Panoptic Based Graph Captioning

A B.tech Project Guided by :- Dr. Tanima Dutta

Romit Mondal - 18075051
Kayala Nithin Sai - 18075028

Introduction

Image captioning: Image captioning means we need to describe the given image well. Image captioning is one of the most challenging problems which is a computer vision problem and a nlp problem.

Some of the major difficulties faced during image captioning are

- We should be able to generate a variable size sentence it is also called as the text generation problem.
- We need to understand the data/features from the given image this could also been understood.
- This understandings should consists of the features of the image which would be used for text generation.

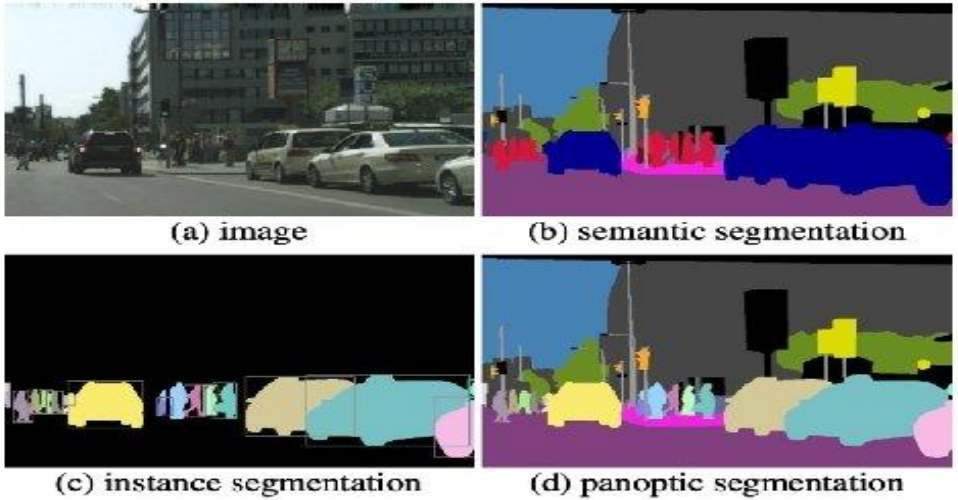
For extracting the features of the graph we would be using the panoptic segmentation.

Introduction (Segmentation)

- **Semantic segmentation** It segments the visual input in order to identify all the objects in the image. It assigns a class for each pixel.
- **Instance segmentation** It classifies each instance to a class prediction. This is also known as the object detection.

Panoptic segmentation It segments the visual input at once and classifies the image into 2 categories

- Things -- the countable instances are called the things
- Stuff -- the uncountable/amorphous background is called stuff



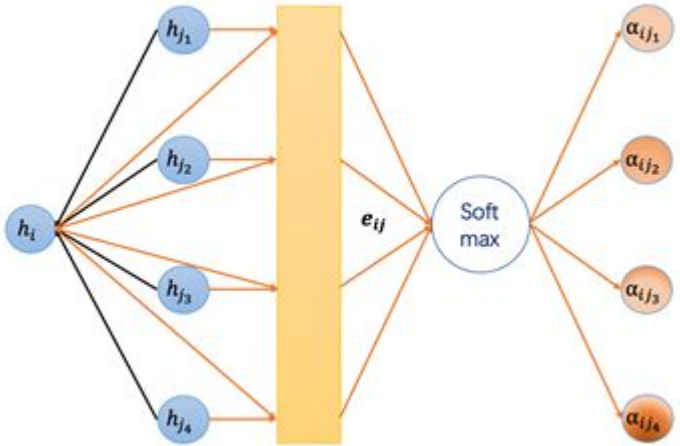
Why Panoptic for feature extraction

- We need the stuff and things features of the image to generate a caption.
- We need to gather as much data as possible.
- We could get the instance features form simple object detection but we would be missing so many features regarding the stuff.
- Similarly if we use the semantic segmentation we would be losing a huge information regarding the objects.
- So we choose the panoptic segmentation.
- When we have both the things and stuff features , then only we can learn the relationship between them using a Graph Attention Network.

Panoptic segmentation assigns two labels to each of the pixels of an image – (i) semantic label (ii) instance id. The pixels having the same label are considered belonging to the same semantic class and instance id's differentiate its instances. **Unlike instance segmentation**, each pixel in panoptic segmentation has a unique label corresponding to instance which means there are no overlapping instances.

Introduction (GAT)

- We use Graph Attention Network [7] as a means to understand the relationship between different instances/stuffs and hence helps in forming the sentence for image captioning [8].
- GAT introduces the attention mechanism as a substitute for the statically normalized convolution operation.
- Below are the equations to compute the node embedding hi(l+1) of layer l+1 from the embeddings of layer l.
- GAT uses weighting neighbor features with feature dependent and structure-free normalization, in the style of attention.



$$z_i^{(l)} = W^{(l)} h_i^{(l)}, \quad (1)$$

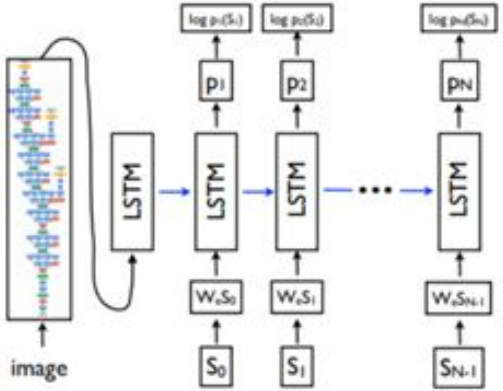
$$e_{ij}^{(l)} = \text{LeakyReLU}(\vec{a}^{(l)^T} (z_i^{(l)} || z_j^{(l)})), \quad (2)$$

$$\alpha_{ij}^{(l)} = \frac{\exp(e_{ij}^{(l)})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik}^{(l)})}, \quad (3)$$

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} z_j^{(l)} \right), \quad (4)$$

Introduction (Decoder)

- The Decoder architecture was proposed in a paper titled “Show and Tell: A Neural Image Caption Generator” by Google in 2015[1].
- The image shows the architecture.
- The image feature vector ‘I’ is inserted into the LSTM sequence at t=-1, only once. After that from t=0.
- the sequence of word vectors are input. The final output word of each step is the word with maximum probability obtained from the hidden state at that particular step.



Dataset

We have used MS COCO Dataset 2014 using COCO API for our training purpose .

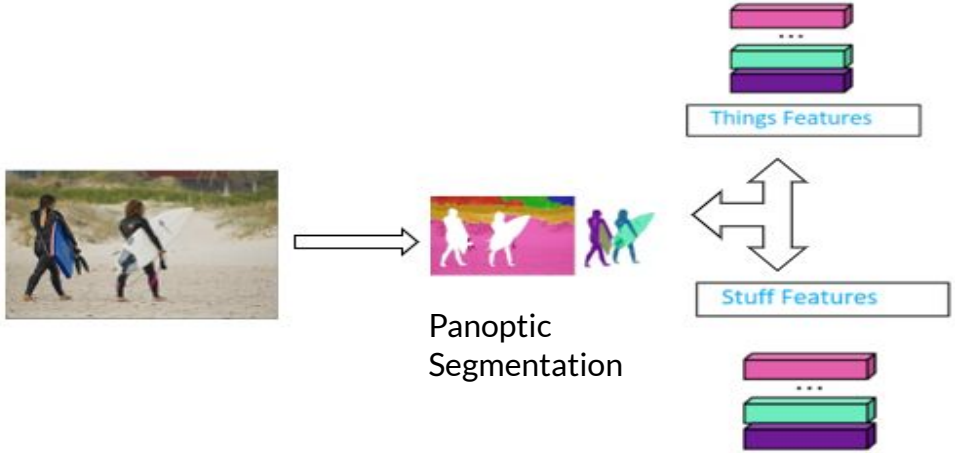
In the 2014 COCO Dataset , there are:

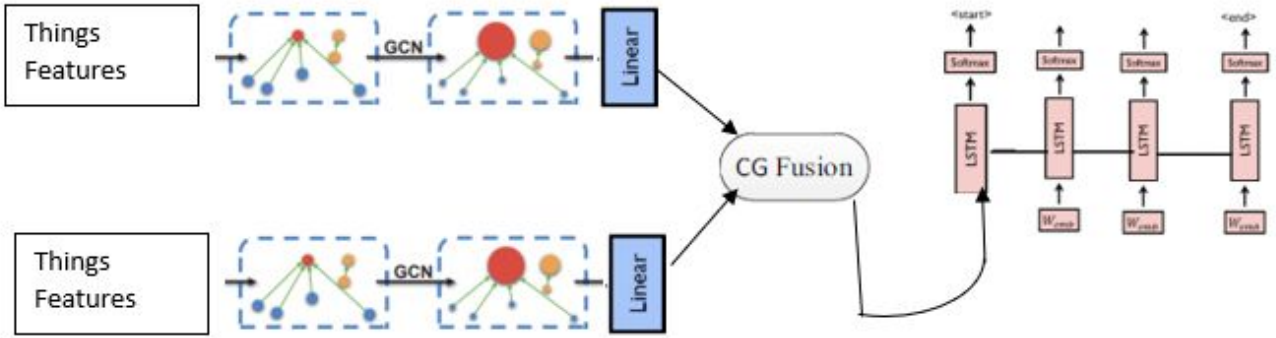
- 83K Training images
- 41K Validation images
- 41K Testing images

We are performing 3 main tasks with our Dataset . They are as follows :

1. Object Detection - each image has a set of bounding boxes and masks representing the objects.
2. Panoptic Segmentation - each pixel of the image has a class id and an instance id.
3. Image Captioning - Each image has at least 5 captions in the annotations.

MODEL DESIGN





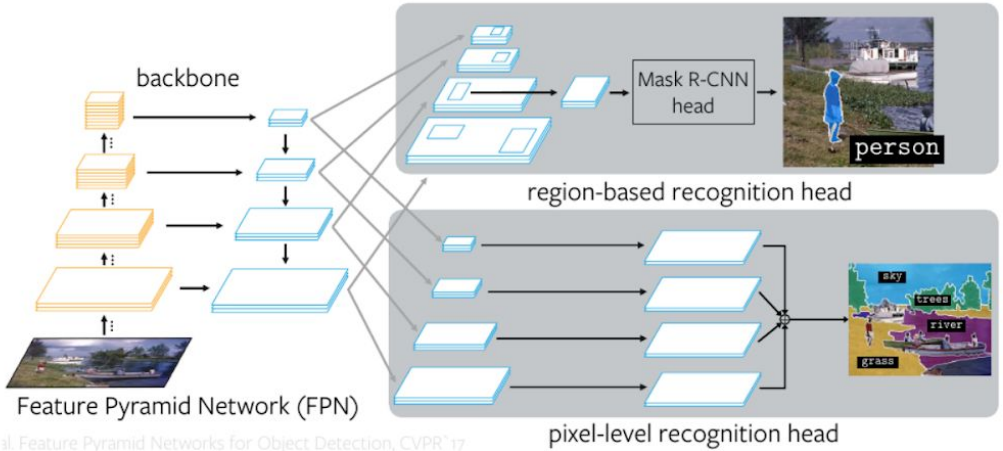
Proposed Methodology

- Building Panoptic Segmentation Model
- Extracting ROI Features
- Graph Attention Network
- Context Gating
- Decoder Recurrent Neural Network (RNN)

Building Panoptic Segmentation Model

We are using Detectron2 by Facebook AI to obtain the following:

- No. of instances(n)
- Bounding box coordinates for each instance
- Semantic Segmentation Mask
- Instance Segmentation Mask for each instance
- Confidence Score for classification of objects



Model in-depth : The idea is to use FPN for multi-level feature extraction as backbone, which is to be used for region-based instance segmentation as in case of Mask R-CNN, and add a parallel dense-prediction branch on top of same FPN features to perform semantic segmentation [9].

- Consider we input an image (h,w,3) where 3 is the number of channel BGR.
- The Panoptic Segmentation model will detect n instances and for each instance/stuff , bounding box coordinates (x1,y1,x2,y2).

So , it will return a tensor of size (n,4) for both things/stuff.

Extracting ROI Features

- We will obtain the backbone layer of the Detectron2 model .
- The role of the backbone network is to extract feature maps from the input image.

Input : image of size (batch_size,3,h,w)

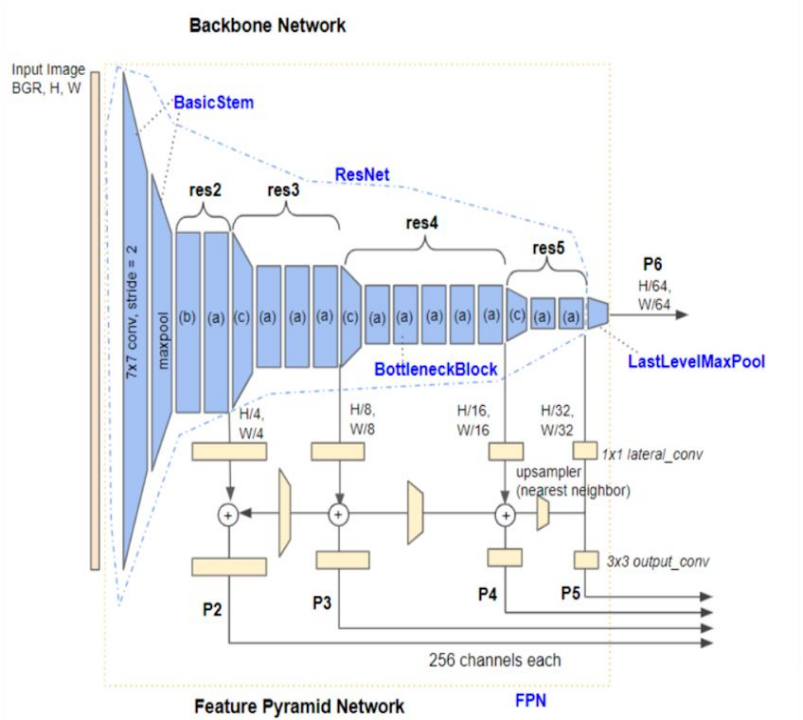
Output : A dictionary of tensors . Each with size (batch_size,256,h/s,w/s).

Here , channels is 256 by default

s is stride (p2-> 4, p3->8,p4->16,p5->32)

- We want to extract P2,P3,P4,P5 features from the FPN as shown in the figure.
- To extract the corresponding features form the bounding boxes we scale the corresponding bounding boxes according to the stride and extract the roi align features.

Adding all the roi features from p2,p3,p4,p5 , and flattening each instance/stuff features to convert (n,256,7,7) into (n,12544) . So we get each instance/stuff feature vectors which can be passed into the Graph as node features.



Graph Attention Network

Input is a list of n roi feature vectors . Each can be treated to be a node in the graph .
Initially the Graph is considered fully connected with no. of nodes equal to no. of instances in the images .

Equation (1) : $$z_i^{(l)} = W^{(l)} h_i^{(l)}, (1)$$ The first is a representation of linear transformation.

Equation (2) : $$e_{ij}^{(l)} = \text{LeakyReLU}(\vec{a}^{(l)^T}(z_i^{(l)} | z_j^{(l)})), (2)$$

The un-normalized attention score eij is calculated using the embeddings of adjacent nodes i and j. This suggests that the attention scores can be viewed as edge data.

Equation (3) and (4) :

$$\alpha_{ij}^{(l)} = \frac{\exp(e_{ij}^{(l)})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik}^{(l)})},$$

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} z_j^{(l)}\right),$$

The equation (3) constructs the relational feature matrix which are used to understand the relation features between the input features.
The equation (4) updates the relational features of the input features constructed with equation (4).

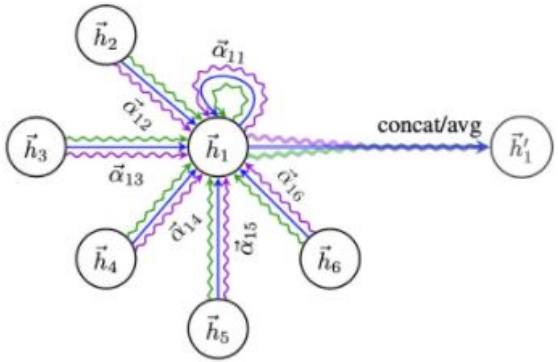
Graph Attention Network

Our Final Graph Attention Network (GAT) Layer Model :

- input dimension = 12544
- output dimension = 200

The output is then passed to a softmax layer .

So, the input of size (n,12544) after passing through GAT layer becomes an output of size (n,200) .



Mostly we consider top 20 instances features . So n=20 for instance .

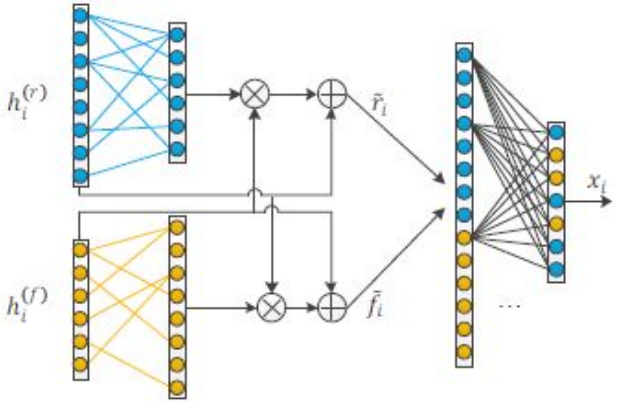
We considered top 10 stuff features . So n=10 for stuff .

Then we flatten the (20,200) instance features from graph to (1,4000) ; the (10,200) stuff features from graph is flattened to (1,2000) which is then passed through a Linear layer to reduce its dimensionality to (1,256).

So , both stuff features and thing feature maps are of dimension (1,256) .

Context Gating

The things features from the graph and the stuff features from the Graph needed to be aggregated to get a better understanding of the image so that we have used a fusion technique known as the context gating let Hi(r) be the things features and Hi(f) be the features of stuff features. The fusion is done with the help of the below equations.



$$\tilde{r}_i = \text{Gating}_r^{(E)}\left(h_i^{(f)}, h_i^{(r)}\right)$$

$$\tilde{f}_i = \text{Gating}_f^{(E)}\left(h_i^{(r)}, h_i^{(f)}\right)$$

Here r' and f' are the gated results and with the below equation these are further fused to get the final context gated result.

$$\text{Gating}\,(x, y) = \sigma\,(wx + b)\,y + y, \qquad x_i = w^{(E)}\left(\left[\tilde{r}_i, \tilde{f}_i\right] + b^{(E)}\right),$$

The two 1*256 feature maps from things and stuffs are passed into the context fusion gate and we get a single feature map of dimension (1,256) , which can then be passed to the decoder.

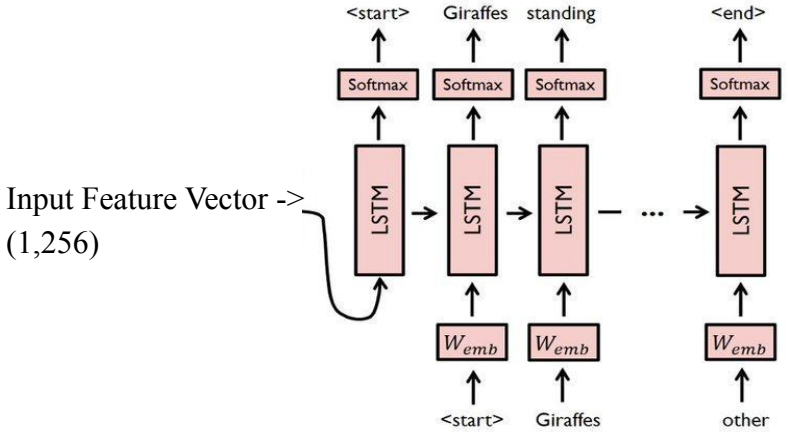
Decoder Recurrent Neural Network (RNN)

- The Sequence Processor model expects input sequences with a predefined length (max sequence length) which are fed into an Embedding layer that uses a mask to ignore padded values. This is followed by an LSTM layer with 256 memory units.
- The words in the caption are embedded to a higher dimensional space of (1,256).
- The input feature vector is of dimension (1,256). So it is concatenated with the word embeddings.
- This concatenated embeddings of size (padded caption length +1 , 1, 256) is passed to the LSTM.

This is then fed to a Dense neuron layer of size = vocab_size that makes a softmax prediction over the entire output vocabulary for the next word in the sequence. The output of the Dense Layer is (max(caption lengths),) and it is our predicted caption. This means the output is already padded up to the max caption length. Targets is the list of caption words which is padded up to the max caption length. This is done using pack_padded_sequence.

- Loss is calculated with respect to outputs against targets with cross entropy

This is how our model works taking an image as input and producing a caption output.



Training Details

1. **Graph Attention Network Model** :

The Graph Attention network has input features of size N * F where N means the number of things/stuff and F means the dimension of the input feature map, the output will be a learned relational feature map between these nodes, the N being 20 for things and 10 for stuff the output being N * F' where F is 12544 for input and F' is 200.

2. **Decoder Model** :

```
[55] print(decoder)

     DecoderRNN(
       (embed): Embedding(9956, 256)
       (lstm): LSTM(256, 512, batch_first=True)
       (linear): Linear(in_features=512, out_features=9956, bias=True)
     )
```

3. **Linear Layer and its Weights and Parameters after training** :

The dimensions for things dense layer is 4000 * 256.

The dimensions for the stuff dense layer is 2000 * 256

4. **Roi_Align Model** :

The roi align converts the features within the bounding box to 7 * 7 features maps with 256 channels.

```
[65] print(roi_align)

     ROIAlign(output_size=(7, 7), spatial_scale=1, sampling_ratio=2, aligned=True)
```

Results

- Bleu score for simple encoder-decoder our model: 17.37
- Bleu score for encoder-decoder paper: 27.7 (Show and tell)

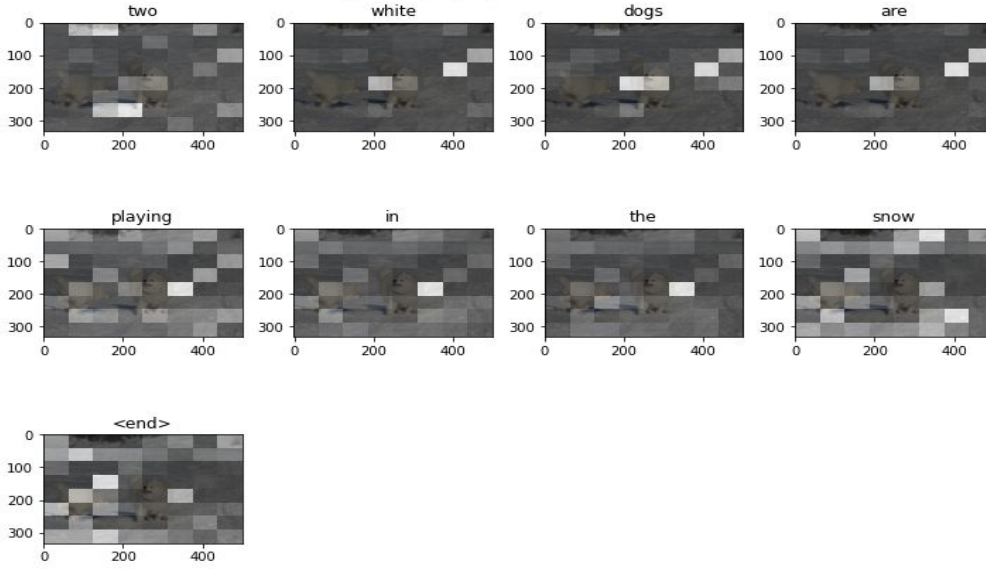
- Simple Encoder Decoder (Base Model)
- Encoder-Decoder with Attention Mechanism and Beam Search
- Panoptic based Graph Captioning

Simple Encoder Decoder (Base Model) :

```
BLEU SCORE:0.6389431042462724
Target Caption:<start> A group of player playing soccer on a field . <end>
Predicted Caption:<start> a group of people playing soccer on a field . <end>
```


```
BLEU SCORE:0.368740304976422
Target Caption:<start> A dog trying to catch a frisbee in a field. <end>
Predicted Caption:<start> a man is standing in the grass with a frisbee . <end>
```

BELU score: 72.59795291154771
Real Caption: Two white dogs are playing in the snow
Prediction Caption: two white dogs are playing in the snow



Real Caption: motorbike jumping pole
Prediction Caption: group of people are standing on the ground

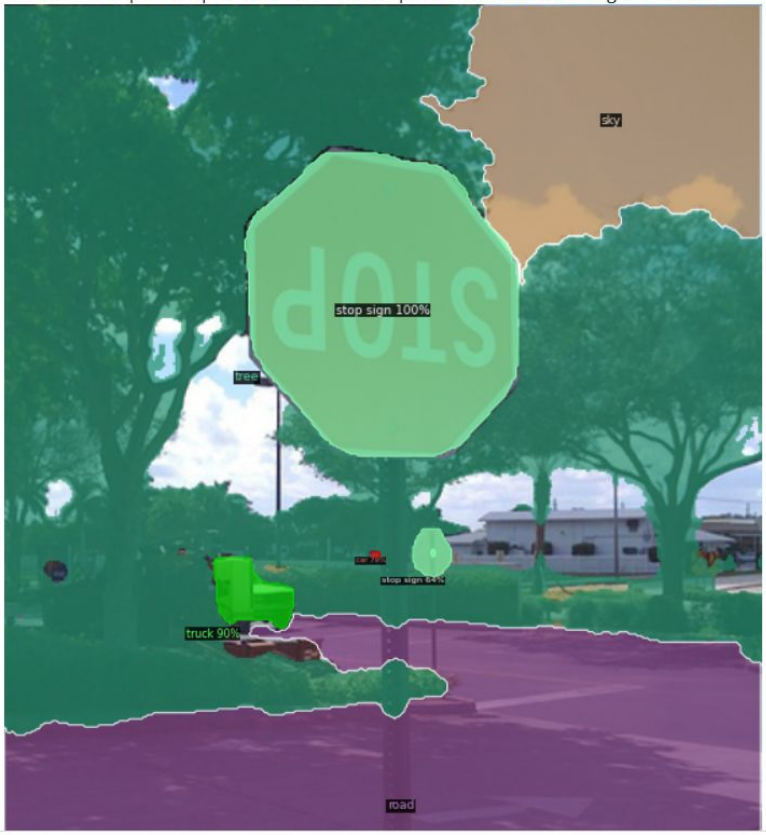
Encoder-Decoder with Attention Mechanism and Beam Search

Panoptic based graph captioning

```
BLEU SCORE:0.7071067811865476
Target Caption:<start> A man sitting on a bench in a park . <end>
Predicted Caption:A man sitting on a bench . <end>
/usr/local/lib/python3.7/dist-packages/nltk/translate/bleu_score.p
Corpus/Sentence contains 0 counts of 2-gram overlaps.
BLEU scores might be undesirable; use SmoothingFunction().
  warnings.warn(_msg)
```


```
BLEU SCORE:0.21470779802151024
Target Caption:<start> A man playing tennis . <end>
Predicted Caption:standing bowl full of green and white . <end>
```


```
BLEU scores might be undesirable; use SmoothingFunction().
  warnings.warn(_msg)
BLEU SCORE:0.13986904558770089
Target Caption:A stop street sign on the road
Predicted Caption:up a blue and white photo of a street sign . <end>
```



Conclusion and future scope

In conclusion , in our B.Tech Project , we have worked on an image captioning model which gives human-like captions because of the following :

- A Panoptic Segmentation Model which extracts all the ‘things’ features and bounding box coordinates.
- An ROI_ALIGN feature extractor which gives the raw features for each bbox coordinate.
- A Graph Attention Network which not only learns from our input roi features but also understands the relationships among various nodes. This helps us attain captions which are more human-like.
- An LSTM Decoder one-to-many model which predicts the image captions .

Future Scope :

- The model needs further training.
- It can be extended to a real-time video captioning model.

References

[1] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: A neural image caption generator,” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, doi: 10.1109/cvpr.2015.7298935.

[2] Z. Liu and J. Zhou, *Introduction to Graph Neural Networks*. Morgan & Claypool Publishers, 2020.

[3] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, “Panoptic Segmentation,” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, doi: 10.1109/cvpr.2019.00963.

[4] M. M. Chanu, “A Deep Learning Approach for Object Detection and Instance Segmentation using Mask RCNN,” *Journal of Advanced Research in Dynamical and Control Systems*, vol. 12, no. SP3. pp. 95–104, 2020, doi: 10.5373/jardcs/v12sp3/20201242.

[5] W. Cai, Z. Xiong, X. Sun, P. L. Rosin, L. Jin, and X. Peng, “Panoptic Segmentation-Based Attention for Image Captioning,” *Applied Sciences*, vol. 10, no. 1. p. 391, 2020, doi: 10.3390/app10010391.

[6] X. Yang, K. Tang, H. Zhang, and J. Cai, “Auto-Encoding Scene Graphs for Image Captioning,” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, doi: 10.1109/cvpr.2019.01094.

[7] Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y, “Graph Attention Networks,” *arXiv*, Oct. 30, 2017.

[8] P. Zhou and M. Chi, “Relation Parsing Neural Network for Human-Object Interaction Detection,” *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, doi: 10.1109/iccv.2019.00093.

[9] H. Liu *et al.*, “An End-To-End Network for Panoptic Segmentation,” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, doi: 10.1109/cvpr.2019.00633.

Thank You

By :

Romit Mondal - 18075051

Kayala Nithin Sai - 18075028

We would like to extend our gratitude to our tutors **Deepali** Mam and **Sonali** Mam for their unending support and motivation.