
# Title: Exploring Image Restoration Techniques: The Application of ViTDeblur

**By: Azamat Ilyassov**

***Course: BSc (Hons) Data Science and Analytics***

**Project code: *PJR40***

**Supervisor: *Alaa Mohasseb***

**May 2024**

**Word count: approx 9500**

Please tick

| | |
|---|---|
| ❑ | I give permission for my project to be published in the University library and/or be made available to other students as examples of previous work. (optional). |
| ❑ | I confirm that I have read and understood the University Rules in respect of plagiarism and student misconduct. |
| ❑ | I declare that this work is entirely my own. Each quotation or contribution cited from other work is fully referenced. |
| | |

Date: _____03/05/2024_____

## Acknowledgements

First of all, I would like to thank my supervisor, Alaa Mohasseb, for all her support. She always had a trust in me and always encouraged and motivated me to keep pushing forward when I felt like everything was bad.

Second of all, I would like to thank my friends, with whom I have been discussing my project ideas with, inspiring and intellectually challenging me, which helped me to get through hardships I faced during my project.

In the end, I want to thank my family for being supportive and understanding.

## Abstract

In this paper a pre-trained ViT is used for image deblurring problems, combining strengths of self-attention learning and Convolutional Neural Networks (CNN). The architecture combined ViT encoding and CNN-based decoding tasks for image restoration. Results obtained highlighted pros and cons of using ViT in combination with CNN. This paper also shortly reviewed main studies regarding image deblurring and the use of Transformer architecture in computer vision tasks, comparing it with classic CNN methodologies.

# Table of contents

**Acknowledgements** 3
**Abstract** 4
**Table of contents** 5
**List of figures** 8
**List of tables** 9
**Chapter 1: Introduction** 10
1.1 Research aims and objectives 11
1.2 Project limitations and risk assessment 11
**Chapter 2: Literature review** 12
2.1 Convolutional Neural Networks 12
2.2 Single stage approaches 13
2.2.1 Image Deblurring using GAN 13
2.2.1.1 Generator 13
Figure 1. Generator architecture drawn for (Li, 2023) 14
2.2.1.2 Discriminator 14
Figure 2. Discriminator architecture (Li, 2023) 15
2.2.1.3 Conclusion 15
2.2.2 DeblurGAN 15
2.2.2.1 Architecture 15
Figure 3. DeblurGAN generator architecture 16
2.2.2.2 Loss 16
2.2.2.3 Training and results 17
Table 1. DeblurGAN benchmarks results (Kupyn et al., 2018) 18
2.2.2.4 Benchmark object detection using YOLO 18
Table 2. DeblurGAN YOLO benchmarks results (Kupyn et al., 2018) 18
2.2.2.5 Conclusion 18
2.2.3 DeblurGAN-v2 18
2.2.3.1 Architecture 19
Figure 4. DeblurGAN-v2 architecture Orest Kupyn et al. (2019) 19
2.2.3.2 Feature Pyramid Deblurring 19
2.2.3.3 Backbone 19
2.2.3.4 Discriminator Loss 19
2.2.3.5 Global and Local scalers 20
2.2.3.6 Overall training loss 20
2.2.3.7 Results 20
Table 3. DeblurGAN-v2 performance evaluation (Orest Kupyn et al., 2019) 21
2.2.3.8 Conclusion 21
2.3 Multi stage models 21
2.3.1 Scale-recurrent Network 21
2.3.1.1 Scale-recurrent Network (SRN) 21
Figure 5. Scale Recurrent Network architecture (Tao et al., 2018) 22

2.3.1.2 Encoder/Decoder ResBlocks....22
2.3.1.3 Loss....23
2.3.1.4 Experiments....23
Table 4. SRN evaluation results (Tao et al., 2018)....23
2.3.1.5 Conclusion....23
2.3.2 MRPNet....24
Figure 6. MRPNet (Zamir et al., 2021)....24
2.3.2.1 Encoder-decoder....24
2.3.2.2 Original Resolution Network....25
2.3.2.3 Cross-stage Feature Fusion....25
2.3.2.4 Supervised Attention Model....25
Figure 7. a-encoder decoder subnetwork, b - ORB structure that is the main component of ORSNet blocks, GAP - global average pooling, c - CSFF between first two stages, d - CSFF between last two stages (Zamir et al., 2021)....26
Figure 8. Supervised Attention Model design (Zamir et al., 2021)....26
2.3.2.5 Loss....26
2.3.2.6 Experiments and results....26
Table 5. MRPNet evaluation results (Zamir et al., 2021)....27
2.3.2.7 Conclusion....27
2.4 Multi stage vs single stage approaches comparison....27
2.5 Transformer based models....28
2.5.1 Transformer - basics....28
2.5.1.1 Architecture....28
Figure 9. Transformer model architecture proposed by Vaswani et al. (2017)... 29
2.5.1.2 Attention head and multi-headed solution....29
Figure 10. Scaled-dot product attention of (Vaswani et al., 2017)....30
2.5.2 Vision Transformer....30
Figure 11. Vision Transformer (ViT) model architecture (Dosovitskiy et al., 2020)....30
2.5.3 Restormer model....31
Figure 12. Restormer model architecture (Zamir et al., 2022)....31
2.5.3.1 Multi-Dconv Head Transposed Attention....31
2.5.3.2 Gated-Dconv Feed-Forward Network....32
2.5.3.3 Training and results....32
Table 6. Restormer evaluation results (Zamir et al., 2022)....33
2.5.3.4 Conclusion....33
2.5.4 Uformer....33
2.5.4.1 Locally-enhanced Window Transformer....33
Figure 14. Locally-enhanced Feed-Forward Network (Wang et al., 2021)....34
2.5.4.2 Multi-Scale restoration modulator....34
2.5.4.3 Uformer’s performance benchmark....35
Table 7. Uformer evaluation results (Wang et al., 2021)....35
2.5.4.4 Conclusion....35

2.5.5 SAM-Deblur..... 35
2.5.5.1 Segment Anything Model.....35
Figure 15. Foundation model pipeline for image segmentation (Kirillov et al., 2023).....36
2.5.5.2 NAFNet..... 36
Figure 16. a - Channels attention implemented by Hu et al. (2019), b - Simplified Channel Attention (Chen et al., 2022), c - Simple gate (Chen et al., 2022).....37
Figure 17. NAFNet architecture, proposed by Chen et al. (2022).....37
2.5.5.3 Mask Average Pooling Unit.....37
2.5.5.4 Results.....37
Table 8. SAM-Deblur evaluation results (Li et al., 2023).....37
2.5.5.5 Conclusion..... 37
2.6 Blurred Image to Video Restoration.....37
Figure 18. Model architecture (Purohit et al., 2019).....39
2.7 Summary..... 39
**Chapter 3: Dataset.....40**
**Chapter 4: Methodology..... 40**
4.1 Architecture.....40
Figure 19. ViTDeblur architecture, where ⊕ is pointwise addition of tensors.. 41
4.2 ViTDecoder.....41
Figure 20. ViTDecoder design, where ⊕ is pointwise addition of tensors.....41
4.2.1 Upsampling blocks.....41
Figure 21. a) upsampling block, b) Decoder Block..... 41
4.2.2 Decoding blocks..... 41
4.3 Loss..... 41
**Chapter 5: Training performance and Experiments..... 42**
5.1 First approach.....42
5.1.1 Training.....42
Figure 22. Training and validation losses logs during ViTDebluradam training.. 43
Figure 23. Losses after each 10 steps.....43
5.1.2 Evaluation..... 44
5.2 Second approach.....44
5.2.1 Training.....44
Figure 24. Training and validation losses logs during ViTDeblurSGD training... 44
Figure 25. Losses after each 10 steps.....45
5.2.2 Evaluation..... 45
5.3 Visual Testing.....45
Figure 25. Visual comparisons of image deblurring using the ViTDebluradam model. Image on the left is a blurred input, the middle image is a sharp image recovered from the trained model and the image on the right is the real sharp image..... 47
Figure 26. Visual comparisons of image deblurring using the ViTDeblurSGD model, same arrangement as in Figure 25..... 48

5.4 Discussion..... 49
**Chapter 6: Conclusion..... 49**
**References..... 50**
**Appendix..... 54**
Appendix A: Project Initiation Document..... 54
Appendix B: Gantt chart.....62
Appendix C:..... 63

# List of figures

Figure 1. Generator architecture drawn for (Li, 2023).....14
Figure 2. Discriminator architecture (Li, 2023).....15
Figure 3. DeblurGAN generator architecture..... 16
Figure 4. DeblurGAN-v2 architecture Orest Kupyn et al. (2019)..... 19
Figure 5. Scale Recurrent Network architecture (Tao et al., 2018).....22
Figure 6. MRPNet (Zamir et al., 2021).....24
Figure 7. a-encoder decoder subnetwork, b - ORB structure that is the main component of ORSNet blocks, GAP - global average pooling, c - CSFF between first two stages, d - CSFF between last two stages (Zamir et al., 2021).....26
Figure 8. Supervised Attention Model design (Zamir et al., 2021).....26
Figure 9. Transformer model architecture proposed by Vaswani et al. (2017)... 29
Figure 10. Scaled-dot product attention of (Vaswani et al., 2017)..... 30
Figure 11. Vision Transformer (ViT) model architecture (Dosovitskiy et al., 2020).....30
Figure 12. Restormer model architecture (Zamir et al., 2022).....31
Figure 14. Locally-enhanced Feed-Forward Network (Wang et al., 2021)..... 34
Figure 15. Foundation model pipeline for image segmentation (Kirillov et al., 2023).....36
Figure 16. a - Channels attention implemented by Hu et al. (2019), b - Simplified Channel Attention (Chen et al., 2022), c - Simple gate (Chen et al., 2022).....37
Figure 17. NAFNet architecture, proposed by Chen et al. (2022).....37
Figure 18. Model architecture (Purohit et al., 2019).....39
Figure 19. ViTDeblur architecture, where ⊕ is pointwise addition of tensors.. 41
Figure 20. ViTDecoder design, where ⊕ is pointwise addition of tensors.....41
Figure 21. a) upsampling block, b) Decoder Block..... 41
Figure 22. Training and validation losses logs during ViTDebluradam training.. 43
Figure 23. Losses after each 10 steps.....43
Figure 24. Training and validation losses logs during ViTDeblurSGD training... 44
Figure 25. Losses after each 10 steps.....45

Figure 25. Visual comparisons of image deblurring using the ViTDebluradam model. Image on the left is a blurred input, the middle image is a sharp image recovered from the trained model and the image on the right is the real sharp image..... 47
Figure 26. Visual comparisons of image deblurring using the ViTDeblurSGD model, same arrangement as in Figure 25..... 48

# List of tables

Table 1. DeblurGAN benchmarks results (Kupyn et al., 2018).....18
Table 2. DeblurGAN YOLO benchmarks results (Kupyn et al., 2018).....18
Table 3. DeblurGAN-v2 performance evaluation (Orest Kupyn et al., 2019) 21
Table 4. SRN evaluation results (Tao et al., 2018)..... 23
Table 5. MRPNet evaluation results (Zamir et al., 2021).....27
Table 6. Restormer evaluation results (Zamir et al., 2022)..... 33
Table 7. Uformer evaluation results (Wang et al., 2021)..... 35
Table 8. SAM-Deblur evaluation results (Li et al., 2023)..... 37

# Chapter 1: Introduction

Since the development of the first computer vision techniques in the latter half of the 20th century, researchers have been devising a variety of methods for image processing tasks. Key challenges in the field include image recognition, semantic segmentation, object localisation and detection. Models developed to solve these tasks are generally trained on static images free of artefacts, such as blurred edges or noise. However, real world images often contain anomalies, such as haze, noise, motion blur, defocus blur, raining, which degrade the performance of image segmentation and classification. Object detection algorithms and problems that require the target to be properly visible are even more vulnerable to such artefacts. Hence,image restoration algorithms have been developed in order to improve performance of image processing applications under these conditions. There is a lot of variance with the extent of  blurred and corrupted images. Addressing these issues, image restoration approaches evolved from Wiener filters  (Wiener, 1949) to modern Deep Learning techniques using various architectures such as Convolutional Neural Networks (CNN) (Lecun et al., 1998) or Transformers (Vaswani et al., 2017). Each of these methods mentioned have their own advantages and disadvantages which will be discussed later in this paper.

In this paper we explore the use of a Vision Transformer (ViT) architecture (Dosovitskiy et al., 2020) to deblur images. More precisely we use a pre-trained ViT to encode images to restore a sharp image from its blurred counterpart. In addition to this we evaluate two types of losses. The reasoning behind choosing a ViT as a base model is to use the combination of ViT encoders with CNN based decoder on blurred images and observe the ability of convolutional layers to capture local features and ViT's strength of capturing long-term global context. The model is named ViTDeblur.
The model presented in this paper is trained on a very small Kaggle dataset created by Aleksey Alekseev (2019). Other existing research in the literature uses much bigger datasets that are traditionally used for training and benchmarks evaluation of their models' performances. These datasets as well as evaluation metrics used for benchmarking and testing are to be shortly introduced to add up for a context before reviewing related works.

**Datasets:**

- GoPro dataset - introduced by Nah et al. (2016) along with a dynamic deblurring multi stage model. As the name implies - dataset consists of images generated from GoPro 240 frames per second (fps) videos.
- RealBlur-J and RealBlur-R - these two datasets were introduced by Rim et al. (2020). Images in this dataset are aligned and were gathered using a dual camera system.
- Human Aware Image Deblurring (HIDE) - introduced by Shen et al. (2020) and the authors pointed out the weak point of GoPro dataset - capturing long range views and ignoring close-up shots, while HIDE dataset covers both wide-range and close-up blurred shots that are also aware of human motion, hence the name.

**Metrics:**

- Peak signal-to-noise ratio (PSNR). “The PSNR block computes the peak signal-to-noise ratio, in decibels, between two images. This ratio is used as a quality measurement between the original and a compressed image. The higher the PSNR, the better the quality of the compressed, or reconstructed image. The mean-square error (MSE) and the peak signal-to-noise ratio (PSNR) are used to compare image compression quality. The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. The lower the value of MSE, the lower the error.” (VisibleBreadcrumbs, 2020) The following metric can be formed the following way:

$$MSE = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M * N}$$

$$PSNR = 10 \log_{10}\left(\frac{R^2}{MSE}\right)$$

Where $I_1$ and $I_2$ are the original and predicted images respectively, M and N are image sizes, R is the maximum values of the data range.

- Structural Similarity Index (SSIM). “As a specific example, we develop a measure of structural similarity (SSIM) that compares local patterns of pixel intensities that have been normalised for luminance and contrast” (Wang et al., 2004)

## 1.1 Research aims and objectives

This paper mostly focuses on observing how well a pre-trained ViT would perform in an image restoration task and find out its weaknesses and strengths. This paper aims to answer the following questions and achieve the following goals:

- What are the current baseline principles used in modern computer vision problems?
- What are the current methods and approaches to solving image restoration tasks?
- What are the limitations of modern approaches? What are their strengths?
- How well would a pre-trained ViT perform in image restoration? Assess feasibility of such an approach.

## 1.2 Project limitations and risk assessment

Using ViT to solve image deblurring problems has great potential, because of how well the model performed and promising results of such approach in image recognition. It is also worth mentioning how well the Transformer model (Vaswani et al., 2017) showed itself in natural language processing. Possible limitations are computational complexity and time limits required in order to train models, because deep learning models are very expensive to train, which depends on the size of the dataset, complexity of the model and training parameters and this may require advanced hardware and good graphical accelerators. Due to self-attention blocks capturing long range pixel dependencies, computation cost increases quadratically with the increment of images resolutions.

# Chapter 2: Literature review

As was stated before, image restoration is a long standing area of computer vision problems and deblurring images is one of them. The pioneer of image restoration and denoising turned out to be an American computer scientist and mathematician - Norbert Wiener.
Norbert Wiener set up a foundation for Image restoration and made significant changes that affected the area of image processing in 1949, when (Wiener, 1949) was published. Here the Wiener filter was presented as a way to deal with blurred and noisy images by minimising overall mean squared error (*WIENER FILTERING*, n.d.). The main issue of the Wiener filter is the sensitivity to small changes in signals that were used to design filters (Vastola & Poor, 1983).
With the development of computer science approaches, rise of machine learning methodologies and development of first neural networks as well as the integration of the term artificial intelligence approaches to such a problem and its classification have changed.

## 2.1 Convolutional Neural Networks

Computer vision is one of the oldest fields of study in computer science, artificial intelligence and machine learning areas. Image processing was first managed by such machine learning techniques as support-vector machines for classification problems.

The development of deep learning techniques led to Convolutional Neural Networks (CNN), first introduced by LeCun et al. (1998) alongside the first CNN-based image classification architecture LeNet-5 for handwritten digits recognition. It was noted that fully-connected networks could be trained in order to solve image recognition tasks, though this may result in multiple units with various weight patterns located at different positions in images. This would require an enormous amount of training samples. Another reason CNNs were presented is because traditional fully-connected networks failed to extract the understanding of local features and spatial information of pixels that are highly locally correlated.

Their seminal paper introduced CNNs consisting of three pillars: local receptive fields, shared weights and sub-sampling. Local receptive fields, also known as kernels, have a predefined

size. These kernels share the same set of weights for a particular feature in order to detect this feature in other locations of the input, hence the shared weight principle was implemented to hold the CNN structure. Sub-sampling layers helps with local averaging and reducing the resolution of the feature map, because once the feature has been detected it is potentially dangerous to have the information about its position, only approximate position, because positions for the same feature are most likely to vary (Lecun et al., 1998) . Overall, the input image is processed using receptive fields and sub-sampled, the output states are to be stored at corresponding locations, akin to a mathematical convolution, hence the namesake - a convolution neural network.

The idea behind a combination of convolution and sub-sampling was first explored by Fukushima (1980) who introduced a structure similar to modern convolutional layers - neocognitron. Such an algorithm was able to learn without a “teacher”, through learning by being presented with a number of stimulus patterns, which is similar to feature extraction and mapping to convolutional layers.

Nowadays, most computer vision problems are dealt with by various neural networks architectures that either involve CNNs or are fully based on them. The following section contains a summarisation of works that used CNNs solely or with the use of other techniques in order to deal with the problem of image deblurring.

## 2.2 Single stage approaches

Single stage approaches do not use any subnetworks and are usually used for high level computer vision tasks (Zamir et al., 2021).

### 2.2.1 Image Deblurring using GAN

One of the simplest examples of single stage approaches was by Li (2023). In this work a Generative Adversarial Network (GAN) (Goodfellow et al., 2014) was used in order to make an image deblurring model.

The GAN model uses two models - generative model and discriminative model. Generative model generates noisy input that corresponds to the distribution that is similar to the distribution in the training set. Discriminative model’s role is to classify whether the sample has come from training or a generated set. So in the end it is a game of maximising discriminants' probability of error and minimising generative model’s errors.

#### 2.2.1.1 Generator

Generator network here was designed using ResNet blocks (He et al., 2015), that consist of convolution, batch normalisation (Ioffe & Szegedy, 2015) and rectified linear unit (ReLU) (Fukushima, 1969) activation layers totalling at 9 residual blocks for a generator network.

| Generator | K | S | $C_{in}$ | $C_{out}$ |
|---|---|---|---|---|
| input_1 | 7 | 1 | 3 | 3 |
| conv2d | 7 | 1 | 3 | 64 |
| conv2d_1 | 3 | 2 | 64 | 128 |
| conv2d_2 | 3 | 2 | 128 | 256 |
| conv2d_3 | 3 | 1 | 256 | 256 |
| conv2d_4 | 3 | 1 | 256 | 256 |
| conv2d_5 | 3 | 1 | 256 | 256 |
| conv2d_6 | 3 | 1 | 256 | 256 |
| conv2d_7 | 3 | 1 | 256 | 256 |
| conv2d_8 | 3 | 1 | 256 | 256 |
| conv2d_9 | 3 | 1 | 256 | 256 |
| conv2d_10 | 3 | 1 | 256 | 256 |
| conv2d_11 | 3 | 1 | 256 | 256 |
| conv2d_12 | 3 | 1 | 256 | 256 |
| conv2d_13 | 3 | 1 | 256 | 256 |
| conv2d_14 | 3 | 1 | 256 | 256 |
| conv2d_15 | 3 | 1 | 256 | 256 |
| conv2d_16 | 3 | 1 | 256 | 256 |
| conv2d_17 | 3 | 1 | 256 | 256 |
| conv2d_18 | 3 | 1 | 256 | 256 |
| conv2d_19 | 3 | 1 | 256 | 256 |
| conv2d_20 | 3 | 1 | 256 | 256 |
| conv2d_21 | 3 | 1 | 256 | 128 |
| conv2d_22 | 3 | 1 | 128 | 64 |
| conv2d_23 | 7 | 1 | 64 | 3 |

Figure 1. Generator architecture drawn for (Li, 2023)

#### 2.2.1.2 Discriminator

The main goal of the discriminator is to distinguish between real or fake image (Goodfellow et al., 2014), Li (2023) proposed an architecture to address this, using convolutional layers, leaky ReLU (Maas et al., 2013), sigmoid and tanh activation layers. The architecture is depicted in Figure 2.

| Discriminator | K | S | $C_{in}$ | $C_{out}$ |
|---|---|---|---|---|
| input_1 | 4 | 2 | 3 | 3 |
| conv2d_24 | 4 | 2 | 3 | 64 |
| conv2d_25 | 4 | 2 | 64 | 64 |
| conv2d_26 | 4 | 2 | 64 | 128 |
| conv2d_27 | 4 | 2 | 128 | 256 |
| conv2d_28 | 4 | 1 | 256 | 512 |
| conv2d_29 | 4 | 1 | 512 | 1 |

Figure 2. Discriminator architecture (Li, 2023)

#### 2.2.1.3 Conclusion

This work applied the GAN model as a solution to the problem of corrupted images restoration. Generator yields a fake sharp image from the blurred counterpart and the discriminator is provided with ground truth and generated value and differentiates which image is sharp and which is not.

### 2.2.2 DeblurGAN

Kupyn et al. (2018) proposed a more defined and optimised solution in terms of architecture, loss and training approach. The model itself is very simple. In short, sharp image $\mathbf{I_s}$ is recovered from blurry image $\mathbf{I_b}$ using a CNN-based model that is supposed to be a Generative model. During the training phase the discriminator model, or critic model is presented and used in order to train both networks in GAN manner.
Additionally, a new approach was also proposed to find loss between predicted and target images along with a new way to evaluate performance of deblurring models using object detection problems, in this case, using the YOLO model on images with various blurring kernels.

#### 2.2.2.1 Architecture

Generator network is very similar to the network used for style transfer that was built by Johnson et al. (2016). Generator is built using two strided convolutions with stride ½, nine Residual Blocks of (He et al., 2015) and two transposed convolutions. Each Residual Blocks contains one convolution, normalisation and ReLU activation layers. ResOut connection was presented here - global skip-connection that adds initial input to the output.
Discriminator network is based on Wasserstein GAN with gradient penalty (WGAN-GP). Even though it is not used in an actual model architecture - it is used during the training stage.

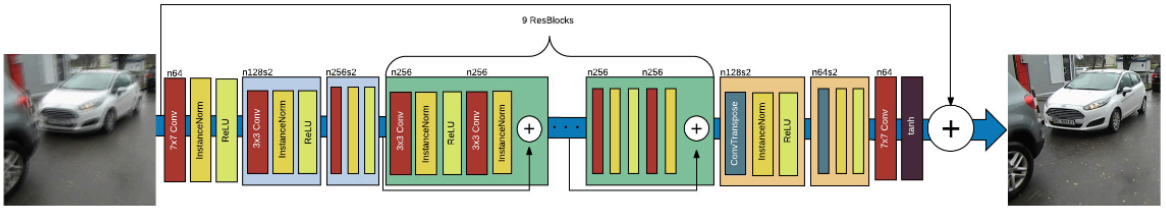


Figure 3. DeblurGAN generator architecture.

#### 2.2.2.2 Loss

The proposed approach to finding loss values combines both adversarial loss and content loss principles.

Instead of using basic and traditional approach of using classical adversarial loss initially proposed in the original GAN paper (Goodfellow et al., 2014), i.e.

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Zhu et al., (2017) proved to have other ways to be more efficient, when cycle consistency loss was suggested as an additional to the whole adversarial loss solution:

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G,F) = {} & \mathbb{E}_{x\sim p_{\text{data}}(x)}\big[\|F(G(x)) - x\|_1\big] \\ & + \mathbb{E}_{y\sim p_{\text{data}}(y)}\big[\|G(F(y)) - y\|_1\big].\end{aligned}$$

Hence, WGAN-GP (Gulrajani et al., 2017) was used during training. WGAN (Arjovsky et al., 2017) algorithm solves the following problem:

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)]$$

The problem with WGAN is that without careful tuning of threshold $c$ weights and gradients might explode or vanish completely. Gulrajani et al. (2017) applied the following coefficient penalty to WGAN:

$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[D(\tilde{\boldsymbol{x}})] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[D(\boldsymbol{x})]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{\boldsymbol{x}}}}\left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2\right]}_{\text{Our gradient penalty}}.$$

Thus, calculated loss for this approach is:

$$\mathcal{L}_{GAN} = \sum_{n=1}^{N} -D_{\theta_D}(G_{\theta_G}(I^B))$$

As for content loss, Kupyn et al. (2018) used principles used for constructing perceptual loss (Johnson et al., 2016), that was initially used for style transfer. Perceptual loss of (Johnson et al., 2016) is a pre-trained image classification network, which is a 16-layered Visual Geometry Group (VGG) network (Simonyan & Zisserman, 2015). One of perceptual losses (Johnson et al., 2016) there is one that calculates **feature loss**:

$$\ell_{feat}^{\phi,j}(\hat{y},y) = \frac{1}{C_j H_j W_j} \|\phi_j(\hat{y}) - \phi_j(y)\|_2^2$$

And another one, in order to boost loss between target image and predicted image (Johnson et al., 2016) also proposed to penalise styles differences, such as colours, textures etc. using the following Gram matrix:

$$G_j^{\phi}(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'}.$$

In order to attain resulting **style-reconstruction loss** - using predicted and target images that are processed through Gram matrix - squared Frobenius norm was applied:

$$\ell_{style}^{\phi,j}(\hat{y},y) = \|G_j^{\phi}(\hat{y}) - G_j^{\phi}(y)\|_F^2.$$

Thus, using these formulas the following content-loss was obtained:

$$\mathcal{L}_X = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^S)_{x,y} - \phi_{i,j}(G_{\theta_G}(I^B))_{x,y})^2$$

(Kupyn et al., 2018) states that the reason for such a complicated procedure is that using Mean Squared Error (MSE) does not converge pixels in a way that would be useful and meaningful.

#### 2.2.2.3 Training and results

(Kupyn et al., 2018) trained their three model variations using GoPro and COCO datasets and evaluated them - $\text{DeblurGAN}_{\text{Wild}}$ - trained on GoPro images, $\text{DeblurGAN}_{\text{Synth}}$ trained on COCO synthetically generated images, $\text{DeblurGAN}_{\text{Comb}}$ combined both.
(Kupyn et al., 2018) achieved the following results:

| DeblurGAN | | | |
|---|---|---|---|
| Metric | Wild | Synth | Comb |
| PSNR | 27.2 | 23.6 | 28.7 |
| SSIM | 0.954 | 0.884 | 0.958 |
| Time | 0.85s | | |

*Table 1. DeblurGAN benchmarks results (Kupyn et al., 2018)*

#### 2.2.2.4 Benchmark object detection using YOLO

(Journal Of L A T E X Class & Files, (2015) proposed a dehazing model and used object detection using You Only Look Once (YOLO) framework (Redmon et al., 2016) as a benchmark in order to evaluate the dehazing model's performance.
Sajjadi et al. (2017) proposed image quality enhancement model that used the same benchmarking principle.

The same YOLO benchmark was used in Kupyn et al. (2018).
YOLO is one the State-of-the-Art (SOTA) model that solves object detection issues in computer vision. Even though this model is extremely efficient, it is trained on limited datasets and when the image is blurred - performance drops and artefacts are to be found. The following are results comparing YOLO model's performance on blurred and deblurred images:

| Method | precision | recall | f1-score |
|---|---|---|---|
| Blurred image | 0.821 | 0.437 | 0.570 |
| DeblurGAN Wild | 0.764 | 0.631 | 0.691 |
| DeblurGAN Synth | 0.801 | 0.517 | 0.628 |
| DeblurGAN Comb | 0.671 | 0.742 | 0.704 |

*Table 2. DeblurGAN YOLO benchmarks results (Kupyn et al., 2018)*

#### 2.2.2.5 Conclusion

Kupyn et al., (2018) used GAN structure and applied it to image restoration tasks, implementing residuals blocks in a generator network. Architecture is relatively simple and easy to understand, therefore in order for the model to be capable of generating a sharp image from blurred input advanced loss techniques were designed using a pre-trained model for training, which requires some experience and proper theoretical understanding of principles used. (Kupyn et al., 2018) also tested their approach on object detection tasks, quite successfully.

### 2.2.3 DeblurGAN-v2

Model defined and trained by Kupyn et al., (2018) turned out to be a very efficient and fast generating sharp and perceptually pleasing image from blurred inputs. Orest Kupyn et al. (2019) proposed an updated version of DeblurGAN in order to push the boundaries of GAN for image deblurring tasks.
In contrast to this approach  a new more efficient generator framework was proposed by Orest Kupyn et al. (2019). New framework uses the Feature Pyramid Networks (FPN) approach

(Lin et al., 2017) for object detection problems. In the generator network a new FPN based backbone was added.

#### 2.2.3.1 Architecture

The overall generator architecture is presented in Fig 4.

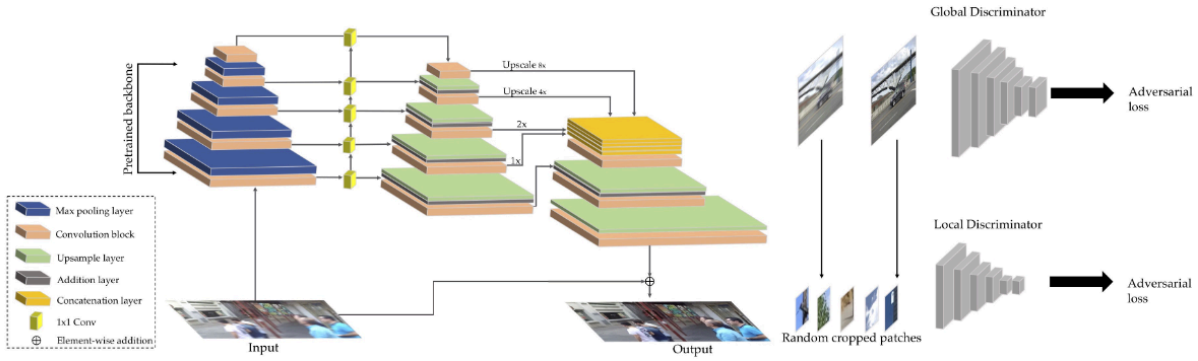



Figure 4. DeblurGAN-v2 architecture Orest Kupyn et al. (2019)

#### 2.2.3.2 Feature Pyramid Deblurring

As was mentioned earlier, the generative model uses a newly proposed FPN (Lin et al., 2017) for object detection. Here FPN uses both bottom-up and top-down approaches. Bottom-up task extracts features and gains semantic context information. Top-down approach reconstructs higher spatial resolution from the data that was extracted from bottom-up solutions.

The output is later stacked and passed on to two additional upsampling convolutional layers in order to preserve image resolution and add output to input - global skip connection, the same principle that was described by Kupyn et al. (2018) and Mao et al. (2017).

#### 2.2.3.3 Backbone

Orest Kupyn et al. (2019) trained two variations of DeblurGAN-v2 with two different backbones in FPN. One is built and trained using **Inception-ResNet-v2** (Szegedy et al., 2016) the other one uses **MobileNet v2** (Sandler et al., 2019).

As for MobileNet v2 all convolution layers were replaced with Depth-wise Separable Convolutions in order to decrease memory usage and computing cost and hence **MobileNet-DSC** is the second backbone variation.

#### 2.2.3.4 Discriminator Loss

Jolicoeur-Martineau (2018) suggested a relativistic discriminator that would estimate that real data is more realistic than the data generated synthetically. This would send a before-hand knowledge that half of the batch that has been passed is fake. Such an approach proved to be computationally efficient and more stable than other GAN discriminators.

Generally, GANs have the following discriminator and generator adversarial losses:

$$L_D^{GAN} = \mathbb{E}_{x_r \sim \mathbb{P}} [f_1(C(x_r))] + \mathbb{E}_{x_f \sim \mathbb{Q}} [f_2(C(x_f))]$$

$$L_G^{GAN} = \mathbb{E}_{x_r \sim \mathbb{P}} [g_1(C(x_r))] + \mathbb{E}_{x_f \sim \mathbb{Q}} [g_2(C(x_f))]$$

To make the loss relativistic, instead of $a(C(x_r))$, where $a$ is an activation function, $a(C(x_r)-C(x_f))$ is presented. This way a new class of discriminators looks like the following:

$$L_D^{RGAN} = \mathbb{E}_{(x_r, x_f) \sim (\mathbb{P}, \mathbb{Q})} [f_1(C(x_r) - C(x_f))] + \mathbb{E}_{(x_r, x_f) \sim (\mathbb{P}, \mathbb{Q})} [f_2(C(x_f) - C(x_r))]$$

$$L_G^{RGAN} = \mathbb{E}_{(x_r, x_f) \sim (\mathbb{P}, \mathbb{Q})} [g_1(C(x_r) - C(x_f))] + \mathbb{E}_{(x_r, x_f) \sim (\mathbb{P}, \mathbb{Q})} [g_2(C(x_f) - C(x_r))]$$

Orest Kupyn et al. (2019) used relativistic GAN method in order to “wraps” up LSGANs (Mao et al., 2017), thus obtaining the following *RaGAN-LS* loss:

$$\begin{aligned} L_D^{RaLSGAN} = & \mathbb{E}_{x \sim p_{data}(x)} \left[ (D(x) - \mathbb{E}_{z \sim p_z(z)} D(G(z)) - 1)^2 \right] \\ & + \mathbb{E}_{z \sim p_z(z)} \left[ (D(G(z)) - \mathbb{E}_{x \sim p_{data}(x)} D(x) + 1)^2 \right] \end{aligned}$$

#### 2.2.3.5 Global and Local scalers

Orest Kupyn et al. (2019) decided to use PatchGAN (Isola et al., 2018) as a global discriminator. And in order to boost the extraction of meaningful spatial information another scaling technique was applied that works the same way as the work Isola et al. (2018) proposed for local feature extraction and managing patches.

#### 2.2.3.6 Overall training loss

The following loss function was used in training:

$$L_G = 0.5 * L_p + 0.006 * L_X + 0.01 * L_{adv}$$

$L_{adv}$ returns both global and local losses. The same content loss $L_x$ that was proposed by Orest Kupyn et al. (2019) uses the same perceptual loss designed by Johnson et al. (2016). And here, in contrast to loss built and proposed by Kupyn et al. (2018), pixel-space or pixel-wise loss $L_p$ was proposed, using MSE, because as was confirmed by Ledig et al. (2016) they tend to produce over-smoothed results, which can cause issues with perceptual quality of outputs, which has been dealt with using perceptual loss mentioned earlier in the work done by Johnson et al. (2016).

#### 2.2.3.7 Results

The following is the table showing results obtained after testing DeblurGAN-v2 with different backbones on Restore dataset:

| | PSNR | SSIM |
|---|---|---|

| | | |
|---|---|---|
| Degraded | 22.056 | 0.873 |
| DeblurGAN | 26.435 | 0.892 |
| DeblurGAN-v2(Inception-ResNet-v2) | 26.916 | 0.894 |
| DeblurGAN-v2(MobileNet-DSC) | 25.412 | 0.891 |

*Table 3. DeblurGAN-v2 performance evaluation (Orest Kupyn et al., 2019)*

#### 2.2.3.8 Conclusion

Orest Kupyn et al. (2019) had a more complex approach that used a pre-built architecture as a backbone and stacked tensors are stacked and run through local and global discriminators. In addition to using FPN, a perceptual loss structure of Johnson et al. (2016) that uses a pre-trained VGG was added to the overall loss. Such approaches have promising performance results and are not as computationally costly as classic basic approaches.

## 2.3 Multi stage models

For such CNN-based models the approach is using lighter and smaller subnetworks that are used to build the whole architecture. Such models show their efficiency by breaking down the task into several subtasks. Mainly, such models have an approach of using identical subnetworks at each stage of processing inputs.

### 2.3.1 Scale-recurrent Network

Orest Kupyn et al. (2019) proposed a very efficient and accurate approach to deblurring images. In the evaluation part various models with different backbones were compared to the results and overall performance of model built and trained (Tao et al., 2018). One of the novel ideas presented in this paper is sharing weights and parameters across the network to increase the speed of training. Along with that the recurrent modules were presented, which stored beneficial data using hidden states and recurrently learning and adjusting weights multiple times.

#### 2.3.1.1 Scale-recurrent Network (SRN)

The following figure illustrates the SRN architecture:

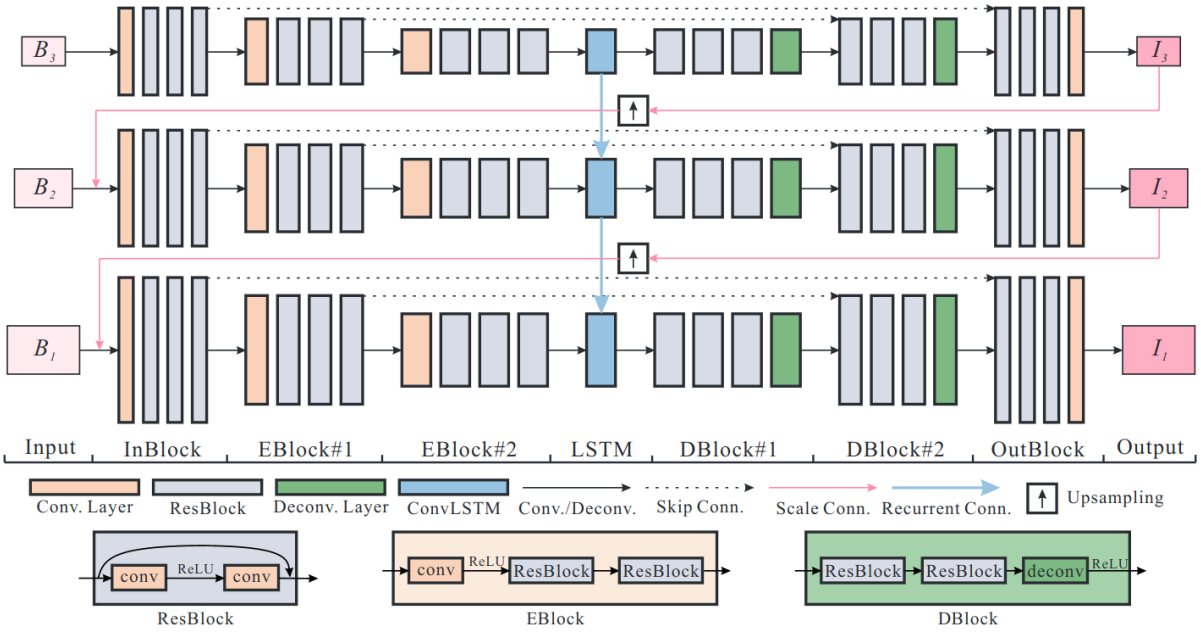



Figure 5. Scale Recurrent Network architecture (Tao et al., 2018)

And the following equation explains how the network works:

$$\mathbf{I}^i, \mathbf{h}^i = \mathbf{Net}_{SR}(\mathbf{B}^i, \mathbf{I}^{i+1\uparrow}, \mathbf{h}^{i+1\uparrow}; \theta_{SR})$$

$\mathbf{B}^i$ - blurred image, $\mathbf{I}^i$ - sharp image, $\mathbf{h}^i$ - hidden state. $\mathbf{Net}_{SR}$ - recurrent network with $\theta_{SR}$ as a set of shared parameters that run across the network.
Recurrent networks may take different variations, but Tao et al. (2018) achieved the best results by using long-short term memory (LSTM), or in case of image processing - ConvLSTM. In order to adapt the output from the previous scale to the current scale using deconvolution layers, image resizing and sub-pixel convolution (Shi et al., 2016) techniques would be a good idea. However, in order to keep the network's efficiency and simplicity - simple bilinear interpolation was exploited.

#### 2.3.1.2 Encoder/Decoder ResBlocks

Encoder/Decoder networks are networks that use Convolution layers in order to extract feature maps by decreasing input size and increasing the number of channels during the encoding phase. During decoding the output of the encoder is upsampled. Usually, such networks are implemented using skip-connections. Such an approach is simple and proved its efficiency (Liu et al., 2017; Su et al., 2016; Xin et al., 2017; Xu et al., 2017). Even with all that, there are still issues that need to be addressed. First, a deblurring task requires more layers to be stacked, since a deblurring task requires a receptive field to be larger. On top of that feature maps that are extracted from middle layers of encoders would contain very little spatial information to properly reconstruct the image.

Second problem comes from the first issue as well - time complexity. (Tao et al., 2018) notices the more layers we have - the more parameters and weights there are to update, hence, the network's convergence becomes slower.
In order to deal with challenges mentioned above, Tao et al. (2018) suggests using modified Residual Blocks, presented in He et al. (2015). Custom encoder and decoder blocks are proposed, Encoder ResBlocks (EBlocks) and Decoder ResBlocks (DBlocks).

EBlocks consist of one convolution layer with stride 2 followed by ResBlocks. DBlocks is the vice versa - ResBlocks followed by a convolution layer. ResBlocks consist of two convolution layers.

#### 2.3.1.3 Loss

(Tao et al., 2018) was inspired by the following Euclidean loss and used it for each scale:

$$\mathcal{L} = \sum_{i=1}^{n} \frac{\kappa_i}{N_i} \|I^i - I_*^i\|_2^2$$

$I^i$ and $I_*^i$ are deblurred images and observed samples. (Tao et al., 2018) considered L2 worth mentioning as good enough loss for generating sharp images.

#### 2.3.1.4 Experiments

There are different varieties of the proposed approach built and applied by Tao et al. (2018). SR - scaler-recurrent network, which is used for the following architectures:

- SR-Flat - convolution layers instead of encoder/decoder structure
- SR-RB - ResBlocks instead of EBlocks and DBlocks
- SR-ED - simple 2 convolution layer blocks replace ResBlocks
- SR-EDRB1, SR-EDRB2, SR-EDRB3 share the same structure but with 1, 2 and 3 ResBlocks respectively

Results of experiments are portrayed in the following table:

| Model | SR-Flat | SR-RB | SR-ED | SR-EDRB 1 | SR-EDRB 2 | SR-EDRB 3 |
|---|---|---|---|---|---|---|
| PSNR | 27.53 | 28.11 | 29.06 | 28.60 | 29.32 | 29.98 |
| SSIM | 0.8886 | 0.8991 | 0.9170 | 0.9082 | 0.9204 | 0.9254 |

*Table 4. SRN evaluation results (Tao et al., 2018)*

#### 2.3.1.5 Conclusion

Tao et al. (2018) has a simple and straightforward multi stage architecture that processes blurred images at several scales and each scale has a simple and clear design. Loss function used in training is straightforward as well finding a difference between predicted result and ground truth. SRN proved to be precise and quick to converge and is considered to be a SOTA model for image restoration that sets up a high bar for other future solutions.

### 2.3.2 MRPNet

Zamir et al. (2021) states that multistage models that employ encoder-decoder subnetworks manage to get a very good contextual information, which is really important when dealing with image restoration, denoising or deblurring tasks, although the cost of such achievement is spatial information, because encoder-decoder networks use the principle of several image downsampling and upsampling techniques. On the other hand, single stage models described in (Anwar et al., 2020; Dong et al., 2015; Zhang et al., 2017; Zhang et al., 2018) excel at achieving good spatial information gain, however, they tend to be robust to semantic details. At this point, it is either we get good spatial information gain or reliable contextual understanding, but not both at the same time. In order to make up for such a gap, a three stage model was proposed - multi-stage progressive image restoration architecture, named MRPNet (Zamir et al., 2021). In the first two stages subnetworks that are based on encoder-decoders were employed and at the last stage a subnetwork that was operating on the original image without changing its size or dimensions was applied.
At each stage deblurred images **X** are obtained by adding up blurred images **R** and residual images **I**. Whole architecture is portrayed in Figure 6.

$$\mathbf{X} = \mathbf{R} + \mathbf{I}$$

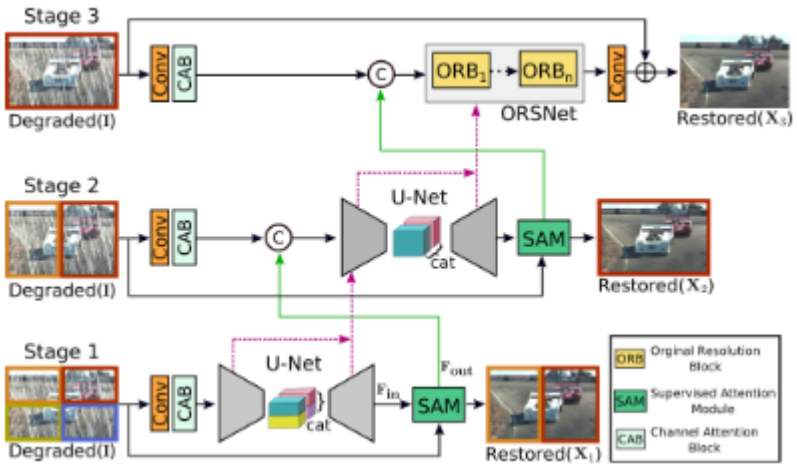



Figure 6. MRPNet (Zamir et al., 2021)

#### 2.3.2.1 Encoder-decoder

As was mentioned before, the first two stages employ encoder-decoder subnetworks. These subnetworks are based on U-Net architecture presented in Ronneberger et al. (2015). At the beginning of each stage, according to Zamir et al. (2021), Channel Attention Blocks (CABs), that are based on methods provided by Xing & Zhang (2019) were implemented in order to extract features and are portrayed in Figure 7(a). In the encoder-decoder subnetwork skip connections are run through CABs as well. After this whole process at the end bilinear interpolation for upsampling followed by a convolution layer.

#### 2.3.2.2 Original Resolution Network

For the last third stage a Original-Resolution Subnetwork (ORSNet) was built, consisting of multiple Original-Resolution Blocks (ORBs). ORBs themselves consist of multiple CABs. ORSNet does not use any downsampling techniques and operates on the input without changing its shape. The subnetwork helps extract spatially enriched features. Subnetwork design is depicted on Figure 7(b).

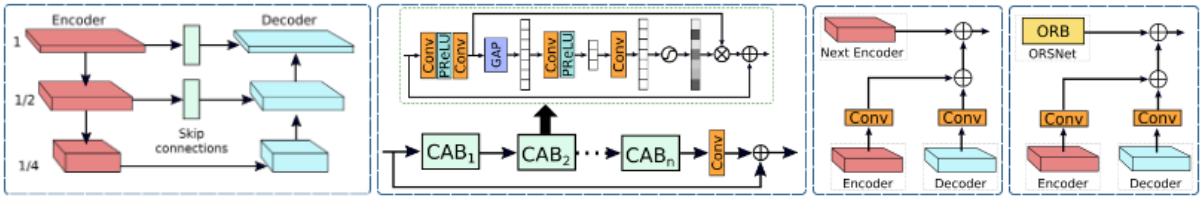
#### 2.3.2.3 Cross-stage Feature Fusion

There are two additional subnetworks between encoder-decoders and between encoder decoder and ORSNet that are called cross-stage feature fusion (CSFF). Outputs from one stage are passed through 1x1 convolution onto the next stage. This module reduces the information loss, improves information gain in features of the next stage and eases the flow of information in the whole architecture. CSFF units located between first and last two stages are portrayed in Figure 7(c) and Figure 7(d) respectively.

#### 2.3.2.4 Supervised Attention Model

In order to mitigate less informative features from the current stage and pass them onto the next stage, between every two stages, (Zamir et al., 2021) implemented a Supervised Attention Model that was used to get useful features through convolution layers and using ground-truth loss and attention maps. The Supervised Attention Model's design is depicted on Figure 8.

The Supervised-Attention Model network runs extracted features from the current stage and gets residual image **R** using 1x1 convolution and adds it to the degraded image **I**, thus obtaining **X**. After getting ground-truth loss **X** is passed onto the attention maps block, which is then multiplied to **R** image pointwise. And this output is then added again to the same **R** residual image.




(a) (b) (c) (d)

Figure 7. a-encoder decoder subnetwork, b - ORB structure that is the main component of ORSNet blocks, GAP - global average pooling, c - CSFF between first two stages, d - CSFF between last two stages (Zamir et al., 2021)

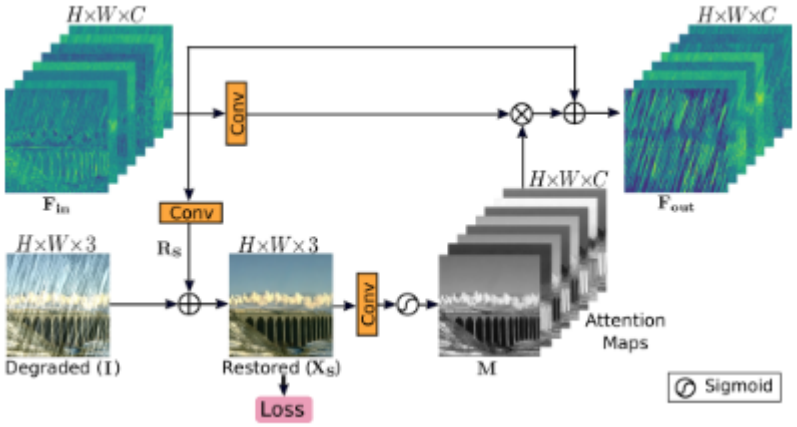



Figure 8. Supervised Attention Model design (Zamir et al., 2021)

### 2.3.2.5 Loss

Zamir et al. (2021) combined two types of losses - Charbonnier and edge losses. Charbonnier loss (Charbonnier et al., 1994) and can be formed the following way:

$$\mathcal{L}_{char} = \sqrt{\|\mathbf{X}_S - \mathbf{Y}\|^2 + \varepsilon^2}$$

$X_S$ here is the obtained sharp image and Y is the observed image, ε here is the constant that was denoted as $10^{-3}$ by Zamir et al. (2021)

Edge loss has the following formulation that (Zamir et al., 2021) came up with:

$$\mathcal{L}_{edge} = \sqrt{\|\Delta(\mathbf{X}_S) - \Delta(\mathbf{Y})\|^2 + \varepsilon^2}$$

Δ here is the Lagrangian operator. Together combined the above losses form the following loss employed by Zamir et al. (2021):

$$\mathcal{L} = \sum_{S=1}^{3} [\mathcal{L}_{char}(\mathbf{X}_S, \mathbf{Y}) + \lambda \mathcal{L}_{edge}(\mathbf{X}_S, \mathbf{Y})]$$

### 2.3.2.6 Experiments and results

The following table is the table of results for MRPNet obtained from evaluating on HIDE, GoPro, RealBlur-R and RealBlur-J datasets.

| MRPNet | PSNR | SSIM |
|---|---|---|
| GoPro | 32.66 | 0.959 |
| HIDE | 30.96 | 0.939 |

| | | |
|---|---|---|
| RealBlur-R | 39.31 | 0.972 |
| RealBlur-J | 31.76 | 0.922 |

*Table 5. MRPNet evaluation results (Zamir et al., 2021)*

Vision Transformer (ViT) is a combination of computer vision and Natural Language Processing (NLP) principles.

#### 2.3.2.7 Conclusion

Zamir et al. (2021) presented a multi-stage model that is capable of solving numerous image restoration tasks, such as image deraining, denoising and deblurring. This approach is an excellent example that defines a multi-stage approach. Numerous subnetwork architectures were implemented and combined together using UNet like shape in a single stage. This model is thorough and does not use any pre-built structures, just like SRN. Loss used for training is simple and straightforward as well. Such an approach is straightforward and is able to outperform some single stage approaches, like DeblurGAN. The main advantage is the number of layers and number of parameters, that compensate for the simplicity of error loss methods. This may lead to memory issues of computational cost, compared to single shot architectures.

## 2.4 Multi stage vs single stage approaches comparison

As it is possible to observe - single stage approaches tend to have a backbone that employs popular frameworks and restructures them for a deblurring task as it is done by Kupyn et al. (2018) or use a pre-trained model for both the model structure and loss during training as done by Orest Kupyn et al. (2019) using perceptual loss with other loss approaches.

When it comes to multi stage approaches compared to single shot methods, they tend to use principles of basic algorithms presented by other authors for different computer vision tasks instead of using pre-trained models and frameworks of known famous models. It is worth noting that loss functions for such approaches tend to be relatively simple and straightforward (Tao et al., 2018; Zamir et al., 2021).

Single stage models tend to be quicker and less complex in terms of structure, however training techniques require more involvement and attention to dig into, while multi stage approaches have, at first, complex approach, but is also simple and straightforward as most of them are mainly based on using CNNs and a lot of scaling operations. Not to mention the training process is also simple.

Both of these approaches have their own ups and downs. However, as it was mentioned before, these methods are built using CNN structure, which are good for learning local context and dependencies, which in case of image deblurring might lead to inaccuracies and should be compensated with either more complex structure or advanced training techniques. Hence, the idea of combining the extraction of local dependencies and the ability of learning global context using self-attention blocks might bring fruitful results to such a problem.

## 2.5 Transformer based models

Vaswani et al. (2017) presented a novel approach for natural language processing tasks in order to process sequential data and getting attention features. The approach is known as the Transformer model with the Self-Attention (SA) as the core component. Dosovitskiy et al. (2020) used the approach proposed in (Vaswani et al., 2017) and proposed Vision Transformer (ViT) architecture for image recognition tasks. As noted by Wang et al. (2021) noticed that SA effectively captures long-range pixel interactions, but its computational complexity increases quadratically with the spatial resolution, which makes it vulnerable to high-resolution images (Zamir et al., 2022; Wang et al., 2021).

### 2.5.1 Transformer - basics

In order to add more context to transformers that are being talked about a lot and dig more into the roots of deblurring images using self-attention blocks, the original Transformer model introduced by Vaswani et al. (2017) and ViT model proposed by Dosovitskiy et al. (2020) are to be shortly described and reviewed.

#### 2.5.1.1 Architecture

Transformer model architecture (Vaswani et al., 2017) consists of encoders and decoders that are full of stacked self-attention blocks and pointwise, fully-connected layers with normalisation. The whole architecture can be viewed on Figure 9.

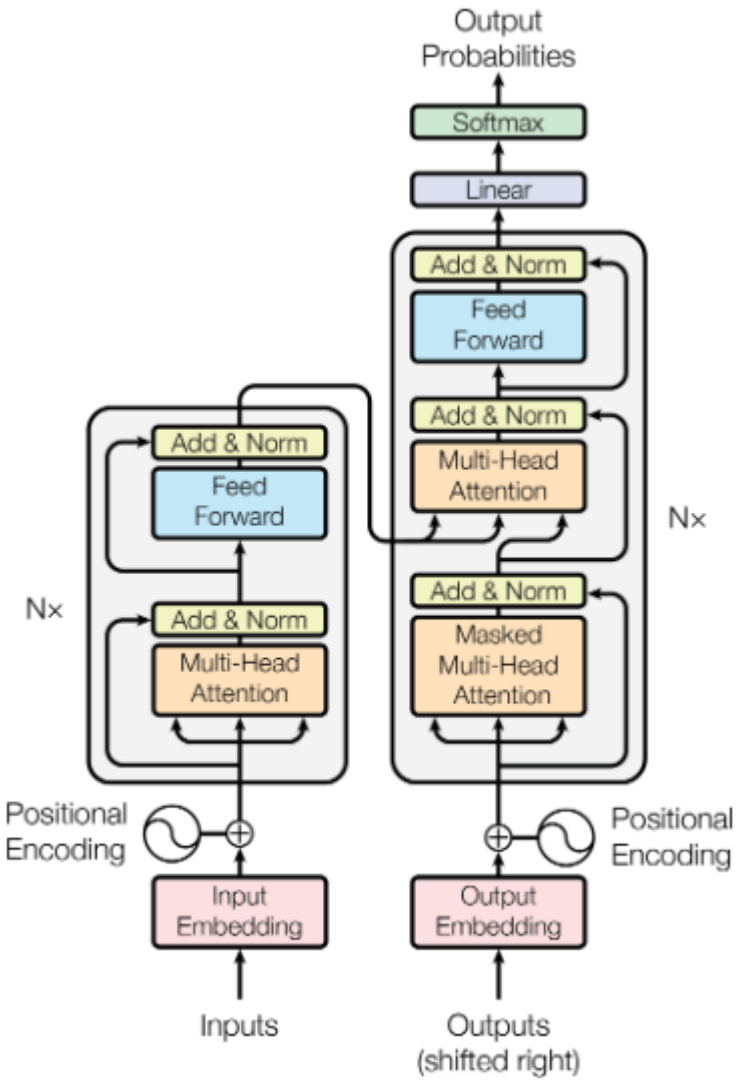



Figure 9. Transformer model architecture proposed by Vaswani et al. (2017)

#### 2.5.1.2 Attention head and multi-headed solution

"An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key." (Vaswani et al., 2017)

Such a formulation of the attention head is depicted in the following notation:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

The design of the flow in the whole architecture is depicted on Figure 10.

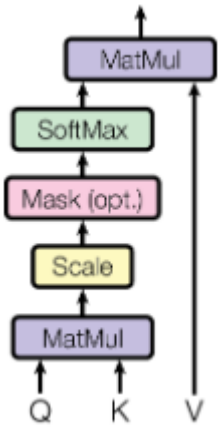



Figure 10. Scaled-dot product attention of (Vaswani et al., 2017)

## 2.5.2 Vision Transformer

Dosovitskiy et al., (2020) successfully used Transformer architecture (Vaswani et al., 2017) as close to the original as possible adapted to image processing tasks.
The ViT architecture is depicted on Figure 11.

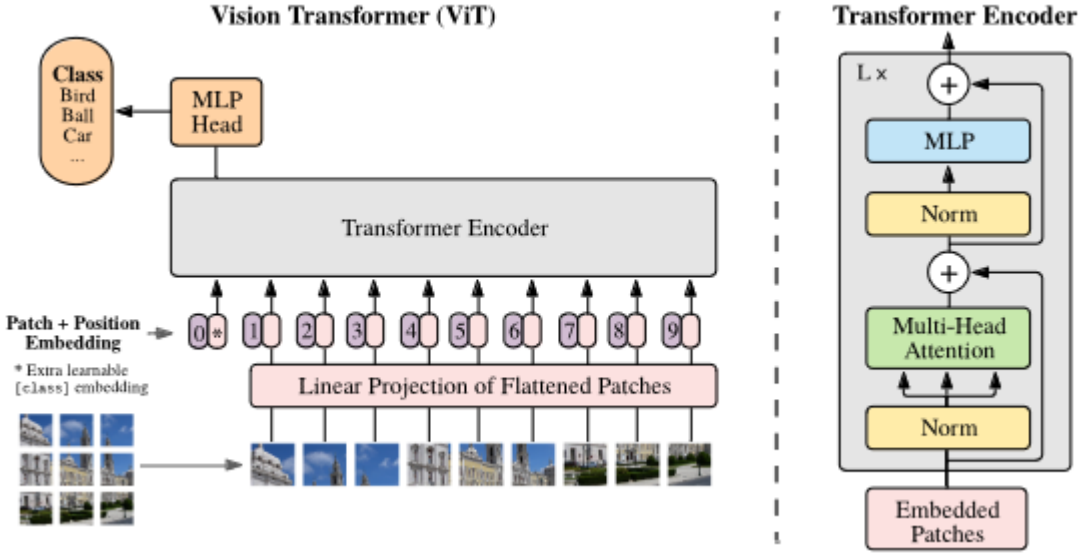



Figure 11. Vision Transformer (ViT) model architecture (Dosovitskiy et al., 2020)

In order to make the input for the transformer, images of size HxWxC (height, width, channels respectively) were flattened 2D patches of size 16x16. Patches are combined with 1D patch embeddings, which have remained the same since (Dosovitskiy et al., 2020) observed no performance gains from adding advanced 2D embedding layers.

Transformer block contains multiple Multi-head self-attention (MSA) heads and Multi-layer Perceptron (MLP) blocks with GELU activations that alternate with each other as intended by Dosovitskiy et al. (2020)

Another contrasting feature is that in CNNs local information of each layer is extracted, while for ViT includes locality in MLP blocks and globally learning layers in Transformers block.

The following works described mostly use the principle of Transformers or use ViT, but indirectly as it was for Li et al (2023) that used the Segment Anything Model (Kirillov et al., 2023), for boosting the deblurring combined with mean average pooling layers and Non-Linear Activation Free network (NAFnet) (Chen et al., 2022).

## 2.5.3 Restormer model

Zamir et al. (2022) proposed a new multi-Dconv head 'transposed' attention(MDTA) block and used it instead of traditional SA blocks along with a new version of traditional feed-forward network (FN) used by conventional Transformer structure - gated Dconv FN (GDFN). These novelties were used to build Restormer - an encoder-decoder Transformer that would be able to keep the effectiveness of keeping global connectivity extraction on large-scaled images. The architecture shown in Figure 12.

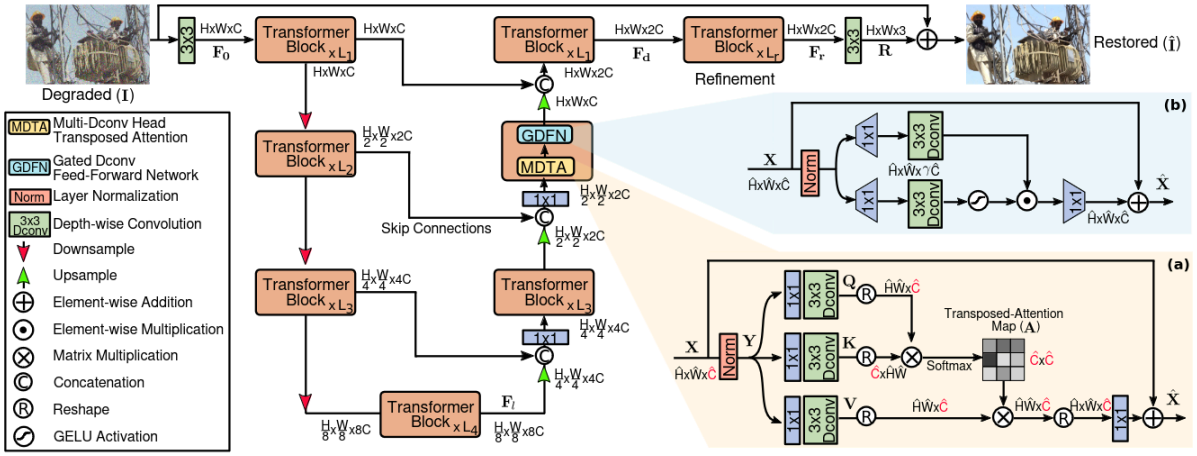



Figure 12. Restormer model architecture (Zamir et al., 2022)

### 2.5.3.1 Multi-Dconv Head Transposed Attention

As it was mentioned before, traditional SA architecture tends to have the overall complexity grow quadratically the higher the spatial resolution gets. Zamir et al. (2022) proposed their version of attention block - MDTA that has a similar structure to SA block, however the query (**Q**), key (**K**) and value (**V**) projections are generated from image **Y** using 1x1 depth-wise convolution $\mathbf{W}_p$ for local context extraction and 3x3 convolution $\mathbf{W}_d$ to gather channel-wise spatial contextual information. $\mathbf{Q} = \mathbf{W}_d^q \mathbf{W}_p^q \mathbf{Y}$, $\mathbf{K} = \mathbf{W}_d^k \mathbf{W}_p^k \mathbf{Y}$, and $\mathbf{V} = \mathbf{W}_d^v \mathbf{W}_p^v \mathbf{Y}$ are then reshaped into $\hat{\mathbf{Q}}, \hat{\mathbf{K}}, \hat{\mathbf{V}}$ tensors respectively to generate an attention map **A**:

$$\text{Attention}\left(\hat{\mathbf{Q}}, \hat{\mathbf{K}}, \hat{\mathbf{V}}\right) = \hat{\mathbf{V}} \cdot \text{Softmax}\left(\hat{\mathbf{K}} \cdot \hat{\mathbf{Q}} / \alpha\right)$$

$$\hat{\mathbf{X}} = W_p \text{ Attention } \left(\hat{\mathbf{Q}}, \hat{\mathbf{K}}, \hat{\mathbf{V}}\right) + \mathbf{X}$$

Here $\hat{\mathbf{X}}$ and $\mathbf{X}$ are output and input features respectively. The whole MDTA structure is portrayed on Figure 12(a).

"This strategy provides two key advantages. First, it emphasises on the spatially local context and brings in the complimentary strength of convolution operation within our pipeline. Second, it ensures that the contextualised global relationships between pixels are implicitly modelled while computing covariance-based attention maps." (Zamir et al., 2022)

#### 2.5.3.2 Gated-Dconv Feed-Forward Network

The original Transformer's FN has two fully connected layers connected by a non-linear unit. A novel FN structure was constructed to controls the information flow - GDFN which also has two convolutional 1x1 layers. First layer expands and the second one reduces the number of feature channels. And similar to conventional FN they are connected by Gaussian Error Linear Unit (GELU) described in (Hendrycks & Gimpel, 2020). This principle is referred to by Zamir et al. (2022) as a gating mechanism.
GDFN can be formulated as:

$$\text{Gating}(\mathbf{X}) = \phi(W_d^1 W_p^1(\text{LN}(\mathbf{X}))) \odot W_d^2 W_p^2(\text{LN}(\mathbf{X}))$$

$$\hat{\mathbf{X}} = W_p^0 \text{ Gating } (\mathbf{X}) + \mathbf{X}$$

The following annotation are referred by Zamir et al. (2022) as ⊙ - element-wise multiplication, Ø - GELU, LN - layer normalisation. This notation has design that is depicted in Figure 12(b).

"The gating mechanism in GDFN controls which complementary features should flow forward and allows subsequent layers in the network hierarchy to specifically focus on more refined image attributes, thus leading to high-quality outputs." (Zamir et al., 2022)

#### 2.5.3.3 Training and results

A novel way was used to train their Transformer model - progressive learning. Small patched inputs and big batches are passed at the start and the more model trains - patches become bigger while batch size smaller.
Model achieved the following results:

| Restormer | GoPro | HIDE | RealBlur-R | RealBlur-J |
|---|---|---|---|---|
| PSNR | 32.92 | 31.22 | 36.19 | 28.96 |
| SSIM | 0.961 | 0.942 | 0.957 | 0.879 |

*Table 6. Restormer evaluation results (Zamir et al., 2022)*

#### 2.5.3.4 Conclusion

Zamir et al. (2022) implemented their own version of self-attention blocks adapted to effectively processing images without risking the increment of computational complexity quadratically and added a gating mechanism for refining weights. Attention blocks capture long-range dependencies and gated FFN is designed for local information gain. This model proved to be efficient in learning global context using convolutional attention, thus preventing high computational costs.

### 2.5.4 Uformer

Another excellent example of Transformer structure's use in image restoration techniques is Unformer model designed, implemented and trained by Wang et al. (2021)
The model's architecture uses their unique SA blocks and the whole design is based on U-net (Ronneberger et al., 2015). Figure 13(a) demonstrates the whole pipeline with quite unique methods: Locally Enhanced Window (LeWin) and multi-stage learnable modulator designed by Wang et al. (2021)

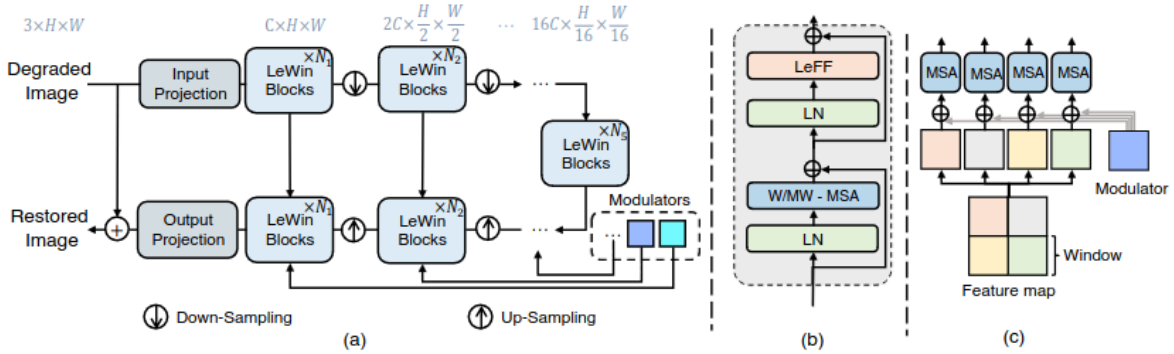



Figure 13. a-Uformer architecture, b-LeWin design, c-Window-based Multi-head Self-Attention (Wang et al., 2021)

#### 2.5.4.1 Locally-enhanced Window Transformer

Wang et al. (2021) and Zamir et al. (2022) stated that traditional SA blocks usually struggle at keeping reasonable computational cost at higher resolution images in order to gain attention and context globally between all tokens. According to Wang et al. (2021) another problem with SAs is that they are limited to capturing local dependencies.
LeWin was designed to address issues mentioned earlier. LeWin itself consists of two core structures: non-overlapping Window-based Multi-head Self-Attention (W-MSA) and Locally-enhanced Feed-Forward Network (LeFF). The whole LeWin process could be represented the following way:

$$\mathbf{X}'_l = \text{W-MSA}(\text{LN}(\mathbf{X}_{l-1})) + \mathbf{X}_{l-1}$$

$$\mathbf{X}_l = \text{LeFF}(\text{LN}(\mathbf{X}'_l)) + \mathbf{X}'_l$$

**Window-based Multi-head Self-Attention (W-MSA).** It works as follows:

$$\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^N\}$$

X is a 2D feature map being split into  a number N of non-overlapping windows.

$$\mathbf{Y}_k^i = \text{Attention}(\mathbf{X}^i\mathbf{W}_k^Q, \mathbf{X}^i\mathbf{W}_k^K, \mathbf{X}^i\mathbf{W}_k^V)$$

$$\hat{\mathbf{X}}_k = \{\mathbf{Y}_k^1, \mathbf{Y}_k^2, \cdots, \mathbf{Y}_k^M\}$$

Attention is gathered throughout non-overlapping windows and concatenated into output $\hat{\mathbf{X}}_k$

$\mathbf{W_k^Q}$, $\mathbf{W_k^K}$, $\mathbf{W_k^V}$ are query, key and value projections respectively for the k-th head. Methods were also used, provided by Liu et al. (2021) and Shaw et al. (2018), to modify the method attention, calculated the following way:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{SoftMax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} + \mathbf{B})\mathbf{V}$$

**B** is the relative position bias. And according to Wang et al. (2021) W-MSA was designed to deal with computational complexity for various image resolutions.

**Locally-enhanced Feed-Forward Network.** When explaining the reason for designing the LeFF (Wang et al., 2021) was referring to solving the problem of gaining proper local context addressing works of (Wu et al., 2021) and (Yuan et al., 2021). It would also be worth mentioning Restormer model's author - (Zamir et al., 2022) who also addressed this issue and solved it with their GDFN structure. The LeFF structure is depicted on Figure 14, all layers are connected by GELU activation function.

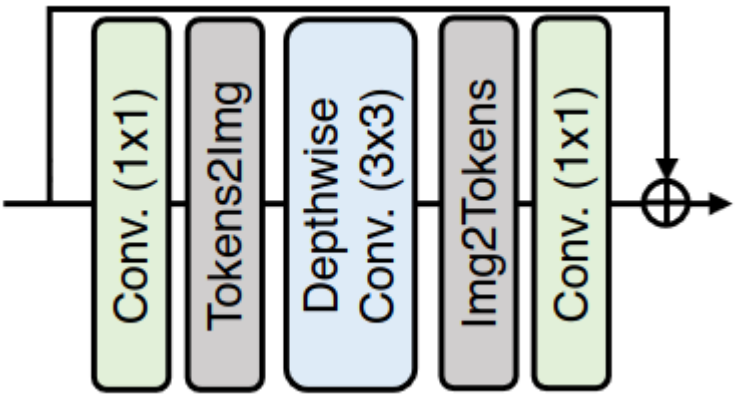



Figure 14. Locally-enhanced Feed-Forward Network (Wang et al., 2021)

### 2.5.4.2 Multi-Scale restoration modulator

(Wang et al., 2021) added a modulator to each LeWin block in the decoder part as a learnable tensor of size MxMxC, where M - size of the image and C-number of learnable

channels. (Wang et al., 2021) narrated this tensor to be a bias shared for all non-overlapping windows in W-MSA. The modulator provided a performance boost by adjusting feature maps at each stage.

#### 2.5.4.3 Uformer's performance benchmark

| Uformer | GoPro | HIDE | RealBlur-R | RealBlur-J |
|---|---|---|---|---|
| PSNR | 32.97 | 30.83 | 36.22 | 29.06 |
| SSIM | 0.967 | 0.952 | 0.957 | 0.884 |

*Table 7. Uformer evaluation results (Wang et al., 2021)*

#### 2.5.4.4 Conclusion

Wang et al., (2021) proposed a UNet shaped network consisting of LeWin blocks. LeWin blocks are transformer blocks solving the problem of self-attention layers' computational costs mentioned by Zamir et al. (2022) using local windows. Novel LeFF FFN technique was implemented for LeWin blocks to be capable of gaining local context along with long-range dependencies and attention maps from the use of W-MSA. Uformer introduced an architecturally comprehensible and computationally reasonable model that combines advantages of extracting local features and capturing global information from both CNN and SA structures respectively.

### 2.5.5 SAM-Deblur

(Li et al., 2023) used a non-standard approach to image deblurring - boosting the existing deblurring state-of-the-art (SOTA) model with segmentation maps generated by segmentation model to regularise solution space.

(Li et al., 2023) decided to use the Segment Anything Model (Kirillov et al., 2023) for semantic priors processing due to its robustness to blurred images. Those priors are then run through Mask Average Pooling (MAP) layers and then processed using Nonlinear Activation Free Network (NAFNet).

#### 2.5.5.1 Segment Anything Model

(Kirillov et al., 2023) presented not only a simple model architecture, but also new tasks that empowered training and made it more productive and efficient for the whole task of semantic segmentation. The whole foundation model for segmentation tasks is depicted on Figure 15.

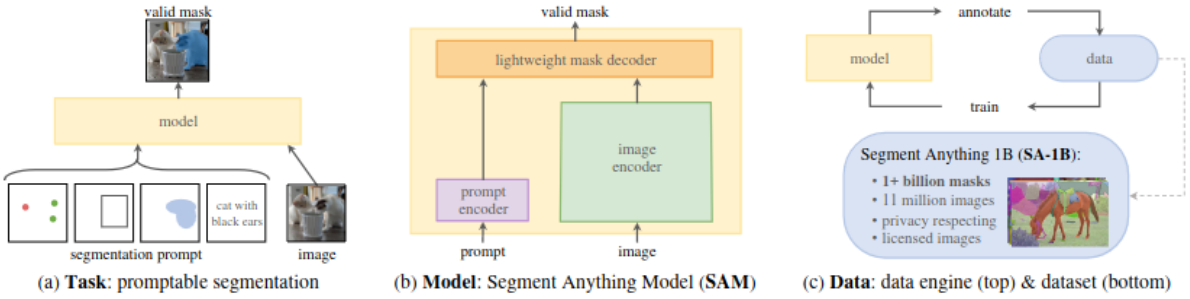



Figure 15. Foundation model pipeline for image segmentation (Kirillov et al., 2023)

(Kirillov et al., 2023) was inspired by NLP tasks that involved prompting and involved the same principles into segmentation tasks - providing masks according to an input prompt, even if the description provided seems to be vague and unclear it would provide something as close to it as possible and after predicting masks they are compared to ground truths. This yields a very efficient pre-training algorithm.

Kirillov et al. (2023) provided a very simple model in terms of architecture. As illustrated on Figure 15(b) there are only three key components: image encoder, prompt encoder and mask decoder. Image encoder phase employs ViT in order to process images. Prompt encoder sums up embedded sparse prompts (points and boxes, etc.) represented through positional encoding and dense prompts (masks) summed up point wise with embedded images. Decoder employs a modified version of Transformer decoder's structure (Vaswani et al., 2017). "Our modified decoder block uses prompt self-attention and cross-attention in two directions (prompt-to-image embedding and vice-versa) to update all embeddings. After running two blocks, we up-sample the image embedding and an MLP maps the output token to a dynamic linear classifier, which then computes the mask foreground probability at each image location" (Kirillov et al., 2023)

### 2.5.5.2 NAFNet

Chen et al., (2022) presented a UNet shaped model that combines several principles of different approaches, including those listed and described above, yielding a model that has low intra- and inter-block complexity with modified gated linear units (GLU) (Shazeer, 2020) as layers that multiply two linear transformations of the input and are connected using activation functions. Chen et al., (2022) used convolution, GELU activation function and a skip-connection in their blocks that are stacked in a UNet style. Depthwise convolutions constructed as simplified channel attention layers in plain blocks also play a role of self-attention blocks, based on channel attention layers (Hu et al., 2019) in order to increase the effectiveness of learning global information and is depicted on Figure 16(b).

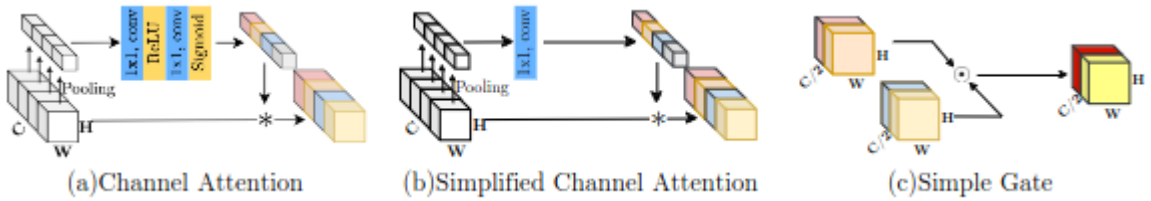


Figure 16. a - Channels attention implemented by Hu et al. (2019), b - Simplified Channel Attention (Chen et al., 2022), c - Simple gate (Chen et al., 2022)

Simple Gate depicted on Figure 16(c) activation function is, as was mentioned before, a modified GLU structure that multiplies feature maps. The whole model architecture design is shown on Figure 17.

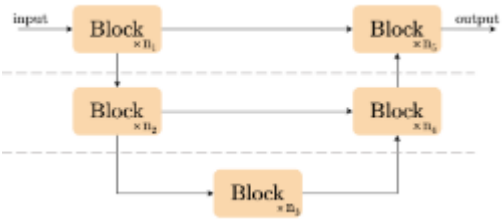



Figure 17. NAFNet architecture, proposed by Chen et al. (2022)

#### 2.5.5.3 Mask Average Pooling Unit

Mask Average Pooling (MAP) unit is the main component proposed by Li et al. (2023). At this point SAM's masks are used to boost the deblurring process. "MAP facilitates a rich interaction between image information and the segmentation masks generated by SAM, producing localised blur priors that are subsequently channel-concatenated with the original image and fed into the NAFNet." (Li et al., 2023)

#### 2.5.5.4 Results

| Metrics | GoPro | RealBlur |
|---|---|---|
| PSNR | 32.83 | 26.62 |
| SSIM | 0.96 | 0.867 |

*Table 8. SAM-Deblur evaluation results (Li et al., 2023)*

#### 2.5.5.5 Conclusion

Li et al. (2023) proposed a unique approach of using a pre-trained SAM for data preprocessing for semantic priors extraction, designed MAP layers for image and masks interaction and used a pre-built NAFNet, consisting of channel attention blocks in UNet shaped manner. This approach is an excellent portrayal of using transformer based models for image restoration problems, both directly (MAP and NAFNet) and indirectly (SAM image encoder).

## 2.6 Blurred Image to Video Restoration

There are only a few works that explore this problem. The first few authors (Jin et al., 2018; Purohit et al., 2019) who produced works of such kind - restoring video from motion

blurred images, did it by extracting directly from the network output and their works are worth mentioning as additional ways the deblurring problem is handled using modern approaches with the help of technological and theoretical advancements.

On the other hand, such articles as Zhang & Yang, (2015), used the vice versa principle - whole video deblurring. According to them motion between consecutive frames is established as well as the deblurring in each frame is done.

Purohit et al. (2019) presented a novel approach to the task of deblurring - video generation from the sequence of blurred images. Authors proposed a two-stage approach based on using Convolutional Neural Networks (CNNs). First stage is essentially an encoder network. Purohit et al. (2019) proposed an architecture that has two auto-encoders that are described as following: Recurrent Video Encoder (RVE) and Blurred Image Encoder (BIE). The second stage is a video decoder, based on recurrent networks architectures, that uses the outputs of BIE and deblurred sharp frames for video restoration - Recurrent Video Decoder (RVD). Recurrent networks proved to be highly efficient in sequence modelling, like for speech recognition tasks, like Graves et al. (2013). Therefore, Purohit et al. (2019) used Convolutional Long Short Term Memory (ConvLSTM) structure (SHI et al., 2015) to keep the benefits of sequence modelling using recurrent networks and to remain aware of spatial information across the image. Initially (Purohit et al., 2019) trains RVE and RVD pair using the following reconstruction loss:

$$\mathcal{L}_j = \sum_{n=1}^{N} \left| \hat{x}_{n,j} - x_{n,j} \right|_1$$

RVE-RVD network is pre-trained and then can be later on used to guide the training of BIE-RVD, a network that generates a video as well. BIE generates the same motion data as RVE does and passes it onto RVD. Training such a BIE-RVD pair from scratch would pose a threat to accuracy and training efficiency.

As described in Purohit et al. (2019) RVE consists of convolutional encoders that extract feature maps that are later one being passed on to a ConvLSTM module. As for BIE (Purohit et al., 2019) constructed for extracting motion features by capturing local motion.
RVD is divided into two phases: Flow Encoder and Flow Decoder connected by ConvLSTM
The whole architecture for the network of (Purohit et al., 2019) is described in Figure 18.

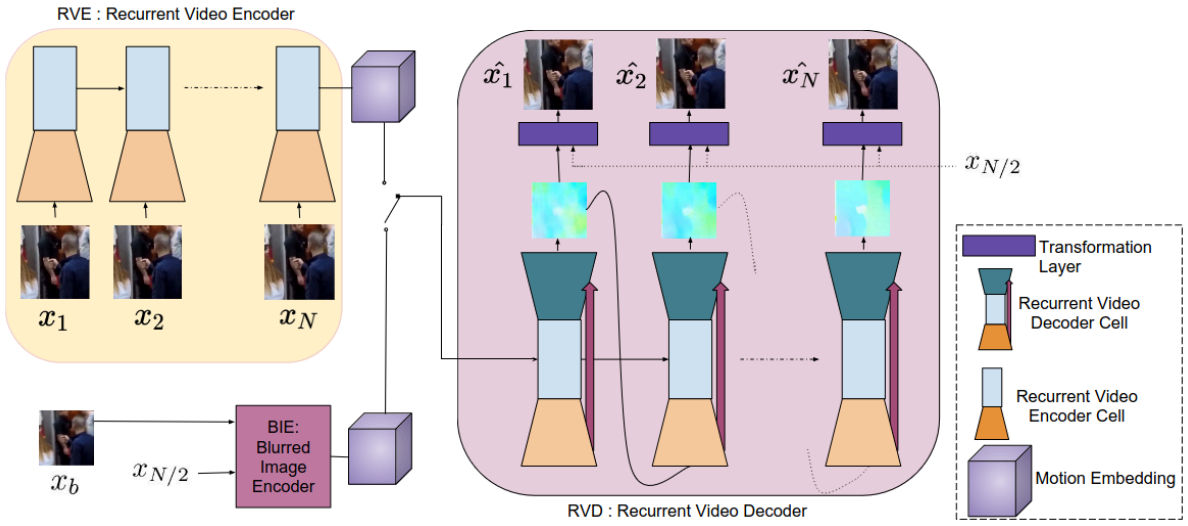



Figure 18. Model architecture (Purohit et al., 2019)

Image deblurring task achieved PSNR score of 30.58 and 0.941 for SSIM. For single image deblurring, the average time of convergence is 0.02 seconds and weight of the model is 17.9 megabytes(MB).

Results of benchmark experiments for a model trained by Purohit et al, (2019) were compared to the results and efficiency of Jin et al. (2018). In Jin et al. (2018) authors trained several neural networks models to generate a video frame by frame. The middle frame network is divided into three parts: extraction, refinement and fusion of features. In order to predict 7 frames 4 networks were used. 1 network generates a middle frame, while the other networks are used for middle-symmetric pairwise frames generation (Jin et al., 2018). Compared to the approach described in Jin et al. (2018), Purohit et al. (2019) proposed one that was shown to have had less issues with motion estimation consistency and had less local errors. Since the approach of Purohit et al. (2019) had two sub-tasks, one of which was responsible for resolving the task of deblurring and the other one was refining motion estimation, the resulting model had a more effective learning process as well as more promising results.

## 2.7 Summary

In reviewing a series of research papers regarding the problem of image deblurring it is clear that CNN-based solutions proved to be effective in alleviating blurring artefacts, enhancing image quality and generating a sharp image. However, an in-depth analysis and comparison of CNN-based approaches with Transformer based methodologies revealed that CNNs lack feasibility in capturing long-range dependencies, because of limited receptive fields and are robust to varying inputs, due to sharing weights for the same feature over all the locations. Transformers are good at capturing global context and weights are adaptable to input variability, but computational complexity increases quadratically with high resolution images, which might be expensive. Therefore Transformer models are modified or applied indirectly in order to avoid high computational costs.

Going forward, I am going to draw an inspiration from the works by Dosovitskiy et al., (2020), Zamir et al. (2022) and Wang et al. (2021), particularly their approach to creating attention blocks using CNNs in order to create a Decoder task for a pre-trained ViT image encoder. We will combine strengths of ViT in capturing global context for encoding and CNN's strong capability of extracting and learning local dependencies for decoding.

# Chapter 3: Dataset

Dataset that was used for training and testing is collected and uploaded into Kaggle by (Aleksey Alekseev, 2019) as an open source dataset. It consists of 350 triplets, which means there are 350 sharp, motion blurred and defocused images, which equals to 1050 images in total. However, for research only purposes and due to lack of technical hardware required only 350 motion blurred - sharp image pairs would be used, or 700 images in total. Alekseev states that the dataset itself is eligible to be used for image deblurring purposes, but due to the fact that triplets are not "pixel-to-pixel" PSNR and SSIM metrics might not be as trustworthy as with other datasets, hence visual samples are to be provided in experiments section.

# Chapter 4: Methodology

In order to achieve goals and overcome all the challenges set up by this problem, whole model and dataset management was done in Kaggle notebooks and its working space. Due to the nature of this project, it was decided to use online graphic accelerators kernels using NVIDIA TESLA P100 Graphic Processing Unit (GPU) provided by Kaggle platform.

In order to access images, the os package was used in order to handle system paths and access image paths. Using os package in combination with pandas the pandas data frame was created, consisting of sharp - blurred image pairs.
In order to read images and convert them into tensors of appropriate size, Python Imaging Library (PIL), Torchvision's transforms and numpy were used to create a data preprocessing pipeline. Pipeline is reading images using PIL, converts them to numpy arrays and transforms them to normalised tensors of appropriate size.

As it was stated before, the base behind the proposed method is the use of a ViT, pre-trained on ImageNet-1K dataset (Russakovsky et al., 2015), in combination with a CNN-based decoder. According to the official documentation, ViT extracts features using Transformer encoder block variation that is as close as possible. ViT learns global information from local windows and patches, which is necessary when dealing with degraded images.

## 4.1 Architecture

ViTDeblur is a single stage model and its architecture consists of a pre-trained ViT and CNN-based decoder, ViTDecoder. Architecture is depicted in Figure 19.

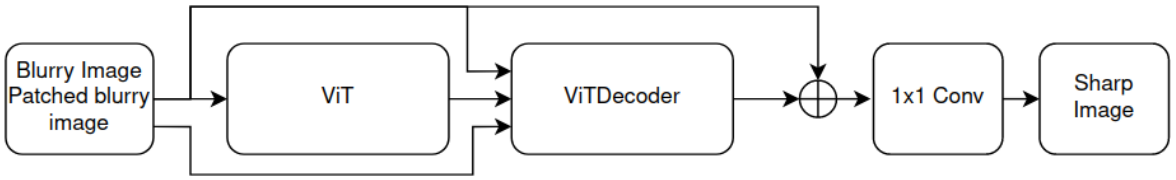



Figure 19. ViTDeblur architecture, where ⊕ is pointwise addition of tensors

## 4.2 ViTDecoder

ViTDecoder has two phases - upsampling phase and decoding phase. After ViT features are extracted, they are added to a patched blurred image, which would allow the model to apply attention maps to a patched blurred image and upsample the output tensor. Upsampled tensor is added to a full-scale blurry image for post-resampling information gain and to prepare the image for decoding blocks.

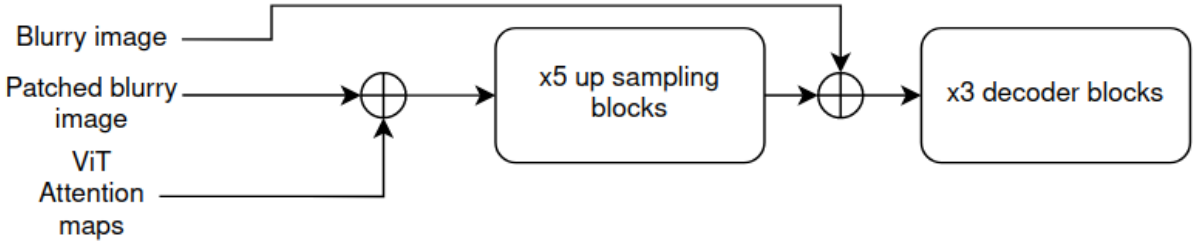



Figure 20. ViTDecoder design, where ⊕ is pointwise addition of tensors

### 4.2.1 Upsampling blocks

Upsampling phase provides pixel wise accuracy during decoding of the image. Decoder has 5 upsampling blocks, each consisting of bilinear interpolation upscaling layer and a 1x1 point wise convolution layer.


Figure 21. a) upsampling block, b) Decoder Block

### 4.2.2 Decoding blocks

Upsampled tensors are then added to the blurred image in order to apply learned attention maps and extract useful gradients for decoding. Each decoding block consists of a 1x1 point wise convolution layer that is activated using the GELU function.

## 4.3 Loss

The ViTDeblurring model is trained on two loss methods. First basic approach is using L1 loss:

$$L1 = \sum |y_{true} - y_{predicted}|$$

This loss function is a base loss used for style transfer architectures that are also focused on getting the image output from the image input (Neural Transfer Using PyTorch — PyTorch Tutorials 2.3.0+Cu121 Documentation, n.d.). L1 loss is also considered to be a content loss at this point.
Second optimisation loss used in training uses classical approach with Mean Squared Error loss, or as referred by Kupyn et al. (2018) is a content loss and is formed the following way:

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$

# Chapter 5: Training performance and Experiments

Dataset was split classically into a 80/20 ratio, which means 80% training shuffled data and 20% validating split. 20% of validation data were also split 50/50 into validation for training and evaluation using PSNR and SSIM metrics. Batch size during training was 1, which means the model was training on each image separately.
As it was mentioned before, the proposed approach has two variations trained on different losses and different training approaches: 1 - l1 loss with Adam optimiser (Kingma & Ba, 2014), 2 - content loss with Stochastic Gradient Descent (SGD) optimiser (Robbins & Monro, 1951).

During the first approach, the model is trained for 100 epochs using Adam optimisation algorithm with a starting learning rate of 0.01 that would be divided by 0.5 each time after 25 epochs passed using StepLR (*StepLR — PyTorch 1.9.0 Documentation*, n.d.). This variation is labelled as $\mathrm{ViTDeblur}_{adam}$.

For the second approach the model is trained for 100 epochs using SGD with a learning rate starting at value 0.1 with the same StepLR scheduler with a step size of 25 and update coefficient of 0.1, labelled as $\mathrm{ViTDeblur}_{SGD}$.

## 5.1 First approach

### 5.1.1 Training

The following figure provides an overview of training and evaluation batches error losses log during training:

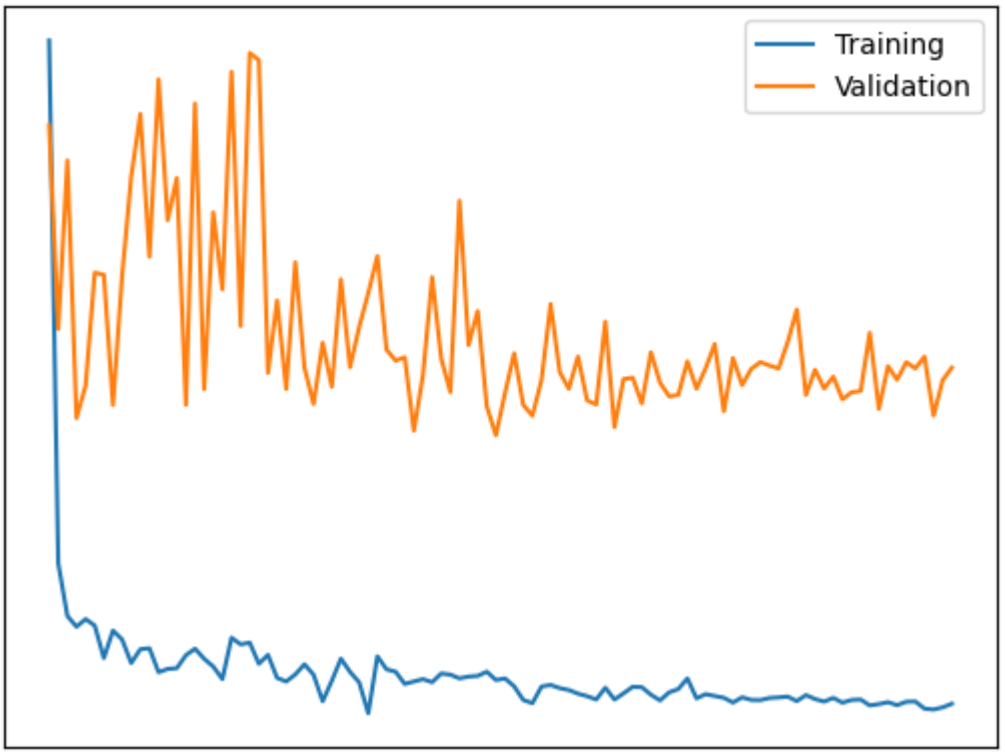



Figure 22. Training and validation losses logs during ViTDeblur$_{adam}$ training

Figure 23 provides average values of training and validation error losses for each 10th training epoch:

```
Epochs 10 train loss 0.7044688289186785 val loss 0.7428026258945465
Epochs 20 train loss 0.7032052205077239 val loss 0.7339262383324759
Epochs 30 train loss 0.7035435885723148 val loss 0.724887410231999
Epochs 40 train loss 0.7028178431093692 val loss 0.7285763255187443
Epochs 50 train loss 0.7031277875282935 val loss 0.7224246493407658
Epochs 60 train loss 0.7018547289073467 val loss 0.7251781923430306
Epochs 70 train loss 0.7024132775941065 val loss 0.7256102332047054
Epochs 80 train loss 0.7017131358385086 val loss 0.7279243205274855
Epochs 90 train loss 0.7015929988452366 val loss 0.725913838829313
Epochs 100 train loss 0.7012593468917268 val loss 0.7277588742119926
```

Figure 23. Losses after each 10 steps

Validation loss has a tendency for fluctuations, but is generally gradually decreasing and so does the fluctuation range, while training loss has more noticeable decline of the loss curve, which would have implied the overfitting problem, but according Figure 22, validation loss log's decrement is observable, even though it is barely visible. But generally the model adapts to unseen data. One of the most obvious reasons for that is because image triplets (in our case, pairs) are not pixel-to-pixel, which consequently leads to a sharp image being different compared to its blurred counterpart.. Another suggested solution is that If there were to be added more training epochs and learning rates are managed using more advanced techniques -

error loss curve would have been more steep. Training took 3 hours and the model has 87156316 numbers of parameters.

### 5.1.2 Evaluation

ViTDeblur$_{adam}$ achieved PSNR score of 48.45 and SSIM - 0.9 with a mean convergence time 0.16 seconds.

## 5.2 Second approach

### 5.2.1 Training

The following figure provides an overview of training and evaluation batches error losses log during training:

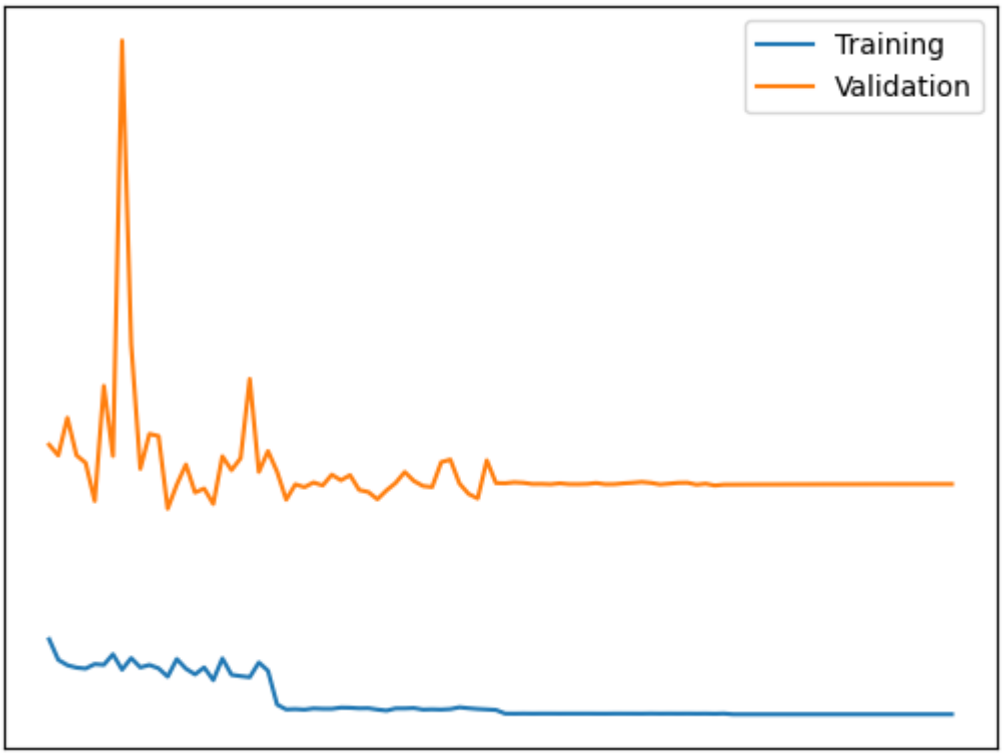



Figure 24. Training and validation losses logs during ViTDeblur$_{SGD}$ training

Figure 25 provides average values of training and validation error losses for each 10th training epoch:

```
Epochs 10 train loss 0.9169600901326963 val loss 1.0606631210872106
Epochs 20 train loss 0.916790051598634 val loss 1.009409669467381
Epochs 30 train loss 0.8938816487789154 val loss 0.9973993080002921
Epochs 40 train loss 0.8938934911042452 val loss 1.0022024137633188
Epochs 50 train loss 0.8931809975100415 val loss 0.9971536202090127
Epochs 60 train loss 0.891496932346906 val loss 0.9967295553003038
Epochs 70 train loss 0.8914771856474025 val loss 0.9972449541091919
Epochs 80 train loss 0.8912030022591353 val loss 0.9965124368667603
Epochs 90 train loss 0.8912016725433725 val loss 0.9966228944914681
Epochs 100 train loss 0.8912018106452056 val loss 0.9966705816132682
```

Figure 25. Losses after each 10 steps

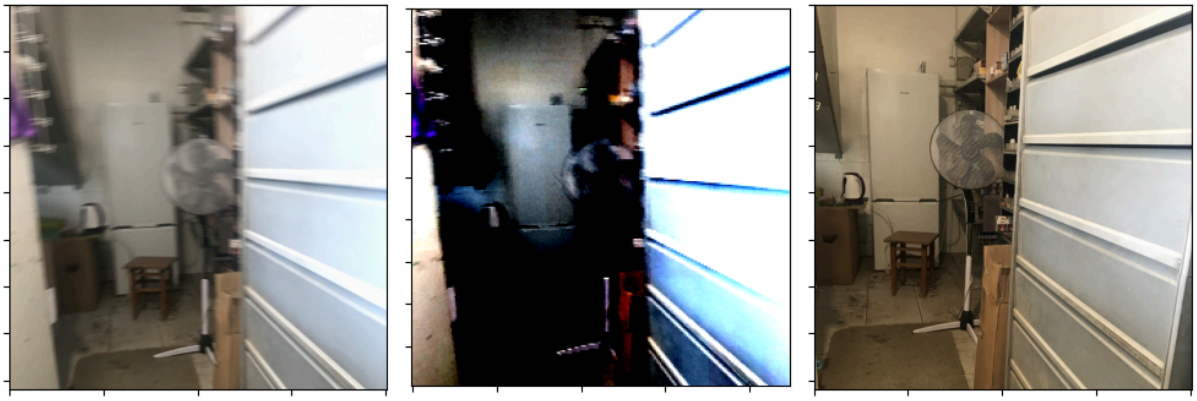
Figure 24 shows that training loss has less fluctuations with lesser intensity with decreasing error rate. Validation loss had some abrupt spikes of error loss at the very start, but later on the loss curve starts to even out and turns into a straight line, the same as the training loss curve. Main reason behind a slower rate of loss changes is the learning rate update coefficient of 0.1. With step sizes of 25 and 100 epochs, the learning rate between the last 25 epochs was between $10^{-4}$ and $10^{-5}$. Training took 2 hours and the model has 86250900 parameters.

### 5.2.2 Evaluation

ViTDeblur$_{SGD}$ achieved PSNR score of 48.75 and SSIM - 0.91 with a mean convergence time 0.15 seconds.

## 5.3 Visual Testing

As previously stated, images are not pixel-to-pixel, hence original scenes are slightly different compared to each other which means that such metrics as PSNR and SSIM might not be a good portrayal of model's performance. Therefore observing predicted results visually would give more insights into the models' performances. Figure 25 and Figure 26 show sets of images which provide a clear depiction of the effects achieved by the both variations of models and offer visual insight of their performance in enhancing image quality.




Midea

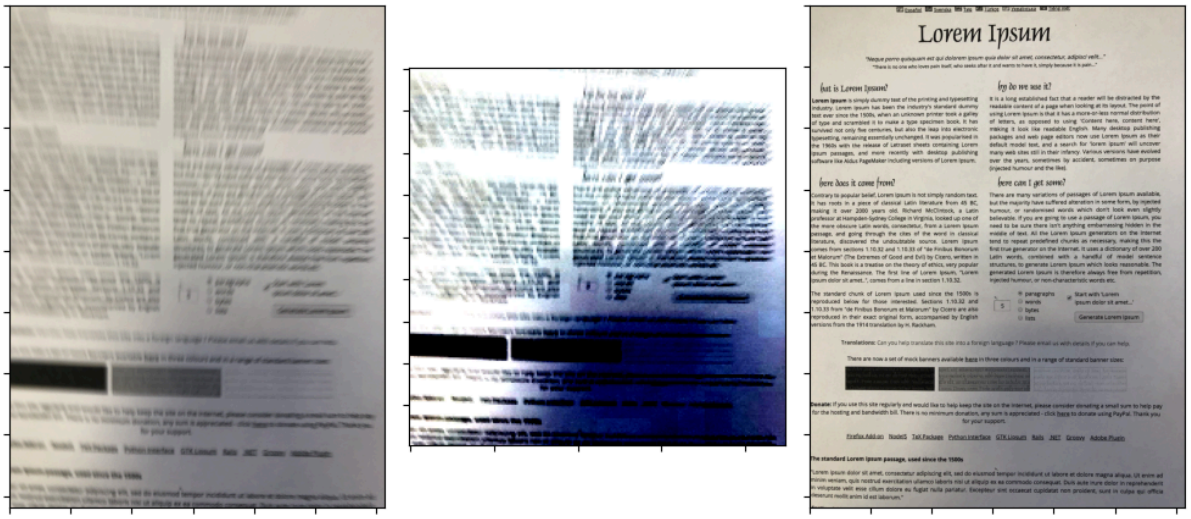



Figure 25. Visual comparisons of image deblurring using the ViTDeblur$_{adam}$ model. Image on the left is a blurred input, the middle image is a sharp image recovered from the trained model and the image on the right is the real sharp image.

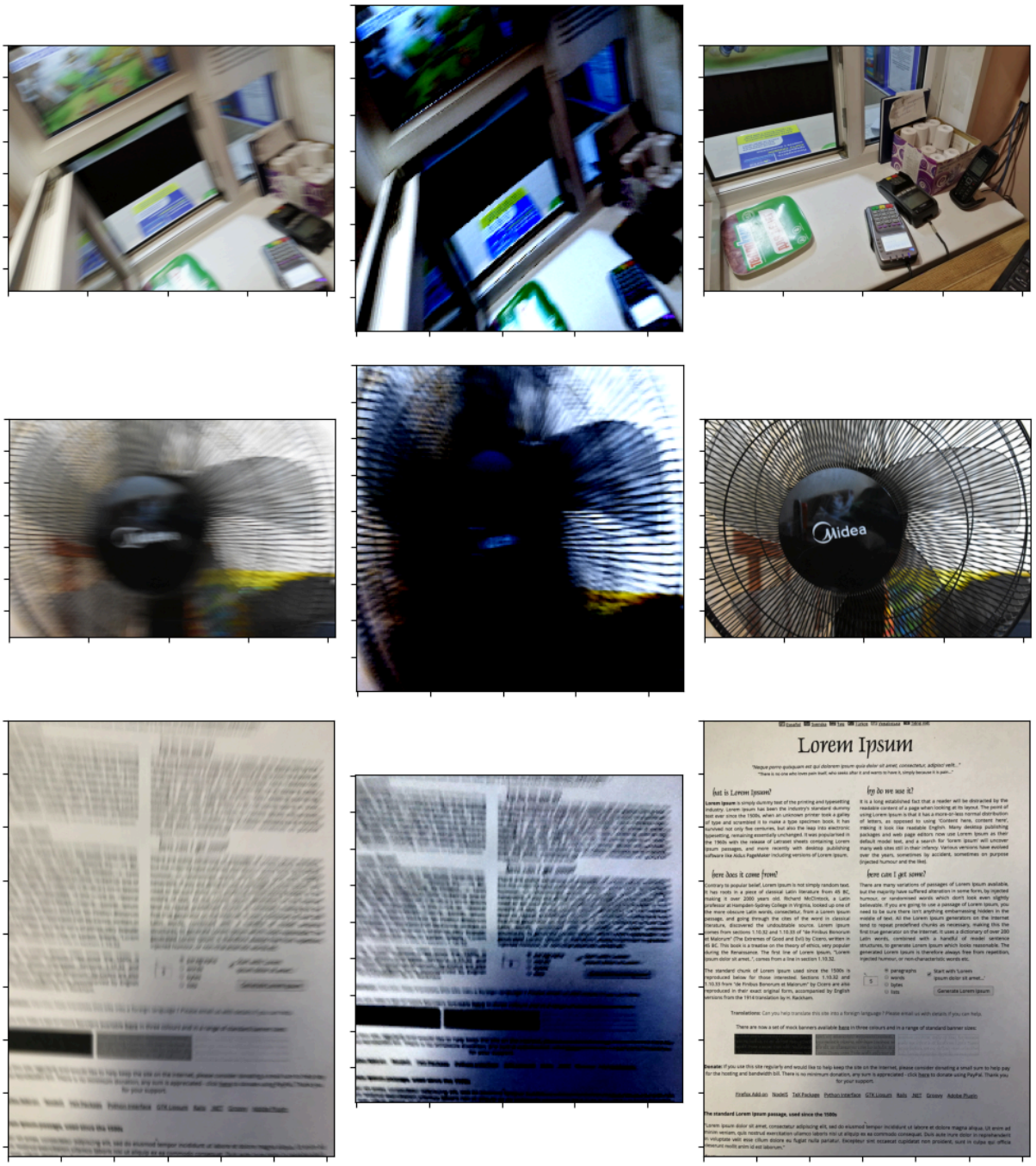



Figure 26. Visual comparisons of image deblurring using the ViTDeblur$_{SGD}$ model, same arrangement as in Figure 25.

From both sets of images, it is clear that the overall blurring defect is mitigated, and even some detailed parts appear to be sharper than they are in a blurred input. However, both ViTDeblur variations face challenges in capturing finer details, especially in areas with shadows, shaded or appear dark in a relatively dark space. Perfect example of this issue is the middle image on the second row in both Figure 25 and 26 - the shaded area is not even visible. When it comes to darker or shaded areas, the model fails to capture details due to similarity in colours as well. Areas with shadows tend to be bigger, as shown on the restored image in the last row, even though blurring is less intense.

It is worth mentioning that $\text{ViTDeblur}_{SGD}$ has better, smoother and overall slightly more pleasing results than $\text{ViTDeblur}_{adam}$. The result of the second approach handles shaded regions and dark objects slightly better than the first method.

## 5.4 Discussion

ViTDeblur model consists of pre-trained ViT and ViTDecoder. ViTDecoder is composed of upsampling and decoding blocks. ViTDeblur was trained on Adam optimiser and SGD. Both variations proved the feasibility of base ViT for image encoding and global context gain in combination with CNN-based decoders, showing their strength in capturing local dependencies. Although, due to weakness and structural simplicity of ViTDecoder, ViTDeblur is vulnerable to shaded areas and dark objects, it is clear that deblurring artefact is noticeably mitigated.

# Chapter 6: Conclusion

Image restoration is a field of study in computer vision that deals with degraded images and is important, because corrupted images may affect the performance of other image processing tasks, such as classification, object detection or localisation. One of such domestic issues that happens occasionally is the blurring during motion.

There are a number of approaches to solving such a task: classic algorithms, machine learning and deep learning techniques. It all started with Wiener's filter progressing to current methods using modern Neural Networks architectures.

Most Neural Network structures solving computer vision tasks are mostly based on CNN layers and divided into two types - multistage and single shot models. They have their own pros and cons, be it architectural complexity, advancement of training algorithms or resulting performance stats, such as memory or time requirements.

When the Transformer model designed to solve NLP tasks was introduced, researchers started to try and apply attention blocks to image processing tasks. This led to the development of ViT model. This set up an example and a baseline for other computer vision tasks and several models were developed that applied self-attention principles to restoring images.

Self-attention based models showed excellent performance and ability to learn and capture long range dependencies and global context and adapt to any input, while CNN layers have limited receptive fields size which prevents such approach from capturing global context, and since all weights are shared it is hard to adapt to various inputs.

In this paper, the ViTDeblur model was introduced. ViTDeblur uses pre-trained ViT for image encoding and ViTDecoder for image decoding. ViT demonstrated the feasibility of capturing long-term dependencies in combination with ViTDecoder's ability of learning local context using CNN. Model was trained on a small Kaggle dataset presented by Aleksey Alekseev (2019) using two approaches to training with different losses and optimisation solvers.

# References


[1] Wiener, N. (1949). Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications. The MIT press.

[2] *WIENER FILTERING*. (n.d.). Www.owlnet.rice.edu. https://www.owlnet.rice.edu/~elec539/Projects99/BACH/proj2/wiener.html

[3] Vastola, K. S., & Poor, H. V. (1983). An analysis of the effects of spectral uncertainty on wiener filtering. *Automatica*, *19*(3), 289–293. https://doi.org/10.1016/0005-1098(83)90105-X(https://www.sciencedirect.com/science/article/pii/000510988390105X)

[4] Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., Yang, M.-H., & Shao, L. (2021). Multi-Stage Progressive Image Restoration. *ArXiv:2102.02808 [Cs]*. https://arxiv.org/abs/2102.02808

[5] Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. https://arxiv.org/pdf/1505.04597.pdf

[6] Xing, X., & Zhang, D. (2019). Image Super-Resolution Using Aggregated Residual Transformation Networks With Spatial Attention. *IEEE Access*, *7*, 92572–92585. https://doi.org/10.1109/access.2019.2927238

[7] Li, Z. (2023, December 15). *Image Deblurring using GAN*. ArXiv.org. https://arxiv.org/abs/2312.09496

[8] He, K., Zhang, X., Ren, S., & Sun, J. (2015, December 10). *Deep Residual Learning for Image Recognition*. ArXiv.org. https://arxiv.org/abs/1512.03385

[9] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). *Generative Adversarial Nets*. https://arxiv.org/pdf/1406.2661.pdf

[10] Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., & Matas, J. (2018, April 3). *DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks*. ArXiv.org. https://doi.org/10.48550/arXiv.1711.07064

[11] Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution. ArXiv.org. https://arxiv.org/abs/1603.08155

[12] Arjovsky, M., Chintala, S., & Bottou, L. (2017). *Wasserstein GAN*. ArXiv.org. https://arxiv.org/abs/1701.07875

[13] Zhu, J.-Y., Park, T., Isola, P., Efros, A., & Research, B. (2017). *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. https://arxiv.org/pdf/1703.10593.pdf

[14] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved Training of Wasserstein GANs. ArXiv:1704.00028 [Cs, Stat]. https://arxiv.org/abs/1704.00028

[15] Simonyan, K., & Zisserman, A. (2015). *Published as a conference paper at ICLR 2015 VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION*. https://arxiv.org/pdf/1409.1556.pdf

[16] Journal Of L A T E X Class, & Files. (2015). *An All-in-One Network for Dehazing and Beyond*. *14*(8). https://arxiv.org/pdf/1707.06543.pdf

[17] Sajjadi, M. S. M., Schölkopf, B., & Hirsch, M. (2017, July 30). *EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis*. ArXiv.org. https://doi.org/10.48550/arXiv.1612.07919
[18] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. https://doi.org/10.1109/cvpr.2016.91
[19] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. *ArXiv:1612.03144 [Cs]*. https://arxiv.org/abs/1612.03144
[20] Orest Kupyn, Martyniuk, T., Wu, J., & Wang, Z. (2019). DeblurGAN-v2: Deblurring (Orders-of-Magnitude) Faster and Better. ArXiv (Cornell University). https://doi.org/10.48550/arxiv.1908.03826
[21] Mao, X., Li, Q., Xie, H., Lau, R., 4, Z., & Smolley, S. (2017). *Least Squares Generative Adversarial Networks*. https://arxiv.org/pdf/1611.04076.pdf
[22] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2016). *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*. https://arxiv.org/pdf/1602.07261.pdf
[23] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2019). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. https://arxiv.org/pdf/1801.04381.pdf
[24] Jolicoeur-Martineau, A. (2018, September 10). *The relativistic discriminator: a key element missing from standard GAN*. ArXiv.org. https://doi.org/10.48550/arXiv.1807.00734
[25] Isola, P., Zhu, J.-Y., Zhou, T., Efros, A., & Research, B. (2018). *Image-to-Image Translation with Conditional Adversarial Networks*. https://arxiv.org/pdf/1611.07004.pdf
[26] Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., & Shi, W. (2016). *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*. ArXiv.org. https://arxiv.org/abs/1609.04802
[27] Tao, X., Gao, H., Wang, Y., Shen, X., Wang, J., & Jia, J. (2018). Scale-recurrent Network for Deep Image Deblurring. ArXiv (Cornell University). https://doi.org/10.48550/arxiv.1802.01770
[28] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., & Wang, Z. (2016). Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. *ArXiv:1609.05158 [Cs, Stat]*. https://arxiv.org/abs/1609.05158
[29] Liu, Z., Yeh, R. A., Tang, X., Liu, Y., & Agarwala, A. (2017). Video Frame Synthesis using Deep Voxel Flow. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1702.02463
[30] Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., & Wang, O. (2016, November 25). *Deep Video Deblurring*. ArXiv.org. https://doi.org/10.48550/arXiv.1611.08387
[31] Xin, T., Gao, H., Liao, R., Wang, J., & Jia, J. (2017). Detail-revealing Deep Video Super-resolution. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1704.02738
[32] Xu, N., Price, B., Cohen, S., & Huang, T. (2017). Deep image matting. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2970-2979).

[33] Purohit, K., Shah, A., & Rajagopalan, A. N. (2019, March 9). *Bringing Alive Blurred Moments*. ArXiv.org. https://doi.org/10.48550/arXiv.1804.02913
[34] Jin, M., Meishvili, G., & Favaro, P. (2018, April 11). *Learning to Extract a Video Sequence from a Single Motion-Blurred Image*. ArXiv.org. https://doi.org/10.48550/arXiv.1804.04065
[35] Zhang, H., & Yang, J. (2015). Intra-frame deblurring by leveraging inter-frame camera motion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4036-4044)
[36] Graves, A., Mohamed, A. R., & Hinton, G. (2013, May). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 6645-6649). Ieee.
[37] SHI, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W., & WOO, W. (2015). *Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting*. Neural Information Processing Systems; Curran Associates, Inc. https://papers.nips.cc/paper_files/paper/2015/hash/07563a3fe3bbe7e3ba84431ad9d055af-Abstract.html
[38] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, Ł., & Polosukhin, I. (2017). *Attention Is All You Need*. https://arxiv.org/pdf/1706.03762.pdf
[39] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ArXiv:2010.11929 [Cs].
[40] Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., & Yang, M.-H. (2022). Restormer: Efficient Transformer for High-Resolution Image Restoration. *ArXiv:2111.09881 [Cs]*. https://arxiv.org/abs/2111.09881
[41] Hendrycks, D., & Gimpel, K. (2020). Gaussian Error Linear Units (GELUs). *ArXiv:1606.08415 [Cs]*. https://arxiv.org/abs/1606.08415
[42] Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., & Li, H. (2021). Uformer: A General U-Shaped Transformer for Image Restoration. ArXiv:2106.03106 [Cs]. https://arxiv.org/abs/2106.03106
[43] Shaw, P., Uszkoreit, J., & Vaswani, A. (2018). Self-Attention with Relative Position Representations. *ArXiv:1803.02155 [Cs]*. https://arxiv.org/abs/1803.02155
[44] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. https://arxiv.org/pdf/2103.14030.pdf
[45] Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., & Zhang, L. (2021). CvT: Introducing Convolutions to Vision Transformers. ArXiv:2103.15808 [Cs]. https://arxiv.org/abs/2103.15808
[46] Yuan, K., Guo, S., Liu, Z., Zhou, A., Yu, F., & Wu, W. (2021). Incorporating Convolution Designs into Visual Transformers. *ArXiv:2103.11816 [Cs]*. https://arxiv.org/abs/2103.11816
[47] O'shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. https://arxiv.org/pdf/1511.08458.pdf

[48] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36(4), 193–202. https://doi.org/10.1007/bf00344251
[49] Lecun, Y., L Eon Bottou, Bengio, Y., & Patrick Haaner Abstract|. (1998). Gradient-Based Learning Applied to Document Recognition. PROC. OF the IEEE. http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf
[50] Li, S., Liu, M., Zhang, Y., Chen, S., Li, H., Dou, Z., & Chen, H. (2023, December 17). SAM-Deblur: Let Segment Anything Boost Image Deblurring. ArXiv.org. https://doi.org/10.48550/arXiv.2309.02270
[51] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment Anything. ArXiv:2304.02643 [Cs]. https://arxiv.org/abs/2304.02643
[52] Chen, L., Chu, X., Zhang, X., & Sun, J. (2022, August 1). Simple Baselines for Image Restoration. ArXiv.org. https://doi.org/10.48550/arXiv.2204.04676
[53] Charbonnier, P., Blanc-Feraud, L., Aubert, G., & Barlaud, M. (1994, November). Two deterministic half-quadratic regularization algorithms for computed imaging. In Proceedings of 1st international conference on image processing (Vol. 2, pp. 168-172). IEEE.
[54] Aleksey Alekseev. (2019). *Blur dataset*. Kaggle.com. https://www.kaggle.com/datasets/kwentar/blur-dataset
[55] Nah, S., Kim, T. H., & Lee, K. M. (2016, December 7). Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring. ArXiv.org. https://arxiv.org/abs/1612.02177v2
[56] Rim, J., Lee, H., Won, J., & Cho, S. (2020). Real-world blur dataset for learning and benchmarking deblurring algorithms. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16 (pp. 184-201). Springer International Publishing.
[57] Shen, Z., Wang, W., Lu, X., Shen, J., Ling, H., Xu, T., & Shao, L. (2020, January 19). Human-Aware Motion Deblurring. ArXiv.org. https://doi.org/10.48550/arXiv.2001.06816
[58] VisibleBreadcrumbs. (2020). Mathworks.com. https://www.mathworks.com/help/vision/ref/psnr.html
[59] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing, 13(4), 600–612. https://doi.org/10.1109/tip.2003.819861
[60] Neural Transfer Using PyTorch — PyTorch Tutorials 2.3.0+cu121 documentation. (n.d.). Pytorch.org. https://pytorch.org/tutorials/advanced/neural_style_tutorial.html#style-loss
[61] StepLR — PyTorch 1.9.0 documentation. (n.d.). Pytorch.org. https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.StepLR.html
[62] Kingma, D. P., & Ba, J. (2014, December 22). Adam: A Method for Stochastic Optimization. ArXiv.org. https://arxiv.org/abs/1412.6980
[63] Shazeer, N. (2020). GLU Variants Improve Transformer. ArXiv:2002.05202 [Cs, Stat]. https://arxiv.org/abs/2002.05202
[64] Hu, J., Shen, L., Albanie, S., Sun, G., & Wu, E. (2019). Squeeze-and-Excitation Networks. ArXiv:1709.01507 [Cs]. https://arxiv.org/abs/1709.01507

[65] Robbins, H., & Monro, S. (1951). A Stochastic Approximation Method. The Annals of Mathematical Statistics, 22(3), 400–407. https://doi.org/10.1214/aoms/1177729586
[66] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. ArXiv:1409.0575 [Cs]. https://arxiv.org/abs/1409.0575v3
[67] Kaggle. (2022). Kaggle: Your Home for Data Science. Kaggle.com. https://www.kaggle.com/
[68] Fukushima, K. (1969). Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements. IEEE Transactions on Systems Science and Cybernetics, 5(4), 322–333. https://doi.org/10.1109/tssc.1969.300225
[69] Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ArXiv.org. https://arxiv.org/abs/1502.03167v3
[70] Maas, A., Hannun, A., & Ng, A. (2013). Rectifier Nonlinearities Improve Neural Network Acoustic Models. https://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf
[71] Anwar, S., Khan, S., & Barnes, N. (2020, March 23). *A Deep Journey into Super-resolution: A survey*. ArXiv.org. https://doi.org/10.48550/arXiv.1904.07523
[72] Dong, C., Loy, C. C., He, K., & Tang, X. (2015). Image Super-Resolution Using Deep Convolutional Networks. *ArXiv:1501.00092 [Cs]*. https://arxiv.org/abs/1501.00092
[73] Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. IEEE Transactions on Image Processing, 26(7), 3142–3155. https://doi.org/10.1109/tip.2017.2662206
[74] Zhang, Y., Tian, Y., Kong, Y., Zhong, B., & Fu, Y. (2018). Residual Dense Network for Image Restoration. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1812.10477

# Appendix

## Appendix A: Project Initiation Document


# School of Computing
# Final Year Engineering Project

# Project Initiation Document

**Azamat Ilyassov**

**BSc(Hons) Data Science and Analytics**

**Image DeBlurring using ViT**

1. Basic details

| | |
|---|---|
| Student name: | Azamat Ilyassov |
| Draft project title: | Image DeBlurring using ViT |
| Course and year: | Bsc(Hons) Data Science and Analytics, 3rd year |
| Project supervisor: | Alaa Mohasseb |
| Client organisation: | |
| Client contact name: | |

2. Degree suitability

During my first year I was introduced to the very basics of the Machine Learning area along with data manipulation techniques in R during a Practical Data Science course. This inspired me to move forward in this direction. Initially I started to gain more expertise on data manipulation using Pandas and NumPy packages in Python.
During my second year I started to gain even more expertise in machine learning algorithms, data mining and deep learning algorithms.
I gained a lot of interest in neural network algorithms. During Placement year after my working hours I was mostly spending time reading research papers on various topics in the AI area, such as Generative AI, Natural Language Processing, Computer Vision models.
In the end ViT(Vision Transformer) model raised my interest, after the release of SAM(Segment Anything Model). Hence, I decided that it would be beneficial for the main idea of my project.
Investigating and contributing knowledge to the methods for understanding data was in the roots of a Data Science industry since the establishment of the industry on the market and scientific community. Machine Learning or Deep Learning, in this case, is heavily relied on within the area to retrieve beneficial information from large amounts of data, especially image data.
My project's topic perfectly fits my course and ideal application of all my expertise I have attained in my modules.

3. The project environment and problem to be solved

The main goal of this project is to be able to help improve image quality, be it images with a lot of noise(unnecessary signals), blurred images or photos taken in a movement.

Hence, my research paper would raise an interest among AI researchers and enthusiasts, in the first place. Further on it also could be used in photo editing apps or restoration of old photos, old graphics or in game development industry in order to improve graphics in games.
Main question of this research project - *what is the best way to improve image quality using computer vision algorithms? Are current methods enough? Is ViT model going to make big changes in the CV area?*
There are many CV models that are able to perform this kind of task. The most common one is the Image Autoencoder model. I want to add to the current knowledge and train a model that will use the Vision Transformer model's backbone for this task.

4. Project aim and objectives

**Aim:**
The main goal of my project is to develop a model for images that are very noisy, blurred or photos taken during movement.

**Objectives:**

- Learn in-depth about Computer Vision area and how it was changed with the rise of AI
- Learn basics of Computer Vision in the context of Neural Networks(CNN structure, basic algorithms for various tasks)
- Research the topic of ViT(Vision Transformers) and derivatives of this work, models that were produced using the backbone of this model and their relation to the topic of my Research project
- Investigate SOTA(State of the Art) approach to training a model
- Research how attention based CV models would fit into image deblurring tasks

5. Project constraints

Project deadline might constrict the model's accuracy and quality.

6. Facilities and resources

- Computational power
  Strong machine with good GPU and RAM
- Library

7. Log of risks

| No | Description | Likelihood (high, medium, low) | Impact | Mitigation/Avoidance |
|---|---|---|---|---|
| 1 | Laptop failure | Medium | Delays and data loss | Back ups files, models and review |
| 2 | Driver failure | Medium | Delays and data | Cloud storage |

| | | | | |
|---|---|---|---|---|
| | | | loss | |
| 3 | Computing facilities booked | Medium | Delays | Book in advance and manage time carefully |
| 4 | Dataset not available | Medium | Delays | Search for alternatives |

8. Project deliverables

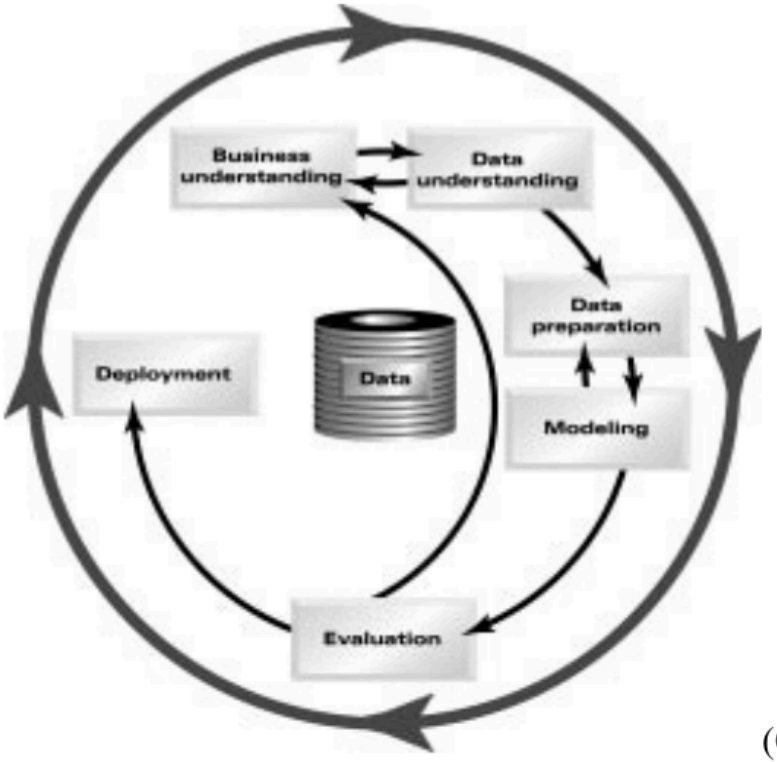
DeBlurring with ViT project is a research project that aims to create a model to deblur an image and present a report that would have a review of all the literature related to the topic.

9. Project approach

The most optimal and popular approach for a research project that is related to the Data Science, Machine Learning and Deep Learning industry is CRISP-DM.

It has 6 stages:

- Business Understanding - basic, but thorough understanding of a task
- Data Understanding - in principle it is the same as Business Understanding. You need to understand your data and how it fits for your project's constraints and environment
- Data Preparation - analysis of given data, building preprocessing pipelines, data augmentations, scaling
- Modelling - implementation of a machine/deep learning model
- Evaluation - evaluating model's performance and its accuracy
- Deployment - preparing to use models in products, for example in web app API. Not applicable in this case, since the aim of this project is to produce a model, add to knowledge, compare results and build a report




(Chapman et al, 2000)

10. Project tasks and timescales

Project would be mainly divided into several stages that would consist of sub-stages.

Project starting phase for initiation of the project. Researching chosen topic, literature review, background research. Data collection and preparation of data transformation pipelines. Modelling - implementation of a model and training. Evaluation, comparing results with previous works. Report stage.

Detailed Gantt Chart is in the appendix A

| No | Stage | Dates | Main Tasks |
|---|---|---|---|
| 1 | Project initiation | 18/09/23 - 2/10/23 | Choose topic and supervisor |
| 2 | Project initiation | 2/10/23 - 20/10/23 | PID document creation |
| 3 | Topic Research | 20/10/23 - 17/11/23 | Gathering all important literature |
| 4 | Topic Research | 30/10/23 - 22/12/23 | Background research and literature review |
| 5 | Data Collection & Preparation | 11/12/23 - 05/01/24 | Gathering dataset |
| 6 | Data Collection & Preparation | 25/12/23 - 09/02/24 | Analysis, understanding and gaining insights on data |

| | | | |
|---|---|---|---|
| 7 | Data Collection & Preparation | 08/01/24 - 09/02/24 | Building data pipeline |
| 8 | Modelling | 18/12/23 - 26/01/24 | Designing method and model architecture |
| 9 | Modelling | 15/01/24 - 22/03/24 | Model's architecture and training process implementation |
| 10 | Give presentation | 02/02/24 | |
| 11 | Modelling | 12/02/24 - 22/03/24 | Training and testing, evaluating model's performance |
| 12 | Evaluation | 11/03/24 - 12/04/24 | Evaluating model's performance, comparing results, descriptive analysis of achieved results |
| 13 | Dissertation build up | 10/11/23 - 26/04/24 | Literature Review along with report where all of the information about design and finding is found |
| 14 | Submit report | 03/05/2024 | |
| 15 | Give presentation | 17/05/2024 | |

11. Supervisor meetings

Me and my supervisor decided on online weekly meetings, 12.30pm every Tuesday. However, a meeting could be cancelled or moved to another time during the week.

12. Legal, ethical, professional, social issues

This research is purely educational and would be available for public use. All of the licences of the data used would not affect the project. Data must be carefully picked, so that it would fall under all ethical and legal norms. The deliverables of this project would not be misused. The data to be used will not contain someone else's work of art, does not have human objects or animals, so there should be no issues with legal, professional, ethical or social point of view.

13. Permission

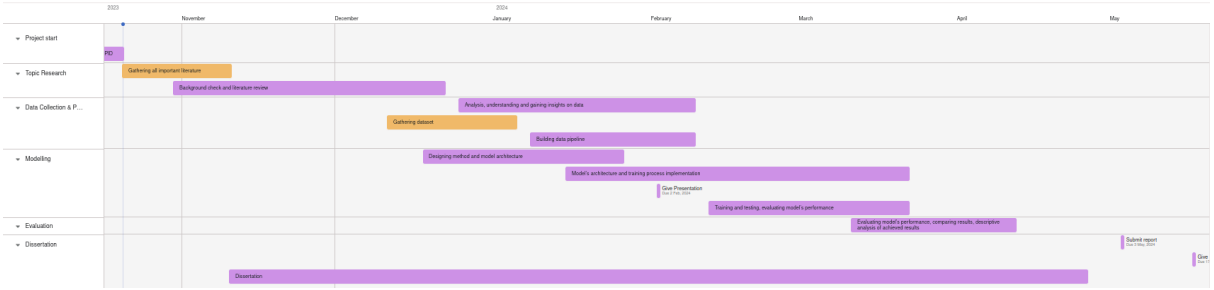
Please tick

❏ I give permission for my PID to be made available to other students as examples of previous work.

❏ I do not give permission for my PID to be made available to other students as examples of previous work.

Date: ___20/10/2023_____

# Appendix B: Gantt chart



# Appendix C: Ethics Certificate


## Certificate of Ethics Review

**Project title:** Image DeBlurring using ViT

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Name**: | Azamat Ilyassov | **User ID**: | UP957417 | **Application date**: | 05/12/2023 15:50:40 | **ER Number**: | TETHIC-2023-107044 |

You must download your referral certificate, print a copy and keep it as a record of this review.

The FEC representative(s) for the **School of Computing** is/are **Elisavet Andrikopoulou, Kirsten Smith**

It is your responsibility to follow the University Code of Practice on Ethical Standards and any Department/School or professional guidelines in the conduct of your study including relevant guidelines regarding health and safety of researchers including the following:

- University Policy
- Safety on Geological Fieldwork

It is also your responsibility to follow University guidance on Data Protection Policy:

- General guidance for all data protection issues
- University Data Protection Policy

Which school/department do you belong to?: **School of Computing**
What is your primary role at the University?: **Undergraduate Student**
What is the name of the member of staff who is responsible for supervising your project?: **Alaa Mohasseb**
Will financial inducements (other than reasonable expenses and compensation for time) be offered to participants?: No
Are there risks of significant damage to physical and/or ecological environmental features?: No
Are there risks of significant damage to features of historical or cultural heritage (e.g. impacts of study techniques, taking of samples)?: No
Does the project involve animals in any way?: No
Could the research outputs potentially be harmful to third parties?: No
Could your research/artefact be adapted and be misused?: No
Will your project or project deliverables be relevant to defence, the military, police or other security organisations and/or in addition, could it be used by others to threaten UK security?: No
Please read and confirm that you agree with the following statements: I confirm that I have considered the implications for data collection and use, taking into consideration legal requirements (UK GDPR, Data Protection Act 2018 etc.), I confirm that I have considered the impact of this work and and taken any reasonable action to mitigate potential misuse of the project outputs, I confirm that I will act ethically and honestly throughout this project

### Supervisor Review

As supervisor, I will ensure that this work will be conducted in an ethical manner in line with the University Ethics Policy.

Supervisor comments:

Supervisor's Digital Signature: **alaa.mohasseb@port.ac.uk** Date: **05/12/2023**

# Certificate of Ethics Review

**Project title:** Image DeBlurring using ViT

| **Name**: | Azamat Ilyassov | **User ID**: | UP957417 | **Application date**: | 05/12/2023 15:50:40 | **ER Number**: | TETHIC-2023-107044 |
|---|---|---|---|---|---|---|---|

You must download your referral certificate, print a copy and keep it as a record of this review.

The FEC representative(s) for the **School of Computing** is/are **Elisavet Andrikopoulou, Kirsten Smith**

It is your responsibility to follow the University Code of Practice on Ethical Standards and any Department/School or professional guidelines in the conduct of your study including relevant guidelines regarding health and safety of researchers including the following:

- University Policy
- Safety on Geological Fieldwork

It is also your responsibility to follow University guidance on Data Protection Policy:

- General guidance for all data protection issues
- University Data Protection Policy

Which school/department do you belong to?: **School of Computing**
What is your primary role at the University?: **Undergraduate Student**
What is the name of the member of staff who is responsible for supervising your project?: **Alaa Mohasseb**
Will financial inducements (other than reasonable expenses and compensation for time) be offered to participants?: No
Are there risks of significant damage to physical and/or ecological environmental features?: No
Are there risks of significant damage to features of historical or cultural heritage (e.g. impacts of study techniques, taking of samples)?: No
Does the project involve animals in any way?: No
Could the research outputs potentially be harmful to third parties?: No
Could your research/artefact be adapted and be misused?: No
Will your project or project deliverables be relevant to defence, the military, police or other security organisations and/or in addition, could it be used by others to threaten UK security?: No
Please read and confirm that you agree with the following statements: I confirm that I have considered the implications for data collection and use, taking into consideration legal requirements (UK GDPR, Data Protection Act 2018 etc.), I confirm that I have considered the impact of this work and and taken any reasonable action to mitigate potential misuse of the project outputs, I confirm that I will act ethically and honestly throughout this project

## Supervisor Review

As supervisor, I will ensure that this work will be conducted in an ethical manner in line with the University Ethics Policy.

Supervisor comments:

Supervisor's Digital Signature**:** **alaa.mohasseb@port.ac.uk** Date**: 05/12/2023**