Object Aware Approaches in Removing Motion Blur

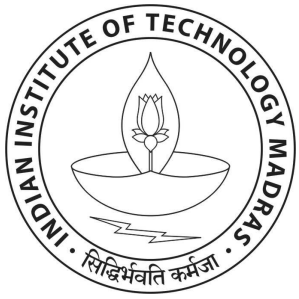
A Project Report

submitted by

JOEL ABHISHEK ROGERS

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY AND MASTER OF TECHNOLOGY



DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

May 2020

THESIS CERTIFICATE

This is to certify that the thesis titled Object Aware Approaches in Removing Motion Blur, submitted by **Joel Abhishek Rogers (EE15B132)**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology and Master of technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. A.N.Rajagopalan
Research Guide
Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

ACKNOWLEDGEMENTS

I wish to thank my primary thesis advisor Prof. A.N Rajagopalan from the Department of Electrical Engineering, at IIT Madras, for his guidance on streamlining the principal idea behind this thesis. I am grateful for the discussions I had with him about the latest developments in the field of image deblurring which helped build up the thesis to its current state.

I would like to thank my lab mate Kuldeep Purohit for his timely assistance whenever I ran up against a wall in my research. His expertise in deblurring models was invaluable in developing this thesis.

I am grateful for the mentor ship provided by my lab mate Maitreya Suin. All the ideas in this thesis were borne out of discussions with him. It was my pleasure to discuss ideas and learn more about his branch of research.

ABSTRACT

KEYWORDS: Deep Learning, Convolutional Neural Nets, Image Deblurring, Synthesizing Motion Blur, Object Detection, Attention, Self-attention

In dynamic scenes, every object of interest is affected by varying degrees of motion blur. Moreover, such objects of interest maybe different for different applications in the same scene. We strive to disentangle the objects of interest from the background of the scene and reconstruct the sharp image by employing a model with an encoder-decoder architecture.

For the purpose of training this model, we construct a custom data set by synthesizing motion blur to a pre-existing data set which is used widely in the realm of object detection. We demonstrate performing object attentive deblurring on the custom data set by generating region proposals using a Region Proposal Network (RPN).

We observe the effect of integrating a self-attention module to the encoder-decoder chain to leverage the global context of the image along with the regions of interest to deblur the image. We hope to inspire further research in speeding up patch wise processing by using custom CUDA Kernels to provide a boost in computation time.

We hope that this report introduces the reader to approaches by which attention can be worked into any similar deep learning application.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

IITM	Indian Institute of Technology, Madras
FG	Foreground
BG	Background
VOC	Visual Objects Class
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
SepConv	Separable Convolution
ROI	Region Of Interest
R-CNN	Region Convolutional Neural Network
FPN	Feature Pyramid Network
RPN	Region Proposal Network
NMS	Non Maximal Suppression
SGD	Stochastic Gradient Descent
IoU	Intersection over Union
PSNR	Peak Signal to Noise Ratio
ReLU	Rectified Linear Unit
SA	Self-attention

CHAPTER 1

INTRODUCTION

In this thesis, we develop a encoder-decoder model for the purpose of treating motion blur in dynamic scenes. By leveraging the recent advances in the realm of object detection we hope to integrate a differentiable, supervised object/region of interest detection module to perform patchwise deblurring. With the attention module, we hope to construct a model that with a single pass targets and deblurs only certain regions/objects of interest that attracts the viewer's attention. For this purpose we also construct a custom data set by modifying a classical object detection data set to suit our needs.

1.1 Deep Learning In Deblurring

Loss of image information in all present day imaging systems is an unavoidable problem. For reasons that may be intrinsic like faulty anti-aliasing filters, problems with the lens of the camera or that may be extrinsic like the motion of the objects in the scene, shake in the camera, loss of focus etc., the resulting images produced invariably turn out blurry. Removing this blur is thus an ubiquitous problem that the field of computer vision has tackled in many different ways.

We further concern ourselves with removing motion blur in this thesis, i.e. blur that is caused due to the relative motion between the objects in the scene and the camera. With the arrival of convolutional neural nets, deep learning methods have been shown to provide better results (according to benchmarking metrics like PSNR and SSIM) than other conventional non-learning based methods in removing motion blur. These methods either estimate certain parameters of the blur kernel as in Chakrabarti (2016), Gong *et al.* (2017) and Sun *et al.* (2015) which then allows for the recovery of the sharp image or are kernel free, where the sharp image is reconstructed directly from the given blurred image. Such kernel free, end to end methods either employ an encoder decoder type of architecture to reconstruct the sharp image as in Nah *et al.* (2017) and Tao *et al.* (2018) or employ generative models which are trained with adversarial loss as in Kupyn

et al. (2018). A survey of the above methods in Sahu *et al.* (2019) goes ahead to show that the kernel free, end to end methods outperform the kernel based ones.

While handling dynamic scenes with non-uniform blur, these end to end methods still attempt to deblur the entire scene, along with the various objects in the foreground and the background simultaneously. Only a few methods like Pan *et al.* (2016) pioneer object motion blur kernels. The recent work on Human Aware Deblurring in Shen *et al.* (2019) shows that explicitly discriminating the humans from the background in a dynamic scene using a *deferentiable, supervised human aware model* followed by a multi-head encoder-decoder model result in comparable results with the sate of the art.

1.2 Differentiable attention in Neural Nets

Differentiable neural attention is a topic which has been gaining a lot of attention in recent times. It serves to mimic the human cognitive attention mechanism, to selectively focus attention on the most visually informative parts of an image. It is widely used in a variety of tasks in computer vision like image captioning, scene recognition, question answering etc. In all these tasks, the attention mechanism is learned in a goal-driven, end to end manner allowing the network to concentrate on the most relevant parts of the input for it's intended task.

Attention has been used in image deblurring tasks by generating segmentation masks to direct focus. In Shen *et al.* (2019), a human-aware attention module is put forth that explicitly decodes foreground human information by generating a soft human mask, that directs the attention of the encoder decoder chain that follows it. Moreover, this human-aware attention module is made to learn from annotated data in a supervised manner.

1.3 Self-attention

In the seminal paper Vaswani *et al.* (2017), self attention was introduced. It saw increasing use in the field of natural language processing where each word in the sentence undergoes attention computation with every other word in the sentence. This enabled

the network to learn dependencies between words along the sentence enabling the network to learn semantic information from sentences producing impressive results in tasks like English to German and German to English translation. Moreover, the transformer model for self attention proposed in Vaswani *et al.* (2017), all the calculations can be done via matrix computations and can be parallelized. This allows for easy integration into any deep learning model and weights of the self attention layer can be learned in an end to end manner.

1.4 Proposed network architecture

In this thesis we go about designing an encoder-decoder based deblurring model that can be trained in an end to end manner. Taking a leaf out of Shen *et al.* (2019), we add a *differentiable attention model* which we then train in a supervised way to generate patches of interest in the blurred dynamic scene.

By constructing a custom data set, we bring supervision into the picture with attention being directed to the regions that are of interest as dictated by the bounding boxes in our custom data set. With the use of recurrent neural networks in Mnih *et al.* (2014), the authors showed how by mimicking the human attention mechanism, information relevant to the task can be built up by focusing visual attention to specific regions in the image. This network went on to outperform convolutional networks that processed the entire image in image classification tasks. We decide the regions of interest to focus visual attention in our task of deblurring to be objects like animals, humans, vehicles etc. that stand out to be distinguishable from the back ground, as these are the things that capture a viewer's attention when he/she views a scene.

We use this custom data set to train an attention module which takes the form of a region proposal network in a supervised manner. We focus on deblurring all the objects in the foreground of any scene disregarding the blur in the background. A basic block diagram of the proposed model architecture with the direction of data flow is shown in Figure 1.1

Deblurring requires a large receptive field (global knowledge) and since we are discarding the background we lose long-range spatial dependencies that may exist is the image. This is usually tackled by using a scaled structure and a large number of con-

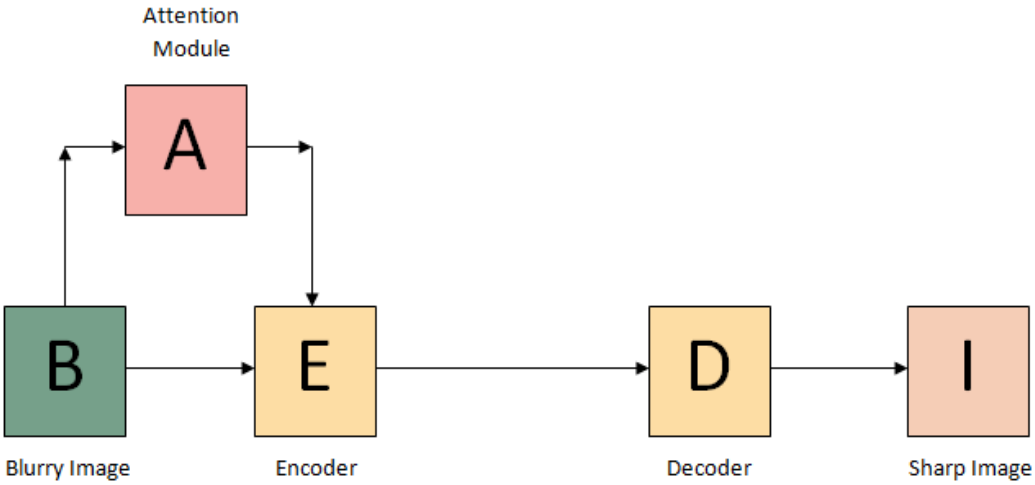


Figure 1.1: Simplified block diagram of proposed network

volutional layers with residual connections in between them, but this comes at the price of increase in the number of parameters for all those extra layers. We propose the addition of a self attention model in the encoder-decoder chain where the patched regions of interest would undergo attention computation with features extracted from the entire blurred image. We hope to leverage the feature maps generated from the attention module and capture long range spatial dependencies (global context) with the addition of this model. A basic block diagram of the model architecture with a self attention module along with the direction of data flow is shown in Figure 1.2

1.5 Contributions of this Thesis

The contributions of this thesis are threefold:

1. We provide a feasible process by which a custom data set with blurred images with bounding boxes around them can be constructed from an classical object detection/classification data set.
2. We perform object attentive motion deblurring on the custom data set and compare the results with a baseline model without the attention module.
3. We explore the possibility of self attention being a viable method of capturing the global context and compare its results with the other models.

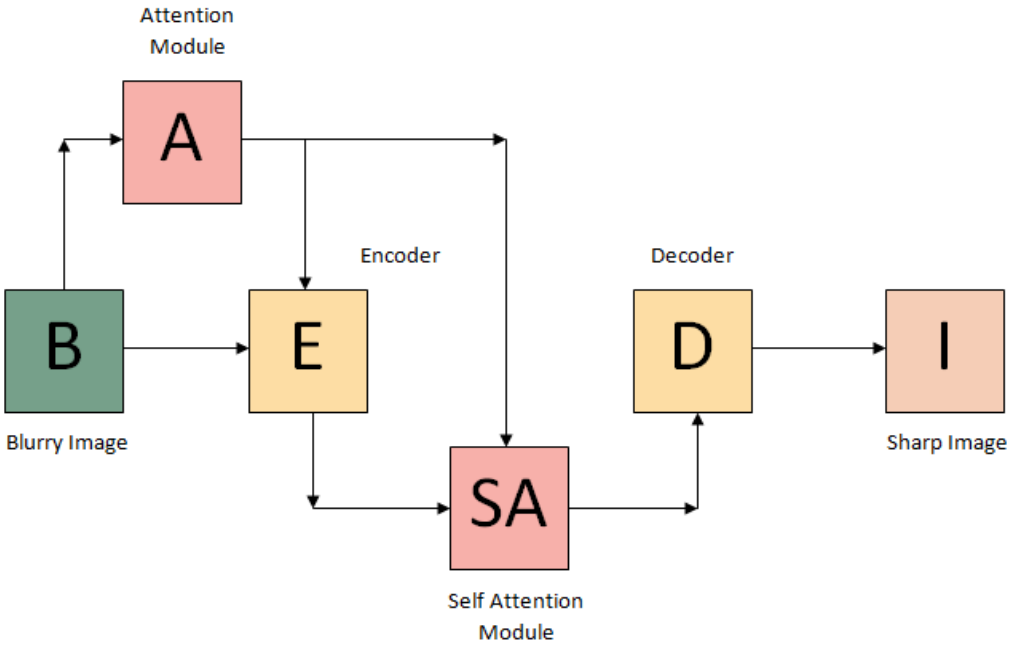


Figure 1.2: Simplified block diagram of proposed network with self attention

1.6 Outline of this Thesis

The remainder of the thesis is set out as follows:

- In chapter 2, we elaborate the motivation behind choosing and the process by which a classical object detection data set (ImageNet by Russakovsky *et al.* (2015)) is artificially blurred.
- In chapter 3, we provide the motivation for choosing and the results upon training a Region Proposal Network (RPN) introduced in Ren *et al.* (2015) to be our attention module.
- In chapter 4, we compare the results of the object attentive deblurring model thus developed with a baseline model.
- In chapter 5, we add in a self attention model and discuss and provide the results thus obtained.
- In the final chapter, we briefly discuss future work in the field, and the potential impact of such research in the field.

CHAPTER 2

CUSTOM DATASET

For the purpose of this thesis we require an Image deblurring dataset that also has to have ground truth annotation bounding boxes for the objects in the scene. Naturally we look towards Object detection and classification datasets. These datasets while equipped with ground truth annotation in the form of bounding boxes lack blurry images. With the absence readily available datasets that meet our requirements we go about preparing our own custom dataset by synthesizing motion blur over an object detection dataset.

2.1 Choice of Dataset

2.1.1 Image Deblurring Dataset

Image deblurring has experienced remarkable progress in recent years. One of the critical factors bootstrapping this progress is the availability of large-scale datasets. Early works directly convolved sharp images with a set of pre-defined motion kernels to synthesize blurry images. Examples of such datasets are the BM4CS dataset proposed in Kohler *et al.* (2012) and the larger dataset of over 1000 images constructed by Sun *et al.* (2015) sourced from the PASCAL VOC dataset. Though widely used, such patch-wise generated datasets yield discrete approximations of real blurry images with pixel-wise heterogeneous blurs. Recently, to construct a more real blurry image dataset, several researchers like Niklaus *et al.* (2017) have generated dynamic blurred images by averaging multiple successive frames captured by high frame-rate video cameras.

Since we wish to synthesize real motion blurs over our objects of interest, we turn our attention to Niklaus *et al.* (2017), where frame interpolation is performed by convolving each pixel with a spatially adaptive interpolation kernel.

2.1.2 ImageNet VID

Unfortunately, since our method of synthesizing motion blur relies on frame interpolation, the base dataset that we synthesize motion blur has to be in video format. Most of the Object detection and classification datasets consist of static images.

The ImageNet VID dataset is a widely used Object classification and scene classification dataset prepared by Russakovsky *et al.* (2015) and released as part of the Large Scale Visual Recognition Challenge to evaluate object detection and classification algorithms on videos. It contains ground truth annotations for 30 different classes of objects which include different classes of animals, vehicles and aircraft. It has a training/validation/testing split of 1952/281/358 unique videos with annotations for each frame in the video. We construct our custom dataset by synthesizing motion blur over the ImageNet VID dataset.

2.2 Dataset Construction

2.2.1 Synthesizing Motion Blur

We make use of the pre-trained frame interpolation network (SepConv) described in Niklaus *et al.* (2017). The SepConv network takes in two frames in as its input and generates an interpolated third frame that lies exactly in between the two input frames. To generate a realistic motion blur, we pick out three consecutive frames which possess noticeable movement of the object of interest. The middle frame of each triplet can be used as the ground truth for the final blurred image. We apply the SepConv network, with the first and second frames of the triplet frames as input and then with the second and third frames of the triplet as input. This synthesizes two in between frames, one in between the first and second frame and the other in between the second and third frame of the chosen triplet. We then apply this same process recursively using the real and newly-interpolated frames as input, generating more interpolated frames in between them. This process is carried out 4 times in total, which results in a sequence of 33 frames. The recursive application of the SepConv net generates a total of 15 frames in between each image in our triplet. The 33 frames are all then averaged to produce a synthetically motion blurred image. The flow is depicted pictorially in Figure 2.1.

A similar method is employed in creating an artificially blurred dataset by Brooks and Barron (2019) and is shown to produce realistically blurred images.

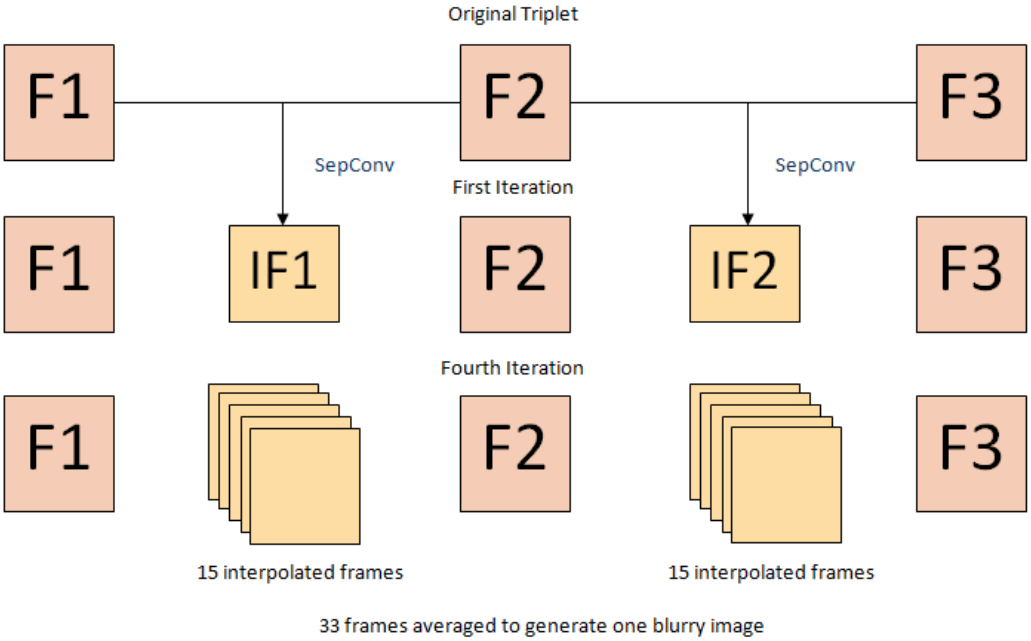


Figure 2.1: Synthesizing motion blur for a one triplet of frames; F denotes Frame and IF denotes Interpolated Frame

For each video snippet in the ImageNet VID dataset, we pick out a triplet and apply the blurring process described above. The bounding box data of the middle frame is taken to be the bounding box associated with the blurred image created. Care has been taken while picking out the triplets so that the object exhibits sufficient motion to be blurred artificially by interpolation.

2.2.2 Dataset Statistics

For each video snippet in the ImageNet VID dataset we generate a pair of images - one sharp and one artificially blurred with ground truth annotations. The custom dataset thus synthesized has a total of 2591 sharp and blurred image pairs.

The training, validation and testing split of ImagNet VID is maintained to ensure equal distribution of all object classes across all the sub-splits. The custom dataset thus boasts a training/validation/testing split of 1952/281/358 image pairs. This split is used to train the attention module which will be described in the following section.

Figure 2.2: Some examples of sharp images and their corresponding artificially blurred images from our custom dataset with their ground truth bounding boxes
We can observe that the frame interpolation method employed results in producing realistic motion blur.

CHAPTER 3

ATTENTION MODULE

With the availability of a dataset with bounding box annotations over the blurry objects, we go about constructing a model that can generate object proposals given an image. Since, we are training the model with data that has been blurred we hope to nudge the model towards making proposals over blurred objects in an image.

3.1 Choice of Attention Module

Naturally, Object detection and region of interest (ROI) proposal nets come to mind. Initial attempts in designing the attention module strove to classify the object as well on top of localizing it, with the aim of using the classification data to improve the deblurring chain.

Anchor based approaches like Fast R-CNN introduced by Ren *et al.* (2015), Grid R-CNN by Lu *et al.* (2019), RetinaNet by Lu *et al.* (2018) along with a non-anchor based approach in the form of RepPoints by Yang *et al.* (2019) were trained and evaluated on the custom dataset created. The base skeleton was the same across all the networks considered, namely a pretrained 50 layer Residual network (ResNet-50) along with a FPN introduced in Lin *et al.* (2017). As expected the RepPoints network performed poorly as the blurring made point set representation of objects difficult. The anchor based methods resulted in good localization of the blurred objects but classification performed poorly particularly for small animals (cats, dogs, lizards, birds etc.) as upon blurring they are hard to distinguish from each other along with a high degree of blur being usually associated with them as they have a high frequency of movement in the videos.

3.1.1 Region Proposal Network

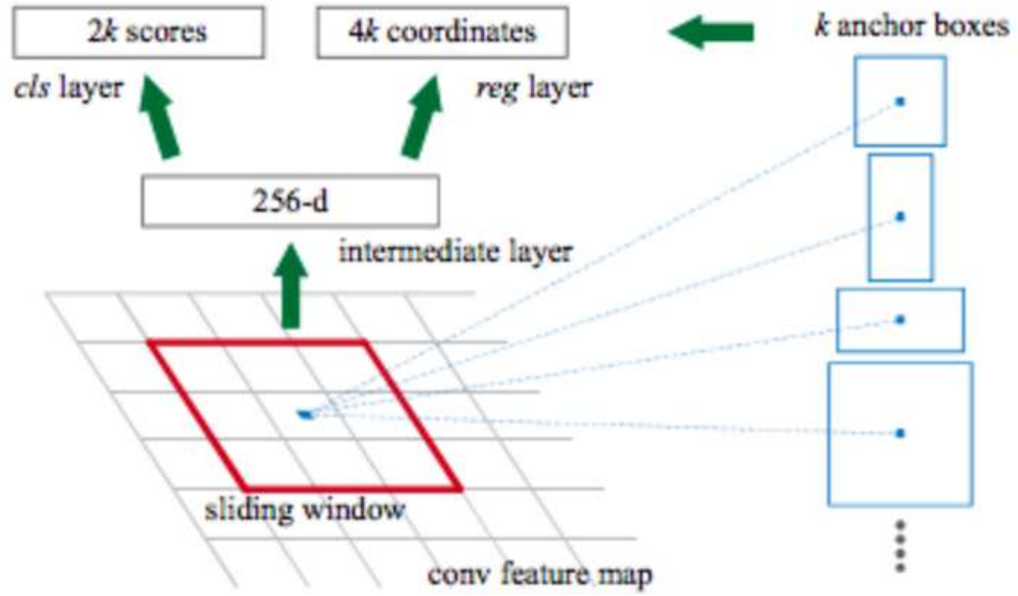
Taking the poor classification results into consideration along with an absence of a feasible framework that can leverage the classification data of the objects for better deblurring, we turn our attention to ROI proposal nets.

RPN developed in Ren *et al.* (2015) is the first stage in a Fast R-CNN object detection model that generates region proposals and ranks them in decreasing order of objectness score i.e. the highers the objectness score of a region box, higher the probability that it contains an object. The time cost of generating region proposals is much smaller in RPN than conventional methods like selective search van de Sande *et al.* (2011) and has the added benefit that it learns to detect regions defined by the ground truth bounding boxes, so it can be adapted to localizing blurry objects in a scene unlike classical object proposal methods like selective search which does not possess a learning component in its makeup. RPN consists of a regressor and a classifier. By sliding a window over the feature maps generated from the image, the classifier determines the probability of a proposal having an object and the regressor regresses the coordinates of the proposal. A simplified functioning of the RPN is depicted in Figure 3.1. Our attention module is closely modeled on RPN.

3.2 Architecture of the attention module

The region proposal network consists of a pre-trained ResNet-50 as its feature extraction network. The feature maps extracted are fed into a FPN which creates 256 dimensional feature map at four different scales at its output. At each scale, a 3x3 convolution is applied (which is the sliding window) followed by a two separate 1x1 convolutions for objectness prediction and bounding box regression. These 3x3 and 1x1 convolution layers are collectively called the RPN head. The architecture described along with the data flow is shown in figure 3.2.

The RPN head outputs a lot of region proposals which are then ranked according to their objectness score in descending order. NMS (Non Maximal Suppression) is then applied so that we may retain only the most confident ones. We further prune the region proposals by discarding those with an $IOU > 0.7$.



Credits: Original Research Paper

Figure 3.1: From Ren *et al.* (2015), RPN head in action

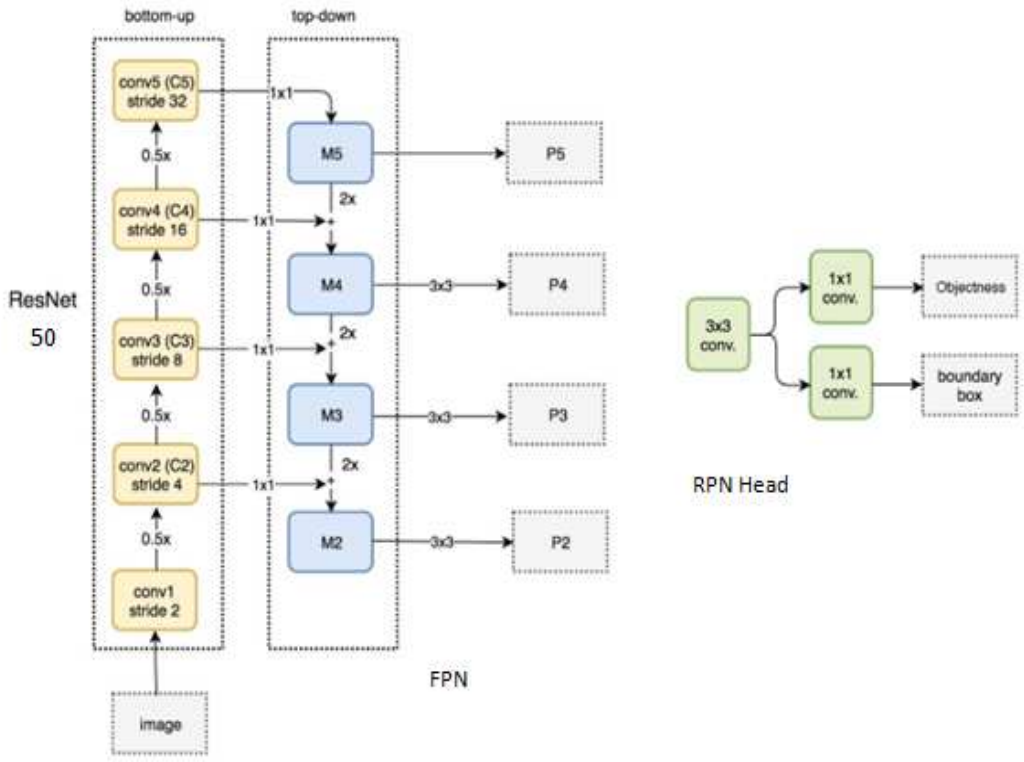


Figure 3.2: Modified from Hui (2018), A simplified flow of data in a RPN

3.3 Implementation details

The RPN network described was then trained upon our custom dataset. There are a total of 1952 training images and 281 images in our validation set. The loss functions are smooth L1 loss for regressing the bounding boxes and cross entropy loss for classifying the region proposals. We minimize the loss functions using SGD with an initial learning rate of 2.5×10^{-3} and a weight decay parameter of 1×10^{-4}. We train for a total of 12 epochs until convergence.

A few results on images from our test split are displayed in Figures 3.3 and 3.4.

Figure 3.3: Region proposals for the classes bird, cat, cow and elephant

Figure 3.4: Region proposals as green bounding boxes for the classes bicycle and car

The attention module thus takes in an image and gives a set of bounding boxes

along with the corresponding confidence score for each bounding box at its output. These bounding boxes are fed forward to the deblurring chain which will be described in the following chapter.

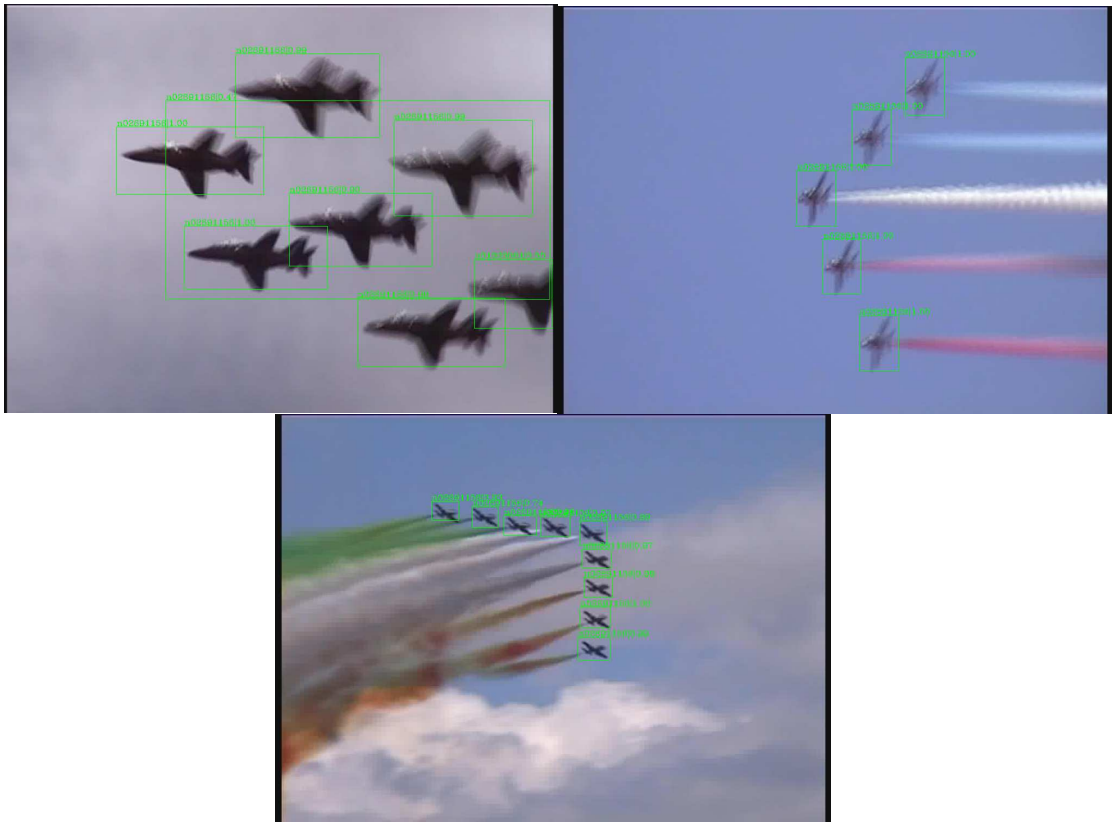


Figure 3.5: Region proposals as green bounding boxes for the class of airplanes

CHAPTER 4

OBJECT ATTENTIVE DEBLURRING

4.1 Network Architecture

4.1.1 Deblurring Network

The deblurring network is an encoder-decoder style network which is modelled after the U-Net architecture introduced in Ronneberger *et al.* (2015). The architecture is symmetric and consists of two major parts - the left part called the contracting path and the right part called the expansive path. The contracting path consists of multiple instances of a 2D double convolution layer followed by a max pooling layer of kernel size 2 i.e. downsampling by 2. A double convolution layer is a sequence consisting of a 2D convolution layer of kernel size 3 and stride 1 followed by a ReLU layer and then by the same 2D convolution layer and ReLU layer repeated once. A simplified block diagram is shown in Figure 4.1.

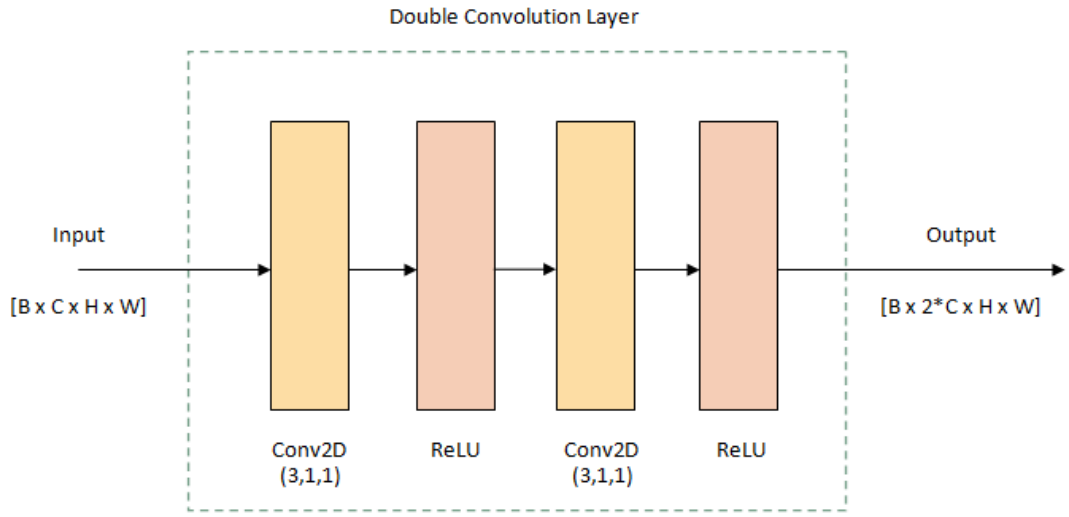


Figure 4.1: Double convolution layer in the contracting path (in the expansive path, halves the number of dimensions)

The expansive path consists of multiple instances of a 2D transposed convolution layer with kernel size 2 and stride 2 i.e. upsampling by 2 followed by another 2D double

convolution layer. Each double convolution layer in the contracting path double the number of dimensions and in the expansive path halve the number of dimensions while each transposed convolution layer along with upsampling the feature map by 2, cuts the number of dimensions by half. There are also skip connections which concatenate feature maps from the contracting path with the feature maps of the expansive path. At the end of the expansive part there is a 2D convolution with a kernel of size one and stride one. The entire network is depicted in the form of a block diagram in Figure 4.2. It is accompanied by the dimensions of an input image as it flows through the network.

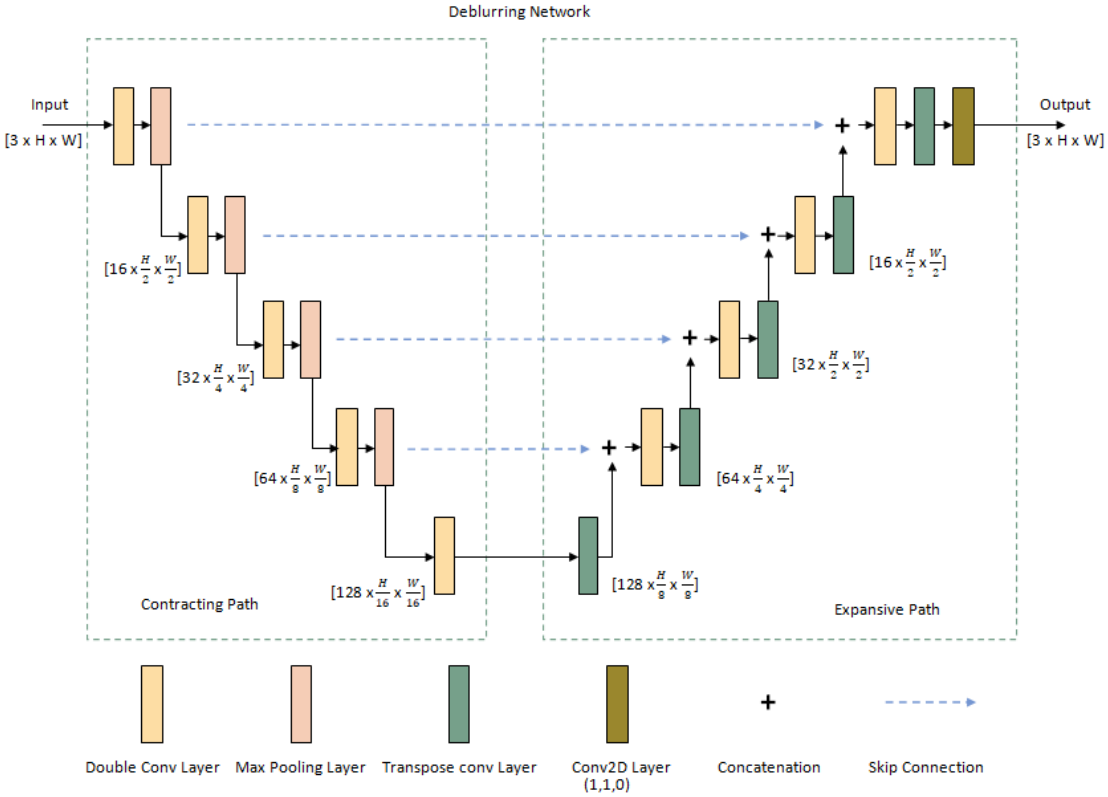


Figure 4.2: The deblurring U-Net architecture

The network is fed a blurry image after it's average is subtracted from it. The sharp image is recovered by summing the blurry image and output of the deblurring net. At the end we add back the average which we removed from the blurry image at the beginning. This net described till now will serve as the baseline deblurring net as it does not have any form of attention built into it.

4.1.2 Integrating the attention module

The blurry image is first fed into our attention module i.e. the RPN described in the previous chapter. We further prune the multitude of region proposals that the RPN outputs by performing NMS with a certain threshold.

Training Pathway

Attention is worked into the network by modifying the input to the deblurring net using the object proposal bounding boxes (high confidence ones). Since the training set consists of images of varying size, we train the deblurring net by taking patches of fixed size (128x128 in our case) from the blurry images and create batches of multiple patches to feed into the network. In the baseline net without any attention, the batching algorithm randomly picks patches of the appropriate size (128x128) from the input blurry image.

However, in our object attentive deblurring variant the batching algorithm takes in the object proposal bounding boxes and the blurry image as its input. It randomly picks one high confidence object proposal from the set of proposals and then the patching is carried out still at random but its made sure that the patch lies within the bounding box or at least some part of the bounding box intersects with the patch chosen. If only a part of the bounding box intersects with the chosen patch the part of the patch that isn't within the bounding box is replaced with zeros. Thus the patches that are fed forward to train the deblurring net have image information from only within the bounding box chosen. A simplified block diagram of the process is shown in Figure 4.3, and some example patches are shown in Figure 4.4.

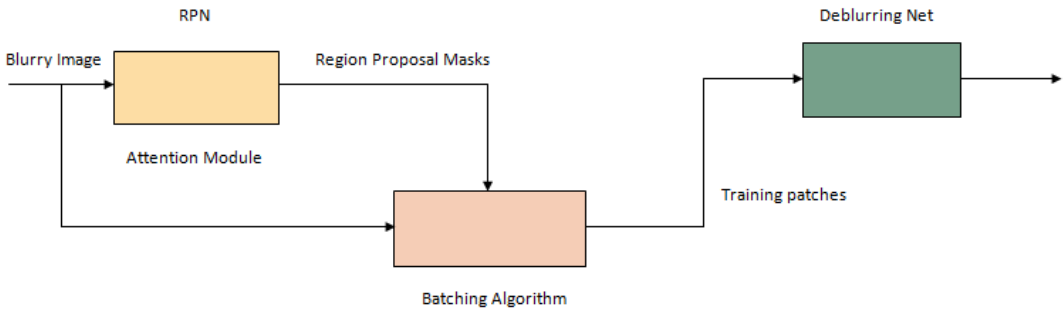


Figure 4.3: Data flow during training

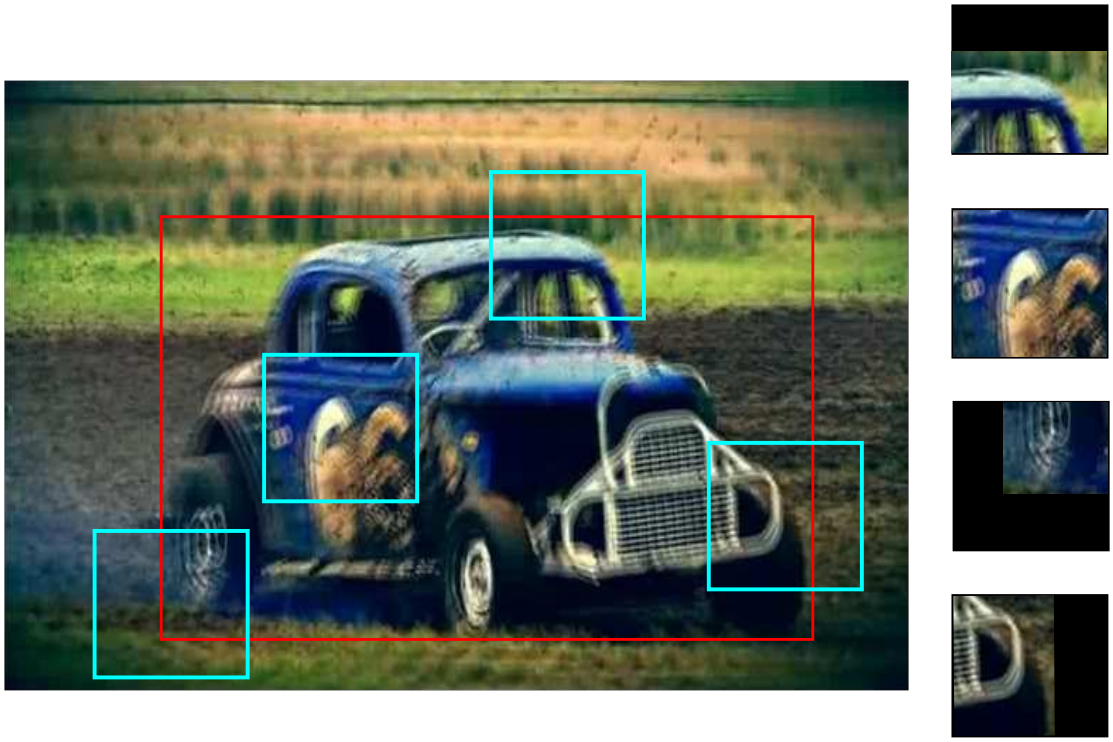


Figure 4.4: Example of training patches generated from a blurry image
The red bounding box is the output of the RPN The cyan bounding boxes are the patches that the algorithm may randomly chooses.

The corresponding ground truth patch from our data set also has the same bounding box applied to it as a mask, setting everything that does not lie within it to zero. Then once the blurry patches are passed through the deblurring network, we optimize the L1 loss between the resultant output patch and the corresponding ground truth patches(with the mask applied).

Testing Pathway

After training the net, the data flow during testing it on novel data is as follows. The blurry input image is passed through the RPN and the region proposals are obtained as bounding box coordinates. The top ones are picked out (by confidence scores) and applied to the input images as masks i.e. any part of the original image that do not lie within the region proposals is set to zero. This masked out image is then fed into the deblurring net. The output added to the input after applying an inverse of the region proposal mask gives us an image that is deblurred only at the patches that our attention module has picked out to contain an object. The data flow during testing a novel image is shown in Figure 4.5.

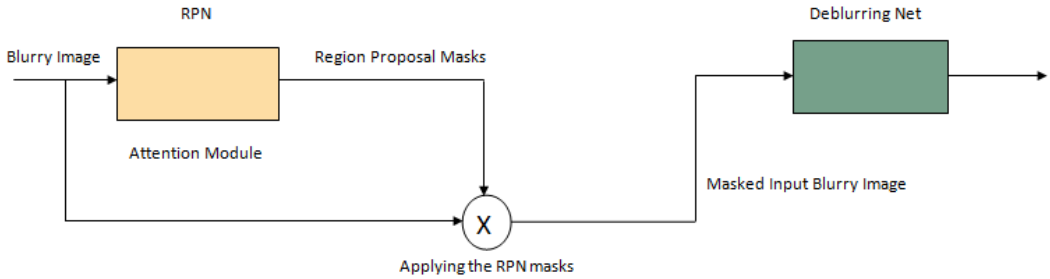


Figure 4.5: Data flow during testing with a novel image

4.2 Implementation Details

We train the deblurring network described upon our custom data set. We work with a total of 1952 training image pairs and 281 validation image pairs of varying sizes. Each image pair consists of a blurry image and the corresponding sharp image.

We crop a patch of 128x128 for each image pair and use a batch of 16 for each iteration. The deblurring net is trained by minimizing the L1 loss between the sharp image patch and output patch of the deblurring net after both the image patches are multiplied with the mask we obtain from the RPN (attention module). The patching algorithm ensures the patches for training lie within the attention mask in some capacity. We use the Adam optimizer introduced in Kingma and Ba (2014), with an initial learning rate of 1×10^{-4}. We train for a total of 300 epochs until convergence.

Identical training parameters are used to train the baseline net which is just the deblurring net without the attention module. Here the patching algorithm chooses patches at random from the blurry image without being subject to any restrictions.

CHAPTER 5

ADDING IN SELF-ATTENTION

5.1 Network Architecture

5.1.1 Self-Attention layer

Efficient Deblurring requires a large receptive field (global context) and by masking out regions that aren't of interest to us in the BG we lose out on a bulk of useful image information that can help us deblur the object patches better.

We turn to the self-attention model introduced in Vaswani *et al.* (2017). The transformer model for self attention proposed in it has been adapted to our needs. The transformer is based on the multi-head attention mechanism. A simplified diagram is depicted in Figure 5.1.

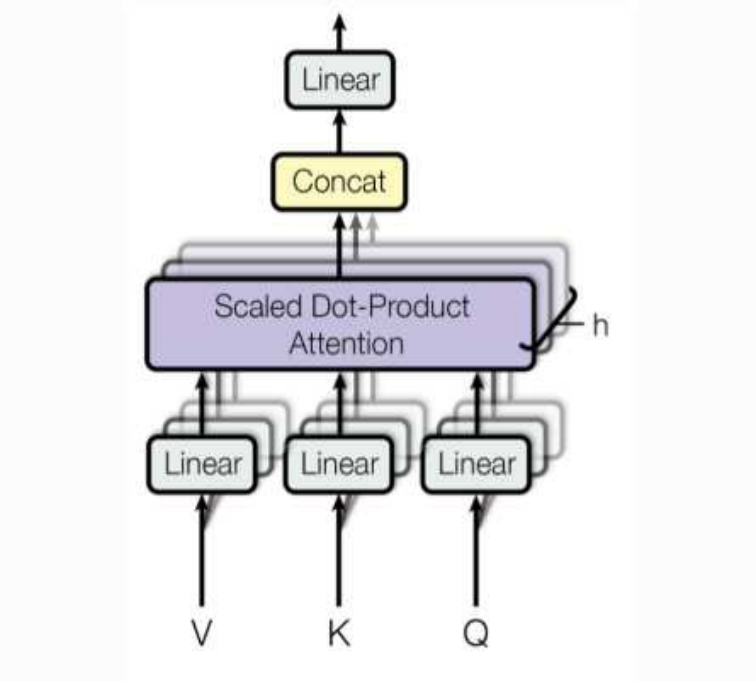


Figure 5.1: Multi-head self attention transformer (Image from Vaswani *et al.* (2017)

The transformer views the encoded representation of the input as a set of key-value pairs (K, V). Now given a set of queries Q, the transformer computes for each query a

scaled dot-product attention score. The output is a weighted sum of the values, where the weight assigned to each value is determined by the dot-product of the query with all the keys.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{n}})V \tag{5.1}$$

Now multi-head attention instead of computing the attention score only once, the multi-head mechanism runs through the scaled dot-product attention multiple times in parallel. The independent attention outputs are then simple concatenated and transformed linearly to the required dimensions. Quoting Vaswani *et al.* (2017), "multi-head attention allows the model to jointly attend to information from different representation sub spaces at different positions. With a single attention head, averaging inhibits this".

$$MultiHead(Q, K, V) = [head_1; head_2; ...; head_n]W^O \tag{5.2}$$

$$head_i = Attention(QW^Q, KW^K, VW^V) \tag{5.3}$$

where W^Q, W^K, W^V and W^O are parameter matrices to be learnt.

In the key and value pair (K, V) are generated from the same feature map/vector we end up with self-attention. We add a self-attention transformer to the lowest layer of the deblurring U-net, at the point where the contracting path ends and the expansive path begins. The feature map at the end of the contracting path after flattening it is our Query vector. The Key and Value vectors are obtained by flattening a feature map from the feature pyramid network (part of the RPN that processes the entire image). Thus, every pixel in the patch we process in the deblurring net looks into every other pixel in the entire image and undergoes attention calculation. This way we make use of the image information we miss out on by masking out the BG while still focusing our attention on regions of interest in the blurry image. A block diagram of the deblurring U-Net with the self attention layer added in is shown in Figure 5.2.

This network is trained in the same way that was described in section 4.1. Since at every level in the feature pyramid network in our attention module, the number of dimensions is 256, we can use any feature map from the feature pyramid network as the input to our feature pyramid network.

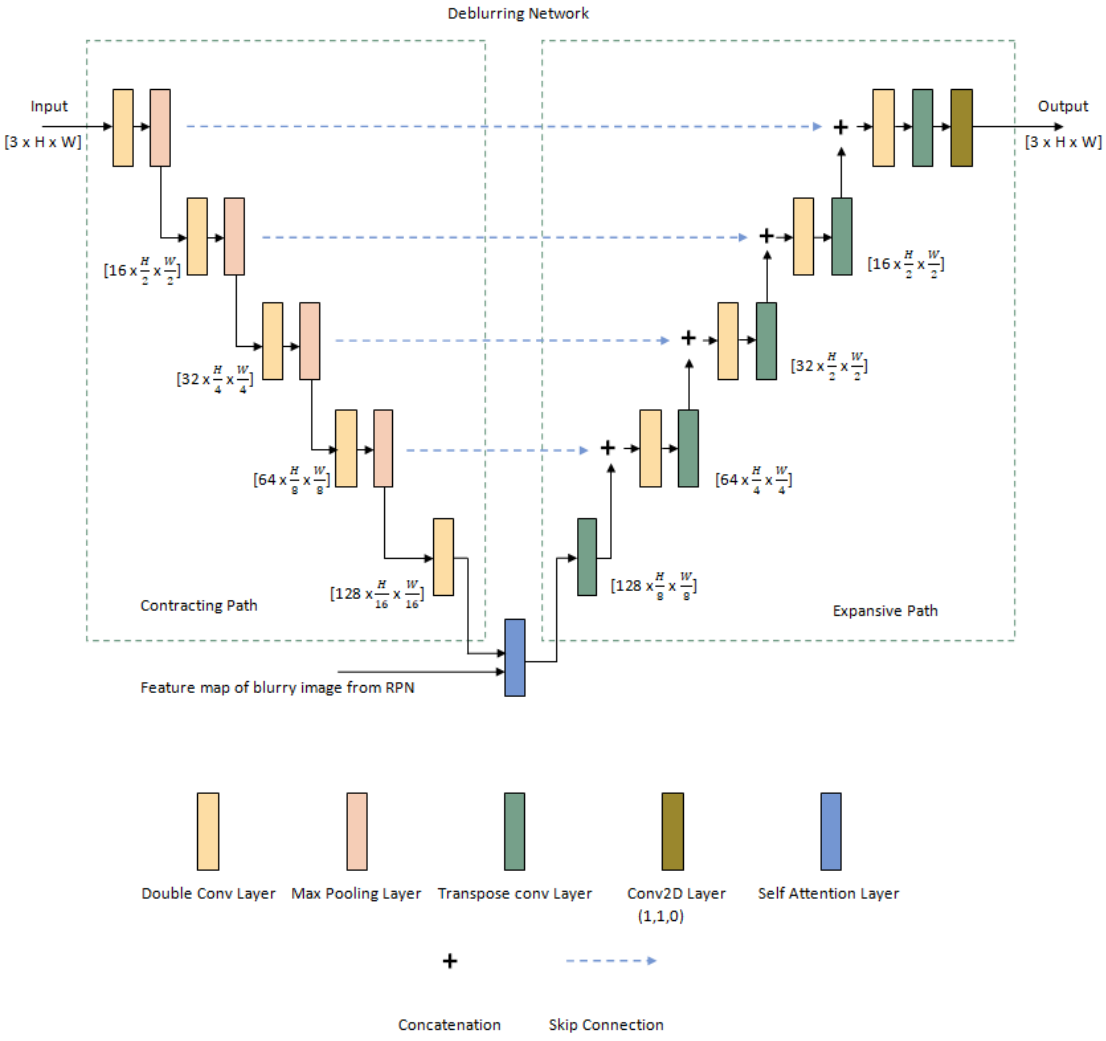


Figure 5.2: The deblurring U-Net architecture with self attention

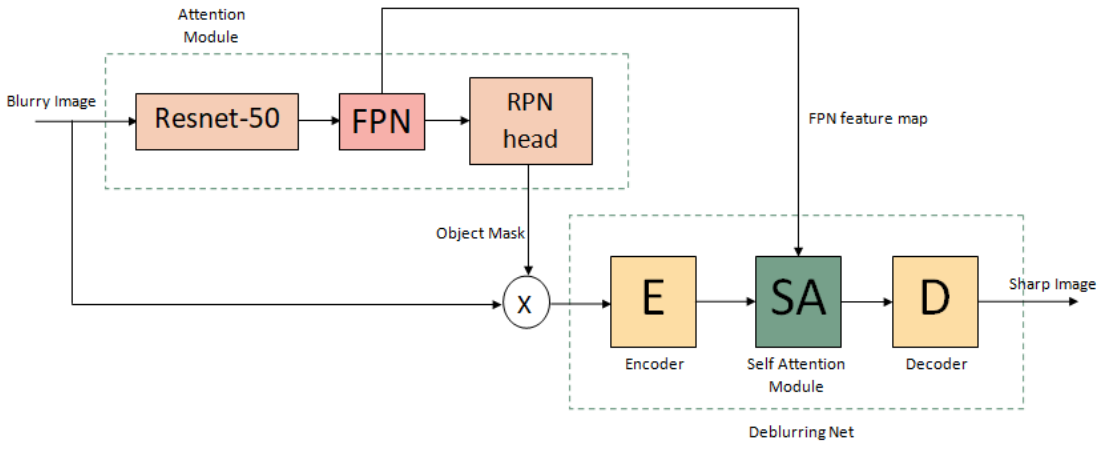


Figure 5.3: Top down view of the entire network

Greater the dimensions of the feature map chosen, higher the computation time incurred. In the implementation that was tested in the thesis, feature maps from the second level from the top of the FPN were used. A top down view of the entire network is included in figure 5.3.

5.2 Implementation Details

This network is also trained on our custom data set with a training/validation split of 1952 to 281 image pairs.

We use the Adam optimizer introduced in Kingma and Ba (2014), with an initial learning rate of 1×10^{-4}, to minimise the L1 loss between the blurry and sharp patch. We run the net for a total of 300 epochs until convergence.

Evaluation Metrics

For quantitative evaluation, Peak Signal-to-Noise-Ratio (PSNR), a standard metric used in the field is adopted.

$$PSNR = 10.log(\frac{MAX_I^2}{MSE}) \tag{5.4}$$

where MAX_I is the maximum possible pixel value in the image, and MSE is the mean squared error pixel-wise error between the noise free mono chrome image I and it's noisy approximation K.

For an mxn image, MSE is given by

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} {}^{n-1}[I(i,j) - K(i,j)]^2 \sum_{j=0}$$

In our case the ground truth sharp image is I and K is the image that our deblurring net generates after processing its blurry pair. Since, we are concerned with the performance of attention based deblurring in this thesis, we perform the PSNR calculation only over the region of interests that the RPN outputs (over the objects in the image). We repeat the calculation for each channel and average the results to report the PSNR.

Comparison of results

We create a benchmark set of 15 test image pairs from the test split of our custom data set and run the baseline deblurring net without attention, the deblurring net with the RPN and the deblurring net with the RPN and the self attention layer and tabulate the PSNR obtained.

Table 5.1: Table comparing the PSNR reported by the different nets

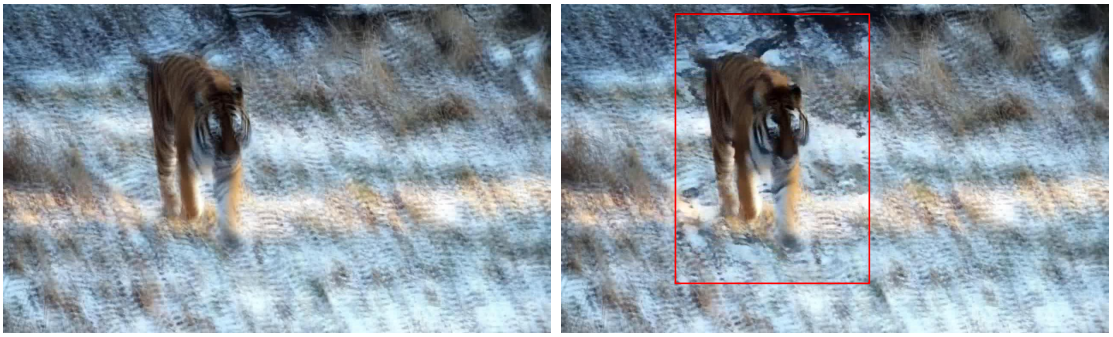
Model	PSNR	Inference Time (s/img)
Baseline Deblurring U-Net	30.506	0.1375
Deblurring U-Net + RPN	32.834	0.216
Deblurring U-Net + RPN + SA	30.799	0.342

As expected we see a rise in inference time per image as we add in the RPN and the self-attention layer due to the extra computation associated with the layers. We see that with the RPN added in, our PSNR is better than the baseline and this model outperforms the one with the self-attention layer added in as well.

Several results from the test set

For this section, we shall denote the attention module integrated deblurring net with the term Attention net and the self-attention variant of it with the term Self-attention net.

Figure 5.4 to 5.8 compare and contrast visually across several object classes, the quality of deblurring achieved by the models described in this thesis.



(A) Blurry Image (B) Attention Net

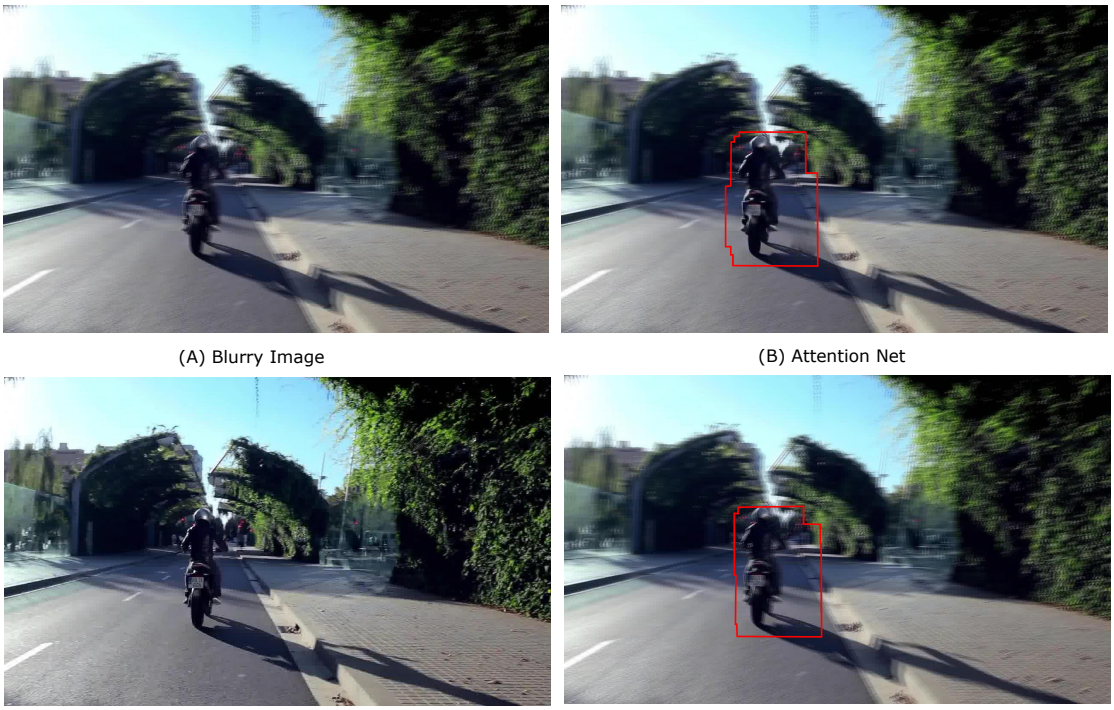
(C) Ground Truth (D) Self-attention Net

Figure 5.4: Object class - Tiger.

(A) Blurry Image (B) Attention Net

(C) Ground Truth (D) Self-attention Net

Figure 5.5: Object class - motorbike.



(A) Blurry Image (B) Attention Net

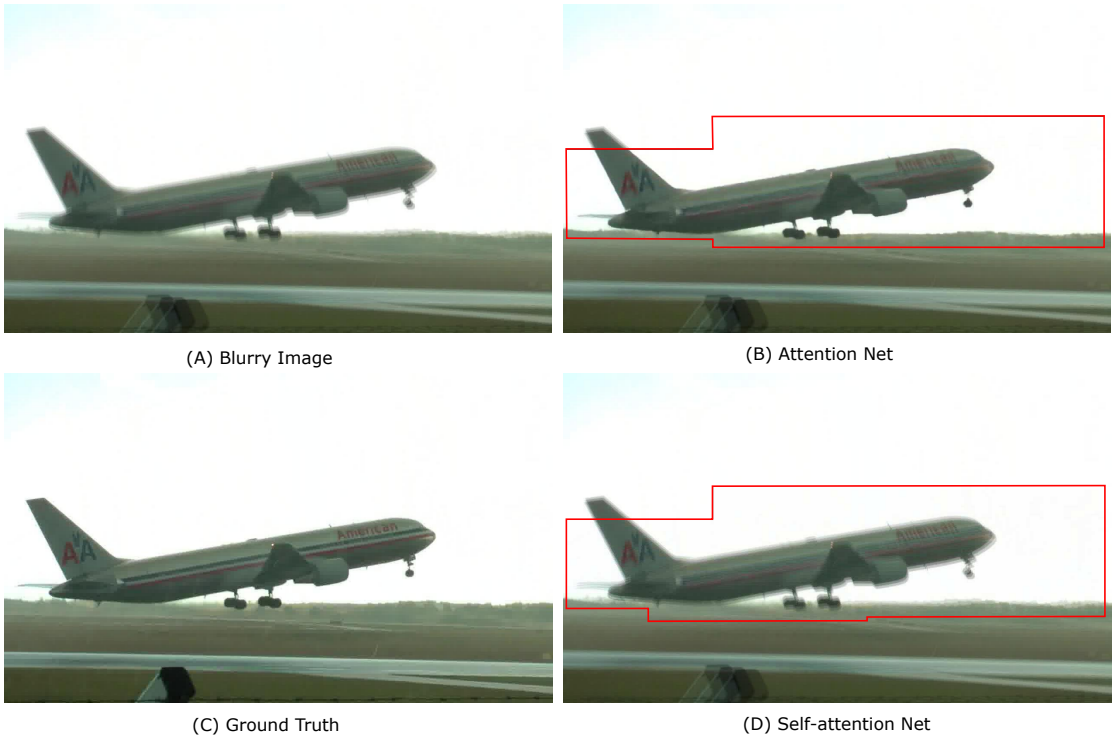
(C) Ground Truth (D) Self-attention Net

Figure 5.6: Object class - motorbike.

(A) Blurry Image (B) Attention Net

(C) Ground Truth (D) Self-attention Net

Figure 5.7: Object class - Bicycle.



(A) Blurry Image (B) Attention Net

(C) Ground Truth (D) Self-attention Net

Figure 5.8: Object class - airplane.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, once we have identified the patches in the blurry image, we go ahead treating the blur by processing all such patches together with the BG blacked out. Even in the latest attention based deblurring works like Human Aware Deblurring Shen *et al.* (2019), once the humans in the seen are masked out as the FG, there are separate branches for treating the FG and BG but all the human patches identified are porcessed together in the FG branch irrespective of the degree of blur the individual patches might have experienced. This leads us to formulate a couple of ideas to further improve attention based deblurring.

6.1 Ideas for future work

Patch-wise processing

In all such attention based approaches, once the patches of interest are identified, instead of processing all the patches together we may characterize each patch and selectively process them in different ways to further improve the sharpness of our outputs.

For example, in a dynamic scene consisting of a multitude of moving objects, it is natural for some to be blurred much more than others. Once our attention module has picked out the object patches, we can characterize the degree of blur in each patch and then depending on the characterization of each patch process them in different ways (different models(encoder-decoder chains) depending on the size of the patch and degree of blur experienced by the object in the patch).

Along with feature sharing between these different encoder-decoder branches a more sharp image can be reconstructed at the end by merging the individual results all together. One of the drawbacks of such a framework would be the increase in model size. Also, we would expect an increase in computation time as instead of a single pass

to deblur the image, each patch has to now be processed separately and then integrated together. This increase in computation time can be mitigated by the following idea.

Custom CUDA Kernels

The idea of spatially adaptive kernels during convolution has been explored recently in Su *et al.* (2019), with the weights of the filter being applied over each pixel during convolution are multiplied by a spatially varying kernel with parameters that can be learned from the local features of the pixel. In our case however, we require each patch to be subject to not only a different filter but a series of such filters depending on the blur we characterize over it.

This can be tackled by developing custom kernels in CUDA. Instead of indexing each patch from the original image and processing them by passing them through different CNNs, we instead would pass all the patches to a custom CUDA kernel which would process certain patches in certain predetermined ways. This way we still process the entire image in a single pass with this custom kernel as it will selectively process each patch in the image as per our design. This will save us the computational time we would have incurred if we had processed the patches separately with different models.

Once the different types of patches and the different ways by which such patches need to be processed are decided upon, implementing such a custom kernel can be explored upon.

6.2 Conclusion

In this thesis, we have thus provided for a reliable framework to synthesize motion blur over an video data of choice and have applied the same to the ILSRVC ImageNet VID data set. This custom data set can be used for further research in the field of attention based deblurring,

The novel attention based deblurring model proposed in the thesis has been shown to outperform the baseline deblurring model without any attention. This paves the way for object targeted deblurring applications using the concept described in this thesis as the cornerstone.

Self-attention has been explored as a viable option in bringing in the global context of an image to improve deblurring. Unfortunately, the results indicate a drop in PSNR with self-attention layer integrated in. We believe a better method in bringing in the global context would increase the receptive field around our regions of interest and result in sharper images. We hope the ideas of patch-wise characterisation followed by processing and custom kernel based methods would inspire future researchers to carry on this idea of object attentive deblurring.

REFERENCES

1. **Brooks, T.** and **J. T. Barron** (2019). Learning to synthesize motion blur. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

2. **Chakrabarti, A.** (2016). A neural approach to blind motion deblurring. *CoRR, abs/1603.04771*. URL `https://arxiv.org/abs/1603.04771`.

3. **Gong, D.**, **J. Yang**, **L. Liu**, **Y. Zhang**, **I. Reid**, **A. v. d. H. C. Shen**, and **Q. Shi** (2017). From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

4. **Hui, J.** (2018). Understanding feature pyramid networks for object detection. URL `https://medium.com/@jonathan_hui/`.

5. **Kingma, D. P.** and **J. Ba** (2014). Adam: A method for stochastic optimization. *CoRR abs/1412.6980*. URL `https://arxiv.org/abs/1412.6980`.

6. **Kohler, R.**, **M. Hirsch**, **B. Mohler**, **B. Scholkopf**, and **S. Harmeling** (2012). Recording and playback of camera shake: Bench marking blind deconvolution with a real-world database. *The IEEE European Conference on Computer Vision (ECCV)*.

7. **Kupyn, O.**, **V. Budzan**, **M. Mykhailych**, **D. Mishkin**, and **J. Matas** (2018). Deblurgan: Blind motion deblurring using conditional adversarial networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

8. **Lin, T.-Y.**, **P. Dollar**, **R. Girshick**, **K. He**, **B. Hariharan**, and **S. Belongie** (2017). Feature pyramid networks for object detection. *The IEEE International Conference on Computer Vision (ICCV)*.

9. **Lu, X.**, **B. Lia**, **Y. Yue**, **Q. Li**, and **J. Yan** (2018). Focal loss for dense object detection. *CoRR, abs/1708.02002*. URL `https://arxiv.org/abs/1708.02002`.

10. **Lu, X.**, **B. Lia**, **Y. Yue**, **Q. Li**, and **J. Yan** (2019). Grid rcnn. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

11. **Mnih, V.**, **N. Heess**, **A. Graves**, and **K. Kavukcuoglu** (2014). Recurrent models of visual attention. *CoRR, abs/1406.6247*. URL `http://arxiv.org/abs/1406.6247`.

12. **Nah, S.**, **T. H. Kim**, and **K. M. Lee** (2017). Deep multi-scale convolutional neural network for dynamic scene deblurring. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

13. **Niklaus, S.**, **L. Mai**, and **F. Liu** (2017). Video frame interpolation via adaptive separable convolution. *The International Conference on Computer Vision (ICCV)*.

14. **Pan, J.**, **Z. Hu**, **Z. Su**, **H.-Y. Lee**, and **M. Yang** (2016). Soft-segmentation guided object motion deblurring. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

15. **Ren, S.**, **K. He**, **R. B. Girshick**, and **J. Sun** (2015). Faster R-CNN: towards real-time object detection with region proposal networks. *The IEEE International Conference on Computer Vision (ICCV)*.

16. **Ronneberger, O.**, **P. Fischer**, and **T. Brox** (2015). U-net: Convolutional networks for biomedical image segmentation. *CoRR abs/1505.04597*. URL `http://arxiv.org/abs/1505.04597`.

17. **Russakovsky, O.**, **J. Deng**, **H. Su**, **J. Krause**, **S. Satheesh**, **S. Ma**, **Z. Huang**, **A. Karpathy**, **A. Khosla**, **M. Bernstein**, **A. C. Berg**, and **L. Fei-Fei** (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, **115**(3), 211–252.

18. **Sahu, S.**, **M. K. Lenka**, and **P. K. Sa** (2019). Blind deblurring using deep learning: A survey. *CoRR, abs/1907.10128v1*. URL `https://arxiv.org/pdf/1907.10128v1.pdf`.

19. **Shen, Z.**, **W. Wang**, **J. Shen**, **H. Ling**, **T. Xu**, and **L. Shao**, Human-aware motion deblurring. *In IEEE International Conference on Computer Vision (ICCV)*. 2019.

20. **Su, H.**, **V. Jampani**, **D. Sun**, **O. Gallo**, **E. Learned-Miller**, and **J. K. Ba** (2019). Pixel-adaptive convolutional neural networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

21. **Sun, J.**, **W. Cao**, **Z. Xu**, and **J. Ponce** (2015). Learning a convolutional neural network for non-uniform motion blur removal. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

22. **Tao, X.**, **H. Gao**, **X. Shen**, **J. Wang**, and **J. Jia** (2018). Scale-recurrent network for deep image deblurring. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

23. **van de Sande, K. E. A.**, **J. R. R. Uijlings**, **T. Gevers**, and **A. W. M. Smeulders** (2011). Segmentation as selective search for object recognition. *The IEEE International Conference on Computer Vision (ICCV)*.

24. **Vaswani, A.**, **N. Shazeer**, **N. Parmar**, **J. Uszkoreit**, **L. Jones**, **A. N. Gomez**, **L. Kaiser**, and **I. Polosukhin** (2017). Attention is all you need. *CoRR, abs/1706.03762*. URL `http://arxiv.org/abs/1706.03762`.

25. **Yang, Z.**, **S. Liu**, **H. Hu**, **L. Wang**, and **S. Lin** (2019). Reppoints: Point set representation for object detection. *The IEEE International Conference on Computer Vision (ICCV)*.