ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Khoa Công nghệ Thông tin

REPORT

Nhóm 8

Project: Visual Question Answering

Môn: Khai thác dữ liệu văn bản và ứng dụng

Lớp: *22CNTThuc*

Giảng viên phụ trách môn học:

Nguyễn Trần Duy Minh

Lê Thanh Tùng

Thành Phố Hồ Chí Minh – 2025

1. Thành viên trong nhóm:

Bảng Thông tin nhóm & vai trò:

MSSV	Họ Tên	Vai trò
22127326	Nguyễn Tuấn Phong	Nhóm trưởng
22127328	Đỗ Đức Phú	Thành viên
22127233	Trần Hoàng Linh	Thành viên
20127451	Nguyễn Nhật Cảnh	Thành viên

2. Tổng quan về project:

Trong những năm gần đây, lĩnh vực Visual Question Answering - VQA đã trở thành một trong những hướng nghiên cứu quan trọng trong trí tuệ nhân tạo, kết hợp giữa thị giác máy tính và xử lý ngôn ngữ tự nhiên. Nhiệm vụ chính của VQA là xây dựng một hệ thống có khả năng hiểu nội dung của một hình ảnh và trả lời các câu hỏi liên quan bằng văn bản tự nhiên.

Dự án này tập trung vào việc xây dựng và đánh giá một hệ thống VQA dựa trên các mô hình tiên tiến thuộc dòng Transformer đa phương thức. Cụ thể, nhóm nghiên cứu đã triển khai hai mô hình nổi bật:

- ViLT (Vision-and-Language Transformer)
- BEiT-3 (Bidirectional Encoder representation from Image Transformers)
- Zeroshot / Fewshot sử dụng Gemini model.

3. Dataset:

a. Crawl data:

Để xây dựng một bộ dữ liệu đa dạng và phù hợp cho bài toán Visual Question Answering (VQA), nhóm đã tiến hành thu thập ảnh và các thông tin liên quan từ nhiều nguồn mở. Việc đa dạng hóa nguồn ảnh giúp mô hình có khả năng tổng quát hóa tốt hơn trên nhiều tình huống thực tế khác nhau.

- **Quá trình thực hiện:**

1. Khởi tạo cơ sở dữ liệu (SQLite)

- Tạo CSDL SQLite chứa bảng `embeddings` với 3 cột:
    - `id`: khóa chính.
    - `url`: URL ảnh (duy nhất).
    - `embedding`: vector nhúng ảnh (dưới dạng BLOB).

- Nếu CSDL đã tồn tại, chỉ mở kết nối.

2. Sao chép dữ liệu đã có

- Sao chép ảnh và file CSDL từ thư mục cũ sang thư mục mới, dùng khi chạy tiếp sau lần đầu.

3. Lưu embedding ảnh vào DB

- Lưu vector embedding vào bảng `embeddings`.
- Nếu URL đã tồn tại, bỏ qua để tránh trùng lặp.

4. Kiểm tra ảnh có trùng không

- Duyệt từng embedding trong DB và so sánh cosine similarity với ảnh mới.
- Nếu vượt qua ngưỡng `threshold` thì ảnh được coi là trùng → không lưu.

5. Tạo embedding cho ảnh

- Dùng mô hình ViT (Vision Transformer) để sinh embedding cho ảnh đầu vào.
- Chuẩn hóa vector đầu ra.

6. Tải ảnh từ các nguồn khác nhau

Thông qua các hàm tải ảnh với hỗ trợ API key rotation, retry logic, và pagination state trả ra ảnh theo yêu cầu.

7. Xử lý 1 lô ảnh (batch)

Lần lượt xử lý ảnh theo các bước:

- Lặp qua danh sách URL ảnh.
- Tải ảnh tạm thời, sinh embedding.
- Kiểm tra trùng (bằng cosine similarity).
- Nếu không trùng: lưu ảnh và embedding.
- Nếu trùng: bỏ qua và xóa ảnh tạm.

8. Thực thi tự động theo ngày

- Quét từng danh mục (`categories`) và tải thêm ảnh nếu chưa đủ.
- Gọi `hàm ở mục 7` để xử lý và lưu.
- Có thể dùng cho Google, Pexels, Openverse, Pixabay.

9. Thống kê số lượng ảnh để kiểm tra số lượng

- Đếm số lượng ảnh theo từng danh mục.
- In tổng số ảnh và số danh mục có ảnh.

Tóm tắt quy trình chính:

1. Tạo DB nếu chưa có.
2. Duyệt từng danh mục ảnh.
3. Lấy URL ảnh từ nguồn API.
4. Tải ảnh về tạm thời.
5. Sinh embedding bằng ViT.
6. Kiểm tra trùng lặp bằng cosine similarity.
7. Nếu không trùng → lưu ảnh + embedding.
8. Lặp cho đến khi đủ số lượng ảnh mong muốn.

- **Các nguồn data:**
    - Google Image: Sử dụng truy vấn theo từ khóa để thu thập ảnh đại diện cho nhiều chủ đề khác nhau
    - Pixabay: Một thư viện ảnh miễn phí với giấy phép Creative Commons. Ảnh từ Pixabay có chất lượng cao và chủ đề đa dạng, phù hợp với bài toán VQA không cần xử lý bản quyền phức tạp.
    - Pexels: Nền tảng cung cấp ảnh và video miễn phí. Tương tự Pixabay, Pexels hỗ trợ truy vấn qua API và cung cấp metadata hữu ích.
    - Openverse: Một thư viện dữ liệu mã nguồn mở thuộc Creative Commons, giúp truy xuất ảnh có giấy phép rõ ràng. Dữ liệu từ đây rất hữu ích cho các nghiên cứu học thuật và mô hình mã nguồn mở.

b. VQA dataset:

Trong quá trình xây dựng hệ thống Visual Question Answering (VQA), nhóm đã đầu tư thời gian để thu thập và xử lý bộ dữ liệu hình ảnh phục vụ cho huấn luyện và đánh giá mô hình. Bên cạnh việc crawl ảnh tự động từ các nguồn ảnh công khai như Google Images, Pixabay, Pexels, và Openverse, nhóm cũng đặc biệt chú trọng đến việc tạo lập tập kiểm thử (test set) một cách cẩn thận và có kiểm soát.

Phân phối dữ liệu đồng đều giữa các tập:

- Các tập Train, Test và Eval có cấu trúc phân phối rất giống nhau ở cả ba khía cạnh: loại câu hỏi, loại câu trả lời, và nội dung câu hỏi.
- Điều này đảm bảo rằng mô hình huấn luyện sẽ không gặp hiện tượng data drift hoặc distribution mismatch khi đánh giá.

Tính đa dạng ở mức loại câu trả lời:

- Bộ dữ liệu bao gồm nhiều kiểu câu trả lời như: Yes/No, Numeric, Color, Position, Object, Other – giúp mô hình có cơ hội học nhiều dạng thông tin khác nhau.\
- Đặc biệt, việc có cả câu hỏi về vị trí và màu sắc cho phép mô hình tiếp cận với các khía cạnh không chỉ nhận dạng mà còn quan hệ không gian và thuộc tính.

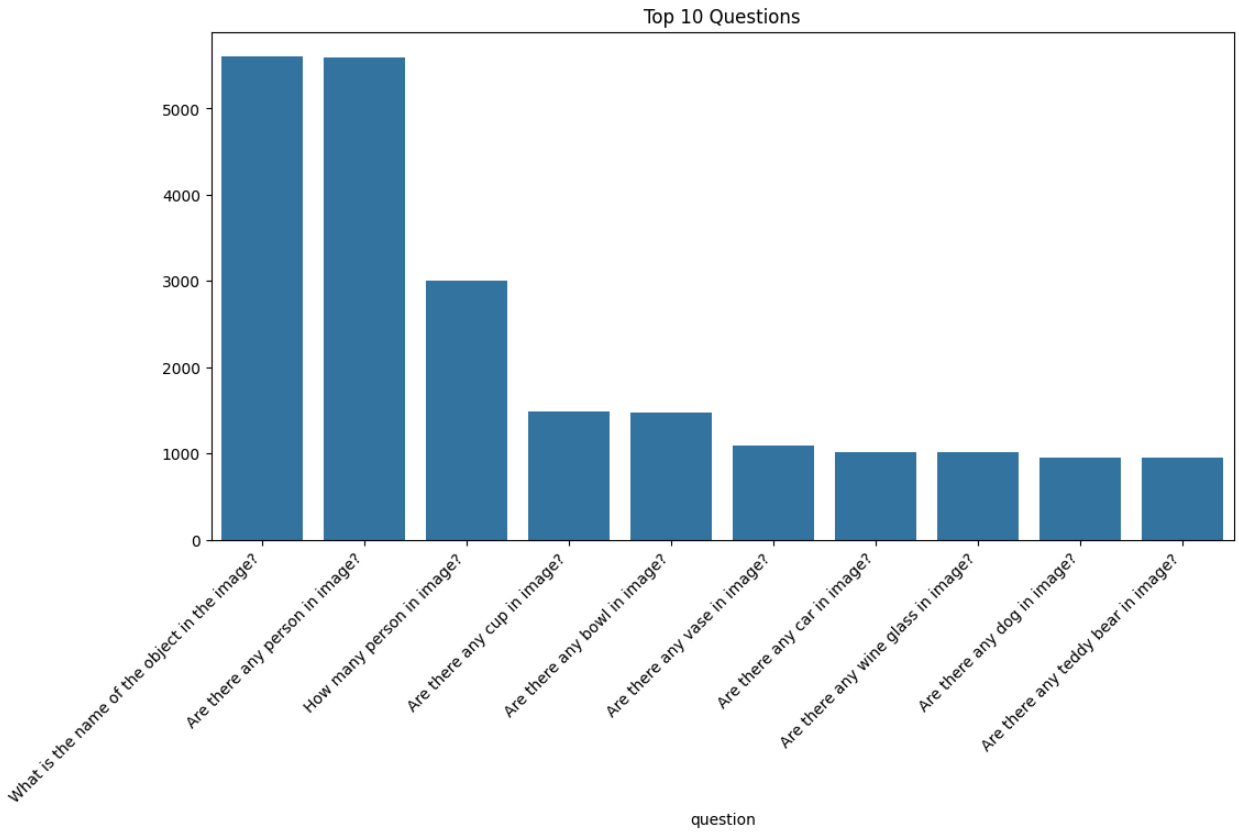
Quy mô tương đối lớn:

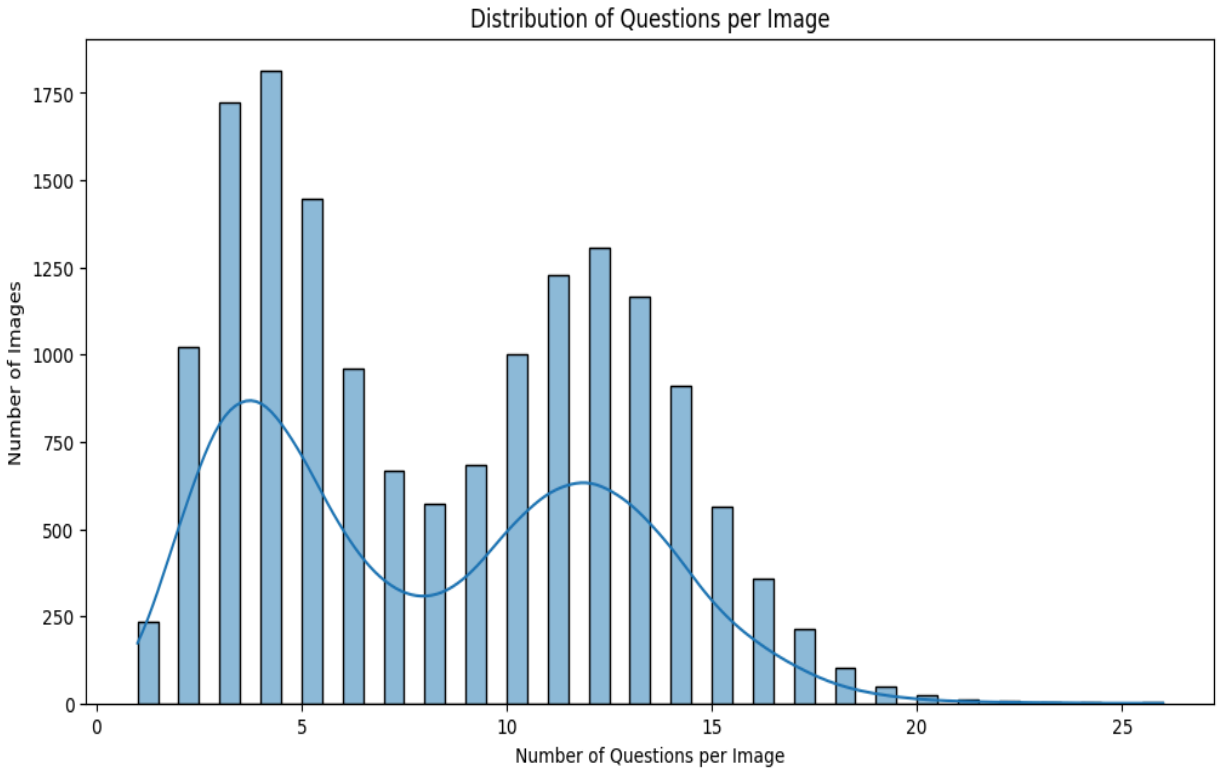
- Số lượng câu hỏi và đáp án đủ lớn trong tất cả các loại để mô hình học được xu hướng ngôn ngữ và hình ảnh phổ biến.

- Tập train:

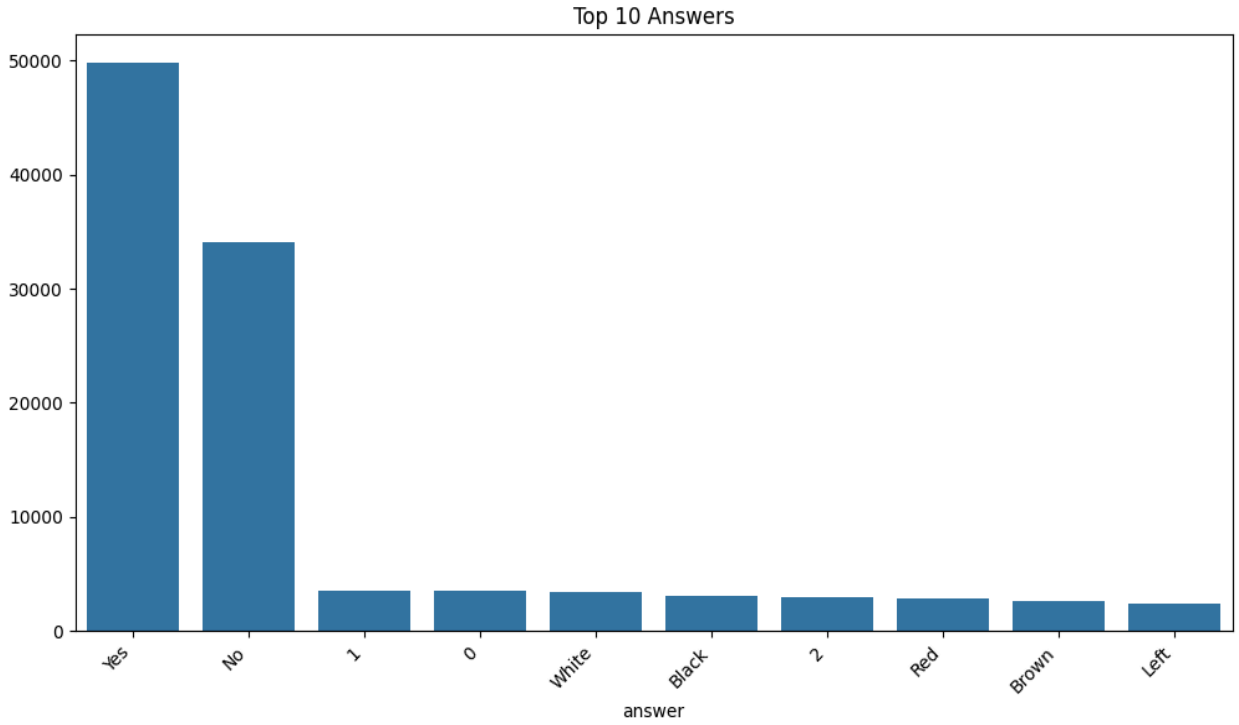
- Unique images: 16065
- Unique questions: 41840
- Unique answers: 106
- Total question: 131703
- Total answer: 131703

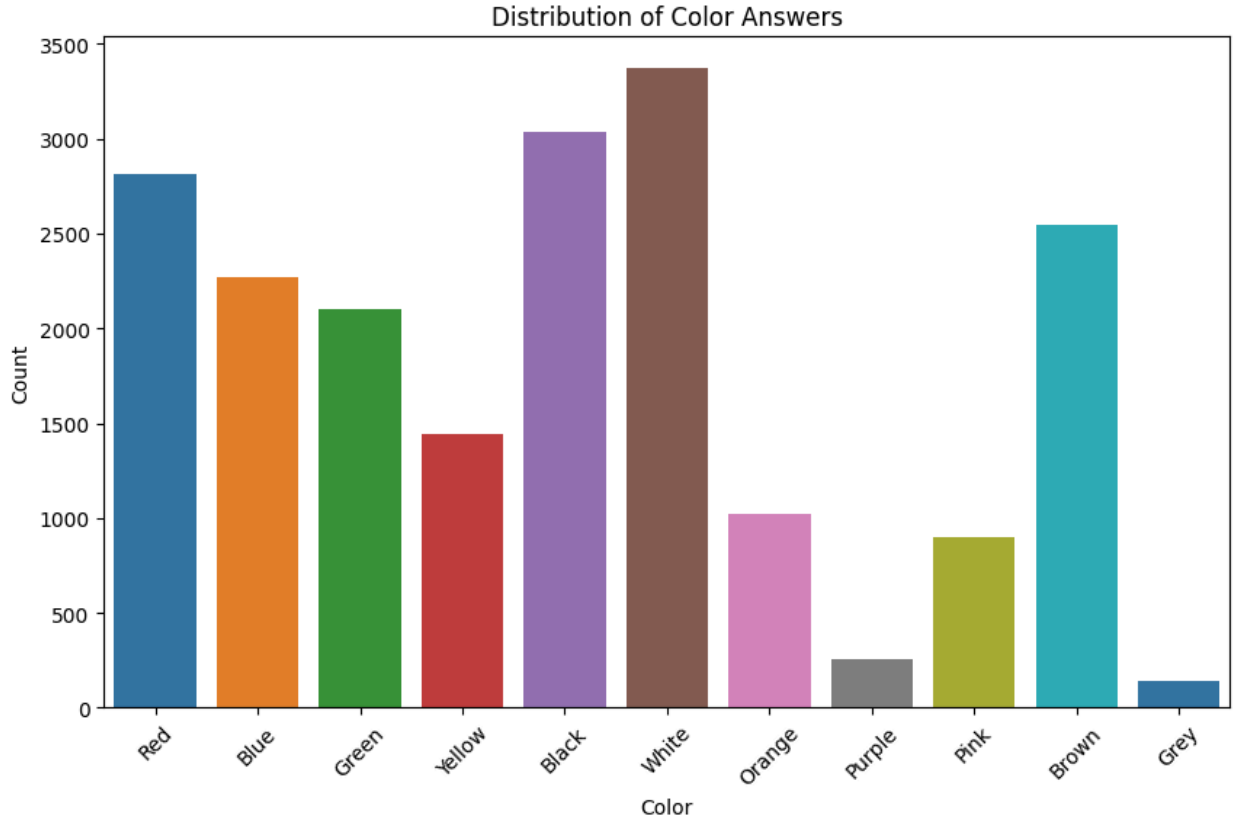
EDA for questions:

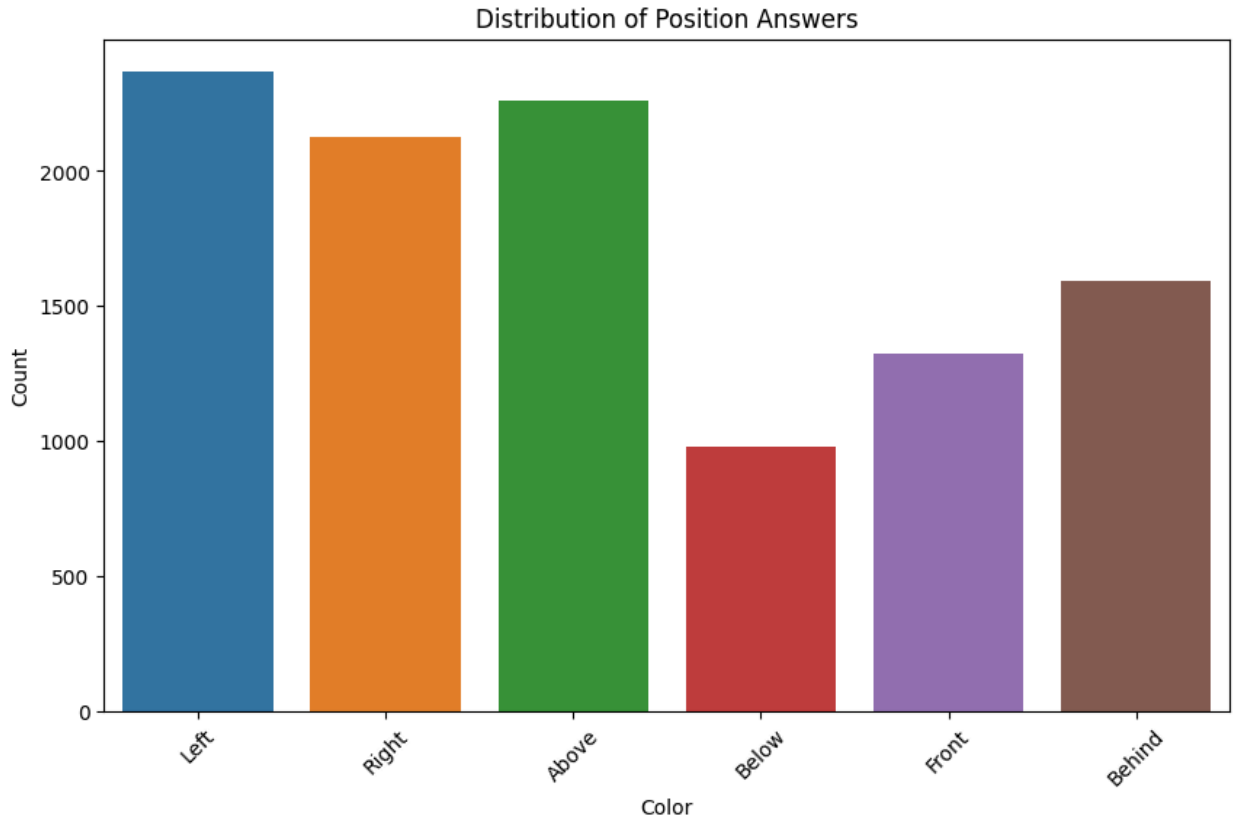


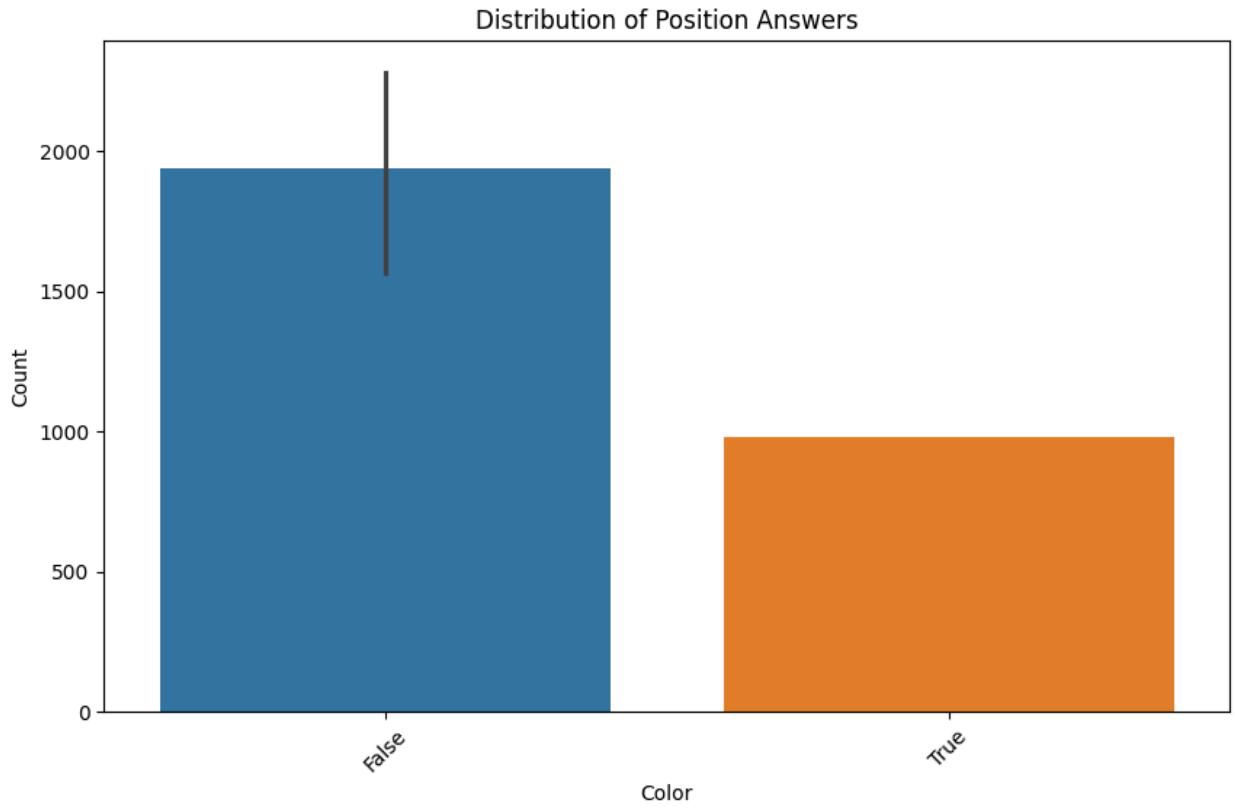


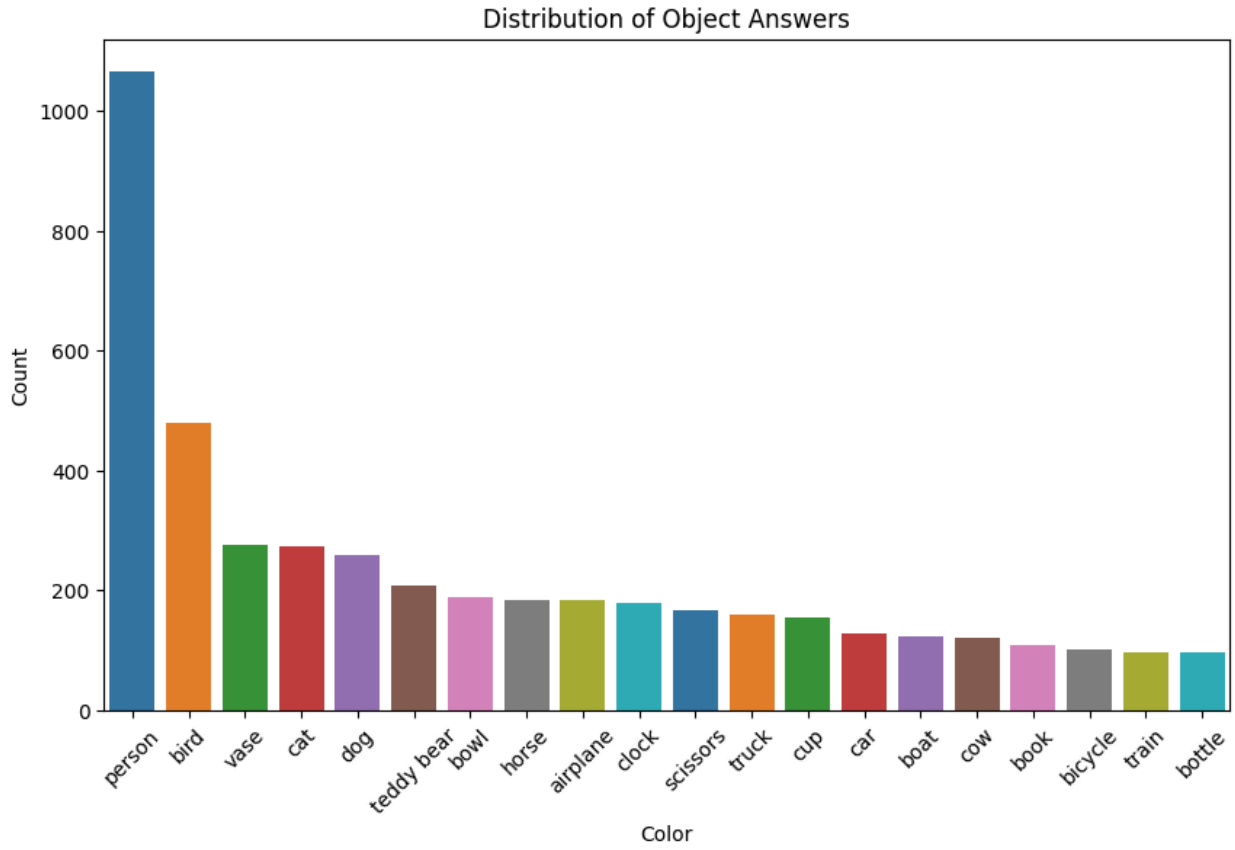
EDA for answers:

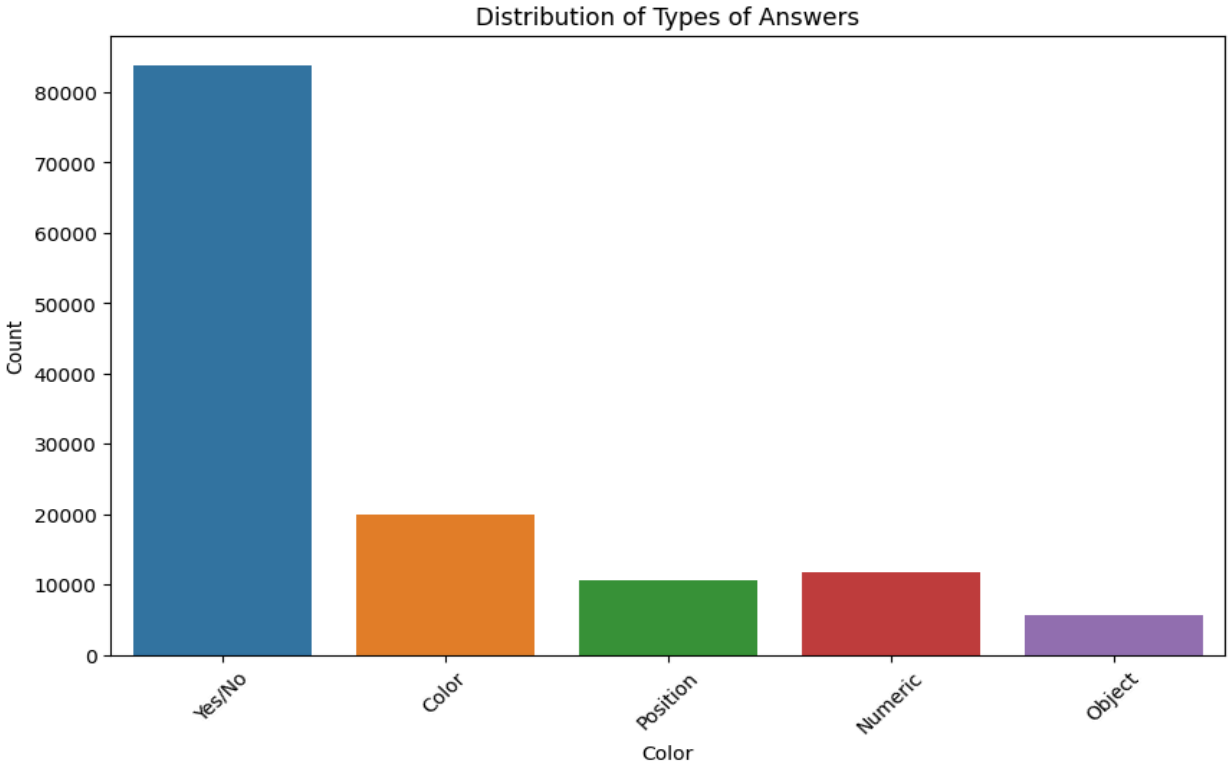

Top 10 Answers
50000
40000
30000
20000
10000
0
Yes
No
1
0
White
Black
2
Red
Brown
Left
answer

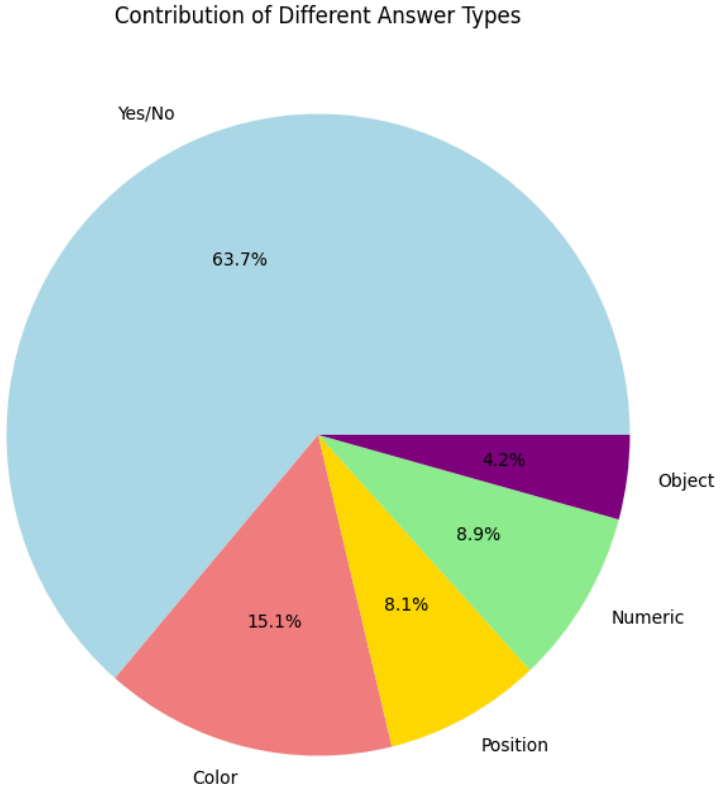

Distribution of Color Answers
3500
3000
2500
2000
1500
1000
500
0
Count
Red
Blue
Green
Yellow
Black
White
Orange
Purple
Pink
Brown
Grey
Color

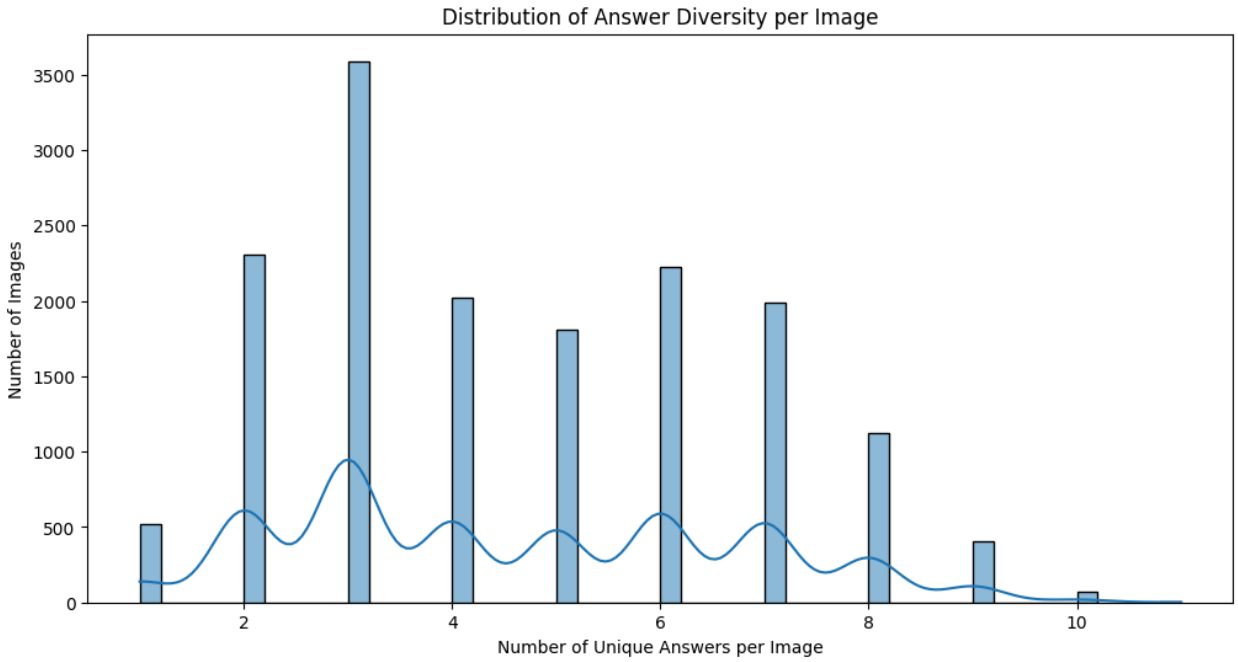

Distribution of Position Answers
Count
0
500
1000
1500
2000
Left
Right
Above
Below
Front
Behind
Color

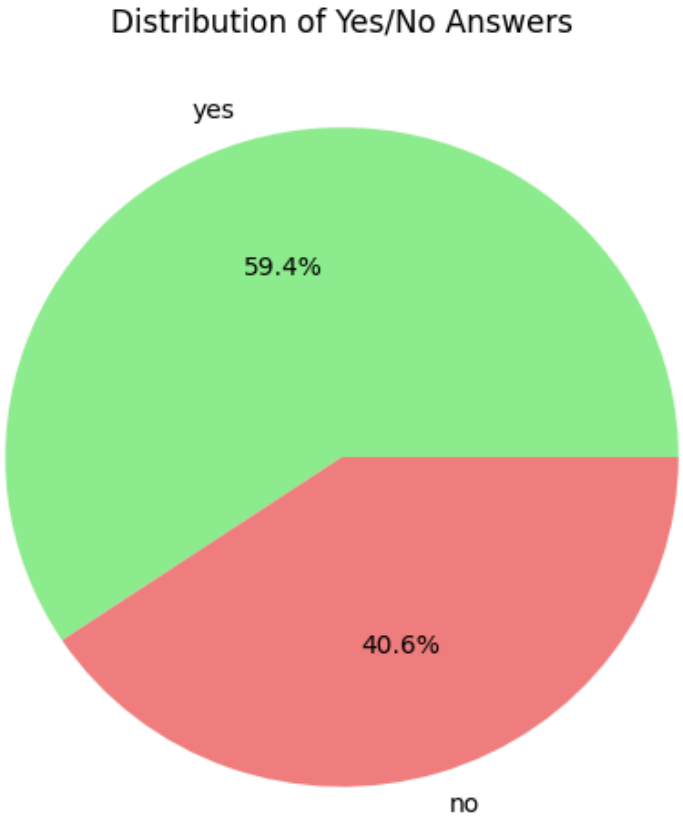

Distribution of Position Answers
Count
0
500
1000
1500
2000
False
True
Color


Distribution of Object Answers
Count
0
200
400
600
800
1000
person
bird
vase
cat
dog
teddy bear
bowl
horse
airplane
clock
scissors
truck
cup
car
boat
cow
book
bicycle
train
bottle
Color


Distribution of Types of Answers
Count
80000
70000
60000
50000
40000
30000
20000
10000
0
Yes/No
Color
Position
Numeric
Object
Color


Contribution of Different Answer Types
Yes/No
63.7%
4.2%
Object
8.9%
Numeric
8.1%
15.1%
Position
Color

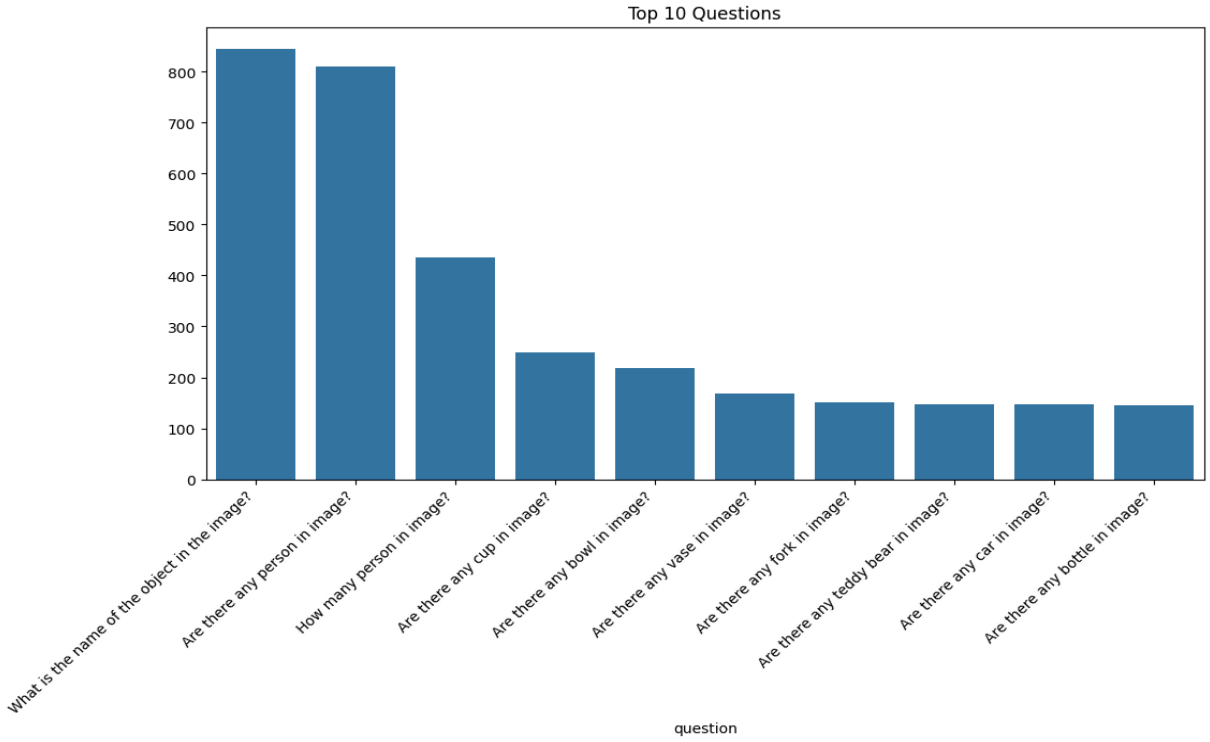


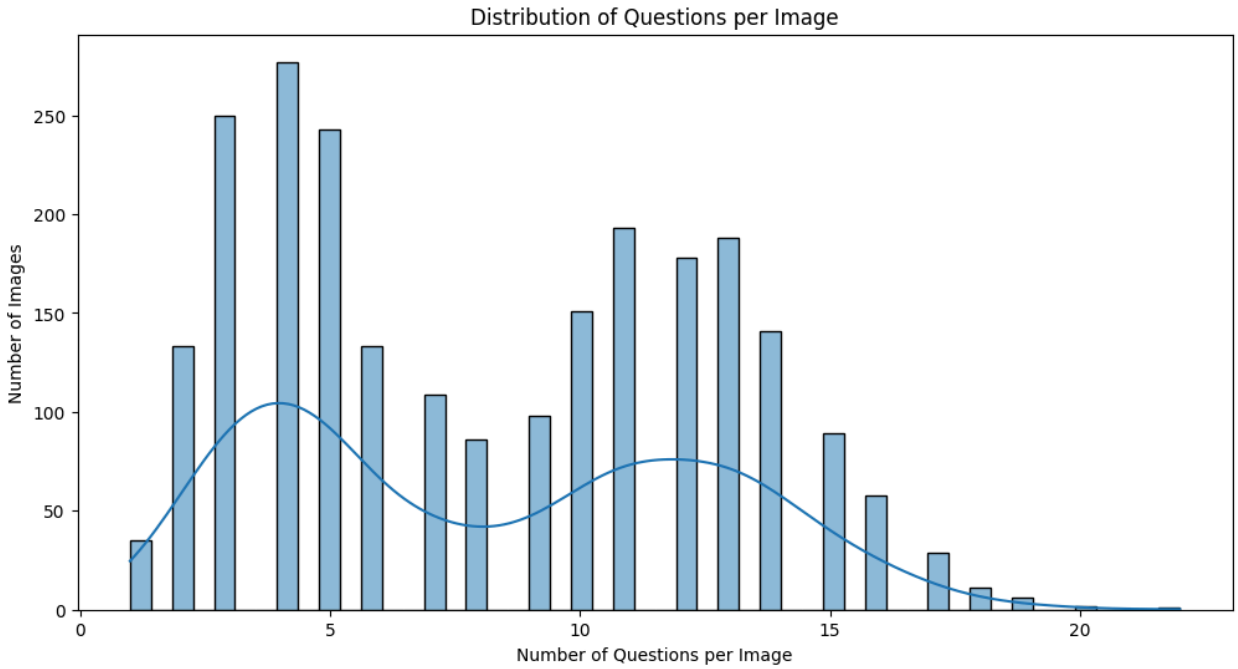


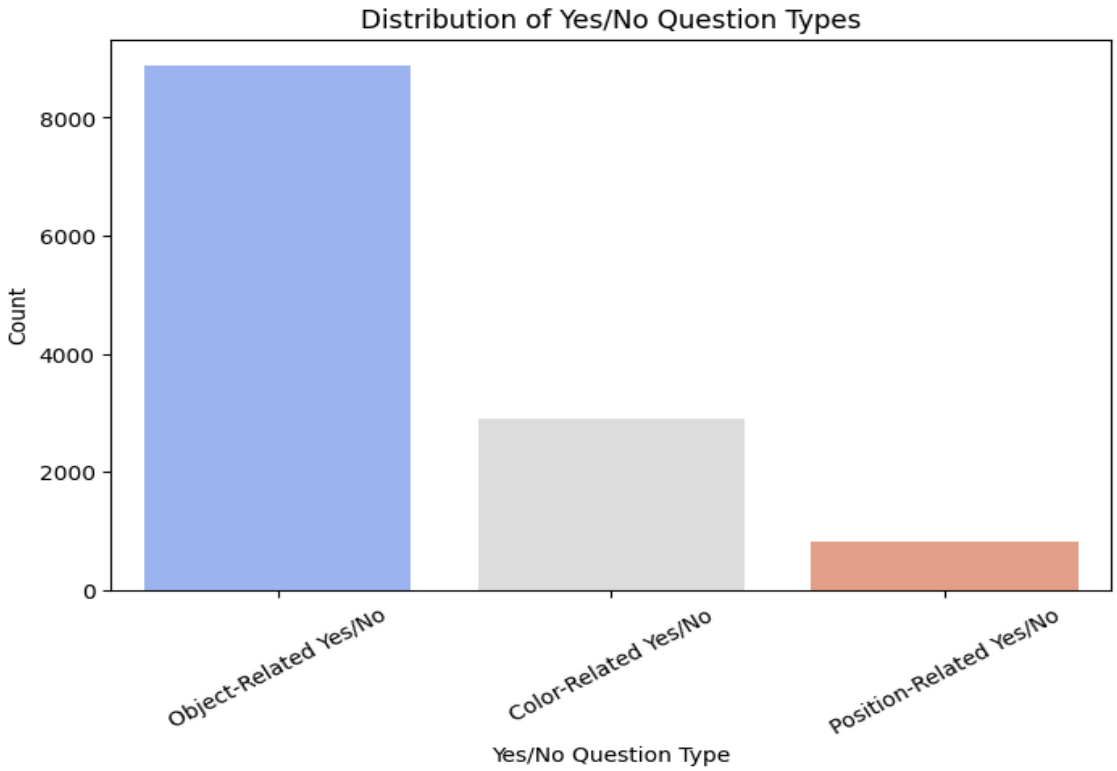
- **Tập val:**
  - Unique images: 2411

- Unique questions: 7863
- Unique answers: 82
- Total question: 19823
- Total answer: 19823

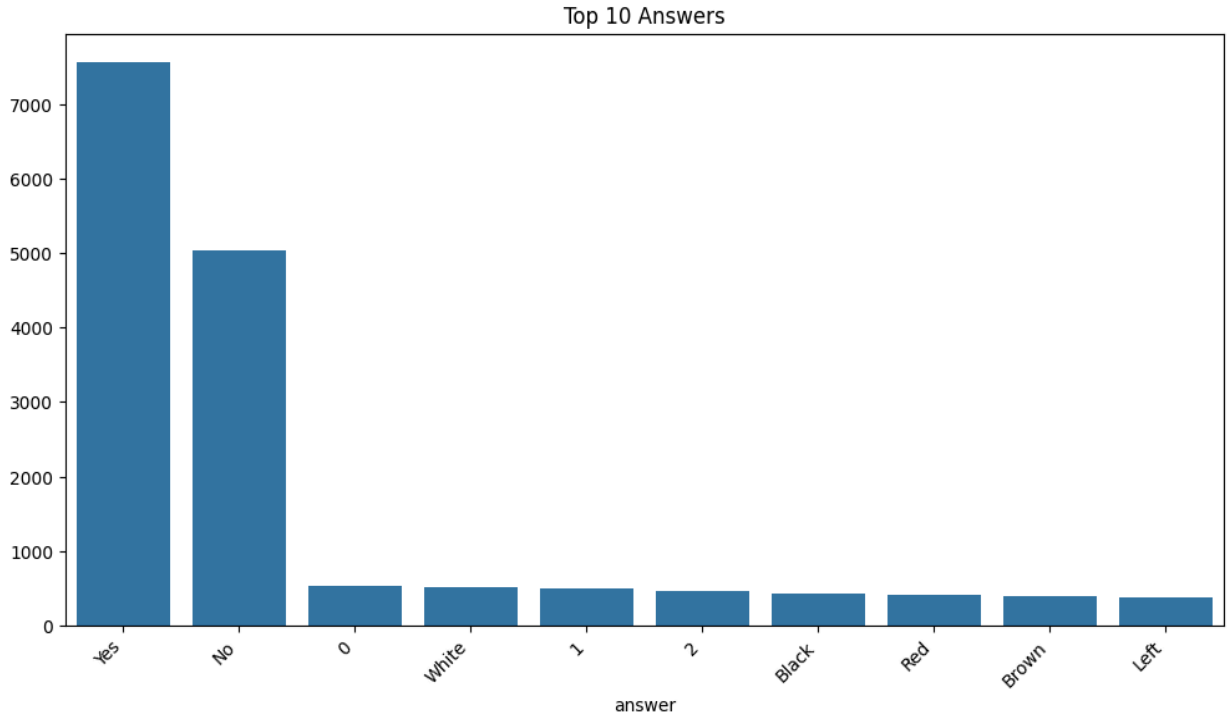
EDA for questions:

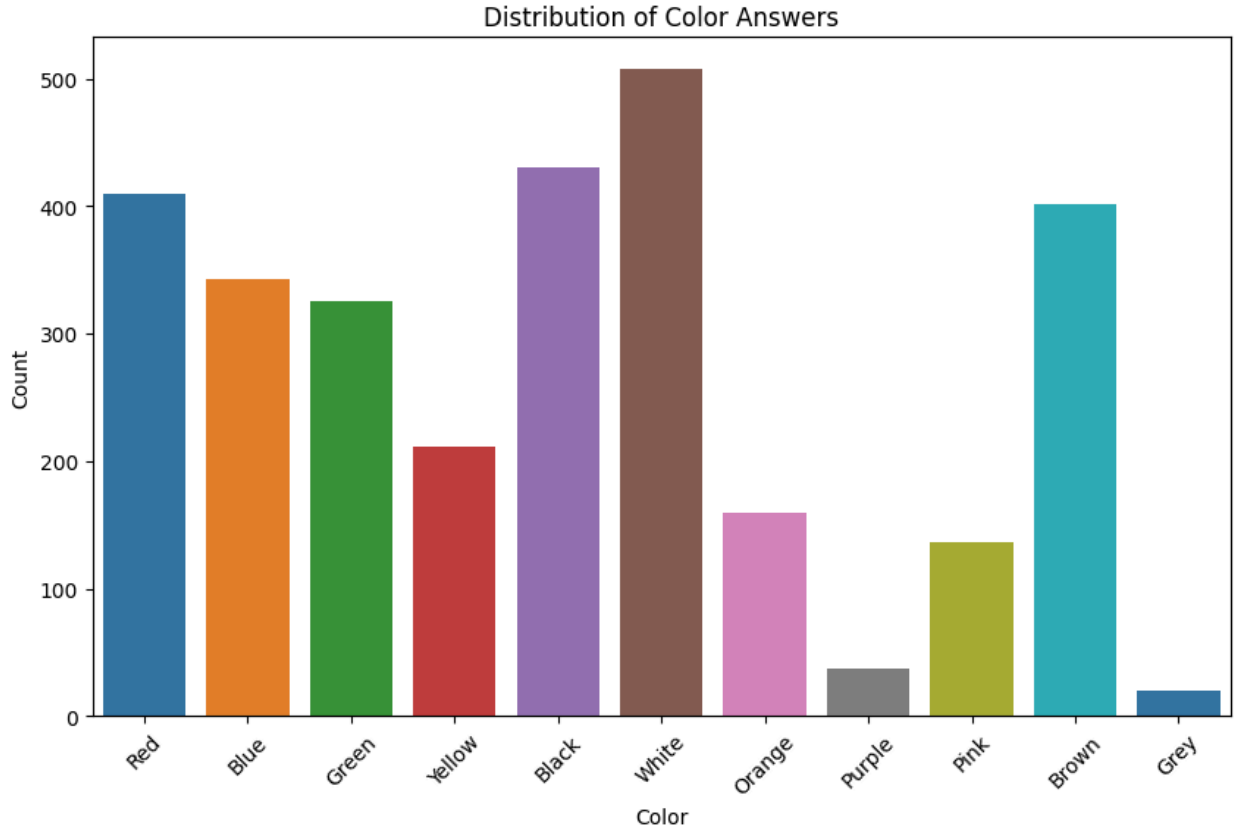


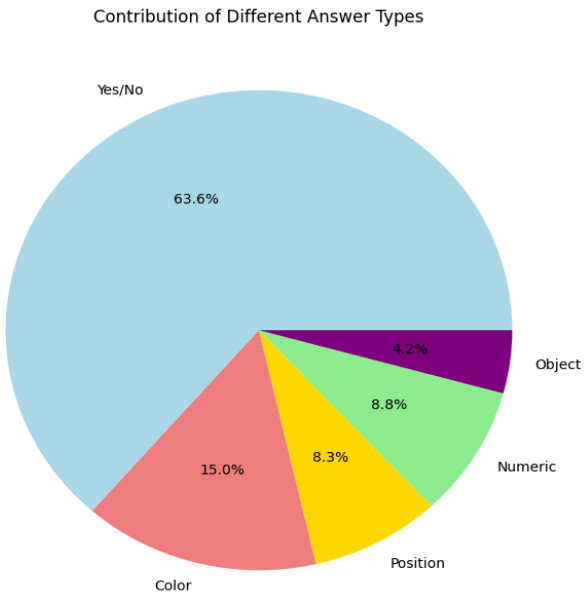


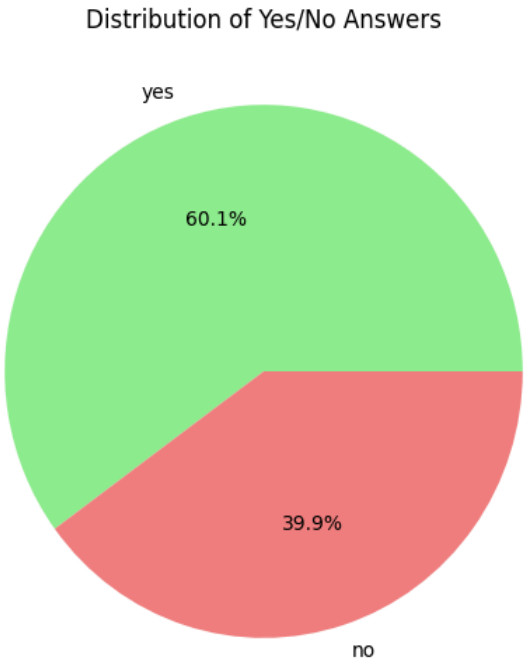


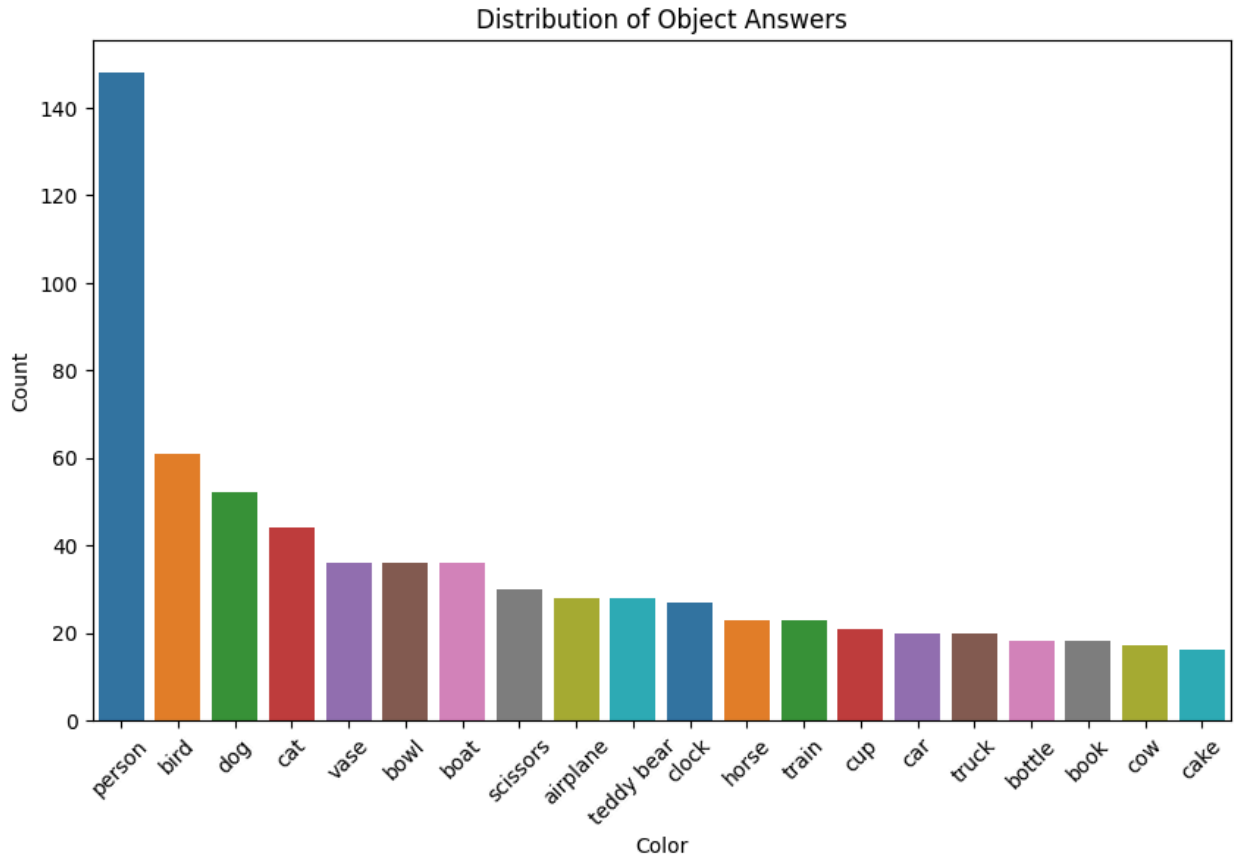
EDA for answers:


Top 10 Answers
7000
6000
5000
4000
3000
2000
1000
0
Yes
No
0
White
1
2
Black
Red
Brown
Left
answer


Distribution of Color Answers
500
400
300
200
100
0
Count
Red
Blue
Green
Yellow
Black
White
Orange
Purple
Pink
Brown
Grey
Color


Contribution of Different Answer Types
Yes/No
63.6%
4.2%
Object
8.8%
Numeric
8.3%
15.0%
Position
Color

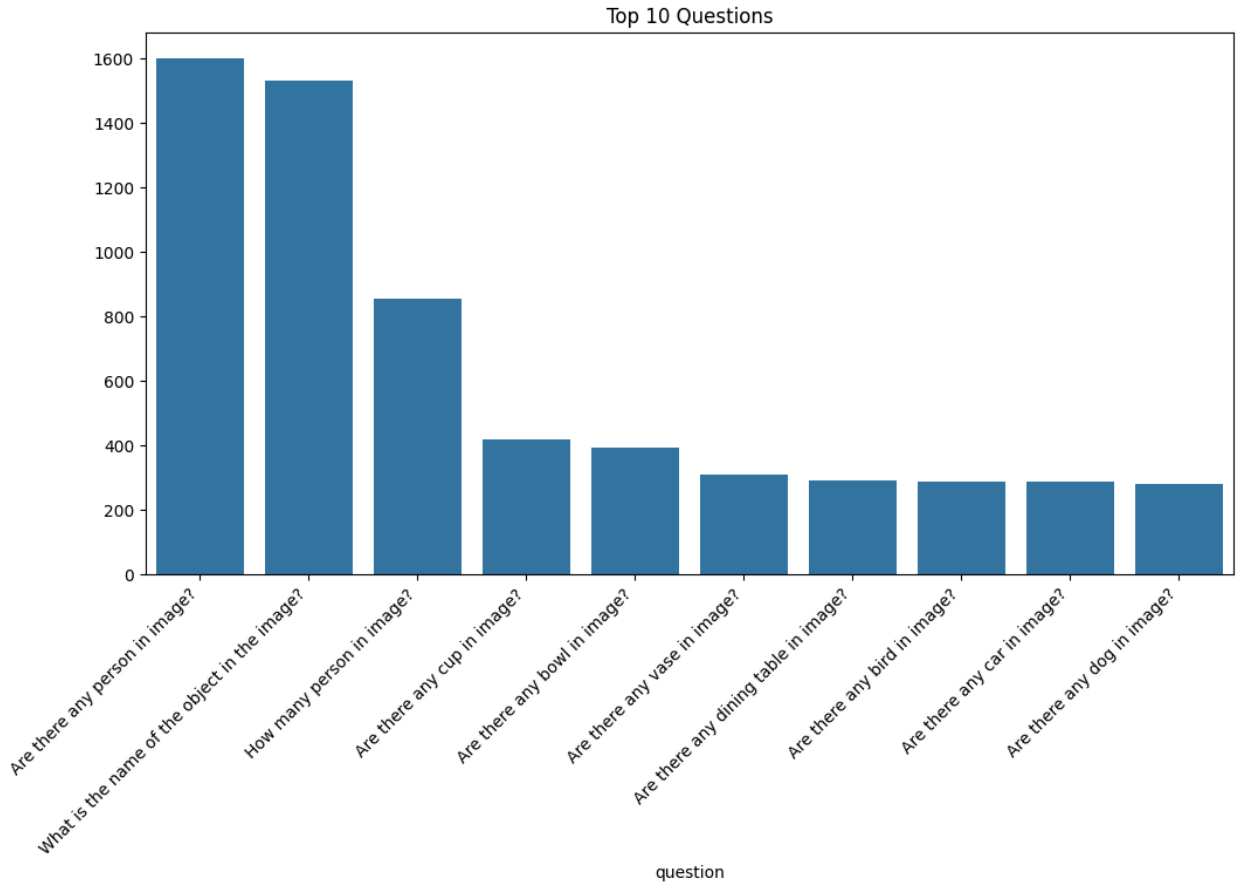

Distribution of Yes/No Answers
yes
60.1%
39.9%
no

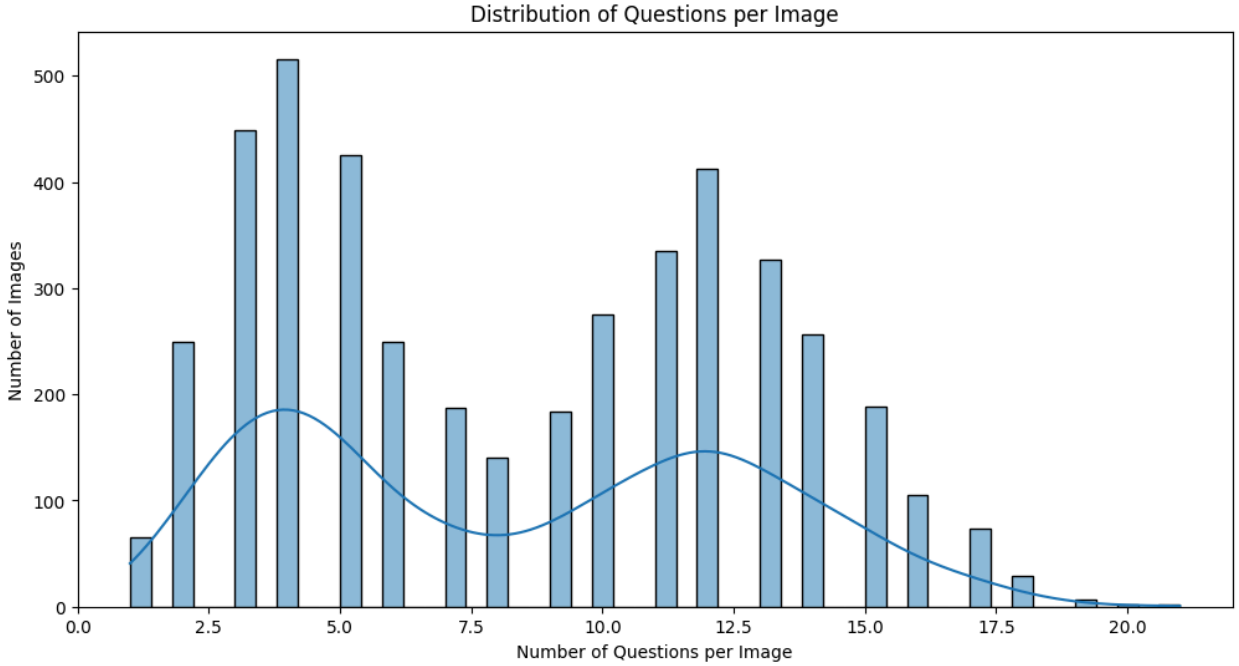


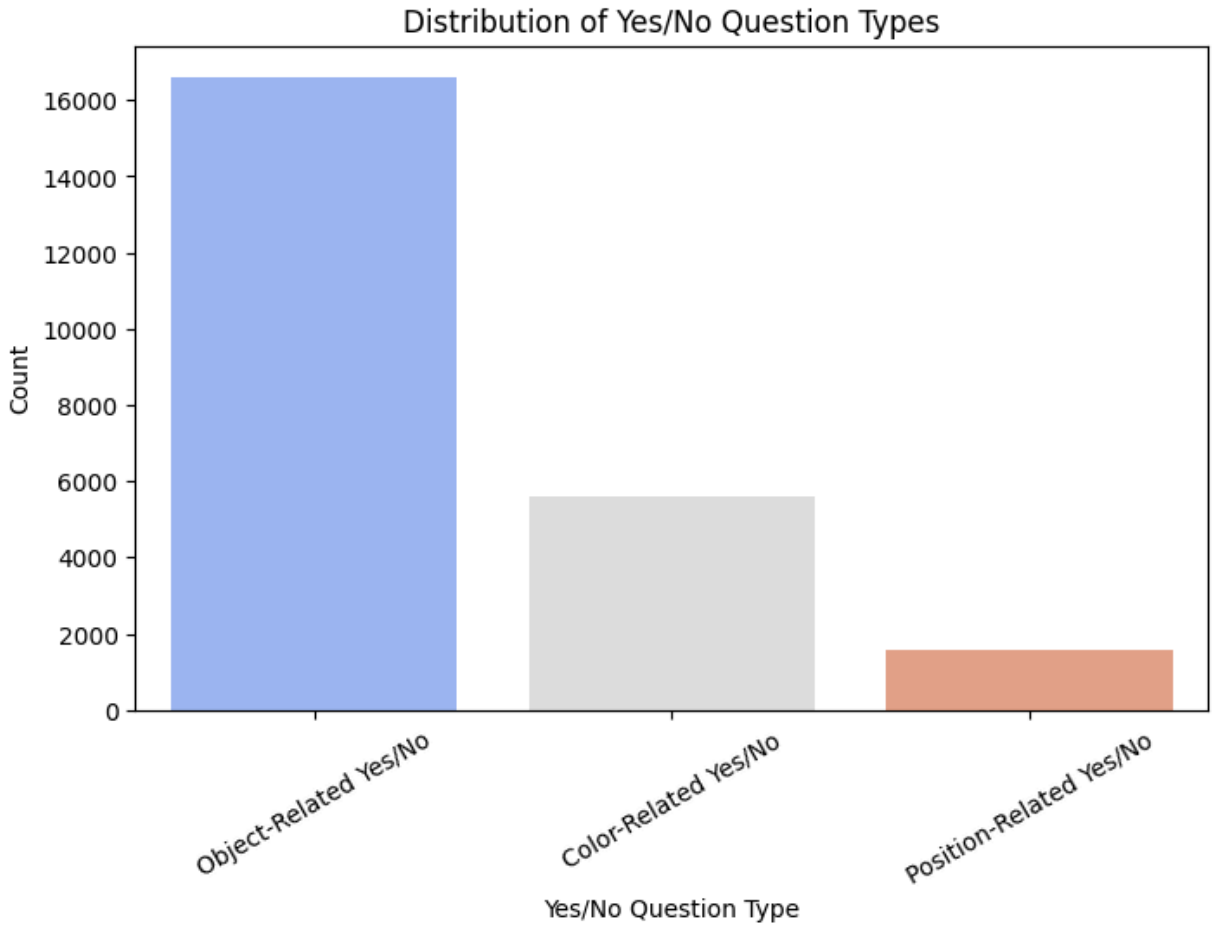
- ## Tập test:
    - Unique images: 4475
    - Unique questions: 13984
    - Unique answers: 92
    - Total question: 37371
    - Total answer: 37371

    Tập test này đã được nhóm tự kiểm tra chi tiết.

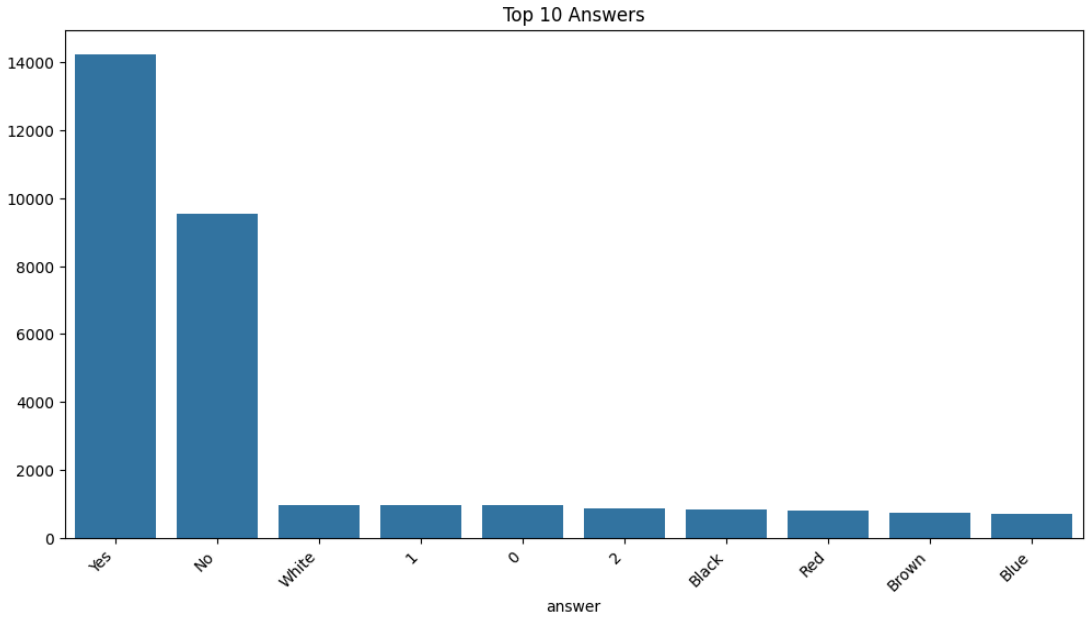
    EDA for questions:

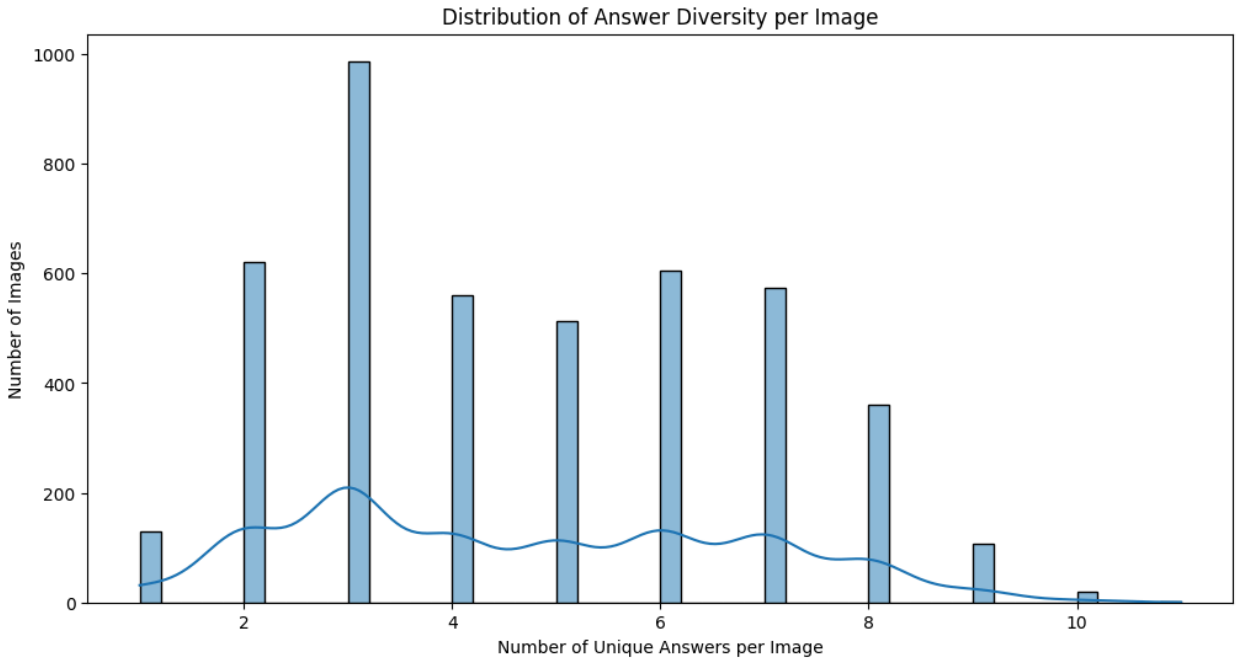

Top 10 Questions
1600
1400
1200
1000
800
600
400
200
0
Are there any person in image?
What is the name of the object in the image?
How many person in image?
Are there any cup in image?
Are there any bowl in image?
Are there any vase in image?
Are there any dining table in image?
Are there any bird in image?
Are there any car in image?
Are there any dog in image?
question

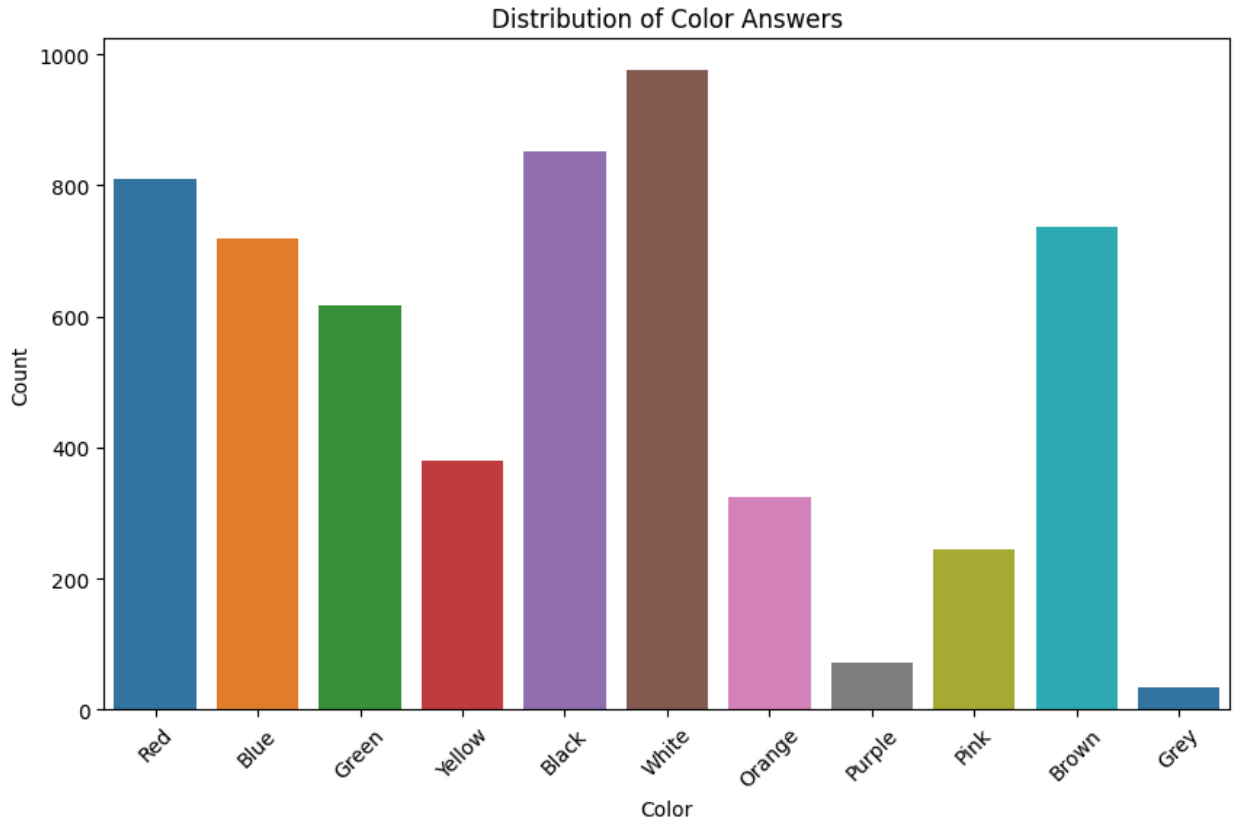

Distribution of Questions per Image
Number of Images
500
400
300
200
100
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Number of Questions per Image

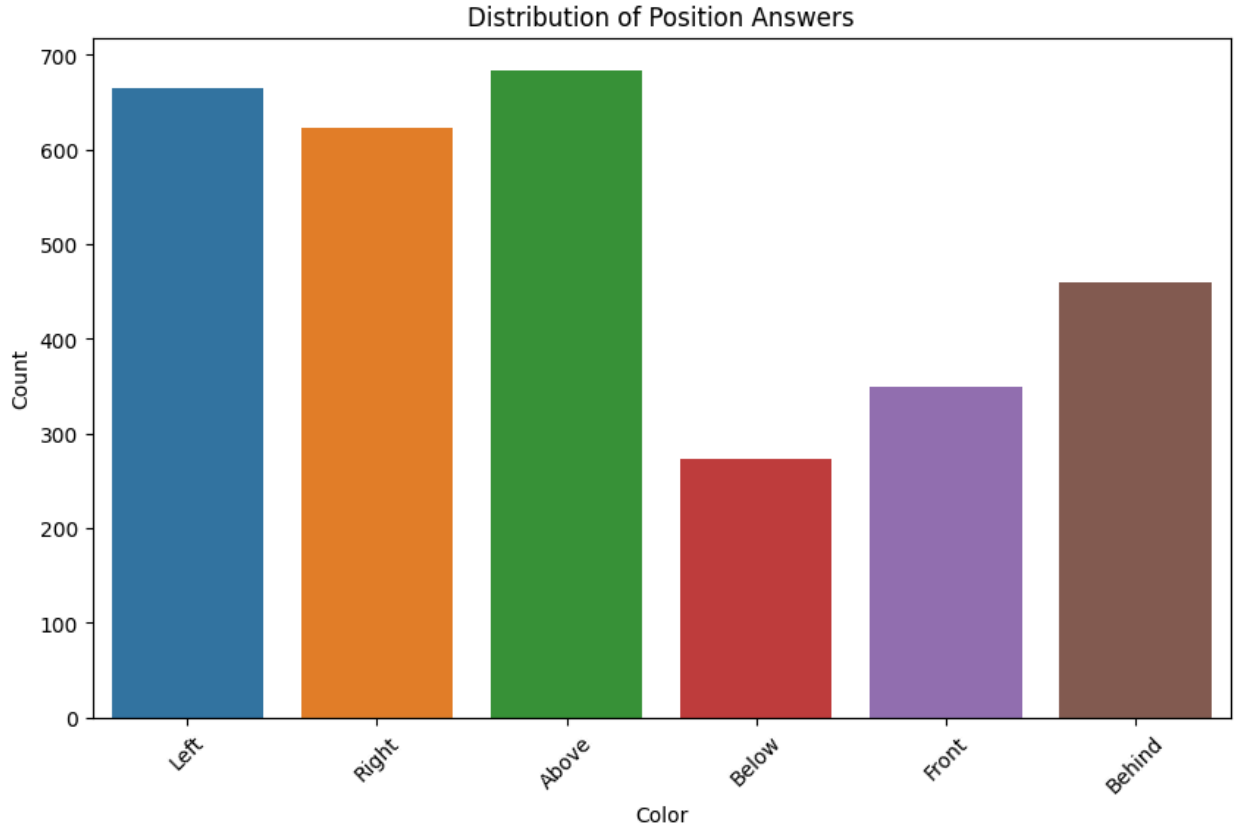


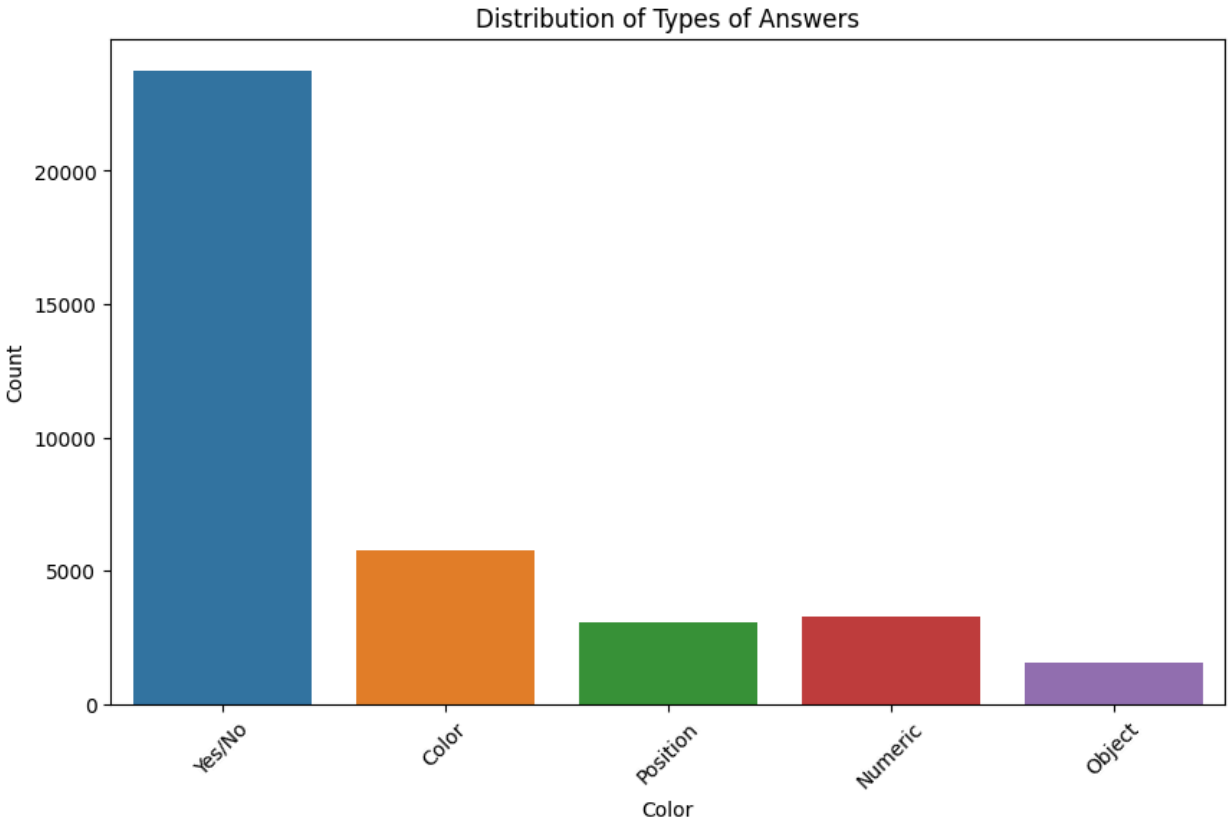
EDA for answers:

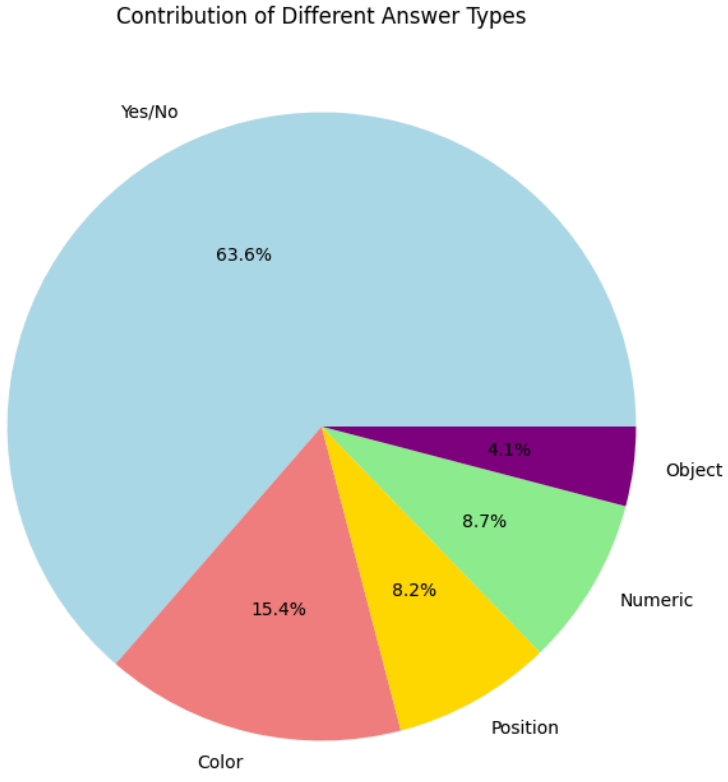

Top 10 Answers
14000
12000
10000
8000
6000
4000
2000
0
Yes
No
White
1
0
2
Black
Red
Brown
Blue
answer

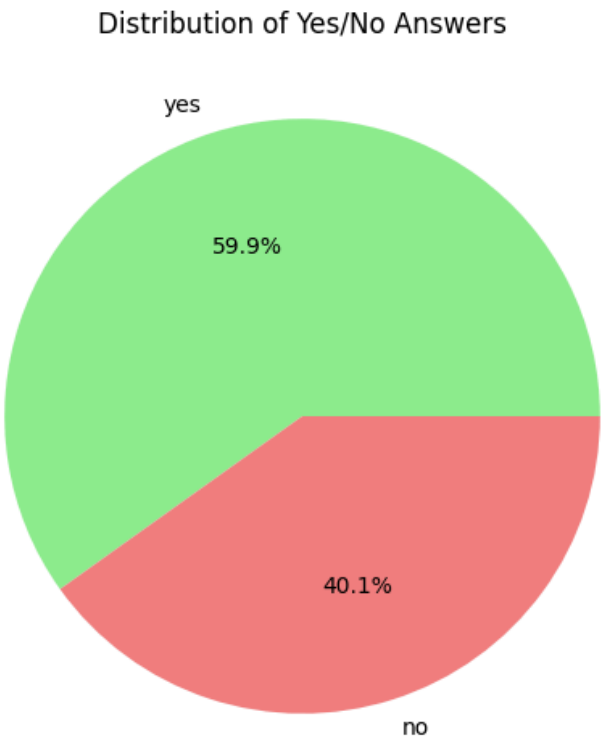

Distribution of Answer Diversity per Image
1000
800
600
400
200
0
Number of Images
2
4
6
8
10
Number of Unique Answers per Image

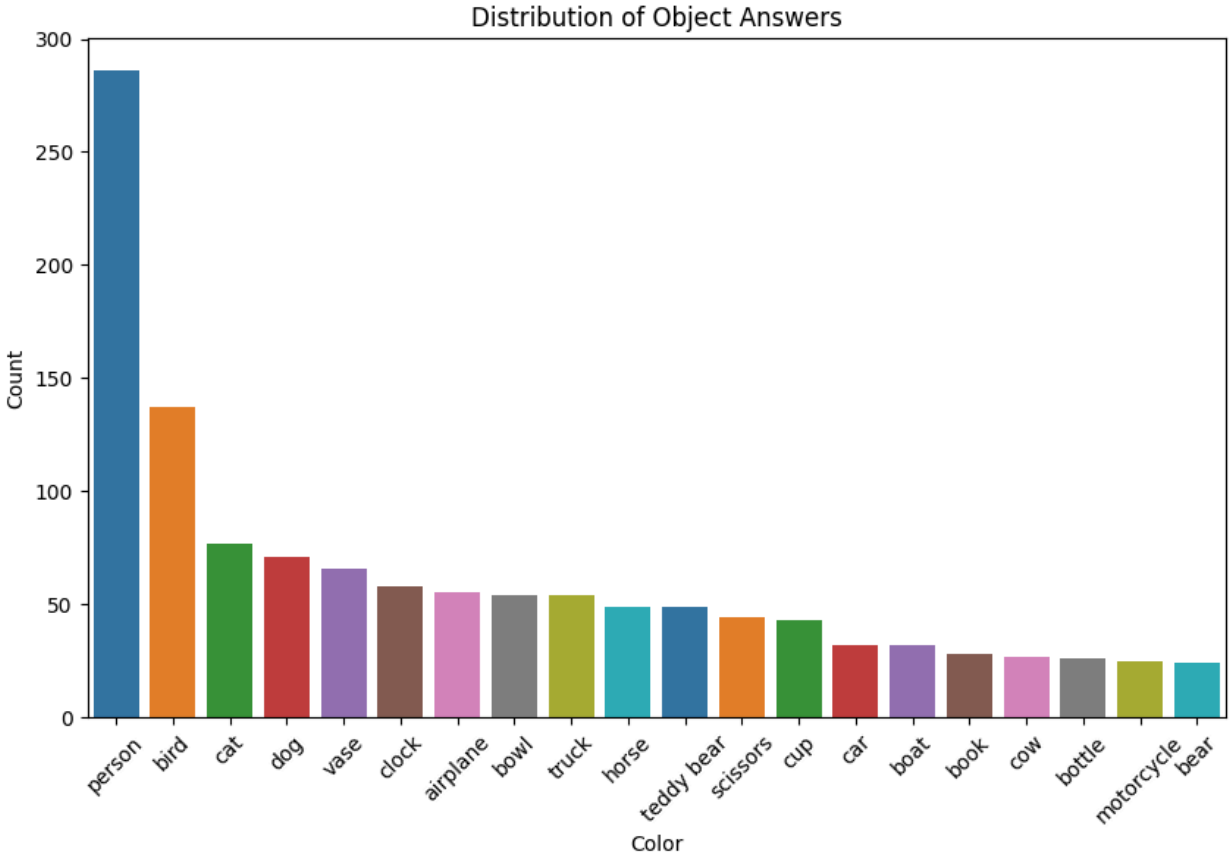

Distribution of Color Answers
Count
Color
0
200
400
600
800
1000
Red
Blue
Green
Yellow
Black
White
Orange
Purple
Pink
Brown
Grey

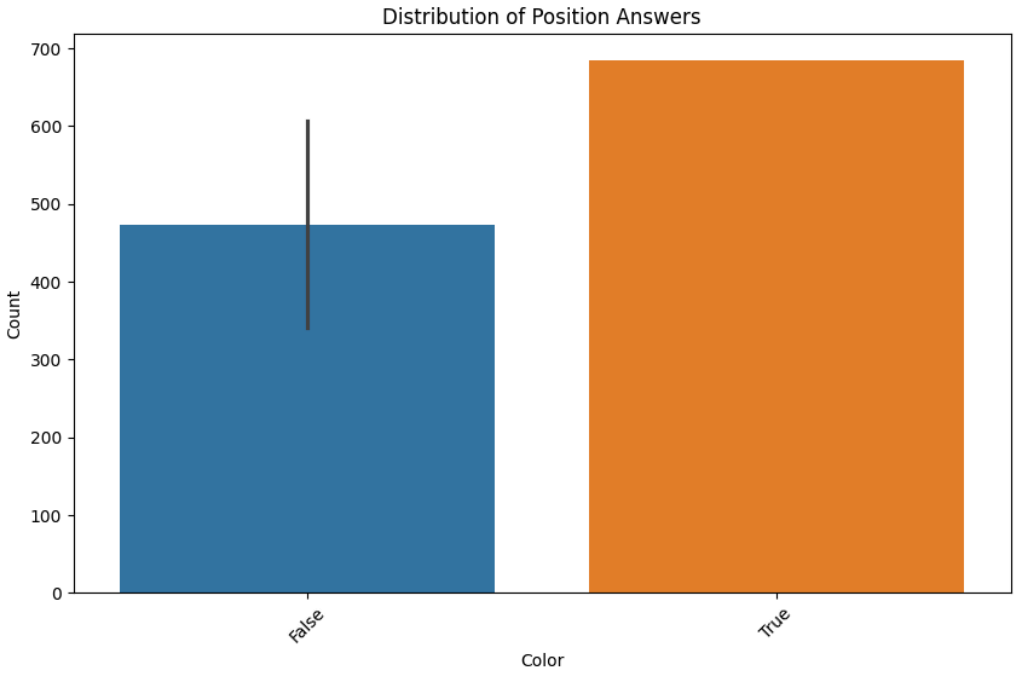

Distribution of Position Answers
Count
Color
0
100
200
300
400
500
600
700
Left
Right
Above
Below
Front
Behind


Distribution of Types of Answers
Count
20000
15000
10000
5000
0
Yes/No
Color
Position
Numeric
Object
Color


Contribution of Different Answer Types
Yes/No
63.6%
4.1%
Object
8.7%
Numeric
8.2%
15.4%
Position
Color


Distribution of Yes/No Answers
yes
59.9%
40.1%
no


Distribution of Object Answers
Count
Color
0
50
100
150
200
250
300
person
bird
cat
dog
vase
clock
airplane
bowl
truck
horse
teddy bear
scissors
cup
car
boat
book
cow
bottle
motorcycle
bear



4. Models:

a. ViLT:

Trong project này, nhóm lựa chọn ViLT (Vision-and-Language Transformer) làm mô hình baseline đầu tiên cho bài toán Visual Question Answering (VQA), bởi vì ViLT có thiết kế nhẹ, không sử dụng CNN để trích xuất đặc trưng ảnh như các mô hình truyền thống mà thực hiện fusion trực tiếp giữa ảnh và câu hỏi bằng Transformer.

- ***Tổng quan:*** ViLT được đề xuất trong bài báo "ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision" (Kim et al., 2021), với các điểm nổi bật:
    - Không sử dụng backbone trích xuất đặc trưng ảnh như ResNet hay ViT. Thay vào đó, ViLT chuyển ảnh thành chuỗi patch embeddings trực tiếp bằng Linear Projection, sau đó kết hợp với token văn bản qua một Transformer duy nhất.
    - Token hóa hình ảnh: Ảnh đầu vào được chia thành các patch nhỏ (ví dụ 32x32), mỗi patch được ánh xạ sang vector bằng một lớp tuyến tính, tương tự như trong ViT.
    - Token hóa văn bản: Câu hỏi được mã hóa bằng tokenizer giống như trong BERT.
    - Các token ảnh và token câu hỏi sau đó được nối chuỗi (concatenate) lại với nhau và đưa vào một Transformer encoder, học mối quan hệ giữa nội dung hình ảnh và ngôn ngữ.
- ***Ưu điểm:***
    - Hiệu quả tính toán cao: Do không dùng CNN hoặc ViT pre-trained cho ảnh, ViLT chạy nhanh hơn các mô hình kết hợp 2 nhánh (dual-stream).
    - End-to-end: ViLT huấn luyện hoàn toàn đầu-cuối, không cần vùng đặc trưng (region-based) như trong LXMERT hay ViLBERT.
- ***Cách nhóm sử dụng:***

- Load pretrained model từ hugging face, thay số lớp ở head layer bằng số lớp trong dữ liệu.
- Tạo 1 bộ mapping cho answer.
- Sử dụng CrossEntroy thay cho BinaryCrossEntropy vì dữ liệu là single class.
- Huấn luyện mô hình và giám sát với Wandb.
- Lưu best model dựa trên loss của tập validation.

b. Beit-3:

Mô hình BEiT-3 (Bidirectional Encoder representation from Image Transformers - version 3) là một mô hình đa phương thức mạnh mẽ, được thiết kế để xử lý đồng thời cả văn bản và hình ảnh trong một kiến trúc thống nhất.

- ***Tổng quan:***
  - Unified Transformer: BEiT-3 sử dụng cùng một Transformer backbone để xử lý cả văn bản và hình ảnh, không tách biệt 2 nhánh riêng cho từng modality như nhiều mô hình cũ.
  - Học đa nhiệm (Multi-task Pretraining): Mô hình được huấn luyện trước trên nhiều tác vụ khác nhau.
  - Token hóa ảnh: Ảnh đầu vào được mã hóa thành chuỗi các patch embeddings bằng ViT-like encoder, sử dụng tokenizer thị giác đặc biệt gọi là dALL-E tokenizer để chuyển ảnh sang các mã số discrete.
  - Token hóa văn bản: Văn bản được mã hóa bằng BERT-style tokenizer như thường lệ.
  - Các token từ cả hai nguồn (image + text) sau đó được nối chuỗi và đưa vào chung một encoder, giúp mô hình hiểu được sự tương tác giữa hình ảnh và văn bản một cách hiệu quả.
- ***Ưu điểm:***
  - Hiệu suất vượt trội: BEiT-3 đạt SOTA (State of the Art) trên nhiều benchmark đa phương thức.
  - Tổng quát hóa tốt: Nhờ huấn luyện trước với chiến lược đa nhiệm, BEiT-3 có khả năng tổng quát mạnh mẽ, giúp cải thiện hiệu suất fine-tune trên tập dữ liệu mới.
  - Khả năng xử lý ảnh và văn bản mạnh: Việc chia sẻ backbone duy nhất cho cả 2 loại dữ liệu giúp tối ưu mối quan hệ giữa ảnh và câu hỏi.
- ***Cách nhóm sử dụng:***
  - Clone source code của BEiT-3 trên github.
  - Thay đổi cách load data để phù hợp với dữ liệu của nhóm.
  - Load pretrained model từ checkpoint đã được finetune trên tập VQAv2, thay số lớp ở head layer bằng số lớp trong dữ liệu.
  - Sử dụng CrossEntroy thay cho BinaryCrossEntropy vì dữ liệu là single class.
  - Huấn luyện mô hình và giám sát với Wandb.
  - Lưu best model dựa trên loss của tập validation.

c. Zero-shot/Few-shot với Gemini:

- **Zero-shot:** là phương pháp cho phép mô hình thực hiện một nhiệm vụ chưa từng thấy trước đó, mà không cần được huấn luyện trực tiếp trên tập dữ liệu của nhiệm vụ đó. Mô

hình được kỳ vọng tận dụng hiểu biết tổng quát đã học từ nhiều nhiệm vụ khác nhau (ví dụ: qua pre-training trên tập lớn) để suy luận trong tình huống mới.

- **Few-shot:** là phương pháp cho phép mô hình học từ một số lượng nhỏ ví dụ mẫu (ví dụ: 10–100) trước khi thực hiện tác vụ. Mục tiêu là tận dụng khả năng tổng quát hóa đã được học từ pre-training, đồng thời "định hướng" mô hình với một vài ví dụ phù hợp với tác vụ cụ thể.
- **Cài đặt:** Trong dự án này, nhóm đã sử dụng mô hình Gemini (phiên bản gemini-2.0-flash) từ Google để thực hiện bài toán Visual Question Answering (VQA) dưới hai chế độ. Mô hình được cấu hình sử dụng phiên bản "gemini-2.0-flash" – một mô hình đa phương thức hỗ trợ xử lý cả văn bản lẫn hình ảnh.
    - Zero-shot: đưa trực tiếp ảnh và câu hỏi sau đó nhận câu trả lời.
    - Few-shot:
        - Mỗi ví dụ gồm: Ảnh, câu hỏi, câu trả lời.
        - Sau khi thêm 10 ví dụ, ảnh người dùng + câu hỏi mới sẽ được nối vào cuối prompt.
        - Mô hình gemini-2.0-flash được gọi với prompt dạng hỗn hợp: ảnh + văn bản.

5. Demo (Implementation):

Nhóm xây dựng ứng dụng đơn giản để tương tác trực quan với các mô hình VQA (Visual Question Answering), cho phép người dùng:

- Tải lên ảnh tùy chọn.
- Đặt câu hỏi liên quan đến ảnh.
- Nhận câu trả lời từ các mô hình VQA như ViLT, BEiT-3 và Gemini (Google AI).

- **Cấu trúc tổng quát của ứng dụng:**
    - Ứng dụng được phát triển bằng Streamlit — một framework đơn giản cho việc xây dựng giao diện web tương tác trong Python. Giao diện gồm các thành phần chính:
        - Thanh bên để chọn mô hình
        - Vùng tải ảnh
        - Ô nhập câu hỏi
        - Phần hiển thị kết quả trả lời
    - Ứng dụng được tích hợp các mô hình mà nhóm đã nêu trên để thực hiện tác vụ VQA.
- **Luồng hoạt động trong ứng dụng:**
    - Người dùng tải ảnh ➜ ảnh được hiển thị.
    - Nhập câu hỏi liên quan đến ảnh.
    - Chọn mô hình xử lý từ sidebar.
    - Gọi hàm get_format_response() để xử lý ảnh + câu hỏi tùy theo mô hình.
    - Trả về kết quả dưới dạng câu trả lời hiển thị trên giao diện.

6. The results:

Bảng Thông tin kết quả các model:

Type	Model	Accuracy	Precision	Recall	F1-score

Fine-tune	ViLT	0.690	0.668	0.690	0.669
Fine-tune	**Beit-3**	**0.824**	**0.801**	**0.824**	**0.807**
Zeroshot	Gemini 2.0 flash	0.66	0.66	0.66	0.666
Fewshot	Gemini 2.0 flash	0.76	0.76	0.76	0.76

7. Conclusion:

- BEiT-3 cho kết quả vượt trội hơn rõ rệt so với ViLT ở tất cả các chỉ số:
    - Accuracy: BEiT-3 đạt 82.4% so với 69.0% của ViLT.
    - F1-score: BEiT-3 đạt 0.807 so với 0.669 của ViLT, thể hiện sự cân bằng tốt giữa precision và recall.
    - Precision/Recall: BEiT-3 cao hơn ~13% so với ViLT, cho thấy khả năng phân loại và nhận diện tốt hơn.
- Việc fine-tune đã giúp cải thiện rõ rệt hiệu suất so với base model.
- ViLT dù nhẹ và nhanh hơn, nhưng không đạt hiệu quả cao bằng BEiT-3 trong môi trường thực tế, đặc biệt với tập dữ liệu phức tạp hơn.

=> BEiT-3 là mô hình phù hợp hơn cho bài toán VQA trong thực tế, nhờ vào kiến trúc mạnh và khả năng học biểu diễn ngữ nghĩa sâu sắc. ViLT tuy nhanh, nhưng cần thêm cải tiến nếu muốn đạt hiệu suất tương đương.

8. Resources:

- Kim et al., "ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision", 2021. - https://arxiv.org/pdf/2102.03334
- Image as a Foreign Language: BEIT Pretraining for All Vision and Vision-Language Tasks - https://arxiv.org/pdf/2208.10442
- Models:
    - https://huggingface.co/dandelin/vilt-b32-finetuned-vqa
    - https://www.kaggle.com/models/phong2004/beit-3-finetune-custom-vqa/PyTorch/default/2
    - https://www.kaggle.com/models/phong2004/beit-3-vqa-finetune/PyTorch/default/3
    - https://huggingface.co/phonghoccode/vilt-vqa-finetune-pytorch
- Dataset:
    - https://www.kaggle.com/datasets/phong2004/final-vqa-dataset
    - https://www.kaggle.com/datasets/ppdddd/vqa-dataset