Scale-Aware Cascading for Semantic Segmentation

Salim Ahmed Ali and Dr B G Prasad
Department of Computer Science & Engineering, B. M. S. College of Engineering, Bengaluru, India

salimahmed.scs19@bmsce.ac.in, bgprasad.cse@bmsce.ac.in

Abstract. Semantic segmentation is an important technology commonly used in medical imaging, autonomous driving vehicles, and backgrounds for virtual meetings. Scale Aware approaches have become the standard when it comes to the semantic segmentation domain of Machine Learning. Multiple image scales are passed through the network allowing the result to use the regular CNN layers such as max-pooling as well as convolution layers. Also, a cascading hierarchy of attention has been shown to improve the accuracy of models for such segmentation tasks. The combination of both these approaches has been shown to greatly improve the accuracy of such models. A side effect of using the cascading approach is that the model turns out to use less memory in comparison to previous approaches. Auto-labelling engines are also helpful in generalizing the model further. The cityscapes dataset used here is a useful data bank as it consists of a myriad of situations where the model can be trained and tested on. This paper presents the tested results of such a segmentation model and incremental modifications to the model pipeline to understand and improve upon the existing architecture.

Abbreviations. CNN - Convolutional Neural Network, IoU - Intersection over Union, ResNet - Residual Network, mAP - mean Average Precision, DCNN - Deep Convolutional Neural Network, ASPP - Atrous Spatial Pyramid Pooling, FCN - Fully Convolutional Network, ReLU - Rectified Linear Unit, IoT - Internet of Things

1. Introduction
Common computer vision problems related to scene understanding [1] are image classification, object detection, and segmentation. Image classification is the simple task of asking whether a specific object exists in the image frame or not. Object detection is to envelop different classes of objects with bounding boxes (dog, car, road, etc). Semantic segmentation goes a step further by asking whether a given pixel belongs to a set of N classes. These classes could be objects like roads, lanes, cycles, vehicles, people, buildings, etc. The result of semantic segmentation is that the objects are enclosed by exact contour boundaries. Semantic segmentation plays a very important role in the development of robotics, self-driving cars, etc.

Another form of the semantic segmentation problem is that of Instance Segmentation. Instance segmentation associates itself with distinguishing between each object belonging to the same class (e.g., vehicle_1, vehicle_2, …., vehicle_N based on a class of Vehicles).
In Figure 1, we can see a representation of the classical scale-aware model. The model consists of at least two scaled input image representations (e.g., 0.5x and 2x). The green bounding circle represents the sensitivity of the model to decipher at that scale. For example, at a scale of 0.5x, the larger person object is contextually better understood by the model in comparison to the smaller person object. The opposite can be seen at the 2x scale where the smaller person object is better understood for the segmentation model. These scaled images get passed along the model to CNN components which is generally a resnet_50 model. Together with this, a score map component generates a score for the scale s. This score map is brought to the resolution of the image with the help of bilinear interpolation. The attention model is trained to understand the scaled features by their contextual importance. The result of the element-wise multiplication is the semantically segmented mask for the class. In our case, we have shown a single person class to display the characteristics of the scale-aware model.

It has been shown that scale-aware segmentation models are found to outperform regular segmentation models with the use of multiple scaled images of the input raw image. Higher scale images are shown to produce better fine detail prediction. In comparison, lower-scale images are better at predicting larger contextual data. This can be seen in the implementation of various models such as DeepLab [2] which uses Atrous convolutions to achieve a similar result. Atrous convolutions intelligently use the contextual information available to them in the scene with the use of dilations. It is known that a dilation rate of 1 is a normal convolution whereas any dilation rate >1 is considered as a truly Atrous convolution.

Since multi-scale contextual data is required for the predictions, the negative effects of using a single input image are minimized. These scaled images are passed onto a pooling layer. This is generally found out to be either one of max-pooling or average-pooling. The max-pooling approach has been found to lose a lot of information as for every pixel, only one of the N scales present is chosen. A better approach would be to use a weighted combination of the different scales. The other approach that has been used is that of average-pooling. This also suffers from a problem where we merge large-scale contextual information with smaller-scale information. If the class object is better predicted at a larger scale, then average-pooling will merge this data with the lower scale data and generate an average. This generally produces a better result in comparison to max-pooling, but some contextual information is lost as we are averaging better scales with worse ones.
Figure 2. Scale-Aware cascading attention used at the training stage.

Figure 3. Scale-Aware cascading attention structure used at the Inference stage.

To address the above-stated issues, an attention-based approach has been proposed by Tao [3] where the attention model is combined with the scale-aware approach while following a cascading layering. In this paper, we will be using the words hierarchy and cascading interchangeably. This approach has been found out to be well suited to the segmentation scene and also handles the issues which the max-pooling and average-pooling approaches had. The model has been shown to lower the memory load by up to 4x which immensely helps during the training phase.

In figure 2, we see the scale-aware cascading structure inspired by the work of Tao [3] which uses two scales for the training stage. In our tests, we vary the count of scales available using a python module. The attention and segmentation components are used whenever an extra scale is added to the cascading structure. The last cascade layer uses just the segmentation component. The element-wise multiplication and addition are similar to the classical scale-aware model.

The scale-aware cascading attention model provides multiple scaled images to the training phase. Each scaled image is passed through a pipeline consisting of a segmentation and attention model. The segmentation model is a regular D-CNN meant for the segmentation task. The cascading is such that the lower scale attention component is allowed to choose how much of the higher scaled image is to be considered in its stage. The largest scale image at the end of the cascading layers, as it is not involved in choosing the attention component for a much larger image, drops the attention stage. This can be seen in figure 2 and figure 3 as well. Cascading layers can also be easily set up to utilize system memory sparingly in comparison to existing models.

2. Literature Survey

Scale-aware [4] model considers a methodology that uses an attention-based approach that understands to weigh the scale-aware features at every pixel location in the image. A major advantage of this approach is that it provides higher performance than in the case where a single-scale input image is provided. Application of the attention model to the multi-scale pipeline allows for the model to generate better results in comparison to the regular max-pooling and average-pooling results. The model also shows that adding further supervision allows the model to improve on its predictions for each image size.

The Hierarchical Multi-Scale Attention model [3] proposes an attention-based mechanism that allows for the merging of differing scale predictions. This model uses an attention mechanism that follows a cascading structure. This in turn allows the model to train faster. It reduces memory consumed when utilizing the model for training in comparison to existing models. It also allows us to train with larger crop sizes(scales) which leads to greater model accuracy.

Self-training with noisy student model [5] improved on current models by increasing the noise component in the student model, resulting in the name NoisyStudent. This allows it to learn and improve to an extent greater than the teacher model’s knowledge. The self-training approach is a simple and effective algorithm to utilize the unlabelled datasets at much greater scales. Confirmation bias
mentioned in deep semi-supervised learning [6] uses self-training, which is known to be a classical approach in the context of semi-supervised learning tasks. It adds multiple sources of noise to the student component. It also makes sure that the student models are at the very least the same size as that of the teacher. In most implementations, the student model will have more layers than the teacher model. This shows that it is a possibility where we use unlabelled data to strategically increase both the accuracy and robustness of ImageNet models.

A common semantic segmentation model which uses a video propagation [7] technique uses a prediction-focussed approach to upscale the training input data. It does this by generating newly created training data to improve the results of current segmentation models. It demonstrates that training such models on labelled image datasets that are modified by these generated samples allows it to significantly improve upon the accuracy of the model.

Several different architectures were studied as part of this literature survey to gauge the benefit of the Multi-Scale Attention model over existing architectures.

2.1. U-Net

2.1.1. Achievements.
- The model is known to produce noteworthy results even with a small training sample set.
- Produces better segmentation maps compared to previous models.
- The model is aware of its location as well as the contextual data simultaneously.

2.1.2. Limitations.
- Lower scale features aren’t obtained correctly and result in irregularly or coarsely segmented image maps.
- For the training stage, the number of parameters is significant and thus increases the initial setup for model training.
- Skip connections are working at the same level or scale. Thus, aggregation at different scales isn’t considered [8].

2.2. Multi-Scale Attention [4]

2.2.1. Achievements.
- Merging multi-scale (zoomed in and zoomed out) [9][10] features with an attention model increases the performance compared to max-pooling or average baselines.

2.2.2. Limitations.
- Adds redundant information usage as similar low-level features are obtained multiple times at varying scales. i.e. encoder-decoder architectures [11].

2.3. DeepLab [2]

2.3.1. Achievements.
- ASPP [12] works at multiple scales by applying the pyramidal approach when generating the feature map.

2.3.2. Limitations.
- A limitation that exists in the DeepLab architecture is that the ASPP blocks end up taking up significant chunks of memory and space on the machine in comparison to the case when we do not use it.

2.4. FCN
2.4.1. Achievements.
- Input image size need not be fixed which allows this model to handle varying input cases. This offers greater flexibility in choosing where such a model can be deployed.

2.4.2. Limitations.
- Generated results have rough and fuzzy boundaries, making it difficult to implement in some real-world use cases.

3. Methodology
The architecture for model training is of the classical training pipeline. Labelled data was collected from the Cityscapes dataset. Also, unlabelled data was obtained from other imaging sites. This dataset was passed through the pre-processing module to apply transforms, add noise, apply mean-subtraction, etc. The trained model is passed onto a scaling generation module which generates the scales based on a passed parameter. The Hyperparameters are set for the model to train on and then the model is trained. figure 4 shows the training pipeline for a two-scale training scenario. The model training is done using a tool known as runx.

Here we propose incremental changes to the cascading scale-aware attention model by testing out standard scale sizes based on the image context. The model was also tested on several cascading layer counts. Also, current auto-labelling engines using hard thresholding approaches have been used to improve on the cityscapes dataset for the model which consists of 20,000+ coarsely bounded city-based semantically segmented images. This allows us to provide improved training data for our model to learn from in comparison to the limited set of 5000 finely predicted images. The coarsely segmented images are passed through the auto-labeller and this uses a pre-trained model to predict and also generate the fine segmentation images.

The hierarchical multi-scale attention architecture [3] consists of a ResNet-50 model for the main trunk of the model. The segmentation and attention components are very similar to each other in that they both use a CNN such as ResNet-128 [13]. The difference being that the segmentation component produces an N-channel output whereas the attention component produces a single channel output. The final result is a channel depth of the number of classes to be predicted. The Semantic head consists of a 3x3 kernel filter, followed by Batch Normalization. This is followed up with the ReLU activation function. This layering repeats another time followed finally by a 1x1 convolution filter.

We test the model for differing cascading levels while keeping the rest of the hyperparameters constant. This allows us to find the model best suited for the scene being tested. It also tells us how high the model’s IoU values can be.

Formulae used:

\[
Precision = \frac{TP}{TP + FP} 
\]

\[
Recall = \frac{TP}{TP + FN} 
\]

\[
Average Precision(AP) = Area under the precision and recall curve 
\]

Here, TP = True Positive, FP = False Positive, and FN = False Negative. Average Precision can also be written as AP.

In the classical multi-scale attention approach as shown in figure 1, the attention masks are all learned at the same time for all different scales provided. This is different in comparison to the cascading approach which finds relative attention between scales that are adjacent to each other. Also, the memory usage in comparison to the classical approach is much less. Since we use scales 0.25x, 1x and 4x, the extra memory usage is \(4.25 \times (0.25^2 + 4^2)\) for the classical scale aware approach. The cascading attention
approach uses a lower memory footprint. Both the precision (1) and recall (2) values of the model meet baseline requirements. The AP (3) value of the model is also within the threshold limits.

Figure 4. Architecture Pipeline for model training.

4. Experimental Setup
The experimental setup consisted of two P4000 Quadro GPUs each specified with 6GB of Video RAM. It includes a RAM specification of 8GB. The processor used was an Intel-based core i5 7400 processor. The entire pipeline as shown in figure 4 was set up on this benchmarking machine.

The dataset used here is the cityscapes dataset which consists of 5000+ images of real-world vehicle cam situations with dense pixel annotations. Together with this, another set of 20,000+ coarse annotations are present in the dataset to train our segmentation model. These are colour-coded annotations for 19 classes such as Vehicles, Pedestrians, Roads, Traffic signs, etc. Together with this dataset, we have included varying images from other general web sources as well as online segmentation training data banks. To add to this, labelling applications such as LabelMe were used to produce edge case labelled data so that the model is better generalized and able to handle most edge-case situations. The data pre-processing steps included here are the standard ones. These include scaling of the image to the ResNet model input size of 255x255 followed by normalization to a floating-point representation.

The test iterations were conducted 3 times for each iteration to produce an average value and remove any inconsistencies that may exist in the different runs of the experiments.

Another step added to the pre-processing was to generate the images of different crop sizes, varying dimensions, zoom levels, as well as adding noise to the input raw images to generalize the model. A scaling module is introduced to generate the different scales for the model to train on. Here we use the scales of 0.25x, 1x, and 4x to train the model.

5. Results
For the Cityscapes dataset, the number of epochs was set to 35 and the learning rate was set to 0.05. These parameters were kept constant while the number of cascading layers was varied. A module was developed to increase the number of cascading layers for the generation of the logs.

Table 1. Predicted results when varying the number of cascading/hierarchical layers.

| Test Iteration | Cascading Layers | Batch Size | Training Duration. (hours) | mAP |
|----------------|------------------|------------|-----------------------------|-----|
| 1              | 2                | 1000       | 13                          | 0.65|
| 2              | 100              | 1000       | 14                          | 0.72|
| 3              | 1000             | 1000       | 13                          | 0.68|

Initially, the number of layers is set to a low of 2 cascading layers at the inference stage as seen in table 1. The segmentation map generated as a result can distinguish the different features but has a difficult time isolating some of the fine details. Also, the larger features such as the tree in the background aren’t correctly segmented. As we increase the number of layers to 100, we see a significant
increase in the segmentation map of finer features. The lamppost and also the person riding the vehicle are correctly predicted. Increasing the number of layers further to 1000 layers starts to produce unwanted artefacts and the predictions are generally worse. We find that a cascading layer count of 100 is found to be close to the optimal value for the cityscapes dataset. This Standard Deviation for predicted value can be $\pm 10$. This testing shows us that increasing the hierarchical layers up to a certain point is helpful, and then the results diminish when adding further layers. The memory usage was within the expected range of the Hierarchical attention model. The results produced are shown in table 2. Also, the IoU values increase by 0.3% with the usage of the cascading model. The auto-labelling step allows the model to increase performance significantly by 0.7%. This is known to add a 1.0% increase over the baseline values.

The tests conducted here could be further extended to more powerful GPU machines to replicate and improve the results. The pre-trained model is known to produce a mIoU of 84 for three scales.

| Hierarchical/ Cascading Layers | Segmentation map Result |
|-------------------------------|-------------------------|
| 2                             |                         |
| 100                           |                         |
| 1000                          |                         |

**6. Conclusion**

Semantic Segmentation is an important computer vision problem that has its uses in medical image diagnosis, self-driving cars, scene cropping, etc. The solutions that semantic segmentation has come to solve have made great contributions to the field of Machine Learning and Computer vision.

In this paper, we have discussed the multi-scale attention model on how it has improved with the use of applying the cascading approach. The cascading attention model allows for the reduction of memory usage and also ends up generalizing the model for the use cases thrown at it. Altering the number of hierarchical layers allows us to find a balance between the number of layers as well as the memory consumption for the model to train and perform inference.

Moreover, auto-labelling engines have also helped improve the performance of the models by providing better training data. This has led to the coarsely labelled data being better used in training the model in its augmented form rather than being directly used which would have reduced the model performance.

Furthermore, modifications to the attention model can be done as the hierarchical attention model uses the same CNN architecture as the segmentation model.

Lastly, the use of hardware-accelerated deployment of the model will be instrumental in the deployment of such architectures on low power, high-performance IoT devices such as edge devices.
References

[1] Everingham, Mark, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. "The pascal visual object classes challenge: A retrospective." International journal of computer vision 111, no. 1 (2015): 98-136.

[2] Chen, Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." IEEE transactions on pattern analysis and machine intelligence 40, no. 4 (2017): 834-848.

[3] Tao, Andrew, Karan Sapra, and Bryan Catanzaro. "Hierarchical multi-scale attention for semantic segmentation." arXiv preprint arXiv:2005.10821 (2020).

[4] Chen, Liang-Chieh, Yi Yang, Jiang Wang, Wei Xu, and Alan L. Yuille. "Attention to scale: Scale-aware semantic image segmentation." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3640-3649. 2016.

[5] Xie, Qizhe, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. "Self-training with noisy student improves imagenet classification." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10687-10698. 2020.

[6] Arazo, Eric, Diego Ortego, Paul Albert, Noel E. O’Connor, and Kevin McGuinness. "Pseudo-labeling and confirmation bias in deep semi-supervised learning." In 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1-8. IEEE, 2020.

[7] Zhu, Yi, Karan Sapra, Fitsum A. Reda, Kevin J. Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. "Improving semantic segmentation via video propagation and label relaxation." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8856-8865. 2019.

[8] Zhou, Zongwei, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. "Unet++: Redesigning skip connections to exploit multiscale features in image segmentation." IEEE transactions on medical imaging 39, no. 6 (2019): 1856-1867.

[9] Florack, Luc, Bart Ter Haar Romeny, Max Viergever, and Jan Koenderink. "The Gaussian scale-space paradigm and the multiscale local jet." International Journal of Computer Vision 18, no. 1 (1996): 61-75.

[10] Arbelaez, Pablo, Michael Maire, Charless Fowlkes, and Jitendra Malik. "Contour detection and hierarchical image segmentation." IEEE transactions on pattern analysis and machine intelligence 33, no. 5 (2010): 898-916.

[11] Sinha, Ashish, and Jose D dolz. "Multi-scale self-guided attention for medical image segmentation." IEEE journal of biomedical and health informatics 25, no. 1 (2020): 121-130.

[12] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Spatial pyramid pooling in deep convolutional networks for visual recognition." IEEE transactions on pattern analysis and machine intelligence 37, no. 9 (2015): 1904-1916.

[13] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.