Deep Fourier Up-Sampling in image De-raining and Segmentation

Yiwei Gui
University of Michigan
EECS 556
yiweigui@umich.edu

Hefeng Zhou
University of Michigan
EECS 556
zhouhf@umich.edu

Jiadong Hao
University of Michigan
EECS 556
haojd@umich.edu

Yiting Wang
University of Michigan
EECS 556
yitinw@umich.edu

Honor Code: Our team will give attribution for any figures used in our documents and will cite all code sources (beyond those already built into Julia Base or Matlab toolboxes).

Abstract

Spatial up-sampling is a critical component in many deep learning models for image processing, yet traditional methods such as interpolation only consider local information. This project explores improvements to Deep Fourier Upsampling, a frequency-domain approach that supports up-sampling using the global frequency domain information. We propose several enhanced variants, and evaluate their performance on two tasks: image de-raining with LPNet and image segmentation with DeepLabV3. Experimental results show that our Fourier attention-based variant significantly improves de-raining quality, while both periodic and attention-based Fourier Up-sampling enhance segmentation accuracy. These findings demonstrate the versatility of Fourier-based Up-sampling across diverse vision tasks.

Keywords: Fourier Up-sampling, self attention, image segmentation

1 Introduction

1.1 Overviews

Nowadays, spatial up-sampling is extensively adopted in multi-scale modeling approaches. Traditional spatial up-sampling operators such as interpolation and transposed convolutions heavily rely on local pixel correlations, which often lead to blurred edges or lost high-frequency details. In contrast, Fourier domain inherently supports global modeling, which provides a brand new area for up-sampling. However, up-sampling in the frequency domain faces challenges due to the absence of invariant properties and local texture similarity; using naive interpolation in the spatial domain will cause errors.

1.2 Related work

Spatial Up-sampling Spatial down- and up-sampling operations form the backbone of many modern convolutional neural network architectures in computer vision tasks. Models like U-Net [1] construct multi-scale representations by down-sampling in the encoder and then restoring spatial resolution through up-sampling in the decoder. Similarly, architectures utilizing feature pyramids [2][3][4][5] and image pyramids [6][7][8] rely heavily on spatial up-sampling to achieve multi-scale feature fusion. However, previous up-sampling methods only take effect on the spatial domain and focus primarily on local pixel relationships, largely overlooking the potential of performing up-sampling in the frequency domain, where global context modeling is naturally supported.

Spatial-Fourier Interaction Previous work has integrated Fourier transforms into deep neural networks to harness frequency-domain information. Some approaches apply the discrete Fourier transform (DFT) to convert spatial features into the frequency domain, allowing models to leverage spectral characteristics to enhance performance [9][10]. Others utilize the convolution theorem to accelerate computation via fast Fourier transform (FFT). For instance, FFC [11] replaces conventional convolution with a hybrid spatial-Fourier interaction. Similarly, spectral pooling proposed in [12] reduces feature resolution by truncating high-frequency components in the Fourier domain. However, these methods typically perform spatial-Fourier interaction at a single resolution and do not address the more complex challenge of multi-resolution frequency-domain modeling.

1.3 Solutions and Advantages

Solutions The paper "Deep fourier Up-sampling" developed FourierUp, which is a learnable operator that can be directly integrated into existing neural networks. It works by first transforming a low-resolution feature map from the spatial domain into the Fourier domain using a 2D discrete Fourier transform. Once in the Fourier domain, specialized upsampling techniques—such as 1)periodic padding, 2)area interpolation, or 3)corner interpolation—are applied to increase the resolution of the frequency representation according to theoretical transform rules verified in the paper [13]. Finally, an inverse Fourier transform is performed to convert the enhanced frequency map back into the spatial domain, yielding a high-resolution feature map that retains both the global context and the fine details of the original image.

Advantages FourierUp offers a powerful alternative to conventional spatial up-sampling by leveraging the global modeling capabilities inherent to the frequency domain. Unlike traditional methods that rely on local pixel relationships Fourier up-sampling operates on a global scale, it captures long-range dependencies and multi-scale frequency patterns through the spectral convolution theorem. This enables a more robust integration of features across different resolutions, which is particularly beneficial for complex computer vision tasks that demand both precise local detail and coherent global context. Furthermore, while previous approaches have limited the interaction between spatial and Fourier representations to a single resolution scale, Deep Fourier Up-Sampling extends this capability across multiple scales. This multi-scale fusion enhances the recovery of fine textures and object boundaries and thus improves overall performance. Also, FourierUp can be easily integrated into existing networks and improves performance across diverse applications.

2 Quantitative Performance Prediction

Our experiments consist of two stages. In stage 1, we will modify the FourierUp components and apply them to LPNet[14], a model for image de-raining, to verify the improvements. In stage 2, we will apply the modified FourierUp sampling modules to image segmentation using DeepLabv3[15] as the base model, which is not mentioned in the original paper [13], to verify the versality of the modules.

Stage 1 Aligned with the original paper [13], we use the Rain200H dataset [16] as the testing sets. To evaluate the performance of the models, we employ Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM), which are widely used in image restoration tasks.

PSNR is defined as:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{\text{MSE}} \right) \tag{1}$$

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

where:

- MAX_I is the maximum possible pixel value of the image.
- The Mean Squared Error (MSE) is defined as the average squared difference between the original image I and its approximation K:

SSIM is defined as:

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{3}$$

where:

- μ_x, μ_y are the pixel sample mean of x and y;
- σ_x^2, σ_y^2 are the sample variances of x and y;
- σ_{xy} is the sample covariance between x and y;
- $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ two variables to stabilize the division with weak denominator, where L is the dynamic range of pixel values.

We expect our variants to outperform the original FourierUp whose performance is shown in Fig. 1. The four model settings are defined as follows [13]:

(1) Original: the baseline without any changes;

(2) FourierUp-AreaUp: replacing the original model's spatial up-sampling with the union of the Area-Interpolation variant of our FourierUp and the spatial up-sampling itself;

(3) FourierUp-Padding: replacing the original model's spatial up-sampling operator with the union of the Periodic-Padding variant of our FourierUp and the spatial up-sampling itself;

(4) Spatial-Up: replacing the variants of FourierUp in configurations 2 and 3 with spatial up-sampling.

Model	Configurations	Rain200H		Rain200L	
		PSNR	SSIM	PSNR	SSIM
LPNet	Original	22.907	0.775	32.461	0.947
	Spatial-Up	22.956	0.777	32.522	0.950
	FourierUp-AreaUp	<u>22.163</u>	<u>0.783</u>	<u>32.681</u>	<u>0.954</u>
	FourierUp-Padding	**23.295**	**0.786**	**32.835**	**0.956**

Figure 1: Performance comparison of LPNet with different up-sampling methods.

Stage 2 To evaluate our modified FourierUp on DeepLabv3, we choose the VOC2012 [17] as the dataset. We use accuracy, mean accuracy, and mean intersection over union (mIoU) as the evaluation metrics.

The Accuracy is defined as:

$$\text{Acc} = \frac{\sum_{c=1}^{C} TP_c}{\sum_{c=1}^{C} (TP_c + FN_c)} \tag{4}$$

The Mean Accuracy is defined as:

$$\text{MeanAcc} = \frac{1}{C}\sum_{c=1}^{C}\text{Acc}_c = \frac{1}{C}\sum_{c=1}^{C}\frac{TP_c}{TP_c + FN_c} \tag{5}$$

- TP_c is the number of true positives for class c;
- FN_c is the number of false negatives for class c;
- C is the total number of semantic classes.

The mIoU is defined as:

$$\text{IoU}_c = \frac{TP_c}{TP_c + FP_c + FN_c} \tag{6}$$

$$\text{mIoU} = \frac{1}{C}\sum_{c=1}^{C}\text{IoU}_c \tag{7}$$

where:

- TP_c is the number of true positives for class c;
- FP_c is the number of false positives;
- FN_c is the number of false negatives;
- C is the total number of semantic classes.

Based on our integration of deep Fourier up-sampling via a fusion module, we anticipate at least a 3% improvement over the baseline DeepLabV3+, which achieves around 90% on the VOC2012 testing dataset at 10,000 epochs.

3 Methods and Theory

3.1 Mathematical formula of the Fourier-Up variants

Below shows the mathematical theory behind the three variants of the Deep Fourier Up-sampling[13], including 1) periodic padding, 2) area interpolation and 3) corner interpolation.

Definitions $f(x, y) \in \mathbb{R}^{2M\times 2N}$ is the 2-times zero-inserted up-sampled version of $g(x, y) \in \mathbb{R}^{M\times N}$ in the spatial domain. Let $F(u, v) \in \mathbb{R}^{2M\times 2N}$, $G(u, v) \in \mathbb{R}^{M\times N}$ denote their corresponding Fourier transforms. Let $H(u, v) \in \mathbb{R}^{2M\times 2N}$ be the 2-times area-interpolation up-sampled Fourier transform of $G(u, v)$, and $h(x, y) \in \mathbb{R}^{M\times N}$ denote its inverse Fourier transform.

Periodic padding $F(u, v) = F(u + M, v) = F(u, v + N) = F(u + M, v + N)$ and $G(u, v) = \frac{F(u,v)}{4}$, where $u = 0, 1, 2, \ldots, N - 1$ and $v = 0, 1, 2, \ldots, M - 1$. Then $F(u, v)$ is the periodic padding of $G(u, v)$, where $G(u, v)$ is exactly one quarter of $F(u, v)$ with the value being $\frac{1}{4}$ times decayed.

Area interpolation Let $H(2u, 2v) = H(2u + 1, 2v) = H(2u, 2v + 1) = H(2u + 1, 2v + 1) = G(u, v)$ with $u = 0, 1, \ldots, M - 1$ and $v = 0, 1, \ldots, N - 1$. Then

$$\begin{aligned}
h(x, y) &= \frac{A(x, y)}{4}\, g(x, y) \\
h(x + M, y) &= \frac{A(2M - x, y)}{4}\, g(x, y) \\
h(x, y + N) &= \frac{A(x, 2N - y)}{4}\, g(x, y) \\
h(x + M, y + N) &= \frac{A(2M - x, 2N - y)}{4}\, g(x, y)
\end{aligned} \tag{8}$$

where $A(x,y) = 1 + e^{\frac{i\pi x}{M}} + e^{\frac{i\pi y}{N}} + e^{i\pi\left(\frac{x}{M}+\frac{y}{N}\right)}$, and $x = 0, 1, \ldots, M-1$, $y = 0, 1, \ldots, N-1$.

Corner interpolation

Theorem-1. Suppose the shifted F_G^{shift} of the Fourier map $G \in \mathbb{R}^{M\times N}$ as

$$F_G^{shift}(u', v') = G(u - \frac{M}{2}, v - \frac{N}{2}), \tag{9}$$

where $u' = 0, 1, \ldots, M-1$ and $v' = 0, 1, \ldots, N-1$, it holds that the inverse Fourier transform f_g^{shift} of F_G^{shift}

$$f_g^{shift}(x,y) = (-1)^{(x+y)} g(x,y), \tag{10}$$

where $x = 0, 1, \ldots, M-1$ and $y = 0, 1, \ldots, N-1$.

Theorem-2. Suppose the corner interpolated F_G^{cor} of the Fourier map $G \in \mathbb{R}^{M\times N}$, it holds that the inverse Fourier transform f_g^{cor} of F_G^{cor}

$$f_g^{cor}(x,y) = g\left(\frac{x'}{2}, \frac{y'}{2}\right) \exp\left(j\pi\left(\frac{x'}{2} + \frac{y'}{2}\right)\right) \frac{(-1)^{(x+y)}}{4}, \tag{11}$$

where $x' = 2x$ and $y' = 2y$, $x = 0, 1, \ldots, M-1$ and $y = 0, 1, \ldots, N-1$.

Note that rather than directly using the transformation rules derived above, the original paper apply some learnable mechanics like using a 1×1 convolution filter to make the transformation more flexible.

3.2 Innovated Fourier-Up Variants

Based on the original three variants of Fourier-Up variants, we innovate new variants to further improve their efficiency. Basically, our modifications can be seperated into three categories according to the module architecture as shown in 2.

1) **Original Fourier Up**: The baseline without any changes.
2) **Fourier_Up_Tuned**: Fine-tuned original Fourier Up module.
   pad_theory: To verify the effectiveness of learnable convolution over fixed mapping, this variant replaces the learnable convolution kernel by the fixed value 1/4 stated in formula 3.1.

   pad_larger_kenel: The Fourier transform operates globally, meaning that each point in the Fourier spectrum is influenced by information from the entire input image. By incorporating larger convolution kernels, the system may become more robust to noise. Hence for this variant, we replace the 1*1 convolution to 3*3 with padding 1.
3) **Fourier_Up_Stacked**: Each of the three original variants has their own strength and weakness as stated in [13]. For example, the periodic-padding variant, while effective in maintaining structural consistency, is prone to introducing boundary artifacts. On the other hand, the area-interpolation variant, though better at preserving high-frequency details, may result in the distortion of fine image structures. In order to leverage the stregth of different variants, we stack two different Fourier Up variants and combine those result with the original upsampling using a 1*1 convolution layer. We build *pad_corner* and *pad_area*, which combines periodic padding with corner interpolation and area interpolation respectively.
4) **Fourier_Up_Atten**: Through experiments, we noticed that not all frequencies contribute equally to image reconstruction – some frequency bands are more important for preserving texture, while some nearly contribute nothing. Self-attention can learn long-range dependencies across the frequency map, allowing the module to focus on critical frequencies and suppress less useful ones, which greatly leverages the global nature of the Fourier domain.

   Technically, we first apply a 1*1 convolution to project each RGB channel into

three separate feature maps (query, key and value) and flatten them, hence each query, key and value has a dimension of *[b (batch), c (channel), h*w (height*width)]*. Then we apply batch matrix multiplication to the permuted query (*[b, h*w, c]*) and the key (*[b, c, h*w]*), then apply the softmax to get the attention score (*[b, h*w, h*w]*). Finally, we perform batch matrix multiplication between the attention score and permuted value (*[b, h*w, c]*) and reshape the output to *[b, c, h, w]*.

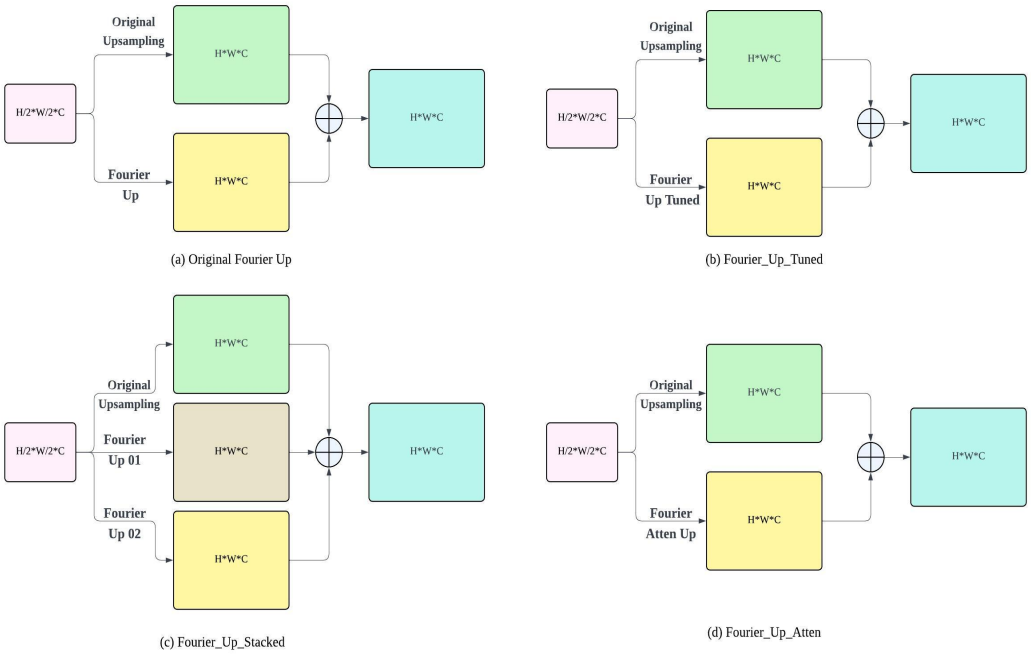


Figure 2: The architecture of the three configurations

3.3 LPNet architecture

The architure of LPNet for image deraining is shown in 3. It first decomposes the rainy input into a multi-scale Laplacian pyramid and feeds each level through a "convolution+residual" sub-network to predict the clean high-frequency detail at that scale; these cleaned Laplacian levels are then up-sampled and summed into a Gaussian pyramid, whose base image is the final derained output. In total, there are four Up-sample modules where we apply our Fourier-Up Variants.

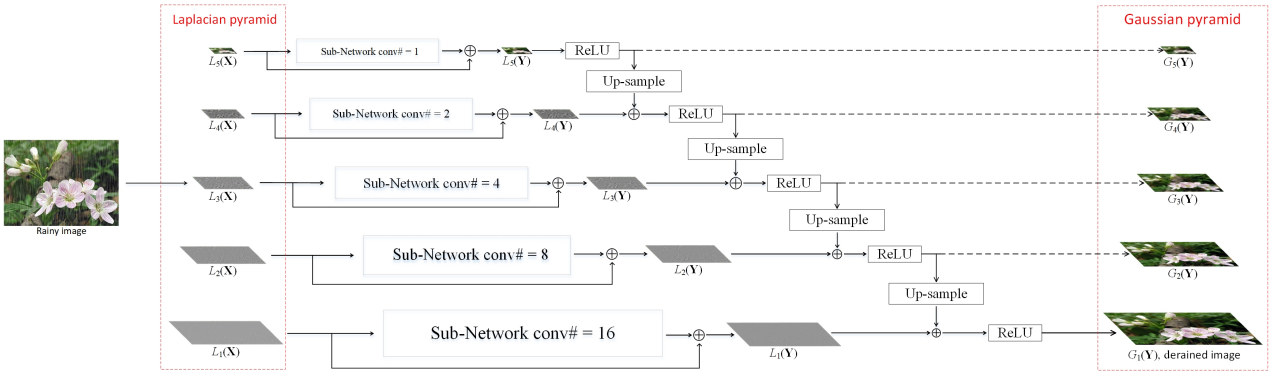


Figure 3: LPNet Architecture

3.4 DeepLab architecture

The architecture of DeepLabv3 for image segmentation is shown below 4. The two marked "Upsample by 4" block is where we apply our Fourier-Up Variants.

Since all the variants are designed for $\times 2$ up-sampling, to match the feature map size, we use the same component twice. However, the original bilinear interpolation in DeepLabv3 generates 513×513 feature maps while our variants gives 512×512. Hence we just apply the nearest neighbor interpolation to fill the last pixel. Due to the extremely high dimension (256 channels), we apply average or max pooling to the keys and values to shrink the size of each feature map to half.

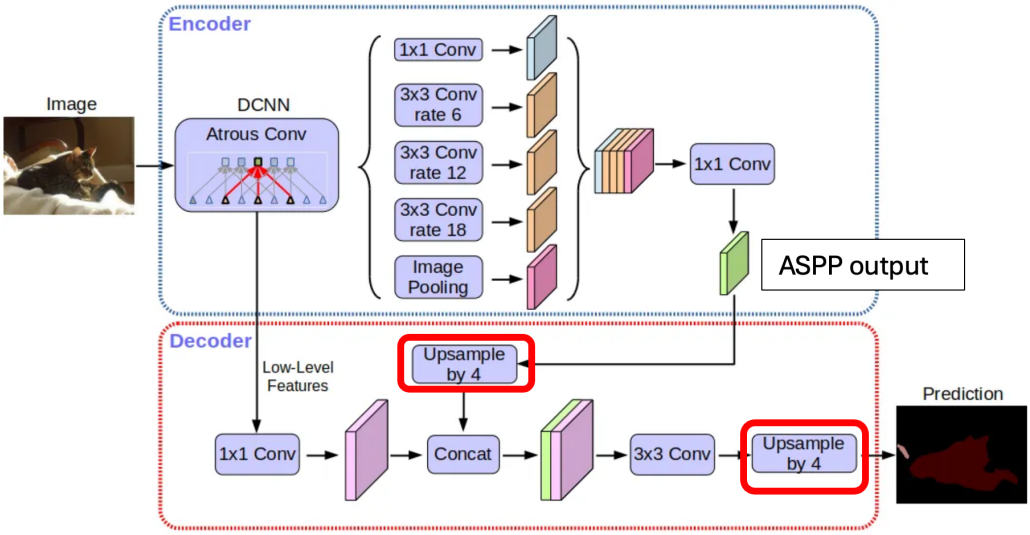


Figure 4: DeepLabV3 Architecture

4 Experimental results

4.1 Stage 1: Modify Fourier-Up variants (LPNet for image de-raining)

4.1.1 Dataset

Training: RainTrainH [13] (1542 rained and ground truth image pairs)
Testing: Rain200H [13] (200 rained and ground truth image pairs)

4.1.2 Training PSNR diagram

As shown in 5, we have three main observations:
1. *pad_attention* (red) outperforms the others most of the time, which indicates the attention machanism can help to generate meaningful frequency representation much more efficiently than convolution operation.
2. *pad_area* (blue) and *pad_corner* (green) are slightly above the *pad* (yellow), suggesting that stacking two variants can help to improve the results.
3. *pad_theory* (purple) has the lowest performance among all variants, which means that using learnable parameter gives much more flexibility than using the fixed transition rule.

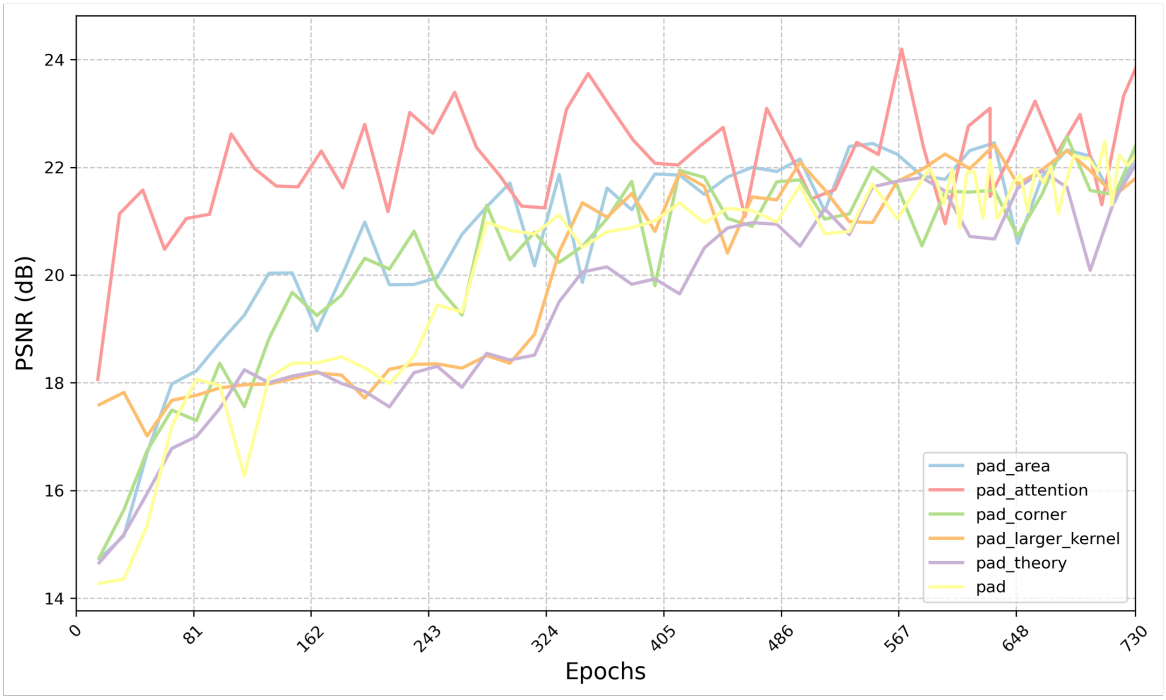


Figure 5: Training results for different modified Fourier-up variants

4.1.3 Fine-tuning of pad_atten

We have tried multiple hyperparameter settings of the pad_atten variant, and in the training PSNR diagram above we just show the best model we have. The performance is measured by the average PSNR in the last 200 epochs in the following table1.

From Exp a,b,c, we determine the best dropout rate to be 0.3. From Exp b,d,e, we determine best no. of heads is 1. As the no. of heads increase, the training accuracy increase but for the test accuracy, it drops significantly, which infers that overfitting happens. From Exp b,g,h, we find the best learning rate is 0.01. Hence, the best model hyperparameter setting is the setting in Exp b.

Table 1: Quantitative testing result comparison of image de-raining.

Exp no.	**no.of heads**	**dropout**	**learning rate**	**Avg PSNR**
a	1	0.1	0.01	21.6784
b	1	0.3	0.01	**22.7907**
c	1	0.5	0.01	22.1276
d	4	0.3	0.01	21.3468
e	8	0.3	0.01	19.5392
g	1	0.3	0.02	22.0962
h	1	0.3	0.05	22.1892

4.1.4 Testing results

During the 730 training epochs, we save the best model and apply them to the test set. The results are shown in table 2.

Table 2: Quantitative testing result comparison of image de-raining.

Model	**Configurations**	**Rain200H** PSNR	**Rain200H** SSIM
LPNet	**pad_attention**	**23.5167**	**0.7957**
	pad_theory	22.3602	0.7453
	pad	22.6593	0.7486
	pad_larger_kernel	22.8351	0.7453
	pad_corner	23.0357	0.7563
	pad_area	22.8203	0.7497

4.1.5 Visualization of de-raining results

(a) is the rained picture, (b) is the ground truth, (c), (d), (e) and (f) are the results that using the original *pad*, *pad_larger_kernel*, *pad_corner* and *pad_attention* variants respectively.

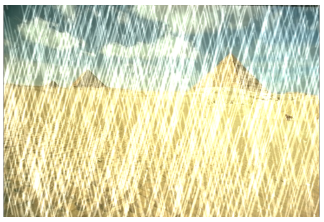
For (b), we observed that the original *pad* method has poor performance in removing raindrops. The details on sky, pyramid, and desert are also highly blurry. Additionally, the color is of high deviation to the ground truth, much dimmer than others.

For (d), it maintains the original yellow and green colors of the ground truth. But it performs badly in removing the raindrops, especially for the pyramid and desert.

For (e), compared with (c), it can reduce more rain, but the color of it dimmer than the ground truth.

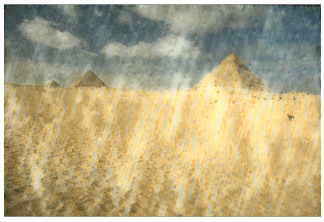
From (f), we can see *pad_attention* method performs very well, it removes almost 80-90% of the raindrops and greatly preserves original color of the ground truth.

In conclusion, by comparing four methods, the *pad_attention* performs the best. It can not only preserve the color but also the details. *pad_larger_kernel* performs well on preserving the color and *pad_corner* performs well on de-raining.

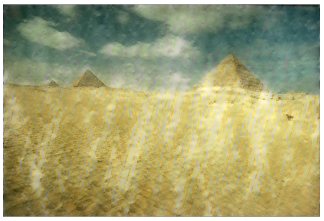


(a) Rainy image

(b) Ground Truth

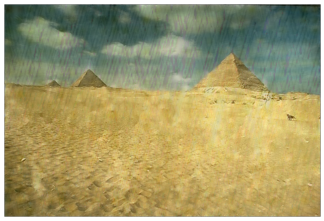


(c) Pad



(d) Pad Larger Kernel

(e) Pad Corner



(f) Pad attention

4.2 Stage 2: Apply Fourier-Up variants to image segmentation (DeepLabv3)

4.2.1 Dataset

Training: VOC2012 + Augmented (10,582 images for training and 1,449 images for validation) [17]
Testing: VOC2012 (1,449 images) [17]

4.2.2 Training performance diagram

As shown in 7, the overall accuracy and mean IoU of both the *pad* (blue) and the *pad_attention* (green) outperforms the original DeepLabv3 using spatial up-sampling, which proves the effectiveness of the

Fourier-Up variants in improving the model performance of various image processing tasks.
Another interesting observation is that during the first half of training, the performance of *pad_attention* is worse than the other two. This is probably because attention incorporates much more parameters, which makes it hard to train in the initial stage. This also explains why at the end of 10,000 iterations the *pad_attention* doesn't outperform the *pad* as stage 1 shows. Original DeepLabv3 is trained for 30,000 iterations to converge, and due to the time limitation, we only train 10,000 iterations for comparison. We believe *pad_attention* will outperform *pad* if trained enough iterations.

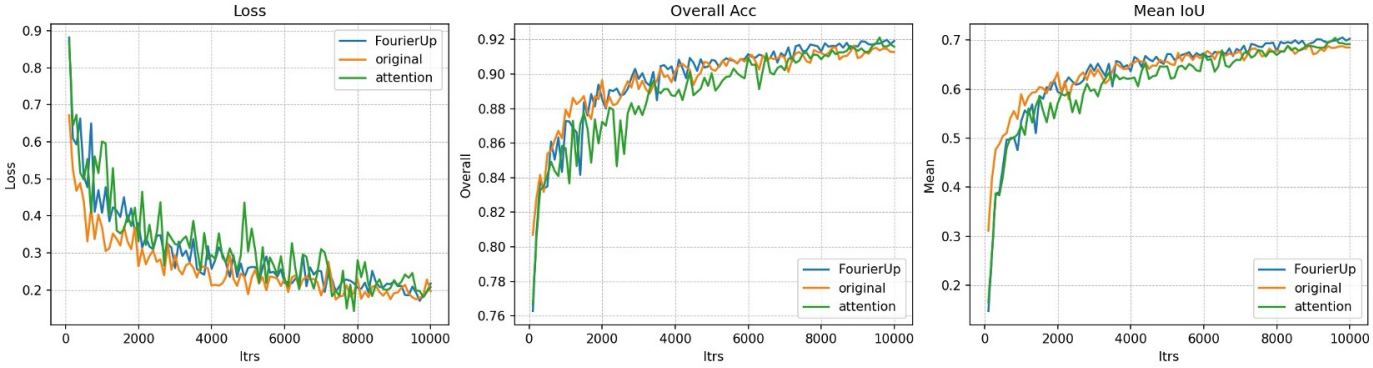


Figure 7: Comparison of three settings' training performance

4.2.3 Testing results

During the 10,000 training iterations, we save the best model and apply them to the test set. The results are shown in table 3.

Table 3: Quantitative Comparisons of DeepLabV3 with different Upsampling module.

Upsample Methods	**Overall Accuracy**	**Mean Accuracy**	**Mean IoU**
Original	0.9113	0.8157	0.6775
pad	**0.9215**	**0.8363**	**0.7106**
pad_attention	0.9202	0.8360	0.7047

4.2.4 Visualization of segmentation results

(b) is the ground truth picture, while (a) is the corresponding target segmentation result. Basically, there are four regions to segment, the left and right chairs, the child and the table.

For (c), we observed that the original DeepLabv3 performs poorly in segmentation of the two chairs. It just detects a small portion of the left chair while completely ignores the right one. And it cannot give a good segmentation on the child's arm on the left.
For (d), we can observe that the *pad_attention* method works better than the original model, especially the detail of the left chair and arm of the child, but it ignores the right chair as well.

For (e), we can see that the *pad* method does the best, because it does detect the details of the child, table and both chairs, though the right one is incomplete.

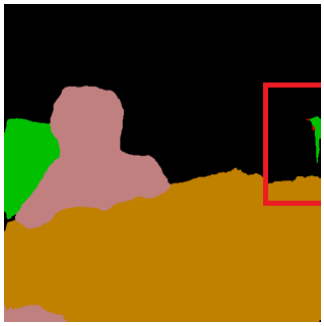
In conclusion, from both the testing scores and the visualization, the *pad* and the *pad_attention* methods do better on image segmentation than the vanilla DeepLabv3.

(a) Target Performance

(b) Ground Truth

(c) Original

(d) Attention



(e) Fourier Pad

5 Conclusions and future work

In this project, we explored enhancements to Deep Fourier Up-sampling and demonstrated its versatility across two key computer vision tasks: image de-raining and semantic segmentation. By proposing several novel variants—such as stacked Fourier modules and frequency attention—we extended the theoretical and practical capabilities of FourierUp beyond its original form.

Experimental results on the Rain200H dataset revealed that our attention-enhanced variant (`pad_attention`) consistently outperformed others in terms of PSNR and SSIM, validating the hypothesis that global frequency-aware modeling improves restoration quality. Likewise, applying FourierUp variants to DeepLabv3 segmentation on the VOC2012 dataset yielded higher overall accuracy and mIoU than baseline bilinear up-sampling.

These results underscore the generality and robustness of Fourier-based up-sampling methods when integrated into deep neural networks. The Fourier domain's ability to globally model long-range dependencies proves valuable in preserving fine image details and improving structural consistency.

Our code is publicly available at: https://github.com/Verbsius/EECS556_25WN

Future Work

Several promising directions remain for future research:

1. **Extended Training for Attention Modules:** Our attention-based variants underperformed the pad variant at 10,000 training epoch in image segmentation. We believe running extended training (e.g., 30,000+ iterations) could unlock their full potential.
2. **Multi-Scale Fourier Attention:** Inspired by pyramid and transformer-based models, developing hierarchical frequency attention modules could enhance multi-resolution fusion across scales.
3. **Cross-Domain Applications:** While we focused on de-raining and segmentation, applying FourierUp to tasks like super-resolution, depth estimation, or medical imaging could further validate its generality.

4. **Hardware Efficiency:** Future work may explore low-rank or quantized versions of the Fourier attention modules to reduce computational overhead on edge devices, especially for the pad_attention variant.

Overall, our innovations in Fourier-based up-sampling demonstrate a strong foundation for future improvements in both performance and applicability across diverse vision problems.

References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[2] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 2017. [Online]. Available: https://arxiv.org/abs/1612.03144

[3] S.-W. Kim, H.-K. Kook, J.-Y. Sun, M.-C. Kang, and S.-J. Ko, "Parallel feature pyramid network for object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. [Online]. Available: https://openaccess.thecvf.com/content_ECCV_2018/papers/Seung-Wook_Kim_Parallel_Feature_Pyramid_ECCV_2018_paper.pdf

[4] S. S. Seferbekov, V. I. Iglovikov, A. V. Buslaev, and A. A. Shvets, "Feature pyramid network for multi-class land segmentation," *CoRR*, vol. abs/1806.03510, 2018. [Online]. Available: http://arxiv.org/abs/1806.03510

[5] L. Zhu, Z. Deng, X. Hu, C.-W. Fu, X. Xu, J. Qin, and P.-A. Heng, "Bidirectional feature pyramid network with recurrent attention residual modules for shadow detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. [Online]. Available: https://openaccess.thecvf.com/content_ECCV_2018/papers/Lei_Zhu_Bi-directional_Feature_Pyramid_ECCV_2018_paper.pdf

[6] Y. Pang, T. Wang, R. M. Anwer, F. S. Khan, and L. Shao, "Efficient featurized image pyramid network for single shot detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. [Online]. Available: https://arxiv.org/abs/1910.07721

[7] Z. Liu, G. Gao, and L. Sun, "Ipg-net: Image pyramid guidance network for object detection," *CoRR*, vol. abs/1912.00632, 2019. [Online]. Available: http://arxiv.org/abs/1912.00632

[8] J. Luo, J. Liu, J. Lin, and Z. Wang, "A lightweight face detector by integrating the convolutional neural network with the image pyramid," *Pattern Recognit. Lett.*, vol. 133, pp. 180–187, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:216349657

[9] Y. Yang and S. Soatto, "FDA: fourier domain adaptation for semantic segmentation," *CoRR*, vol. abs/2004.05498, 2020. [Online]. Available: https://arxiv.org/abs/2004.05498

[10] J.-H. Lee, M. Heo, K.-R. Kim, and C.-S. Kim, "Single-image depth estimation based on fourier domain analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018/papers_backup/Lee_Single-Image_Depth_Estimation_CVPR_2018_paper.pdf

[11] L. Chi, B. Jiang, and Y. Mu, "Fast fourier convolution," in *Neural Information Processing Systems*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:227276693

[12] O. Rippel, J. Snoek, and R. P. Adams, "Spectral representations for convolutional neural networks," 2015. [Online]. Available: https://arxiv.org/abs/1506.03767

[13] M. Zhou, H. Yu, J. Huang, F. Zhao, J. Gu, C. C. Loy, D. Meng, and C. Li, "Deep fourier up-sampling," *Advances in Neural Information Processing Systems*, vol. 35, pp. 22 995–23 008, 2022. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/91a23b3e6a2ebaad62e17d0269f88c6b-Paper-Conference.pdf

[14] X. Fu, B. Liang, Y. Huang, X. Ding, and J. Paisley, "Lightweight pyramid networks for image deraining," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 1794–1807, 2020. [Online]. Available: https://arxiv.org/abs/1805.06173

[15] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017. [Online]. Available: https://arxiv.org/abs/1706.05587

[16] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive image deraining networks: A better and simpler baseline," 2019. [Online]. Available: https://arxiv.org/abs/1901.09221

[17] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge 2012 (voc2012) results," http://host.robots.ox.ac.uk/pascal/VOC/voc2012/, 2012.