Semantic Image Segmentatio: Using Knowledge Graphs to Improve Local Feature Mapping in Convolutional Networks

1st Samuel Mcmurray
Computer Science
Jönköping University
Jönköping, Sweden
mcsa22oo@student.ju.se

2nd Fatima Umar
Computer Science
Jönköping University
Jönköping, Sweden
umfa22or@student.ju.se

***Abstract*—Image segmentation is a challenging computer vision task due to the understanding of the world at the semantic level. The use of knowledge graphs within a convolotional network to improve global feature representation further enhancing local features. This paper implements a Symbolic Graph Reasoning layer within a convolution network, evaluating the performance against a U-Net ResNet-101 model as well as looking at the feature maps within the Symbolic Graph Reasoning layer. The results showing that the Symbolic Graph Reasoning layer out preforms the ResNet-101 for both F-Measure and MeanIoU by roughly 5-7%. Unfortunatly due to the input size no conclusive evidence could be found as to whether the Symbolic Graph Reasoning layer had an impact based on the feature mappings due to the shapes of the layers, more experiments would need to be conducted. Although, based on the quantitative and visual results it would appear to have an impact on the model.**

***Index Terms*—Semantic Segmentation, Deep Learning, Convolutional Neural Network, Graph Reasoning, Computer Vision**

I. Introduction

Image segmentation is one of the more challenging tasks as stated in Yanming Guo et al.[2]in computer vision as segmenting an image requires an understanding of the at the semantic level of the world and the things that are present in it. As described by Shervin Minaee et al.[7] is done by classifying an object with semantic label at the pixel level, thus being computationally more demanding as compared to the classification of an entire image like that of image classification. Since the begging of image segmentation many algorithms and methods have been developed, however in recent years the use of Deep Learning(DL) has seen substantial improvements over earlier methods. The Convolutional Neural Network(CNN) architecture is one of the most commonly used architectures within the computer vision tasks due to their improved performance over others. Many models have been developed based on the CNN architecture within the field to improve on different aspects include Fully Convolutional Networks(FCNs), Encoder-Decoder, CNNs with graphical models and many more. The models with the most notoriety today are AlexNet, VGGNet, ResNet, SegNet, and DeepLab.

Liang et al.[6] reasoned that recognizing objects within convolutional networks advances could be hindered due to the lack of explicit reasoning over contexts and high level semantics. The argument that global semantic coherency can be achieved by the incorporation of commonsense human knowledge into feature representation to learn beyond local convolutions. This is achieved by integrating three modules into a CNN as a Symbolic Graph Reasoning(SGR) layer. The first module is the local to semantic voting in which the local features are used to map them to different symbolic nodes through voting. The second module is the graph reasoning layer enhances the features by performing graph reasoning over all the symbolic nodes that were generated from the first module using an embedded vocabulary and a knowledge graph. The third semantic to local module takes the evolved symbolic nodes from the graph reasoning and learns the appropriate associations between those and the local features.

The experiment conducted by Xiaodan Liang et al.[6] was conducted on three different datasets COCO-stuff which contained 182 semantic concepts, PASCAL-Context which contained 59 semantic concepts, and ADE20k which contained 150 semantic concepts, resulting in a concept graph of 340 concepts. In addition to the semantic segmentation of the datasets the authors also included a classification problem on the CIFAR-100 dataset with 100 classes. Their SGR model used the ResNet101 as the backbone convolution network with the pretrained weights of ImageNet implementing Atrous Spatial Pyramid Pooling(ASPP) between the ResNet block and the SGR layer. The results of each of the datasets showing improvements over other models tested on the Mean Intersection of Union(mIoU) of roughly 2-3% in addition multiple configurations of the SGR model which saw more minimal improvements. The classification results were stated to be comparable to the state of the art models seeing improvement over the baseline models of the ResNet and DenseNet-100 of roughly 4%, noting that the number of parameters were significantly less.

Although the experiment Xiaodan Liang et al.[6] had conducted was quite vast incorporating varying datasets and comparing various models including state of the art models at the time further investigation into the model needs to be considered. The individual mappings within the SGR layer were not evaluated on how well they had performed whether the symbolic nodes being incorporated had an effect. Additionally the knowledge graph was not investigated as to whether improvements can be made based on the knowledge graph by expanding it or incorporating more meaning into it. Due to the limitations of the project only the mapping will be investigated in context to the symbolic nodes and the new outputs.

In this paper the SGR model is implemented to the greatest extent possible, comparing the model based on meanIoU, class accuracy against U-Net ResNet-101 model, and investigate the feature mappings produced by the SGR layer. The results showing that the implementation does improve performance over the baseline model. Unfortunately due to the resizing of the the layers within the SGR the outputs are not observable in many cases until the final output. In addition while observing the feature mapping from the ASPP to the final output layer, the input size severely hinders the deep network as when the ASPP begins the size of the image is 4x4. With that more experiments would be needed to investigate whether the reshaping of the features is needed or whether measures can be taken to change the convolutions and the outputs so that they take the correct shape then are reshaped after or prior. Additionally, this experiment should further be undertaken with more context images with more classes available with images of larger size.

II. Background and Related work

According to Shervin Minaee et al.[7] the most popular type of segmentation models are the encoder-decoder architectures, although they have limitations due to the loss of resolution through the encoding process resulting in a loss of fine grained image information. Some models such as DeConvNet and SegNet recover this information while others such as HRNet try to maintain those high resolution representations. U-Net and V-Net are another type of encoder-decoders that are also heavily inspired by FCNs which were initially developed for the medical domain segmentation but have been adopted for their uses.

As stated by Muhammad Shafiq et al.[8] A deep Residual Network(ResNet) is a type of CNN that uses residual connections for the input of from the previous layer to be added to the output of the current layer. According to Kaiming He et al.[3] ResNet were designed to solve the problem that occurs within deep networks known as the degradation problem where the accuracy gets saturated and then degrades. By using the residual mapping to optimize the unreferenced mapping, this allowing for deeper networks with improved accuracy gains over plain networks with no residual blocks.

Xia Li et al.[5] developed a model for semantic segmentation which had a ResNet101 backbone called Spatial Pyramid Graph Reasoning. Differing from other techniques incorporating graph reasoning the original feature space organized as a pyramid. In this model the original features are down sampled then up sampled and the output is summed together until the original feature is added back again, in between the down and up samples graph reasoning is performed. In addition a data-dependent improved Laplacian is used with a attention diagonal matrix for better distance metric within the graph. It was found that this model showed good performance gain outperforming other methods with little to no increase in computational costs.

III. Method

The experiment itself is intended to follow the implementation set out by XiaodanLiang et al.[6], as to provide an optimal review of the original as developed. Some changes were necessary to accommodate for our hardware/software or where information provided is ambiguous in nature. The knowledge graph itself is not defined within the article only stating the total vocab, in addition the embedding is briefly spoken of no mention of how or if it was trained was given.

A. Preprocessing

In semantic segmentation each of the images has a ground truth, the ground truth is either an RGB image or a grayscale image. The value for each of the pixels within the image is associated to a particular class, in a grayscale image it is from 0-255 in terms of RGB it is 3-channels from 0-255 for each channel. Each class needs to be mapped to a color or value for continuity between all the images within the dataset, in RGB this is a colormap. The output in the model for multi-class semantic segmentation is a convolution layer with the filters representing the number of classes that uses a softmax activation function. In order for the loss to be calculated a mask needs to be made for each of the classes within a ground truth image. A mask is a binary representation for each of the pixels 0 being black and 1 being white, if the object in an RGB ground-truth image is the class of the current mask being made all the pixels of the corresponding color will be 1 all other pixels will be 0. After all the masks are created they are stacked to creat a channel dimension based on the number of classes present in the particular dataset.

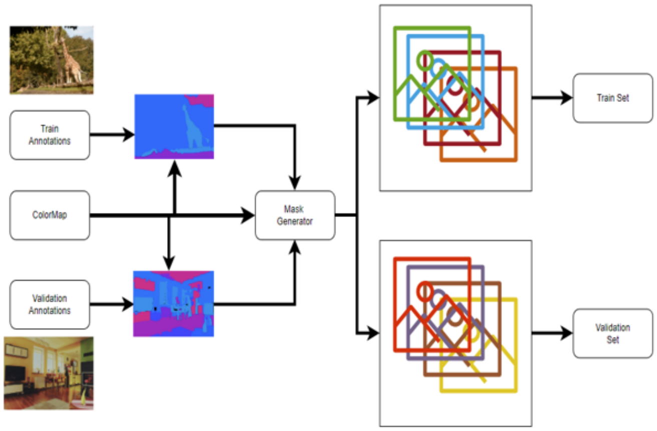

Fig. 1. Creation of Ground-Truth Images and Masks

In the case of COCO-stuff the RGB ground-truth images can be generated using the annotation JSON file in conjunction with the COCO API. For each of the classes present in an image a colormap can be used in order to mark all the colors on the image based on the associated classes. Found in Fig. 1 the training and validation annotations are used to create a RGB ground-truth image with a color map that same color map is then used to create a mask that has each color representing a class as a channel in that mask for the training and validation sets.

B. Artous Spatial Pyramid Pooling

Presented in Liang-Chieh Chen et al.[1] Deep CNNs spatial resolution of feature maps are reduced to a significant degree in these networks due to the repeated combination of max-pooling and striding within consecutive layers. The typical solution for such a problem is requires more time and memory, but another solution is to use atrous convolution allowing for setting a desirable resolution to any layer. This is done by introducing zeros in between the up sampled filter samples values taking only into account the non-zero values. This allows for the feature responses of neural network to be controlled in terms of the spatial resolution.

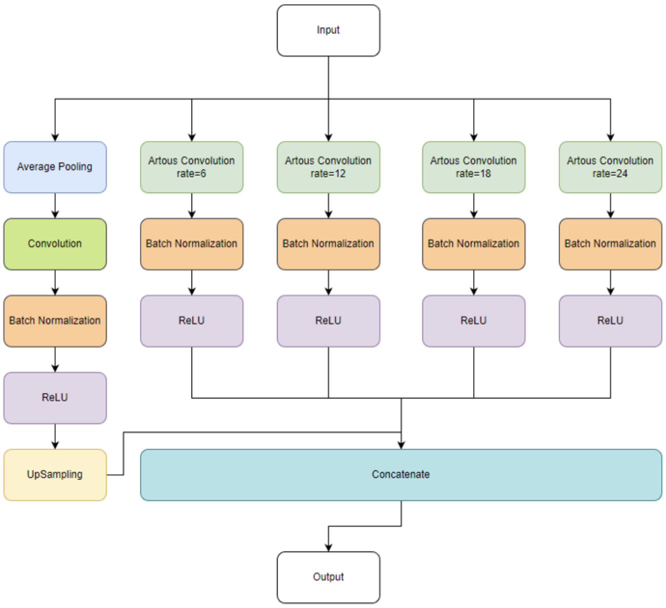

Fig. 2. Atrous Spatial Pyramid Pooling prior to Semantic Graph Reasoning

Spatial Pyramid Pooling(SPP) found in Kaiming et al.[4] allows for variable sized inputs, in doing so allows for different scales and aspect ratios. In turn this makes it possible to resize the input image to any scale which in traditional methods plays an important role for accuracy of deep networks. Liang-Chieh Chen et al.[1] took inspiration from this by using multiple atrous convolutional layers in parallel with different sampling rates that are processed and fused to produce a final output. Found in Fig. 2 is the ASPP implementation found in Xiaodan Liang et al.[6] where the rates on the atrous convolution layer is 6,12,18, and 24. The inputs coming from the ResNet convolution block with a channel dimension size of 2048 being reduced down to the desired 256 channel dimension size on the output of the ASPP.

C. Local-To-Semantic Voting Module

The local to semantic voting module as described in Xiaodan Liang et al.[6] first summarizes the encoded global information contained within the local features into representations of symbolic nodes. The characteristics found from the aggregate of the local features in which are correlated to specific semantic meanings corresponding to a specific symbolic node. Found in Fig. 3 the input is the feature tensor where B representing the batch, H representing the height, W representing the width, and D representing the channel dimension of the feature maps. The convolution on the right hand side uses the size of the vocabulary M as the output channel dimension representing each as a class in order to apply the softmax activation function for normalization. The left hand convolution outputs the desired channel dimension which is reshaped by combining H and W into a singular axis, the output from the softmax function on the right side is similarly reshaped by combining the H and the W into a single axis then is transposed for the matrix multiplication of the both the left hand and right hand sides. The output from the matrix multiplication is now that of M which the size being 204 in this implementation and the desired channel dimension, a ReLU activation function is applied and the output is used in the graph reasoning module.

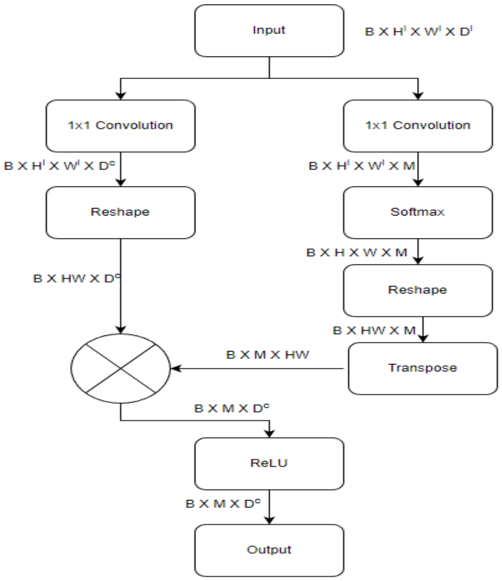

Fig. 3. Local To Semantic Voting Module

D. Graph Reasoning Module

The graph reasoning module handles the evolving of the global representations of the symbolic nodes by incorporating the linguistic embedding and graph in conjunction with the previous module output of those symbolic nodes found in Xiaodan Liang et al.[6]. The implementation changes slightly as the word vectors from Fasttext was not available a simple embedding is used in place with word vectorization. The graph reasoning module shown in Fig. 4 the input using the output of the local to semantic voting module which is concatenated with the word embedding k being the embedding dimensions which is 100 for this implementation the concatenated output is then applied to a linear layer.

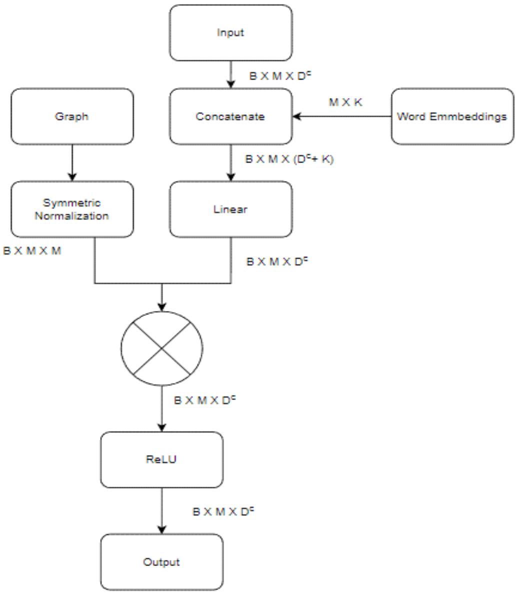

Fig. 4. Graph Reasoning Module

The edge connections from the graph defined by the node adjacency weights which are soft weights, with them being soft weights symmetric normalization is implemented to take the average of the adjacent node features. The output from the linear layer and the symmetric normalization is then matrix multiplied to then a ReLU activation function is applied resulting in the evolved global representations for the output of the graph reasoning layer.

E. Semantic-To-Local Module

The semantic to local module allows for the each of the local feature representations to be boosted based on the evolved global representations of the symbolic nodes as described by Xiaodan Liang et al.[6]. As shown in Fig. 5 the evolved feature representation from the output of the graph reasoning module is used to be concatenated with the original features that are reshaped to the combined H and W in a single axis. The concatenated tensor is then used in the left side convolution layer which has softmax applied then matrix multiplied with the evolved features from the right side convolution, which then has a ReLU activaition function applied and it reshaped to the original shape. The result is then added to the original tensor which is the final output for the SGR layer.

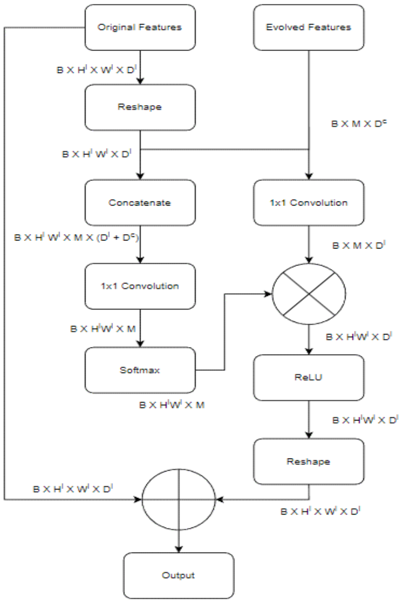

Fig. 5. Semantic to local module

IV. Experiment Implementation

Initially the dataset chosen was COCO-Stuff due to the number of classes and the context aspect that was needed to be captured within the experiment. After several experiments and testing the results of the accuracy, loss, and meanIoU were abnormal, the loss would start low until it hit a threshold of epochs increasing exponentially simultaneously the accuracy would fall. After careful investigation into the masks the coco annotations would not fill all the classes colors, and rather would outline the class shown in Fig. 6. In the face of this the dataset was changed to PASCAL VOC containg 2912 total images, and 21 classes and masks unfortunately due to time constraints using PASCAL Context was not feasible.

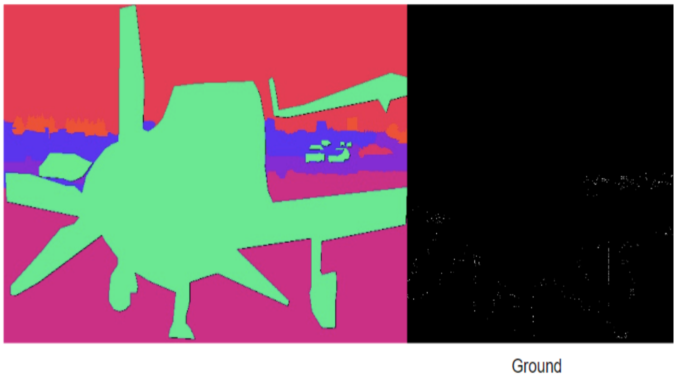

Fig. 6. Problem with some of the COCO-stuff annotations

The experiments were conducted using Tensorflow 2 with RTX 3070 8GB graphics card, the backbone was ResNet-101 pretrained on Imagenet following the implementation by Xiaodan Liang et al.[6]. The ASSP used the dilation rates of 6, 12, 18, and 24 reducing the final residual block from a channel dimension of 2048 to 256 with the SGR layer following, due to VRAM limitations the image size was set to 128 X 128. The optimization was performed with Stochastic Gradient Descent with the learning rate set to 2.5e-3 with the number of epochs set to 100. The word total vocabulary for the word embeddings was 204 the number of embedding dimensions was set to 100. The baseline model that will be compared is the U-Net architecture with the ResNet-101 backbone with the weights being pretrained on Imagenet.

V. Results & Analysis

The results of the U-Net Resnet-101 and the SGR implementations found in Table I shows that for both F-Measure and MeanIoU the SGR implementation performed better by roughly 5-7%. Shown in TableII are the individual classes F-Measure and MeanIoU scores lending some insight into where the models were struggling and where they had succeeded. Both of the models had difficulties with the cow class for segmentation with both performing less than 5% accurately on both metrics, in addition the cows both models found it difficult to segment motorcycles and televisions or computer monitors as well. In terms of success both models performed well in segmenting people, horses, cats, bottles and people, although in terms of performance the SGR model outperformed to a great deal on most of the classes.

TABLE I
Validation Results

Metric	ResNet-101	SGR
F-Measure	57.52%	64.63%
MeanIoU	51.22%	57.99%

TABLE II
Validation Results Class Scores

Class	ResNet F1	ResNet MIoU	SGR F1	SGR MIoU
Background	65.06%	58.95%	74.4%	66.07%
Boat	62.33%	57.14%	66.4%	62.13%
Bicycle	57.23%	51.60%	63.40%	53.64%
Bus	57.57%	49.07%	42.99%	34.70%
Airplane	33.64%	23.79%	78.94%	71.69%
Bottle	61.18%	56.45%	65.92%	60.95%
Bird	54.63%	51.06%	60.37%	55.07%
Car	78.83%	75.14%	77.25%	70.70%
Cat	67.41%	58.77%	76.95%	72.16%
Cow	03.86%	02.37%	04.68%	02.84%
Horse	74.95%	69.90%	78.69%	75.79%
Motorcycle	41.45%	33.14%	33.00%	23.91%
Person	74.47%	69.03%	87.30%	82.86%
Potted Plant	54.45%	45.09%	51.91%	44.01%
TV/Monitor	41.31%	32.80%	47.14%	38.33%
Sofa	47.00%	38.62%	73.92%	68.55%
Sheep	71.73%	65.95%	74.54%	67.82%
Train	63.45%	57.07%	79.11%	73.39%
Chair	60.43%	56.32%	73.29%	63.29%
Dog	63.25%	58.13%	67.83%	59.89%
Dining Table	73.73%	65.16%	79.11%	70.04%

As for the direct predictions found in Fig. 7 the Resnet model seems to have a better shape but it is identifying other classes where they aren't any other classes as for the SGR model it performs well in terms of semantics although its pixel selection is in accurate. The results for the prediction on the accuracy of the pixels could be related to the size of the input used for both the images and the masks.

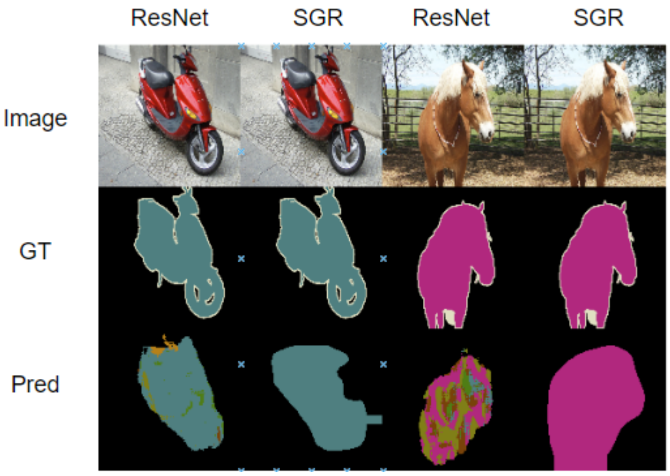

Fig. 7. Comparison of Segmentation Results

Fig. 8. The input for Symbolic Graph Reasoning Layer

Found in Fig. 8 is the input into the SGR layer, the image itself is the moped that is found in Fig. 7. Due to the input size of the images the height and width were reduced down to a 4x4 vector making it difficult to ascertain what the individual mappings contain. Additionally, it is difficult to ascertain whether this had a negative impact on the model.

Found in Fig. 9 is the output from the SGR layer due to the shapes of the tensors within the layer it was not possible to gain the feature mappings without completely changing the model. Within this image a it becomes apparent that the moped is being targeted by several of the feature mappings.

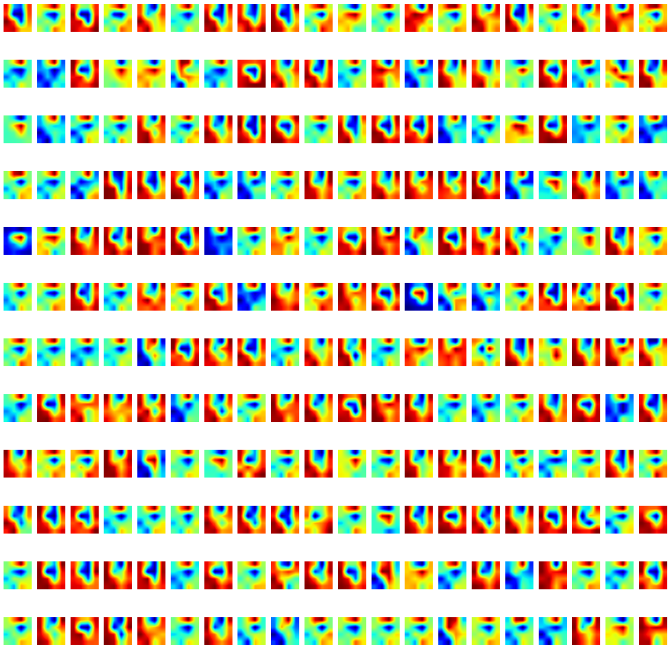

Fig. 9. The output for Symbolic Graph Reasoning Layer

VI. Discussion

This paper focused on the following the implementation of Xiaodan Liang et al.[6] for the SGR layer in the convolution network in order to validate the claims as to whether the implementation does indeed improve local features by global feature representations with symbolic nodes. The implementation was challenging some of the aspects within the paper were ambiguous in nature leaving possibility that the implementation in place is incorrect and not a valid model bringing in to question the validity of the paper. In addition the word embeddings that were done within this experiment were untrained and there was no trainable weights placed on it, the learning rates where also not done using ”poly” which could have negatively impacted the training.

The results themselves leave credence to the fact that the implementation is done in the correct manner, showing relatively good performance on an easy dataset, although when examining the predictions the background classes were less likely to be segmented correctly. This brings into question how well could the model do on a context dataset such as COCO-stuff, PASCAL-Context, or ADE20K. One of the goals for the paper was to investigate the feature mapping of the inner convolutions of the SGR layer but due to how it is implemented in accordance with the original it's simply not possible, although it was possible to get the input and outputs to gain some insights. The SGR layer needs to be further researched with continuing experiments on the knowledge graph, the implementation to make it possible to view the mappings, and the use on varying tasks in computer vision or NLP.

VII. Conclusion

In conclusion the SGR layer outperformed the baseline model of the U-Net ResNet-101 in terms of F-Measure and MeanIoU, although it is still unknown how crucial of a role it played in the results. It is possible the ASPP had contributed to the improvements of the model, further research is needed to investigate the how the mapping of the features works within the layer whether the size of the images would be more beneficial. The SGR layer although initially difficult to implement would seem to be able to be incorporated into any CNN.

References

[1] Liang-Chieh Chen et al. “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2018), pp. 834–848. DOI: 10.1109/TPAMI.2017.2699184.

[2] Yanming Guo et al. “A review of semantic segmentation using deep neural networks”. In: *International Journal of Multimedia Information Retrieval* 7.2 (June 2018), pp. 87–93. ISSN: 2192-662X. DOI: 10.1007/s13735-017-0141-z.

[3] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[4] Kaiming He et al. “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.9 (2015), pp. 1904–1916. DOI: 10.1109/TPAMI.2015.2389824.

[5] Xia Li et al. “Spatial Pyramid Based Graph Reasoning for Semantic Segmentation”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 8947–8956.

[6] Xiaodan Liang et al. “Symbolic Graph Reasoning Meets Convolutions”. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, pp. 1858–1868.

[7] Shervin Minaee et al. “Image Segmentation Using Deep Learning: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.7 (2022), pp. 3523–3542. DOI: 10.1109/TPAMI.2021.3059968.

[8] Muhammad Shafiq and Zhaoquan Gu. “Deep Residual Learning for Image Recognition: A Survey”. In: *Applied Sciences* 12.18 (2022). ISSN: 2076-3417. DOI: 10.3390/app12188972. URL: https://www.mdpi.com/2076-3417/12/18/8972.