Image Deblurring and Denoising Using Deformable Convolution and Large Kernel Attention

Abhishek Sanjay Thosar
athosar@buffalo.edu
University at Buffalo, SUNY

Rishi Jay Joshi
rishijay@buffalo.edu
University at Buffalo, SUNY

Sai Mahesh Pullagura
spullagu@buffalo.edu
University at Buffalo, SUNY

Abstract- Image Deblurring and Image Denoising are well-known problems in Image Restoration. In this paper, we demonstrate a Deformable Convolutional Neural Network[1][2] along with a Large Kernel Attention (LKA) module[10][11][12] for image restoration tasks. Previously, Deformable CNN has produced good results in object detection and image segmentation tasks. It uses spatial sampling locations for learning offsets[1][3] without any additional supervision. LKA learns 2D structure of the images which enables self-adaptive, long-range correlations[5][10][11] when performing self-attention. This work discusses application of Deformable CNNs and LKA modules based on the ResNet backbone architecture. The combination of both of these techniques were yet to be applied to image restoration problems. Our extensive experiments on this novel model has shown comparable performance results in restoring the image from noisy and blurred data. Code: -

Keywords—image restoration, deblurring, denoising, gopro, sidd, convnets, LKA, deformable cnn, PSNR, SSIM

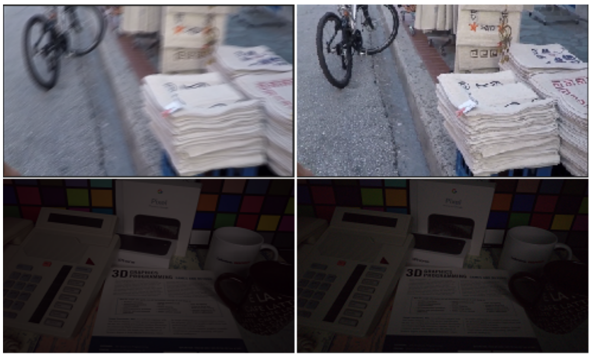


Fig. 1. Image deblurring and image denoising

I. Introduction

Images play a huge role in the daily lives of people and in various industries ranging from photography to space exploration. Scientists and analysts study images obtained from different sources to gain insights about the world, its elements and the universe we inhabit. However, even after having access to state-of-the-art cameras and huge advances in image processing techniques, it is possible to have images that are blurry and noisy. Images from harder to reach areas like outer space are more prone to these defects. It thus becomes difficult to study and analyze these images.

In this paper, we propose and implement a Deep Learning model that performs image restoration tasks on the GoPro and SIDD datasets. These are image deblurring and image denoising. We aim to achieve results that can be easily interpreted and analyzed for scientific purposes as well as for preserving fond memories.

A Convolution Neural Network (CNN)[3][6] is excellent for image recognition but doesn't work so well for geometric transformation. Geometric Transformations transform the positions and orientation of an image to another position and orientation. A Deformable Convolution Neural Network is used in our model, particularly for blurred images. It will help us identify geometric transformations and approximate objects in blurred images. We also use a Large Kernel Attention mechanism which would allow our model to learn the areas where attention needs to be paid in each image. .

II. Related Work

A. Multi-Stage Progressive Image Restoration[13]

This paper proposes a multistage architecture with an encoder and a decoder that progressively learns restoration functions for the input images. The first two stages use the encoder and decoder to extract multistage contextual information. The last stage operates on the original input image resolution and preserves the spatial features. An attention module is also used at every two stages. However, multi-state techniques for image restoration may still yield subpar results.

B. Restormer: Efficient Transformer for High-Resolution Image Restoration[18]

This paper proposes the use of Restormer, an encoder-decoder transformer with design changes in the multi-head attention and feedforward network for performing image restoration tasks. A multi-Dconv head transposed attention module (MDTA) and a gated-Dconv feed-forward network is used.

C. Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections[19]

This paper proposes a "very deep network architecture" consisting of a chain of convolution and deconvolution layers. Convolution layers act as the encoder and deconvolution layers act as the decoder. Skip connections are added to divide the network into several blocks to backpropagate the gradients hence helpling to better train the deep network.

D. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising[20]

This paper proposes the use of residual learning and batch normalization to speed up the training process of a feed-forward denoising convolutional neural network (DnCNNs) and boost denoising performance. The network architecture is designed by modifying the VGG network[24]. It consists of a series of Conv+ReLU and Conv+BN+ReLU layers. This allows the DnCNN to separate the image structure from noise through hidden layers. The depth of the network is set based on the effective patch sizes used in state-of-the-art denoising methods.

E. *Laplacian Filter*

The laplacian filter is used for image sharpening and edge detection. It detects edges in the entire image at once and is applied to an image that has been smoothed with some gaussian smoothing filter.

F. *Bilinear and Bicubic Interpolation*[21]

Bilinear and Bicubic Interpolation are nearest neighbor interpolation techniques. Bilinear interpolation considers the nearest 2x2 pixel values around the unknown pixel. The weighted average of these 4 pixels gives us the final interpolated value and thus a smoother looking image compared to the nearest neighbor. Bicubic interpolation, on the other hand, considers the nearest 4x4 images. In a total of 16 pixels, closer pixels are given more preference. Bicubic interpolation performs better than bilinear interpolation and is the ideal method.

III. Dataset And Features

A. *GoPro*[22]

We evaluate our model performance on blurred images by using the GoPro dataset. GoPro is an image deblurring dataset consisting of 3,214 images, which are divided into 2,103 training images and 1,111 test images. All of these are blurred. The dataset also provides us with pairs of blurred and sharp images which act as the ground truth. The images are of dimension 1280 x 720.

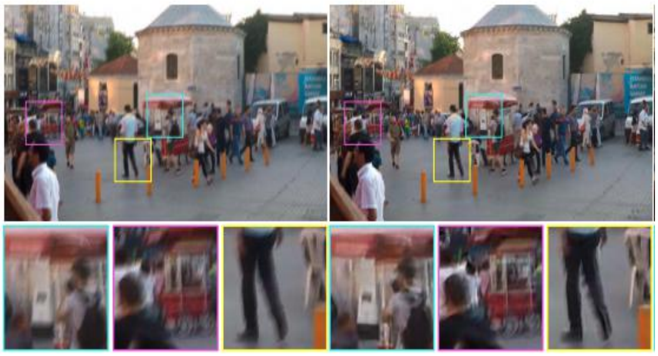


Fig. 2. GoPro Dataset

B. *Smartphone Image Denoising Dataset*[23]

The denoising performance of our mode is evaluated using the SIDD dataset. SIDD contains 30,000 noisy images from 10 scenes under different lighting conditions using five representative smartphone cameras. Ground truth images are provided along with the noisy images.

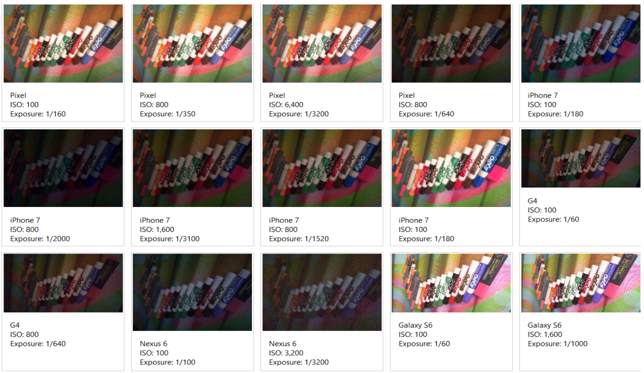

Pixel ISO: 100 Exposure: 1/160
Pixel ISO: 800 Exposure: 1/350
Pixel ISO: 6,400 Exposure: 1/3200
Pixel ISO: 800 Exposure: 1/640
iPhone 7 ISO: 100 Exposure: 1/180
iPhone 7 ISO: 800 Exposure: 1/2000
iPhone 7 ISO: 1,600 Exposure: 1/3100
iPhone 7 ISO: 800 Exposure: 1/1520
iPhone 7 ISO: 100 Exposure: 1/180
G4 ISO: 100 Exposure: 1/60
G4 ISO: 800 Exposure: 1/640
Nexus 6 ISO: 100 Exposure: 1/100
Nexus 6 ISO: 3,200 Exposure: 1/3200
Galaxy S6 ISO: 100 Exposure: 1/60
Galaxy S6 ISO: 1,600 Exposure: 1/1000

Fig. 3. SIDD

IV. Theory

A. *Deformable Convolutional Neural Network*[1]

Convolutional Neural Networks is a very popular deep learning model[3][6] which is commonly used in image/object recognition and classification. Yet, the performance of the model is not great in terms of geometric transformations in object scale, pose and part deformation. Thus, we make use of a slightly modified model of CNN which is capable of learning such geometric transformations for the given data, known as a Deformable Convolution Model.

In deformable convolutions, along with the standard convolution operation, we add a 2-D offset value which helps in factoring the scale of different objects and its respective receptive fields. This allows us to deform the constant receptive field received from the prior activation units and make the offsets learn-able from the preceding feature maps using additional convolutional layers. The addition of the new parameter to the existing model will be less and also it minimizes the computation cost incurred from performing back propagation.

A regular convolution model[6] comprises two steps:

- Use a rectangular kernel for sampling a small region of the feature map .
- Multiplying weights of the rectangular kernel with sampled values and adding them across the kernel gives a single scalar value.

Let K be a 3x3 kernel that samples a small region of the input feature map.

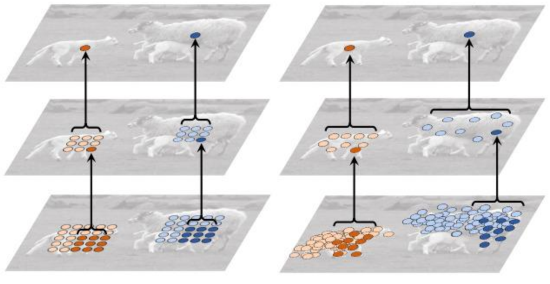
$$K = \{(1,1), (1,0), \ldots, (0,1), (1,1)\} \tag{1}$$

Then the equation of the normal 2d convolution operation will be given as seen below where w is the weights of the kernel, x is the input feature map, y is the output of convolution operation,p_0 is the starting position of each kernel and p is enumerating along with all the positions in K.

$$y(p_0) = \sum_{pn \in K} w(p_n).x(p_0+p_n) \tag{2}$$

Instead of using a simple fixed sampling grid, the deformable convolution introduces 2D offsets to the normal convolution operation depicted above. The Deformable Convolution operation is depicted by the equation below where Δp_n denotes the offsets added to the normal convolution operation.

$$y(p_0) = \sum_{pn \in K} w(p_n).x(p_0+p_n+\Delta p_n) \tag{2}$$



(a) standard convolution (b) deformable convolution

Fig. 4 Receptive field: CNN vs Deformable CNN[1]

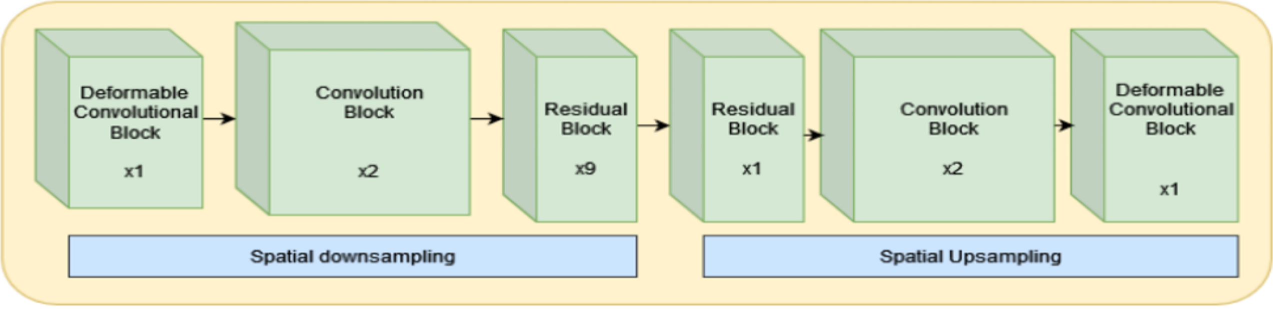


Fig. 5. Model Architecture

Fig 4. shows fixed receptive field in standard convolution and adaptive receptive field in deformable convolution operation.[1]

Now, the sampling is irregular and offset locations is found using the term $p_n + \Delta p_n$. As the offset Δp_n is typically fractional, Eq. (2) is implemented via bilinear interpolation[12] as

$$x(p) = \sum_{p} G(q,p).x(q) \qquad (3)$$

Note that G is two dimensional. It is separated into two one dimensional kernels as

$$G(q,p) = g(q_x,p_x).g(q_y,p_y) \qquad (4)$$

B. *Large Kernel Attention[12]*

A regular kernel layer is a technique that replicates the concept of cognitive attention. It looks to enhance a few parts of the input data while diminishing other parts such that the network focuses more on small but important details of the data. This objective can be achieved using an attention map which indicates the importance of different parts.

The traditional model which could understand the relation between data parts is the self-attention model which captures long range dependency between data parts. However, it has few drawbacks such as high cost computation for quadratic complexity, ignoring the 2D structure of the image and finally focusing only on the spatial adaptability while ignoring the channel dimension adaptability. Hence we chose another model called Large Kernel Attention which could resolve the earlier drawbacks and produce better attention maps.

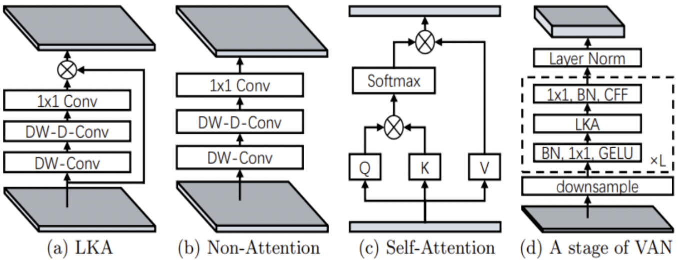


Fig. 6 (a) Structure of large kernel attention. (b) Non-attention module (c) Self-Attention Module (d) Visual Attention network. Difference between (a) and (b) is the element-wise multiplicity.[12]

The LKA model is composed of 3 blocks: a channel convolution (1×1 convolution), a spatial long-range convolution (depth-wise dilation convolution) and a spatial local convolution (depth_wise convolution)[5]. Note that these steps are applied sequentially on the data which results in determining the relation between the data parts at considerably low cost, ultimately assisting in building a better attention map. The main advantage of this model which is a combination of self attention with the convolutional layer is that it achieves adaptability in the channel dimension using dynamic process and neighbor correlations.

The LKA model from Fig.6 can be formulated as follows: -

$$\text{Attention} = Conv_{1x1}(\text{DW-D-Conv}(\text{DW-Conv}(F)))$$

$$\text{Output} = \text{Attention} * F$$

V. Evaluation Metrics

A. *Peak Signal-to-Noise Ratio*

The PSNR metric is used to measure the similarity between images by computing peak signal-to-noise ratio. Higher PSNR ratios imply superior performance of the model and thus greater quality of the compressed images.

$$PSNR = 10log_{10}(\frac{R^2}{MSE}) \qquad (5)$$

B. *Structural Similarity*

As the name suggests, the SSIM metric calculates the structural similarity between images. A value of +1 indicates that the two given images are very similar and a value of -1 indicates that the images are very dissimilar. Often, the ranges are specified between 0 and 1.

$$SSIM(x,y) = [l(x,y)]^{\alpha}.[c(x,y)]^{\beta}.[s(x,y)]^{\gamma} \qquad (6)$$

Here, l(x,y), c(x,y) and s(x,y) refers to luminance, contrast and structure comparisons respectively. α, β and γ describe the importance of each metric.

VI. Method

A. *ResNets[8][9]*

A Deep Residual Learning framework or Resnet is used to address a degeneration problem caused by stacking deep neural network layers together. When the network depth is increased the accuracy is saturated and then decreases rapidly. Instead of hoping each few stacked layers directly fit a desired underlying mapping, we explicitly let these layers fit a residual mapping. It is easier to optimize the residual mapping than to optimize the original, unreferenced mapping. We have used Resnet as our backbone model and we have added 101 layers to it. Resnets are seminal architecture in computer vision deep learning

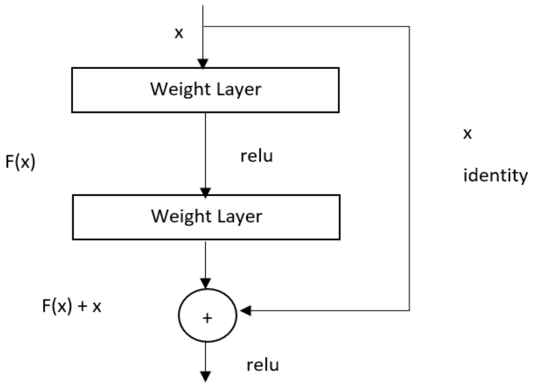


Fig. 7 ResNet

B. Deformable ConvNets

The deformable convolutional neural network was primarily targeted at tasks such as image segmentation, object detection. We apply deformable convolutional blocks in our model to better capture the blurry areas in an image. Offsets in deformable convolution are learned by the model, and are used to give importance to blurry and noisy areas.

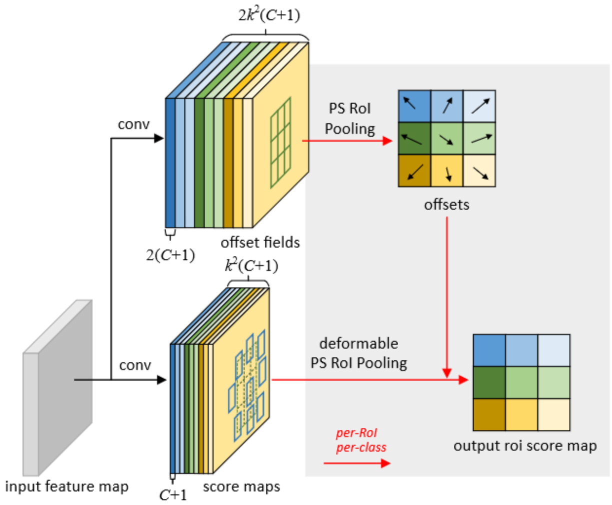


Fig. 8 Deformable convolutional block[1]

C. Kernel Attention

Large kernel attention brings pros of self-attention and large convolutional kernels. Large kernel attention can capture local receptive fields better. It can provide long-range dependencies for feature maps. LKA achieves the adaptability in the spatial dimension as well as in channel dimension.

We use a combination of LKA and Deformable convolution in search for better performance on GoPRO and SIDD datasets for deblurring and denoising tasks.

D. Model Architecture

We used PyTorch to model our framework, loss was back-propagated using Adam optimizer. We created multiple convolutional blocks using a deformable convolutional layer from PyTorch and residual blocks using LKA. We used mean squared loss to calculate PSNR of the predicted image and ground truth.

We modified existing Resnet residual block architecture as described in fig. 5. Residual block contains a 2D convolutional layer followed by batchnorm to center the layer weights around the mean. Input which is then forwarded to LKA block to better learn 2D structural embeddings. MLP layer allows the model to further abstract learned features to add with input residual.

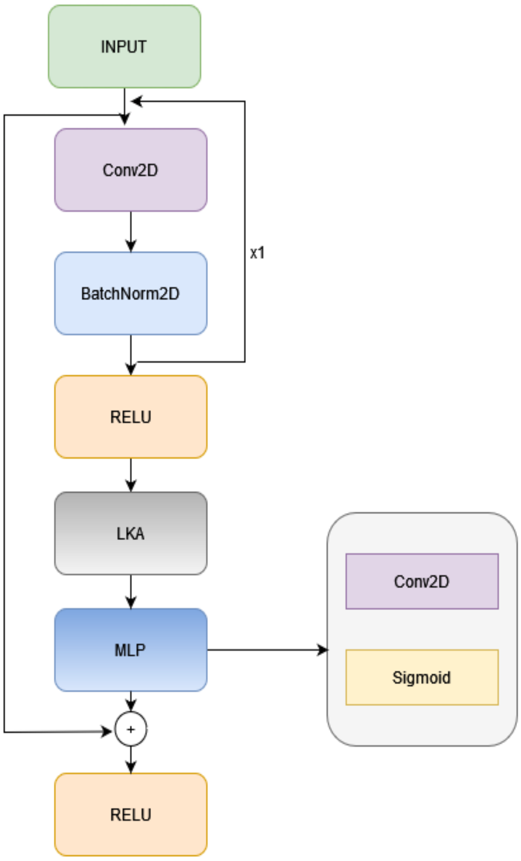


Fig. 9 Modified Resnet block with LKA

A Channel Attention Module performs channel-based attention in convolutional neural networks. We produce a channel attention map by exploiting the inter-channel relationship of features. As each channel of a feature map is considered as a feature detector, channel attention focuses on what is meaningful given an input image.

E. Mean Squared Error - MSE Loss Function

The MSE loss function is given by the formula: -

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \qquad (7)$$

The above function measures the average squared difference between the predicted values and the ground truth values.

VII. Experiments

We have performed various experiments relating to blurring and denoising images using the GoPro and SIDD datasets to evaluate the performance of our models. We have used PSNR and SSIM metrics for evaluation.

A. *Image Deblurring*

Image Deblurring is a task of generating clear / sharp images when a degenerated image is provided as input. The image may contain blur at various points. Our model would identify these defects in the images provided by the GoPro dataset and generate clear images.

B. *Image Denoising*

Image Denoising is a task of generating clear / sharp images when noisy images are given as input. The model uses the SIDD dataset for training and testing purposes and outputs images which are free from noise.

C. *Experimental Setup*

We executed our experiments on Google Colab Pro™ which has 24 GB of GPU RAM available for model training. Our metrics are extracted from the best execution run observed in hyperparameter search. GoPro dataset provides separate train and test sets while in SIDD dataset we used 80:20 split for train and test sets. Model employs data augmentation techniques like transformations, color augmentations and random crop to prevent model overfitting. We used PyTorch dataset and data-loaders class for batched data processing. Model checkpointing is done using the Tensorboard package in which we run our model for 50 epochs in a batch of 10 epochs each time.

Stacking residual deformable convolutional layers give us a large receptive field in the network. Using RELU as an activation function improves the convergence speed of the model.

Table 1.

	Number of parameters			
	Layer block name	***Input Shape***	***Output Shape***	***Per block parameters***
1	Deformable Convolutional block	32x256x256	32x256x256	18,432
2	Residual Block	32x256x256	128x64x64	73,728
3	LKA Residual Block	128x64x64	128x64x64	147,456

Model takes 32x256x256 spatial dimensions for input convolutional layers and the model has a total of 3,573,615 parameters with mentioned input dimensions and 9 residual blocks. Following table gives a summary of per block level estimation of parameters.

D. Training

This section elaborates on setup and hyperparameters employed during the training process. For deblurring we used GoPro dataset and for denoising we used SIDD dataset. Training is done one dataset at a time in a configurable way.

1. *Image deblurring*

The GoPro dataset contains 1111 test pairs of images for which we used a learning rate of 0.01, step size of 1000, batch size of 8, random crop size of 128 and deformable kernel size of 3. For deblurring task model trained for 50 epochs due to computing resource limitations.

2. *Image Denoising*

The SIDD dataset contains 160 pairs of groundtruth and noisy images for which we used learning rate of 0.01, step size of 500, batch size of 8, random crop of defined window size and deformable kernel size of 4. For denoising task, the model trained for 20 epochs due to computing resource limitations.

VIII. Results

In the image restoration task we used PSNR (peak signal to noise ratio) and SSIM (structural similarity index) to measure performance of our model. We executed our model with different combinations of residual blocks, epochs which resulted in varying improved results discussed below.

A. *Image Deblurring*

The following image shows output for our image deblurring task using settings discussed in the experimental setup section. Our observation suggests that increasing the number of epochs helps models in learning better performance on a dataset.

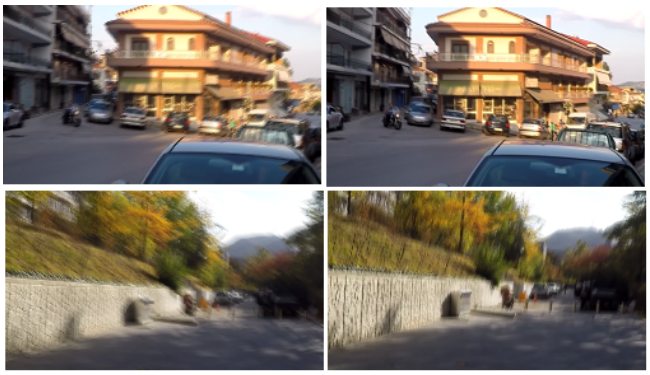


Fig. 10 Image Deblurring Output

From Table 2, we can see that training on more epochs will give improved performance for this model.

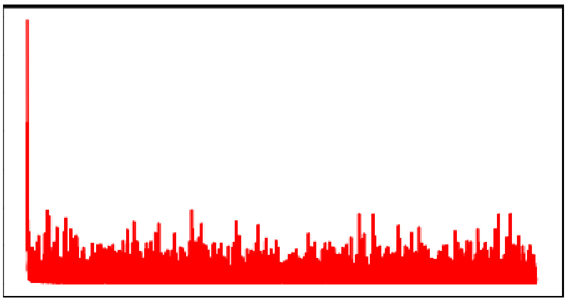


Fig. 11 Deblurring loss function graph

TABLE 2. PSNR AND SSIM FOR IMAGE DEBLURRING

	Image Deblurring on GoPro Dataset				
	Layer block name	***Residual Blocks***	***Epochs***	***PSNR***	***SSIM***
1	Deformable Convolution and LKA kernel size 3	12	50*	26.53	0.89
		9	20	25.40	0.81
2	Deformable convolution and LKA kernel size 4	12	50	26.60	0.90
		9	20	25.20	0.81

* max epochs we were able to run on Google Colab Pro™

B. Image Denoising

Image denoising task is compute intensive due to high resolution images available in images, we computed results only on settings mentioned in experimental setup section.

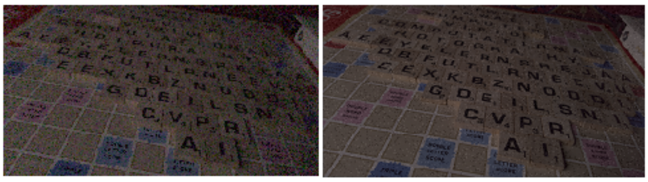


Fig. 12 Image Denoising output

TABLE 3. PSNR AND SIDD FOR IMAGE DENOISING

	Image Denoising on SIDD Dataset				
	Layer block name	***Residual Blocks***	***Epochs***	***PSNR***	***SSIM***
1	Deformable Convolution and LKA kernel size 3	12	50*	26.63	0.88
		9	20	25.73	0.83

* max epochs we were able to run on Google Colab Pro™

IX. CONCLUSION / FUTURE WORK

Through this project, we have demonstrated how deformable convolution neural networks and self-attention layers can be used in conjunction with the ResNet model to generate sharp images from noisy / blurred images. This provides us with increased modeling and training power. We have thus proposed an original approach to deblur and denoise images by using the aforementioned techniques and presented results that are comparable to existing approaches.

In future, more work can be done on the backbone model. We can replace the current ResNet model with different variants like ResNet-152, ResNet-164 or ResNeXt. We can also implement the backbone with some newly introduced models like CBNetV2, VGG-19 and NASNet-A.

REFERENCES

[1] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, Yichen Wei, *Deformable Convolutional Networks, 2017.*

[2] Xizhou Zhu, Han Hu, Stephen Lin, Jifeng Dai, *Deformable ConvNets v2: More Deformable, Better Results, 2019.*

[3] Keiron O'Shea, Ryan Nash, *An Introduction to Convolutional Neural Networks,* 2015.

[4] Min Lin, Qiang Chen, Shuicheng Yan, *Network In Network, 2013.*

[5] François Chollet, Xception: *Deep Learning with Depthwise Separable Convolutions, 2017.*

[6] Sachin Mehta, Mohammad Rastegari, Linda Shapiro, Hannaneh Hajishirzi, ESPNetv2: *A Light-weight, Power Efficient, and General Purpose Convolutional Neural Network, 2019.*

[7] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, Eugenio Culurciello, ENet: *A Deep Neural Network Architecture for Real-Time Semantic Segmentation, 2016.*

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, *Deep Residual Learning for Image Recognition, 2015.*

[9] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, Yun Fu, *Residual Dense Network for Image Restoration, 2018.*

[10] Han Zhang, Ian Goodfellow, Dimitris Metaxas, Augustus Odena, *Self-Attention Generative Adversarial Networks, 2018.*

[11] Jin-Hwa Kim, Jaehyun Jun, Byoung-Tak Zhang, *Bilinear Attention Networks, 2018.*

[12] Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng, Shi-Min Hu, *Visual Attention Network, 2022.*

[13] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, Ling Shao, *Multi-Stage Progressive Image Restoration, 2021.*

[14] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, Timo Aila, *Noise2Noise: Learning Image Restoration without Clean Data, 2018.*

[15] Hang Zhao, Orazio Gallo, Iuri Frosio, Jan Kautz, *Loss Functions for Neural Networks for Image Processing, 2015.*

[16] Kai Zhang, WangMeng Zuo, Shuhang Gu, Lei Zhang, *Learning Deep CNN Denoiser Prior for Image Restoration, 2017.*

[17] Tobias Plötz, Stefan Roth, *Neural Nearest Neighbors Networks, 2018.*

[18] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, *Restormer: Efficient Transformer for High-Resolution Image Restoration, 2022*

[19] Xiao-Jiao Mao, Chunhua Shen, Yu-Bin Yang, *Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections*, *2016*

[20] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, Lei Zhang, *Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising, 2016*

[21] Robert G. Keys, *Cubic Convolution Interpolation for Digital Image Processing, 1981*

[22] GoPro Dataset - https://paperswithcode.com/dataset/gopro

[23] SIDD - https://www.eecs.yorku.ca/~kamel/sidd/

[24] Karen Simonyan, Andrew Zisserman - *Very Deep Convolutional Networks for Large-Scale Image Recognition, 2015*