Self-attention and shape context networks to improve robustness on semantic segmentation tasks

G049 (s2259628, s2239389, s1751472)

Abstract

In this work, we explore semantic image segmentation in a supervised regime to evaluate and improve the robustness of the state-of-the-art DeepLabv3+ architecture across the PASCAL Visual Object Challenge (VOC) 2012 data set. Through experimentation it is found that the baseline model (DeepLabv3+) performance degrades for the image augmentations such as gaussian blur, cutout, and scaling. We propose three models to combat these perturbation within images. The proposed variants are deformable convolution, hierarchical attention, and hierarchical attention paired with a contour detection model. Our proposed networks were able to outperform the baseline MIoU score by 2%, and were more robust to the aforementioned perturbations, evaluated using the Corruption Degradation (CD) metric. We found that our proposed deformable convolution model was more robust to scaling ($CD = 0.9$) compared to the baseline, and additionally, both variations of the hierarchical attention model were robust against perturbations such as cutout and Gaussian blur ($CD = 0.95$).

1. Introduction

Semantic image segmentation is the task of assigning each pixel in an image with the label of the objects present at that location (Long et al., 2015). It is different from two other prevalent tasks in computer vision; classification, which involves detecting whether an object is present in an image, and detection, which involves labelling the bounding box of each object present in an image, in that this tasks requires identifying objects at each pixel.

In recent years, great strides have been made in improving the performance of semantic segmentation through the use of deep learning methods such as fully convolutional neural networks (Long et al., 2015), however these models require access to large amount of pixel-level annotated data, which is known to be notoriously difficult to obtain. Moreover, they have been shown not to be robust against clutter, perturbations, or small changes in input images (Kamann & Rother, 2020a; Vasiljevic et al., 2016; Hendrycks & Dietterich, 2019). Semantic segmentation, however, requires a high level of accuracy as it requires prediction at a pixel level, and errors can lead to catastrophic issues in applied situations.

In this project, we aim to explore the robustness of the DeepLabv3+ semantic segmentation architecture when Gaussian blur, cutout (DeVries & Taylor, 2017), and rescaling based perturbations are applied to images for segmentation, and analyse the degradation in performance using the mean intersection-over-union (MIoU) and corruption degradation (CD) metrics for different intensities of perturbation, and build models that are more robust against these perturbations. We finally provide a comparison of our approach to the baseline in terms of the degradation in performance with perturbations, and analyse aspects that lead to this robustness.

1.1. Motivation

The motivation of this project is to explore the different perturbations to which the state of the art semantic segmentation models are not robust to, and to improve the generalisation on these perturbations. Semantic segmentation has several applications to real-world problems, ranging from autonomous driving (Çağrı Kaymak & Uçar, 2018) and road sign detection, to biomedical cell data analysis, which requires high precision and robustness to noise, clutter and irrelevant information. Failure to correctly segment can directly lead to harmful consequences (for example, occlusions in images for self driving cars might give wrong predictions leading to accidents). Hence ensuring the robustness of semantic segmentation models is crucial.

The authors of the paper (Kamann & Rother, 2020b) tried benchmarking the robustness on the state of the art model used for semantic segmentation (DeepLabv3+). Taking this into consideration we try to explore some perturbations that were not explored such as cutout and the effect of scale, in combination with Gaussian blur, which showed reduction in performance((Kamann & Rother, 2020b). Exploring these augmentations, along with common noise perturbations, will place us in a better position to analyse how well the semantic segmentation models take contextual information into account when faced with clutter or unencountered distortions.

1.2. Research Questions

We propose two new architecture changes in the state of the art model; a deformable convolution based model, an attention based model. These models integrate high and low-level semantic features using contour information, and use spatial, contextual and correlation based information to improve robustness to perturbations such as occlusion and

noise. We further evaluate our proposed models in terms of the average degradation of the mean intersection-over-union (MIoU) accuracy metric, evaluating the MIoU for each class prediction on the augmented data set, aiming to show the improved robustness of our proposed architectures. We are interested in understanding how incorporating spatial coherence, contour and correlation based information makes the model robust to noise and clutter.
In section 2 we discuss the data set used for the experiments (the PASCAL VOC 2012 data set), including the details regarding size, augmentations used, and parameters required for replication. We then discuss our proposed solutions in section 3, with details regarding the proposed remedying architectures, and why we choose to explore these solutions. The implementation and specifications are explained in section 4, and our results and findings are discussed in section 5, giving quantitative details regarding experimentation.

2. Data Set and Task

The baseline model for our tasks is the DeepLabv3+ (Chen, 2018) semantic segmentation architecture. This model is the most recent iteration of the DeepLab (Chen, 2017) architecture series. It adds an encoder-decoder module to further improve segmentation accuracy. Details about the model is explained in 3

The metric we use is the class-level prediction at each pixel, evaluating the mean intersection-over-union (MIoU) over all classes. The MIoU is calculated by taking the mean of the class-wise correct prediction against all predictions (intersection-over-union); the ratio of true-positive classifications to the sum of the true-positive, false-positive, and false-negative classifications. (TensorFlow, 2022b)

$$MIoU = \sum_{c} \frac{TP}{TP + FP + FN} \tag{1}$$

The PASCAL VOC (Visual Object Classes) 2012 (Loshchilov & Hutter, 2017b) data set is being used for the task at hand; which is to probe and evaluate the robustness of the baseline DeepLabv3+ architecture across different augmentations and perturbations applied to the validation set of the same data set, as well as of our proposed remedying architectures. Using cutout, Gaussian blur and rescaling to augment and perturb the PASCAL VOC validation data set, each architecture will perform semantic segmentation on the clean data, as well as the augmented data, and the MIoU metric and its degradation on the augmented data is calculated. The results will be used to determine the robustness of the baseline model to perturbation, as well as the extent to which the remedying architectures improve the robustness in MIoU. Since using the MIoU metric alone does not gives an indication of the degradation in performance, we also use another metric known as mean corruption degradation (Kamann & Rother, 2020b). Together these two metrics help us analyse the robustness of a model.

Using the same method as DeepLabV3+, the images and labels are converted to the TensorFlow TFRecord format and are input to the model. (TensorFlow, 2022a) These records, used to store sequential binary records, were prepared using the TensorFlow data set tools on GitHub (TF).

The PASCAL VOC data set (Loshchilov & Hutter, 2017b) is as a standardised data set of images for object recognition in the PASCAL Visual Object Challenge (VOC); a challenge where the data set is made public, with an annual competition and workshop to obtain the best possible image segmentation accuracy. The data set is accompanied with a ground truth segmentation map for each image, containing the label information of each object present in each pixel with a label 0 for background. The data set is created from Portable Network Graphic images sourced from Flickr (Loshchilov & Hutter, 2017a), consisting of a training and validation data set, each having 20 classes, as well as an additional background class. The most recent and most developed data set, the 2012 release, is being used for this task. This consists of 1,464 train images, 1,449 validation images and 1,456 test images. Data augmentations are done on both train and validation images, which gave us around 10,582 images for training. Because the data set was part of a challenge, the test set was not made public. As a result, to evaluate our performance on perturbed images we used a part of the validation set as our test set.

For this task, the data is pre-processed by subtracting image means first, and changing labels from colour maps to integer labels at each label, representing the object present (for instance, $0 - 21$ for PASCAL VOC where 20 classes of objects are present with one label for background). From here affine transformations including cropping the images between scales $(0.5, 2)$ and horizontal flipping are applied in order to augment the data set. The same augmentations are applied to the image labels as well. All the models are then trained and tested over this augmented training data set.

3. Methodology

In this section we firstly describe the model that we use to evaluate the robustness of the state of the art segmentation models, followed by an explanation of our proposed methods which improves upon its robustness.

3.1. The DeepLabv3+ Encoder-Decoder Architecture

We use the DeepLabv3+ model (Chen, 2018) in our experimentation as a baseline to perform pixel level segmentation using its encoder-decoder architecture(fig 1). The the encoder is a deep convolutional network, which processes images through a backbone convolutional network architecture to allow spatial resolution feature maps, then applies dilated spatial pyramid pooling over these feature maps.
Dilated spatial pyramid pooling is a concatenation of the features obtained by applying a dilated convolution (Yu & Koltun, 2015) operation at different rates at each location of the feature map. The dilated convolution at rate r is effectively a convolution operation, where there is a gap of

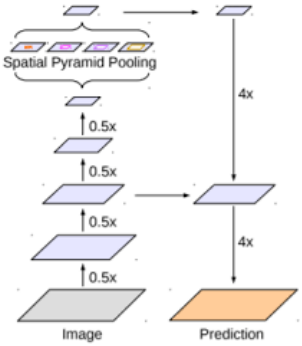


Figure 1. DeepLabv3+ Architecture (Chen, 2018)

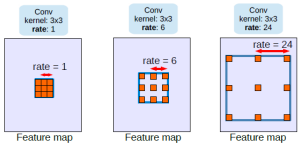


Figure 2. Atrous Convolution (Chen et al., 2017)

r locations between the locations at which the filter weights of convolution is applied. It can be visualised (fig 2) as a convolutional operation which is scaled by r, and filled with holes at intermediate locations to compensate for the absence of weights at those locations. This can be used to aggregate multi-scale contextual information.
In the original paper, the authors use rates of 6, 12, and 18. An image level global pooling is also concatenated with the features obtained from the dilated spatial pyramid, which just concatenates an average of the encoder features to each location in the feature map. These operations help in aggregating contextual information at each location. The decoder upsamples the features back to the original resolution using a bilinear interpolation (Fadnavis, 2014) based method, which samples values using spatial information and generate samples for a larger feature map from a smaller one. Their proposed method performs this upscaling of each feature by 4, and concatenates the obtained features with a low-level convolutional map of the encoder at each location. This is then repeated by upsampling each feature by 4, finally obtaining a feature map matching the original image size. Pixel level class prediction can then be performed on the feature map using the cross-entropy loss function. The robustness of the DeepLabv3+ model is evaluated using different data set perturbations, described in detail in the experiments section. We use the same method for the decoder block in our proposed solutions, with some variations.

3.2. Hierarchical Spatial Attention for Coarse Grained Patch Level Object Prediction

Spatial attention is an effective method for capturing long range interactions between different locations of feature maps, overcoming the limitations of the convolutional operation caused by their limited receptive field sizes (Fu et al., 2019). The authors of the paper (Dosovitskiy et al., 2020) proposed an architecture for hierarchical attention

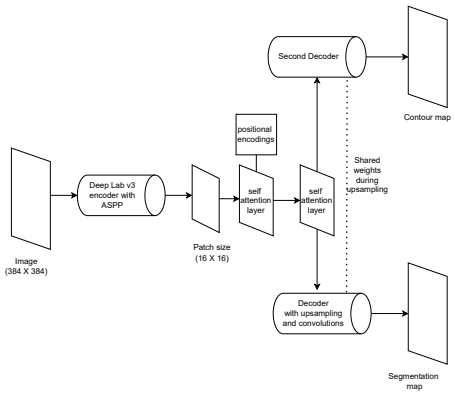


Figure 3. Attention Based model with contour information

for image classification tasks. We employ a variation of this method to predict the objects that are present in each 16x16 patch of an image (obtained when passed through the encoder part of the model), modifying the original objective of classification for the whole image, to per-patch classification, also learning contextual features at each location in the feature map. To do this, we first obtain the features from a deep convolutional network. One such variant often used as a baseline across many papers is the Residual Network (He et al., 2016). Extracted features from the penultimate layer are obtained, providing a feature map $E \in \mathbb{R}^3$ of size $\frac{H}{16} * \frac{W}{16}$, with $C = 256$ channels. The feature maps obtained after these two operations are of size $H' = \frac{H}{16}$, and $W' = \frac{W}{16}$, C, where H and W are the height and width of the original image, and C is the number of channels per location. Thus each resultant location in this map has a receptive field of size 16x16. The features are then processed by a single layer convolutional encoder, which reduces the number of channels from C to R, and passes them through a spatial attention layer (as proposed in the transformer architecture) to predict objects present at each patch. Supposing the features obtained from the last layer of the residual network after processing through the dilated pyramid pooling layer are of size $E := H' * W' * C$, we firstly pass these features through a convolutional layer of kernel size 1x1 with R channels, where $R < C$, to obtain a feature map of size $F := H' * W' * R$. We observe that reducing the number of channels with a final convolutional layer is essential. This is due to the computational expensive of spatial attention, requiring F operations to be performed, scaling with R. A positional encoding P is then added at each location in the feature map to encode the information about the location of each patch, obtained by passing the relative (x, y) coordinate through a single layer network, adding to each location as a vector, generating the features $E + P = e_1 + p_1, e_2 + p_2, e_3 + p_3, ...e_{H'*W'} + p_{H'*W'}$, where $E = e_1, e_2, ..., e_{H'*W'}$ and $P = p_1, p_2, ..., p_{H'*W'}$. This is then repeated to generate three such feature maps, each one processing the original features from the encoder and adding positional encoding P. The resultant feature maps are $K, Q, V \in F$. These encoded feature maps are then passed through the spatial self-attention layer which captures cross correlation between features at each location to

generate new feature maps, according to following equation.

$$A = \text{softmax}(K * \frac{Q}{\sqrt{R}}) * V \quad (2)$$

Finally, the features obtained from A are added with the original feature map E to obtain the final feature map $E' = E + A$. This addition acts as a skip connection to allow gradients to flow through the network, mitigating the vanishing gradient problem. This procedure is repeated 2 times to generate the final feature map of size F, in contrast to the vision transformer (Dosovitskiy et al., 2020), which performs it for $L = 8$ times. This final feature map is passed through a multi-class classification layer to predict all objects present in the corresponding patch. The labels of the objects present in each image are obtained by first average pooling the ground truth segmentation map converted to a 21 dimensional one-hot vector over each patch, and then clipping the value for the one hot label vector to 1 for each object where the value is greater than zero, otherwise we set it to 0. Thus, the model performs a category level prediction on each 16x16 patch of the original image, while also capturing long range interactions and contextual information, further helping the segmentation tasks.

3.3. Hierarchical Supervision to the Decoder

The encoder features obtained as described in the previous section are then upsampled twice, each time by 4x4 using bilinear interpolation (Fadnavis, 2014). Thus the final upsampled feature map is of size 16x16 times the size of the features produced by the encoder, and is of the same size as the original image. In total the loss is added at two stages in our network, the first one being the multi-class loss described in section 5.1, and the pixel level cross entropy loss on the feature maps obtained after the final upsampling operation. Also, we are passing the predictions of patch level coarse gained prediction to the first upsampled feature map. Hierarchical supervision has shown improvement in many methods that use the deconvolution operation for upsampling (Fu et al., 2017). We instead eliminate the deconvolution operation and use bilinear upsampling, followed by convolution. The multi-level supervision used to predict the labels in a hierarchical pyramid at multiple scales has shown to be effective in (Ghiasi & Fowlkes, 2016). This method has also shown to improve the ground truth predictions, by allowing the model to combine both coarse-grained feature prediction, which is more smooth but does not predict well near contours and boundaries due to its smoothness, and the fine-grained predictions, which predicts well near the contours but is more susceptible to noise and irrelevant information.

3.4. Contour and Foreground Background Prediction as an Auxiliary Task

In addition to predicting the objects present at each pixel, we extend the task by adding an additional objective predicting the locations which are at the boundaries of the objects, and if they are in the foreground (on an object) or in the background. This is done by creating shared layers in the encoder, and a separate decoder map for the foreground/background and contour detection. We create shared convolutional layers and pass their output to two decoder pathways, consisting of upsampling and convolution operations, one of which performs the identification of class level prediction, or the original segmentation task, and the other performs an auxiliary foreground background and contour prediction task. Thus, supposing that the newly created layers are $c_1, c_2, ..., c_n$, the outputs from the base ResNet encoder at intermediate layers $b_1, b_2, ..., b_n$ are passed to them so that the outputs of these layers are $c_1 + b_1, c_2 + b_2, ..., c_n + b_n$, and the output from the final layer, i.e., $c_n + b_n$ is passed to a separate upsampling based decoder for predicting foreground, background or contours. For generating labels for this task, we process the ground truth segmentation labels for each image to generate new labels which contain the integers 1,2,3 denoting background, foreground, and contour respectively. To this end, we iterate over each image during the pre-processing step, and compare the label at each location for the ground-truth labels corresponding to each image with its left-shifted label map, shifted 1 pixel to the left, and down-shifted variant, which is shifted up by 1 pixel. If the label is different from the version that has not shifted, then we assign a label 3 to that pixel denoting contour, otherwise we encode it as 1 or 2 denoting background or foreground, depending on whether that location has a classification label greater than zero to denote object presence. The separate decoder head is used to predict the corresponding label from $1, 2$ and 3, as obtained using this procedure, using a similar upsampling and prediction procedure as described in section. Adding this as an auxiliary task allows the spatial attention layer to take information about the object contours and the foreground and background regions, since the attention operation is performed on the joint encoder feature map for the segmentation, and the contour/foreground-background task.

3.5. Scale Invariant Deformable Convolutions

The convolutional operation is a good method to learn location independent features from an image, but these operations do not have the inductive bias to handle scale and shape deformation variation of objects. To this end, we design deformable convolution operation which firstly performs a 1x1 convolution of a large number of channels over the whole feature map, and then perform a max pooling operation at different scales of $4, 8, 12$ at each location on the obtained features, before performing another pointwise convolution on these features. This multi-scaled max pooling followed by 1x1 pointwise convolution can capture the feature statistics even if they are far away in the images. For example if the parts of an object are deformed, but at different scales than usual. The max pooling at multiple scales will only find the activation of a feature at the spatial range where its activation is maximum, hence allowing some flexibility in where the features can be present. We added this operation after the spatial pyramid operation on

the features obtained from decoder. This method is similar to the one proposed in (Dai et al., 2017), however instead of letting another convolutional feature map learn the deformation of filter weights location, we let the max pooling operation capture it at different scales, removing this restriction and adding flexibility, allowing deformation of the learned kernels.

4. Experiments

In all our experiments, we use ResNet-101(He et al., 2016) as backbone which was pre-trained for performing image classification on the ImageNet data set (Russakovsky et al., 2015).

4.1. Training Protocol

Environment - All the experiments were run on a single GPU Tesla T4 available on Google Colab.
Data Set - As mentioned in the data set section we used augmented data set containing 10,582 images derived from train data of 1464 images. This augmented data set contains augmentations including random scaling of images by factors in range $(0.5, 2)$, and random left and right y-axis mirroring during training process. The original image size used was 513x513 by the authors (Chen, 2018), but due to some issues while running our models on GPU we were able to use image size of 384x384 at most. We loaded the data set from the TFRecords that were created as described in and subtracted image means from the images only
Regularization - Following the original training regime mentioned in (Chen et al., 2017), we also added L2 regularization with weight decay of 0.0001 for all convolution layers and dense layers.
Loss function and optimizer - As for the loss function, Sparse Categorical Cross Entropy was used as there are more than one object in the images. We used the Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9, and a poly learning rate policy (3), with power 0.9 and initial learning rate 0.006. Power $\in (0, 1)$ is a hyperparameter controlling learning rate decay, where power magnitude is proportional to decay rate.

$$new\_lr = old\_lr\left(1 - \frac{iteration}{max\_iteration}\right)^{power} \tag{3}$$

Training/Evaluation details- All the models are trained for 54,000 epochs, with a batch size of 8 for training and validation set as per the original training regime. Based on the validation mIoU score, we saved the best model at each epoch. Since the data set is hosted as part of the challenge we were not able to apply perturbations on test set since we were only able to submit our model to the server to get evaluated on the test set and the test labels did not have these perturbations applied. We instead report the results on the whole validation set of 1,449 images containing almost same number of test set (1,456 images) for all the experiments. This is observed in other research papers as well (Hendrycks & Dietterich, 2019).

4.2. Perturbations

We now describe the different perturbations and augmentations that we applied over the DeepLabv3+ model, and our proposed variants to test their robustness against these. We apply these augmentations at three severity levels on the validation set of PASCAL images where ground truth segmentation maps were available, and evaluate two metrics to evaluate and compare their robustness that are described below.
Cutout - A regularisation technique used to augment data sets through occlusion, forcing models to rely heavily on context. However, instead of regularisation, we apply it as a perturbation over images to test the robustness of the models against occlusion. This emphasis on context removes focus from identification of frequent identifying features, through the removal of a contiguous region within input images. For our cutout application, we applied it over patches of sizes of $0.1 * H$x$0.1 * W$, $0.2 * H$x$0.2 * W$, and $0.3 * H$x$0.3 * W$, where H and W (384 in our case) denote the height and width of the image respectively. These patches were cut from a location where an object was present, and filled with zeros, thus darkening out the patch and removing image contents. Since the application of cutout removes the actual content from a part of the image, the ground truth label at that part must also be changed. As such we apply the same cutout at the same location in the corresponding ground truth label. Using each of these three variants of the patch sizes for cutout, we compare the performance of the baseline model and our variants by comparing the average MIoU and mean corruption degradation (Kamann & Rother, 2020b) over each size of the cutout patch on the validation data set.
Gaussian noise - Here we add a value sampled from a Gaussian distribution of mean 0 and standard deviation $\sigma \in (4.77, 6, 8)$ to each pixel of the image. We apply this noise over all the pixels of an image sampled from the same Gaussian distribution and then test the robustness of all the model against it. We experiment with the standard deviations $\sigma = 4.77, 6$, and 8 and then calculate the average MIoU and the mean corruption degradation.
Rescaling - To check the robustness of model against different scales and sizes of the objects, we take crops of ratios 0.4, 0.5, 0.6 from several locations of each image as well as the ground truth label map and resized them to the original image size 384 X 384. Then we performed the evaluation of our models on these rescaled images on the validation set. The average MIoU performance and the mean corruption degradation is evaluated and compared for all the models and the results as obtained are compared in the tables 1 and 2.
Evaluation metric for performance measure on perturbed images: The mean corruption degradation (CD) error can be used to test the robustness of a model. It is obtained by calculating the average drop in performance, rather than the vanilla IoU score. To evaluate it, we follow the procedure as mentioned in (Kamann & Rother, 2020b), and firstly calculate the degradation score $D = 1 - MIoU$ and average it over different severity levels of all the aug-

Model	Clean images	Gaussian blur	Cutout	Rescaling
DeepLabv3+	72.99	71.5	71.2	66.7
Deformable convolution	73.1	72.28	71.4	**67.5**
Hierarchical Self-Attention	73.74	**72.8**	**72.26**	65.4
Hierarchical Self-Attention (with Contour Prediction)	72.92	*71.8*	71.1	63.6

Table 1. A comparison of average MIoU of the DeepLabv3+ encoder with the three proposed variants. The average is taken at different severity levels of the corruptions and is calculated for the clean image, Gaussian blur, cutout, scaling and rotations as described in the experiments.

Model	Gaussian blur	Cutout	Rescaling
DeepLabv3+	1	1	1
Deformable convolution	**0.95**	0.99	**0.97**
Hierarchical Self-Attention	**0.95**	**0.96**	1.04
Hierarchical Self-Attention (with Contour Prediction)	**0.98**	1	1.09

Table 2. Mean CD for each model per perturbation, averaged over the different severity levels.

mentations. Then we calculate the same score for a reference model and divide the average with that of the reference model. This gives normalized degradation score which can then give better insights for comparing model robustness.

$$CD_c = \frac{\Sigma^3_{s=1,c} D_{s,c}}{\Sigma^3_{s=1,c} D_{ref,s,c}} \tag{4}$$

In above equation (4), s denotes the severity level of the perturbation and c the corruption type. $D_{ref,s,c}$ is the degradation of the reference model for the given perturbation and severity level which in our case is the baseline DeepLabv3+.

4.3. Experiments on DeepLabv3+

The main aim of this experiment is to recreate the baseline model, and train it from scratch so that the results are homogeneous throughout all the other experiments. In order to do that we used publicly available code which was published by the authors of the original paper (Chen, 2018). We were not able to decode some of the specifics regarding the experiments from the original paper so we used this GitHub link as an extra reference here, as the author was able to recreate the baseline accuracy following almost the same training regime as the original paper(Chen, 2018), but in detail.

In the original paper they used different architectures on different data sets, and got good results when a deeper architecture model is used. Since deeper architecture requires huge computational power to scale, in the model for our experiments we used ResNet-101(He et al., 2016) as backbone architecture trained on ImageNet 1k data set (Russakovsky et al., 2015). (It is mentioned in the paper that when ResNet-101 was used as backbone for encoder part of the model, it gave an MIoU score of 0.77.)

As mentioned in section 3 we used dilation rates of $1, 6, 12, 18$ for Spatial Pyramid Pooling layers to improve the performance of the model and help capture better context from the images. The architecture used is depicted in figure1. The training protocol was then followed and the results were reported in the table 1. It is observed that due to reducing the image size by 25%, we had a decrease in MIoU of 6% compared to original results.

Following the training, we applied the different perturbations described earlier on the validation set and calculated the average MIoU score and the CD score averaged over all the perturbations. We also evaluated the original MIoU on clean images. The results for the experiment are shown in1.

4.4. Deformable Convolution Experiments

In order to investigate ways to improve state of the art model, we tried implementing a deformable convolution block. It was implemented before the Atrous Spatial Pyramid Pooling (ASPP) so that spatial level offsets are learned automatically by the model without any supervision. According to the authors of the paper (Dai et al., 2017), deformable convolution was successful in learning dense spatial information from deep CNN architectures used in most vision problems, such as semantic segmentation and object detection, but their method is not flexible to large deformation of objects.

With the new architecture (a combination of baseline and deformable convolution) trained using the same training protocol, and then evaluated using the perturbations applied similar to the DeepLabV3+ model, we report the results in table 1. We find that the new model performs better than the baseline model for cropped and rescaled validation images, as well as the perturbed images.

4.5. Experiments Using Self-Attention Networks with and without Coupling with Contour Objective

Finally, we performed our experiments on the self-attention based networks where we created two variants, one coupled with identifying the contours and foreground/background regions and one without this additional task. We used two layers of self-attention operation and the K, Q, V as de-

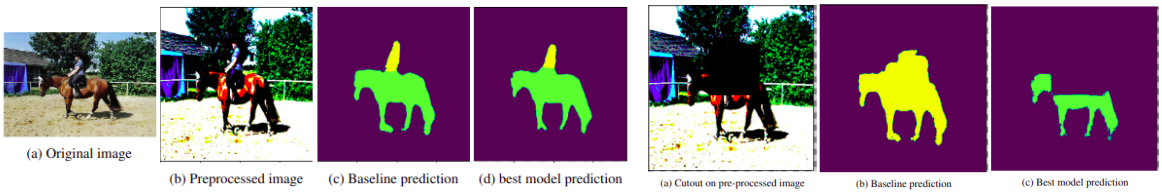


Figure 4. Performance on clean images and cutout versions- baseline vs best performing model (attention+contour)

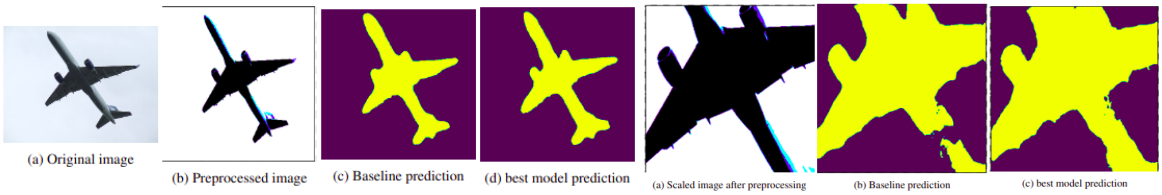


Figure 5. Performance on clean images and scaled versions- baseline vs best model (deformable)

scribed in section 3, had 128 channels. We performed layer normalisation (Ba et al., 2016) during both applications of attention. The class-level predictions obtained at each patch were concatenated after the first upsampling procedure with the original feature maps to allow them access to predictions from patch-level object classification. We then followed the same training protocol and calculated the metrics similar to previous experiments. For coupling with contour and foreground/background objective, the encoder backbone had shared layers up to the attention block before which, we concatenate the features of the two shared layers to calculate K, Q, and V. We tried both with and without creating separate shared layers to use single backbone instead, and it gave poor results. The experiments for this variant were done similarly on the clean and perturbed validation set. The experiments showed that coupling with contour objective improved performance on cutout and occlusions. Further analysis is done in 5.

5. Results/Discussion

After performing our experiments by training the baseline model and our proposed variants over the clean data and their different perturbed version, we summarize the results in tables 1 and 2. Since average MIoU does not tell how robust the model is to different corruptions data, we measure mean corruption degradation of the different models. As we train DeepLabv3+ as our baseline, we consider it as the reference model, and take its corruption degradation as 1. The lower the corruption degradation of a model, the more robust a model is to corruptions. We now summarize our findings based on these results.

- The DeepLabv3+ model is robust to small cutouts of ratio up to 0.1% and small levels of Gaussian blur ($\sigma < 4$), however increasing the noise decreases its MIoU value significantly (table 1) and it is because in gaussian noise add pixel level noise and since we do pixel level classification this might break the image according to the model but looks almost same to human eye.
- From fig 7 as expected when the intensity of perturbations increased the MIoU value decreased but some of the proposed variations had better performance compared to baseline models.
- The vanilla attention based model is the most robust against both Gaussian noise and cutout, which can be observed from its CD value (table 2,fig 7b). It is more robust to Gaussian blur and shows a CD decrease of 0.5 against it. This shows that attention based methods can be a good alternative to improve robustness against these distortions.
- We observed that the attention model when combined with contour information is robust against cutout, and ignores it well for large cutouts as depicted in fig 4. However, the robustness against small cut ratios was almost the same. We believe that this was due to some issue during training and we wish to further analyse this in future to show its robustness against cutouts of all sizes. This shows that adding the contour information can allow the model to ignore irrelevant regions well. Secondly, the vanilla attention based model was robust against the cutout augmentations for all the tested scales (7a) and showed a decrease in CD by about 0.4.
- The deformable convolution based model was the most robust against scale variations(fig 7c,5), and has better performance when several crops were taken from images and rescaled. In the experiments where we take the three scaled variations of the images, the average degradation in MIoU for the deformable convolution was 5.63, whereas it was 6.3 for the baseline model. Moreover, according to our observations, the deformable variant performed better than attention and contour based variants that couldn't handle scale variations well.

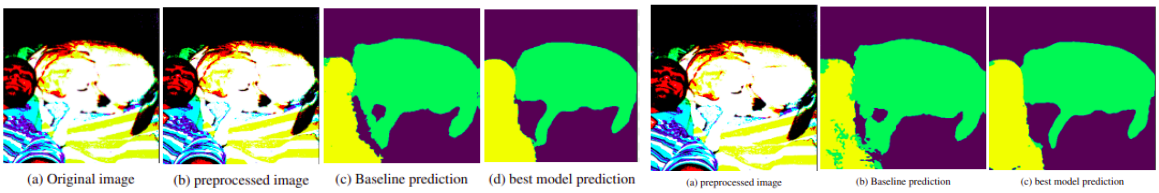
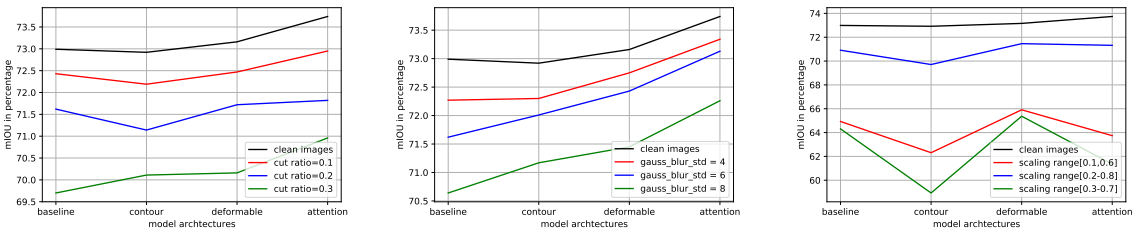


Figure 6. Performance on clean images and image with Gaussian noise- baseline vs best model(attention model)



(a) Cutout (b) Gaussian Blur (c) Image Scaling

Figure 7. Graphs for different variations of perturbations performed

6. Related Works

6.1. Utilizing Context for Semantic Segmentation

Several state of the art models for segmentation use different methods to incorporate the information from context at each position in the generated feature map to improve segmentation. (Zhao et al., 2017) combine features obtained by performing average pooling operation at different scales or kernel sizes at each location to aggregate the contextual information from surrounding regions and objects. (Zhang et al., 2019) use the information of all the objects by first predicting the objects present in an image, and then aggregating this information to each location to refine the features. They also use an additional "semantic encoding loss", which predicts the objects present in the scene through a cross entropy loss. There have been many works which have utilized the self attention mechanism to improve contextual information.

6.2. Attention Based Methods for Semantic Segmentation

The use of attention for capturing the correlations between the features has proved to be useful in many computer vision tasks ranging from object detection, recognition, and classification (Hu et al., 2019). The attention method involves measuring correlations between a pixel and all its surrounding pixels in a feature map, and then performing a summation over those pixels which are most correlated with the current location. This method allows each location to encode information over larger ranges and longer distances, which overcomes the limitation of the small receptive field size of convolution. Recently, these methods have also been shown to improve the results of semantic segmentation on the PASCAL VOC data set. (Fu et al., 2020) proposed a dual attention network to allow each location in a feature map obtained from the output of a ResNet encoder to aggregate the contextual information around it, by attending over all the other locations and channels and gathering salient features from these locations. The transformer model which consists of several layers, each one performing self-attention in each layer, has been highly successful for natural language processing tasks, and has recently shown promise for computer vision tasks as well. (Strudel et al., 2021) and (Zheng et al., 2021) design a transformer based network (Vaswani et al., 2017) to capture correlation at all the levels of a network for performing the task of semantic segmentation, instead of only the outputs of a network. Instead of using the output of any convolutional network as a baseline encoder before performing attention, their model is a fully attention based model, allowing the features at each layer to perform an attention over all its surrounding locations. Their methods also show improved performance on several data sets for the task of segmentation. Recently there have been several works that have explored the robustness of the self-attention based methods and transformer network for image classification (Bhojanapalli et al., 2021). However, improving robustness for the task of semantic segmentation is yet underexplored, and this work aims to provide an analysis of their robustness to different perturbations applied to images.

7. Conclusions

The results of our experiments and our analysis of the corruption degradation shows that hierarchical self attention makes the DeepLabv3+ model robust to distortions and noise in input images. We also observe that adding contour and shape prediction can improve robustness against occlusion through our experiments on cutout. These observations indicate that adding attention based methods can provide more flexibility to the semantic segmentation mod-

els, and allow some noisy regions to be ignored. Adding an auxiliary task of taking the contours into account can enrich the network with shape information, and allow the attention method to take this information into account as we observed in Figure 4.

We are excited to explore these methods further in future by considering larger image sizes, which we were unable to try, and adding the contours and background/foreground information, and attention to multiple layers of the network.

References

Ba, Lei Jimmy, Kiros, Jamie Ryan, and Hinton, Geoffrey E. Layer normalization. *CoRR*, abs/1607.06450, 2016. URL http://arxiv.org/abs/1607.06450.

Bhojanapalli, Srinadh, Chakrabarti, Ayan, Glasner, Daniel, Li, Daliang, Unterthiner, Thomas, and Veit, Andreas. Understanding robustness of transformers for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10231–10241, 2021.

Chen, Papandreou, Kokkinos Murphy Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1706.05587v3*, 2017. URL https://arxiv.org/pdf/1606.00915.pdf.

Chen, Liang-Chieh, Papandreou, George, Schroff, Florian, and Adam, Hartwig. Rethinking atrous convolution for semantic image segmentation, 2017.

Chen, Zhu, Papandreou Schroff Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv preprint arXiv:1802.02611v3*, 2018. URL https://arxiv.org/pdf/1802.02611.pdf.

Dai, Jifeng, Qi, Haozhi, Xiong, Yuwen, Li, Yi, Zhang, Guodong, Hu, Han, and Wei, Yichen. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 764–773, 2017.

DeVries, Terrance and Taylor, Graham W. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Dosovitskiy, Alexey, Beyer, Lucas, Kolesnikov, Alexander, Weissenborn, Dirk, Zhai, Xiaohua, Unterthiner, Thomas, Dehghani, Mostafa, Minderer, Matthias, Heigold, Georg, Gelly, Sylvain, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Fadnavis, Shreyas. Image interpolation techniques in digital image processing: an overview. *International Journal of Engineering Research and Applications*, 4(10):70–73, 2014.

Fu, Jun, Liu, Jing, Wang, Yuhang, and Lu, Hanqing. Stacked deconvolutional network for semantic segmentation, 2017.

Fu, Jun, Liu, Jing, Tian, Haijie, Li, Yong, Bao, Yongjun, Fang, Zhiwei, and Lu, Hanqing. Dual attention network for scene segmentation, 2019.

Fu, Jun, Liu, Jing, Jiang, Jie, Li, Yong, Bao, Yongjun, and Lu, Hanqing. Scene segmentation with dual relation-aware attention network. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Ghiasi, Golnaz and Fowlkes, Charless C. Laplacian pyramid reconstruction and refinement for semantic segmentation. In Leibe, Bastian, Matas, Jiri, Sebe, Nicu, and Welling, Max (eds.), *Computer Vision – ECCV 2016*, pp. 519–534, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46487-9.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Hendrycks, Dan and Dietterich, Thomas. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Hu, Han, Zhang, Zheng, Xie, Zhenda, and Lin, Stephen. Local relation networks for image recognition. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3463–3472, 2019.

Kamann, Christoph and Rother, Carsten. Benchmarking the robustness of semantic segmentation models. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020a. doi: 10.1109/cvpr42600.2020.00885. URL http://dx.doi.org/10.1109/CVPR42600.2020.00885.

Kamann, Christoph and Rother, Carsten. Benchmarking the robustness of semantic segmentation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8828–8838, 2020b.

Long, Jonathan, Shelhamer, Evan, and Darrell, Trevor. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, 2015. doi: 10.1109/CVPR.2015.7298965.

Loshchilov, Ilya and Hutter, Frank. Flickr. *arXiv preprint arXiv:1711.05101*, 2017a. URL https://www.flickr.com/.

Loshchilov, Ilya and Hutter, Frank. The pascal voc project. *arXiv preprint arXiv:1711.05101*, 2017b. URL http://host.robots.ox.ac.uk/pascal/VOC/.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, et al.

Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Strudel, Robin, Garcia, Ricardo, Laptev, Ivan, and Schmid, Cordelia. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7262–7272, 2021.

TensorFlow. Tfrecord and tf.train.example. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jan 2022a. doi: 10.1109/cvpr42600.2020.00885. URL https://www.tensorflow.org/tutorials/load_data/tfrecord.

TensorFlow. tf.keras.metrics.meaniou. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Mar 2022b. doi: 10.1109/cvpr42600.2020.00885. URL https://www.tensorflow.org/api_docs/python/tf/keras/metrics/MeanIoU.

Vasiljevic, Igor, Chakrabarti, Ayan, and Shakhnarovich, Gregory. Examining the impact of blur on recognition by convolutional networks. *CoRR*, abs/1611.05760, 2016. URL http://arxiv.org/abs/1611.05760.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Łukasz, and Polosukhin, Illia. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yu, Fisher and Koltun, Vladlen. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

Zhang, Hang, Zhang, Han, Wang, Chenguang, and Xie, Junyuan. Co-occurrent features in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 548–557, 2019.

Zhao, Hengshuang, Shi, Jianping, Qi, Xiaojuan, Wang, Xiaogang, and Jia, Jiaya. Pyramid scene parsing network, 2017.

Zheng, Sixiao, Lu, Jiachen, Zhao, Hengshuang, Zhu, Xiatian, Luo, Zekun, Wang, Yabiao, Fu, Yanwei, Feng, Jianfeng, Xiang, Tao, Torr, Philip HS, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6881–6890, 2021.

Çağrı Kaymak and Uçar, Ayşegül. A brief survey and an application of semantic image segmentation for autonomous driving, 2018.