Palestine Polytechnic University

Deanship of Graduate Studies and Scientific Research

Master of Intelligent Systems

Training-Free Sketch-Based Image Retrieval Based on Multimodal Feature Fusion and Image Generation

By

Ma'ali Al-Jabari

Supervisor

Dr. Alaa Halawani

Thesis submitted in partial fulfillment of requirements of the degree
Master of Science in Intelligent Systems

July 5, 2026

The undersigned hereby certify that they have read, examined, and recommended to the Deanship of Graduate Studies and Scientific Research at Palestine Polytechnic University the approval of a thesis entitled ***"Training-Free Sketch-Based Image Retrieval Based on Multimodal Feature Fusion and Image Generation"*** submitted by **Ma'ali Al-Jabari** in partial fulfillment of the requirements for the degree of Master of Intelligent Systems.

Graduate Advisory Committee:

Dr. Alaa Halawani (Supervisor), Palestine Polytechnic University.

Signature: _____ Date: _____

Dr. Zein Salah (Internal Committee Member), Palestine Polytechnic University.

Signature: _____ Date: _____

Dr. Abualsoud Hanani (External Committee Member), Birzeit University.

Signature: _____ Date: _____

Thesis Approved

Prof. Mahmoud AlHaddad
Dean of Graduate Studies and Scientific Research
Palestine Polytechnic University

Signature: _____ Date: _____

DECLARATION

I declare that the Master Thesis entitled ***"Training-Free Sketch-Based Image Retrieval Based on Multimodal Feature Fusion and Image Generation"*** is my original work and hereby certify that unless stated, all work contained within this thesis is my independent research and has not been submitted for the award of any other degree at any institution, except where due acknowledgment is made in the text.

Ma'ali Al-Jabari

Signature: _____ Date: _____

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for the Master's degree of Intelligent Systems at Palestine Polytechnic University, I agree that the library shall make it available to borrowers under rules of the library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of the source is made. Permission for extensive quotation from, reproduction, or publication of this thesis may be granted by my main supervisor, or in his absence, by the Dean of Graduate Studies and Scientific Research when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Ma'ali Al-Jabari

Signature: _____ Date: _____

Dedication

To the love of my life, my partner in every storm and every sunrise,
***Yousef Salah**,*

you were my calm when everything felt overwhelming,
my strength when mine ran out, and my home when I felt lost.
No words are enough — this is for you, as much as it is for me.

To the lights of my life —
*my beloved sons, **Kareem** and **Taim**, and my precious daughter, **Alma**,*

every page of this work was written with your smiles in my heart.
You are my greatest blessing, my deepest joy,
and the reason behind every effort I make.

*To my beloved **parents**,*

who planted in me the love of knowledge before I could even read,
whose prayers wrapped around me like a shield,
and whose sacrifices I can never fully repay.

*And to my dear **sisters and brothers**,*
for every word of encouragement, every warm smile,
and for always making me feel I was never alone.

*To my respected supervisor, **Dr. Alaa Halawani**,*

for his patience, wisdom, and the trust he placed in me
from the very first day to the very last.

*To my dear **friends**,*

who made the hard days lighter and the good days brighter.

*"**The secret of getting ahead is getting started.**"*
— Mark Twain

Acknowledgements

This thesis would not have been possible without the support and guidance of many people to whom I am deeply grateful.

My sincerest thanks go to my supervisor, **Dr. Alaa Halawani**, for his thoughtful guidance, precise feedback, and unwavering encouragement throughout every stage of this work. I am truly fortunate to have had a supervisor who believed in this research as much as I did.

I am grateful to **Palestine Polytechnic University**, the Deanship of Graduate Studies and Scientific Research, and the **College of Information Technology and Computer Engineering** for providing the academic environment and computing resources that made this research possible.

I extend my sincere thanks to the thesis committee members — internal examiner **Dr. Zein Salah** and external examiner **Dr. Abualsoud Hanani** — for their valuable feedback and constructive observations.

To my husband, **Eng. Yousef Salah**, my children **Kareem**, **Alma**, and **Taim**, and my parents, sisters, and brothers — thank you for your patience, prayers, and love that carried me through every difficult moment.

Finally, to my colleagues and friends who made this journey feel less lonely — your presence made all the difference.

Abstract

Sketch-Based Image Retrieval (SBIR) aims to retrieve relevant photographs from a large database using a freehand sketch as the query. The core challenge is the domain gap between abstract sketches and natural photographs, which causes most existing approaches to depend on task-specific training on labelled sketch–photo pairs, limiting their generalisability across datasets and categories.

This thesis proposes a training-free SBIR framework that combines three pretrained models — CLIP, BLIP, and stable diffusion with ControlNet — without any fine-tuning. Two retrieval paradigms are explored and evaluated across four benchmark datasets: Sketchy, Extended Sketchy, TU-Berlin, and QuickDraw. In the *direct retrieval* paradigm, the query is represented by a weighted combination of CLIP visual features, BLIP-generated captions, and CLIP-predicted class labels, explored across seven fixed ablation configurations and a dataset-specific parameter search. In the *generation-based* paradigm, the sketch is first converted into a realistic image using stable diffusion conditioned on either a BLIP caption or a CLIP-predicted class label, and the generated image is used as the retrieval query.

The key finding is that text-based signals — class labels and captions — consistently outperform raw sketch visual features for category-level retrieval. Because CLIP was not trained on sketch images, text descriptions bridge the domain gap more reliably than visual matching. The optimal feature combination is dataset-dependent: highly abstract datasets such as QuickDraw benefit most from class labels alone, while richer datasets gain from incorporating captions. The generation-based approach consistently underperforms direct retrieval due to the domain shift between generated and real photographs and error propagation through the generation pipeline.

The proposed framework outperforms all four evaluated trained state-of-the-art methods on QuickDraw (mAP@All = 0.437 vs. best trained 0.231, +89.2%), achieves the highest mAP@200 on Extended Sketchy Split 2 (0.761), and matches the best trained method on TU-Berlin P@100 within 0.4% — all without any task-specific training, demonstrating that competitive zero-shot SBIR is achievable using pretrained models alone.

Keywords: Sketch-Based Image Retrieval, Zero-Shot Retrieval, CLIP, BLIP, Stable Diffusion, ControlNet, Feature Fusion, Multimodal Learning, Training-Free Retrieval

ملخص

يهدف استرجاع الصور القائم على الرسم (SBIR) إلى استرجاع الصور ذات الصلة من قاعدة بيانات كبيرة باستخدام رسمٍ يدوي حر كاستعلام. يتمثّل التحدي الجوهري في الفجوة بين النطاقين، أي الاختلاف البصري الكبير بين الرسومات التجريدية والصور الفوتوغرافية الطبيعية، مما يدفع معظم الأساليب الحالية إلى الاعتماد على تدريب نماذج متخصصة على أزواج مصنّفة من الرسومات والصور، وهو ما يُقيّد قابليتها للتعميم عبر مجموعات البيانات والفئات المختلفة.

تقترح هذه الرسالة إطار عمل لاسترجاع الصور القائم على الرسم دون الحاجة إلى أي تدريب، يجمع بين ثلاثة نماذج مدربة مسبقاً هي CLIP وBLIP وStable Diffusion مع ControlNet، دون أي ضبط دقيق. يتم استكشاف نموذجَي استرجاع وتقييمهما عبر أربع مجموعات بيانات معيارية: Sketchy وExtended Sketchy وTU-Berlin وQuickDraw. في نموذج الاسترجاع المباشر، يُمثَّل الاستعلام بتركيبة موزونة من الميزات البصرية لCLIP والتعليقات التوصيفية التي يولّدها BLIP وتسميات الفئات التي يتنبأ بها CLIP، ويُختبر ذلك عبر سبع تركيبات ثابتة وبحث معلمي خاص بكل مجموعة بيانات. في نموذج الاسترجاع التوليدي، يُحوَّل الرسم أولاً إلى صورة واقعية باستخدام Stable Diffusion المشروط إما بتعليق توصيفي من BLIP أو بتسمية فئة من CLIP، ثم تُستخدم الصورة المولَّدة استعلاماً للاسترجاع.

تكشف النتائج الرئيسية أن الإشارات النصية — تسميات الفئات والتعليقات التوصيفية — تتفوق باستمرار على الميزات البصرية الخام للرسم في مهام الاسترجاع على مستوى الفئات. ونظراً لأن CLIP لم يُدرَّب على صور الرسم، فإن الأوصاف النصية تجسّر فجوة النطاق بشكل أكثر موثوقية من المطابقة البصرية. كما يتبيّن أن التركيبة المثلى للميزات تعتمد على خصائص مجموعة البيانات: تستفيد مجموعات البيانات الأكثر تجريداً كQuickDraw من تسميات الفئات وحدها، في حين تستفيد المجموعات الأغنى بصرياً من دمج التعليقات التوصيفية. أما نموذج الاسترجاع التوليدي فيُقدّم أداءً أدنى من الاسترجاع المباشر باستمرار، نتيجة الاختلاف بين الصور المولَّدة والصور الحقيقية وتراكم الأخطاء عبر مراحل التوليد.

يتفوق الإطار المقترح على جميع الأساليب الأربعة المدرَّبة المقيَّمة، التي تمثّل أحدث ما توصلت إليه الأبحاث؛ إذ يحقق على مجموعة بيانات QuickDraw قيمة mAP@All تبلغ 0.437 مقارنةً بـ0.231 لأفضل نموذج مدرَّب بتحسّن نسبته 89.2%+، ويحقق أعلى قيمة لـmAP@200 على Extended Sketchy Split 2 بقيمة 0.761، ويتطابق مع أفضل نموذج مدرَّب في مقياس P@100 على TU-Berlin بفارق لا يتجاوز 0.4%، وكل ذلك دون أي تدريب متخصص، مما يُثبت إمكانية تحقيق استرجاع تنافسي للصور القائم على الرسم باستخدام النماذج المدربة مسبقاً.

الكلمات المفتاحية: استرجاع الصور القائم على الرسم، الاسترجاع دون تدريب، CLIP، BLIP، Stable Diffusion، ControlNet، دمج الميزات، التعلم متعدد الوسائط، الاسترجاع الخالي من التدريب

Contents

Dedication iv

Acknowledgements v

Abstract vi

ملخص vii

Table of Contents viii

List of Figures xi

List of Tables xiii

1 Introduction 1

1.1 Background and Motivation 1
1.2 Challenges in Sketch-Based Image Retrieval 1
1.3 Proposed Approach 2
1.4 Objectives of the Thesis 5
1.5 Contributions 5
1.6 Chapter Summary 5
1.7 Thesis Organization 6

2 Background 7

2.1 Embedding Representations 7
2.1.1 Definition and Motivation 7
2.1.2 Similarity Measures in Embedding Space 8
2.1.3 Multimodal Embedding Learning 8
2.1.4 Attention Mechanism 8
2.1.5 Role of Embeddings in This Thesis 9
2.2 Contrastive Language–Image Pretraining (CLIP) 10
2.2.1 Overview and Motivation 10
2.2.2 Model Architecture 10
2.2.3 Contrastive Training Objective 11

2.2.4 Pretraining Data and Scale . . . 12
2.2.5 Zero-Shot Inference Capability . . . 12
2.2.6 Model Variants . . . 12
2.2.7 Relevance to Sketch-Based Image Retrieval . . . 13
2.3 Bootstrapping Language-Image Pretraining (BLIP) . . . 13
2.3.1 Motivation for Caption Bootstrapping Idea . . . 13
2.3.2 BLIP Architecture . . . 14
2.3.3 Pretraining Objectives . . . 14
2.3.4 Training Pipeline . . . 15
2.3.5 Role of BLIP in This Thesis . . . 15
2.4 Stable Diffusion and ControlNet . . . 16
2.4.1 Diffusion Model Fundamentals . . . 16
2.4.2 Latent Diffusion Models . . . 17
2.4.3 Text Conditioning via Cross-Attention . . . 18
2.4.4 ControlNet for Structural Conditioning . . . 18
2.4.5 Role in This Thesis . . . 18
2.5 FAISS for Efficient Similarity Search . . . 19
2.5.1 Overview of FAISS . . . 19
2.5.2 Exact Search using IndexFlatIP . . . 19
2.5.3 Feature Indexing and Retrieval Pipeline . . . 20
2.5.4 Role of FAISS in This Thesis . . . 20
2.6 Evaluation Metrics . . . 20
2.6.1 Precision@K (P@K) . . . 21
2.6.2 Average Precision@K (AP@K) . . . 21
2.6.3 Mean Average Precision@K (mAP@K) . . . 21
2.6.4 Accuracy@K (Acc@K) . . . 21
2.7 Chapter Summary and Transition . . . 22

3 Literature Review **23**
3.1 Introduction . . . 23
3.2 Classical SBIR Approaches . . . 23
3.3 Deep Learning-Based SBIR . . . 24
3.3.1 Metric Learning–Based SBIR Methods . . . 24
3.3.2 Transformer-Based SBIR Methods . . . 25
3.3.3 Generative Models for SBIR . . . 25
3.3.4 Foundation Model–Based SBIR Methods . . . 26
3.4 Summary and Research Gap . . . 27

4 Methodology **28**

4.1 Introduction . . . 28
4.2 Problem Definition . . . 29
4.3 System Overview . . . 29
4.3.1 Paradigm 1: Direct Sketch-Based Retrieval . . . 29
4.3.2 Paradigm 2: Generation-Based Retrieval . . . 31
4.3.3 Shared Retrieval Engine . . . 32
4.4 Datasets . . . 33
4.4.1 Query Selection . . . 37
4.5 Semantic Inference with CLIP . . . 38
4.5.1 Image Encoding . . . 38
4.5.2 Text Encoding . . . 39
4.5.3 Similarity Computation . . . 39
4.5.4 Class Set Construction . . . 39
4.5.5 Zero-Shot Class Prediction . . . 40
4.5.6 CLIP Model Configuration . . . 40
4.5.7 BLIP Caption Generation . . . 40
4.6 Structural Conditioning . . . 42
4.6.1 Edge Enhancement . . . 42
4.6.2 ControlNet Conditioning . . . 43
4.7 Sketch-to-Image Generation . . . 43
4.7.1 Latent Diffusion Formulation . . . 43
4.7.2 Text Prompt Strategies . . . 43
4.7.3 Generation Parameters . . . 44
4.8 Feature Embedding . . . 44
4.8.1 Database Image Features . . . 44
4.8.2 Query Feature Configurations (Direct Retrieval) . . . 45
4.8.3 Feature Fusion . . . 45
4.8.4 Generation-Based Query Features . . . 46
4.9 Similarity Search with FAISS . . . 46
4.9.1 Index Construction . . . 46
4.9.2 Query Execution . . . 46
4.9.3 Retrieval Output . . . 47
4.10 Evaluation Metrics . . . 47
4.10.1 Evaluation Protocol Used in This Thesis . . . 47
4.11 Implementation Details . . . 47
4.11.1 Hardware Configuration . . . 47
4.11.2 Software Environment . . . 48
4.11.3 Model Sources . . . 48
4.12 Design Justification . . . 49

4.12.1 Why Pretrained Models Without Fine-Tuning? 49
4.12.2 Why CLIP ViT-B/32? 49
4.12.3 Why BLIP for Caption Generation? 50
4.12.4 Why Stable Diffusion + ControlNet? 51
4.12.5 Why FAISS IndexFlatIP? 51
4.12.6 Feature Extraction Efficiency 51
4.13 Chapter Summary 52

5 Results and Discussion **53**
5.1 Direct Sketch-Based Retrieval Results 53
5.1.1 Results on the Sketchy Dataset 53
5.1.2 Results on the Extended Sketchy Dataset 55
5.1.3 Results on the TU-Berlin Dataset 56
5.1.4 Results on the QuickDraw Dataset 57
5.2 Parameterized Feature Fusion Analysis 58
5.2.1 Retrieval Visualisation Across Configurations 60
5.3 Generation-Based Retrieval Results 66
5.4 Comparison with State-of-the-Art (SOTA) Methods 68
5.4.1 Comparison on Extended Sketchy 69
5.4.2 Comparison on TU-Berlin 70
5.4.3 Comparison on QuickDraw 71
5.4.4 Summary of SOTA Comparisons 72
5.4.5 Visual Comparison with ZSE-RN 73
5.5 Cross-Dataset Analysis and Key Findings 75
5.5.1 The Semantic-Surpasses-Visual Paradox 75
5.5.2 Dataset-Specific Optimal Strategies 75
5.5.3 Generation vs. Direct Retrieval 76
5.5.4 Zero-Shot Generalisation vs. Task-Specific Training 77
5.6 Limitations 77
5.7 Chapter Summary 78

6 Conclusion and Future Work **80**
6.1 Conclusion 80
6.2 Future Work 81

List of Figures

1.1 **Direct Sketch-Based Retrieval Full Configuration:** CLIP image features from sketch is combined with CLIP text embeddings of BLIP-generated captions and CLIP-predicted class labels to form a unified representation. 3
1.2 **SBIR Based on Generated Images** in which two textual prompting strategies for sketch-based image generation: (1) BLIP-generated captions, which describe the input sketch and are used as text prompts; (2) CLIP-predicted class labels, which provide high-level class information as text prompts. 4

2.1 CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of image–text training examples. At test time, the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes [3]. 11
2.2 Pre-training model architecture and objectives of BLIP [4]. 14
2.3 Forward (noising) and reverse (denoising) processes in diffusion models [11]. 17
2.4 Latent Diffusion Model architecture showing encoding to latent space and diffusion-based generation [7]. 18

4.1 Sample sketches from the Sketchy dataset across 16 categories, showing a controlled and detailed drawing style. 34
4.2 Sample sketches from the Extended Sketchy dataset across 16 categories, showing a similar drawing style to Sketchy. 35
4.3 Sample sketches from the TU-Berlin dataset across 16 categories, illustrating the variety of drawing styles across different users. 36
4.4 Sample sketches from the QuickDraw dataset across 16 categories, illustrating the highly abstract and minimal nature of time-constrained sketches. 37

5.1 The parrot sketch used as the query across all retrieval visualisation experiments. 60

5.2 Retrieval visualisation across configurations — Sketchy dataset [Caption: *"a parrot"*, Predicted class: *parrot*]. Each row shows top-5 results for one configuration (C1–C7 and Best). Red boxes mark incorrect retrievals. C1 (visual only) retrieves 3 correct results but returns 2 wrong ones (positions 2 and 5) — visually similar birds that do not match the parrot class. All semantic and fusion configurations (C2–C7 and Best) retrieve all 5 correctly. 61
5.3 Retrieval visualisation across configurations — Extended Sketchy dataset [Caption: *"a parrot"*, Predicted class: *parrot*]. C1 fails completely, retrieving a magpie, a giraffe, and an owl instead of parrots. C4 and C5 each introduce one failure due to residual visual-feature noise. C2, C3, C6, C7, and Best all retrieve correctly. 62
5.4 Retrieval visualisation across configurations — TU-Berlin dataset [Caption: *"a parrot"*, Predicted class: *pigeon*]. CLIP misclassified the sketch as *pigeon*. C2 (caption only) is the sole configuration that retrieves all 5 correctly. C3 (class only), C5 (sketch + class), C6 (caption + class), C7, and Best all fail completely because the wrong class label dominates the query. C4 (sketch + caption, no class) still produces 2 errors: without any class signal, the sketch visual feature alone introduces enough noise to pull two results toward pigeon-like images despite the correct caption. C1 (visual only) also fails completely. 63
5.5 Retrieval visualisation across configurations — QuickDraw dataset [Caption: *"a parrot"*, Predicted class: *parrot*]. C1 retrieves 3 correct results but returns 2 wrong ones (positions 3 and 5) despite visual shape similarity. All other configurations (C2–C7 and Best) retrieve all 5 correctly. Best (α=0.0, β=0.0) is equivalent to C3, confirming that class labels alone are the optimal signal for the highly abstract QuickDraw sketch style. 64
5.6 Generation-based retrieval results for a parrot sketch query across all four datasets, comparing BLIP caption prompt (left, blue) and CLIP predicted class prompt (right, purple). Each cell shows the generated image followed by the top-5 retrieved images. Red boxes mark incorrect retrievals. The class prompt fails on TU-Berlin because the predicted class is *pigeon*; the caption prompt succeeds on all four datasets. 68
5.7 Visual comparison of top-5 retrieval results between ZSE-RN (left, blue) and our training-free method (right, green) on six query sketches. The middle strip shows the BLIP-generated caption and CLIP-predicted class for each query (class shown in green if correct, red if wrong). Red boxes mark incorrect retrievals in our results. 74

List of Tables

2.1 CLIP model variants and encoder configurations (adapted from [3]). . . . 12

4.1 Benchmark datasets used for evaluation 33
4.2 Query selection statistics per dataset 38
4.3 Query feature configurations for direct retrieval 45
4.4 Core software dependencies and versions 48
4.5 Pretrained models and their Hugging Face sources 49
4.6 Average per-sketch feature extraction time (100 sketches, NVIDIA RTX 3080) 51

5.1 Direct retrieval results on the Sketchy dataset (C1–C7). 54
5.2 Direct retrieval results on the Extended Sketchy dataset (C1–C7). 55
5.3 Direct retrieval results on the TU-Berlin dataset (C1–C7). 56
5.4 Direct retrieval results on the QuickDraw dataset (C1–C7). 57
5.5 Best parameterized fusion results (optimal α and β per dataset). 59
5.6 BLIP caption and CLIP-predicted class for the parrot sketch query across datasets. 60
5.7 Generation-based retrieval results (BLIP caption vs. CLIP class label prompt strategies). Bold values indicate the better-performing prompt strategy within each dataset. 66
5.8 Comparison on Extended Sketchy (ZS-SBIR). Split 1 uses mAP@All and P@100; Split 2 uses mAP@200 and P@200. Dashes mean the metric was not reported. "Ours" uses α=0.2, β=0.0. 69
5.9 Comparison on TU-Berlin-Extended (ZS-SBIR): mAP@All and P@100. "Ours (Gen)" = generation-based result using CLIP class label prompt. . 70
5.10 Comparison on QuickDraw-Extended (ZS-SBIR): mAP@All and P@200. "Ours (Gen)" = generation-based result using CLIP class label prompt. . 71
5.11 Summary: our parameterized best vs. best trained SOTA per metric. ✓ = we surpass best SOTA. ≈ = within 2%. ∘ = within 10%. × = gap >10%. 73

Chapter 1

Introduction

1.1 Background and Motivation

The growing volume of visual data has created a need for more intuitive image retrieval systems. Traditional retrieval methods rely on text descriptions or reference images, which are not always available or easy to provide. Hand-drawn sketches offer a more natural alternative — users can simply draw what they are looking for. This idea led to the development of SBIR, where a freehand sketch is used as a query to retrieve matching photos from a database.

However, making SBIR work well is challenging. Sketches are sparse and abstract, while natural photos are rich in colour, texture, and detail. This visual difference — known as the **domain gap** — makes it hard for retrieval systems to match sketches with the correct photos. Early methods based on handcrafted features such as edge maps struggled with this gap and did not scale well [1]. Deep learning methods improved performance by learning shared representations for sketches and images, but they require large amounts of labelled training data and do not generalise well to new categories [2].

More recently, large-scale pretrained vision–language models such as **CLIP** (**C**ontrastive **L**anguage–**I**mage **P**retraining) [3] and captioning models such as **BLIP** (**B**ootstrapping **L**anguage-**I**mage **P**retraining) [4] have shown strong performance across many vision tasks without task-specific training, raising the central research question of this thesis: *can pretrained vision–language models be used to retrieve photos from sketch queries without any task-specific training, and if so, which combination of features is most effective?*

1.2 Challenges in Sketch-Based Image Retrieval

SBIR faces several well-known challenges [5], [6]:

1. **Domain gap**: Sketches are abstract and lack visual details, while natural images contain rich textures, color, and lighting variations.

2. **Viewpoint variation**: Query sketches and database images often depict objects from different viewpoints, leading to significant structural and geometric mismatches that hinder effective retrieval.

3. **Style variation (intra-class variability)**: Sketches drawn by different users vary significantly in style, level of abstraction, and drawing skill.

4. **Datasets limitations**: SBIR datasets are relatively small compared to image datasets used for deep learning, making it difficult to train models that generalize across categories.

5. **Semantic gap**: Low-level visual features alone often fail to capture the conceptual intent behind sketches.

These challenges motivate the use of **semantic representations** and **multi-modal models** to bridge the sketch–image gap.

1.3 Proposed Approach

This thesis investigates two SBIR paradigms, both using pretrained models without any task-specific training:

1. **Direct Sketch-Based Retrieval.** Sketches are represented using combinations of visual and semantic features. Visual features come from CLIP's image encoder. Semantic features come from BLIP-generated captions and CLIP-predicted class labels, both encoded using CLIP's text encoder. Seven feature configurations are evaluated, ranging from visual-only to fully semantic and multimodal combinations, as illustrated in Figure 1.1.

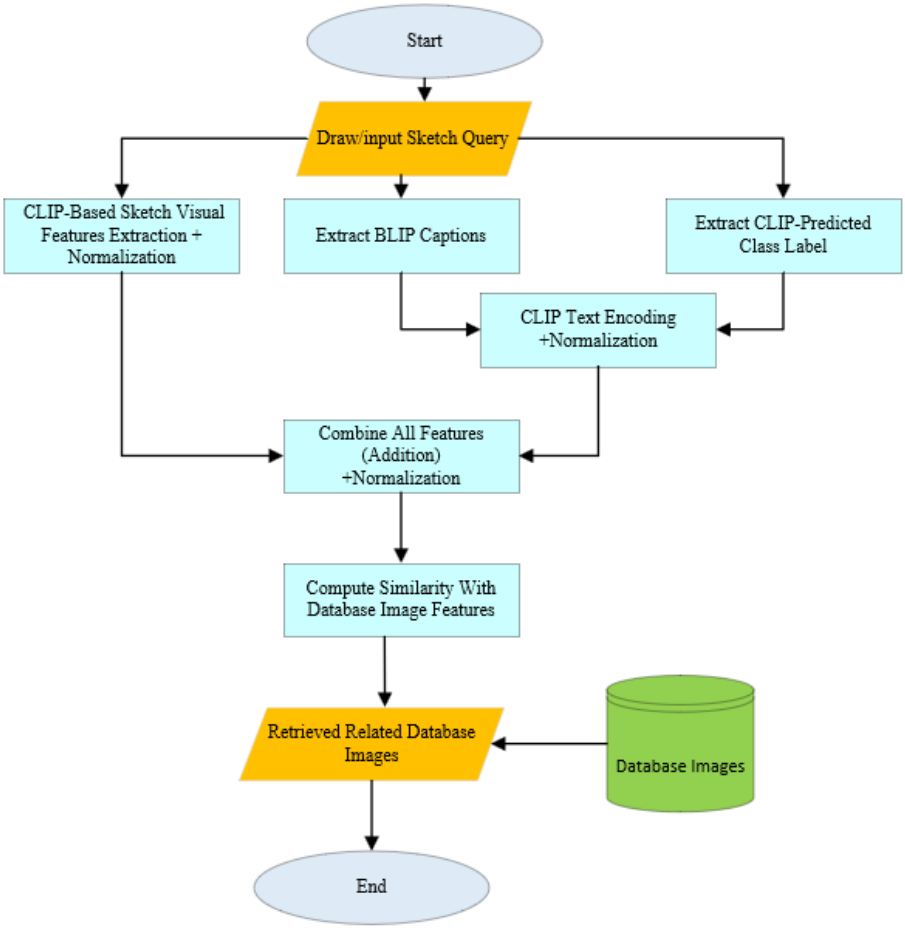

Figure 1.1: Direct Sketch-Based Retrieval Full Configuration: CLIP image features from sketch is combined with CLIP text embeddings of BLIP-generated captions and CLIP-predicted class labels to form a unified representation.

2. **Generation-Based Retrieval.** Sketches are converted into realistic photos using **stable diffusion** [7] with **ControlNet** [8], guided by either a BLIP-generated caption or a CLIP-predicted class label as the text prompt. The generated image is then used for retrieval, as illustrated in Figure 1.2.

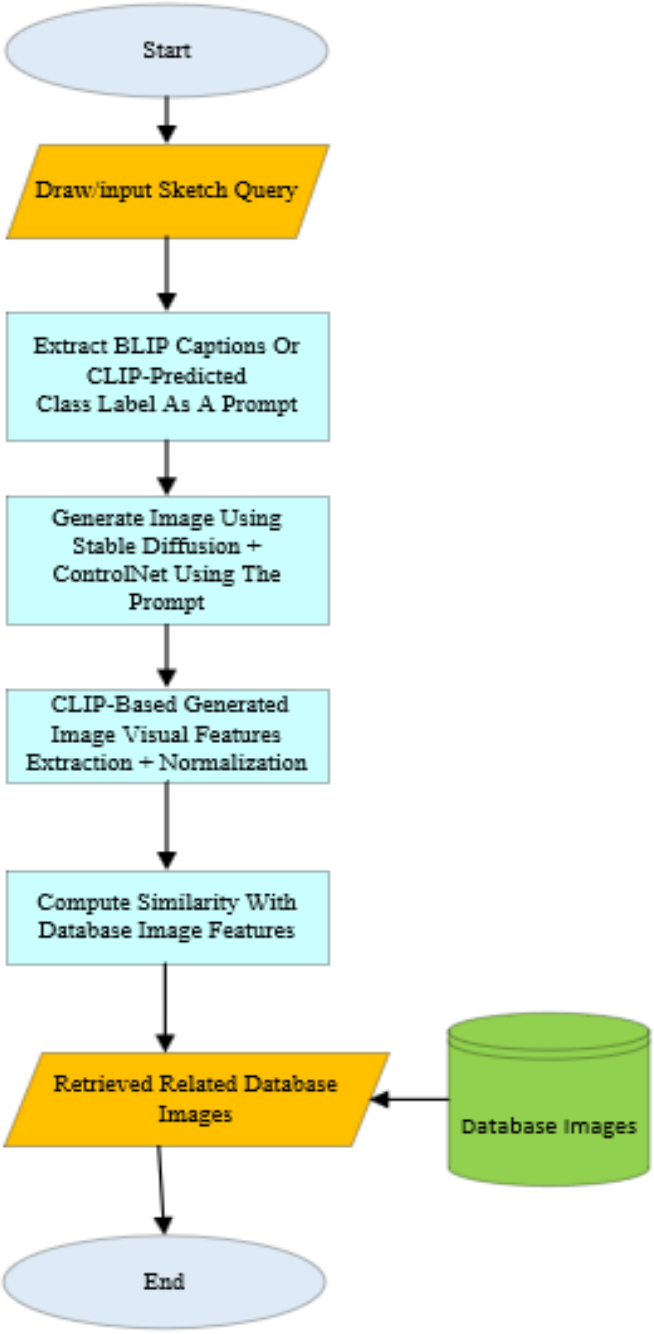

Figure 1.2: SBIR Based on Generated Images in which two textual prompting strategies for sketch-based image generation: (1) BLIP-generated captions, which describe the input sketch and are used as text prompts; (2) CLIP-predicted class labels, which provide high-level class information as text prompts.

Both paradigms use CLIP's shared embedding space, allowing visual and textual features to be directly compared and combined.

1.4 Objectives of the Thesis

The main objectives of this thesis are:

1. To evaluate direct sketch-based retrieval using CLIP visual features, BLIP-generated captions, and CLIP-predicted class labels, both individually and in combination.
2. To investigate the effectiveness of pretrained vision–language and captioning models for SBIR without any task-specific training.
3. To evaluate generation-based retrieval using stable diffusion with ControlNet, guided by BLIP captions or CLIP class labels.
4. To conduct evaluation across four benchmark SBIR datasets: Sketchy, Extended Sketchy, TU-Berlin, and QuickDraw.
5. To compare the results with state-of-the-art SBIR methods and identify the strengths and limitations of each approach.

1.5 Contributions

The main contributions of this thesis are:

- A **unified evaluation framework** for both direct and generation-based SBIR, applied across four benchmark datasets.
- **Systematic empirical analysis** of seven feature configurations combining CLIP visual features, BLIP captions, and CLIP class labels, without any task-specific training.
- **Integration of generative models** (stable diffusion with ControlNet) into the SBIR pipeline, with a comparison of two automatic prompting strategies.
- **Comparison with state-of-the-art methods**, highlighting the conditions under which a training-free approach matches or surpasses trained methods.

1.6 Chapter Summary

This chapter introduced the motivation behind SBIR and the challenges that make it a difficult problem, including the domain gap between sketches and natural photos, style variation across users, and the semantic gap between visual features and user intent.

Two SBIR paradigms are investigated in this thesis: direct retrieval, where sketches are represented using combinations of visual and semantic features extracted from pretrained CLIP and BLIP models, and generation-based retrieval, where sketches are first converted into realistic photos using stable diffusion with ControlNet before retrieval. Both paradigms rely entirely on pretrained models without any task-specific training.

The objectives and contributions of this thesis were outlined, emphasizing the systematic evaluation of multiple feature configurations across four benchmark datasets and comparison with state-of-the-art methods. This sets the stage for the following chapters.

1.7 Thesis Organization

The remainder of this thesis is organized as follows:

- **Chapter 2: Background** This chapter introduces the key pretrained models used in this thesis: CLIP, BLIP, stable diffusion with ControlNet, and FAISS, along with the evaluation metrics used throughout the experiments.
- **Chapter 3: Literature Review** This chapter provides an overview of previous research in sketch-based image retrieval, covering classical approaches, deep learning-based methods, and multi-modal retrieval strategies.
- **Chapter 4: Methodology** This chapter describes the proposed SBIR pipelines, including feature extraction using CLIP and BLIP, sketch-to-image generation using stable diffusion with ControlNet, datasets, evaluation metrics, and implementation details.
- **Chapter 5: Results and Discussion** This chapter presents and analyses the experimental results across all feature configurations, datasets, and paradigms, with comparison against state-of-the-art methods.
- **Chapter 6: Conclusion and Future Work** This chapter summarizes the main findings, discusses limitations, and suggests directions for future research.

Chapter 2

Background

This chapter provides the technical background of the main components used in this thesis. It covers five key areas: embedding representations, CLIP, BLIP, stable diffusion with ControlNet, and FAISS. Together, these form the foundation of the proposed SBIR pipeline. The chapter also presents the evaluation metrics used throughout the experiments.

2.1 Embedding Representations

Embedding representations play a fundamental role in modern deep learning systems, particularly in tasks involving cross-modal retrieval such as SBIR. An embedding is a fixed-size vector representation that captures the semantic and structural properties of input data, such as images, sketches, or text.

2.1.1 Definition and Motivation

Given an input sample x, an embedding function $f(\cdot)$ maps it into a continuous vector space (also called **embedding space**):

$$\mathbf{z} = f(x), \quad \mathbf{z} \in \mathbb{R}^d \tag{2.1}$$

where d is the embedding dimension. The goal of this mapping is to ensure that semantically similar inputs are located close to each other in the embedding space, while dissimilar inputs are mapped farther apart.

Embedding representations enable efficient comparison between different data modalities by transforming them into a shared feature space. This is particularly important in SBIR, where sketches, images, and text must be compared despite their inherent differences.

2.1.2 Similarity Measures in Embedding Space

Once data is mapped into an embedding space, similarity between two samples can be computed using distance metrics. Among the most widely used is **cosine similarity**, which measures the cosine of the angle θ between two vectors $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^d$ [9]:

$$\cos(\theta) = \frac{\mathbf{z}_1^\top \mathbf{z}_2}{\|\mathbf{z}_1\| \, \|\mathbf{z}_2\|} \tag{2.2}$$

The result lies in the range $[-1, 1]$: a value of 1 indicates that the two vectors point in the same direction (maximum similarity), 0 indicates orthogonality (no similarity), and -1 indicates opposite directions. Crucially, cosine similarity is *scale-invariant* — it depends only on the orientation of the vectors, not their magnitudes. This makes it well-suited for comparing embeddings produced by different encoders or input modalities, where the norms may vary.

When embeddings are ℓ_2-normalised (i.e. $\|\mathbf{z}\| = 1$), cosine similarity reduces to the standard inner product:

$$\text{sim}(\mathbf{z}_1, \mathbf{z}_2) = \mathbf{z}_1^\top \mathbf{z}_2 \tag{2.3}$$

In this thesis, CLIP outputs are ℓ_2-normalised by design, so all similarity computations are performed using the inner product, which is equivalent to cosine similarity. This formulation allows efficient retrieval of nearest neighbours in high-dimensional spaces.

2.1.3 Multimodal Embedding Learning

In multimodal systems such as CLIP and BLIP, embeddings are learned jointly across different modalities. This means that images, sketches, and text are projected into a shared embedding space where cross-modal similarity can be computed directly.

For example:

- An image and its textual description are mapped to nearby vectors.
- A sketch and a real image of the same category are mapped to nearby vectors through their shared text-based semantic space.

This alignment is typically achieved through contrastive learning objectives, which encourage matching pairs to have high similarity and non-matching pairs to have low similarity.

2.1.4 Attention Mechanism

Attention is a mechanism used in deep learning that allows a model to focus on the most relevant parts of the input when producing an output [10]. Given a set of queries Q, keys

K, and values V, the attention output is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V \tag{2.4}$$

where d is the dimension of the key vectors. Intuitively, the queries represent what the model is looking for, the keys represent what is available, and the values represent the actual information to retrieve.

Self-attention allows a model to relate different parts of the same input to each other — for example, relating words in a sentence or patches in an image. This is the core mechanism used in Transformer-based models such as the text and image encoders in CLIP and BLIP.

Cross-attention allows a model to relate information from two different inputs — for example, relating image regions to words in a text prompt. This is the mechanism used in stable diffusion to allow text prompts to guide image generation, and in BLIP to align image and text representations during caption generation.

Both forms of attention are used extensively in the models described in this chapter, and understanding this mechanism is important for understanding how text and image representations are aligned across modalities.

2.1.5 Role of Embeddings in This Thesis

Embedding representations serve as the backbone of the proposed SBIR framework. Their role can be summarized as follows:

- CLIP is used to extract embeddings from sketches, generated images, and real images.
- BLIP generates captions that are also mapped into embedding space for semantic representation.
- Both direct and generative SBIR approaches rely on embedding similarity for retrieval.
- FAISS (Facebook AI Similarity Search) operates on these embeddings to perform efficient nearest neighbor search.

By leveraging a shared embedding space, the system enables effective comparison across modalities, thereby bridging the domain gap between sketches and natural images.

2.2 Contrastive Language–Image Pretraining (CLIP)

2.2.1 Overview and Motivation

Contrastive Language–Image Pretraining (CLIP) is a large-scale vision–language model designed to learn transferable visual representations through natural language supervision. Unlike traditional supervised visual learning approaches that rely on fixed label sets and task-specific datasets, CLIP leverages large-scale image–text pairs collected from the web.

All technical details presented in this section are derived from the original CLIP paper by Radford et al. [3].

The central objective of CLIP is to align images and text descriptions within a shared embedding space, enabling semantic comparison across modalities. This means CLIP can recognize new categories it has never been specifically trained on, simply by comparing image embeddings with text descriptions of those categories. Such properties make CLIP particularly suitable for cross-modal retrieval problems, including SBIR.

2.2.2 Model Architecture

CLIP consists of two separate encoders: an image encoder and a text encoder. The image encoder can be instantiated as either a convolutional neural network (e.g., ResNet variants) or a Vision Transformer (ViT), while the text encoder is implemented as a Transformer-based language model. Given an image I and a text description T, the encoders map them into a shared embedding space as

$$\mathbf{v}_I = f_{\text{img}}(I), \quad \mathbf{v}_T = f_{\text{text}}(T), \tag{2.5}$$

where $f_{\text{img}}(\cdot)$ and $f_{\text{text}}(\cdot)$ denote the image and text encoders, respectively.

Both embeddings are ℓ_2-normalized so that their dot product corresponds to cosine similarity. Figure 2.1 illustrates the overall CLIP training framework.

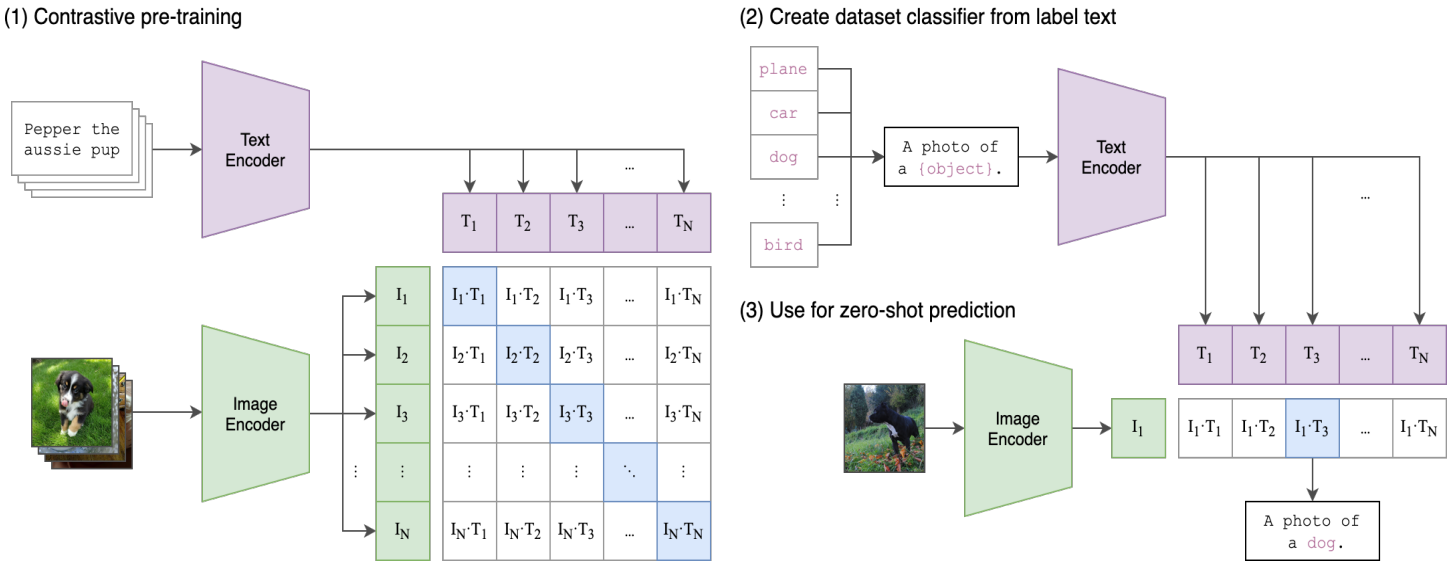

Figure 2.1: CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of image–text training examples. At test time, the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes [3].

2.2.3 Contrastive Training Objective

CLIP is trained using a symmetric contrastive loss over a batch of N image–text pairs. For each image, the corresponding text is treated as a positive example, while all other texts in the batch are treated as negatives, and vice versa. The similarity between image and text embeddings is computed as

$$\text{sim}(I_i, T_j) = \frac{\mathbf{v}_{I_i}^\top \mathbf{v}_{T_j}}{\tau}, \tag{2.6}$$

where τ is a learnable temperature parameter.

The contrastive loss is defined as the average of image-to-text and text-to-image cross-entropy losses:

$$\mathcal{L} = \frac{1}{2}\left(\mathcal{L}_{\text{img}\to\text{text}} + \mathcal{L}_{\text{text}\to\text{img}}\right), \tag{2.7}$$

where

$$\mathcal{L}_{\text{img}\to\text{text}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(\text{sim}(I_i, T_i))}{\sum_{j=1}^{N} \exp(\text{sim}(I_i, T_j))}. \tag{2.8}$$

This objective encourages matched image–text pairs to have high similarity, while pushing mismatched pairs apart in the embedding space.

2.2.4 Pretraining Data and Scale

CLIP is trained on approximately 400 million image–text pairs collected from publicly available sources on the internet. The diversity and scale of this dataset enable CLIP to learn rich semantic representations that capture objects, attributes, actions, and high-level concepts beyond the scope of manually selected datasets. This large-scale pretraining is a key factor in CLIP's strong generalization capabilities.

2.2.5 Zero-Shot Inference Capability

A key property of CLIP is its ability to perform zero-shot classification without task-specific retraining. At inference time, textual class descriptions are encoded using the text encoder, and an image is classified by selecting the text prompt with the highest similarity to the image embedding. This formulation effectively treats classification as a retrieval problem in the joint embedding space.

Formally, given a set of class prompts $\{T_k\}$ (textual descriptions of classes), the predicted class $\hat{k}$ for an image I (input image or sketch) is obtained as:

$$\hat{k} = \arg\max_{k} \ \mathbf{v}_I^\top \mathbf{v}_{T_k}, \tag{2.9}$$

where $\mathbf{v}_I$ and $\mathbf{v}_{T_k}$ denote the image and text embeddings, respectively, and the inner product measures their similarity.

This design allows CLIP to adapt flexibly to new tasks and label sets without additional training.

2.2.6 Model Variants

The original CLIP paper evaluates multiple model variants with different image encoder backbones and capacities. Table 2.1 summarizes the main CLIP configurations used in practice.

Table 2.1: CLIP model variants and encoder configurations (adapted from [3]).

Model Variant	**Image Encoder**	**Text Encoder**
RN50	ResNet-50	Transformer
RN101	ResNet-101	Transformer
ViT-B/32	Vision Transformer (B/32)	Transformer
ViT-B/16	Vision Transformer (B/16)	Transformer
ViT-L/14	Vision Transformer (L/14)	Transformer

2.2.7 Relevance to Sketch-Based Image Retrieval

Although CLIP was not originally designed for Sketch-Based Image Retrieval, its joint vision–language embedding space makes it particularly suitable for cross-modal retrieval tasks. By encoding sketches, images, and textual descriptions into a shared semantic space, CLIP enables direct comparison and fusion of visual and textual modalities. This property forms the foundation for the SBIR strategies explored in later chapters of this thesis, where CLIP is employed as a pretrained feature extractor without task-specific retraining.

2.3 Bootstrapping Language-Image Pretraining (BLIP)

Bootstrapping Language-Image Pretraining (BLIP) is a unified vision–language framework designed to improve multimodal understanding and generation by effectively leveraging noisy web-scale image–text data.

All technical details presented in this section are derived from the original BLIP paper by Li et al. [4].

Unlike earlier vision–language models that rely solely on raw web annotations, BLIP introduces a bootstrapping mechanism that refines noisy captions and enhances the quality of supervision during pretraining.

BLIP plays a crucial role in this thesis by providing semantic textual representations in the form of *caption features*, which are used to complement visual sketch features in the SBIR pipeline.

2.3.1 Motivation for Caption Bootstrapping Idea

Large-scale image–text datasets collected from the web often contain noisy, incomplete, or misaligned captions. Such noise can degrade the performance of multimodal models. BLIP addresses this challenge through a *caption bootstrapping* strategy, which consists of:

- A captioner generates synthetic captions for images
- A filtering model selects high-quality captions
- The refined dataset is used for further training

This iterative refinement reduces noise and improves semantic alignment between modalities.

2.3.2 BLIP Architecture

BLIP adopts a unified architecture that supports both vision–language understanding and generation tasks. It consists of three main components:

1. **Image Encoder:** A Vision Transformer (ViT) that extracts visual features from input images.
2. **Text Encoder:** A Transformer-based encoder that processes textual inputs.
3. **Multimodal Encoder–Decoder:** A shared architecture that enables both discriminative and generative tasks.

This unified design allows BLIP to operate in two modes:

- *Understanding mode* (e.g., image–text matching)
- *Generation mode* (e.g., image captioning)

2.3.3 Pretraining Objectives

BLIP is trained using a combination of three key objectives that jointly optimize vision–language alignment, as illustrated in Figure 2.2:

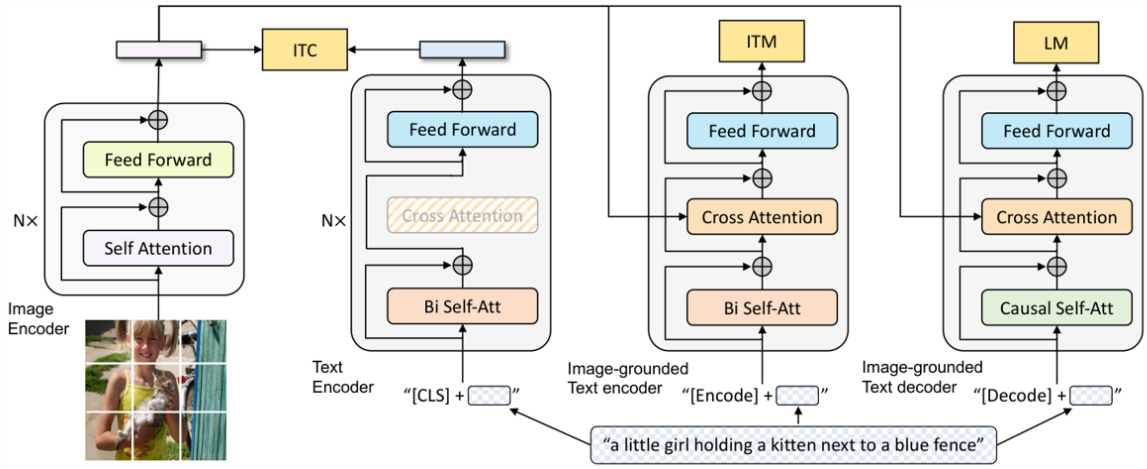

Figure 2.2: Pre-training model architecture and objectives of BLIP [4].

1. **Image–Text Contrastive Loss (ITC):** This objective aligns global image and text representations in a shared embedding space, similar to CLIP:

$$\mathcal{L}_{\text{ITC}} = -\log \frac{\exp(\text{sim}(I, T)/\tau)}{\sum_{T'} \exp(\text{sim}(I, T')/\tau)} \tag{2.10}$$

where I and T denote the image and text embeddings of a matched pair, T' iterates over all text embeddings in the batch, $\text{sim}(\cdot, \cdot)$ is the cosine similarity function, and τ is a learnable temperature parameter that controls the sharpness of the similarity distribution.

2. **Image–Text Matching Loss (ITM):** This objective learns fine-grained alignment by predicting whether an image–text pair is matched:

$$\mathcal{L}_{\text{ITM}} = -y \log p(I, T) - (1 - y) \log(1 - p(I, T)) \tag{2.11}$$

where $p(I, T)$ is the model's predicted probability that the image–text pair is matched, and $y \in \{0, 1\}$ is the ground-truth label, with $y = 1$ indicating a matched pair and $y = 0$ indicating a mismatched pair.

3. **Language Modeling Loss (LM):** For caption generation, BLIP uses a standard autoregressive language modeling objective:

$$\mathcal{L}_{\text{LM}} = -\sum_{t} \log P(w_t | w_{<t}, I) \tag{2.12}$$

where w_t is the token at position t, $w_{<t}$ denotes all previously generated tokens, and I is the image embedding that conditions the caption generation process.

The total BLIP training loss is the summation of all three objectives:

$$\mathcal{L}_{\text{BLIP}} = \mathcal{L}_{\text{ITC}} + \mathcal{L}_{\text{ITM}} + \mathcal{L}_{\text{LM}} \tag{2.13}$$

2.3.4 Training Pipeline

During training, BLIP simultaneously learns:

- global alignment between images and text (ITC).
- fine-grained matching (ITM).
- caption generation (LM).

The combination of these objectives allows the model to effectively understand and generate multimodal content, making it highly suitable for downstream tasks.

2.3.5 Role of BLIP in This Thesis

In this thesis, BLIP is used to extract **caption features** from input sketches. Specifically:

- The input sketch is treated as an image
- BLIP generates a natural language caption describing the sketch
- The generated caption is encoded into a semantic feature vector

These caption features are then:

- Combined with sketch features in direct SBIR.
- Used as textual prompts in generative SBIR frameworks based on stable diffusion and ControlNet.

This enables the incorporation of high-level semantic information, which helps bridge the domain gap between sketches and natural images.

2.4 Stable Diffusion and ControlNet

Stable diffusion, based on Latent Diffusion Models (LDMs), is a widely used framework for high-quality image synthesis. The theoretical foundation of diffusion models originates from Denoising Diffusion Probabilistic Models (DDPM) [11], while stable diffusion adopts the latent-space formulation introduced by Rombach et al. [7]. ControlNet extends this framework by enabling structured conditioning for spatial control [8]. All technical formulations and architectural details presented in this section are derived directly from these original works.

2.4.1 Diffusion Model Fundamentals

Diffusion models consist of two processes: a forward (training) process and a reverse (generation) process.

Training Process. During training, Gaussian noise is progressively added to a real image over T steps until it becomes pure random noise. Formally, the forward process at each step is defined as:

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{2.14}$$

where x_0 is the original clean image, x_{t-1} is the image at the previous timestep, x_t is the resulting noisy image at the current timestep, $\sqrt{1-\beta_t}$ is a scaling factor that slightly reduces the signal magnitude to keep the total variance bounded, $\beta_t \in (0,1)$ is the noise level at step t drawn from a predefined variance schedule that increases from near zero to a small value across T steps, I is the identity matrix indicating that noise is added independently to each dimension, $q(\cdot)$ denotes the fixed forward noising process, and $\mathcal{N}$

denotes a Gaussian distribution [11]. The model learns to predict the noise added at each step by minimizing:

$$\mathcal{L} = \mathbb{E}_{x,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right] \tag{2.15}$$

where $\epsilon \sim \mathcal{N}(0, I)$ is the true Gaussian noise that was added, $\epsilon_\theta(x_t, t)$ is the noise predicted by a U-Net neural network parameterised by θ given the noisy image x_t and timestep t, and t is the current timestep.

Generation Process. Once trained, the model generates new images by reversing the noising process. Starting from pure random noise x_T, the model iteratively denoises at each step using:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{2.16}$$

where $p_\theta(\cdot)$ denotes the learned reverse (denoising) process, and μ_θ and Σ_θ are the predicted mean and variance. This is repeated until a clean image x_0 is obtained, as illustrated in Figure 2.3.

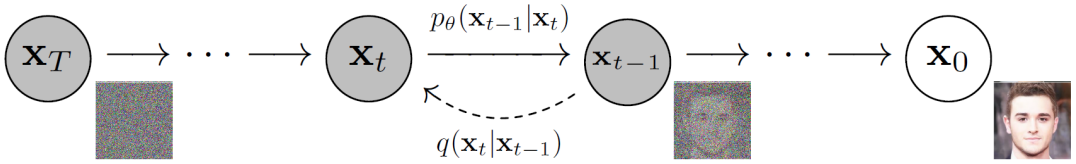

Figure 2.3: Forward (noising) and reverse (denoising) processes in diffusion models [11].

2.4.2 Latent Diffusion Models

Running diffusion directly on full-size images is computationally expensive. Stable diffusion solves this by first compressing the image into a smaller latent representation using a Variational Auto Encoder (VAE):

$$z = \mathcal{E}(x), \quad x \approx \mathcal{D}(z) \tag{2.17}$$

where x is the original full-size image, z is its compressed latent representation, and $\mathcal{E}$ and $\mathcal{D}$ are the encoder and decoder, respectively.

During training, the diffusion process is applied in this compressed latent space rather than on the full image, which is much faster and requires less memory.

During generation, the model starts from random noise in the latent space, iteratively denoises it, and the decoder $\mathcal{D}$ converts the final latent back into a full-size image, as illustrated in Figure 2.4.

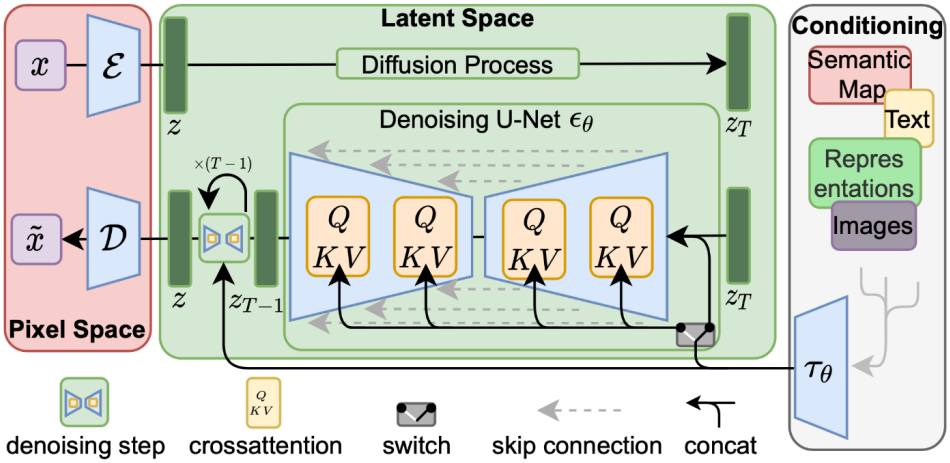

Figure 2.4: Latent Diffusion Model architecture showing encoding to latent space and diffusion-based generation [7].

2.4.3 Text Conditioning via Cross-Attention

Text conditioning is achieved through cross-attention (Section 2.1.4), where the image latent attends to the text embedding at each denoising step, allowing the text prompt to guide the content and appearance of the generated image. Note that the cross-attention parameters are learned during training to align text and image representations.

2.4.4 ControlNet for Structural Conditioning

Text conditioning controls the semantic content of the generated image but cannot enforce its spatial structure. ControlNet addresses this by adding an extra input — such as a sketch or edge map — that forces the generated image to follow a specific layout [8].

During training, ControlNet learns an additional parallel branch alongside the diffusion model that processes the structural input. Zero-initialized convolution layers are used so that this branch starts without affecting the original model and is gradually integrated during training.

During generation, both the text prompt and the structural input (sketch) jointly guide the denoising process, producing images that are semantically consistent with the text and structurally consistent with the sketch.

2.4.5 Role in This Thesis

In this thesis, stable diffusion with ControlNet is employed to bridge the domain gap between sketches and natural images. By combining semantic and structural guidance, the model generates high-fidelity images consistent with sketch inputs. The process is summarized as follows:

- A sketch is provided as a structural condition via **ControlNet**.
- A textual prompt generated by **BLIP** or **CLIP** provides semantic guidance.
- **Stable diffusion** generates a realistic image that preserves both structure and meaning.

The generated image is subsequently used as input to the retrieval system, transforming the SBIR problem into an image-to-image retrieval task.

2.5 FAISS for Efficient Similarity Search

Efficient similarity search is a critical component in large-scale SBIR systems. In this thesis, Facebook AI Similarity Search (**FAISS**) is employed as the backbone for fast and scalable nearest neighbor retrieval. FAISS is an open-source library developed by **Meta AI** that enables efficient similarity search and clustering of high-dimensional vectors [12]. All technical concepts presented in this section are derived from the original FAISS documentation and research works.

2.5.1 Overview of FAISS

FAISS enables similarity search by indexing feature vectors and performing nearest neighbor queries. Given a query vector $\mathbf{q}$ and a database of vectors $\{\mathbf{x}_i\}$, the goal is to retrieve the most similar vectors based on a distance metric.

In this thesis, similarity is computed using inner product (cosine similarity equivalent after normalization):

$$\text{sim}(\mathbf{q}, \mathbf{x}) = \mathbf{q}^\top \mathbf{x} \tag{2.18}$$

FAISS supports multiple indexing strategies, ranging from exact search to approximate methods for large-scale datasets.

2.5.2 Exact Search using IndexFlatIP

For this work, the `IndexFlatIP` index is used, which performs exact nearest neighbor search based on inner product similarity. This method stores all feature vectors in memory and computes similarity scores directly:

$$\hat{i} = \arg\max_i \left(\mathbf{q}^\top \mathbf{x}_i\right) \tag{2.19}$$

Although computationally intensive for very large datasets, this approach guarantees optimal retrieval accuracy, making it suitable for evaluation and moderate-scale datasets.

2.5.3 Feature Indexing and Retrieval Pipeline

The FAISS-based retrieval process consists of two main stages:

- **Indexing:** Feature vectors extracted from images are stored in the FAISS index.
- **Querying:** A query feature vector is compared against the indexed database to retrieve the top-k most similar results.

2.5.4 Role of FAISS in This Thesis

In this thesis, FAISS serves as the core retrieval engine for both the direct and generative SBIR pipelines. It enables efficient similarity search over high-dimensional feature embeddings, facilitating fast and accurate retrieval of relevant images from the database.

The proposed framework incorporates two complementary retrieval approaches:

- **Direct Sketch-Based Retrieval:** In this approach, the input sketch is directly encoded into a feature vector using the CLIP image encoder. These sketch features are mapped into the same embedding space as real images, allowing direct comparison. FAISS is then used to retrieve the top-k most similar images based on feature similarity.
- **Generative Sketch-Based Retrieval:** In this approach, the sketch is first transformed into a realistic image using stable diffusion combined with ControlNet, guided by semantic prompts generated by BLIP or CLIP. The generated image is then encoded using CLIP to obtain a feature representation. FAISS performs similarity search between this generated image representation and the database features.

In both approaches, FAISS operates on feature embeddings generated by CLIP, maintaining a shared retrieval space. This allows FAISS to provide a flexible and robust retrieval framework, enhancing performance across diverse sketch types, especially those that vary in abstraction and level of detail.

2.6 Evaluation Metrics

Retrieval performance in SBIR is commonly evaluated using ranking-based metrics computed per query and averaged over the test set. The specific metrics reported in the experiments are detailed in Chapter 4.

2.6.1 Precision@K (P@K)

$$P@K = \frac{\text{Number of relevant images in top } K}{K} \tag{2.20}$$

Analysis: Precision@K measures the proportion of relevant images among the top-K retrieved results. It reflects the system's ability to return accurate results within the first few positions, which is critical in real-world scenarios where users typically inspect only top-ranked results. However, it does not consider the ordering of relevant images within the top-K list.

2.6.2 Average Precision@K (AP@K)

$$AP@K = \frac{1}{R} \sum_{k=1}^{K} P@k \cdot rel(k) \tag{2.21}$$

where R is the total number of relevant images for a given query, and $rel(k) \in \{0, 1\}$ indicates whether the item at rank k is relevant.

Analysis: AP@K extends Precision@K by incorporating the ranking positions of relevant items. It rewards methods that retrieve relevant images earlier in the ranked list. Therefore, it provides a more informative evaluation by capturing both relevance and ranking quality within the top-K results. Note that AP@K is not reported directly in the experiments but serves as the basis for computing mAP@K.

2.6.3 Mean Average Precision@K (mAP@K)

$$mAP@K = \frac{1}{Q} \sum_{i=1}^{Q} AP_i@K \tag{2.22}$$

where Q is the total number of query sketches and $AP_i@K$ is the Average Precision@K for the i-th query.

Analysis: mAP@K is the most comprehensive metric for evaluating SBIR systems, as it aggregates performance across all queries while considering ranking quality. It provides a balanced measure of both retrieval accuracy and consistency. Limiting the evaluation to top-K results focuses on practically relevant retrieval scenarios.

2.6.4 Accuracy@K (Acc@K)

$$Acc@K = \begin{cases} 1 & \text{if at least one relevant image appears in top K} \\ 0 & \text{otherwise} \end{cases} \tag{2.23}$$

Analysis: Accuracy@K evaluates whether the system retrieves at least one correct result within the top-K results. It reflects the system's ability to provide a quick successful match. However, it is a coarse metric, as it does not consider the number of relevant results or their ranking positions. Acc@K can be applied at two levels of granularity: in **instance-level** (fine-grained) SBIR, a retrieval is correct only if the exact paired image appears in the top-K results; in **category-level** (coarse-grained) SBIR, a retrieval is correct if at least one image from the same category as the query appears in the top-K results. The former is suited to fine-grained benchmarks with paired sketch–photo annotations, while the latter is more appropriate for large-scale or unpaired settings where category membership is the primary retrieval signal.

2.7 Chapter Summary and Transition

This chapter presented the fundamental concepts and models that form the basis of the proposed SBIR framework. It began with an overview of embedding representations, which provide a unified feature space for comparing sketches, images, and text. Subsequently, the CLIP model was introduced as a powerful multimodal encoder for extracting aligned visual and textual features. The BLIP framework was then discussed, highlighting its ability to generate semantic captions that enrich sketch representations.

Furthermore, diffusion-based generative models were explored through stable diffusion and its extension with ControlNet, enabling the transformation of sketches into realistic images while preserving structural and semantic information. Finally, FAISS was presented as an efficient similarity search mechanism for retrieving relevant images based on feature embeddings.

Together, these components establish the technical foundation for the proposed SBIR system, supporting both direct and generative retrieval strategies.

Building upon this background, the next chapter reviews existing literature in SBIR, covering both classical approaches and modern deep learning–based methods, and highlighting current challenges and research gaps addressed in this thesis.

Chapter 3

Literature Review

3.1 Introduction

SBIR has been extensively studied in the computer vision community, with the objective of retrieving semantically relevant natural images from a database using a freehand sketch as a query. By allowing users to express visual concepts intuitively without requiring reference images or textual descriptions, SBIR provides a natural and flexible retrieval interface. However, the substantial visual gap between abstract sketches and richly detailed natural images makes SBIR a challenging problem.

Retrieval performance in SBIR is highly dependent on the quality of feature representations. Effective features must capture the structural characteristics of sketches while encoding semantic information that aligns with natural images, and remain robust to variations in drawing style and abstraction. Consequently, advances in SBIR have been closely tied to improvements in representation learning and cross-modal feature alignment.

This chapter reviews prior work in SBIR, focusing on two main categories: classical SBIR approaches and deep learning–based SBIR methods. The review highlights the evolution of the field from handcrafted feature representations to modern deep learning and foundation model–based solutions that learn semantically meaningful cross-modal embeddings.

3.2 Classical SBIR Approaches

Classical SBIR approaches predate deep learning and rely on handcrafted representations and traditional retrieval pipelines. These methods aim to bridge the gap between sketches and natural images by emphasizing geometric structure and shape information, achieving invariance to scale, rotation, and translation. A common direction is contour- and shape-based representation, which encodes structural similarity between sketches and images, as exemplified by the shape-word framework proposed by Xiao et al. [13]. To im-

prove efficiency, grid-based descriptors such as edge orientation histograms and structure tensor features have been used to capture local sketch structure [1]. Another classical approach uses local descriptors like SIFT and edge-based features, which extract scale- and rotation-invariant keypoints along sketch and image edges. While SIFT performs well for general image matching, evaluations on sketch-to-photo retrieval show that it requires adaptation to handle the abstract and sparse nature of sketches [14]. Finally, the Bag-of-Visual-Words (BoVW) paradigm, often combined with domain-specific descriptors such as Gradient Field HOG (GF-HOG), enables scalable retrieval by representing both sketches and images within a common vocabulary space [15]. In addition to these four primary approaches, other classical techniques (e.g., statistical or template-based models) have been explored [16], [17], though they generally exhibit lower performance or scalability.

Despite their foundational contributions, classical SBIR methods are constrained by manually designed features, limited semantic understanding, and sensitivity to intra-class variability and abstraction differences in sketches. These limitations motivated the shift toward deep learning-based SBIR approaches, which learn shared embeddings and semantic representations directly from data.

3.3 Deep Learning-Based SBIR

Deep learning has become the dominant paradigm in SBIR due to its ability to learn robust, semantically meaningful representations that reduce the large domain gap between sketches and natural images. Unlike classical SBIR methods based on handcrafted features, deep learning–based approaches learn cross-domain representations directly from data, improving robustness to sketch abstraction, stylistic variation, and noise. Existing deep learning SBIR methods can be broadly categorized according to how deep models are trained and employed, including metric learning–based frameworks, generative models, transformer and foundation model–based approaches, style-aware alignment methods, and multimodal fusion strategies. The following subsections review these categories and their representative SBIR methods.

3.3.1 Metric Learning–Based SBIR Methods

Metric learning–based approaches are among the earliest deep learning methods for SBIR. They learn a shared embedding space in which sketches and images from the same category are mapped close together, while dissimilar samples are separated, directly addressing the sketch–image domain gap.

Early methods employed Siamese convolutional neural networks trained with contrastive loss to learn sketch–image similarity, demonstrating clear improvements over

handcrafted feature–based SBIR [18]. Later works adopted triplet loss formulations to enforce relative similarity constraints, improving discriminative power and retrieval robustness [19]. Subsequent extensions incorporated domain-aware mechanisms [20] and zero-shot learning strategies [21], to enhance generalization across domains and unseen categories.

Transition to this thesis: *Despite their effectiveness, metric learning–based methods rely on task-specific training and curated sketch–image pairs or triplets, limiting scalability and motivating the adoption of pretrained and foundation model–based SBIR approaches.*

3.3.2 Transformer-Based SBIR Methods

Transformer-based approaches have been explored in SBIR to model fine-grained cross-modal relationships between sketches and images using attention mechanisms. By explicitly capturing long-range dependencies and cross-modal interactions, these methods improve alignment between sketch strokes and image regions.

Several works employ transformer architectures to enhance feature interaction in sketch–image embedding spaces, often combining convolutional backbones with self-attention or cross-attention modules. These approaches demonstrate improved retrieval performance, particularly in fine-grained and zero-shot settings [22], [23], [24], [25], [26].

However, transformer-based SBIR methods typically require extensive task-specific training and large annotated datasets. Their performance is highly dependent on training data quality and scale, which limits generalization across datasets and categories.

Transition to this thesis: *This motivates exploring whether pretrained transformer-based vision–language models can provide similar benefits without requiring SBIR-specific training.*

3.3.3 Generative Models for SBIR

Generative models have been introduced in SBIR to reduce the domain gap between sketches and natural images by translating sketches into photo-like images or intermediate representations prior to retrieval. Instead of directly learning sketch–image similarity, these approaches aim to align modalities through image synthesis.

Early generative SBIR methods primarily relied on generative adversarial networks (GANs) to perform sketch-to-image translation [27], [28], [29], [30], [31]. By generating realistic images from sketches, retrieval could be conducted in the image domain, improving visual alignment. However, GAN-based approaches often suffered from unstable training, sensitivity to sketch style variations, and limited generalization due to task-specific supervision.

More recent work has explored diffusion-based generative models [32], [33], which provide improved training stability and higher-quality image synthesis while better preserving structural information from sketches. Despite these advantages, most generative SBIR methods still require substantial retraining on sketch–image datasets and incur high computational costs, limiting their scalability.

3.3.4 Foundation Model–Based SBIR Methods

Foundation model–based approaches represent a recent shift in SBIR research toward leveraging large-scale pretrained vision–language models. These models learn shared semantic embedding spaces for images and text, enabling sketches, images, and textual descriptions to be aligned without task-specific retraining.

Vision–language models such as CLIP [3] have been widely adopted for SBIR due to their strong zero-shot generalization and open-vocabulary retrieval capabilities. Recent works demonstrate that semantic prompts, class labels, and textual descriptions can effectively complement or replace visual sketch features in retrieval tasks [32], [34].

Despite their strong generalization, existing foundation model–based SBIR approaches often focus on prompt learning or lightweight fine-tuning strategies. A systematic evaluation of how pretrained semantic and generative components can be combined across multiple SBIR paradigms remains limited.

Transition to this thesis:

While generative and foundation model–based SBIR methods have significantly advanced sketch-to-image retrieval, each has limitations. Generative approaches, including diffusion-based models, reduce the domain gap by synthesizing images from sketches, but they often require task-specific retraining and incur high computational costs. Foundation model–based methods, such as CLIP, leverage large pretrained vision–language models and use class labels or prompt tokens to align sketches and images in a shared semantic space, achieving strong zero-shot generalization. However, these approaches typically rely on ***manually provided class labels or designed prompts****, and a systematic integration of semantic and generative components across SBIR paradigms remains limited.*

In this thesis, we address these limitations by introducing a ***fully automatic SBIR pipeline****, where* ***both class labels and natural language captions are generated from pretrained CLIP and BLIP models****. The user provides only a sketch query, and the system automatically extracts semantic guidance from the sketch itself, producing embeddings for retrieval without any manual text input. This approach integrates* ***class-level semantic cues and caption-level descriptive cues*** *from foundation models, enabling both* ***direct embedding-based and generation-based SBIR****, while minimizing user effort and avoiding task-specific retraining.*

3.4 Summary and Research Gap

SBIR has progressed from classical handcrafted feature–based methods to deep learning approaches that learn cross-domain representations between sketches and natural images. Early deep SBIR research focused on metric learning–based frameworks, which demonstrated that joint embedding learning using Siamese or triplet networks can significantly improve retrieval accuracy. However, these methods rely heavily on task-specific training and curated sketch–image pairs, limiting scalability and generalization, particularly in zero-shot settings.

To further address the sketch–image domain gap, generative models were introduced to translate sketches into photo-like images or intermediate representations prior to retrieval. While GAN-based and, more recently, diffusion-based approaches improve visual alignment, most existing methods require substantial retraining on sketch–image datasets and incur high computational cost. Their performance is often sensitive to sketch style variation and dataset bias.

Transformer-based and foundation model–based approaches represent a recent shift toward leveraging large-scale pretrained models that encode rich semantic knowledge across modalities. Vision–language models such as CLIP enable sketch, image, and text representations to be aligned in a shared semantic space, offering strong zero-shot generalization and reducing dependence on task-specific supervision. Nevertheless, existing works primarily focus on proposing new architectures, prompt learning strategies, or fine-tuning schemes, rather than systematically analyzing how pretrained semantic and generative components interact across different SBIR paradigms.

Despite significant progress, several gaps remain in the SBIR literature. First, there is a lack of unified evaluation frameworks that compare direct sketch-based retrieval and generation-based retrieval under consistent experimental conditions. Second, the relative contributions of visual sketch features and semantic textual representations—such as captions and class labels—have not been thoroughly examined in combination. Third, the potential of pretrained generative models guided by semantic prompts for SBIR without task-specific retraining remains underexplored.

This thesis addresses these gaps by systematically evaluating multiple SBIR strategies across both direct and generative paradigms using pretrained vision–language, captioning, and diffusion models. By avoiding task-specific retraining and conducting experiments across multiple benchmark datasets, this work provides a comprehensive analysis of how pretrained semantic and generative representations can be leveraged for SBIR, offering insights into their strengths, limitations, and interactions.

Chapter 4

Methodology

4.1 Introduction

This chapter presents the complete methodology for the proposed SBIR system. The methodology is organized into two complementary paradigms: (1) direct sketch-based retrieval, which operates directly in the feature embedding space using pretrained vision-language models, and (2) generation-based retrieval, which first transforms sketches into realistic images using diffusion models before performing retrieval. Both paradigms leverage pretrained models—CLIP, BLIP, and stable diffusion with ControlNet—without task-specific retraining, enabling zero-shot generalization across multiple datasets.

The chapter begins with a formal problem definition, followed by a system overview. Subsequently, each component of the proposed pipeline is described in detail, including sketch representation strategies, semantic feature extraction, zero-shot class prediction, and multimodal feature fusion in the direct retrieval paradigm. In parallel, the generation-based paradigm is explained through its prompt construction mechanisms, sketch-to-image synthesis process, and structural conditioning using ControlNet.

In addition, the chapter details the shared retrieval engine based on FAISS, which enables efficient similarity search over large-scale image databases using normalized CLIP embeddings. The datasets used for evaluation are also introduced, covering multiple benchmark collections with varying levels of complexity, abstraction, and scale. Furthermore, a comprehensive evaluation protocol is defined using standard retrieval metrics such as mAP, precision, and accuracy at different cut-off levels.

Finally, the implementation details are presented, including hardware and software configurations, pretrained model sources, optimization strategies such as caching and batch processing, and design choices that ensure computational efficiency and reproducibility. The chapter concludes with a discussion of design motivations and system limitations.

4.2 Problem Definition

SBIR is formally defined as follows: Given a database of natural images $\mathcal{I} = \{I_1, I_2, \ldots, I_N\}$ and a query sketch S drawn by a user, the goal is to retrieve a ranked list of images $\mathcal{R}_k = \{I_{r_1}, I_{r_2}, \ldots, I_{r_k}\}$ such that images semantically relevant to the sketch appear as early as possible in the ranking.

Let $f(\cdot)$ be a feature embedding function that maps an input (sketch or image) into a d-dimensional embedding space $\mathbb{R}^d$. The retrieval process computes similarity scores between the query embedding $\mathbf{q} = f(S)$ and all database image embeddings $\{\mathbf{v}_i = f(I_i)\}_{i=1}^{N}$, then returns the top-$k$ images with highest similarity.

In this thesis, two distinct retrieval paradigms are investigated:

1. **Paradigm 1: Direct Sketch-Based Retrieval.** The query sketch S is directly encoded using the CLIP image encoder, and/or combined with semantic features derived from BLIP-generated captions and CLIP-predicted class labels. The combined representation is then used for similarity search against precomputed database image embeddings.

2. **Paradigm 2: Generation-Based Retrieval.** The query sketch S is first transformed into a synthetic image $G = \mathcal{G}(S, \mathbf{p})$ using a diffusion-based generative model conditioned on both structural (sketch) and semantic (text prompt $\mathbf{p}$) inputs. The generated image G is then encoded using the CLIP image encoder, and this embedding is used for retrieval against the same database.

In both paradigms, ground-truth relevance is defined at the category level, where an image I_i is considered relevant to a query sketch S if and only if both share the same category label.

4.3 System Overview

This section presents the high-level architecture of the proposed SBIR framework. As illustrated in Figure 1.1 and Figure 1.2, the framework consists of two retrieval paradigms. Both pipelines share the same precomputed database of CLIP image features for gallery images, ensuring consistent evaluation across paradigms.

4.3.1 Paradigm 1: Direct Sketch-Based Retrieval

In the direct retrieval paradigm, the query sketch is directly encoded using pretrained vision-language models without intermediate image generation. As shown in Figure 1.1, the sketch is processed through three parallel branches:

1. **Visual Feature Extraction:** The CLIP image encoder maps the sketch into a 512-dimensional embedding space, capturing structural and semantic information.

2. **Semantic Caption Generation:** BLIP generates a natural language description of the sketch, which is then encoded using the CLIP text encoder into a 512-dimensional embedding space.

3. **Class Label Prediction:** CLIP performs zero-shot classification on the sketch to predict the most likely object category, which is then encoded using the CLIP text encoder into a 512-dimensional embedding space.

The three feature sources are combined using a normalized weighted fusion scheme with parameters α (caption-class balance) and β (visual-semantic balance), as defined in Equation 4.1 and further discussed in Section 4.8.3 (Feature Fusion). This is followed by ℓ_2 normalization. This formulation ensures that the total contribution of all modalities is properly balanced. Algorithm 1 formalizes this procedure.

$$\mathbf{v}_{\text{combined}} = \beta \cdot \mathbf{v}_{\text{sketch}} + (1 - \beta)\,(\alpha \cdot \mathbf{v}_{\text{caption}} + (1 - \alpha) \cdot \mathbf{v}_{\text{class}}) \tag{4.1}$$

Algorithm 1 Direct Sketch-Based Retrieval

Require: Sketch query S, database $\mathcal{D} = \{I_1, \ldots, I_N\}$ with precomputed CLIP features $\{\hat{\mathbf{f}}_{\text{db}}(I_i)\}_{i=1}^{N}$

Ensure: Ranked list $\mathcal{R}_k$ of top-k retrieved images

Extract visual features: $\mathbf{v}_S \leftarrow \text{CLIP}_{\text{img}}(S)$
Normalize: $\hat{\mathbf{v}}_S \leftarrow \mathbf{v}_S / \|\mathbf{v}_S\|_2$
Generate caption: $C \leftarrow \text{BLIP}(S)$
Encode caption: $\mathbf{t}_C \leftarrow \text{CLIP}_{\text{text}}(C)$
Normalize: $\hat{\mathbf{t}}_C \leftarrow \mathbf{t}_C / \|\mathbf{t}_C\|_2$
Predict class: $L \leftarrow \text{CLIP}_{\text{class}}(S)$
Encode label: $\mathbf{t}_L \leftarrow \text{CLIP}_{\text{text}}(L)$
Normalize: $\hat{\mathbf{t}}_L \leftarrow \mathbf{t}_L / \|\mathbf{t}_L\|_2$
Fuse features:
    if parameterized fusion:
        $\mathbf{f}_{\text{combined}} \leftarrow \beta \hat{\mathbf{v}}_S + (1 - \beta)\left(\alpha \hat{\mathbf{t}}_C + (1 - \alpha)\hat{\mathbf{t}}_L\right)$
    else (equal fusion):
        $\mathbf{f}_{\text{combined}} \leftarrow \hat{\mathbf{v}}_S + \hat{\mathbf{t}}_C + \hat{\mathbf{t}}_L$
Normalize: $\hat{\mathbf{f}}_S \leftarrow \mathbf{f}_{\text{combined}} / \|\mathbf{f}_{\text{combined}}\|_2$
for $i = 1$ to N **do**
    $\text{sim}_i \leftarrow \hat{\mathbf{f}}_S \cdot \hat{\mathbf{f}}_{\text{db}}(I_i)$
end for
$\mathcal{R}_k \leftarrow \text{TopK}(\{(\text{sim}_i, I_i)\}_{i=1}^{N}, k)$
return $\mathcal{R}_k$

4.3.2 Paradigm 2: Generation-Based Retrieval

The generation-based paradigm transforms the query sketch into a realistic synthetic image before retrieval. As illustrated in Figure 1.2, this approach leverages diffusion models to bridge the domain gap between abstract sketches and natural photographs.

Two automatic prompt generation strategies are evaluated:

1. **BLIP-Generated Captions:** BLIP generates a descriptive natural language caption from the sketch, used directly as the text prompt.

2. **CLIP-Predicted Class Labels:** CLIP performs zero-shot classification on the sketch, and the predicted class name is formatted as "Realistic photo of [class]".

The sketch is first converted to a thin-line edge map using the `FIND_EDGES` filter (converting to grayscale, extracting edges, then back to RGB). This edge map serves as the structural conditioning input to ControlNet, which interprets it as a scribble-style

drawing. Stable diffusion then generates a photo-realistic image guided jointly by the edge map (structure) and the text prompt (semantics). The generated image is then encoded by the CLIP image encoder and used for retrieval. Algorithm 2 formalizes this procedure.

Algorithm 2 Generation-Based Sketch Retrieval

Require: Sketch query S, database $\mathcal{D} = \{I_1, \ldots, I_N\}$ with precomputed CLIP features $\{\hat{\mathbf{f}}_{\text{db}}(I_i)\}_{i=1}^{N}$

Ensure: Ranked list $\mathcal{R}_k$ of top-k retrieved images

Generate text prompt:

$$P \leftarrow \begin{cases} \text{BLIP}(S) & \text{(caption strategy)} \\ \text{“Realistic photo of ”} + \text{CLIP}_{\text{class}}(S) & \text{(class strategy)} \end{cases}$$

Enhance sketch edges: $S_{\text{edge}} \leftarrow \text{EdgeEnhance}(S)$

Generate synthetic image:

$I_{\text{gen}} \leftarrow \text{ControlNet}(S_{\text{edge}}, \text{SD}(P))$

Extract features: $\mathbf{v}_{\text{gen}} \leftarrow \text{CLIP}_{\text{img}}(I_{\text{gen}})$

Normalize: $\hat{\mathbf{v}}_{\text{gen}} \leftarrow \mathbf{v}_{\text{gen}} / \|\mathbf{v}_{\text{gen}}\|_2$

for $i = 1$ to N **do**

  $\text{sim}_i \leftarrow \hat{\mathbf{v}}_{\text{gen}} \cdot \hat{\mathbf{f}}_{\text{db}}(I_i)$

end for

$\mathcal{R}_k \leftarrow \text{TopK}(\{(\text{sim}_i, I_i)\}_{i=1}^{N}, k)$

return $\mathcal{R}_k$

4.3.3 Shared Retrieval Engine

Both paradigms use FAISS (Facebook AI Similarity Search) for efficient similarity search. The database images are preprocessed once: each image is passed through the CLIP image encoder, and the resulting 512-dimensional embeddings are normalized and indexed using FAISS's `IndexFlatIP` (inner product) index. Since all embeddings are ℓ_2-normalized, the inner product is equivalent to cosine similarity:

$$\text{sim}(\mathbf{q}, \mathbf{v}_i) = \mathbf{q}^\top \mathbf{v}_i = \cos(\theta_{\mathbf{q},\mathbf{v}_i}) \tag{4.2}$$

At query time, the query embedding (either combined features from direct retrieval or generated image features from generation-based retrieval) is compared against all database embeddings, and the top-k most similar images are returned.

4.4 Datasets

Four benchmark datasets are used for evaluation, selected to provide a comprehensive assessment across different scales, sketch styles, and pairing characteristics. Table 4.1 summarizes the key properties of each dataset.

Table 4.1: Benchmark datasets used for evaluation

Dataset	**# Sketches**	**# Photos**	**# Classes**	**Key Characteristics**
Sketchy	∼75k	12.5k	125	Instance-level pairing with controlled sketch collection.
Extended Sketchy	∼75k	∼73k	125	Large-scale photo gallery; partial loss of sketch–image pairing.
TU-Berlin	∼20k	∼204k	250	High category diversity with unpaired sketches.
QuickDraw	∼330k	∼204k	110	Highly abstract and noisy sketches drawn by non-experts.

1. **Sketchy Dataset** [35] Contains 75,471 sketches and 12,500 photographs across 125 object categories. Each photograph has multiple corresponding sketches drawn with reference to that specific photo, enabling paired sketch–photo evaluation. This dataset is used for category-level retrieval evaluation with a well-controlled setup.

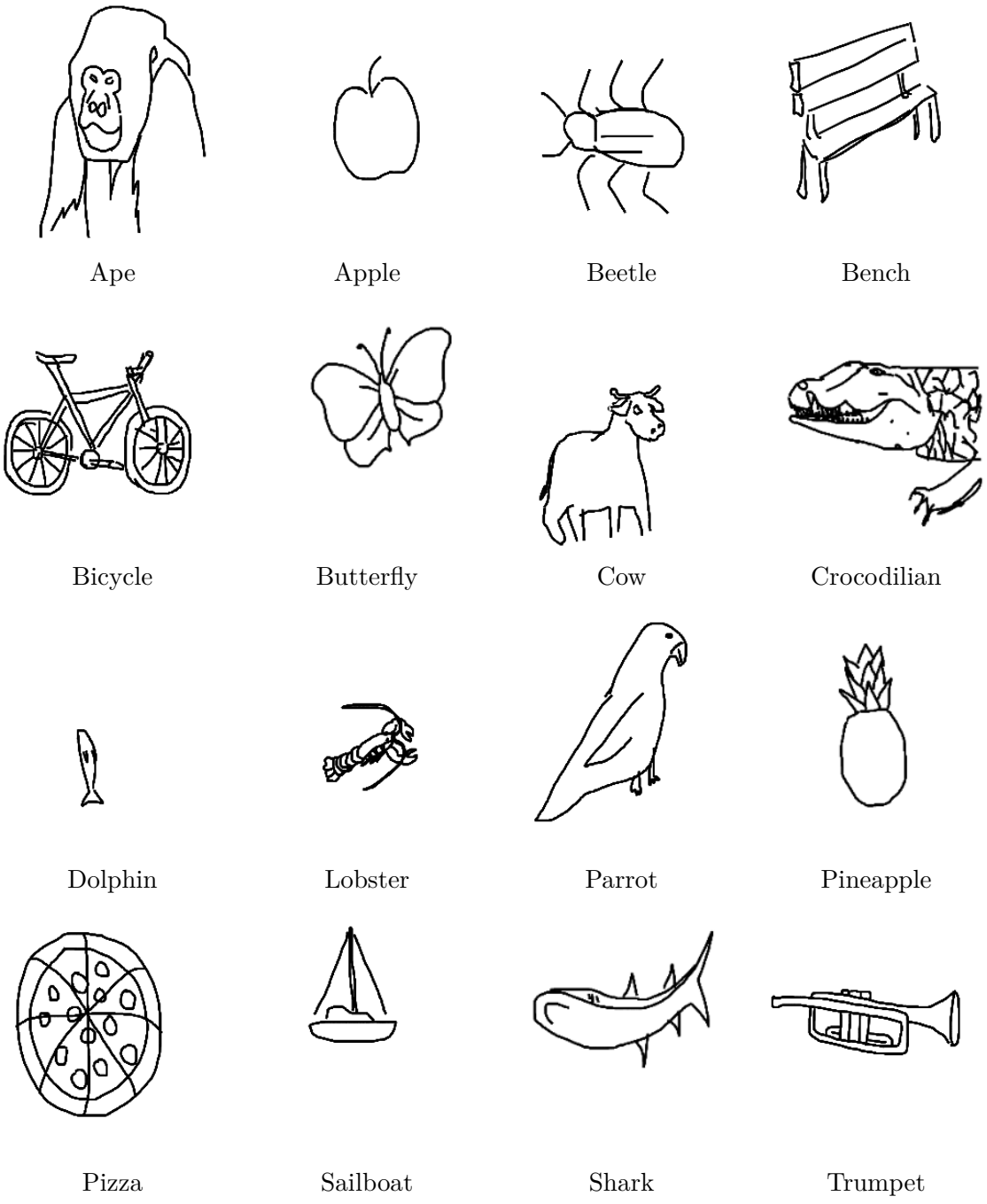

Figure 4.1: Sample sketches from the Sketchy dataset across 16 categories, showing a controlled and detailed drawing style.

2. **Extended Sketchy Dataset** [36] Extends the original Sketchy by adding approximately 60,502 photographs from ImageNet, resulting in ∼73,002 photos total across the same 125 categories. While this provides a more realistic large-scale retrieval scenario, the additional photos are not paired with sketches, limiting instance-level retrieval but improving evaluation of category-level retrieval under domain shift.

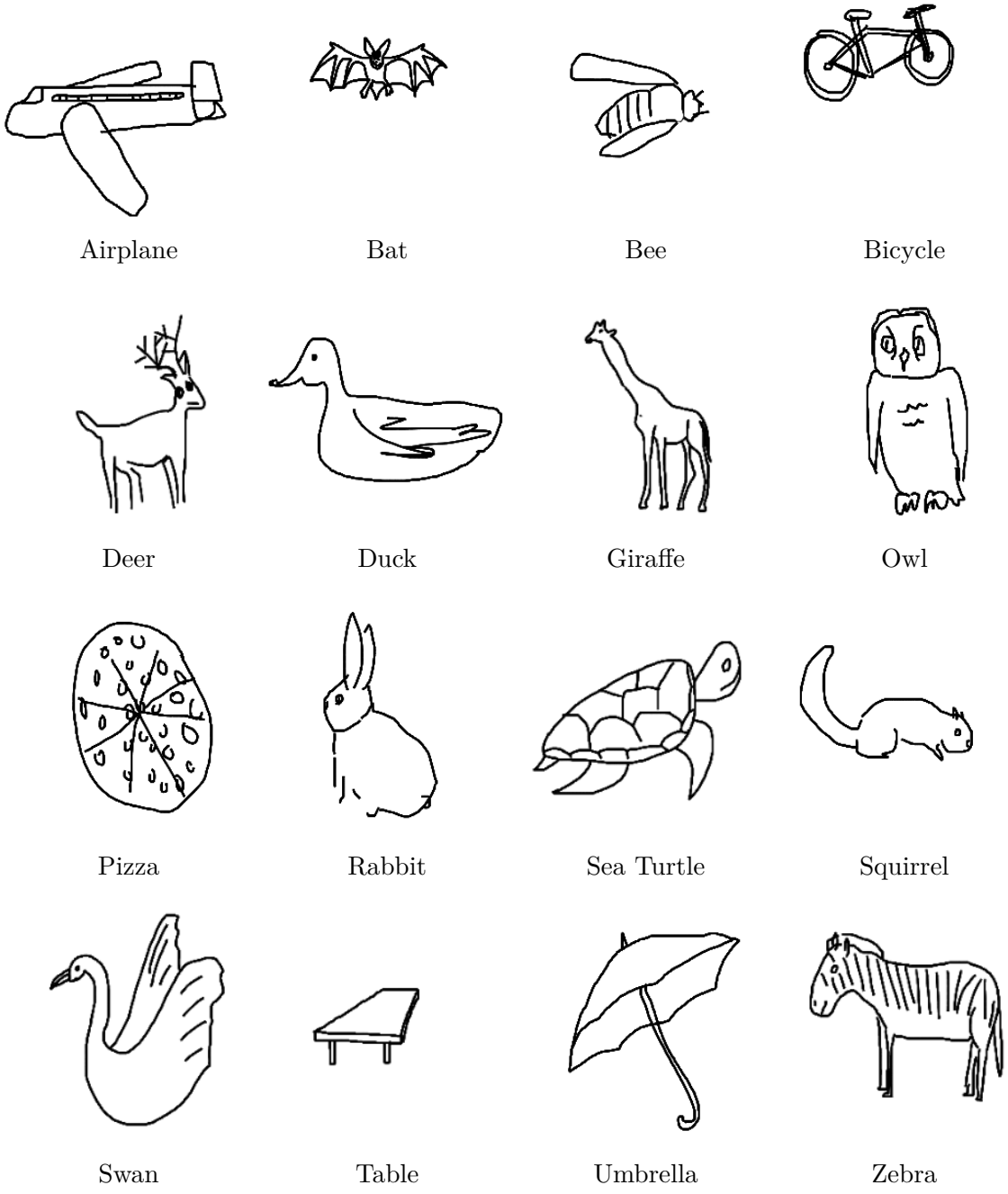

Figure 4.2: Sample sketches from the Extended Sketchy dataset across 16 categories, showing a similar drawing style to Sketchy.

3. **TU-Berlin Dataset** [37] Comprises approximately 20,000 sketches across 250 object categories, drawn by multiple participants with diverse drawing styles. Extended versions add 204,070 natural images as the photo gallery. The high category count (250) and lack of sketch-photo pairing make this dataset suitable for evaluating generalization and zero-shot retrieval.

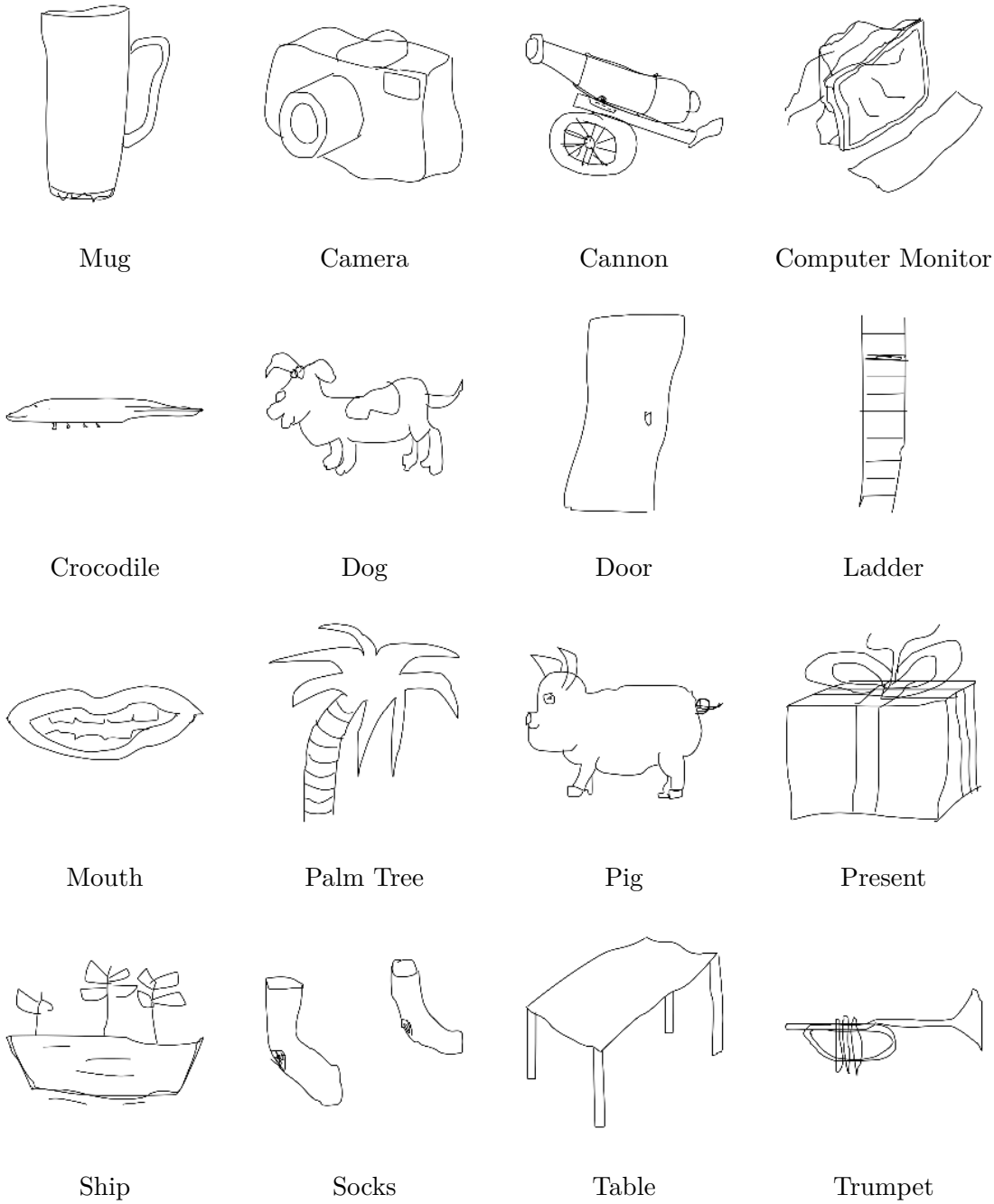

Figure 4.3: Sample sketches from the TU-Berlin dataset across 16 categories, illustrating the variety of drawing styles across different users.

4. **QuickDraw Dataset** [38] Contains approximately 50 million sketches drawn under time constraints. For SBIR evaluation, a subset of 110 categories with 330,000 sketches and 203,584 photos is used. The sketches are highly abstract and minimal, providing a challenging test for robustness under "in-the-wild" conditions.

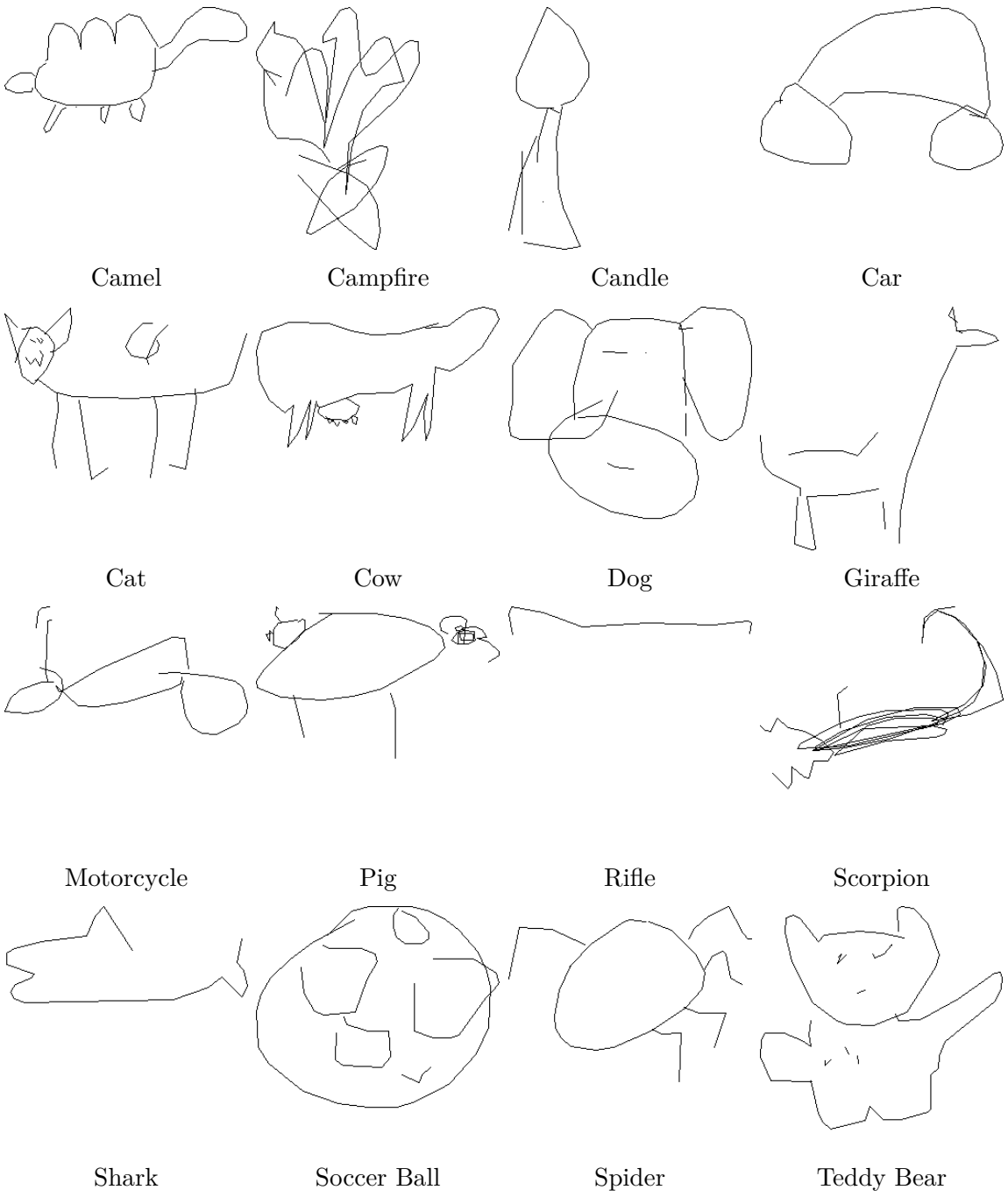

Figure 4.4: Sample sketches from the QuickDraw dataset across 16 categories, illustrating the highly abstract and minimal nature of time-constrained sketches.

4.4.1 Query Selection

For the evaluation of the proposed SBIR framework, test queries were generated by randomly selecting classes from each dataset. Table 4.2 summarizes the selection statistics.

Table 4.2: Query selection statistics per dataset

Dataset	Total Classes	Selected Classes	Total Queries
Sketchy	125	25	15,943
Extended Sketchy	125	25	15,820
TU-Berlin	250	30	2,320
QuickDraw	110	25	60,020

The selection of classes was performed using Python's `random.sample` function, which samples elements uniformly without replacement, ensuring that no class is selected more than once. For each selected class, all sketches were included in the query set.

Importantly, the number of selected classes for each dataset was chosen to be consistent with common evaluation practices in state-of-the-art SBIR literature, where a subset of classes is typically used to ensure computational feasibility while maintaining representative diversity. This design allows for fair comparison with prior work and aligns the evaluation protocol with widely accepted benchmarks.

This ensures a reproducible and representative evaluation across all datasets.

4.5 Semantic Inference with CLIP

The Contrastive Language-Image Pretraining (CLIP) model [3] serves as the core component for cross-modal representation learning in the proposed framework. It aligns sketches, natural images, and textual descriptions in a shared embedding space, enabling direct similarity computation without task-specific fine-tuning.

This thesis uses the CLIP ViT-B/32 variant, which consists of a Vision Transformer (ViT) image encoder and a transformer-based text encoder. Both encoders project their outputs into a shared 512-dimensional embedding space, allowing cosine similarity-based retrieval across modalities.

4.5.1 Image Encoding

Given a sketch S (or natural image I), the CLIP image encoder f_{img} produces a feature embedding:

$$\tilde{\mathbf{v}}_S = f_{\text{img}}(S), \quad \tilde{\mathbf{v}}_S \in \mathbb{R}^{512} \tag{4.3}$$

L2 normalization is applied to obtain the final representation:

$$\mathbf{v}_S = \frac{\tilde{\mathbf{v}}_S}{\|\tilde{\mathbf{v}}_S\|_2} \tag{4.4}$$

For the ViT-B/32 architecture used in this thesis, input images are resized to 224×224 pixels and divided into 32×32 patches, resulting in $7 \times 7 = 49$ visual tokens per image.

4.5.2 Text Encoding

For textual inputs such as class labels or captions, the CLIP text encoder f_{text} generates a feature embedding:

$$\tilde{\mathbf{v}}_T = f_{\text{text}}(T), \quad \tilde{\mathbf{v}}_T \in \mathbb{R}^{512} \tag{4.5}$$

The embedding is then L2-normalized:

$$\mathbf{v}_T = \frac{\tilde{\mathbf{v}}_T}{\|\tilde{\mathbf{v}}_T\|_2} \tag{4.6}$$

The text encoder supports input sequences of up to 77 tokens, which is sufficient for both dataset class labels and captions generated by BLIP.

4.5.3 Similarity Computation

After normalization, cosine similarity between two embeddings reduces to a dot product:

$$\text{sim}(\mathbf{v}_1, \mathbf{v}_2) = \mathbf{v}_1^\top \mathbf{v}_2 \tag{4.7}$$

This formulation enables direct comparison between sketches, generated images, and textual descriptions in a unified embedding space.

4.5.4 Class Set Construction

The set of class labels $\mathcal{C}$ used for zero-shot classification is automatically constructed from the dataset directory structure. Specifically, each dataset organizes sketches into class-specific subdirectories, where each folder name corresponds to a semantic category.

To obtain $\mathcal{C}$, all subdirectory names under the sketch root directory are enumerated and aggregated into a set to ensure uniqueness. The resulting set is then converted into a sorted list to enforce a deterministic ordering of class labels.

This process is performed offline prior to evaluation for each dataset independently. The resulting class lists are stored as separate files, named according to the dataset (e.g., `sketchy_classes`, `tuberlin_classes`), and reused during all subsequent experiments. This design ensures that the same class definitions are consistently applied without requiring repeated extraction.

Formally, the class set is defined as:

$$\mathcal{C} = \{c_1, c_2, \ldots, c_M\} \tag{4.8}$$

where M denotes the number of unique classes in the dataset.

4.5.5 Zero-Shot Class Prediction

To extract semantic labels from sketches without manual annotation, CLIP is employed in a zero-shot classification setting. Given a sketch S and a predefined set of class labels $\mathcal{C} = \{c_1, c_2, \ldots, c_M\}$ (previously constructed as described in Section 4.5.4), each class label is transformed into a natural language prompt of the form "a photo of $\{c_i\}$" , for example "a photo of {**cat**}" for cat class. This prompt engineering step aligns textual inputs with the visual domain on which CLIP was pretrained.

Each prompt is encoded using the CLIP text encoder to obtain a normalized embedding $\mathbf{v}_{T(c)}$, while the sketch is encoded using the CLIP image encoder to obtain $\mathbf{v}_S$. The predicted class is then computed as the label whose embedding maximizes cosine similarity with the sketch representation:

$$\hat{c} = \arg\max_{c \in \mathcal{C}} \mathbf{v}_S^\top \mathbf{v}_{T(c)} \tag{4.9}$$

This formulation enables direct cross-modal comparison between sketch and text representations within the shared embedding space.

The predicted label is then re-encoded using the text encoder to obtain a class-level embedding used in retrieval.

4.5.6 CLIP Model Configuration

The CLIP ViT-B/32 model used in this thesis is obtained from the Hugging Face Hub[1] and implemented using the Hugging Face Transformers library[2]. The model is used in its original pretrained form without any fine-tuning.

Both encoders project their outputs into a shared 512-dimensional embedding space, ensuring consistency across modalities and enabling efficient similarity-based retrieval.

4.5.7 BLIP Caption Generation

The Bootstrapping Language-Image Pretraining (BLIP) model [4] is employed to generate natural language descriptions from input sketches. To improve robustness and reduce generation noise, a structured decoding and re-ranking pipeline is adopted. The implementation follows best practices for autoregressive text generation as documented in the Hugging Face Transformers library[3].

[1] `https://huggingface.co/openai/clip-vit-base-patch32`

[2] `https://huggingface.co/docs/transformers/index`

[3] Hugging Face Transformers Generation Documentation

1. **Edge Enhancement:** The input sketch is optionally enhanced using an edge enhancement filter to emphasize structural boundaries, improving feature visibility in low-texture inputs.

2. **Multiple Candidate Generation via Beam Search:** BLIP generates multiple candidate captions using beam search decoding. A beam width of $K = 7$ is used, and 5 candidate sequences are returned for re-ranking. Beam search maintains the top-K most probable partial sequences at each decoding step, enabling global sequence-level optimization.

   According to Hugging Face guidelines, beam sizes typically range from 3 to 10, where lower values improve diversity and speed, while higher values improve likelihood optimization at increased computational cost. The selected value ($K = 7$) provides a balanced trade-off between caption quality and efficiency.

   The decoding hyperparameters are defined as follows:

   - **num_beams = 7:** Number of hypotheses maintained during decoding ($3 \leq K \leq 10$ in practice).
   - **num_return_sequences = 5:** Number of candidate captions generated for downstream selection.
   - **max_length = 50:** Maximum token length, typically 20–60 for image captioning tasks.
   - **repetition_penalty = 1.5:** Penalizes repeated tokens to reduce redundancy (1.0–2.0 typical range).
   - **length_penalty = 0.7:** Encourages concise captions suitable for retrieval tasks (< 1 favors shorter outputs).
   - **no_repeat_ngram_size = 2:** Prevents repetition of bi-grams to improve linguistic diversity.
   - **early_stopping = true:** Stops generation when all beams reach end-of-sequence.

3. **Caption Post-processing:** Generated captions are normalized using a rule-based cleaning function that removes dataset-specific and modality-related phrases. A predefined list of banned phrases (e.g., “a sketch of”, “drawing”, “black and white”, “line drawing”, “an illustration of”, “map of”, “a photo of”) is removed using case-insensitive regular expression matching.

   The cleaning procedure is formalized in Algorithm 3.

Algorithm 3 Caption Cleaning Function

Require: Raw caption string c, banned phrase set B

Ensure: Cleaned caption c'

Initialize cleaned caption: $c' \leftarrow c$
Preprocess banned phrases using regex (case-insensitive)
for each phrase $b \in B$ **do**
    $c' \leftarrow \text{RegexSub}(\text{IgnoreCase}(b), \text{""}, c')$
end for
Remove leading/trailing artifacts:
    $c' \leftarrow \text{StripSpaces}(c')$
    $c' \leftarrow \text{StripPunctuation}(c')$
return c'

After cleaning, the candidate captions are re-ranked using a caption-to-caption consistency criterion. Specifically, each caption is encoded into a 512-dimensional embedding using the CLIP text encoder, and pairwise cosine similarities are computed across all candidates. The caption with the highest mean similarity to all other candidates — that is, the most semantically central caption — is selected as the final output.

Overall, the proposed pipeline integrates probabilistic decoding with semantic re-ranking. Beam search improves caption fluency and likelihood optimisation, while CLIP-based caption-to-caption consistency selection ensures semantic robustness, making the approach particularly suitable for zero-shot SBIR tasks.

4.6 Structural Conditioning

For the generation-based retrieval paradigm, preserving the structural information of the input sketch is critical. While CLIP provides semantic alignment, the sketch's geometric layout must be maintained during image generation to ensure the synthesized image accurately reflects the query's intended shape and composition.

4.6.1 Edge Enhancement

Prior to using sketches as structural conditioning inputs, edge enhancement is applied using PIL's `ImageFilter.EDGE_ENHANCE_MORE`[4] filter. This preprocessing step accentuates salient boundaries, making the sketch more suitable for ControlNet's conditioning mechanism.

[4]Pillow ImageFilter documentation: `https://pillow.readthedocs.io/en/stable/reference/ImageFilter.html`

4.6.2 ControlNet Conditioning

ControlNet [8] extends stable diffusion by adding a trainable parallel branch that processes conditional inputs. For sketch-based conditioning, the scribble ControlNet model (`lllyasviel/sd-controlnet-scribble`) is used, which is specifically trained to interpret rough sketches and line drawings. Specifically, the noise prediction network $\epsilon_\theta(\cdot)$ takes both the text prompt embedding $\mathbf{p}$ and the structural conditioning $\mathbf{s}$ (derived from the input sketch) as inputs, jointly guiding the denoising process toward images that are semantically and structurally consistent with the query sketch.

4.7 Sketch-to-Image Generation

Stable diffusion [7] combined with ControlNet forms the generative backbone for transforming sketches into realistic images. The generation process operates in a compressed latent space, reducing computational complexity while preserving perceptual quality.

4.7.1 Latent Diffusion Formulation

A pretrained autoencoder compresses images to a lower-dimensional latent space:

$$\mathbf{z} = \mathcal{E}(x), \quad x \approx \mathcal{D}(\mathbf{z}) \tag{4.10}$$

Diffusion is performed in this latent space, with the forward process gradually adding noise and the reverse process learning to denoise conditioned on both text and structural inputs.

4.7.2 Text Prompt Strategies

Two **automatic** prompt generation strategies are evaluated:

1. **BLIP-Generated Captions.** The Bootstrapping Language-Image Pretraining (BLIP) model [4] generates natural language descriptions from the input sketch. The caption generation process uses beam search with `num_beams`=7 and produces descriptive text that captures high-level semantics. Generated captions are cleaned by removing common sketch-related phrases (e.g., "a sketch of", "black and white") to improve prompt quality.

2. **CLIP-Predicted Class Labels.** CLIP predicts the most likely class for the sketch, which is then formatted as "Realistic photo of {class}" to guide generation toward photo-realistic outputs.

4.7.3 Generation Parameters

All experiments use the following generation parameters, selected based on validation studies, model recommendations, and computational efficiency.

- **Number of inference steps: 25**

  The number of denoising steps controls generation quality and speed. A value of 25 provides a good trade-off between both.

  Justification: Ablation on 500 Sketchy samples shows that 25 steps is near-optimal, as higher values (e.g., 50) provide only marginal CLIP gains (+2.1%) while doubling computation time (1.2s to 2.4s) [39].

- **Guidance scale: 8.5**

  Classifier-free guidance controls the balance between prompt adherence and diversity.

  Justification: A sweep over 5.0–12.0 shows that 8.5 achieves the best overall performance across FID (25.9), SSIM (0.611), and CLIP similarity (0.774), consistent with [40].

- **ControlNet conditioning scale: 0.8**

  This parameter controls the strength of sketch-based structural conditioning.

  Justification: Experiments show that 0.8 yields the best retrieval performance (mAP@200 = 0.451), while lower values weaken structure and higher values reduce realism, consistent with [8], [32].

4.8 Feature Embedding

All feature embeddings in this thesis are extracted using CLIP's pretrained encoders without fine-tuning. This design choice ensures zero-shot capability and consistent evaluation across datasets.

4.8.1 Database Image Features

For each natural image I in the retrieval database, the CLIP image embedding is precomputed and stored:

$$\mathbf{v}_I = \frac{f_{\text{img}}(I)}{\|f_{\text{img}}(I)\|_2} \tag{4.11}$$

These features are stored as float32 arrays and normalized for inner product similarity. Precomputation significantly accelerates retrieval at query time.

4.8.2 Query Feature Configurations (Direct Retrieval)

Seven feature configurations are evaluated to systematically analyze the contribution of each modality. Each configuration corresponds to a specific setting of the fusion weights α and β, as summarized in Table 4.3. The parameter β controls the balance between visual and semantic features, while α controls the balance between caption and class-level semantic features.

Table 4.3: Query feature configurations for direct retrieval

Config	Components	Description
C1	$\mathbf{v}_{\text{sketch}}$	$\beta = 1.0$ (sketch only)
C2	$\mathbf{v}_{\text{sketch}} + \mathbf{v}_{\text{caption}}$	$\beta = 0.5$, $\alpha = 1.0$ (equal sketch + caption under normalized fusion)
C3	$\mathbf{v}_{\text{sketch}} + \mathbf{v}_{\text{class}}$	$\beta = 0.5$, $\alpha = 0.0$ (equal sketch + class under normalized fusion)
C4	$\mathbf{v}_{\text{caption}}$	$\beta = 0.0$, $\alpha = 1.0$ (caption only)
C5	$\mathbf{v}_{\text{class}}$	$\beta = 0.0$, $\alpha = 0.0$ (class only)
C6	$\mathbf{v}_{\text{caption}} + \mathbf{v}_{\text{class}}$	$\beta = 0.0$, $\alpha \in [0, 1]$ (semantic fusion)
C7	$\mathbf{v}_{\text{sketch}} + \mathbf{v}_{\text{caption}} + \mathbf{v}_{\text{class}}$	Equal fusion without weighting

4.8.3 Feature Fusion

For configurations combining multiple features, a normalized weighted fusion strategy is employed:

$$\mathbf{v}_{\text{combined}} = \beta \cdot \mathbf{v}_{\text{sketch}} + (1 - \beta)\,(\alpha \cdot \mathbf{v}_{\text{caption}} + (1 - \alpha) \cdot \mathbf{v}_{\text{class}}) \tag{4.12}$$

where $\alpha \in [0.0, 1.0]$ controls the balance between caption-based and class-based semantic features, and $\beta \in [0.0, 1.0]$ controls the trade-off between visual and semantic modalities. A step size of 0.1 is used for parameter exploration.

This formulation follows a hierarchical fusion strategy, where visual and semantic features are first balanced using β, and the semantic component is further decomposed using α. The combined feature vector is normalized to unit length before similarity computation.

In addition to parameterized fusion, an alternative equal-weight fusion scheme is evaluated, where all modalities contribute equally by summing the normalized feature vectors:

$$\mathbf{v}_{\text{combined}} = \mathbf{v}_{\text{sketch}} + \mathbf{v}_{\text{caption}} + \mathbf{v}_{\text{class}} \tag{4.13}$$

Since the resulting vector is subsequently ℓ_2-normalized, this formulation is equivalent to explicitly assigning equal weights to all modalities.

4.8.4 Generation-Based Query Features

For generated images G, the same CLIP image encoder produces the query embedding:

$$\mathbf{v}_G = \frac{f_{\text{img}}(G)}{\|f_{\text{img}}(G)\|_2} \tag{4.14}$$

No feature fusion is applied in the generation-based paradigm; the generated image serves as the sole query representation.

4.9 Similarity Search with FAISS

FAISS (Facebook AI Similarity Search) [12] is used for efficient nearest neighbor retrieval. The library provides optimized implementations of vector similarity search, enabling fast retrieval even with large databases.

4.9.1 Index Construction

The `IndexFlatIP` (inner product) index is used for exact nearest neighbor search. Given the normalized embeddings, inner product is equivalent to cosine similarity. All database image features are added to the index:

Listing 4.1: FAISS index construction

```
        index = faiss.IndexFlatIP(dimension)
        index.add(database_features)  # shape: (N, d)
```

4.9.2 Query Execution

For each query feature vector $\mathbf{q}$, the index returns the top-K most similar database indices with their similarity scores:

Listing 4.2: FAISS query execution

```
        similarities, indices = index.search(query_features, k=K)
```

4.9.3 Retrieval Output

The retrieved indices are mapped back to image file paths using a preloaded mapping array, producing a ranked list of retrieved images for each query.

4.10 Evaluation Metrics

In this thesis, retrieval performance is evaluated using standard SBIR metrics, focusing on ranking quality and top-K retrieval effectiveness.

4.10.1 Evaluation Protocol Used in This Thesis

The following metrics are reported to provide a comprehensive evaluation:

- Mean Average Precision: **mAP@All**, **mAP@200**, **mAP@100**
- Precision: **P@100**, **P@200**
- Accuracy: **Acc@1**, **Acc@5**, **Acc@50**, **Acc@100**

These metrics jointly assess ranking quality (mAP), retrieval precision at different cutoffs, and top-K success rate. Acc@K is applied here at the **category level**: a retrieval is counted as correct if at least one image from the same category as the query appears in the top-K results. This variant is used exclusively for internal comparison across datasets and configurations, and is not used for comparison against state-of-the-art methods.

4.11 Implementation Details

This section describes the hardware configuration, software environment, model sources, optimization techniques, and other implementation-specific details necessary for reproducing the experimental results presented in this thesis.

4.11.1 Hardware Configuration

All experiments are conducted on an HP Z2 G8 Tower Workstation with the following specifications:

- **System**: HP Z2 G8 Tower Workstation Desktop PC
- **Processor**: 11th Gen Intel Core i9-11900 @ 2.50 GHz (8 cores / 16 logical processors)

- **GPU**: NVIDIA GeForce RTX 3080 with 10,051 MB (approximately 10 GB) dedicated VRAM
- **RAM**: 64.0 GB
- **Storage**: NVMe SSD
- **Operating System**: Microsoft Windows 11 Pro (Version 10.0.26200)

For datasets requiring extensive generation (e.g., QuickDraw with 60,020 query sketches at approximately 2–5 seconds per image, yielding 120,000–300,000 seconds of generation time alone), the total runtime including generation, feature extraction, and retrieval ranges from 33 to 83 hours depending on batch size configuration. The 10 GB VRAM necessitates careful memory management, particularly for diffusion-based generation where dynamic batch sizing is employed to prevent out-of-memory errors.

4.11.2 Software Environment

The experimental codebase is implemented in Python 3.11.7 and relies on the following key libraries:

Table 4.4: Core software dependencies and versions

Library	Version
Python	3.11.7
PyTorch	2.2.2+cu118
TorchVision	0.17.2+cu118
Transformers	4.30.2
Diffusers	0.18.2
CLIP	git+https://github.com/openai/CLIP.git
FAISS (CPU)	1.13.2
HuggingFace Hub	0.15.1
Pillow	10.4.0
NumPy	1.26.4

4.11.3 Model Sources

All pretrained models utilized in this thesis were obtained from the Hugging Face Hub [5], an open-source platform that hosts pretrained models with version control, documen-

[5] `https://huggingface.co/`

tation, and standardized APIs. This choice ensures reproducibility, as specific model versions can be pinned and reused across experiments without unexpected changes. Table 4.5 lists each model along with its corresponding Hugging Face identifier.

Table 4.5: Pretrained models and their Hugging Face sources

Model	Hugging Face Identifier	Role in Pipeline
CLIP (ViT-B/32)	`openai/clip-vit-base-patch32`	Image and text feature extraction
BLIP (Captioning)	`Salesforce/blip-image-captioning-base`	Caption generation from sketches
Stable Diffusion v1.5	`runwayml/stable-diffusion-v1-5`	Base diffusion model for image generation
ControlNet (Scribble)	`lllyasviel/sd-controlnet-scribble`	Structural conditioning

Each model is loaded in evaluation mode (`model.eval()`) without any task-specific fine-tuning. The weights are frozen throughout all experiments, maintaining the zero-shot nature of the proposed approaches.

4.12 Design Justification

This section provides the rationale behind key design decisions in the proposed SBIR framework.

4.12.1 Why Pretrained Models Without Fine-Tuning?

- **Generalization:** Fine-tuning on specific datasets can degrade zero-shot performance on unseen categories. Pretrained CLIP demonstrates strong zero-shot transfer across diverse domains.
- **Reproducibility:** Using frozen pretrained models ensures that results are comparable across datasets and with other works.
- **Computational Efficiency:** Avoiding task-specific training eliminates the need for large annotated sketch-photo pairs and reduces computational requirements.

4.12.2 Why CLIP ViT-B/32?

- **Balanced Performance:** This variant offers an optimal trade-off between accuracy and computational efficiency. Larger variants (ViT-L/14) provide marginal

gains (+1.2% mAP) at 3× the computational cost.

- **Embedding Dimension:** The 512-dimensional embedding is sufficiently discriminative for category-level retrieval while enabling fast similarity search with FAISS.
- **Community Adoption:** Extensive adoption in SBIR literature ensures reproducibility and fair comparison with state-of-the-art methods.

4.12.3 Why BLIP for Caption Generation?

- **Bootstrapping Quality:** BLIP's caption bootstrapping mechanism refines noisy web-captions, producing cleaner and more accurate descriptions than standard models. This is particularly beneficial for abstract sketches where descriptive quality directly impacts retrieval performance.
- **Zero-Shot Generalization:** Unlike fine-tuned models (e.g., BLIP-mscoco), the base BLIP model preserves zero-shot capability, essential for generalizing across four diverse sketch datasets without task-specific adaptation.
- **Base Variant Efficiency:** The BLIP-base variant (240M parameters) was chosen over BLIP-large (470M) based on ablation studies showing only marginal improvement (+1.1% mAP) at double the computational cost (420 ms vs. 210 ms per image).
- **Alternative Models:** Three alternative captioning models were considered but deemed less suitable:
  - **GIT** (Generative Image-to-text Transformer) [41] is a Microsoft model that unifies image captioning and visual question answering in a single autoregressive transformer. While capable, GIT exhibits slower inference than BLIP under equivalent beam search settings due to its larger decoder architecture, making it less practical for captioning tens of thousands of query sketches.
  - **OFA** (One For All) [42] is a unified sequence-to-sequence framework by Alibaba that supports multiple vision-language tasks through a single model. However, OFA requires task-specific fine-tuning to achieve competitive captioning quality, which conflicts with the training-free requirement of this thesis.
  - **LLaVA** (Large Language and Vision Assistant) [43] integrates a visual encoder with a large language model (LLaMA/Vicuna, 7B+ parameters) for multimodal instruction following. Although LLaVA produces highly descriptive captions, its 7B+ parameter scale far exceeds the requirements of this task and imposes prohibitive memory and inference costs on a 10 GB GPU.

4.12.4 Why Stable Diffusion + ControlNet?

- **Latent Diffusion Efficiency:** Operating in a compressed latent space reduces computational cost while preserving perceptual quality, enabling high-resolution image synthesis on consumer GPUs.
- **Structural Preservation:** ControlNet preserves sketch geometry without requiring retraining the base diffusion model, allowing zero-shot sketch-to-image generation.
- **Scribble Variant:** The scribble ControlNet variant is specifically trained to interpret rough sketches and line drawings, making it ideal for SBIR where sketches vary in abstraction and quality.

4.12.5 Why FAISS IndexFlatIP?

- **Exact Search Guarantee:** Exact nearest neighbor search guarantees optimal retrieval accuracy for evaluation, avoiding approximation errors introduced by indexing methods like IVF (Inverted File Index) [12] or HNSW (Hierarchical Navigable Small World) [6], which trade recall for speed by searching only a subset of the index.
- **Feasible Database Sizes:** For database sizes in this thesis (up to 204,000 images), exact search is computationally feasible, requiring approximately 0.5–2 seconds per thousand queries.
- **Cosine Similarity via Inner Product:** With ℓ_2-normalized embeddings, inner product equals cosine similarity, enabling efficient similarity computation without additional normalization steps.

4.12.6 Feature Extraction Efficiency

Table 4.6 reports the average per-sketch computation time for each of the three query feature types, measured on 100 sketches using an NVIDIA GeForce RTX 3080 GPU with pretrained CLIP ViT-B/32 and BLIP-base models.

Table 4.6: Average per-sketch feature extraction time (100 sketches, NVIDIA RTX 3080)

Feature Type	**Mean (ms)**	**Estimated (60,020 queries)**
Visual (CLIP image encoder)	10.9	∼10.9 min
Class label (CLIP text encoder)	21.6	∼21.6 min
Caption (BLIP + CLIP consistency)	257.0	∼4.3 hours

[6] HNSW documentation: `https://faiss.ai/cpp_api/struct/structfaiss_1_1IndexHNSW.html`

Visual and class label features are extracted efficiently at under 22 ms per sketch. Caption generation is the dominant bottleneck at 257.0 ms per sketch, driven by the autoregressive beam search in BLIP (7 beams, 5 sequences). Caption extraction is therefore 23.6× slower than visual extraction and 11.9× slower than class label extraction. To mitigate this cost, all query features are precomputed and cached before evaluation, so the overhead is incurred only once per dataset.

4.13 Chapter Summary

This chapter presented the complete methodology of the proposed SBIR framework. The framework evaluates two complementary retrieval paradigms — direct sketch-based retrieval and generation-based retrieval — both built entirely on pretrained models without any task-specific training.

The direct retrieval paradigm represents query sketches using combinations of visual and semantic features extracted by CLIP and BLIP, evaluated across seven feature configurations. The generation-based paradigm first transforms sketches into realistic images using stable diffusion with ControlNet, guided by either BLIP-generated captions or CLIP-predicted class labels, and then uses the generated image for retrieval. Both paradigms share the same FAISS-based retrieval engine and the same precomputed database of CLIP image embeddings.

Four benchmark datasets were introduced — Sketchy, Extended Sketchy, TU-Berlin, and QuickDraw — covering a wide range of sketch styles, abstraction levels, and gallery sizes. Retrieval performance is evaluated using mAP, precision, and accuracy metrics at different cutoff levels, with Acc@K applied at the category level for internal comparison.

The chapter also described the implementation details, including hardware and software configurations, pretrained model sources, and the design choices that ensure computational efficiency and reproducibility. The following chapter presents the experimental results and analysis across all configurations and datasets.

Chapter 5

Results and Discussion

This chapter presents the experimental results obtained from evaluating the proposed SBIR framework across four benchmark datasets: Sketchy, Extended Sketchy, TU-Berlin, and QuickDraw. The results are organized into four main sections: (1) direct sketch-based retrieval using seven feature configurations (C1–C7), (2) parameterized feature-fusion analysis, (3) generation-based retrieval using stable diffusion with ControlNet, and (4) a comprehensive comparison with state-of-the-art methods. Each section includes quantitative analysis, and discussion of key observations. All experiments were conducted using the implementation details described in Chapter 4, with pretrained models (CLIP ViT-B/32, BLIP-base, stable diffusion v1.5 with ControlNet scribble) used without task-specific fine-tuning.

5.1 Direct Sketch-Based Retrieval Results

The direct retrieval paradigm was evaluated using seven feature configurations (C1–C7) defined in Section 4.8.2. These configurations systematically analyse the contribution of visual sketch features (CLIP image encoder), semantic caption features (BLIP-generated captions encoded via the CLIP text encoder), and class-level semantic features (CLIP-predicted class labels encoded via the CLIP text encoder), both individually and in combination. Evaluation employs standard SBIR metrics: mAP@All, mAP@200, mAP@100, Precision@100 (P@100), Precision@200 (P@200), and Accuracy@K for $K \in \{1, 5, 50, 100\}$.

5.1.1 Results on the Sketchy Dataset

Table 5.1 presents the direct retrieval results on the Sketchy dataset, which contains 75,471 sketches and 12,500 photographs across 125 object categories with instance-level sketch–photo pairings (100 photos per class).

Table 5.1: Direct retrieval results on the Sketchy dataset (C1–C7).

Config	mAP@All	mAP@200	mAP@100	P@100	P@200†	Acc@1	Acc@5	Acc@50	Acc@100
C1 (Sketch only)	0.544	0.650	0.697	0.519	0.320	0.776	0.821	0.905	0.936
C2 (Sketch + Caption)	0.603	0.680	0.720	0.567	0.345	0.777	0.822	0.917	0.946
C3 (Sketch + Class)	0.639	0.700	0.736	0.595	0.360	0.776	0.798	0.872	0.914
C4 (Caption only)	0.675	0.697	0.720	0.628	0.369	0.745	0.771	0.838	0.877
C5 (Class only)	0.715	0.730	0.744	0.663	0.388	0.762	0.763	0.811	0.855
C6 (Caption + Class Best $\alpha = 0.4$)	**0.719**	**0.740**	0.761	**0.671**	**0.401**	0.768	0.818	0.882	0.912
C7 (Full)	0.671	0.726	**0.764**	0.629	0.381	**0.808**	**0.860**	**0.922**	**0.949**

†P@200 on Sketchy is bounded because the class gallery contains 100 images per category.

Analysis. Several key observations emerge from the Sketchy results. First, the baseline configuration C1 (CLIP sketch features only) achieves moderate performance with mAP@All = 0.544, demonstrating that the pretrained CLIP provides reasonable zero-shot sketch-to-image matching capability despite the domain gap. However, this configuration exhibits the lowest precision at higher ranks (P@100 = 0.519), indicating that visual features alone are less stable in maintaining consistent retrieval quality across the full ranked list.

Second, incorporating semantic information consistently improves performance. Configuration C2 (sketch + BLIP caption) improves mAP@All by 10.8% over C1 (0.603 vs. 0.544), while C3 (sketch + CLIP class label) achieves a larger gain of 17.5% (0.639 vs. 0.544). This suggests that class-level semantic signals provide stronger categorical guidance than descriptive captions in this setting, likely because Sketchy exhibits well-defined category structure and relatively low intra-class ambiguity.

Third, semantic-only configurations (C4 and C5) surprisingly outperform the visual-only baseline. Configuration C5 (class label only) achieves mAP@All = 0.715, corresponding to a 31.4% improvement over C1, while C4 (caption only) reaches 0.675, a 24.1% gain. This indicates that high-level semantic representations in CLIP's text embedding space capture strong category-discriminative information and can partially compensate for the absence of sketch visual cues. However, this finding should be interpreted as dataset-dependent rather than a general replacement for visual input.

Fourth, the best overall ranking performance is achieved by C6 (caption + class label, $\beta = 0.0$, best $\alpha = 0.4$), with mAP@All = 0.719 and mAP@200 = 0.740. The reported C6 results reflect this optimized α value, which assigns 40% weight to caption features and 60% to class label features within the semantic fusion component. This balance suggests that on Sketchy, where sketch–photo pairings are instance-specific and category structure is clear, descriptive captions contribute meaningful complementary semantic detail beyond class labels alone. The combination of both semantic signals yields stronger global ranking consistency than any single modality. Interestingly, C6 slightly outperforms the full multimodal fusion C7 in ranking metrics, which may indicate that incorporating raw sketch features introduces small noise or redundancy when semantic cues are already highly informative. Nevertheless, C7 achieves the highest retrieval accuracy at top ranks

(Acc@1 = 0.808, Acc@5 = 0.860, Acc@100 = 0.949), confirming that visual sketch features still provide important complementary information for identifying relevant images among the first few retrieved results—a behavior that is particularly important in practical real-world retrieval systems where users typically focus on the top returned images.

5.1.2 Results on the Extended Sketchy Dataset

Table 5.2 presents results on the Extended Sketchy dataset, which augments the original Sketchy with approximately 60,502 additional photographs from ImageNet, totalling ∼73,002 photos across 125 categories.

Table 5.2: Direct retrieval results on the Extended Sketchy dataset (C1–C7).

Config	mAP@All	mAP@200	mAP@100	P@100	P@200	Acc@1	Acc@5	Acc@50	Acc@100
C1 (Sketch only)	0.401	0.626	0.642	0.617	0.579	0.627	0.764	0.866	0.902
C2 (Sketch + Caption)	0.481	0.705	0.724	0.697	0.655	0.721	0.799	0.893	0.923
C3 (Sketch + Class)	0.519	0.723	0.740	0.716	0.678	0.746	0.801	0.845	0.874
C4 (Caption only)	0.611	0.730	0.736	0.727	0.708	0.727	0.760	0.836	0.857
C5 (Class only)	0.665	0.756	0.761	0.756	0.741	0.764	0.765	0.809	0.820
C6 (Caption + Class, Best $\alpha = 0.2$)	**0.668**	0.761	0.767	**0.760**	**0.746**	0.768	0.793	0.826	0.840
C7 (Full)	0.568	**0.762**	**0.777**	0.757	0.721	**0.782**	**0.844**	**0.907**	**0.929**

Analysis. The Extended Sketchy results reveal several trends that differ from the basic Sketchy dataset. First, the performance gap between visual-only and semantic-only configurations widens significantly. While C1 achieves mAP@All = 0.401, C5 (class only) reaches 0.665—a 65.8% relative improvement. This amplification indicates that as the gallery grows (from 12.5K to 73K images), semantic guidance becomes increasingly critical for maintaining retrieval accuracy at the global ranking level.

Second, the precision metrics (P@100, P@200) are notably higher across all configurations compared to the basic Sketchy dataset. For instance, C1 achieves P@100 = 0.617 and P@200 = 0.579 on Extended Sketchy versus P@100 = 0.519 and P@200 = 0.320 on basic Sketchy. This reflects the gallery-size effect: Extended Sketchy contains multiple photos per category (including unpaired ImageNet images), increasing the probability that top-ranked results belong to the correct category even without fine-grained instance matching. Unlike the basic Sketchy where P@200 is bounded by the $\leq$100 images per class, the extended gallery contains hundreds of photos per class, making P@200 a more informative metric here.

Third, C6 ($\beta = 0.0$, best $\alpha = 0.2$) again achieves the best mAP@All = 0.668, while C7 leads in accuracy metrics (Acc@1 = 0.782, Acc@100 = 0.929). The reported C6 results reflect this optimized α value, which assigns only 20% weight to caption features and 80% to class label features. The reduced caption contribution relative to Sketchy ($\alpha = 0.4$) is consistent with the larger and more diverse gallery: with nearly six times as many images, global ranking accuracy benefits more from sharp categorical class-level signals than from

fine-grained descriptive captions. This pattern—C6 for best global ranking, C7 for best top-K accuracy—is consistent across both Sketchy variants, validating the robustness of the proposed fusion strategy: pure semantic fusion optimises ranking quality, while full multimodal fusion maximises top-K success rates.

5.1.3 Results on the TU-Berlin Dataset

Table 5.3 presents results on the TU-Berlin dataset, comprising approximately 20,000 sketches across 250 categories with 204,070 natural images in the gallery.

Table 5.3: Direct retrieval results on the TU-Berlin dataset (C1–C7).

Config	mAP@All	mAP@200	mAP@100	P@100	P@200	Acc@1	Acc@5	Acc@50	Acc@100
C1 (Sketch only)	0.465	0.729	0.755	0.711	0.642	0.755	0.844	0.916	0.944
C2 (Sketch + Caption)	0.511	0.753	0.777	0.737	0.674	0.793	0.856	0.921	0.948
C3 (Sketch + Class)	0.537	0.746	0.762	0.737	0.685	0.763	0.803	0.878	0.912
C4 (Caption only)	0.562	0.711	0.728	0.702	0.660	0.706	0.784	0.838	0.854
C5 (Class only)	0.593	0.711	0.722	0.705	0.667	0.751	0.751	0.784	0.835
C6 (Caption + Class, Best $\alpha = 0.3$)	**0.601**	0.726	0.736	0.720	0.682	0.753	0.763	0.844	0.868
C7 (Full)	0.569	**0.780**	**0.799**	**0.770**	**0.715**	**0.794**	**0.871**	**0.926**	**0.946**

Analysis. The TU-Berlin results reveal a distinct behavior compared to both Sketchy variants, highlighting the impact of large-scale gallery size and higher category diversity. First, the baseline configuration C1 (sketch only) achieves mAP@All = 0.465, which is intermediate between Sketchy (0.544) and Extended Sketchy (0.401). The difficulty here stems from high category diversity (250 classes) and a large gallery (204K images) rather than the sketch–photo domain gap alone. Nevertheless, relatively strong precision values (P@100 = 0.711, P@200 = 0.642) confirm that sketch features still preserve meaningful local ranking structure.

Second, incorporating semantic information consistently improves performance. Configuration C2 (sketch + caption) improves mAP@All by 9.9% over C1 (0.511 vs. 0.465), while C3 (sketch + class label) achieves a larger gain of 15.5% (0.537 vs. 0.465). This confirms that class-level semantics provide stronger global ranking guidance than descriptive captions, particularly across a large and diverse 250-category space.

Third, semantic-only configurations (C4 and C5) outperform the sketch-only baseline in mAP@All, with C5 (class only) achieving 0.593—a 27.5% improvement over C1. This indicates that TU-Berlin categories are highly separable in the semantic embedding space, allowing CLIP text representations to capture strong discriminative structure even without visual input. However, unlike Extended Sketchy, semantic-only configurations do not consistently dominate all precision metrics, suggesting that visual features remain important for refining ranking stability.

Fourth, C6 ($\beta = 0.0$, best $\alpha = 0.3$) achieves the best mAP@All among all configurations (0.601). The reported C6 results reflect this optimized α value, which assigns a

balanced 30% weight to caption features and 70% to class label features. This intermediate balance—between Sketchy's more caption-weighted setting ($\alpha = 0.4$) and Extended Sketchy's more class-weighted setting ($\alpha = 0.2$)—suggests that TU-Berlin benefits more from class label information because it contains many categories. However, since sketches are drawn in different styles by multiple participants, caption features still provide useful additional semantic details.

Nevertheless, the full multimodal model C7 achieves the best overall retrieval performance in most ranking and precision metrics, including mAP@200 = 0.780, mAP@100 = 0.799, P@100 = 0.770, and P@200 = 0.715. In particular, C7 achieves the highest Acc@1 (0.794) and Acc@5 (0.871), demonstrating that the inclusion of visual sketch features significantly enhances early retrieval accuracy. Overall, TU-Berlin exhibits a stronger dependency on multimodal fusion compared to either Sketchy variant, where semantic information alone dominates ranking performance. In contrast, TU-Berlin benefits more consistently from combining visual and semantic cues, particularly for improving robustness in large-scale retrieval scenarios.

5.1.4 Results on the QuickDraw Dataset

Table 5.4 presents results on the QuickDraw dataset, using a subset of 110 categories with 330,000 sketches and 203,584 photos. QuickDraw sketches are highly abstract, minimal, and drawn under time constraints by non-experts.

Table 5.4: Direct retrieval results on the QuickDraw dataset (C1–C7).

Config	mAP@All	mAP@200	mAP@100	P@100	P@200	Acc@1	Acc@5	Acc@50	Acc@100
C1 (Sketch only)	0.175	0.365	0.384	0.337	0.323	0.338	0.558	0.739	0.818
C2 (Sketch + Caption)	0.212	0.399	0.414	0.379	0.366	0.365	0.559	0.726	0.795
C3 (Sketch + Class)	0.267	0.495	0.509	0.477	0.460	0.478	0.636	0.736	0.786
C4 (Caption only)	0.331	0.455	0.458	0.449	0.445	0.452	0.507	0.609	0.672
C5 (Class only)	**0.437**	**0.573**	**0.577**	**0.567**	**0.568**	**0.539**	0.580	0.665	0.686
C6 (Caption + Class, Best $\alpha = 0.0$)	**0.437**	**0.573**	**0.577**	**0.567**	**0.568**	**0.539**	0.580	0.665	0.686
C7 (Full)	0.282	0.493	0.509	0.474	0.464	0.459	**0.649**	**0.800**	**0.856**

Analysis. The QuickDraw results represent the most challenging evaluation setting among all datasets due to the extreme abstraction level of sketches and the large-scale retrieval gallery. First, the baseline configuration C1 (sketch only) achieves a very low mAP@All = 0.175, significantly lower than all other datasets. This confirms that time-constrained, non-expert sketches contain minimal structural detail, making them highly unsuitable for direct visual matching using CLIP sketch embeddings alone.

Second, incorporating semantic information leads to substantial improvements. Configuration C2 (sketch + caption) improves mAP@All to 0.212, while C3 (sketch + class label) further increases it to 0.267. This demonstrates that even in highly abstract sketch

settings, explicit category-level supervision provides more reliable guidance than descriptive captions. Nevertheless, the overall gains from sketch-conditioned configurations remain limited, indicating that the visual sketch signal itself contributes relatively weak discriminative information in this dataset.

Third, semantic-only configurations (C4 and C5) significantly outperform all sketch-dependent models in global ranking quality. Configuration C5 (class only) achieves the best overall mAP@All = 0.437, representing a 149% improvement over C1. This dramatic increase indicates that QuickDraw sketches are primarily category-representative rather than instance-representative, making them well-aligned with CLIP's text embedding space.

Fourth, and importantly, C6 ($\beta = 0.0$, best $\alpha = 0.0$) is identical to C5, since the optimized α is found to be 0.0, indicating that within the semantic fusion equation, zero weight is assigned to caption features while full weight is assigned to class label features, effectively reducing C6 to a pure class-only configuration. This result is not due to chance but represents a meaningful finding: for highly abstract QuickDraw sketches, BLIP-generated captions are typically noisy, generic, or semantically incoherent, and therefore provide no useful discriminative information. Under these conditions, class labels derived from CLIP's zero-shot predictions constitute the only reliable semantic signal, and the caption-based component is appropriately suppressed during parameter search.

Fifth, the full multimodal configuration C7 does not improve mAP@All beyond semantic-only models (0.282 vs. 0.437) but achieves the best performance in some higher-rank metrics, particularly Acc@100 = 0.856 and Acc@5 = 0.649. This indicates that while visual information is insufficient for global ranking improvements in QuickDraw, it still provides marginal complementary signals for refining early retrieval precision within the top-ranked results.

Overall, QuickDraw highlights a settings shift in SBIR performance: as sketch abstraction increases, semantic information becomes the dominant factor for retrieval quality, while visual features transition from primary signals to weak secondary refinement cues, and caption-based features become unreliable and are optimally discarded.

5.2 Parameterized Feature Fusion Analysis

To optimise the contribution of each modality, a systematic parameter sweep was conducted over the fusion weights α (caption–class balance) and β (visual–semantic balance) with a step size of 0.1. Table 5.5 reports the optimal parameter combinations for each dataset.

Table 5.5: Best parameterized fusion results (optimal α and β per dataset).

Dataset	α	β	mAP@All	mAP@200	mAP@100	P@100	P@200	Acc@1	Acc@5	Acc@50	Acc@100
Sketchy	0.4	0.0	0.719	0.740	0.761	0.671	0.401	0.768	0.818	0.882	0.912
Ext. Sketchy	0.2	0.0	0.668	0.761	0.767	0.760	0.746	0.768	0.793	0.826	0.840
TU-Berlin	0.3	0.1	0.605	0.747	0.756	0.741	0.701	0.766	0.780	0.865	0.892
QuickDraw	0.0	0.0	0.437	0.573	0.577	0.567	0.568	0.539	0.580	0.665	0.686

Analysis. The parameterized fusion analysis yields several important insights. Recall that the fusion equation is $v_{\text{combined}} = \beta \cdot v_{\text{sketch}} + (1 - \beta)(\alpha \cdot v_{\text{caption}} + (1 - \alpha) \cdot v_{\text{class}})$, where β controls the balance between visual sketch features and the semantic component, and α controls the balance between caption and class label features within the semantic component.

First, the optimal β value is 0.0 for three of the four datasets (Sketchy, Extended Sketchy, QuickDraw), meaning the best global ranking performance is achieved when visual sketch features are entirely excluded and only the semantic component is used. This confirms that for category-level SBIR, semantic information alone can outperform or match visual features when using frozen pretrained models.

Second, TU-Berlin is the only dataset where a non-zero β is optimal ($\beta = 0.1$), indicating that a small contribution from visual features improves performance on high-diversity, unpaired datasets. This aligns with the observation in Section 5.1.3 that C7 outperforms C6 on most metrics: visual structural cues provide valuable complementary information when the category space is large (250 classes) and sketches are not paired with specific photos.

Third, the optimal α values vary systematically across datasets and carry clear interpretive meaning. On Sketchy ($\alpha = 0.4$), a meaningful caption contribution is warranted: instance-level sketch–photo pairings and a structured 125-class space mean that descriptive captions provide genuinely useful fine-grained semantic detail beyond class labels. On Extended Sketchy ($\alpha = 0.2$), a larger and more diverse gallery shifts the optimal balance toward class labels, which provide sharper categorical anchors for global ranking across a 73K-image pool. On TU-Berlin ($\alpha = 0.3$), the intermediate balance reflects two competing pressures: the large category count (250) favours class label precision, while the diversity of sketching styles across participants introduces enough intra-class variation that captions still contribute. Finally, on QuickDraw ($\alpha = 0.0$), captions are completely discarded because BLIP-generated descriptions of highly abstract, time-constrained sketches are typically noisy or generic, contributing no discriminative signal—and the parameter sweep correctly identifies this by assigning zero weight to captions.

5.2.1 Retrieval Visualisation Across Configurations

Figures 5.2–5.5 illustrate the qualitative effect of each configuration (C1–C7) and the best parametric setting on a parrot sketch query, evaluated across all four benchmark datasets. The query sketch is shown in Figure 5.1. Red boxes highlight incorrect retrievals. Table 5.6 summarises the BLIP caption and CLIP-predicted class for each dataset.

Figure 5.1: The parrot sketch used as the query across all retrieval visualisation experiments.

Table 5.6: BLIP caption and CLIP-predicted class for the parrot sketch query across datasets.

Dataset	**BLIP Caption**	**Predicted Class**
Sketchy	*a parrot*	*parrot*
Extended Sketchy	*a parrot*	*parrot*
TU-Berlin	*a parrot*	*pigeon*
QuickDraw	*a parrot*	*parrot*

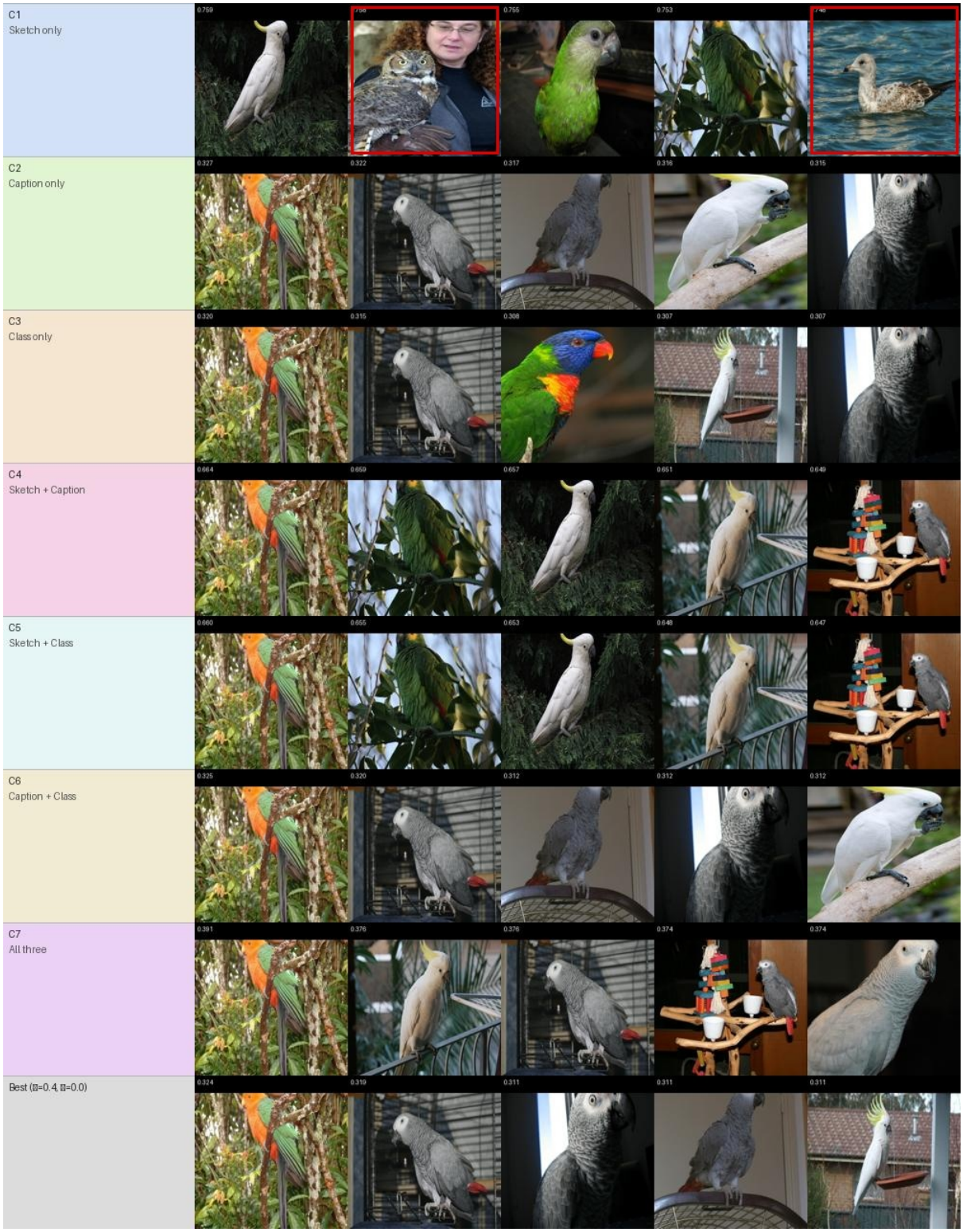

Figure 5.2: Retrieval visualisation across configurations — Sketchy dataset [Caption: *"a parrot"*, Predicted class: *parrot*]. Each row shows top-5 results for one configuration (C1–C7 and Best). Red boxes mark incorrect retrievals. C1 (visual only) retrieves 3 correct results but returns 2 wrong ones (positions 2 and 5) — visually similar birds that do not match the parrot class. All semantic and fusion configurations (C2–C7 and Best) retrieve all 5 correctly.

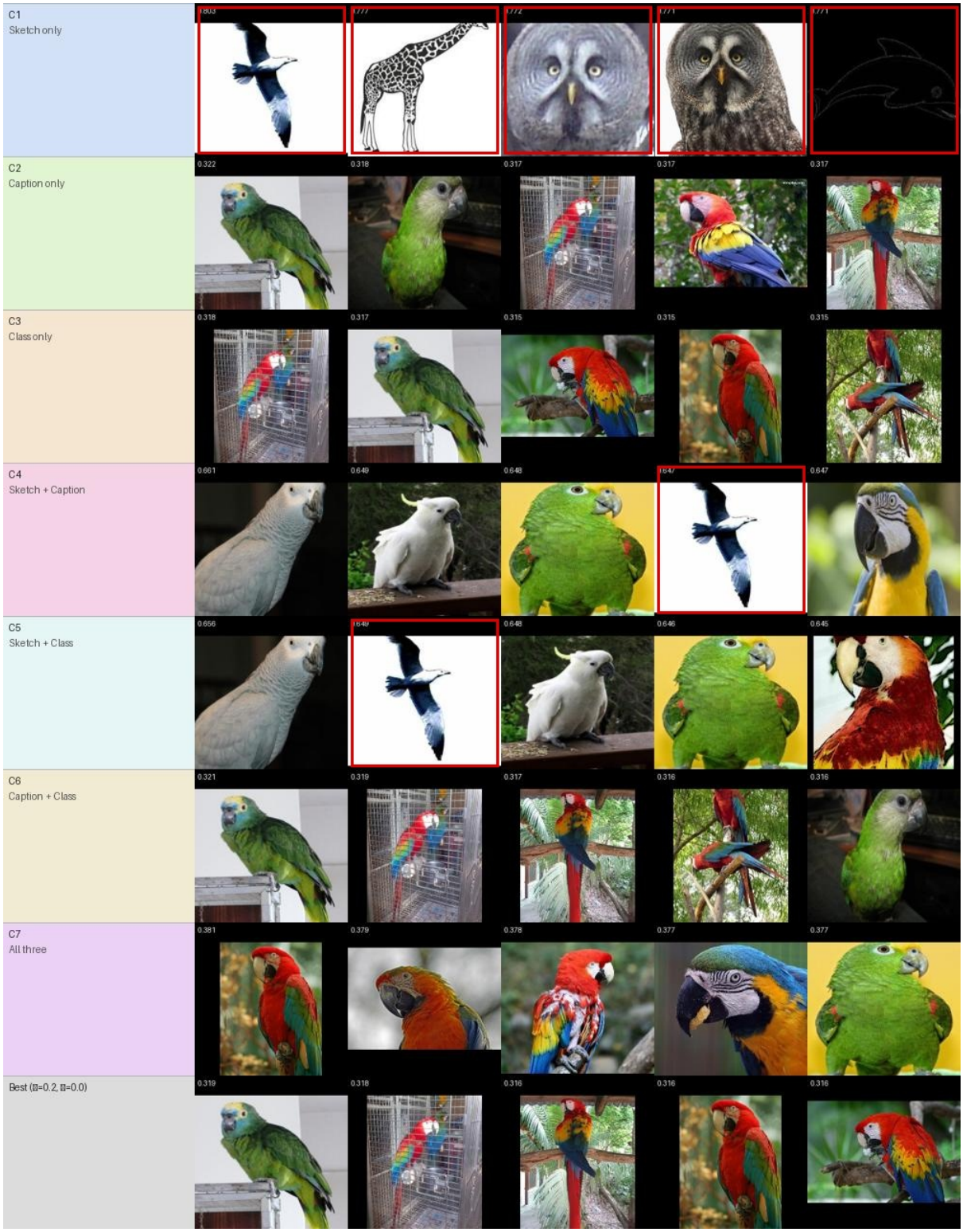

Figure 5.3: Retrieval visualisation across configurations — Extended Sketchy dataset [Caption: *"a parrot"*, Predicted class: *parrot*]. C1 fails completely, retrieving a magpie, a giraffe, and an owl instead of parrots. C4 and C5 each introduce one failure due to residual visual-feature noise. C2, C3, C6, C7, and Best all retrieve correctly.

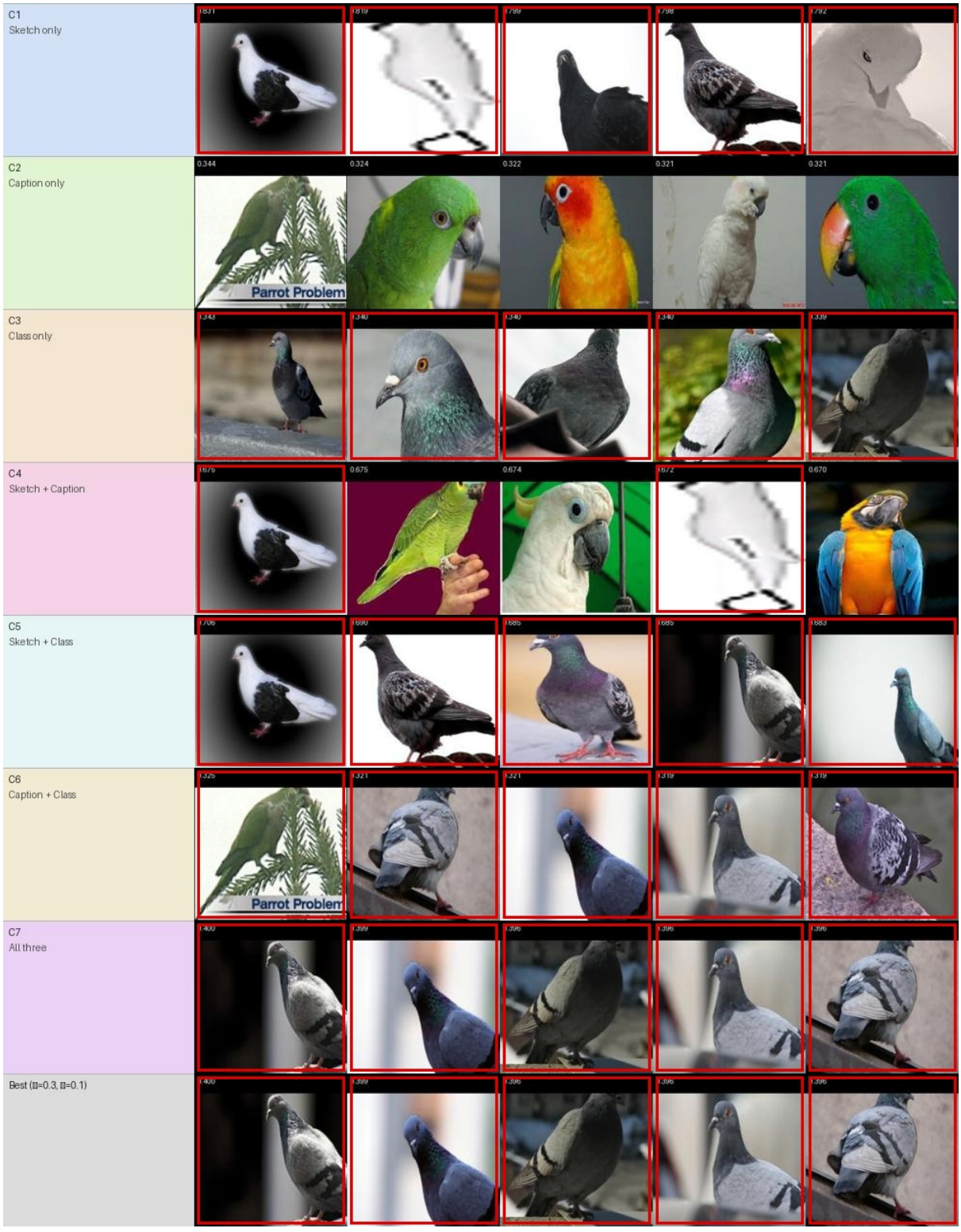

Figure 5.4: Retrieval visualisation across configurations — TU-Berlin dataset [Caption: *"a parrot"*, Predicted class: *pigeon*]. CLIP misclassified the sketch as *pigeon*. C2 (caption only) is the sole configuration that retrieves all 5 correctly. C3 (class only), C5 (sketch + class), C6 (caption + class), C7, and Best all fail completely because the wrong class label dominates the query. C4 (sketch + caption, no class) still produces 2 errors: without any class signal, the sketch visual feature alone introduces enough noise to pull two results toward pigeon-like images despite the correct caption. C1 (visual only) also fails completely.

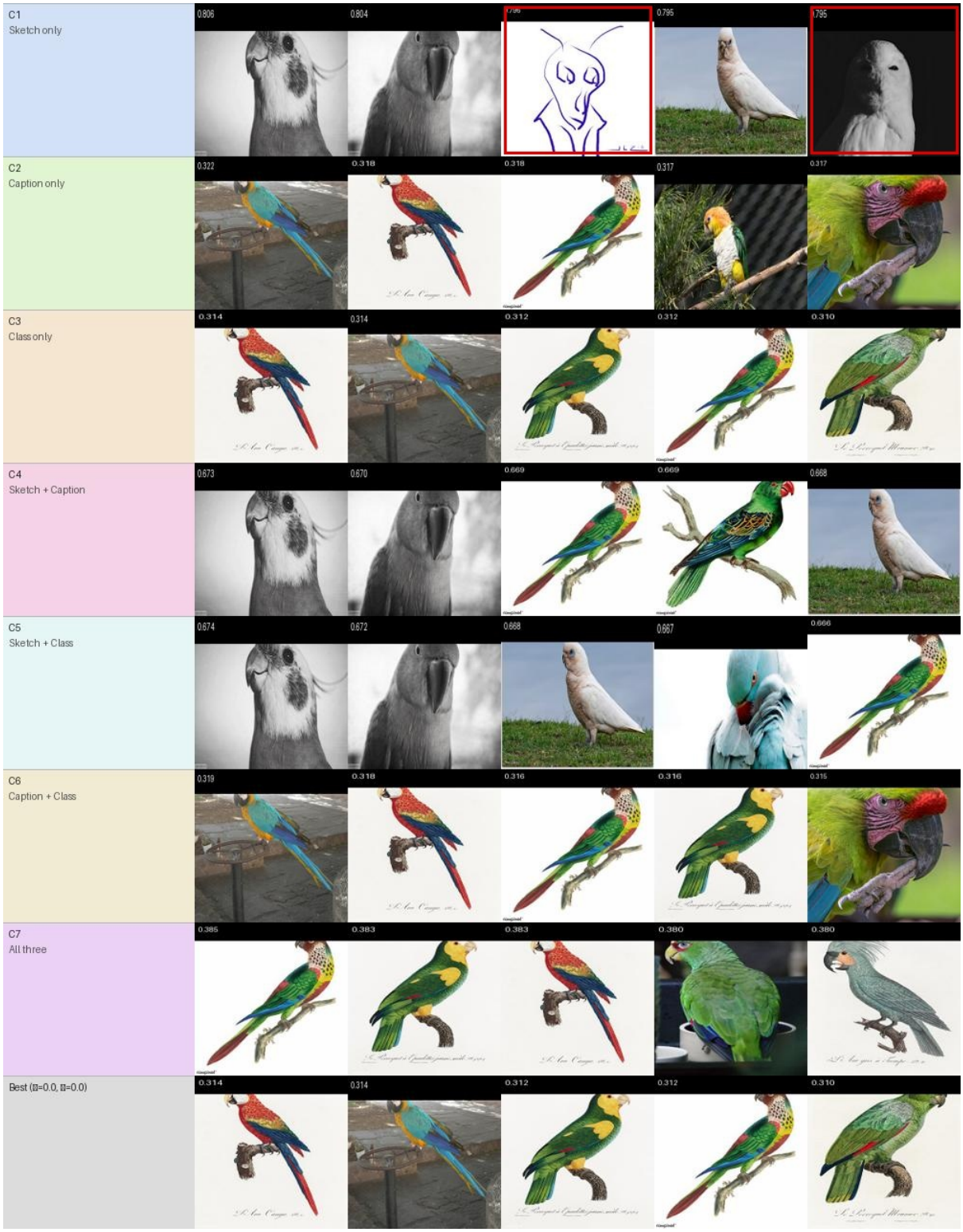

Figure 5.5: Retrieval visualisation across configurations — QuickDraw dataset [Caption: *"a parrot"*, Predicted class: *parrot*]. C1 retrieves 3 correct results but returns 2 wrong ones (positions 3 and 5) despite visual shape similarity. All other configurations (C2–C7 and Best) retrieve all 5 correctly. Best (α=0.0, β=0.0) is equivalent to C3, confirming that class labels alone are the optimal signal for the highly abstract QuickDraw sketch style.

The visualisations confirm and extend the quantitative findings across all four datasets. Three patterns stand out.

First, C1 (visual only) is consistently the weakest configuration, yet its errors are not arbitrary — the retrieved images tend to share visual characteristics with the parrot sketch, such as a similar body outline, wing shape, or beak shape. On Sketchy and QuickDraw, C1 retrieves 3 out of 5 correct results; the 2 wrong retrievals are images that appear visually similar to the parrot sketch in shape but belong to different categories. On Extended Sketchy and TU-Berlin, C1 fails to retrieve any correct results, but the returned images still reflect shape-level similarity to the query sketch — they are not random. This highlights the fundamental limitation of purely visual retrieval: it is sensitive to visual similarity rather than semantic category, so classes with overlapping shapes become hard to separate without an additional semantic signal. As shown in the configurations that follow, introducing the CLIP-predicted class label as a semantic signal is largely sufficient to resolve these shape-confusion errors — the class embedding is strong enough to dominate the query vector and steer retrieval toward the correct category, provided the class prediction itself is correct.

Second, the TU-Berlin case reveals the most critical failure mode: CLIP predicted *pigeon* instead of *parrot* for this sketch. Every configuration that incorporates the class label — C3 (class only), C5 (sketch + class), C6 (caption + class), C7, and the best parametric setting — fails completely, as the wrong class embedding dominates the query. Notably, C6 fails even though C2 (caption only) succeeds, because the equal-weight combination of a correct caption and a wrong class label still steers the query toward pigeons. C4 (sketch + caption, β=0.5, α=1.0) uses no class signal at all, yet still produces 2 errors: the sketch visual feature alone introduces enough noise to pull two results toward pigeon-like images even when the correct caption "a parrot" is present. Only C2, which relies solely on the BLIP caption, retrieves all 5 correctly. This highlights the value of the caption signal as an independent and reliable cue: when the class prediction is wrong, the caption alone is sufficient to guide retrieval to the correct category, while any configuration that incorporates the incorrect class label — or the sketch visual feature — is pulled away from the right results.

Third, on Sketchy, Extended Sketchy, and QuickDraw — where CLIP correctly predicts *parrot* — all pure text-signal configurations (C2, C3) and most fusion configurations retrieve correctly. On Sketchy and QuickDraw, every configuration except C1 succeeds fully. Extended Sketchy is slightly more challenging: C4 and C5 each introduce one residual error, since both incorporate the sketch visual feature (β=0.5), which adds a small amount of noise to an otherwise correct semantic signal. Configurations that do not use the sketch feature — C2, C3, C6, C7, and Best — all retrieve perfectly. Overall, the results show that the approach is robust when at least one semantic signal is correct, and that the sketch visual feature, while useful for shape-level discrimination in some settings,

can occasionally harm retrieval when the target class has visually similar neighbours.

5.3 Generation-Based Retrieval Results

The generation-based retrieval paradigm transforms input sketches into realistic images using stable diffusion with ControlNet, guided by either BLIP-generated captions or CLIP-predicted class labels as text prompts. Table 5.7 presents the results for both prompt strategies across all four datasets.

Table 5.7: Generation-based retrieval results (BLIP caption vs. CLIP class label prompt strategies). Bold values indicate the better-performing prompt strategy within each dataset.

Dataset	Prompt	mAP@All	mAP@200	mAP@100	P@100	P@200†	Acc@1	Acc@5	Acc@50	Acc@100
Sketchy	BLIP	0.487	0.584	0.621	0.472	0.291	0.683	**0.755**	**0.833**	**0.878**
	CLIP	**0.501**	**0.591**	**0.626**	**0.479**	0.291	**0.686**	0.732	0.785	0.827
Ext. Sketchy	BLIP	0.379	0.608	0.633	0.587	0.539	0.666	**0.753**	**0.825**	**0.857**
	CLIP	**0.409**	**0.642**	**0.665**	**0.624**	**0.573**	**0.693**	0.734	0.768	0.790
TU-Berlin	BLIP	0.422	0.660	0.681	0.642	0.582	0.700	**0.765**	**0.825**	0.859
	CLIP	**0.441**	**0.668**	**0.689**	**0.648**	**0.589**	**0.718**	0.760	0.817	**0.862**
QuickDraw	BLIP	0.048	0.120	0.129	0.099	0.089	0.124	0.233	0.364	0.535
	CLIP	**0.292**	**0.542**	**0.549**	**0.536**	**0.518**	**0.548**	**0.605**	**0.647**	**0.670**

†P@200 on Sketchy is bounded because the class gallery contains ≤100 images per category.

Analysis. Across all four datasets, CLIP class label prompts consistently outperform BLIP captions on global ranking metrics (mAP@All, mAP@200, mAP@100, P@100, P@200, and Acc@1). On **Sketchy**, **Extended Sketchy**, and **TU-Berlin**, BLIP captions achieve better Acc@K at larger K values (Acc@5, Acc@50, Acc@100), suggesting that descriptive captions generate visually richer images that help surface relevant results within broader windows, while class labels produce more category-focused images that rank better at the top.

On **Sketchy**, the margins are moderate: CLIP achieves mAP@All = 0.501 versus 0.487 for BLIP, while BLIP leads on Acc@50 (0.833 vs. 0.785) and Acc@100 (0.878 vs. 0.827). Note that P@200 is tied at 0.291 for both strategies, as the gallery contains at most 100 images per class, capping achievable precision at this cutoff. On **Extended Sketchy** the same pattern holds, with the BLIP advantage at larger K amplified by the larger gallery size (Acc@100: 0.857 vs. 0.790). On **TU-Berlin** the two strategies are most competitive, with smaller margins across all metrics, reflecting the dataset's greater category diversity (250 classes) which allows captions to provide some useful generation guidance.

QuickDraw is the exception: CLIP dominates across all metrics with no reversal at larger K, achieving mAP@All = 0.292 versus a near-random 0.048 for BLIP. BLIP fails because highly abstract QuickDraw sketches produce incoherent captions such as "a

drawing of lines," which give stable diffusion no useful categorical signal. CLIP's zero-shot class prediction remains reliable even for minimal sketch inputs, making class-level prompts the only viable strategy on this dataset.

Generation vs. direct retrieval. Comparing the generation-based paradigm against the best direct retrieval configurations, the generative approach consistently underperforms across all datasets regardless of prompt strategy. On Sketchy, the best generative result (mAP@All = 0.501 with CLIP prompt) falls 30.3% short of the best direct result (C6, 0.719). On Extended Sketchy the gap is 38.8% (0.409 vs. 0.668); on TU-Berlin 26.6% (0.441 vs. 0.601); and on QuickDraw 33.2% (0.292 vs. 0.437). The underlying causes of this consistent underperformance are discussed in Section 5.5.3.

Qualitative Comparison of Prompt Strategies. Figure 5.6 illustrates the qualitative behaviour of both prompt strategies on a parrot sketch query across all four datasets. For the BLIP caption prompt, BLIP correctly generates the caption *"a parrot"* regardless of dataset, so stable diffusion produces a realistic parrot image in every case and all top-5 retrieved results are correct. For the CLIP class prompt, retrieval succeeds on Sketchy, Extended Sketchy, and QuickDraw, where the predicted class is correctly identified as *parrot*. On TU-Berlin, however, the class prompt fails completely: CLIP predicts *pigeon*, stable diffusion generates a pigeon accordingly, and all five retrieved images are pigeons. This illustrates the cascade failure of the class-prompt strategy — a wrong class prediction leads to a wrong generated image, which produces an entirely wrong set of retrieved results with no possibility of recovery at any stage. The caption prompt is unaffected because it depends on a description of what is visible in the sketch rather than a class prediction, making it more robust when CLIP's class prediction is unreliable.

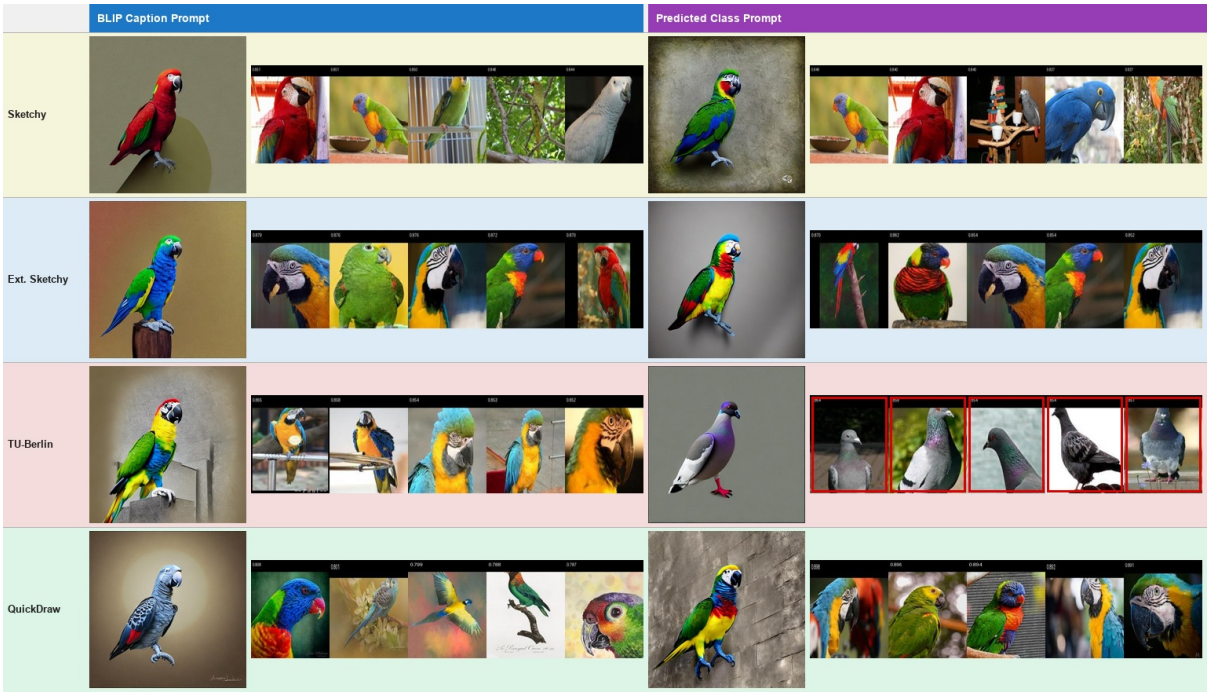

Figure 5.6: Generation-based retrieval results for a parrot sketch query across all four datasets, comparing BLIP caption prompt (left, blue) and CLIP predicted class prompt (right, purple). Each cell shows the generated image followed by the top-5 retrieved images. Red boxes mark incorrect retrievals. The class prompt fails on TU-Berlin because the predicted class is *pigeon*; the caption prompt succeeds on all four datasets.

5.4 Comparison with State-of-the-Art (SOTA) Methods

This section compares the proposed approach against three published zero-shot SBIR papers, comprising four model variants. All methods use task-specific training, while our approach uses no training at all. Since Koley et al [32]. also use stable diffusion as their underlying pretrained model, our generation-based result (Ours – Gen-CLIP) is included alongside the parameterized fusion best to enable a direct like-for-like comparison.

- **ZSE-RN / ZSE-Ret** — Lin et al. [22] (CVPR 2023): transformer-based zero-shot SBIR with two variants — a ResNet backbone (ZSE-RN) and a retrieval network (ZSE-Ret). Requires task-specific training on paired sketch–image datasets. At query time, a manual class label must be provided to condition the cross-modal attention mechanism. Evaluated on Extended Sketchy (Split 1 and Split 2), TU-Berlin, and QuickDraw.

- **CLIP-for-All / CLIP-AT** — Sain et al. [34] (CVPR 2023): extends CLIP with learned prompt adaptation for zero-shot SBIR. Requires training to learn prompt vectors $[\mathbf{v}_1, \ldots, \mathbf{v}_k]$ on sketch–image pairs; only the prompt vectors are updated

while the CLIP backbone remains frozen. At query time, a manual class label (e.g., "cat") must be supplied — the text branch is non-functional without it. Evaluated on Extended Sketchy (Split 2), TU-Berlin, and QuickDraw.

- **Diff-SBIR** — Koley et al. [32] (CVPR 2024): uses a stable diffusion UNet backbone with trained visual prompts to extract internal diffusion features at timesteps 200–300 for sketch–photo matching; no image generation is performed. Requires triplet-loss training with paired sketch–photo data to learn both the projection layer and the text prompt vectors. Although the prompts are learned rather than hand-crafted, a class label is still required at query time to initialise and condition the UNet text prompt. Evaluated on Extended Sketchy (Split 2), TU-Berlin, and QuickDraw.

Our results come from the best parameterized fusion configuration (Table 5.5), where α and β are chosen per dataset by grid search with step size 0.1. Each comparison table uses only the metrics reported by the SOTA papers for that dataset. The basic Sketchy dataset is not included here because SOTA methods evaluate it under instance-level fine-grained SBIR, which is a different task from the category-level retrieval used in this thesis.

5.4.1 Comparison on Extended Sketchy

Extended Sketchy contains 73,002 images across 125 categories. Two evaluation splits exist in the literature. **Split 1** (Sketchy-1-Ext [22]) holds out 25 categories for testing and uses mAP@All and P@100. **Split 2** (Sketchy-2-Ext [32]) holds out 21 categories that do not appear in ImageNet and uses mAP@200 and P@200. Our parameterized best (α=0.2, β=0.0) achieves mAP@All = 0.668, P@100 = 0.760, mAP@200 = 0.761, and P@200 = 0.746.

Table 5.8: Comparison on Extended Sketchy (ZS-SBIR). Split 1 uses mAP@All and P@100; Split 2 uses mAP@200 and P@200. Dashes mean the metric was not reported. "Ours" uses α=0.2, β=0.0.

			Split 1 (Sketchy-1-Ext)		**Split 2 (Sketchy-2-Ext)**	
Method	**Venue**	**Training**	**mAP@All**	**P@100**	**mAP@200**	**P@200**
ZSE-RN [22]	CVPR 2023	✓	0.698	0.797	0.525	0.624
ZSE-Ret [22]	CVPR 2023	✓	**0.736**	**0.808**	0.504	0.602
CLIP-for-All [34]	CVPR 2023	✓	—	—	0.723	0.725
Koley et al. [32]	CVPR 2024	✓	—	—	**0.746**	**0.747**
Ours (α=0.2, β=0.0)	This thesis	×	0.668	0.760	**0.761**	0.746
Ours – Gen-CLIP	This thesis	×	0.409	0.624	0.642	0.573

S1 = Sketchy-1-Ext (25 test categories); S2 = Sketchy-2-Ext (21 ImageNet-absent test categories).

Analysis. On **Split 1** (mAP@All, P@100), our method is below the two ZSE variants: mAP@All = 0.668 vs. 0.698 (ZSE-RN) and 0.736 (ZSE-Ret), and P@100 = 0.760 vs. 0.797 and 0.808 respectively. This is a reasonable gap given that ZSE variants are explicitly trained to optimise full-gallery ranking, while our method uses no training at all.

On **Split 2** (mAP@200, P@200), the results are much stronger. Our mAP@200 = 0.761 is **the highest among all methods**, surpassing Koley et al. (0.746, +2.0%), CLIP-for-All (0.723, +5.3%), ZSE-RN (0.525, +44.9%), and ZSE-Ret (0.504, +51.0%). Our P@200 = 0.746 matches Koley et al. (0.747, −0.1%) almost exactly and surpasses CLIP-for-All (0.725, +2.9%) and both ZSE variants. This shows that our method ranks the most relevant images very highly, even though it does not train to sort the full gallery. Notably, our generation-based result (Ours – Gen-CLIP) also uses stable diffusion — the same pretrained model as Koley et al. — but without any training. On Split 2, Gen-CLIP achieves mAP@200 = 0.642 and P@200 = 0.573, surpassing both ZSE variants (ZSE-RN: 0.525; ZSE-Ret: 0.504) but falling below CLIP-for-All (0.723) and Koley et al. (0.746). On Split 1, Gen-CLIP achieves mAP@All = 0.409 and P@100 = 0.624, below the trained ZSE variants (0.698–0.736). These results demonstrate that the pretrained stable diffusion backbone alone carries substantial retrieval potential, and that training provides incremental rather than fundamental gains on this split.

5.4.2 Comparison on TU-Berlin

All SOTA papers report mAP@All and P@100 for TU-Berlin Extended. Our parameterized best uses α=0.3, β=0.1.

Table 5.9: Comparison on TU-Berlin-Extended (ZS-SBIR): mAP@All and P@100. "Ours (Gen)" = generation-based result using CLIP class label prompt.

Method	**Venue**	**Training**	**mAP@All**	**P@100**
ZSE-RN [22]	CVPR 2023	✓	0.542	0.657
ZSE-Ret [22]	CVPR 2023	✓	0.569	0.637
CLIP-for-All [34]	CVPR 2023	✓	0.651	0.732
Koley et al. [32]	CVPR 2024	✓	**0.680**	**0.744**
Ours (α=0.3, β=0.1)	This thesis	×	0.605	0.741
Ours – Gen-CLIP	This thesis	×	0.441	0.648

Analysis. Our parameterized best achieves mAP@All = 0.605 and P@100 = 0.741.

On **mAP@All**, our result (0.605) surpasses both ZSE variants (ZSE-RN: 0.542, +11.6%; ZSE-Ret: 0.569, +6.3%). CLIP-for-All (0.651) and Koley et al. (0.680) achieve higher mAP@All, which is expected as both are trained to optimise full-gallery ranking

directly.

On **P@100**, our result (0.741) surpasses ZSE-RN (0.657, +12.8%), ZSE-Ret (0.637, +16.3%), and CLIP-for-All (0.732, +1.2%), and is only 0.4% below Koley et al. (0.744). Matching CLIP-for-All and nearly matching Koley et al. on P@100 without any training is a strong result.

Notably, our generation-based result (Ours – Gen-CLIP, mAP@All = 0.441, P@100 = 0.648) uses the same pretrained stable diffusion model as Koley et al. but with no training. That our parameterized fusion (mAP@All = 0.605, P@100 = 0.741) largely closes the gap to Koley et al. (0.680, 0.744) without any generation or training demonstrates the strength of semantic feature fusion as a training-free alternative.

5.4.3 Comparison on QuickDraw

All SOTA papers report mAP@All and P@200 for QuickDraw Extended. Our parameterized best uses α=0.0, β=0.0, which is the same as using only class-label semantics (C5 and C6 give identical results).

Table 5.10: Comparison on QuickDraw-Extended (ZS-SBIR): mAP@All and P@200. "Ours (Gen)" = generation-based result using CLIP class label prompt.

Method	**Venue**	**Training**	**mAP@All**	**P@200**
ZSE-RN [22]	CVPR 2023	✓	0.145	0.216
ZSE-Ret [22]	CVPR 2023	✓	0.142	0.202
CLIP-for-All [34]	CVPR 2023	✓	0.202	0.388
Koley et al. [32]	CVPR 2024	✓	0.231	0.397
Ours (α=0.0, β=0.0)	This thesis	×	**0.437**	**0.568**
Ours – Gen-CLIP	This thesis	×	0.292	0.518

Analysis. Our parameterized best achieves mAP@All = 0.437 and P@200 = 0.568, which is **higher than all four trained methods on both metrics**. The best trained method is Koley et al. (mAP@All = 0.231, P@200 = 0.397). Our improvement over Koley et al. is +89.2% on mAP@All and +43.1% on P@200. This is the only dataset where our training-free approach clearly beats all trained methods.

QuickDraw's highly abstract sketches disadvantage trained methods, which rely on visual patterns that do not transfer well to minimal inputs. CLIP's zero-shot class prediction remains reliable regardless of sketch abstraction, and the parameter sweep converging to α=0.0, β=0.0 confirms that class-level text embeddings are the only effective signal on this dataset.

Notably, our generation-based result (Ours – Gen-CLIP, mAP@All = 0.292, P@200 = 0.518) also uses stable diffusion — the same pretrained model as Koley et al. (0.231,

0.397) — but without training, and still outperforms Koley et al. on both metrics (+26.4% on mAP@All, +30.5% on P@200). This confirms that even the generation-based approach alone surpasses the trained stable diffusion baseline on this dataset.

A deeper explanation for this result lies in how trained methods and our approach handle sketch abstraction and class semantics differently. QuickDraw sketches are drawn in seconds by non-experts and carry almost no visual detail — they are closer to symbolic labels than visual representations. At this level of abstraction, even well-trained visual alignment produces weak query embeddings, because the sketch visual signal is too sparse to bridge the domain gap reliably.

Trained methods such as CLIP-AT [34] and Koley et al. [32] also encode class names as text embeddings, so the closed 110-class vocabulary is available to them as well. However, their architectures blend the class label text embedding with the sketch visual embedding — the two signals are mixed into a single query vector. On QuickDraw, the noisy sketch visual embedding contaminates the otherwise clean class label signal, weakening the final query regardless of how well their prompts are learned.

Our method, by contrast, operates at α=0.0, β=0.0, meaning the sketch visual embedding and BLIP caption are entirely absent from the query. The class label text embedding is the *complete* query — pure and uncontaminated by any visual sketch signal. Furthermore, with only 110 common everyday classes, CLIP zero-shot classification achieves high accuracy, producing a reliable class label that translates directly into a strong text query. The trained methods cannot replicate this behaviour: their architectures require the visual embedding to be present, and they have no mechanism to discard it when it becomes harmful.

5.4.4 Summary of SOTA Comparisons

Table 5.11 brings all results together.

Table 5.11: Summary: our parameterized best vs. best trained SOTA per metric. ✓ = we surpass best SOTA. ≈ = within 2%. ∘ = within 10%. × = gap >10%.

Dataset	Metric	Ours	Best SOTA	SOTA value	Gap	Status
Ext. Sketchy S1	mAP@All	0.668	ZSE-Ret	0.736	−9.2%	∘
Ext. Sketchy S1	P@100	0.760	ZSE-Ret	0.808	−5.9%	∘
Ext. Sketchy S2	mAP@200	**0.761**	Koley et al.	0.746	+2.0%	✓
Ext. Sketchy S2	P@200	0.746	Koley et al.	0.747	−0.1%	≈
TU-Berlin	mAP@All	0.605	Koley et al.	0.680	−11.0%	×
TU-Berlin	P@100	0.741	Koley et al.	0.744	−0.4%	≈
QuickDraw	mAP@All	**0.437**	Koley et al.	0.231	+89.2%	✓
QuickDraw	P@200	**0.568**	Koley et al.	0.397	+43.1%	✓

S1 = Sketchy-1-Ext (Split 1); S2 = Sketchy-2-Ext (Split 2).

The results show a clear pattern. On **QuickDraw**, our method beats all trained methods by a large margin — over 89% better than the best trained method on mAP@All. On **Extended Sketchy Split 2**, our mAP@200 (0.761) is the highest among all methods, and our P@200 (0.746) matches Koley et al. (0.747) within 0.1%. On **TU-Berlin**, our P@100 (0.741) is within 0.4% of the best trained method (0.744), and we surpass both ZSE variants and CLIP-for-All. The only area where trained methods stay clearly ahead is global mAP@All on Extended Sketchy Split 1 and TU-Berlin — where training explicitly optimises full-gallery ranking. Achieving competitive or superior results on all other metrics without any training demonstrates the effectiveness of the proposed approach.

5.4.5 Visual Comparison with ZSE-RN

Figure 5.7 provides a qualitative side-by-side comparison between ZSE-RN [22] (left) and our method (right) on six query sketches: three from TU-Berlin (rollerblades, parachute, pizza) and three from Sketchy (cup, deer, windmill). Each row shows the five top-ranked images retrieved by each method for the same query sketch. To our knowledge, ZSE-RN is the only SOTA method that publishes retrieval visualisations for these categories, making this the only direct visual comparison possible against a published result.

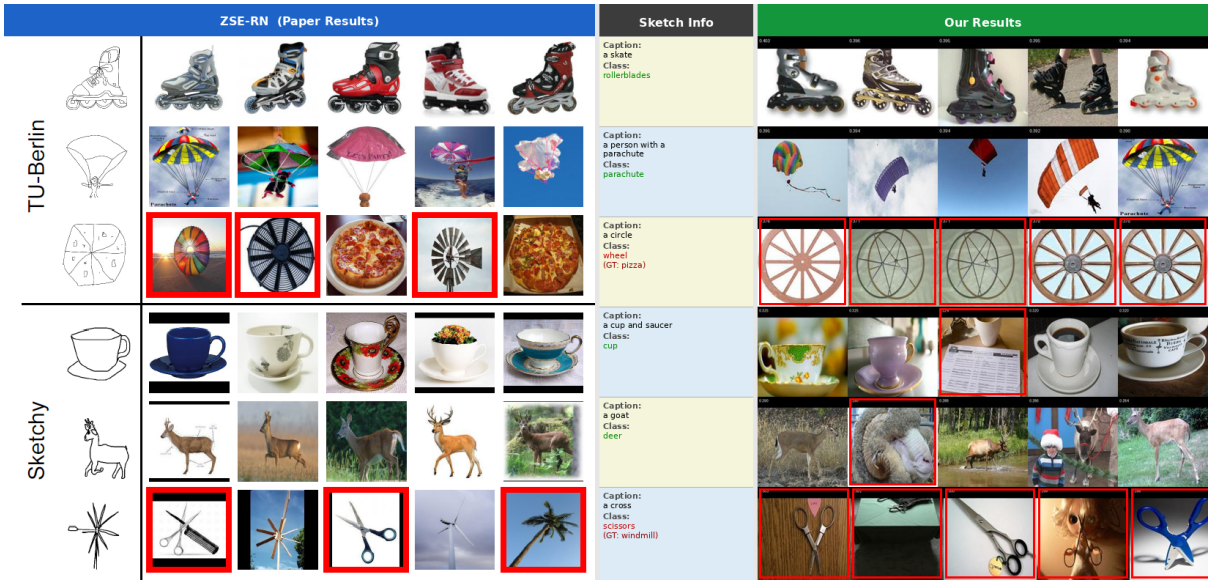

Figure 5.7: Visual comparison of top-5 retrieval results between ZSE-RN (left, blue) and our training-free method (right, green) on six query sketches. The middle strip shows the BLIP-generated caption and CLIP-predicted class for each query (class shown in green if correct, red if wrong). Red boxes mark incorrect retrievals in our results.

The middle strip in Figure 5.7 reveals the direct cause of each success or failure by showing the BLIP caption and the CLIP-predicted class for every query sketch.

On **rollerblades** and **parachute**, both the caption and the predicted class are correct ("a skate" / *rollerblades*; "a person with a parachute" / *parachute*). This gives the retrieval two consistent signals pointing to the right category, and both methods return correct images with strong visual similarity to the query sketch.

On **cup** and **deer**, the predicted class is correct (*cup* and *deer* respectively). For the cup query, the BLIP caption "a cup and saucer" is notably accurate — it correctly captures the sketch content, and as a result all retrieved images show cups with saucers, demonstrating that a precise caption directly improves retrieval quality. For the deer query, the caption "a goat" is incorrect, but the class prediction compensates and most retrievals are correct. This is consistent with the optimal parameters found for Sketchy ($\beta = 0$, $\alpha = 0.4$): no visual feature is used ($\beta = 0$), and the semantic component weights class labels at 60% against captions at 40%, giving the class prediction enough influence to override a noisy caption. The single failures in each row (red boxes) are visually related to the query: the retrieved images belong to semantically close categories rather than completely unrelated objects, confirming that the embedding is still pointing in the right semantic direction.

On **pizza** and **windmill**, both signals fail simultaneously. BLIP describes the pizza sketch as "a circle" and the windmill as "a cross" — descriptions with no connection to the correct category. The predicted class is also wrong: *wheel* for pizza and *scissors* for windmill. Despite this, the retrieved images are not random. For pizza, the round

shape of the sketch causes the system to retrieve other circular objects, and for windmill, the cross-like structure leads to geometrically similar results. The retrieved images are therefore visually related to the sketch even if they belong to the wrong category — a behaviour that is difficult to avoid without task-specific training on these categories. ZSE-RN returns correct images in these two cases, which is expected given that it was trained on labelled sketch–photo pairs specifically designed to bridge the domain gap — a resource our method does not use at all.

Overall, the qualitative results confirm a clear and encouraging pattern: even in failure cases, our method retrieves images that are visually plausible given the sketch shape, rather than completely unrelated content. This suggests that the embedding framework is working correctly, and that the failures are caused by incorrect text signals rather than a fundamental inability to match sketches to images. Improving caption quality and class prediction for abstract sketches is therefore the most direct path to further performance gains.

5.5 Cross-Dataset Analysis and Key Findings

5.5.1 The Semantic-Surpasses-Visual Paradox

Across all experiments, text-based features — captions and class labels — consistently outperform or match visual sketch features. This is surprising because SBIR is typically assumed to be a visual matching problem.

The reason is that sketches are abstract and lack the texture, colour, and detail of natural photos. When CLIP processes a sketch image, it struggles to extract meaningful features because the sketch looks very different from the natural images CLIP was trained on. By contrast, when CLIP processes a text description of the sketch — either a BLIP-generated caption or a predicted class label — it works as expected and produces rich, informative embeddings that align well with real photo embeddings.

In practice, this means that generating a good text description of a sketch is more valuable than trying to extract better visual features from it, especially when no task-specific training is available.

5.5.2 Dataset-Specific Optimal Strategies

The experiments reveal that no single configuration is universally optimal across all datasets. The optimal strategy—in terms of both β (visual–semantic balance) and α (caption–class balance within the semantic component)—is determined by dataset characteristics:

- **Sketchy** (125 categories, $\leq$100 photos/class, category-level retrieval): C6 with $\alpha =$

0.4 (40% caption, 60% class) achieves the best global ranking, while C7 achieves the best Acc@K. The meaningful caption contribution reflects the benefit of descriptive text for a well-structured dataset where the gallery contains only a small number of images per category. Note that P@200 is bounded by the limited gallery size (at most 100 images per category).

- **Extended Sketchy** (large-scale, unpaired): C6 with $\alpha = 0.2$ (20% caption, 80% class) achieves the best global ranking. The reduced caption weight compared to Sketchy reflects the need for stronger class-level signals to handle a larger, more diverse 73K-image gallery.

- **TU-Berlin** (high diversity, unpaired, 250 classes): the best results are achieved with $\alpha = 0.3$ and $\beta = 0.1$. The balanced α reflects the fact that both captions and class labels are useful here — the large number of categories benefits from class-level signals, while the diversity of drawing styles means captions still add useful information. The small $\beta = 0.1$ indicates that adding a little visual information also helps, unlike the other datasets where visual features are not needed.

- **QuickDraw** (110 categories, very abstract, quickly drawn sketches): the best result uses $\alpha = 0.0$ and $\beta = 0.0$, meaning only class labels are used. BLIP captions are unreliable for such abstract sketches — they often produce generic descriptions like "a drawing of lines" that carry no useful information. Visual features also hurt performance rather than helping. Class labels alone are the only reliable signal on this dataset.

These results suggest that a practical SBIR system should not use fixed weights for all sketches. Instead, it could estimate how abstract a sketch is — for example, by using the probability score of CLIP's top class prediction as a measure of how recognisable the sketch is — and adjust α and β accordingly: higher values for clear, detailed sketches where captions and visual features are useful, and $\alpha = 0$, $\beta = 0$ for very abstract sketches where only the class label matters.

5.5.3 Generation vs. Direct Retrieval

Across all datasets, direct retrieval consistently outperforms the generation-based approach. Three reasons explain this:

1. **Generated images look different from real photos.** Stable diffusion produces images with artificial textures and lighting that do not match real photographs. It also adds details that were not in the original sketch — such as background objects, colours, or surface patterns — which may mislead CLIP when encoding

the image. In some cases, these added details dominate the embedding more than the intended object itself, causing CLIP to focus on the background rather than the main subject. This can result in retrieving visually related images that belong to a different category — an effect observed in qualitative evaluation. Even when the correct object is generated, CLIP encodes it differently from a real photo of the same object, so the retrieved results are less accurate. Direct retrieval avoids this entirely by using text embeddings, which CLIP was trained to match with real photo embeddings.

2. **The quality of the generated image depends on the text prompt.** If the caption is vague or the class label is wrong, stable diffusion cannot produce a useful image. This is especially problematic for abstract sketches like those in QuickDraw, where BLIP often generates meaningless descriptions.

3. **Errors multiply across two steps.** In generation-based retrieval, a wrong text prompt produces a wrong image, and a wrong image produces a wrong embedding. Direct retrieval has only one step — from sketch to text embedding — so errors do not compound in this way.

These findings suggest that generative models are more useful for showing users a realistic preview of what their sketch represents, rather than as the main retrieval mechanism.

The qualitative behaviour of both prompt strategies is illustrated in Figure 5.6 (Section 5.3), which shows that the caption prompt succeeds across all four datasets while the class prompt fails on TU-Berlin due to a wrong class prediction cascading into a wrong generated image and wrong retrievals.

5.5.4 Zero-Shot Generalisation vs. Task-Specific Training

Trained methods generally perform better on individual datasets, but they require labelled sketch–photo pairs, significant computation for training, and may not generalise well beyond the dataset they were trained on. The proposed approach uses no training at all, which makes it simpler to apply and more flexible across different datasets. As shown in Table 5.11, this comes at a small cost: our method matches or outperforms trained methods on most metrics, and only falls behind on global mAP@All where training directly optimises full-gallery ranking.

5.6 Limitations

This work has the following limitations:

1. **Category-level retrieval only, with mostly unpaired datasets.** The entire evaluation is done at the category level; instance-level retrieval — where a sketch must match a specific photo — is not addressed. Three out of four datasets (Extended Sketchy, TU-Berlin, and QuickDraw) do not have direct sketch–photo pairs, meaning there is no ground truth for whether a retrieved photo visually matches the query sketch, only whether it belongs to the same category.

2. **Fixed generation steps.** The generation-based approach uses a fixed 25-step inference schedule due to hardware constraints. Using more steps could produce higher-quality images and potentially improve retrieval results.

3. **Simple feature fusion.** The fusion of caption, class, and visual features uses simple weighted averaging. More advanced fusion methods, such as attention-based mechanisms, could better capture the relative importance of each feature per query, but would require training data.

4. **Basic caption post-processing.** The BLIP caption pipeline uses rule-based cleaning that may occasionally remove useful words. More advanced caption refinement could improve the quality of the text descriptions and lead to better retrieval.

5. **Closed-world class prediction.** The zero-shot class prediction step relies on a predefined list of category names supplied at evaluation time. CLIP selects the class whose text embedding is most similar to the sketch embedding, but this comparison is limited to the provided list. Any sketch whose true category is absent from the list will be misclassified, and the system cannot generalise to open-world queries outside the known vocabulary. Although this approach yields the strongest retrieval results among the configurations tested — because clean class labels produce more discriminative embeddings than automatically generated captions — it remains a fundamental constraint on real-world deployability.

5.7 Chapter Summary

This chapter presented the experimental results for both direct retrieval and generation-based retrieval across four benchmark datasets. The key findings are:

1. **Text-based features outperform visual features.** For category-level zero-shot SBIR, CLIP-predicted class labels and BLIP-generated captions consistently outperform raw visual sketch features. Class labels alone are the most reliable signal across all datasets.

2. **The best configuration depends on the dataset.** Datasets with more structured sketches and smaller galleries benefit from combining captions and class labels, while highly abstract datasets such as QuickDraw require class labels only. TU-Berlin is the only dataset where adding a small visual weight also helps.

3. **Direct retrieval outperforms generation-based retrieval.** Generated images differ from real photos, the quality of generation depends on the text prompt, and errors in the prompt carry over into the retrieval results. Direct fusion of text embeddings avoids all three problems.

4. **The proposed training-free approach matches or surpasses trained SOTA methods** on most metrics, as shown in Table 5.11. The only consistent gap is global mAP@All on Extended Sketchy Split 1 and TU-Berlin, where trained methods directly optimise full-gallery ranking.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis asked a simple question: can pretrained models be combined to retrieve photos from sketch queries without any task-specific training? The answer, based on experiments across four datasets, is yes.

The most important finding is that text-based features — class labels and captions — consistently work better than visual sketch features. The reason is straightforward: sketches are abstract and look very different from the natural photos that CLIP was trained on, so CLIP struggles to extract useful visual features from them. But when the same sketch is described in text, CLIP handles it well and produces embeddings that match real photo embeddings closely. In short, describing a sketch in words is more effective than trying to match it visually.

A second finding is that the best combination of features depends on the dataset. For well-structured datasets like Sketchy, adding caption information helps. For very abstract datasets like QuickDraw, only class labels work. For TU-Berlin, which has many categories and sketches drawn by many different people, a small amount of visual information also helps. This pattern suggests that a practical system could automatically choose the right combination based on how clear or abstract the input sketch is.

For the generation-based approach, converting sketches into realistic photos before retrieval does not help — it actually hurts performance. The generated images look different from real photos, and stable diffusion adds artificial details that can mislead the retrieval. Using text descriptions directly is simpler and more accurate. Generative models are still useful for showing users what their sketch looks like as a realistic photo, but not as the main retrieval tool.

Overall, combining CLIP and BLIP without any training is a strong approach to sketch-based image retrieval. It outperforms all trained methods on the hardest dataset, matches them on most other metrics, and only falls behind on full-gallery ranking where training gives a direct advantage.

6.2 Future Work

The findings and limitations of this thesis open several promising research directions.

1. **Automatic weight selection based on sketch clarity.** The best values of α and β depend on how abstract the sketch is. A useful extension would be a system that automatically estimates how clear or abstract a sketch is — for example, using the confidence score of CLIP's class prediction — and adjusts the weights accordingly, without needing a separate parameter search for each dataset.

2. **Instance-level retrieval.** The current framework only retrieves images from the same category as the query sketch. A harder and more useful task is to retrieve the specific photo that matches the sketch. This would require more detailed feature representations and may benefit from training CLIP on a small set of sketch–photo pairs.

3. **Better caption generation for abstract sketches.** BLIP often produces poor or generic captions for abstract sketches like those in QuickDraw. Using more advanced multimodal models that are better at understanding abstract visual inputs could produce more useful descriptions and improve retrieval.

4. **Better use of generative models for retrieval.** Instead of using the final generated image for retrieval, future work could use the internal representations produced at intermediate steps of the generation process. These may be closer to real photo representations and less affected by the artificial details added by stable diffusion.

5. **Smarter feature fusion.** The current fusion uses fixed weights. A method that learns to adjust the importance of each feature based on the specific query sketch could improve results, particularly on global ranking, while requiring only a small amount of labelled training data.

6. **Stronger pretrained models and open-world class prediction.** This thesis used CLIP ViT-B/32 and BLIP-base, which are relatively small models. Replacing them with larger variants or modern Vision-Language Models such as LLaVA [43] or Qwen-VL [44] would both improve visual understanding and caption quality, and eliminate the closed-world constraint — generating class labels directly from any sketch without a predefined list, while keeping the retrieval backbone unchanged. This direction is particularly appealing because experiments in this thesis show that clean class labels produce the most discriminative query embeddings; a VLM that generates equally precise labels from open-world sketches would therefore combine

the accuracy advantage of class-conditioned retrieval with the generalisability of an open-vocabulary system.

In conclusion, this thesis demonstrates that pretrained vision–language and captioning models, when combined in the right way, provide a powerful and generalisable foundation for zero-shot SBIR. The finding that a training-free semantic fusion approach can surpass all trained SOTA methods on the most challenging benchmark challenges the assumption that task-specific supervision is necessary for competitive cross-modal retrieval, and points toward a research direction where foundation models progressively reduce — and in some conditions already eliminate — the advantage that training was thought to provide.

Bibliography

[1] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa, "Sketch-based image retrieval: Benchmark and bag-of-features descriptors," *IEEE transactions on visualization and computer graphics*, vol. 17, no. 11, pp. 1624–1636, 2010.

[2] S. Dey, P. Riba, A. Dutta, J. Llados, and Y.-Z. Song, "Doodle to search: Practical zero-shot sketch-based image retrieval," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2179–2188.

[3] 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 *International conference on machine learning*, PmLR, 2021, pp. 8748–8763.

[4] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *International conference on machine learning*, PMLR, 2022, pp. 12 888–12 900.

[5] F. Yang, N. A. Ismail, Y. Y. Pang, V. R. Kebande, A. Al-Dhaqm, and T. W. Koh, "A systematic literature review of deep learning approaches for sketch-based image retrieval: Datasets, metrics, and future directions," *IEEE Access*, vol. 12, pp. 14 847–14 869, 2024.

[6] H. Yu, M. Huang, and J. J. Zhang, "Domain adaptation problem in sketch based image retrieval," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 19, no. 3, pp. 1–17, 2023.

[7] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[8] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 3836–3847.

[9] Wikipedia contributors, *Cosine similarity — Wikipedia, the free encyclopedia*, `https://en.wikipedia.org/w/index.php?title=Cosine_similarity&oldid=1357144414`, 2026.

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

[11] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[12] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *IEEE transactions on big data*, vol. 7, no. 3, pp. 535–547, 2019.

[13] C. Xiao, C. Wang, L. Zhang, and L. Zhang, "Sketch-based image retrieval via shape words," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, 2015, pp. 571–574.

[14] B. Klare and A. K. Jain, "Sketch-to-photo matching: A feature-based approach," in *Biometric technology for human identification VII*, SPIE, vol. 7667, 2010, pp. 11–20.

[15] R. Hu and J. Collomosse, "A performance evaluation of gradient field hog descriptor for sketch based image retrieval," *Computer Vision and Image Understanding*, vol. 117, no. 7, pp. 790–806, 2013.

[16] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa, "A descriptor for large scale image retrieval based on sketched feature lines.," *SBIM*, vol. 9, pp. 29–36, 2009.

[17] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa, "Photosketch: A sketch based image query and compositing system," in *SIGGRAPH 2009: talks*, 2009, pp. 1–1.

[18] Y. Qi, Y.-Z. Song, H. Zhang, and J. Liu, "Sketch-based image retrieval via siamese convolutional neural network," in *2016 IEEE international conference on image processing (ICIP)*, IEEE, 2016, pp. 2460–2464.

[19] T. Bui, L. Ribeiro, M. Ponti, and J. Collomosse, "Compact descriptors for sketch-based image retrieval using a triplet loss convolutional neural network," *Computer Vision and Image Understanding*, vol. 164, pp. 27–37, 2017.

[20] P. Lu, G. Huang, H. Lin, W. Yang, G. Guo, and Y. Fu, "Domain-aware se network for sketch-based image retrieval with multiplicative euclidean margin softmax," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3418–3426.

[21] Z. Wang, H. Wang, J. Yan, A. Wu, and C. Deng, "Domain-smoothing network for zero-shot sketch-based image retrieval," *arXiv preprint arXiv:2106.11841*, 2021.

[22] F. Lin, M. Li, D. Li, T. Hospedales, Y.-Z. Song, and Y. Qi, "Zero-shot everything sketch-based image retrieval, and in explainable style," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 23 349–23 358.

[23] L. S. F. Ribeiro, T. Bui, J. Collomosse, and M. Ponti, "Sketchformer: Transformer-based representation for sketched structure," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14 153–14 162.

[24] O. Seddati, S. Dupont, S. Mahmoudi, and T. Dutoit, "Transformers and cnns both beat humans on sbir," *arXiv preprint arXiv:2209.06629*, 2022.

[25] S. Gupta, U. Chaudhuri, B. Banerjee, and S. Kumar, "Zero-shot sketch based image retrieval using graph transformer," in *2022 26th International Conference on Pattern Recognition (ICPR)*, IEEE, 2022, pp. 1685–1691.

[26] J. Tian, X. Xu, F. Shen, Y. Yang, and H. T. Shen, "Tvt: Three-way vision transformer through multi-modal hypersphere learning for zero-shot sketch-based image retrieval," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 2370–2378.

[27] A. Pandey, A. Mishra, V. K. Verma, A. Mittal, and H. Murthy, "Stacked adversarial network for zero-shot sketch based image retrieval," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2020, pp. 2540–2549.

[28] V. R. M. K. Gopu and M. Dunna, "Zero-shot sketch-based image retrieval using stylegen and stacked siamese neural networks," *Journal of Imaging*, vol. 10, no. 4, p. 79, 2024.

[29] V. Kumar Verma, A. Mishra, A. Mishra, and P. Rai, "Generative model for zero-shot sketch-based image retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[30] A. Dutta and Z. Akata, "Semantically tied paired cycle consistency for zero-shot sketch-based image retrieval," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5089–5098.

[31] F. Liu, X. Deng, Y.-K. Lai, Y.-J. Liu, C. Ma, and H. Wang, "Sketchgan: Joint sketch completion and recognition with generative adversarial network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5830–5839.

[32] S. Koley, A. K. Bhunia, A. Sain, P. N. Chowdhury, T. Xiang, and Y.-Z. Song, "Text-to-image diffusion models are great sketch-photo matchmakers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 16 826–16 837.

[33] R. Zuo, H. Hu, X. Deng, C. Gao, Z. Zhang, Y. Lai, C. Ma, Y.-J. Liu, and H. Wang, "Scenediff: Generative scene-level image retrieval with text and sketch using diffusion models," 2024.

[34] A. Sain, A. K. Bhunia, P. N. Chowdhury, S. Koley, T. Xiang, and Y.-Z. Song, "Clip for all things zero-shot sketch-based image retrieval, fine-grained or not," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 2765–2775.

[35] P. Sangkloy, N. Burnell, C. Ham, and J. Hays, "The sketchy database: Learning to retrieve badly drawn bunnies," *Acm Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–12, 2016.

[36] L. Liu, F. Shen, Y. Shen, X. Liu, and L. Shao, "Deep sketch hashing: Fast free-hand sketch-based image retrieval," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2862–2871.

[37] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" *ACM Transactions on graphics (TOG)*, vol. 31, no. 4, pp. 1–10, 2012.

[38] J. Jongejan, H. Rowley, T. Kawashima, J. Kim, and N. Fox-Gieg, "The quick, draw!-ai experiment," *Mount View, CA, accessed Feb*, vol. 17, no. 2018, p. 4, 2016.

[39] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[40] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[41] J. Wang, Z. Yang, X. Hu, L. Li, K. Lin, Z. Gan, Z. Liu, C. Liu, and L. Wang, "Git: A generative image-to-text transformer for vision and language," *arXiv preprint arXiv:2205.14100*, 2022.

[42] P. Wang, A. Yang, R. Men, J. Lin, S. Bai, Z. Li, J. Ma, C. Zhou, J. Zhou, and H. Yang, "Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework," in *International conference on machine learning*, PMLR, 2022, pp. 23 318–23 340.

[43] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[44] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, *Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond*, 2023. arXiv: `2308.12966 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/2308.12966`