B.Tech Project Report

Understanding Image Scenes and Videos by Scene Graph Generation

Submitted to the Department of Information Technology

For the partial fulfilment of the degree of Bachelor of Technology in Information Technology

Presented By:

Randhi Nagasurya (2021ITB015)
Somireddy Naveen Kumar Reddy (2021ITB085)
Moru Sai Tirupathi (2021ITB086)

Under the Supervision of:

Dr. Arindam Biswas
Professor
Department of Information Technology
Indian Institute of Engineering Science and Technology, Shibpur

May 14, 2025

Department of Information Technology
Indian Institute of Engineering Science and Technology, Shibpur
West Bengal, India – 711103

CERTIFICATE

This is to certify that we have examined the thesis entitled **"Understanding Image Scenes and Videos by Scene Graph Generation"**, submitted by **Moru Sai Tirupathi** (Roll Number: *2021ITB086*), **Somireddy Naveen Kumar Reddy** (Roll Number: *2021ITB085*), **Randhi Nagasurya** (Roll Number: *2021ITB015*) undergraduate students of **Department of Information Technology** in partial fulfillment for the award of degree of **Bachelor of Technology**. We hereby accord our approval of it as a study carried out and presented in a manner required for its acceptance in partial fulfillment for the under-graduate degree for which it has been submitted. The thesis has fulfilled all the requirements as per the regulations of the institute and has reached the standard needed for submission.

Head of Department	**Supervisor**
Dr. Tuhina Samanta,	Dr. Arindam Biswas,
Dept. of Information Technology,	Dept. of Information Technology,
IIEST, Shibpur.	IIEST, Shibpur.

Examiners:

1. _____
2. _____
3. _____

Place: Shibpur
Date: .....

Acknowledgment

We are profoundly grateful to Dr. Arindam Biswas, Professor, Department of Information Technology, Indian Institute of Engineering Science And Technology, Shibpur, for his valuable guidance, constant support, and encouragement throughout the course of this project. His expertise and suggestions were instrumental in overcoming challenges and achieving our project objectives.

We also extend our sincere thanks to the Department of Information Technology, Indian Institute of Engineering Science And Technology, Shibpur, for providing the necessary infrastructure and resources that enabled us to undertake this project.

Lastly, we are thankful to our families and friends for their unwavering support and motivation, which inspired us to complete this project successfully.

Randhi Nagasurya (2021ITB015)
Somireddy Naveen Kumar Reddy (2021ITB085)
Moru Sai Tirupathi (2021ITB086)

Contents

Abstract

Understanding a visual scene goes beyond recognizing individual objects in isolation. Relationships between objects also constitute rich semantic information about the scene. In this work, we explicitly model the objects and their relationships using **Scene Graphs**, a visually grounded graphical structure of an image or video. Our approach employs a YOLO-based architecture for efficient object detection. This architecture ensures accurate detection of objects, while **Region of Interest (ROI) alignment** further enhances feature representation for each detected object.

Our **Scene Graph Generation (SGG)** framework is capable of extracting meaningful visual and spatial relationships between objects in both images and videos. The model produces a structured graphical representation where objects are represented as nodes, and their interactions are captured as edges, providing a rich semantic understanding of the visual scene. Unlike conventional methods, our approach seamlessly extends to dynamic scenes, making it suitable for real-time video analysis.

Chapter 1

Introduction

The field of computer vision has made remarkable strides over the past few decades in its ability to understand and interpret complex visual data. Initially, image understanding focused on low-level features such as detecting edges, colors, and textures, which were effective in identifying simple patterns. However, these methods were limited in their ability to model the intricacies of real-world scenes. As a result, the focus shifted to higher-level representations capable of modeling relationships and interactions between objects.

In the early stages, object recognition techniques were a primary focus, allowing systems to identify individual objects in an image. However, these systems struggled to provide context or understand the interactions between objects. For example, a system might be able to detect a "cat" or "ball" but would fail to understand their relationship, such as whether the cat was playing with the ball or simply sitting next to it. This limitation led to the emergence of scene understanding, which focused not just on detecting objects, but also on representing how they are related spatially and temporally.

Table 1.1 summarizes the evolution of image understanding from basic object recognition to more complex scene representations:

Stage	**Key Focus**
Early Image Understanding	Low-level features (edges, textures)
Object Recognition	Detection of individual objects
Scene Understanding	Representation of object relationships and context

Table 1.1: Progression of image understanding techniques

One of the most powerful tools that has emerged for scene understanding is the *scene graph*. Scene graphs represent images as a network of nodes (objects) and edges (relationships between objects), providing a rich, structured representation that can capture not only the objects but also their relationships in a visual scene. For instance, a scene graph can describe interactions like "a person sitting on a chair" or "a dog chasing a ball," providing much deeper context than traditional object detection methods.

Scene graphs have become essential in vision-language tasks such as image captioning, visual question answering (VQA), and robotics, where understanding the relationships between objects is critical. One key innovation in scene graph development has been the integration of external knowledge sources, such as WordNet or ConceptNet, which allows scene graphs to reason beyond the visual content of an image. This embedding of knowledge enables models to infer relationships that may not be explicitly visible in the

image. For example, a scene graph can infer that a "cat" is a type of "animal," even if only a part of the cat is visible.

Additionally, hierarchical structures within scene graphs organize objects and relationships into categories, allowing for more efficient reasoning across different contexts. By grouping objects like "dogs" and "cats" under the broader category "animals," scene graphs facilitate better generalization across a range of scenes, improving tasks like VQA.

A pivotal dataset in the development of scene graphs is the Visual Genome dataset, which provides over 100,000 images annotated with detailed scene graphs. These annotations include object categories, their attributes, and relationships, along with region descriptions and question-answer pairs. The Visual Genome dataset has become an invaluable resource for training and evaluating scene graph generation models, significantly advancing the state of the art in image understanding.

In conclusion, scene graphs have become a cornerstone of modern image understanding. By modeling both objects and their relationships in a structured and relational way, scene graphs provide a more comprehensive and nuanced understanding of images. As they continue to evolve, particularly with the incorporation of external knowledge and hierarchical structures, scene graphs will remain a critical tool in advancing vision-language understanding and enabling more accurate and sophisticated models for a wide range of applications.

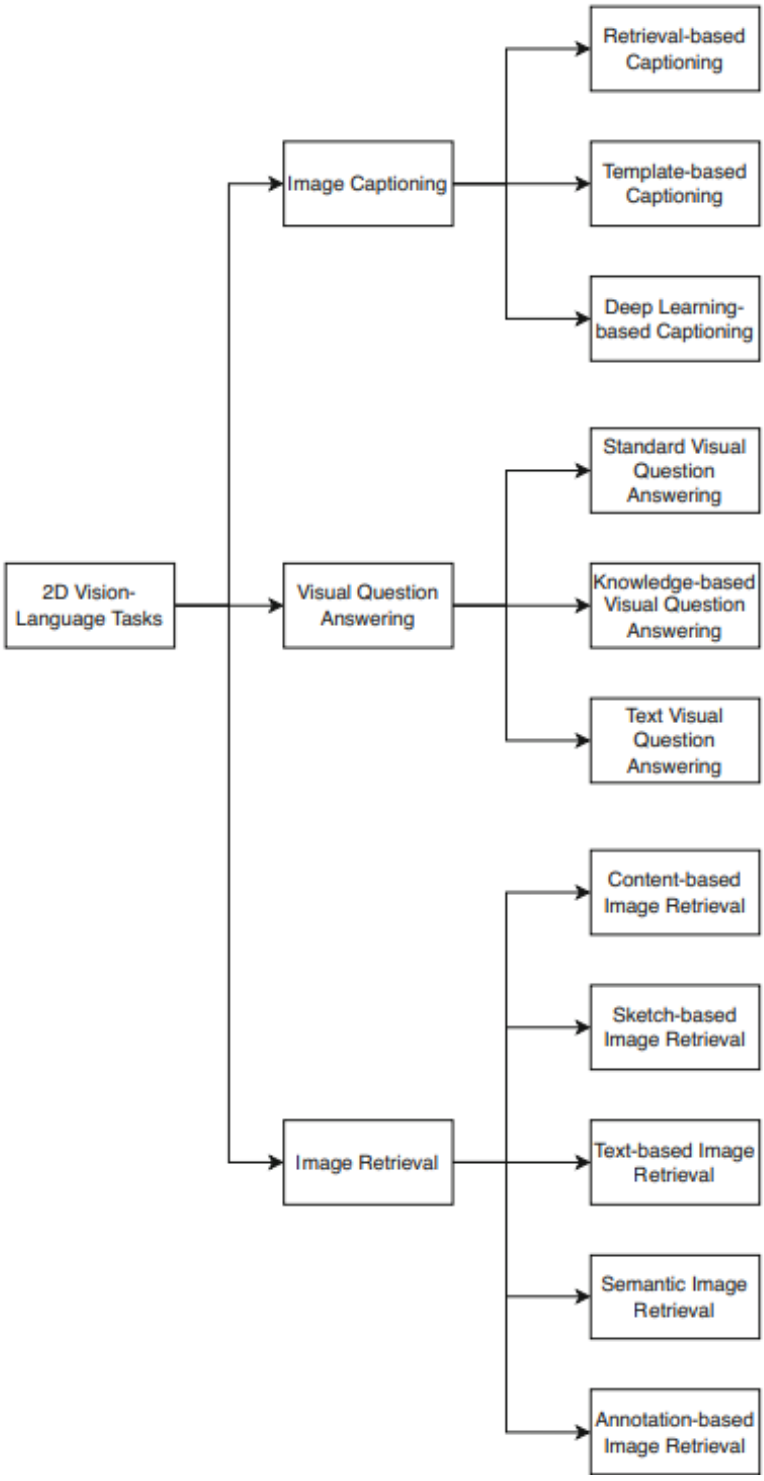

Figure 1.1: 2D Vision-Language Task Taxonomy

Chapter 2

Literature Review

2.1 Visual Relationship Detection with Language Priors

Cewu Lu et al. (ECCV 2016) Cewu Lu et al. introduced the novel task of visual relationship detection, focusing on identifying relationships between object pairs in an image. They proposed a joint optimization framework that simultaneously learns visual features through Convolutional Neural Networks (CNNs) and linguistic relationships using word embeddings and language priors.

- **Dataset:** A new benchmark dataset for visual relationship detection was provided.
- **Framework:** The model combines visual and language models to predict relationships such as *"man-riding-horse"*.
- **Inference:** A probabilistic framework integrates contextual knowledge for accurate relationship prediction.

Component	Methodology	Example Output
Visual Features	CNNs	Object detection
Linguistic Priors	Word Embeddings	Semantic context
Joint Inference	Probabilistic Model	Relationships

Table 2.1: Key Components of the Visual Relationship Detection Model

2.2 Improving Visual Relationship Detection using Semantic Modeling

Bryan A. et al. (ISWC 2017) Bryan et al. enhanced visual relationship detection by incorporating semantic scene descriptions. A joint embedding of visual and textual features enriched the understanding of object-object interactions.

- **Dataset:** Images paired with natural language descriptions.
- **Approach:** Semantic modeling of visual-textual data improves relationship prediction.

2.3 Scene Graph Generation by Iterative Message Passing

Danfei Xu et al. (CVPR 2017) This study proposed a graph neural network (GNN) to generate scene graphs by representing images as graphs, where nodes are objects and edges are relationships.

- **Architecture:** Iterative message passing between nodes.
- **Output:** Scene graphs describing objects and their interconnections.

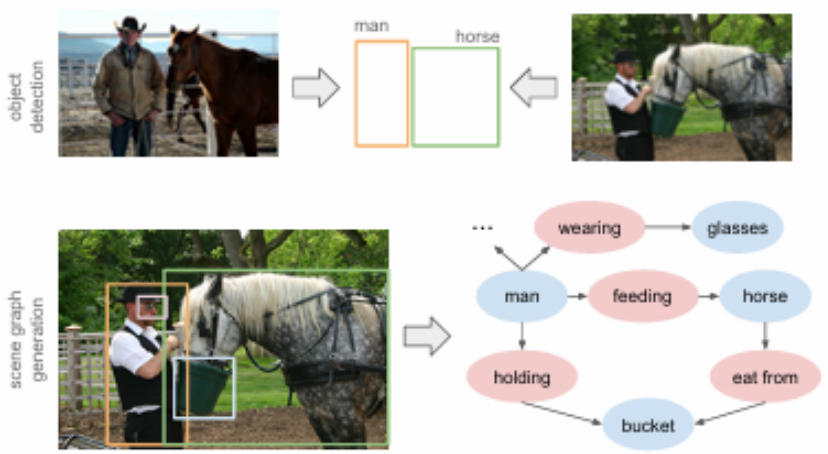

Figure 2.1: A Scene Graph Representation

-

2.4 Generating Triples with Adversarial Networks

Matthew Klawonn et al. (AAAI 2018) Matthew et al. utilized Generative Adversarial Networks (GANs) for scene graph construction, emphasizing triplet generation $(subject, predicate, object)$.

- **Generator:** Produces triplets.
- **Discriminator:** Evaluates triplet plausibility.
- **Training:** Adversarial optimization improves triplet accuracy.

Chapter 3

Panoptic Scene Graph (PSG) Dataset

3.1 Overview

The **Panoptic Scene Graph (PSG)** dataset is a large-scale, richly annotated dataset that unifies *scene graph generation* with *panoptic segmentation*. It is designed to provide detailed annotations of objects, their interactions, and segmentation masks, enabling a holistic understanding of visual scenes.

It includes:

- **Images and Scene Graphs:** Over 49,000 images sourced from the COCO dataset, each annotated with detailed scene graphs.

- **Objects with Panoptic Segmentation:** Every object instance is labeled and assigned a panoptic segmentation mask, allowing precise localization and instance-level distinction.

- **Relationships:** Over 330 types of relationships between object pairs, covering actions, spatial relations, and more.

- **Triplets:** Each image includes multiple `<subject, predicate, object>` triplets, grounding visual relationships in segmentation masks.

This dataset enables research in *scene graph generation*, *vision-language modeling*, *segmentation-aware reasoning*, and *holistic image understanding*.

3.2 Dataset Components

Table 3.1: Summary of PSG Dataset Components

Component	Description	per Image
Objects	Instances with panoptic segmentation and category labels	∼15
Relationships	Annotated pairwise relations among segmented objects	∼10–20
Triplets	`<subject, predicate, object>` entries grounded in segmentation masks	∼18
Panoptic Masks	Per-pixel segmentation of all object and background categories	Full image
Scene Graphs	Graph representation of objects and their relationships	1

3.3 Data Representation

3.3.1 Objects

Objects form the core of the PSG dataset:

- **Panoptic Segmentation:** Every object instance is segmented with per-pixel masks, distinguishing both "things" (countable objects) and "stuff" (amorphous regions).

- **Categories:** Mapped to COCO classes, ensuring compatibility with established benchmarks.

- **Statistics:** An average of 15 object instances per image.

These fine-grained segmentations provide a robust foundation for both object detection and image understanding.

3.3.2 Relationships

Relationships describe how objects interact or relate:

- **Types:** Includes spatial (e.g., "on", "next to"), functional (e.g., "holding", "riding"), and semantic (e.g., "looking at") relationships.

- **Triplet Representation:** Each relationship is expressed as a directed triplet — `(subject, predicate, object)`.

- **Annotation:** Grounded in segmentation masks for both subject and object.

- **Statistics:** Each image contains approximately 18 triplets.

Relationships are essential for constructing structured scene representations and enabling reasoning about object interactions.

3.3.3 Triplets and Scene Graphs

- **Triplets:** Central to the dataset, each triplet connects two segmented objects via a relationship predicate.

- **Grounding:** All entities (subjects and objects) in a triplet are explicitly tied to their panoptic segmentation masks.

- **Scene Graphs:** Each image's collection of triplets forms a comprehensive scene graph, representing the structure and meaning of the visual content.

3.3.4 Panoptic Segmentation

- **Full Coverage:** All pixels in the image are labeled as belonging to a specific object or background class.

- **Unified Labeling:** Combines both semantic segmentation (for "stuff") and instance segmentation (for "things") in a coherent format.

- **Standard:** Follows the panoptic segmentation annotation protocol used in COCO.

This level of annotation enables high-precision visual tasks and integration with other segmentation-based datasets.

3.4 Applications

The PSG dataset supports a wide range of tasks at the intersection of vision and reasoning:

- **Panoptic Scene Graph Generation (PSG Task):** Predict a scene graph where each entity is grounded with a segmentation mask.

- **Scene Graph Prediction with Segmentation:** Integrate object interaction modeling with pixel-level accuracy.

- **Visual Reasoning:** Enable structured reasoning over segmented images and relationships.

- **Compositional Learning:** Study generalization to unseen combinations of object pairs and relationships.

- **Vision-Language Tasks:** Serve as a bridge for tasks like image captioning, VQA, and multimodal grounding.

By combining segmentation and structured relationships, the PSG dataset sets a new standard for holistic visual understanding, offering deep insights into both spatial structure and object interactions in natural images.

Chapter 4

Model Architecture

4.1 Extracting Visual and Spatial Features using YOLO

In this section, we present the feature extraction process using YOLO (You Only Look Once), which is a state-of-the-art object detection framework. Our approach leverages CSPNet for efficient backbone processing, PAN-FPN for multi-scale feature fusion, and a detection head for accurate object localization.

4.1.1 YOLO Backbone - Cross Stage Partial Network (CSPNet)

Purpose: The backbone of our YOLO model is powered by CSPNet (Cross Stage Partial Network), designed to extract hierarchical features from the input image efficiently.

Architecture Overview: CSPNet splits the input feature map into two paths:

- One path undergoes a series of convolutional layers, capturing complex features.
- The other path bypasses these computations, preserving the initial feature information.

After processing, both paths are merged using a 1×1 convolution. This design enhances gradient flow, reduces computation, and maintains a rich feature representation.

Hierarchical Feature Extraction:

- **Low-Level Features (C1):** Detect edges, colors, and textures.
- **Mid-Level Features (C3):** Recognize object parts and shapes.
- **High-Level Features (C5):** Identify complete objects and their semantics.

Output Sizes: The backbone produces feature maps of varying resolutions:

$$C3: \ 80 \times 80 \times 256, \quad C4: \ 40 \times 40 \times 512, \quad C5: \ 20 \times 20 \times 1024$$

Advantages of CSPNet:

- Reduces redundant computation, making the model faster.
- Enhances gradient flow during training, improving learning efficiency.

4.1.2 YOLO Neck – PAN-FPN

Purpose: The YOLO Neck combines two architectures — Path Aggregation Network (PAN) and Feature Pyramid Network (FPN) — to efficiently fuse features from multiple scales.

Architecture Overview:

- **Top-Down Pathway (FPN):**
  - Starts from the deepest layer (high-level features).
  - Gradually upsamples and merges features with shallower layers.
  - Enhances semantic context, making it suitable for detecting small objects.
- **Bottom-Up Pathway (PAN):**
  - Starts from the shallowest layer (low-level features).
  - Gradually downsamples, refining spatial details.
  - Improves localization, making it ideal for large object detection.

Output Sizes: The PAN-FPN produces feature maps of different resolutions:

$$P3: \ 80 \times 80 \times 256, \quad P4: \ 40 \times 40 \times 512, \quad P5: \ 20 \times 20 \times 1024$$

Why PAN-FPN?

- Combines fine-grained details (from shallow layers) with high-level semantics (from deep layers).
- Ensures the model can detect objects of varying sizes, from tiny to large.

4.1.3 YOLO Detection Head

Purpose: The YOLO detection head is responsible for predicting object locations, classes, and confidence scores directly on the fused feature maps from PAN-FPN.

Core Components:

- **Bounding Box Prediction:** Determines object positions (center, width, height).
- **Objectness Score:** Represents the confidence of object presence.
- **Class Probabilities:** Predicts the class (e.g., person, car, dog).

How it Works: The detection head applies a series of lightweight convolutional layers to each feature map (e.g., $P3, P4, P5$) and outputs predictions for each region. This efficient design enables fast and accurate detection.

4.1.4 Non-Maximum Suppression (NMS)

Purpose: NMS is a post-processing step used to eliminate redundant or overlapping bounding boxes, retaining only the most confident predictions.

How it Works:

1. Sort all detected boxes by confidence score.
2. Retain the highest confidence box and suppress boxes with high overlap (based on Intersection over Union, IoU).
3. Repeat until no overlapping boxes remain.

Intersection over Union (IoU): IoU measures the overlap between two boxes:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

4.1.5 Spatial Feature Extraction

Purpose: Extract spatial information for each detected object, which helps in understanding their relative positions.

Spatial Features Include:

- Normalized coordinates of the object's center (x, y).
- Normalized width and height of the object.

Spatial Feature Vector:

$$\mathbf{s} = [x_{norm}, y_{norm}, w_{norm}, h_{norm}]$$

These features help in identifying spatial relationships (e.g., "on top of", "next to").

4.1.6 ROI Align (Region of Interest Align)

Purpose: Extract accurate, object-aligned visual features from the feature maps.

Working Mechanism:

- Maps each detected object's bounding box onto the feature map.
- Samples feature values using bilinear interpolation, maintaining spatial accuracy.
- This ensures that each object's feature representation is correctly aligned, improving recognition.

Why ROI Align?

- Provides precise feature extraction without the misalignment issues of traditional ROI pooling.
- Enhances the model's ability to understand the object's visual context.

4.2 Relation Prediction in Scene Graph Generation

In scene graph generation, each image is parsed into objects and their pairwise relations. After detecting and classifying objects in an image, the goal of *relation prediction* is to label the predicate between each subject-object pair [**?**]. In other words, given a candidate pair of objects (subject s and object o), the model predicts the most likely relation $r \in \mathcal{P}$ (e.g., "person rides horse") with an associated confidence score.

4.2.1 Feature Representation and Fusion

To predict relations, we build a feature vector that combines information from the subject, the object, and their spatial configuration. Concretely, let $\mathbf{f}_s \in \mathbb{R}^d$ and $\mathbf{f}_o \in \mathbb{R}^d$ be the visual feature embeddings (from a CNN) of the subject and object, respectively, and let $\mathbf{f}_{sp} \in \mathbb{R}^k$ encode spatial/geometric information (e.g., normalized bounding-box coordinates, relative distance). In addition, semantic cues (such as object class embeddings) can also be included. All features (visual, spatial and semantic) are then concatenated into a single feature vector:

$$\mathbf{f} = \text{Fuse}(\mathbf{f}_s, \mathbf{f}_o, \mathbf{f}_{sp}) \in \mathbb{R}^m.$$

This multi-modal fusion provides a rich representation of the object pair [**?**].

4.2.2 Neural Classifier and Softmax Scoring

The fused feature $\mathbf{f}$ is fed into a neural classifier (typically several fully-connected layers) to produce a logit vector $\mathbf{z} \in \mathbb{R}^{|\mathcal{P}|}$. For instance, a simple linear classifier has the form

$$\mathbf{z} = W\mathbf{f} + \mathbf{b},$$

where W and $\mathbf{b}$ are learned weights. The network then applies a *softmax* function to convert logits into a probability distribution over relation classes:

$$\hat{p}_r = \frac{\exp(z_r)}{\sum_{p=1}^{|\mathcal{P}|} \exp(z_p)}, \qquad r = 1, \ldots, |\mathcal{P}|, \tag{4.1}$$

where $\hat{p}_r$ is the predicted probability (score) for relation r. The softmax layer ensures that the scores are normalized and sum to 1 [**?**]. During inference, the relation with highest probability is selected as the prediction.

For example, suppose the classifier outputs logits $\mathbf{z} = [2.5,\ 0.5,\ -1.0]$ for three relation

classes {riding, standing, above}. The softmax probabilities are then

$$\sigma(\mathbf{z}) \approx [0.84,\ 0.10,\ 0.06],$$

so the model predicts "riding" with score 0.84. Figure 4.1 outlines this process. Subject features, object features, and spatial features are fused and passed through the classifier and softmax to yield final relation scores.

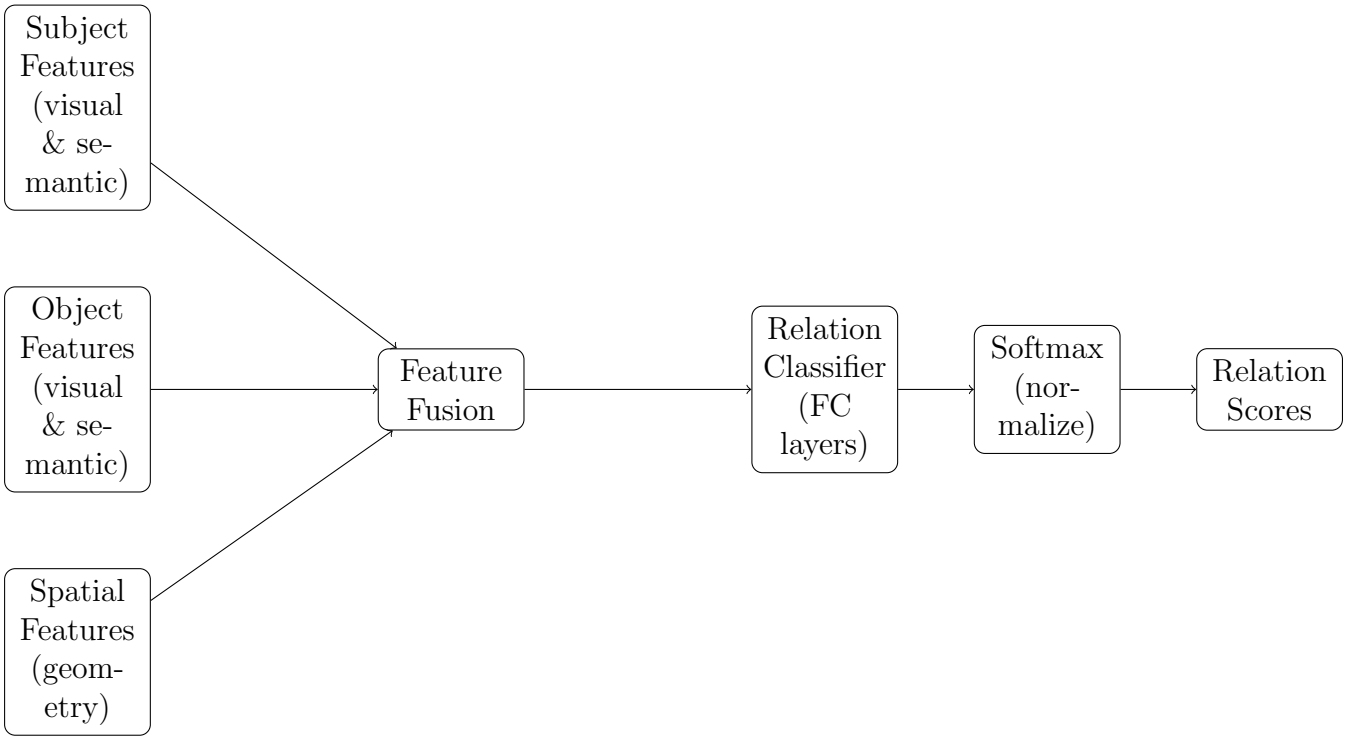

Figure 4.1: Flowchart of relation prediction in scene graph generation. Subject and object visual/semantic features and their spatial features are fused and fed into a neural classifier. The logits are normalized by softmax to produce the relation scores.

Table 4.1 gives a concrete comparison of two real-world scenarios. Each row shows the subject/object pair, the raw logit outputs, and the resulting softmax probabilities and predicted relation. In the first scenario, "riding" has the highest score; in the second, "next-to" is selected.

Table 4.1: Example relation predictions for two scenes. Logits are given for classes (riding, standing, next-to); probabilities are obtained by softmax.

Scene	**Subject**	**Object**	**Logits**	**Softmax**	**Prediction (score)**
Person on horse	Person	Horse	[3.0, 0.2, −1.0]	[0.93, 0.05, 0.02]	riding (0.93)
Person beside horse	Person	Horse	[0.5, 1.0, 2.5]	[0.10, 0.16, 0.74]	next-to (0.74)

In practice, the classifier parameters $W, \mathbf{b}$ are learned from data, and Eq. (4.1) defines the mapping from features to normalized relation scores (e.g., the relation 'under').

Chapter 5

Experimental Results and Analysis

Present your results, graphs, tables, and analysis.

What is Recall@K?

Recall@K is a metric used to evaluate how many correct predictions (typically relationships or triplets in Scene Graph Detection) a model makes among its top K predictions.

Formula:

$$\text{Recall@K} = \frac{\text{Number of correct predicted triplets in top } K}{\text{Total number of ground truth triplets}}$$

Intuition:

If there are 100 ground-truth relationships, and your top 50 predictions include 30 of them:

$$\text{Recall@50} = \frac{30}{100} = 0.30 \quad \text{(or 30\%)}$$

What is mean Recall@K (mR@K)?

Mean Recall@K goes one step further by calculating Recall@K separately for each relationship class, then averaging across all classes.

Formula:

$$\text{mean Recall@K} = \frac{1}{N} \sum_{i=1}^{N} \text{Recall@K for class } i$$

where:

- N is the total number of relationship classes (e.g., "on", "under", "next to", etc.)

Why is mR@K Important?

Recall@K can be dominated by frequent relationships (e.g., "on", "has").

mean Recall@K treats each class equally, no matter how rare it is.

So, mR@K is a better metric when you care about performance on rare relationships, not just the common ones.

Example in SGDet

Suppose your model predicts the following relationships:

- Common ones like *"person-on-horse"*, *"cup-on-table"* (seen many times in training)
- Rare ones like *"book-beside-lamp"*, *"cat-under-chair"* (seen fewer than 10 times)

A model could get high Recall@K by just predicting the common ones well. But the

mean Recall@K would be low if the rare ones are missed, which forces the model to do better across all classes, not just the frequent ones.

Summary

Table 5.1: Comparison of Recall@K and mean Recall@K

Metric	**Focus**	**Biased Toward**	**Good For**
R@K	Overall performance on top-K predicts	Frequent relationships	General performance comparison
MR@K	Average across all relationship classes	None (equal weight per class)	Balanced, fair evaluation

$$\text{mR@K} = \frac{1}{N_{\text{rel}}} \sum_{r=1}^{N_{\text{rel}}} \text{Recall@K}_r \tag{5.1}$$

where:

- mR@K: mean Recall at K
- N_{rel}: Total number of relationship classes
- Recall@K$_r$: Recall for relationship class r, considering top-K predictions per image

Table 5.2: Comparison of VCTree, MOTIFS, PENet, and Our Approach for SGDet mR@50/100

Model	**SGDet mR@50**	**SGDet mR@100**
Motifs	6.6	7.9
VCTree	5.8	6.6
PE-NET	16.7	18.8
Our Approach	12.4	14.5

While Table 5.2 shows that PE-Net outperforms our approach in terms of mean Recall at both 50 and 100 (16.7 and 18.8 vs. 12.4 and 14.5 respectively), it is important to consider the underlying design trade-offs. PE-Net employs a prototype-based embedding framework with a two-shot Faster R-CNN backbone, which is computationally intensive and optimized primarily for static image scene graph generation. In contrast, our model leverages YOLO as the detection backbone, which is significantly faster and computationally lighter. Although this choice results in slightly lower performance metrics on image-based benchmarks, it enables real-time analysis and makes our framework particularly suitable for video scene graph generation, where rapid object detection and low-latency processing are critical. Thus, our approach is better aligned for practical deployment in dynamic environments such as video analytics, surveillance, and streaming-based applications, where speed and efficiency are as crucial as accuracy.

Chapter 6

Conclusions and Future Work

6.1 Conclusion

In this work, we successfully developed a comprehensive Scene Graph Generation (SGG) framework capable of extracting meaningful visual and spatial relationships between objects from both images and videos. Our approach employed a robust YOLO-based architecture for efficient object detection, using Cross Stage Partial Network (CSPNet) as the backbone for hierarchical feature extraction and Path Aggregation Network - Feature Pyramid Network(PAN-FPN) for multi-scale feature fusion. By incorporating precise Region of Interest (ROI) alignment, we ensured accurate feature representation for each detected object, leading to high-quality scene graphs that capture the complex interactions within a visual scene.

The SGG framework effectively converts raw visual data into a structured graphical representation, where objects are represented as nodes and their interactions are modeled as edges. This structured representation lays the foundation for advanced visual understanding and paves the way for more complex computer vision tasks. The flexibility of our framework to work seamlessly with both static images and dynamic videos demonstrates its adaptability, making it a foundational tool for advanced visual understanding tasks.

6.2 Future Work

- **Person-Centric Action Recognition:** We aim to extend the work to not only detect objects but also to recognize and describe the actions of a specific person of interest in a video. This enhancement will involve integrating person re-identification and action recognition modules to generate real-time, context-aware descriptions such as "Person X is walking," "Person X is interacting with another person," or "Person X is picking up an object." This will further elevate the model's capability from static scene understanding to dynamic activity recognition, making it suitable for applications such as automated surveillance, smart video analytics, and personalized activity monitoring.

Chapter 7

References

1. R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, "Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations," *International Journal of Computer Vision (IJCV)*, vol. 123, pp. 32–73, 2017.

2. D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene Graph Generation by Iterative Message Passing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5410–5419, 2017.

3. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 91–99, 2015.

4. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

5. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

6. R. Speer, J. Chin, and C. Havasi, "ConceptNet 5.5: An Open Multilingual Graph of General Knowledge," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 4444–4451, 2017.

7. G. A. Miller, "WordNet: A Lexical Database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

8. J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

9. S. C. Johnson, "Hierarchical Clustering Schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

10. Y. Li, R. Zemel, M. Brockschmidt, and D. Tarlow, "Gated Graph Sequence Neural Networks," in *International Conference on Learning Representations (ICLR)*, 2016.

11. R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural Motifs: Scene Graph Parsing with Global Context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5831–5840, 2018.

12. D. Hendrycks and T. Dietterich, "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations," *arXiv preprint arXiv:1807.01697*, 2018.

13. Y. Guo, L. Gao, X. Wang, Y. Hu, X. Xu, X. Lu, H. T. Shen, and J. Song, "From General to Specific: Informative Scene Graph Generation via Balance Adjustment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16383–16392, 2021.

14. K. Tang, H. Zhang, B. Wu, W. Luo, and W. Liu, "Learning to Compose Dynamic Tree Structures for Visual Contexts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6619–6628, 2019.

15. A. Zareian, S. Karaman, and S.-F. Chang, "Bridging Knowledge Graphs to Generate Scene Graphs," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 606–623, 2020.