in part II, which could view the whole images and sentences as a global features, RES are more fine-grained with input data. The RES algorithms must take the regional features into consideration, which means the relation between each component is very important. Thus, more methods choose the cross-modal attention mechanism to bridge the multimodalities gap in this tasks. In this part, we also give a detailed summary of cross-modal attention module within our benchmark.

CONCLUSION

Brief Summary

This thesis focuses on deep learning based multimodal domain, especially image-text retrieval and referring expression segmentation tasks. There are heterogeneous information features that are difficult to express, and feature distances that are difficult to calculate in image-text retrieval domain. In the RES field, there is a lack of fine-grained semantic information database, benchmarks and evaluation indicators. In view of the above problems, this thesis puts forward effective suggestions and experiments. It can be divided into three parts In Part I, we provide an introduction to the fundamentals of multimodality, which are divided into three chapters: computer vision, natural language processing, and multimodal learning. In Part II, we delve into the study of image-text retrieval and present our proposal of utilizing the IMC loss to enhance the performance of pairwise learning. In Part III we introduce a novel multimodal multi-view database that enables us to examine the multi-view stability of RES. Additionally, we establish a new multi-view metric and benchmark to evaluate its performance.

The main contributions of this thesis can be conclude in:

— This thesis provides a comprehensive analysis of the underlying factors contributing to the emergence of multimodality analysis. We emphasize the connections and distinctions between two prominent multimodal tasks: image-text retrieval and referring expression segmentation. Additionally, we present concrete examples to illustrate the practical applications of these tasks.

— The primary focus of the thesis is to identify and address challenges and bottlenecks in computer vision, natural language processing, multimodal fusion, and to bridge the gaps between various modalities.

— Taking a machine learning perspective, we conduct an in-depth review of the SOTA techniques in multimodal learning. We meticulously summarize the advancements in foundational technologies that drive the progress of multimodal learning.

— The thesis proceeds to categorize the SOTA model structures for image-text retrieval. We then perform a detailed analysis and comparison, examining the re-

spective advantages and disadvantages associated with each category.

— We propose a novel multimodal loss function, referred to as IMC Loss, which builds upon the triplet loss methodology. Unlike conventional approaches, IMC Loss not only considers the distances between different modes in the latent space but also incorporates optimization of distances within the same modality. This unique approach contributes to the continuous improvement of the multimodal performance.

— In this thesis, we create a new multimodal multi-view database that can be automatically generated for the referring expression segmentation task.

— A comprehensive analysis is undertaken to examine the SOTA referring expression segmentation models, placing special emphasis on the significant role of multimodal attention mechanisms at these multimodal models.

— We build a benchmark and new metrics to evaluate the performance and robustness of the referring expression segmentation models on our multimodal multi-view database, thus inverting the models' ability to comprehend high-level semantics.

— The thesis provides comprehensive visual representations of the benchmark and metric results, along with visualizations depicting the model parameters and training process. These visualized data aids the interpretation and analysis of multimodal learning.

Our suggestions and experimental methods can be further refined to enhance efficiency and address multimodal retrieval tasks more effectively. Additionally, when combined with algorithms from related fields, they offer significant scalability.

Perspectives

For future multimodal retrieval work, we will further investigate the design of more advanced networks and loss functions for image-text retrieval, aiming to improve performance. By doing so, we seek to improve the effectiveness of the retrieval process for matching images with corresponding textual information. Image-text retrieval encompasses the representation of diverse modal information, aiming to utilize machine learning techniques to minimize the spatial distribution distance between two modal data. In our previous work, we predominantly employed supervised learning methods, specifically the triplet loss, to train network parameters in a supervised manner. However, our future research endeavors involve exploring unsupervised approaches, such as the VAE [140]

method, which automatically minimizes the distribution distance (KL divergence) of distinct modal features in the latent space. We are also interested in leveraging self-supervised learning techniques. Self-supervised learning, originating from BERT, involves masking specific information and learning its contextual role. In the field of vision, a related self-supervised approach known as MAE [141] is employed, which involves cutting out the images into smaller patches and utilizing masks to infer the reconstructed image using the remaining information. However, the use of self-supervised learning in the multimodal domain remains limited. This domain presents a more complex contextual environment, and establishing it through image-text retrieval poses a significant challenge for future exploration.

For future multimodal referring expression segmentation work, we plan to enhance the data modality of our multi-view database for image segmentation by incorporating additional 3D information, such as depth information or dynamic temporal information, thereby extending the data representation into three-dimensional space. This is a new pioneering field, and there are many follow-up tasks that can be carried out on this foundation work. Even though our database is nearly 56K object and expression pairs train, it has not yet fed fully the large cross-modal models. Larger models may perform better if there are more image-text pairs. After increasing the computing capacity, the database will be expanded, and additional models will be evaluated. We do not distinguish between basic logic and complex expression logic in the referring expression, and in future experiments, we will increase the text's length and the modeling environment's complexity. Currently, we only train on the central view of this multi-view perspective and conduct robustness testing on the four peripheral views. In a future study, all eighty multi-view robustnesses will be investigated. We employ a light field camera array to capture scene images but only experiment with the 2D RES model. In future work, we may add depth information and a benchmark for the light field algorithm. Our metrics enable experimental evidence of an improved understanding of attention module in multimodal latent spaces from a multi-view robustness perspective. In the future, additional metrics will demonstrate this conclusion in greater detail.

For multimodality, serving as an essential component, holds a cornerstone position in the quest for general AI. Its scope encompasses a vast range of tasks, and while we have only scratched the surface of a few basic ones, there exists a multitude of broader and

more complex multimodal tasks yet to be explored. The applications of multimodality are incredibly diverse, extending to domains such as mentioned in previous chapter scientific research, medical imaging, and autonomous driving, each of which holds great promise. Nevertheless, it is important to acknowledge that multimodality also presents its own set of challenges that demand attention. As discussed earlier, issues such as data security and environmental sustainability need to be effectively addressed to fully unlock the potential of multimodal systems.

List of publications

Two conference paper:

— Chen, Jianan and Zhang, Lu and Wang, Qiong and Bai, Cong and Kpalma, Kidiyo, « Review of recent deep learning based methods for image-text retrieval », in 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), 2020.

— Chen, Jianan and Zhang, Lu and Wang, Qiong and Bai, Cong and Kpalma, Kidiyo, «Intra-Modal Constraint Loss for Image-Text Retrieval », in 2022 IEEE International Conference on Image Processing (ICIP), 2022.

One under review paper:

— «CLEVR-Remv: A Generative Multimodal Dataset and Multi-View Benchmark for Referring Expression Segmentation», on Conference on Computer Vision and Pattern Recognition, 2024.

Sources primary

[1] P. Shirley, « Fundamentals of computer graphics », 2018.

[2] W. Zhao, D. Zhou, B. Cao, K. Zhang, and J. Chen, « Adversarial modality alignment network for cross-modal molecule retrieval », *IEEE Transactions on Artificial Intelligence*, pp. 1–12, 2023. DOI: 10.1109/TAI.2023.3254518.

[3] F. Wang, X. Liang, L. Xu, and L. Lin, « Unifying relational sentence generation and retrieval for medical image report composition », *IEEE Transactions on Cybernetics*, vol. 52, *6*, pp. 5015–5025, 2022. DOI: 10.1109/TCYB.2020.3026098.

[4] A. Z. Zhu, V. Casser, R. Mahjourian, H. Kretzschmar, and S. Pirk, « Instance segmentation with cross-modal consistency », *in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 2009–2016. DOI: 10.1109/IROS47612.2022.9982285.

[5] C. Zhou, T. Zhang, Y. Wen, L. Chen, L. Zhang, and J. Chen, « Cross-modal guidance for hyperfluorescence segmentation in fundus fluorescein angiography », *in 2021 IEEE International Conference on Multimedia and Expo (ICME)*, 2021, pp. 1–6. DOI: 10.1109/ICME51207.2021.9428085.

[6] C. Janiesch, P. Zschech, and K. Heinrich, « Machine learning and deep learning », *Electronic Markets*, vol. 31, pp. 685–695, 2021.

[7] C. Cortes and V. N. Vapnik, « Support-vector networks », *Machine Learning*, vol. 20, pp. 273–297, 1995.

[8] L. Rosasco, E. D. Vito, A. Caponnetto, M. Piana, and A. Verri, « Are Loss Functions All the Same? », *Neural Computation*, vol. 16, *5*, pp. 1063–1076, May 2004.

[9] B. E. Boser, I. M. Guyon, and V. N. Vapnik, « A training algorithm for optimal margin classifiers », *in Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, New York, NY, USA: Association for Computing Machinery, 1992, pp. 144–152.

[10] J. R. Smith and S.-F. Chang, « Tools and techniques for color image retrieval », *in Electronic imaging*, 1996.

[11] M. A. Stricker and M. Orengo, « Similarity of color images », *in Electronic imaging*, 1995.

[12] G. Pass, R. Zabih, and J. Miller, « Comparing images using color coherence vectors », *in MULTIMEDIA '96*, 1997.

[13] J. Huang, R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, « Image indexing using color correlograms », *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 762–768, 1997.

[14] D. G. Lowe, « Distinctive image features from scale-invariant keypoints », *International journal of computer vision*, vol. 60, *2*, pp. 91–110, 2004.

[15] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, « Surf: speeded up robust features », *in European conference on computer vision*, Springer, 2006, pp. 404–417.

[16] N. Dalal and B. Triggs, « Histograms of oriented gradients for human detection », *in 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, IEEE, vol. 1, 2005, pp. 886–893.

[17] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, « Orb: an efficient alternative to sift or surf », *in 2011 International conference on computer vision*, IEEE, 2011, pp. 2564–2571.

[18] G. Strang, « Wavelet transforms versus fourier transforms », *Bulletin of the American Mathematical Society*, vol. 28, *2*, pp. 288–305, 1993.

[19] J. Canny, « A computational approach to edge detection », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, *6*, pp. 679–698, 1986. DOI: 10.1109/TPAMI.1986.4767851.

[20] I. Sobel and G. M. Feldman, « An isotropic 3×3 image gradient operator », 1990.

[21] T. Zhang and C. Suen, « A fast parallel algorithm for thinning digital patterns », *Communications of the ACM*, vol. 27, *3*, pp. 236–239, 1984.

[22] Z. Guo and R. W. Hall, « Parallel thinning with two-subiteration algorithms », *Communications of the ACM*, vol. 32, pp. 359–373, 1989.

[23] S. Suzuki and K. Abe, « Topological structural analysis of digitized binary images by border following », *Comput. Vis. Graph. Image Process.*, vol. 30, pp. 32–46, 1985.

[24] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, 4. Springer, 2006, vol. 4.

[25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, « Learning representations by back-propagating errors », *Nature*, vol. 323, pp. 533–536, 1986.

[27] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, « Handwritten digit recognition with a back-propagation network », *in NIPS*, 1989.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, « Gradient-based learning applied to document recognition », *Proc. IEEE*, vol. 86, pp. 2278–2324, 1998.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, « Imagenet classification with deep convolutional neural networks », *Commun. ACM*, vol. 60, *6*, pp. 84–90, May 2017, ISSN: 0001-0782. DOI: 10.1145/3065386. [Online]. Available: https://doi.org/10.1145/3065386.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, « Imagenet large scale visual recognition challenge », *International journal of computer vision*, vol. 115, pp. 211–252, 2015.

[31] K. Simonyan and A. Zisserman, « Very deep convolutional networks for large-scale image recognition », *CoRR*, vol. abs/1409.1556, 2014.

[32] K. He, X. Zhang, S. Ren, and J. Sun, « Deep residual learning for image recognition », *in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[33] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, « Densely connected convolutional networks », *in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269. DOI: 10.1109/CVPR.2017.243.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, « Attention is all you need », *Advances in neural information processing systems*, vol. 30, 2017.

[35] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, « End-to-end object detection with transformers », *in Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Cham: Springer International Publishing, 2020, pp. 213–229, ISBN: 978-3-030-58452-8.

[36] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, « An image is worth 16x16 words: transformers for image recognition at scale », *in International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy.

[37] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, « Swin transformer: hierarchical vision transformer using shifted windows », *in 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9992–10 002. DOI: 10.1109/ICCV48922.2021.00986.

[38] M. T. Pilehvar and J. Camacho-Collados, « Embeddings in natural language processing: theory and advances in vector representations of meaning », *Synthesis Lectures on Human Language Technologies*, 2020.

[39] Y. Bengio, R. Ducharme, and P. Vincent, « A neural probabilistic language model », *in Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp, Eds., vol. 13, MIT Press, 2000. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2000/file/728f206c2a01bf572b5940d7d9a8fa4Paper.pdf.

[40] D. Jurafsky and J. H. Martin, « N-gram language models », *Speech and Language Processing (3rd ed.)*, 2022.

[41] P. Gagniuc, *Markov Chains: From Theory to Implementation and Experimentation*. John Wiley & Sons, 2017, ISBN: 9781119387596. [Online]. Available: https://books.google.fr/books?id=2chjtAEACAAJ.

[42] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, « Learning phrase representations using rnn encoder-decoder for statistical machine translation », *in Conference on Empirical Methods in Natural Language Processing*, 2014.

[43] S. Hochreiter and J. Schmidhuber, « Long short-term memory », *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[44] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, « Efficient estimation of word representations in vector space », *in International Conference on Learning Representations*, 2013.

[45] T. Mikolov, Q. V. Le, and I. Sutskever, « Exploiting similarities among languages for machine translation », *ArXiv*, vol. abs/1309.4168, 2013.

[46] D. Bahdanau, K. Cho, and Y. Bengio, « Neural machine translation by jointly learning to align and translate », *CoRR*, vol. abs/1409.0473, 2014.

[47] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, « BERT: pre-training of deep bidirectional transformers for language understanding », *in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. [Online]. Available: https://aclanthology.org/N19-1423.

[48] A. Williams, N. Nangia, and S. R. Bowman, « A broad-coverage challenge corpus for sentence understanding through inference », *in North American Chapter of the Association for Computational Linguistics*, 2017.

[49] E. F. Tjong Kim Sang and F. De Meulder, « Introduction to the CoNLL-2003 shared task: language-independent named entity recognition », *in Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 2003, pp. 142–147. [Online]. Available: https://aclanthology.org/W03-0419.

[50] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, « Squad: 100,000+ questions for machine comprehension of text », *ArXiv*, vol. abs/1606.05250, 2016.

[51] Y. Zhang and H. Lu, « Deep cross-modal projection learning for image-text matching », *in Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., Cham: Springer International Publishing, 2018, pp. 707–723, ISBN: 978-3-030-01246-5.

[52] A. Karpathy and L. Fei-Fei, « Deep visual-semantic alignments for generating image descriptions », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, *4*, pp. 664–676, 2017. DOI: 10.1109/TPAMI.2016.2598339.

[53] R. Hadsell, S. Chopra, and Y. LeCun, « Dimensionality reduction by learning an invariant mapping », *in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2006, pp. 1735–1742. DOI: 10.1109/CVPR.2006.100.

[54] F. Schroff, D. Kalenichenko, and J. Philbin, « Facenet: a unified embedding for face recognition and clustering », *in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682.

[55] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, « Vse++: improving visual-semantic embeddings with hard negatives », *in Proceedings of the British Machine Vision Conference (BMVC)*, 2018. [Online]. Available: https://github.com/fartashf/vsepp.

[56] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, « Learning transferable visual models from natural language supervision », *in Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 18–24 Jul 2021, pp. 8748–8763. [Online]. Available: https://proceedings.mlr.press/v139/radford21a.html.

[140] D. P. Kingma and M. Welling, « Auto-encoding variational bayes », *CoRR*, vol. abs/1312.6114, 2013.

[141] K. He, X. Chen, S. Xie, Y. Li, P. Doll'ar, and R. B. Girshick, « Masked autoencoders are scalable vision learners », *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15 979–15 988, 2021.

[142] H. Freeman, « On the encoding of arbitrary geometric configurations », *IRE Trans. Electron. Comput.*, vol. 10, pp. 260–268, 1961.

Sources secondary

[58] A. F. Biten, L. Gomez, M. Rusinol, and D. Karatzas, « Good news, everyone! context driven entity-aware captioning for news images », *in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[59] Z. Fan, Z. Wei, S. Wang, and X.-J. Huang, « Bridging by word: image grounded vocabulary construction for visual captioning », *in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 6514–6524.

[61] X. Huang and Y. Peng, « Deep cross-media knowledge transfer », *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8837–8846.

[62] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, « Attngan: fine-grained text to image generation with attentional generative adversarial networks », *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1316–1324.

[63] T. Qiao, J. Zhang, D. Xu, and D. Tao, « Mirrorgan: learning text-to-image generation by redescription », *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1505–1514.

[64] W. Li, P. Zhang, L. Zhang, Q. Huang, X. He, S. Lyu, and J. Gao, « Object-driven text-to-image synthesis via adversarial training », *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 174–12 182.

[65] J. Liu, C. Xu, and H. Lu, « Cross-media retrieval: state-of-the-art and open issues », *Int. J. of Multimedia Intelligence and Security*, vol. 1, pp. 33–52, Jan. 2010. DOI: 10.1504/IJMIS.2010.035970.

[69] Y. Zhang and H. Lu, « Deep cross-modal projection learning for image-text matching », *in The European Conference on Computer Vision (ECCV)*, Sep. 2018.

[71] L. Zhen, P. Hu, X. Wang, and D. Peng, « Deep supervised cross-modal retrieval », *in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[73] F. Liu and R. Ye, « A strong and robust baseline for text-image matching », *in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 169–176. DOI: 10.18653/v1/P19-2023. [Online]. Available: https://www.aclweb.org/anthology/P19-2023.

[74] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, « Generative adversarial nets », *in Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[76] N. Sarafianos, X. Xu, and I. A. Kakadiaris, « Adversarial representation learning for text-to-image matching », *in Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5814–5824.

[78] Z. Wang, X. Liu, H. Li, L. Sheng, J. Yan, X. Wang, and J. Shao, « Camp: cross-modal adaptive message passing for text-image retrieval », *in The IEEE International Conference on Computer Vision (ICCV)*, Oct. 2019.

[79] B. Zhu, C.-W. Ngo, J. Chen, and Y. Hao, « R2gan: cross-modal recipe retrieval with generative adversarial network », *in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[81] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, « Canonical correlation analysis: an overview with application to learning methods », *Neural computation*, vol. 16, *12*, pp. 2639–2664, 2004.

[82] Y. Wang, H. Yang, X. Qian, L. Ma, J. Lu, B. Li, and X. Fan, « Position focused attention network for image-text matching », *in Proceedings of the 28th International Joint Conference on Artificial Intelligence*, ser. IJCAI'19, Macao, China: AAAI Press, 2019, pp. 3792–3798, ISBN: 978-0-9992411-4-1. [Online]. Available: http://dl.acm.org/citation.cfm?id=3367471.3367568.

[84] Z. Ji, Y. Sun, Y. Yu, Y. Pang, and J. Han, « Attribute-guided network for cross-modal zero-shot hashing », *IEEE transactions on neural networks and learning systems*, 2019.

[85] Y. Huang and L. Wang, « Acmm: aligned cross-modal memory for few-shot image and sentence matching », *in The IEEE International Conference on Computer Vision (ICCV)*, Oct. 2019.

[87] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He, « Stacked cross attention for image-text matching », *in The European Conference on Computer Vision (ECCV)*, Sep. 2018.

[90] Y. Huang, Y. Long, and L. Wang, « Few-shot image and sentence matching via gated visual-semantic embedding », *in Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8489–8496.

[91] S. Li, T. Xiao, H. Li, B. Zhou, D. Yue, and X. Wang, « Person search with natural language description », *in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017.

[93] J. Marin, A. Biswas, F. Ofli, N. Hynes, A. Salvador, Y. Aytar, I. Weber, and A. Torralba, « Recipe1m+: a dataset for learning cross-modal embeddings for cooking recipes and food images », *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[94] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Torralba, « Learning cross-modal embeddings for cooking recipes and food images », *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[95] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, « Flickr30k entities: collecting region-to-phrase correspondences for richer image-to-sentence models », *in 2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2641–2649. DOI: 10.1109/ICCV.2015.303.

[96] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, « Microsoft coco: common objects in context », *in Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 740–755, ISBN: 978-3-319-10602-1.

Sources teriary

[97] J. Chen, L. Zhang, C. Bai, and K. Kpalma, « Review of recent deep learning based methods for image-text retrieval », *in 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2020, pp. 167–172. DOI: 10.1109/MIPR49039.2020.00042.

[98] J. Pennington, R. Socher, and C. D. Manning, « Glove: global vectors for word representation », *in Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162.

[99] X. Glorot and Y. Bengio, « Understanding the difficulty of training deep feedforward neural networks », *in Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[100] D. P. Kingma and J. Ba, « Adam: a method for stochastic optimization », *in International Conference on Learning Representations*, 2015.

[101] B. Klein, G. Lev, G. Sadeh, and L. Wolf, « Associating neural word embeddings with deep image representations using fisher vectors », *in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4437–4446. DOI: 10.1109/CVPR.2015.7299073.

[102] X. Lin and D. Parikh, « Leveraging visual question answering for image-caption ranking », *in Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham: Springer International Publishing, 2016, pp. 261–277, ISBN: 978-3-319-46475-6.

[103] L. Zhang, M. Luo, J. Liu, X. Chang, Y. Yang, and A. G. Hauptmann, « Deep top-k ranking for image–sentence matching », *IEEE Transactions on Multimedia*, vol. 22, *3*, pp. 775–785, 2020. DOI: 10.1109/TMM.2019.2931352.

[104] Z. Zheng, L. Zheng, M. Garrett, Y. Yang, M. Xu, and Y.-D. Shen, « Dual-path convolutional image-text embeddings with instance loss », *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, *2*, pp. 1–23, 2020.

[105] F. Liu and R. Ye, « A strong and robust baseline for text-image matching », *in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, 2019, pp. 169–176.

[106] F. Huang, X. Zhang, Z. Zhao, and Z. Li, « Bi-directional spatial-semantic attention networks for image-text matching », *IEEE Transactions on Image Processing*, vol. 28, *4*, pp. 2008–2020, 2019. DOI: 10.1109/TIP.2018.2882225.

[107] Y. Huang, Q. Wu, C. Song, and L. Wang, « Learning semantic concepts and order for image and sentence matching », *in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6163–6171. DOI: 10.1109/CVPR.2018.00645.

[108] M. Zhou, Z. Niu, L. Wang, Z. Gao, Q. Zhang, and G. Hua, « Ladder loss for coherent visual-semantic embedding », *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, *07*, pp. 13 050–13 057, Apr. 2020. DOI: 10.1609/aaai.v34i07.7006. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/7006.

[109] Y. Chen, R. Huang, H. Chang, C. Tan, T. Xue, and B. Ma, « Cross-modal knowledge adaptation for language-based person search », *IEEE Transactions on Image Processing*, vol. 30, pp. 4057–4069, 2021. DOI: 10.1109/TIP.2021.3068825.

[110] J. Chen, L. Zhang, Q. Wang, C. Bai, and K. Kpalma, « Intra-modal constraint loss for image-text retrieval », *in 2022 IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 4023–4027. DOI: 10.1109/ICIP46576.2022.9897195.

[143] C. Liu, Z. Mao, W. Zang, and B. Wang, « A neighbor-aware approach for image-text matching », *in ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3970–3974. DOI: 10.1109/ICASSP.2019.8683869.

Sources quaternary

[111] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy, « Generation and comprehension of unambiguous object descriptions », *in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 11–20. DOI: 10.1109/CVPR.2016.9.

[112] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, « ReferItGame: referring to objects in photographs of natural scenes », *in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 787–798. DOI: 10.3115/v1/D14-1086. [Online]. Available: https://aclanthology.org/D14-1086.

[113] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg, « Modeling context in referring expressions », *ArXiv*, vol. abs/1608.00272, 2016.

[114] B. O. Community, *Blender - a 3d modelling and rendering package*, version 2.83LTS, Stichting Blender Foundation, Amsterdam: Blender Foundation. [Online]. Available: http://www.blender.org.

[115] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick, « Clevr: a diagnostic dataset for compositional language and elementary visual reasoning », *in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1988–1997. DOI: 10.1109/CVPR.2017.215.

[116] R. Liu, C. Liu, Y. Bai, and A. L. Yuille, « CLEVR-Ref+: Diagnosing Visual Reasoning With Referring Expressions », *in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 4180–4189, ISBN: 978-1-72813-293-8. DOI: 10.1109/CVPR.2019.00431. [Online]. Available: https://ieeexplore.ieee.org/document/8954348/ (visited on 10/13/2022).

[117] L. Karazija, I. Laina, and C. Rupprecht, « Clevrtex: a texture-rich benchmark for unsupervised multi-object segmentation », *in Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, J. Vanschoren and S. Yeung, Eds., vol. 1, 2021. [Online]. Available: https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/file/e2c420d928d4bf8ce0ff2ec19b371514-Paper-round2.pdf.

[118] L. Salewski, A. S. Koepke, H. P. A. Lensch, and Z. Akata, « Clevr-x: a visual reasoning dataset for natural language explanations », *in xxAI - Beyond Explainable AI: International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers*, A. Holzinger, R. Goebel, R. Fong, T. Moon, K.-R. Müller, and W. Samek, Eds. Cham: Springer International Publishing, 2022, pp. 69–88, ISBN: 978-3-031-04083-2. DOI: 10.1007/978-3-031-04083-2_5. [Online]. Available: https://doi.org/10.1007/978-3-031-04083-2_5.

[119] Z. Chen, P. Wang, L. Ma, K.-Y. K. Wong, and Q. Wu, « Cops-ref: a new dataset and task on compositional referring expression comprehension », *in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 083–10 092. DOI: 10.1109/CVPR42600.2020.01010.

[120] A. Alvarez-Gila, J. Van De Weijer, Y. Wang, and E. Garrote, « Mvmo: a multi-object dataset for wide baseline multi-view semantic segmentation », *in 2022 IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 1166–1170. DOI: 10.1109/ICIP46576.2022.9897955.

[121] H. Sheng, R. Cong, D. Yang, R. Chen, S. Wang, and Z. Cui, « Urbanlf: a comprehensive light field dataset for semantic segmentation of urban scenes », *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, pp. 7880–7893, 2022.

[122] Y. Li, L. Zhang, Q. Wang, and G. Lafruit, « Manet: multi-scale aggregated network for light field depth estimation », *in ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 1998–2002. DOI: 10.1109/ICASSP40776.2020.9053532.

[123] C. Liu, Z. Lin, X. Shen, J. Yang, X. Lu, and A. Yuille, « Recurrent multimodal interaction for referring image segmentation », *in Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017.

[124] R. Li, K. Li, Y.-C. Kuo, M. Shu, X. Qi, X. Shen, and J. Jia, « Referring image segmentation via recurrent refinement networks », *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

[125] Z. Hu, G. Feng, J. Sun, L. Zhang, and H. Lu, « Bi-directional relationship inferring network for referring image segmentation », *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[126] L. Ye, M. Rochan, Z. Liu, and Y. Wang, « Cross-modal self-attention network for referring image segmentation », *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[127] S. Huang, T. Hui, S. Liu, G. Li, Y. Wei, J. Han, L. Liu, and B. Li, « Referring image segmentation via cross-modal progressive comprehension », *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10 485–10 494, 2020.

[128] L.-C. Chen, G. Papandreou, I. Kokkinos, K. P. Murphy, and A. L. Yuille, « Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 834–848, 2016.

[129] R. Hu, M. Rohrbach, and T. Darrell, « Segmentation from natural language expressions », *in Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham: Springer International Publishing, 2016, pp. 108–124, ISBN: 978-3-319-46448-0.

[130] G. Luo, Y. Zhou, X. Sun, L. Cao, C. Wu, C. Deng, and R. Ji, « Multi-task collaborative network for joint referring expression comprehension and segmentation », *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10 031–10 040, 2020.

[131] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, « Empirical evaluation of gated recurrent neural networks on sequence modeling », *ArXiv*, vol. abs/1412.3555, 2014.

[132] H. Ding, C. Liu, S. Wang, and X. Jiang, « Vision-language transformer and query generation for referring segmentation », *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16 301–16 310, 2021.

[133] A. Kamath, M. Singh, Y. LeCun, I. Misra, G. Synnaeve, and N. Carion, « Mdetr - modulated detection for end-to-end multi-modal understanding », *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1760–1770, 2021.

[134] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, « Bert: pre-training of deep bidirectional transformers for language understanding », *ArXiv*, vol. abs/1810.04805, 2019.

[135] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, « End-to-end object detection with transformers », *in Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Cham: Springer International Publishing, 2020, pp. 213–229, ISBN: 978-3-030-58452-8.

[136] Z. Yang, J. Wang, Y. Tang, K. Chen, H. Zhao, and P. H. S. Torr, « Lavt: language-aware vision transformer for referring image segmentation », *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18 134–18 144, 2021.

[137] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Y. Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, *TensorFlow: large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: `https://www.tensorflow.org/`.

[138] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, « Pytorch: an imperative style, high-performance deep learning library », *in Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[139] P. Krähenbühl and V. Koltun, « Efficient inference in fully connected crfs with gaussian edge potentials », *in NIPS*, 2011.

COLLEGE DOCTORAL BRETAGNE
MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

Titre : Recherche multimodale basée sur l'apprentissage profond

Mot clés : intelligence artificielle, apprentissage profond, recherche multimodale, ensemble de données multimodal

Résumé : Les tâches multimodales jouent un rôle crucial dans la progression vers l'atteinte de l'intelligence artificielle (IA) générale. L'objectif principal de la recherche multimodale est d'exploiter des algorithmes d'apprentissage automatique pour extraire des informations sémantiques pertinentes, en comblant le fossé entre différentes modalités telles que les images visuelles, le texte linguistique et d'autres sources de données. L'entropie de l'information associée à des données diverses pour la même sémantique de haut niveau varie considérablement, présentant un défi substantiel pour les modèles multimodaux.

Les modèles de réseau multimodal basés sur l'apprentissage profond offrent une solution efficace pour relever les difficultés découlant des différences substantielles d'entropie de l'information. Ces modèles présentent une précision et une stabilité impressionnantes dans les tâches d'appariement d'informations multimodales à grande échelle, comme la recherche d'images et de textes.

Dans nos recherches, nous développons une nouvelle base de données multimodale et multi-vues générative spécifiquement conçue pour la tâche de segmentation référentielle multimodale. De plus, nous établissons une référence de pointe (SOTA) pour les modèles de segmentation d'expressions référentielles dans le domaine multimodal. Les résultats de nos expériences comparatives sont présentés de manière visuelle, offrant des informations claires et complètes.

Title: Deep Learning Based Multimodal Retrieval

Keywords: artificial intelligence, deep learning, multimodal retrieval, multimodal dataset

Abstract: Multimodal tasks play a crucial role in the progression towards achieving general artificial intelligence (AI). The primary goal of multimodal retrieval is to employ machine learning algorithms to extract relevant semantic information, bridging the gap between different modalities such as visual images, linguistic text, and other data sources. The information entropy linked to diverse data for the same high-level semantics varies notably, presenting a substantial challenge for multimodal models.

Deep learning-based multimodal network models provide an effective solution to tackle the difficulties arising from substantial differences in information entropy. These models exhibit impressive accuracy and stability in large-scale cross-modal information matching tasks, such as image-text retrieval.

In our research, we develop a novel generative multimodal multi-view database specifically designed for the multimodal referential segmentation task. Additionally, we establish a state-of-the-art (SOTA) benchmark and multi-view metric for referring expression segmentation models in the multimodal domain. The results of our comparative experiments are presented visually, providing clear and comprehensive insights.