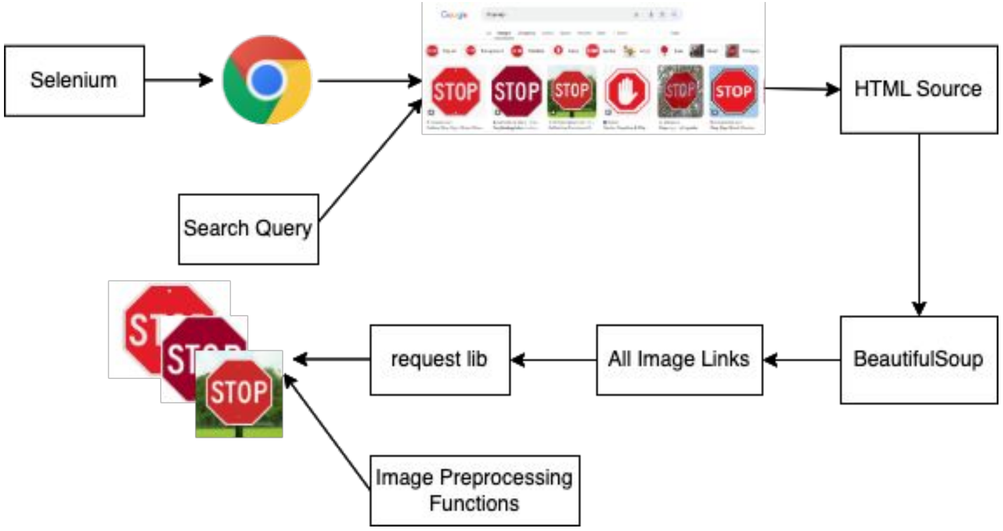
CSE 881 - Road Sign Detection Project

Bao Hoang and Tanawan Premsri

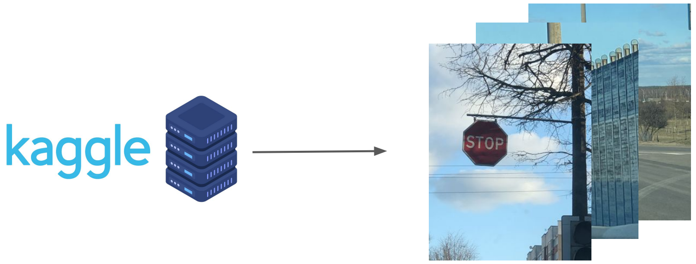
1. Introduction

- Require high-quality road sign detection systems for automobile
- Evaluate multiple architecture models
- Different source of data: Google, Kaggle dataset, Synthetic Image
- Labels: Stop, Crosswalk, Traffic Light, and Speed Limit.

2. Data Collection



Scrapped Image From Google



Kaggle Dataset

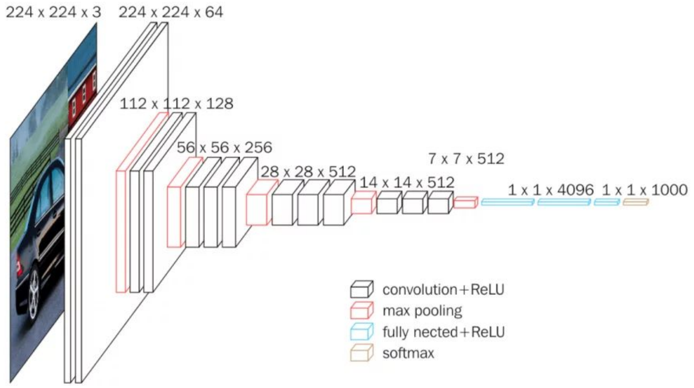
Synthetic Images

3. Models

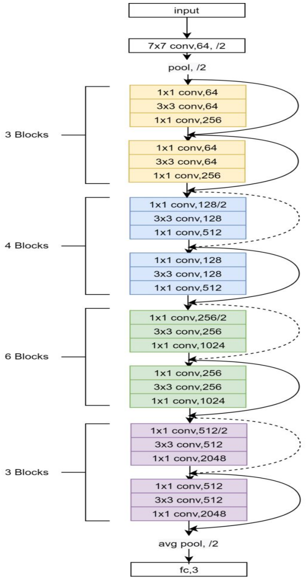
- We implemented 5 computer vision models:
    1. ResNet-50
    2. VGG-16
    3. CLIP
    4. LlavaNext-72B
    5. BLIP + LlavaNext-72B

4. CNN Architectures

- Both ResNet and VGG are widely used CNN architectures, known for their robust performance in image recognition tasks.
- We finetune ResNet-50 and VGG-16 using weights pretrained on the ImageNet-1k dataset.

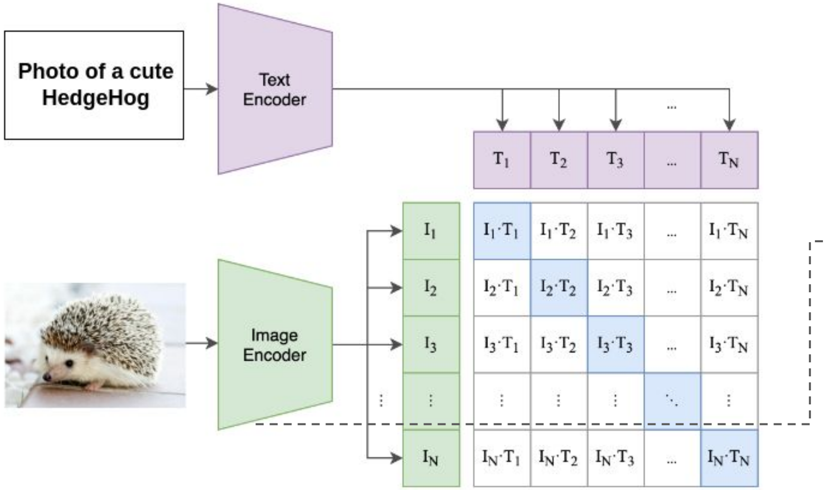


VGG-16

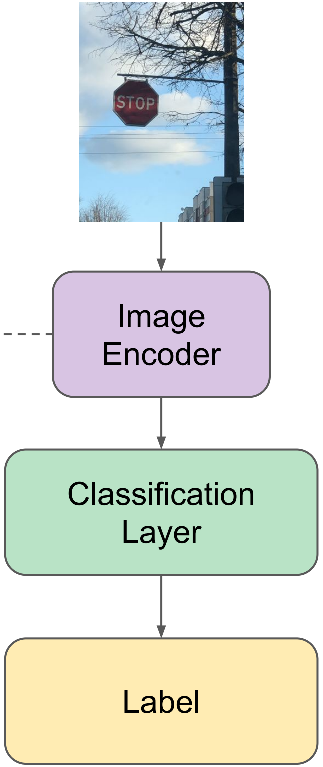


ResNet-50

5. CLIP

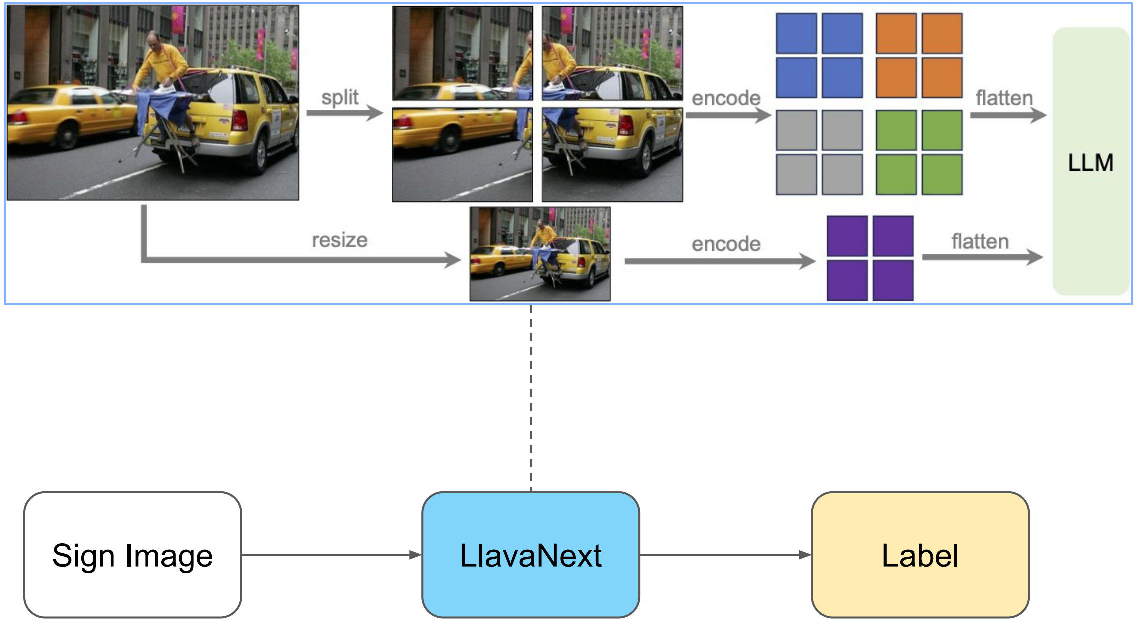


- Normal CLIP model
- Match embedded of text and image



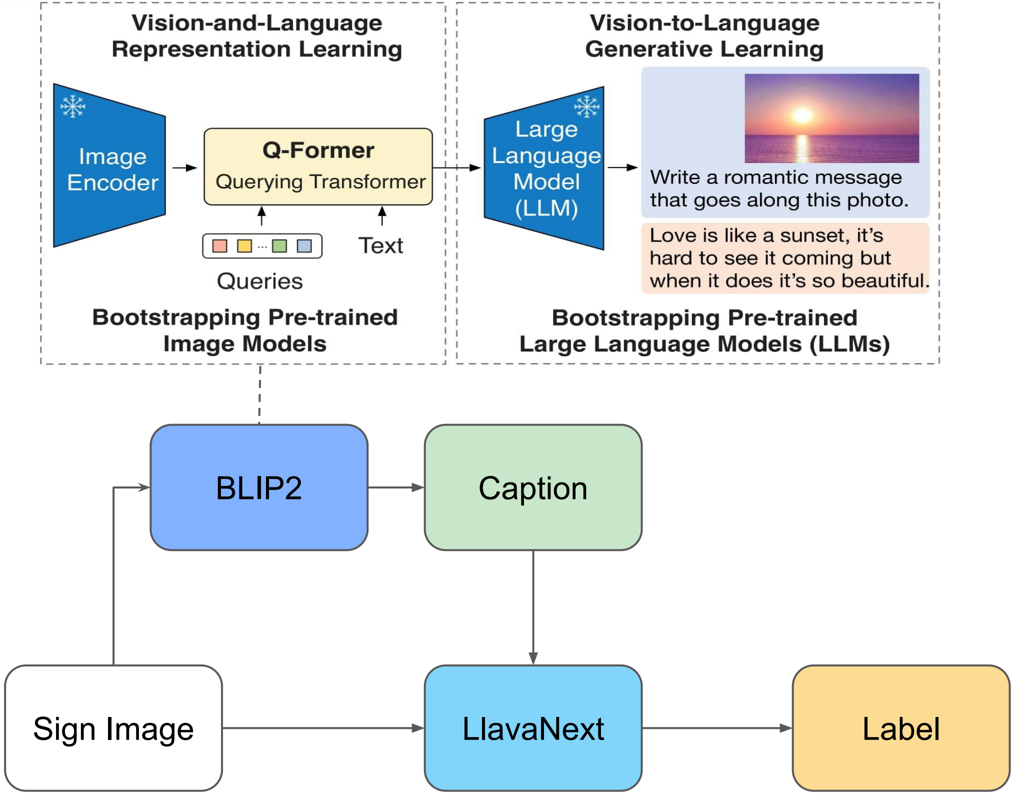
- Training classification layer based on encoder
- To adapt CLIP to our task more effectively

6. Llava-Next



- Combine image encoding with LLMs
- Exceptional performance in multiple tasks and capable of following instructions.
- Checkpoint:Llava-1.6-72b-hf
- Setting: Zero-shot setting

7. Llava-Next + BLIP



- BLIP exceptionally good at generate caption
- Use it to pre-generate augmented information of LlavaNext
- Checkpoint: blip2-opt-2.7b
- Setting: Zero-shot setting

8. Experimental Results

Model	Google images	Kaggle	Overall
VGG16 (Fine-tune on Google)	84.73%	66.67%	83.62%
VGG16 (Fine-tune on Kaggle)	51.75%	**97.56%**	54.55%
VGG16 (Fine-tune on all)	84.25%	94.31%	84.87%
VGG16 (Fine-tune on all + Image gen)	73.22%	94.60%	71.13%
ResNet (Fine-tune on Google)	85.15%	71.54%	84.32%
ResNet (Fine-tune on Kaggle)	53.08%	94.31%	55.60%
ResNet (Fine-tune on all)	**85.58%**	96.75%	**86.26%**
ResNet (Fine-tune on all + Image gen)	68.98%	89.43%	68.10%
CLIP (Train on Google)	83.08%	70.73%	82.33%
CLIP (Train on Kaggle)	33.35%	79.67%	36.18%
CLIP (Train on all)	83.19%	93.49%	83.83%
CLIP	73.12%	78.15%	77.15%
LlavaNext-72B (0-shot)	73.51%	90.00%	74.41%
BLIP + LlavaNext-72B (0-shot)	71.61%	72.72%	71.67%

Table 2: Accuracy of Different Computer Vision Model Architectures on Road Sign Dataset

- The best performance model is ResNet
- Change in image distribution affect model significantly
- Incorporating synthetic images provide negative effect rather than positive
- VLMs is promising even with 0-shot setting

9. Qualitative Results

- We classified the prediction errors into 4 types of error:
    - Type 1: Hard Examples
    - Type 2: Multiple Signs
    - Type 3: Irrelevant or Incorrectly labeled images
    - Type 4: Generation error

Type 1

Type 2

Type 3

9. Qualitative Results

Model	Type 1 Hard Examples	Type 2 Multiple Signs	Type 3 Unusual or incorrectly labeled images	Type 4 generation error
VGG16	22	20	58	0
Resnet	28	17	55	0
CLIP	37	16	47	0
CLIP Classifier	20	40	40	0
Llava	31	21	40	8
BILP + Llava	21	28	43	8

Table 3: Number of Images per Error in Misclassification Examples from Different Computer Vision Model Architectures

- Majority of error from incorrect labels/usual images
- Multiple signs error also contributes to lower score
- Llava still has hallucinations when generating the answer

10. Web Development

Users can upload road sign images and our app can return label of uploaded images

CSE 881 - Road Sign Detection Project

Authors: Bao Hoang and Tanawan Premsri

About the Project

With the rapid progress in autonomous driving technology, detecting and classifying road signs has become a critical task. Road signs provide essential information for safe and efficient navigation, making their accurate detection indispensable for modern autonomous vehicles.

This project leverages cutting-edge **Computer Vision** and **Deep Learning** techniques to build and evaluate high-performance road sign detection models. The models are trained on diverse road sign images collected from Google Images, Google Shopping, and Kaggle, covering 4 categories **Stop**, **Speed Limit**, **Traffic Light**, and **Cross Walk**. For more details, please refer to our source code and the final report at https://github.com/hoangcaobao/CSE881.

Below, you can upload an image of a road sign below to see how well our fine-tuned models (ResNet and VGG) can classify it!

Which Computer Vision Architectures you want to use?

VGG

Upload an image

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

Which Computer Vision Architectures you want to use?

VGG

Upload an image

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

STOP_sign.jpg 26.8KB

Uploaded Image

Uploaded Image Is Stop Sign

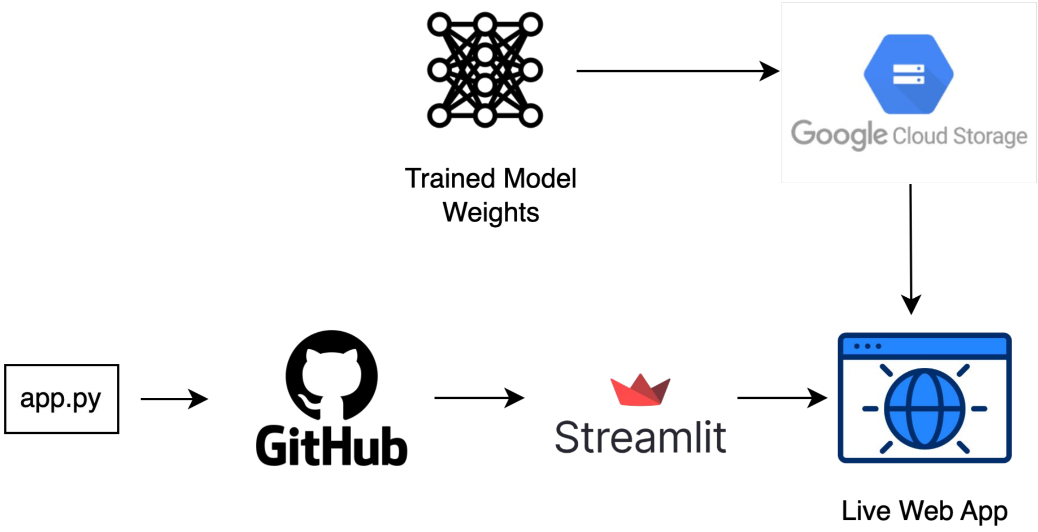
10. Web Deployment

To deploy our web application, we used Streamlit Community Cloud to host the app.

However, Streamlit requires us to upload the code to GitHub.

=> We could not upload the model weights directly to GitHub due to space limitations,

=> We used Google Cloud Storage to store the model weights and downloaded them once the deployment process was complete.



11. Discussion

What we accomplished?

- Collecting the image from different sources
- Evaluate various Computer Vision models
- Analysis on the misclassification images
- Website demonstration the task for best model

Future Directions

- Incorporate more comprehensive images to the road sign dataset
- Divide task and solve by specialize modules
- Develop better prompting strategy for VLM on the road sign detection

Thank you for listening