UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Sequence Modelling with Deep Learning for Visual Content Generation and Understanding

by Zongxin Yang

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Prof. Yi Yang

University of Technology Sydney
Faculty of Engineering and Information Technology

Jul 2021

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Zongxin Yang* declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:

Signature: Signature removed prior to publication.

Date: 16th Jul, 2021

Abstract

Although convolutional neural networks have proven to be effective and stable in image feature learning, sequence modelling is still critical for learning spatial and temporal context information. In an image scenario, different semantic structures can be regarded as a sequence arranged along the horizontal (or vertical) direction. Moreover, in a video scenario, temporal sequence modelling is necessary for understanding inter-frame relationships, such as object movement and occlusion. This thesis explores more effective spatial or temporal sequence modelling for image or video scenario understanding. For the former, an encoder-decoder framework is proposed to split an input scenario into a sequence of spatial features and reconstruct the input. By modelling spatial sequence information, the framework can even predict new scenes with very large scales in length while keeping a consistent style regarding the given input. For video understanding, the thesis processes temporal sequences in a recurrent manner (*i.e.*, frame by frame), which is more memory-efficient. In addition, the thesis proposes to implicitly impose the feature embedding of each target and relative background to be contrastive throughout the temporal sequence, promoting the results of downstream tasks accordingly. Besides, a novel transformation module is designed to model channel relationships for improving intra-frame representation ability. To validate proposed approaches and components, extensive experiments are conducted on image outpainting, instance segmentation, object detection, classification, video classification, and video object segmentation.

ACKNOWLEDGMENTS

Firstly, I would like to thank my supervisor Professor Yi Yang, without whom I would not have been able to complete this thesis. I sincerely appreciate him for his selfless support and patience. He guided me throughout the research by providing tremendous help on his knowledge and insight into the related fields.

I would also like to thank my colleagues at University of Technology Sydney. I would like to thank Yunchao Wei, Linchao Zhu, Xin Yu, Ping Liu, Yu Wu, Qianyu Feng, Peike Li, Chen Liang, Jiaxu Miao, Yuhang Ding, and many others. I was really fortunate to work with them and participate in intellectual conversations with them.

Lastly, I would like to thanks my mother, Gexiu Liao, and my father, Jianlin Yang, for their support and love throughout the years.

LIST OF PUBLICATIONS

1. **Z. YANG**, J. DONG, P. LIU, Y. YANG, AND S. YAN, *Very long natural scenery imageprediction by outpainting*, in ICCV, 2019, pp. 10561-10570.

2. **Z. YANG**, L. ZHU, Y. WU,AND Y. YANG, *Gated channel transformation for visual-recognition*, in CVPR, 2020, pp. 11794-11803.

3. **Z. YANG**, Y. WEI, AND Y. YANG, *Collaborative video object segmentation by foreground-background integration*, in ECCV, Springer, 2020, pp. 332-348.

Table of Contents

List of Figures

Figure **Page**

LIST OF TABLES

TABLE **Page**

CHAPTER 1

INTRODUCTION

1.1 Visual Feature Learning

For visual feature learning, there are two types of basic data, image and video. For the former, it has been demonstrated that Convolutional Neural Networks (CNNs) are effective in fundamental computer vision tasks, such as segmentation [61], classification [35], and detection [61]. A single convolutional layer computes with a small region of neighbouring pixels for each spatial position, which suffers from the problem of local ambiguities [67]. To relieve such a problem, it is proposed to construct deep CNNs and improve the receptive fields by VGGNets [60], using multiple stages of convolutional layers, non-linear layers, and pooling operators. Moreover, a residual connection is introduced by ResNets [29] for building deeper yet stable convolutional architectures. Although deeper CNNs help to enlarge the receptive fields of networks, the learned spatial dependencies are still restricted in a fixed range [69].

Extending 2D CNNs to 3D ones, C3D [40, 68] and many following methods (*e.g.*, [74]) has proven to be effective for modeling spatial-temporal features in video recognition tasks, such as action classification [42]. However, a large amount of computational resource is required to run these methods, and thus most of the researchers only afford to model temporal dependencies in short video sequences, with 16 frames in usual [74]. Besides, all the input video frames must be fed to 3D CNNs simultaneously and in parallel, rather than frame by frame. The latter manner puts less pressure on memory usage for inference.

Compared to CNN-based methods, sequence modelling based methods (*e.g.*, [14, 32]) show a better talent to learn long-term dependencies in Natural Language Processing (NLP) tasks. Moreover, Recurrent Neural Networks (RNNs) [5] process sequential information in a node by node manner instead of in parallel. This one-by-one manner allows the network to receive input data online, and changing the length of the input data will not affect memory usage. In recent years, more and more sequence modelling based methods are introduced to replace or cooperate with CNN-based methods in visual tasks, such as lip reading [16], image captioning [80], localization [39], and image understanding [7].

1.2 Sequence Modeling for Image Generation

Image generation is one of the most fundamental visual tasks and is closely related to sequence modelling. Intuitively, a person asked to paint, draw or otherwise recreate a visual scenario will naturally do so in a sequential, iterative procedure [25]. Following this concept, RNNs have been applied and shown to excel in handwriting generation [24]. DRAW [25] mimicked the foveation of the human eye to construct complex images iteratively, with an RNN-based auto-encoding framework. Furthermore, PixelRNN [69] proposed to sequentially predict the pixels in a scene in the two spatial directions. However, the training and inference of PixelRNN are impractical and inefficient to cover all the pixels when handling high-resolution images.

Differently, CNN-based methods [38, 50, 85] aimed to generate entire scenes at once and employs downsample-upsample pipelines to improve computational efficiency. Compared to RNN-based methods, CNN-based methods can only learn contextual information in fixed extents due to limited receptive fields. Thus they are difficult to make long-range predictions while keeping consistency in each image scene.

In Chapter 3, an effective end-to-end framework is proposed to model long-range dependencies for image generation. Given an input image scene, the method generates unseen contents outside the given image border. The predicted scene should keep semantically harmonious regrading the given scene and can be much larger than the input scene. This setting is also known as image outpainting or image extrapolation. In contrast to popular CNN-based image inpainting [85], outpainting is rarely studied before. There are two major challenges for outpainting: **(1)** The inconsistency between generated and original regions. **(2)** The difficulty in making predictions in high qualities when spatially far away from the input. An encoder-decoder framework is proposed to solve the above two problems, combining recurrent sequence modelling module (LSTM [32]) with CNN architectures. The designed framework can generate highly realistic images with very large scales in length while keeping a consistent style regarding the given input. More than that, a new natural scenery dataset is collected and proposed to evaluate our method's effectiveness. The dataset contains about $6,000$ complicated and diverse natural scenes, including starry sky, riverbank, seaside, valley, snow mountain, etc. Sufficient experiments and ablation studies on this dataset demonstrate the efficiency and effectiveness of the framework.

1.3 Sequence Modeling for Video Understanding

A lot of attention has been attracted by 3D-CNN based methods [22, 40] in video recognition tasks in recent years. However, these methods are usually applicable in modelling temporal information in short videos [74]. Such a characteristic limits the application of 3D-CNN in many Video Semantic Segmentation [92], Video Object Tracking (VOT) [4], and Video Object Segmentation (VOS) [6]. To efficiently process long-term video sequence, many methods for these tasks employs a recurrent and iterative manner, *i.e.*, processing the given video frame by frame.

Chapter 4 focuses on exploring a more efficient and effective temporal sequence modelling framework for the challenging semi-supervised VOS. Given a video sequence and desired objects' masks of the reference frame, semi-supervised VOS is responsible for segmenting objects in a pixel-wise manner across the entire video sequence.

Many VOS works [48, 70], proposed for semi-supervised VOS and based on temporal sequence modelling, have shown promising performance. However, few VOS works focus on modelling the background region's temporal sequence information and only pay attention to finding robust matching mechanisms based on the foreground target (s). The foreground region is intuitively easy to be recognised from a video when accurately removing the background. More than that, video scenarios usually contains a group of similar-looking instances, such as humans, animals, and cars. Under these cases, predictions of one foreground object are easy to be confused by a similar-looking instance in the background.

Unlike previous works that only focus on the foreground object's temporal sequence modelling (s), we argue that the background dependencies are equally important. In Chapter 4, we propose a Collaborative Foreground-Background Integration (CFBI) approach for semi-supervised VOS. In CFBI, the learned feature embeddings from the foreground target and the background region are implicitly compelled to be contrastive, thus promoting the network's representative ability and improving the VOS accuracy. To handle various object scales, both pixel-level and instance-level information types are applied for matching targets between the given reference and the video frames required to predict. Besides, a novel transformation module is designed to model channel relationships in a single frame to improve representation ability. A series of experiments are designed on three popular benchmarks, and CFBI exceeds all the state-of-the-art VOS approaches.

1.4 Contributions

This thesis is organised as follows. After this chapter, Chapter 2 presents the literature review on visual feature learning and covers related recent studies on image generation and video understanding, including image inpainting, image outpainting, video classification, and video object segmentation. In Chapter 3, an encoder-decoder framework by introducing horizontal sequence modelling is proposed to predict very long natural scenery images. We collect a diverse natural scenery dataset and evaluate the model with it on image outpainting. Chapter 4 proposes a video understanding framework for iteratively segmenting the given object from the video frame by frame. The model is evaluated on video object segmentation. Chapter 5 briefly summarises the thesis and show future directions for improvements.

In this thesis, I make the following contributions:

(1) An efficient encoder-decoder network is designed by combining sequence modelling with CNN architectures for image outpainting. Before the proposed framework, there are rare deep learning based methods for image outpainting.

(2) I collect a new outpainting dataset, in which $6,000$ complicated natural scenarios are collected. We validate the effectiveness of the outpainting framework on the proposed dataset.

(3) I consider video background to be equally treated as foreground and propose a video understanding framework for semi-supervised VOS. The proposed methods outperform all the previous strongest approaches on the three most popular benchmarks while maintains an efficient run-time.

(4) I design a generally applicable transformation module for visual understanding. In this unit, explainable and trainable variables are proposed to model global channel relationships in CNNs.

CHAPTER 2

LITERATURE REVIEW

2.1 CNN-based Visual Feature Learning

2.1.1 Image Feature Learning

VGGNets [60] and Inception networks [65] has proved that it is critical for improving the representative ability of image networks by constructing deep convolutional networks. A residual connection was introduced by ResNets [29] to help build convolutional structures with larger depth. The residual connection concept has been demonstrated to be robust and stable in many following methods [30, 79].

Many other works paid attention to another manner, *i.e.*, increasing the diversity of operator composition, for improving the representative ability of the basic convolutional elements (or blocks) within CNNs. InceptionV3 [66] proposed to construct computational elements with multi-branch pooling or convolutional layers. ResNeXt [79] proved that grouped convolutional layers are practical and efficient in improving the cardinality of block transformations. Even though the efforts mentioned above have made significant progress in image feature learning, the learned spatial dependencies are still restricted in a fixed range due to the limited size of convolutional receptive fields [69].

2.1.2 Video Feature Learning

After 2D CNNs achieved much success in image tasks, 3D convolutions [40, 68] (C3D) were introduced for video classification task. Among the methods following C3D, Non-local neural networks [74] (NLNets) designed a non-local operation for modelling global dependencies, which are difficult to capture by convolutional layers. The non-local operation significantly improved the accuracy of video classification [42]. However, a large amount of computational resource is required to run these methods and thus only afford to model temporal dependencies in short video sequences, with 16 frames in usual [74]. In addition, all frames of the input video must be fed to 3D CNNs simultaneously and in parallel, rather than frame by frame. The latter puts less pressure on memory usage.

2.2 Image Generation

2.2.1 RNN-based Image-to-Image Translation

Many approaches based on RNNs have been proposed to learn spatial dependencies for image generation. Graves et al. [24] proved that RNNs are promising in handwriting generation. DRAW [25] mimics the foveation of the human eye to construct complex images with an RNN-based auto-encoding framework iteratively. Furthermore, PixelRNN [69] present an RNN that iteratively generates the pixels in a scene in the two spatial directions. However, the training and inference of PixelRNN are impractical and inefficient to cover all the pixels when handling high-resolution images.

In comparison, we combine both convolutional and recurrent structures into an end-to-end framework in Chapter 3. The input or output sequences of the RNN module are downsampled or upsampled by our encoder-decoder convolutional network. Thus the length of the sequence is significantly reduced.

2.2.2 Generative Adversarial Networks

It has been demonstrated that Generative Adversarial Networks (GANs) [23] is successful in various visual problems, including visual content generation [18, 54], style transfer [91], and image completion [50]. GAN proposed a training pipeline consisting of a discriminator and a generator. The generator was compelled by the discriminator to captures the training data attributes by using an adversarial loss. Diverse variants of GANs have arisen to stabilise the training of GAN. Among them, commonly used WGAN-GP [26] introduced a gradient penalty process and achieved much better training stability. Hence, we utilise WGAN-GP in this thesis for our image outpainting approach.

2.2.3 Image Inpainting

Many preliminary published works for image inpainting [3, 49] employed non-semantic approaches to match local patches and fill the missing hole. Benefit from the development of CNNs, recent methods [36, 50, 82, 85] outperform the traditional methods, not surprisingly. As the scale of the missing hole or patch grows larger, however, the predicted results' performance degrades.

2.2.4 Image Outpainting

Image outpainting or image extrapolation is more challenging than image inpainting due to larger missing regions and less contextual information. There have been many classical methods [44, 62, 87] for image extrapolation or image outpainting. These methods were based on traditional processing methods instead of based on deep learning. Many of the traditional approaches attempted to “search" similar scene(s) from the given candidate(s) and then spatially stitch the matched scene patch(es) with the input scene. These traditional methods have many limitations. First, Constructing handcrafted features is elaborate and complicated. Second, many post-processes, such as local warping [73], are necessary for guaranteeing smooth patch stitching. Third, the searching space of the candidates significantly affects the outpainting results. The prediction will fail if there is no suitable patch in candidates. The framework proposed in Chapter 3 is the first method based on deep learning for solving image outpainting problem to the best of our knowledge.

2.2.5 Image-to-Image Translation

In recent works [23, 38, 58, 91] for image-to-image translation, Deep CNNs have been widely applied. “Pix2Pix" [38] proposed to model pixel-to-pixel mapping by using conditional generative adversarial network [23]. Many methods applied the concept of “Pix2Pix" to related tasks, such as sketch-to-photograph transfer [58] and many other style transfer tasks [20, 91]. There is a notable difference between image-to-image translation and image outpainting. In the former, the input scene and the output one share a similar semantic layout and consistent spatial position. However, in image outpainting, the output scene is larger than the input one. Many pixels, outside the input region and in the output scene, have no corresponding pixels in the input scene.

2.3 Video Understanding

2.3.1 Video Classification

Apart from the above CNN-based methods for video feature learning, many methods investigated temporal sequence modelling in videos for video classification. Long Short-Term Memory [32], which has good stability in long-term sequence modelling, was employed by Ng et al. [86] and Donahue et al. [19]. A Convolutional Gated Recurrent Unit is proposed by Ballas et al. [2] to utilise information from multiple spatial scales of the feature. Srivastava et al. [63] extended the sequence to sequence framework [64] to learn features from consecutive frames and proposed a composite recurrent autoencoder for unsupervised video learning.

2.3.2 Semi-supervised Video Object Segmentation

A lot of frameworks based on temporal sequence modelling have been proposed to process long-term video sequences efficiently iteratively in the field of semi-supervised VOS. Among them, some methods [6, 77] proposed to fine-tune the VOS framework on the given reference frame, which are attached with the ground-truth object mask in semi-supervised VOS. OnAVOS [71] introduced an online fine-tuning approach, *i.e.*, fine-tuning on the reference frame during the inference process. MaskTrack [51] tracked objects by using optical flow to propagate the mask of objects from one frame to the following one. For improving the performance, PReMVOS [46] combined four different deep convolutional networks into a framework, but quite a lot of fine-tuning processes are used for training and inference. Even though the efforts mentioned above have achieved promising results, the fine-tuning at the inference stage heavily slows down these methods' run-time.

To avoid using inefficient online fine-tuning, recent methods (*e.g.*, [12, 83]) focused on improving the effectiveness of temporal sequence modelling. A concept of instance-level embedding was introduced by OSMN [83] for efficiently predicting the object segmentation. PML [11] leveraged a nearest-neighbour classifier to learn pixel-wise embeddings for objects. A pixel-level matching mechanism was proposed by VideoMatch [34] to map the pixel attributes from the reference frame to the current object frame. In addition, FEELVOS [70] proposed a glocal and local matching pipeline for efficient multi-object video segmentation. The local matching is responsible for matching the object's pixels between the previous frame and the current one. STMVOS deployed a memory bank for reading and storing the object attributes from the frames in history. However, extensive simulated data generated from various datasets are necessary for training STMVOS. Without such an elaborate training procedure, the performance of STMVOS drops a lot.

In Chapter 4, both the instance-level and pixel-level information types are utilised to model temporal sequences for VOS. Moreover, a collaborative foreground-background integration approach is proposed to learn more contrastive embedding of the background region additionally. The above-mentioned methods ignore important background matching.

Chapter 3

Natural Scenery Image Generation by Spatial Sequence Modeling

3.1 Introduction

As illustrated in Fig. 3.1, the objective of image outpainting is to predict unseen contents outside the border of a given scene. The predicted scene has to be harmonious and consistent regarding the given scene on both semantic content and spatial configuration. The utilisation of image outpainting is promising in various applications. However, it is still hindered for realistic image outpainting due to this problem's difficulties.

Compared to image inpainting [3, 49], which has to completes missing regions inside the given images, two major difficulties exist in image outpainting. First, it is not easy to smoothly connect the semantic information and spatial layout between the predicted contents and the given image. [73] proposed to warp local textures to generate smooth transition around the boundaries between predictions and inputs. Second, it is hard to make the predicted contents in the distance be harmonious with the given scene because little contextual information can be used to infer the unseen region.

To solve the image outpainting problem, Some preliminary works [44, 62, 73, 87] have been proposed. These methods were based on traditional processing methods instead of based on deep learning. Many of the traditional methods attempted to “search" similar scene(s) from the given candidate(s) and then spatially stitch the matched patch(es) with

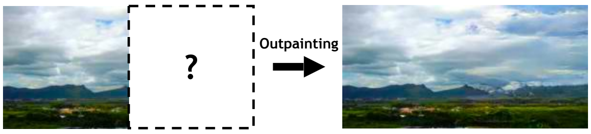


Figure 3.1: An illustration of one-step image outpainting, which predicts unseen contents outside the input region with the identical scale of the input. The semantic information and spatial layout must keep harmonious and consistent between predicted contents and the input scenario.

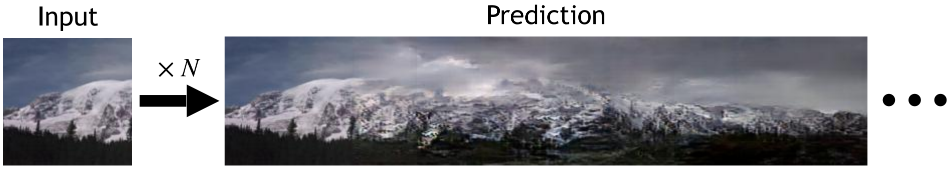


Figure 3.2: An illustration of multi-step image outpainting in horizontal direction for natural scenarios.

the input scene. These traditional methods have many limitations. First, Constructing handcrafted features is elaborate and complicated. Second, many post-processes, such as local warping [73], are necessary for guaranteeing smooth patch stitching. Third, the searching space of the candidates significantly affects the outpainting results. The prediction will fail if there is no suitable patch in candidates.

It has proved that deep encoder-decoder networks are robust in image inpainting [50]. In these networks, the encoder parts are responsible for extracting a high-level convolutional feature from an input image, while the decoder parts will generate and predict a completed image by modelling contextual dependencies. Following the same strategy, we construct an encoder-decoder pipeline for image outpainting. Moreover, some innovative improvements are made in the pipeline to handle the two critical problems mentioned above.

A person asked to draw or paint an image outpainting will naturally and sequentially do so from the image boundary to the faraway distance. Following the same concept, **Recurrent Contextual Transfer (RCT)** is proposed to model spatial sequence information in a single direction (horizontal direction in our default setting). RCT works in a sequence-to-sequence manner, *i.e.*, the spatial sequence of the input image will be transferred to a new spatial sequence for predicted contents. We use RCT to connect the network encoder and decoder. In other words, the spatial sequence length is firstly downsampled by the convolutional encoder. Thus, RCT is much more computationally efficient than PixelRNN [69], which processed spatial sequences at a pixel level. The benefit from the sequence modelling of RCT is that our network can maintain the faraway predictions with the input scene.

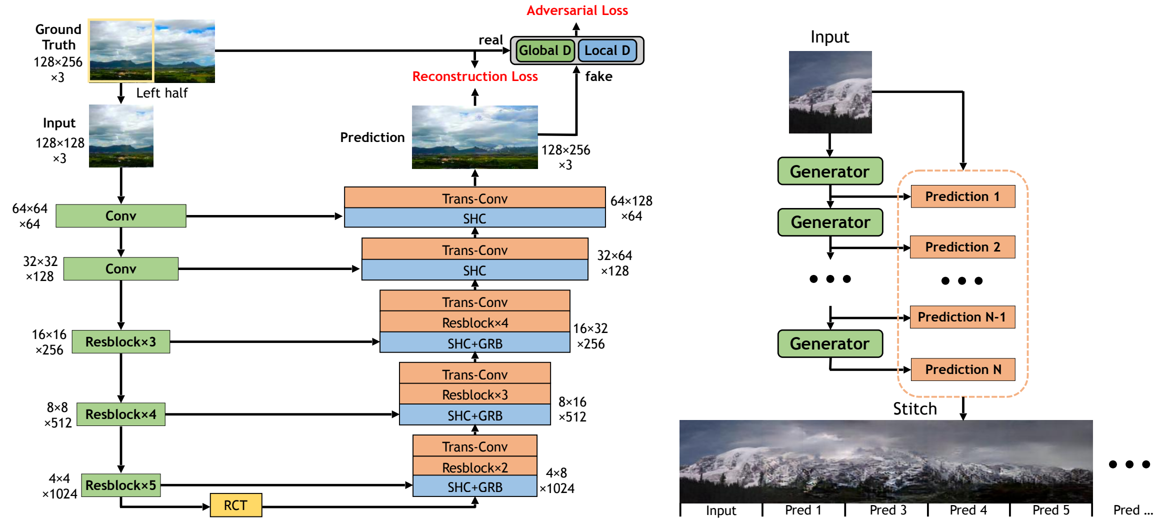
Apart from the spatial sequence modelling, we propose **Skip Horizontal Connection (SHC)** to further fuse the encoder's information with decoder's around the boundary of the given input. SHC constructs horizontal connections at multiple spatial levels between the encoder and decoder. By doing this, the decoder can learn strong spatial dependencies around the input boundary, promoting the smoothness and quality of the predicted contents accordingly.

Combining the advantages of the above *RCT* and *SHC* modules, our outpainting framework is able to predict unseen contents with *extra length* beyond the input boundary. Fig. 3.2 shows a recurrent outpainting approach based on our method. Specifically, the last step's prediction will be used as the input for the next step iteratively. This approach can efficiently predict realistic and smooth predictions with a very large scale in length. Our method is capable of generating content with high qualities, even if the contents are placed far from the given scene, where rare contextual dependencies can be used.

A new dataset with natural scenarios is collected and proposed to evaluate our method's effectiveness. The dataset contains about $6,000$ complicated and diverse natural scenes, including starry sky, riverbank, seaside, valley, snow mountain, etc. Extensive experiments are designed on the natural scenery dataset, and the proposed framework outperforms all competitors [36, 38, 85], not surprisingly.

3.2 The Proposed Approach

Fig 3.3(a) shows an overview of our outpainting framework, which follows the pipeline of GAN [23], *i.e.*, a discriminator and a generator. We follow the commonly-used encoder-decoder framework to build the generator, where Skip Horizontal Connection (SHC) is applied to fuse the encoder's information into the decoder's on multiple spatial scales. Moreover, Global Residual Blocks (GRB) is designed to create a large receptive field to learn more local contextual information. Besides, Recurrent Contextual Transfer is proposed to model spatial sequence information and connect the encoder and decoder.



(a) Overview

(b) Multi-Step Prediction

Figure 3.3: (a) An overview of our outpainting framework, which contains a discriminator and a generator. (b) An illustration of multi-step image outpainting based on our framework. In this iterative manner, we can efficiently predict unseen contents with a very large scale in length.

3.2.1 Encoder-Decoder Generator

In this section, an encoder-decoder generator is designed for image outpainting. Given an input image, the encoder part is responsible for extracting its representative feature and then split the feature into a spatial sequence. The proposed RCT module will transfer the sequence feature from the input region to the predicted unseen region. The decoder part takes the transferred spatial sequence to generate an unseen prediction with the same scale, style, and consistent scene regarding the generator input.

Encoder. We follow the structure of commonly-used *ResNet-50* to build our encoder. In detail, the layers of *ResNet-50* after *conv4_5* are removed. Moreover, we use a convolutional layer (stride= 2) to replace the *max pooling* layer.

Convolutional layers are difficult to propagate global context information from the input region to another predicted region since the convolutional receptive fields' scales are limited [50]. For solving this problem, Context-Encoder [50] employed *fully-connected* layers to build pixel-to-pixel correspondence between the input region and the predicted region. However, the use of *fully-connected* layers restricted Context-Encoder that they can only deal with features with fixed dimensions. Such a restriction decreases the prediction qualities when predictions are required to be large-scale, which is shown in Fig. 3.7(c). Furthermore, *fully-connected* layers utilise a mass of parameters, and

layer	output size	parameters
Conv	64×64×64	4×4, stride=2
Conv	32×32×128	4×4, stride=2
Res-Block×3	16×16×256	first block's stride=2
Res-Block×4	8×8×512	first block's stride=2
Res-Block×5	4×4×1024	first block's stride=2
RCT	4×4×1024	None
GRB+SHC	4×8×1024	dilation=1
Res-Block×2	4×8×1024	None
Dilated-Conv	8×16×512	4×4, stride=2
GRB+SHC	8×16×512	dilation=2
Res-Block×3	8×16×512	None
Dilated-Conv	16×32×256	4×4, stride=2
GRB+SHC	16×32×256	dilation=4
Res-Block×4	16×32×256	None
Dilated-Conv	32×64×128	4×4, stride=2
SHC	32×64×128	None
Dilated-Conv	64×128×64	4×4, stride=2
SHC	64×128×64	None
Dilated-Conv	128×256×3	4×4, stride=2

Table 3.1: The hyper-parameter setting of our encoder-decoder generator.

thus the training process will become inefficient or impractical when handling high-resolution images. Instead of using *fully-connected* layers, we propose a Recurrent Contextual Transfer (RCT) module for modelling spatial dependencies to avoid the above shortcomings.

Recurrent Contextual Transfer. Fig. 3.4 shows the detailed structure of our RCT module, which is responsible for modelling spatial sequence dependencies and transferring the features from the encoder output to the decoder input in a spatially iterative approach. Specifically, we first split the feature maps of the given input into a sequence along a spatial dimension (horizontal dimension in our setting). The input channel dimensions are reduced by one convolutional layer. After that, RCT uses two layers of *LSTM* [32] to iteratively transfer the spatial sequence from the input region to the predicted region. The predicted sequence will be stitched back into a convolutional feature, followed by one convolutional layer for increasing the channel dimensions. Thus, the input feature maps and the output ones of RCT will share the same channel dimension.

The benefit of the recurrent structure in RCT is that we can control the prediction region's scale by adjusting the predicted spatial sequence's length. More than that, we

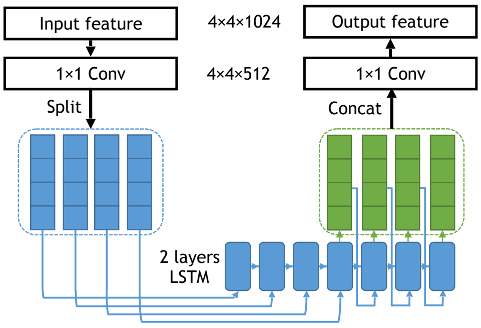
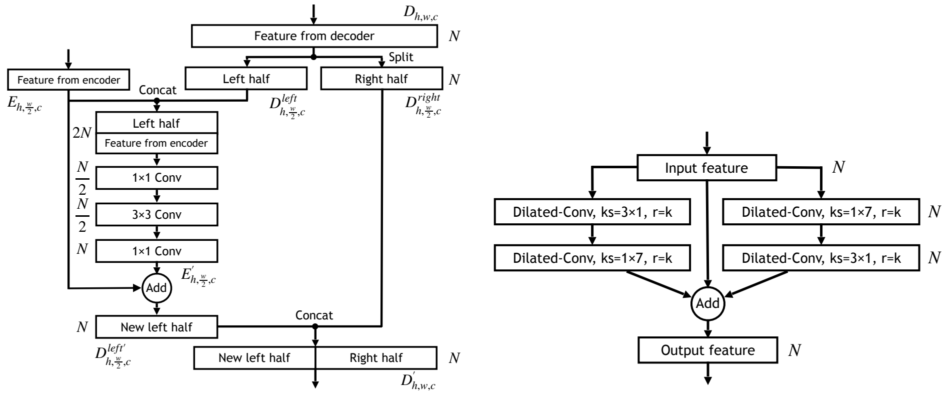


Figure 3.4: An illustration of Recurrent Contextual Transfer (RCT). The input feature maps are split into a sequence along a spatial dimension (horizontal dimension in our setting). Two layers of LSTM [32] are used to iteratively transfer the spatial sequence from the input region to the predicted region.

can efficiently predict scenes with high-quality and very large scale in length by iterating the generator, as shown in Fig. 3.10, 3.11.

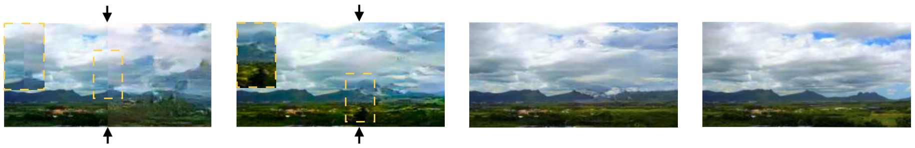


(a) SHC

(b) GRB

Figure 3.5: (a) The structure of Skip Horizontal Connection (SHC). (b) The details of Global Residual Block (GRB). N, and r, and ks denote channel number, dilation rate, and kernel size of convolutional layers separately.

Decoder. The decoder is responsible for decoding the predicted feature maps from RCT and outpainting the given image to an entire image containing an unseen and adjacent region. Without loss of generality, we will predict the right half of an image scene in our default setting, given the left half. To construct the decoder network, we use 5 *transposed-convolutional* layers, which is prevalent in image generation to upsample convolutional features and decrease the number of neurons accordingly. Different from previous works (*e.g.*, [50]), **Skip Horizontal Connection** is proposed to fuse the encoder's information with decoder's further, around the boundary of the given input.



(a) Without SHC (b) Single SHC (c) Full SHC (Ours) (d) Ground-truth

Figure 3.6: (a): Removing all the SHC connections, the predicted results have a clear faultage on the junction line between the given scene and generated scene. (b) Using only a single SHC connection in the decoder's second stage, the junction boundary becomes smoother and move a bit to the predicted region. (c) Applying all the SHC connections as our default setting, the predicted results have an excellently smooth junction line.

Skip Horizontal Connection. U-Net [55] introduced a framework, which concatenates the encoder features with the decoder features at each spatial level. Thus, the encoder information would be easier to assess during the decoding process. Following the same strategy, we propose Skip Horizontal Connection (SHC) to build more connections in our encoder-decoder generator. Unlike U-Net, our SHC does concatenation on a spatial dimension instead of the channel dimension since the decoder's desired spatial region is not consistent with the encoder's one. In our setting, the decoder features have a spatial region outside the input region, which is different from the encoder features.

Fig. 3.5(a) illustrates the detailed structure of SHC. Let $D_{h,w,c}$ and $E_{h,\frac{w}{2},c}$ denote a feature from decoder and a feature from encoder, respectively, a output feature $D^{'}_{h,w,c}$ is computed by SHC. Specifically, we first concatenate $E_{h,\frac{w}{2},c}$ with $D^{left}_{h,\frac{w}{2},c}$, the left half of $D_{h,w,c}$ on the dimension of channels. Second, 3 convolutional layers, which constitute a bottleneck structure like bottleneck Res-Block [29], are used to merge the information in the concatenated feature into a new feature, $E^{'}_{h,\frac{w}{2},c}$. After that, a element-wise residual connection is applied between $E_{h,\frac{w}{2},c}$ and $E^{'}_{h,\frac{w}{2},c}$, and generates $D^{left'}_{h,\frac{w}{2},c}$. Finally, we use $D^{left'}_{h,\frac{w}{2},c}$ to replace the left part of the decoder feature, and thus the encoder information is introduced.

Apart from SHC, *Global Residual Block* is proposed to make large receptive fields and learn a wide range of contextual information. Following ResNets [29], we employ a residual structure, which performs stable during the training process. For reducing computation cost, the normal n×n convolutional layer is replaced with a combination of two convolutional layers with the kernel size of n×1 and 1×n. Furthermore, *dilated-convolutional* layers [84] are used to enlarge the receptive field while keeping coverage and resolution. In our outpainting setting, the horizontal scale is larger and more important than the vertical one. Thus, our GRB modules use a larger kernel size in the

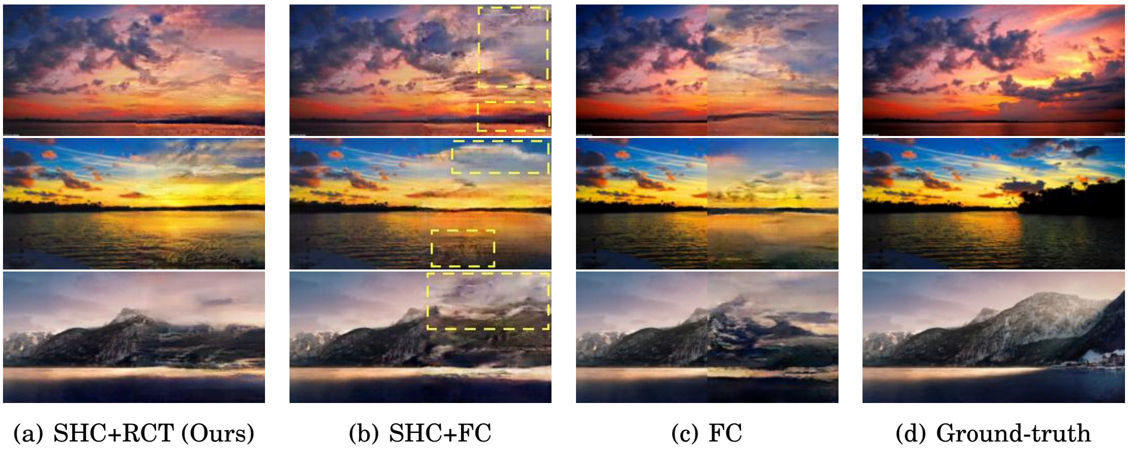


Figure 3.7: Comparison results on the proposed dataset. (d) The ground-truth image scenes. (c) The predictions have clear faultage on the junction line between the given scene and generated scene using a fully-connected layer to transfer information from the encoder to the decoder. (b) Applying SHC layers can mitigate the problem of faultage. However, the predicted contents are still blurred where the prediction position is far from the junction line, as highlighted in yellow boxes. (a) We improve the prediction quality in faraway regions by replacing the fully-connected layer with our proposed RCT, which has a better performance in modelling spatial sequence.

horizontal direction.

3.2.2 Loss Function

Two types of loss functions, *i.e.*, a generative adversarial loss and a masked reconstruction loss, are used in our framework. The latter focuses on the low-level attributes of image scenes and is employed to capture the logical correlations and the overall layout of the given input scene's predicted contents. Differently, the adversarial loss [1, 23, 26] is responsible for improving the quality of scene details, which belong to high-level attributes.

Masked Reconstruction Loss, $\mathscr{L}_{rec}(x)$, is based on a MSE loss applied between the entire generated scene containing the input region $\tilde{x}$ and the ground-truth scene x. The MSE loss can make compel the generator to generate a rough and blurry layout of the predicted scene with regrads to the ground-truth [50]. In formula,

$$\mathscr{L}_{rec}(x) = M \odot \parallel x - \tilde{x} \parallel_2^2, \tag{3.1}$$

where a mask weight, M, is applied to progressive decrease the weight of MSE along the horizontal direction towards the predicted region. The mask applied in loss function

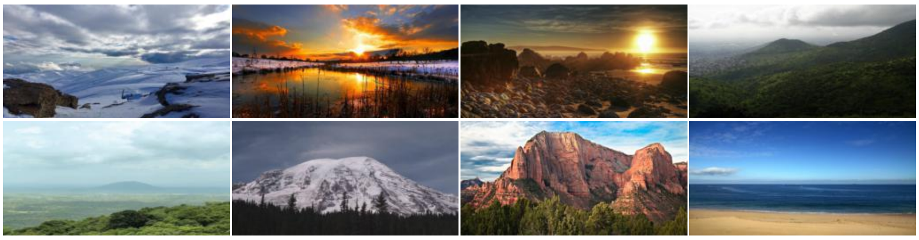


Figure 3.8: Some scenes sampled from the natural scenery dataset. The dataset contains about 6,000 complicated and diverse natural scenes, including starry sky, riverbank, seaside, valley, snow mountain, etc.

has been commonly used in many image generation methods [36, 50, 85]. Intuitively, the prediction should have little correlation with the ground-truth scene where the region is far away from the input boundary. A *cos* function is used to decay M to 0, which is different from the mask function used in previous works. Let d denote the horizontal distance between a position in the predicted region and the input boundary, our mask function is:

$$M(d) = \frac{1 + cos(\frac{d\pi}{W_p})}{2}, \tag{3.2}$$

where W_p denotes the width of the predicted region in the training stage.

Global and Local Adversarial Loss is employed to learn high-level details to solve the blurry problem of MSE loss. Specifically, two types of discriminators, *i.e.*, one global discriminator and one local discriminator, are used to make that the predicted contents can not be distinguished from real scenes. The global discriminator takes the entire image scene, containing both the input region and the predicted region, as input. In comparison, the local discriminator takes only the predicted region. Besides, we follow the same strategy in WGAN-GP [26] to construct each discriminator, glocal or local.

WGAN-GP enforces a muted constraint, a gradient penalty regarding the gradient norm, on the discriminator. Let $\tilde{x} \sim \mathbb{P}_{\tilde{x}}$ denote a random sample from the predicted scenes, the formular of our objective is,

$$\max_G \min_D \lambda_{gp} \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_{\tilde{x}}} [(\| \nabla_{\tilde{x}} D(\tilde{x}) \|_2 - 1)^2] + \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g} [D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r} [D(x)]. \tag{3.3}$$

Thus the formular of our discriminator loss, $\mathscr{L}_{dis}$, is

$$\mathscr{L}_{dis} = \min_D \lambda_{gp} \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_{\tilde{x}}} [(\| \nabla_{\tilde{x}} D(\tilde{x}) \|_2 - 1)^2] + \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g} [D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r} [D(x)]. \tag{3.4}$$

Moreover, the generator loss, $\mathscr{L}_{gen}$, is

$$\mathscr{L}_{gen} = \min_G - \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g} [D(\tilde{x})] \tag{3.5}$$

For the local adversarial loss, $\mathscr{L}_{dis}^{local}$ and $\mathscr{L}_{gen}^{local}$, the real sample x and fake sample $\tilde{x}$ are the generated region (*i.e.*, the right half in our setting) of the ground-truth image and the generated image. In contrast, the global adversarial loss, $\mathscr{L}_{dis}^{global}$ and $\mathscr{L}_{gen}^{global}$, takes the entire ground-truth image and the entire generated scene as inputs.

Finally, the complete adversarial loss for training discriminator, $\mathscr{L}_D$, is

$$\mathscr{L}_D = \mathscr{L}_{dis}^{global} + \mathscr{L}_{dis}^{local}. \tag{3.6}$$

And the complete generator loss, $\mathscr{L}_G$, is

$$\mathscr{L}_G = \lambda_{rec}\mathscr{L}_{rec} + \lambda_{adv}[\beta\mathscr{L}_{gen}^{global} + (1-\beta)\mathscr{L}_{gen}^{local}]. \tag{3.7}$$

We set $\lambda_{gp} = 10$, $\beta = 0.9$, $\lambda_{adv} = 0.002$, and $\lambda_{rec} = 0.998$ in all the experiments.

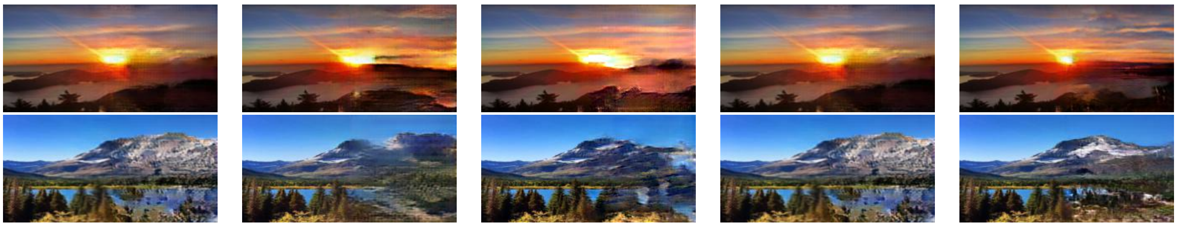
3.3 Results

A new natural scenery dataset is collected and proposed to evaluate our method's effectiveness. The dataset contains about 6,000, 5,000 for training and 1,000 for testing, complicated and diverse natural scenes, including starry sky, riverbank, seaside, valley, snow mountain, etc. About 3,000 scenes are manually selected from SUN dataset [78], while other scenes are collected from the internet. We show some samples of the natural scenery dataset in Fig. 3.8.

Extensive experiments are designed on the proposed natural scenery dataset for evaluating our method. We first show the results on 1-step horizontal generation. Then the results of multi-step generation will demonstrate the powerful ability of our framework in modelling spatial sequence. Notably, our framework predicts natural scenes in only the horizontal direction due to the collected dataset's spatial characteristics. However, our network can theoretically model spatial sequence in any direction after slight modifications in network structure or inference approach.

3.3.1 One-step Generation

During training, Adam optimizer [43] with $\beta_2 = 0.9$ and $\beta_1 = 0.5$ is applied to optimize the losses in Eq. 3.7 and Eq. 3.6. The batch size is 32, and the initial learning rate is 0.0001, which is divided by 10 after 1,000 epochs. The framework will be trained for 1,500 epochs in total. We first warm-up the generator for 1000 iterations, setting $\lambda_{adv} = 0$ and $\lambda_{rec} = 1$. During the rest of training schedule, we use $\lambda_{adv} = 0.002$ and $\lambda_{rec} = 0.998$.



(a) GLC [36] (b) Pix2Pix [38] (c) CA [85] (d) SHC+FC (e) SHC+RCT

Figure 3.9: Comparing our framework with state-of-the-art image generation works on the 1-step generation, ours SHC+RCT framework achieves the best generative quality.

Following the same training strategy of GAN in [1], every time we update the generator parameters, the discriminator parameters are updated n_{cir} times. When the training iteration number is less than 30 or a multiple of 500, we update the discriminator more times, setting $n_{cir} = 30$. In other cases, $n_{cir} = 5$ is used for saving computation. For training, all the images are resized to 144×432, and we employ random cropping and flipping to augment the training data. For testing, all the scenes are in a resolution of 128×256.

In the experiments, popular Fréchet Inception Distance [31] (lower is better) and Inception Score [57] (higher is better) are utilised to evaluate the performance of out-painting.

GRB Number	**FID**	**IS**
0	15.171	2.756
1	14.828	2.765
3 (ours)	**13.713**	**2.852**

Table 3.2: Ablation study of different GRB number. 1 denotes we remove all the GRB blocks except the one where the feature dimensions are $16 \times 32 \times 256$.

Comparison with Previous Works. For evaluating our generation performance, we make comparisons with some latest image generation methods. Fig. 3.9 and Table. 3.3 show the comparison results, and our framework generates the most realistic prediction results, demonstrating our effectiveness in modelling spatial sequence information.

Our method outperforms all the other methods in the FID score, and the IS score is slightly worse than CA [85], as shown in Table. 3.3. The reason is that CA's contextual attention module directly utilised the original region's information to reconstruct the predicted region. However, the contextual attention makes predictions blurry where far

Method	FID	IS
Pix2Pix [38]	19.734	2.825
GLC [36]	14.825	2.812
CA [85]	19.040	**2.931**
FC+SHC	15.186	2.845
RCT+SHC (Ours)	**13.713**	2.852

Table 3.3: Comparison results in FID [31] and IS [57] metrics. In the testing split of proposed natural scenary dataset, the images have an average IS score of 3.387.

away from the input region, as shown in Fig. 3.9, 3.12. Besides, contextual attention results in a worse FID score (19.040 *vs*our 13.713). CA is a suitable approach for small patch prediction but is ineffective in large-scale outpainting.

Ablation Study. Ablation studies are conducted for SHC and RCT to evaluate their performance. Fig. 3.7 shows the qualitative comparisons. Without both SHC and RCT, the predictions have clear faultage on the junction line between the given scene. Applying SHC layers can mitigate the problem of faultage. However, the predicted contents are still blurred, where the prediction position is far from the junction line. We improve the prediction quality in faraway regions by replacing the fully-connected layer, which connects the decoder and encoder, with our proposed RCT, which performs better in modelling spatial sequence. Table.3.2 shows the results of the ablation study of GRB numbers. The quality of results progressively improves with more GRB blocks, which demonstrates the necessity of GRB.

3.3.2 Multi-Step Prediction

For further evaluating our effectiveness in modelling spatial sequence, the models trained in Sec. 3.3.1 are used in the experiments of multi-step generation, where the last step's prediction will be used as the input for the next step iteratively. The predictions from all the steps will be stitched into an entire and large-scale prediction.

Fig. 3.10 shows the qualitative results of multi-step generation in a single horizontal direction, from left to right, while Fig. 3.11 shows the generation in both the horizontal directions, left and right. Our framework is good at maintaining the scene consistency and harmony after multiple steps of prediction. In contrast, Fig. 3.12 shows the results of other competitors [36, 38, 85] on multi-step generation, where the quality of their predictions fastly degrades along with the increasing steps. Besides, our method with

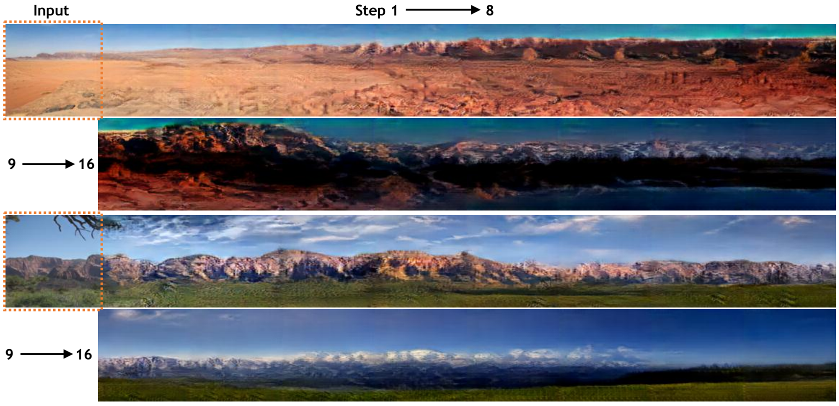


Figure 3.10: We can generate outpainting images with a huge scale in length. Given an image scene with a size of 128×128, our method iteratively generates a 16 times larger scene in the horizontal direction from left to right, leading to a large prediction with a size of 2176 pixels in length. Since the predictions are too long, we divide every prediction into two halves.

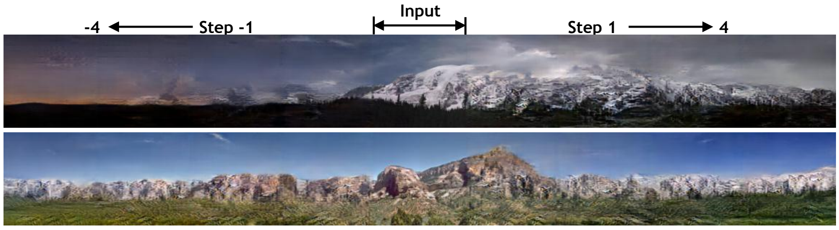


Figure 3.11: We generate outpainting images in both the horizontal directions, left and right. Given an input image, our method iteratively generates a 4 times larger scene for each horizontal direction, leading to a large prediction with a size of 1152 pixels in length.

only a fully-connected (FC) connection maintains a better consistency but will suffer from the blurry problem. All these large-scale results demonstrate that our framework is robust in large-scale image outpainting, and the proposed RCT module is promising in modelling spatial sequence information.

We also prove that our method predicts contents instead of taking over the patches in the input scene. Fig. 3.13 shows a snow mountain prediction starting from a scene with rarely observable contents. Such a case is extremely challenging for traditional non-deep-learning works (e.g., [44, 62, 73, 87]) since it is difficult to find useful patches from such a given seen region, which contains nearly null observable textures.

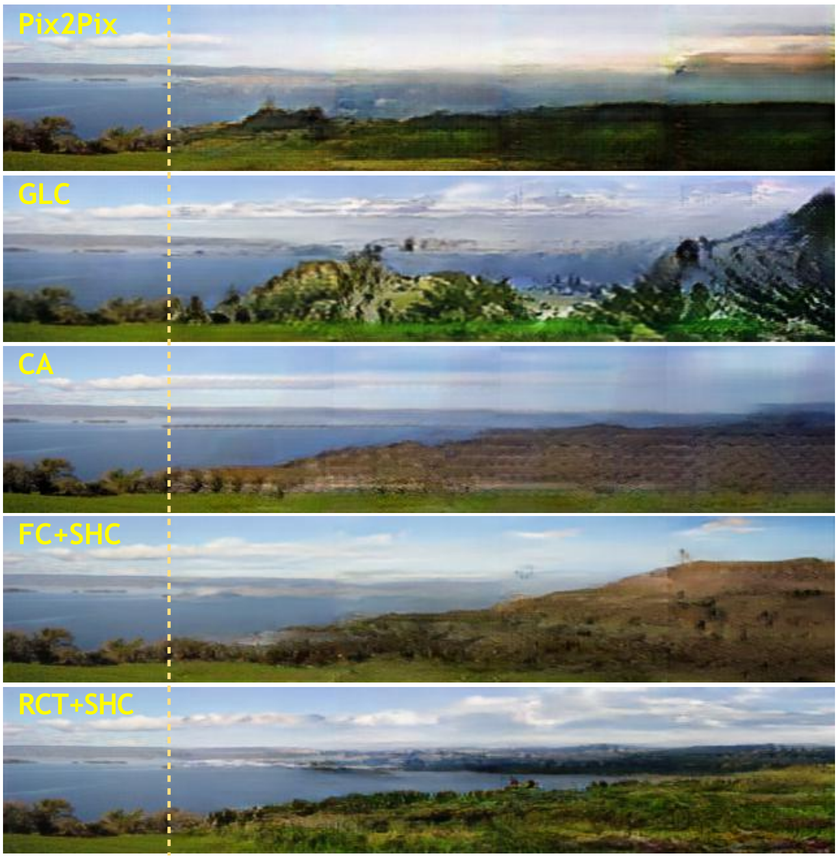


Figure 3.12: Comparison results on multi-step generations.

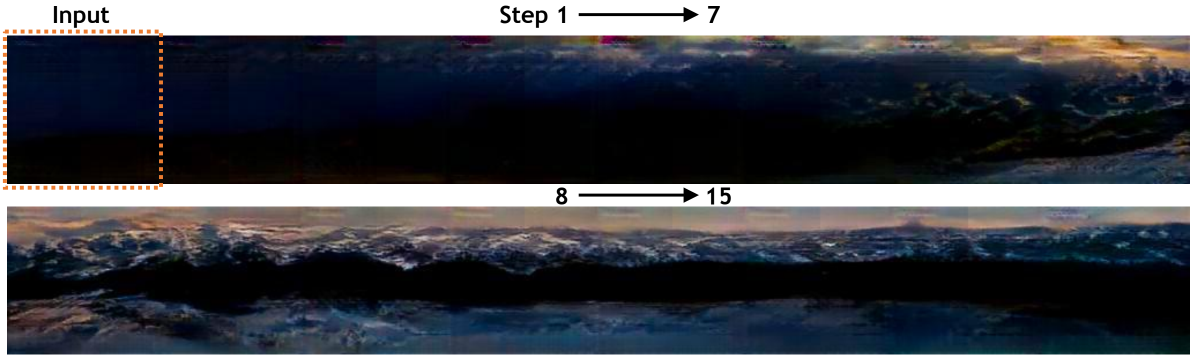


Figure 3.13: A snow mountain prediction starting from a scene with rarely observable contents. Such a case is extremely difficult for traditional non-deep-learning works.

3.4 Conclusion

An efficient generative framework for large scale image generation by modelling spatial sequence is proposed in this chapter. The framework, to the best of our knowledge, is the first framework on the strength of deep neural networks for solving the outpainting problem. By introducing the proposed RCT, SHC, and GRB components, our framework is capable of predicting large-scale scenes with realistic quality. We can also iterate the generator to progressively generate image scenes with extremely large scales in length, which is unprecedented. Besides, we collect a new natural scenery dataset for evaluating the performance of image outpainting. our method outperforms all competitors [36, 38, 85] without accident.

CHAPTER 4

Collaborative Foreground-Background Integration for Video Understanding

4.1 Introduction

Temporal sequence modelling is essential in many video understanding tasks with long-term videos in data, such as Video Object Segmentation (VOS) [6], Video Semantic Segmentation [92], and Video Object Tracking (VOT) [4]. Among these tasks, VOS is one of the basic tasks in video understanding with many valuable applications in computer vision, such as self-driving cars [88] and augmented reality [47]. This chapter explores a more efficient and effective temporal sequence modelling framework for the challenging semi-supervised VOS, which aims to segment objects in a pixel-wise manner across the entire video sequence.

Many preliminary VOS solutions [6, 46, 71] were based on online fine-tuning, *i.e.*, fine-tuning on the first frame in the inference stage. To avoid using inefficient online fine-tuning, recent methods (*e.g.*, [48, 70, 83]) focused on improving the effectiveness of temporal sequence modelling and maintaining a fast run-time. STMVOS deployed a memory bank for reading and storing the object attributes from the frames in history. However, extensive simulated data generated from various large image datasets [13, 21, 27, 45, 59] are necessary for training STMVOS. Without such an elaborate training procedure, the performance of STMVOS drops a lot. FEELVOS [70] proposed a glocal (between the reference and current object frame) and local (between the previous and

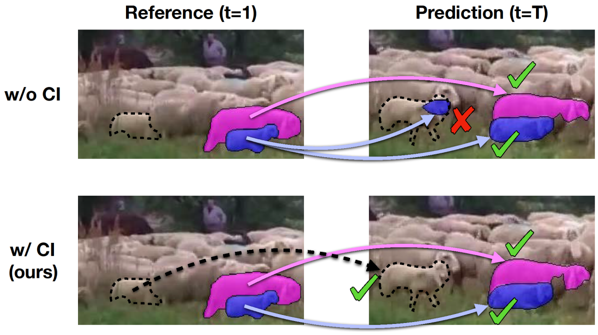


Figure 4.1: CI denotes Collaborative Integration. The dot-line arrow denotes the proposed background matching. In the given video sequence, we have two foreground objects, *i.e.*, pink sheep and blue sheep. Without background matching, the blue sheep's prediction is confused by one similar sheep in the background region, as shown in the top line. By adding the background matching, the confusion problem is relieved, as shown in the bottom line.

current object frame) matching mechanism using pixel-wise feature embedding for efficient multi-object video understanding. The proposed FEELVOS framework is efficient and simple but is not effective as STMVOS in performance.

Although the above-mentioned efforts have achieved notable promotion for VOS, few VOS works focus on modelling the background region's temporal sequence information and only pay attention to finding robust matching mechanisms regarding the foreground target (s). One person is intuitively easy to recognise the foreground object in videos when accurately removing the opposite background region. Furthermore, common video scenes usually focus on a group of similar instances, such as humans, animals, and cars. When targeting similar instances, the missing of matching for background instances will lead to an inevitable problem of confusion in prediction. Fig. 4.1 gives a case of the background confusion problem in our practice. Such a phenomenon motivates us that we have to equally treat both the foreground and background regions for learning more contrastive feature embedding, relieving the background confusion, and thus promoting the VOS performance.

Unlike previous works that only focus on the foreground object's temporal sequence modelling (s), we argue that the background dependencies are equally important. Hence, we propose a Collaborative Foreground-Background Integration (CFBI) approach for semi-supervised VOS. In CFBI, the learned feature embeddings from the foreground target and the background region are implicitly compelled to be contrastive, thus promoting the network's representative ability and improving the VOS accuracy. To handle various object scales, both pixel-level and instance-level information types are applied for matching targets between the given reference and the video frames required to predict. Besides, a novel transformation module is designed to model channel relationships in a single frame to improve representation ability. As to the training process, a balanced random-crop augmentation approach is proposed to avoid biasing the network parameters to the background attributes in datasets. We significantly increase temporal modelling effectiveness by combining all the above methods and components while the framework is still keeping efficient.

For validating the effectiveness of the CFBI framework, we design sufficient experiments on YouTube-VOS 2018 [81], DAVIS 2017 [53], and DAVIS 2016 [52], three of the most popular VOS benchmarks. Without the use of post-processing, online fine-tuning, or simulated data, we outperform all the competitors on all three benchmarks, *i.e.*, the validation splits of YouTube-VOS ($\mathcal{J}$&$\mathcal{F}$ **81.4%**), DAVIS 2016 (**89.4%**), and DAVIS 2017 (**81.9%**). More than that, the performance can be boosted to **82.7%**, **90.1%**, and **83.3%** by applying a flipping & multi-scale augmentation during inference. More ablation studies prove the necessity of each proposed component or method.

4.2 The Proposed Approach

Many previous methods (*e.g.*, [70, 83]) have been proposed to model the foreground region's temporal sequence information. A concept of instance-level embedding was introduced by OSMN [83] for efficiently predicting the segmentation mask of video objects. However, the diversity of features within the objects' appearance details was not considered by using only instance-level embedding vectors, thus leading to imprecise results. In comparison, some methods [11, 70] proposed to learn pixel-wise embedding for matching every pixel of the object, promoting the feature diversity and resulting in more promising performance. However, the pixel-level matching mechanism can only utilise contextual information in a small local range and thus is easy to be confused by some similar textures or instances in the background region.

A solution to the above problems is to incorporate both the instance-level and pixel-level information types into a framework. Besides, it is important to treat both the foreground and background regions equally for learning more contrastive feature embedding. Thus, a collaborative foreground-background integration framework is proposed in this chapter. An overview of the detailed framework is shown in Fig. 4.2. In the framework, the background embedding is considered for collaborating with foreground embedding, thus implicitly encouraging the learned pixel-level embeddings of the back-

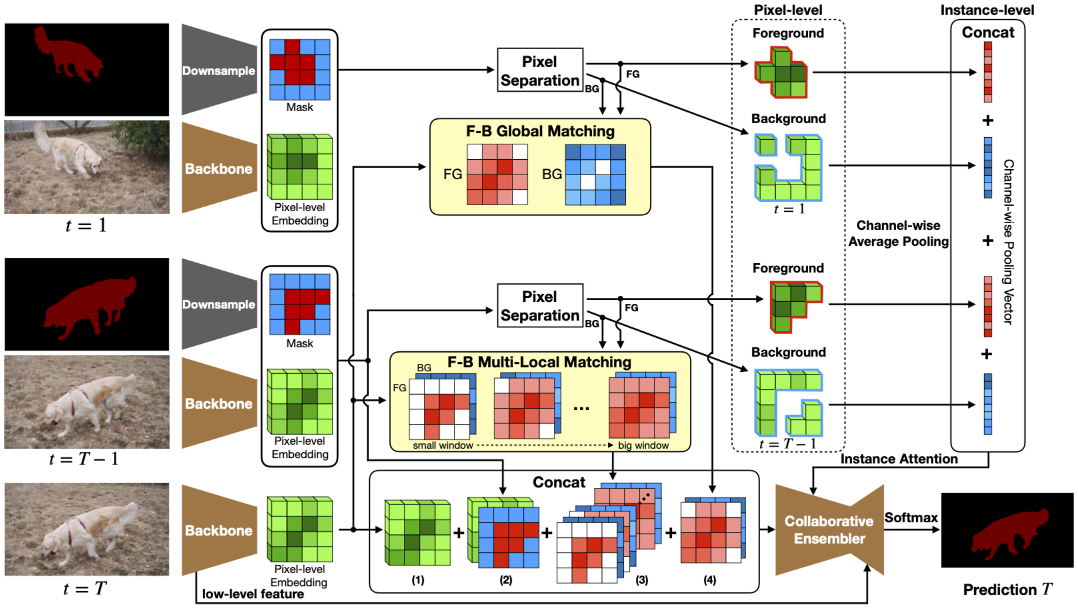


Figure 4.2: CFBI **overview**. F-G, red color, and blue color denote Foreground-Background, foreground region, background region, respectively. The confidence of the matching map with a deeper colour will be higher. Given the current frame ($t = T$), the reference frame with mask ($t = 1$), and the previous frame with mask ($t = T-1$), CFBI will model their temporal dependencies and predict the target segmentation of the current frame. First, we use a backbone to extract pixel-level embeddings for all three frames. The pixel-level embeddings of $t = T-1$ and $t = 1$ will be separated into foreground embeddings and background embeddings regarding the given masks. Then, the instance-level attention and pixel-level matching will be applied to both the foreground embeddings and background embeddings to model temporal sequence information and predict the current frame segmentation. During inference, given the first frame mask, we can predict target masks from the reference frame to the final one by iterating such a prediction process.

ground region and the foreground object to be more diverse in the embedding space. Moreover, both the pixel-level and instance-level matching mechanisms are deployed for modelling temporal sequence information. Besides, we propose a multi-local matching approach to make the pixel-level matching robust to different speeds of object motion, and an instance-level attention module is designed for utilising instance guidance in a more efficient manner. For aggregating all the foreground, background, instance-level, and pixel-level embeddings, an efficient collaborative ensembler with large receptive fields is designed to predict accurate segmentation masks.

4.2.1 Pixel-level Matching

We show the architecture of our pixel-level matching in the middle of Fig. 4.2. We utilise a global-local matching approach [70] for modelling temporal information between the first, the previous, and the current frames. In comparison, the matching mechanism is also applied to the background region, and we propose a multi-window strategy for improving the robustness of the local matching.

For modelling the temporal sequence information in the background region, the foreground pixel-wise distance [70] is modified to distinguish between the background and foreground. Let F_t and B_t denote all the foreground object pixels and the background pixels at time t, the formula of the distance between pixel q at time t and pixel p at time T (the current frame) with regards to their pixel-wise embedding, e_q and e_p, is,

$$D_t(p,q) = \begin{cases} 1 - \frac{2}{1+exp(||e_p - e_q||^2 + b_F)} & \text{if } q \in F_t \\ 1 - \frac{2}{1+exp(||e_p - e_q||^2 + b_B)} & \text{if } q \in B_t \end{cases}, \tag{4.1}$$

where b_B and b_F are trainable background and foreground biases, used to learn different distance functions for background and foreground. Thus, the constrast between the foreground and the background embeddings is strengthened.

Foreground-Background Global Matching. The global matching is applied between the reference frame $t = 1$ and the current frame $t = T$. Let all pixels at time t are in the set $\mathscr{P}_t$, the pixel set of a foreground object o at time t can be denoted as $\mathscr{P}_{t,o} \subseteq \mathscr{P}_t$. Thus, the formula of the foreground global matching between the object pixels of the reference frame and one pixel p of the current frame is,

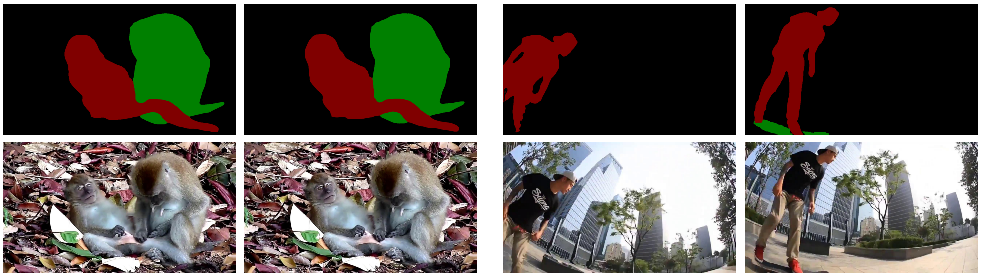
$$G_{T,o}(p) = \min_{q \in \mathscr{P}_{1,o}} D_1(p,q). \tag{4.2}$$

The formula for the background global matching is similar. Let $\overline{\mathscr{P}}_{t,o} = \mathscr{P}_t \backslash \mathscr{P}_{t,o}$ denote the set of relative background pixels regarding the foreground object o, the formula for the background global matching is,

$$\overline{G}_{T,o}(p) = \min_{q \in \overline{\mathscr{P}}_{1,o}} D_1(p,q), \tag{4.3}$$

which is similar to the foreground one.

Foreground-Background Multi-Local Matching. There are diverse movement speeds in video understanding, from nearly static to fast, of foreground objects, as shown in Fig. 4.3. However, FEELVOS utilised only a fixed window of neighbouring pixels to



(a) Slow motion

(b) Fast motion

Figure 4.3: In video understanding, there are diverse movement speeds, from nearly static to fast, of foreground objects. The demonstration video sequences are sampled from YouTube-VOS [81].

conduct the local matching mechanism and hence is difficult to fit various and different motion rates. In contrast, we propose a multi-local matching approach, *i.e.*, learning to select a suitable local window, to make the pixel-level matching robust to different object motion speeds. The matching maps of small windows are extracted from the largest window in a lightweight manner on the implementation. Hence, proposed multi-local matching is as efficient as local matching with a single matching window.

First, $H(p,k)$ and $K = \{k_1, k_2, ..., k_n\}$ denote the neighbourhood pixel set of pixel p in a maximum distance of k and the set of all the neighbourhood scales, respectively, the formula of the foreground multi-local matching between the object pixels of the previous frame $(T-1)$ and one pixel p of the current frame is,

$$ML_{T,o}(p,K) = \{L_{T,o}(p,k_1), L_{T,o}(p,k_2), ..., L_{T,o}(p,k_n)\}, \tag{4.4}$$

where

$$L_{T,o}(p,k) = \begin{cases} \min_{q \in \mathscr{P}^{p,k}_{T-1,o}} D_{T-1}(p,q) & \text{if } \mathscr{P}^{p,k}_{T-1,o} \neq \emptyset \\ 1 & \text{otherwise} \end{cases}, \tag{4.5}$$

and $\mathscr{P}^{p,k}_{T-1,o} := \mathscr{P}_{T-1,o} \cap H(p,k)$ denotes the object pixels placed in the neighborhood window. Similarly, the formula of the background multi-local matching is

$$\overline{ML}_{T,o}(p,K) = \{\overline{L}_{T,o}(p,k_1), \overline{L}_{T,o}(p,k_2), ..., \overline{L}_{T,o}(p,k_n)\}, \tag{4.6}$$

where

$$\overline{L}_{T,o}(p,k) = \begin{cases} \min_{q \in \overline{\mathscr{P}}^{p,k}_{T-1,o}} D_{T-1}(p,q) & \text{if } \overline{\mathscr{P}}^{p,k}_{T-1,o} \neq \emptyset \\ 1 & \text{otherwise} \end{cases}, \tag{4.7}$$

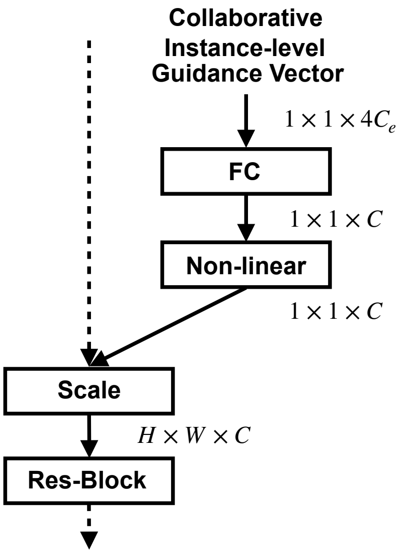


Figure 4.4: The detailed structure of the instance-level attention module. C, W, and H mean channel dimension, width, and height, separately. C_e refers to the channel number of pixel-level embedding extracted from the backbone.

and $\overline{\mathscr{P}}^{p,k}_{T-1,o} := \overline{\mathscr{P}}_{T-1,o} \cap H(p,k)$ denotes the backgournd pixels placed in the neighborhood.

Apart from the global and local matching maps, the pixel-wise embedding and the previous frame mask are also concatenated with the current frame embedding, as shown in the bottom box of Fig. 4.2, followed by our collaborative ensembler to integrate all the temporal information. It has been proved that concatenating the previous frame mask is effective in VOS [70]. In our practice, the framework performance can be further improved by about 0.5% if we concatenate both the previous frame's mask and pixel-wise embedding.

4.2.2 Instance-level Attention

In addition to pixel-level matching, we apply an instance-level attention mechanism to help model temporal sequence information for objects with larger scales, as shown in the right of Fig 4.2

We first divide each pixel-level embedding extracted from the backbone into foreground pixels and background ones regarding the foreground region and background region in the given mask for the reference and previous frames. Then, channel-wise average pooling is applied to each pixel group to transfer the four pixel-level embeddings (*i.e.*, $\mathscr{P}_{1,o}$, $\mathscr{P}_{T-1,o}$, $\overline{\mathscr{P}}_{1,o}$, and $\overline{\mathscr{P}}_{T-1,o}$) into instance-level embedding vectors, and all the four instance-level embedding vectors are concatenated into one final instance-level embedding vector. Hence, such a vector aggregates the instance-level information from both the foreground and background regions and the reference and previous frames.

After generating the instance-level vector, a designed lightweight attention mechanism is applied for controlling the output network, *i.e.*, Collaborative Ensembler (CE). Inspired by channel attention methods (*e.g.*, [33]), we construct a gate module to adjust each feature channel's scale based on the information containing in the instance-level vector. The gate module's architecture is one *fully-connected* layer followed by a non-linear activation function, which is simple yet effective in our practice.

Compared to the pixel-level embedding, the instance-level vector containing global contextual information of the object. Such global information is useful for relieving the problem of local ambiguities [67], which derives from the local receptive fields of features, like the pixel-level embedding.

4.2.3 Collaborative Ensembler (CE)

To aggregate both the instance-level and pixel-level informations, an efficient collaborative ensembler network is designed to make large receptive fields and implicitly model the temporal dependencies on both the foreground and background regions, as shown in the bottom right of Fig. 4.2.

Following recent methods in semantic segmentation [9, 10], collaborative ensembler employs a downsample-upsample pipeline containing three downsampling stages of bottleneck Res-Blocks [29] followed by an ASPP [10] module and an upsampling Decoder [10] module. Dilated convolutional layers are used in Res-Block for capturing more contextual information. The Res-Block numbers for the three stages are 2, 3, 3, respectively. In each stage, the dilation rates of Res-Blocks are 1, 2, 4 (or 1, 2 for Stage 1) in order. Collaborative ensembler takes the pixel-level matching results as input, and the instance-level attention mechanism is applied before each Res-Block of collaborative ensembler.

4.2.4 Gated Channel Transformation (GCT)

To further improve CE's representation ability, a novel transformation module, *i.e.*, Gated Channel Transformation, is designed to model channel-wise and contextual information, as shown in 4.5. In detail, GCT firstly aggregates global contextual embedding by using ℓ_2 norm for each channel. We apply channel-wise embedding weight, $\boldsymbol{\alpha}$, to adjust each embedding's importance. After that, a lightweight ℓ_2 normalisation operation is employed

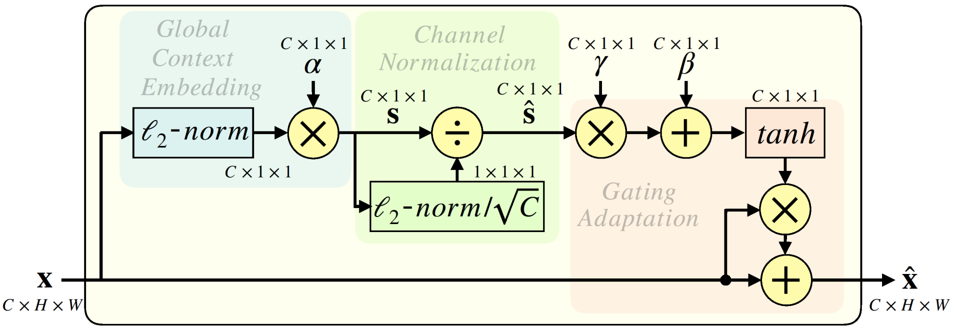


Figure 4.5: The detailed architecture of Gated Channel Transformation.

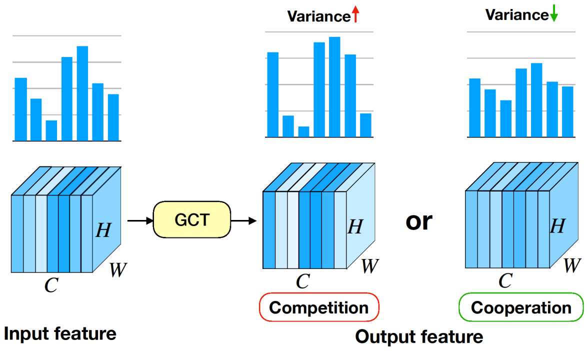


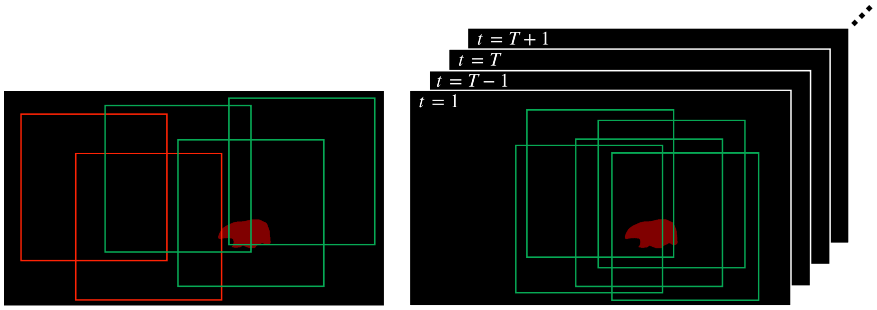
Figure 4.6: By using normalization operation, GCT is able to explicitly model some channel relations, like cooperation (decreasing channel variance) and competition (increasing channel variance).

to adjust the variance of channel activations and thus model channel relations, like competition and cooperation. An illustration is shown in Fig. 4.6. In the normalisation operation, we use channel-wise gating weight and gating bias, $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$, to control the behaviour of relation modelling. When the gating weight of one channel is positively activated, GCT will push the activation of this channel far away from the ℓ_2 norm of all the, like promoting this channel to compete with other channels. Oppositely, when one channel's gating weight is negatively activated, GCT tends to promote this channel to cooperate with the others.

In our experiments, we apply GCT before all the convolutional layers of CE to improve CE's representation ability and thus promote VOS's performance accordingly.

4.2.5 Implementation Details

We adjust the commonly used traditional random cropping and the sampling strategy used in FEELVOS for improving the training effectiveness.



(a) Normal

(b) Balanced

Figure 4.7: (a) If we use traditional random cropping, some cropping boxes (red boxes as shown) will contain little or even no object region. (b) By using the proposed balanced-crop, all the cropping boxes contain enough foreground pixels.

Balanced Random-Crop. In video understanding tasks, the foreground object usually only occupies a small area in the lens. If we use traditional random cropping, as shown in Fig. 4.7, some cropping boxes (red boxes as shown) will contain little or even no object region. More background information is easy to bias the model to background attributes, which is not desired.

In this section, a balanced random-crop strategy is proposed for solving such an issue for video segmentation. The balanced random-crop is applied on a sequence of frames (*i.e.*, the current frame, the reference frame, and the previous frame). For all the frames, the cropping window is consistent and restricted to contain enough foreground object pixels. The restriction approach is computationally efficient. To be specific, the method will judge whether the randomly cropped window contains object pixels or not after cropping. If not, the cropping process will be repeated until the desired condition is met.

Sequential Training. For each training iteration, FEELVOS sampled only one current frame for each video sequence. In this case, all the previous masks are sampled from the ground-truth labels. In contrast, RGMP proposed to predict multiple steps in training, and the predictions in the early steps will be used to guide the segmentation in later steps. This training strategy is more consistent with the test stage, thus leading to better performances in evaluation.

In our experiments, we sample longer video frame sequences for every training iteration. Specifically, we first sample a batch of videos for each iteration. After that, we randomly sample one frame and $N+1$ consecutive frames for each video to be the reference frame, the previous frame, and the current sequence required for prediction. Samely, the predictions in the early steps will be used to guide the segmentation in later

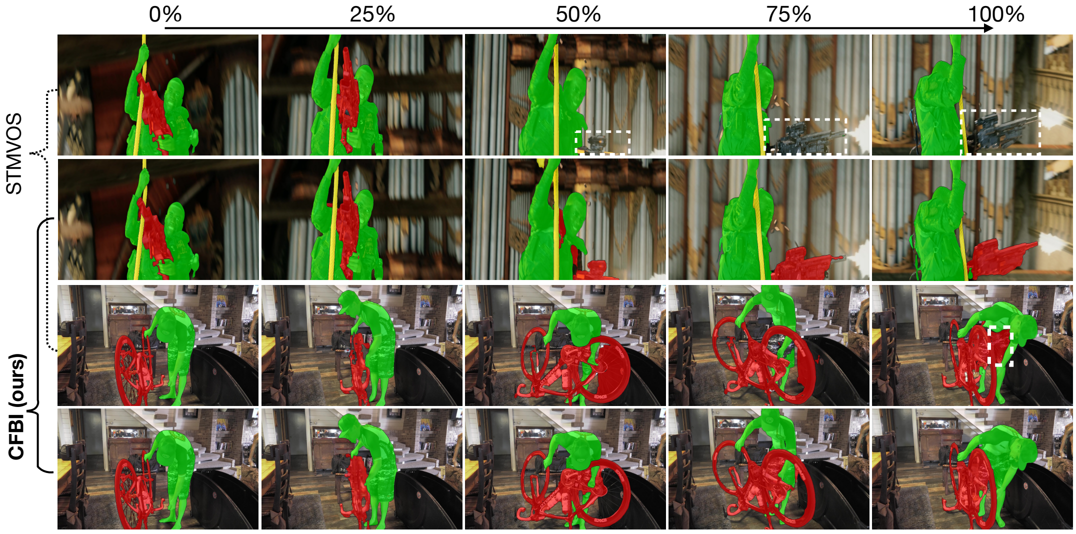


Figure 4.8: On the validation split of DAVIS 2017, CFBI is compared with a state-of-the-art VOS method, STMVOS. STMVOS confuses a part of the person and the bicycle in the second video. Besides, CFBI is successful in segmenting the foreground weapon after blur and occlusion in the top sequence, while STMVOS fails.

steps.

Training Details. The backbone network of CFBI is one of the latest image semantic segmentation networks, DeepLabv3+ [10] based on dilated ResNet-101 [10], which is more efficient than Xception-65 [15] adopted by FEELVOS. The backbone is pre-trained in image classification on ImageNet [17] and in image semantic segmentation on COCO [45]. One depth-wise separable convolution is used to propagate the backbone's final feature into the space of pixel-level embedding.

For VOS training, the segmentation loss is following the bootstrapped cross-entropy loss used in FEELVOS. The window sizes of the multi-local matching are $K = \{2,4,6,8,10,12\}$ in our experiments. During the training of VOS, the BN [37] parameters in the backbone are frozen, and we apply Group Normalization (GN) [75] to collaborative ensembler. GN is robust to small training batch sizes. The foreground bias b_F and background bias b_B are initialised to zero during training. We use a window size of 465×465 for the balanced random-crop and set N to 3 for sequential training. Using longer sequence length will not further improve performance but takes much more computational resources.

The training splits of YouTube-VOS [81] (3471 videos) and DAVIS 2017 [53] (60 videos) are used for training VOS. All the videos are first downsampled to 480P resolution before training. The training optimiser is SGD with a momentum of 0.9. The base learning rate and training iterations are 0.01 and 100,000 for YouTube-VOS experiments or 0.006 and

				Seen		Unseen	
Methods	FT	SD	Mean	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
Validation 2019 Split							
CFBI			**81.0**	**80.6**	**85.1**	**75.2**	**83.0**
CFBI^{MS}			**82.4**	**81.8**	**86.1**	**76.9**	**84.8**
Testing 2019 Split							
MST* [89]		✓	81.7	80.0	83.3	**77.9**	85.5
EMN* [90]		✓	81.8	**80.7**	**84.7**	77.3	84.7
CFBI			81.5	79.6	84.0	77.3	85.3
CFBI^{MS}			**82.2**	80.4	**84.7**	**77.9**	**85.7**
Validation 2018 Split							
AG [41]			66.1	67.8	-	60.8	-
PReM [46]	✓		66.9	71.4	75.9	56.5	63.7
BoLT [72]	✓		71.1	71.6	-	64.3	-
STM^{-} [48]			68.2	-	-	-	-
STM [48]		✓	79.4	79.7	84.2	72.8	80.9
CFBI			**81.4**	**81.1**	**85.8**	**75.3**	**83.4**
CFBI^{MS}			**82.7**	**82.2**	**86.8**	**76.9**	**85.0**

Table 4.1: Compare CFBI with existing frameworks on the validation 2018, validation 2019, and testing 2019 splits of YouTube-VOS [81]. ^{MS} denotes that the flip & multi-scale augmentation is used in the evaluation. *, SD, and FT denote model ensemble, simulated training data, and online fine-tuning, respectively.

50,000 for DAVIS experiments. The batch size is 8 videos or 6 videos for YouTube-VOS or DAVIS, respectively. The data augmentations are a combination of random scaling, the balanced random-crop, and random flipping. The scales used in the multi-scale inference are $\{1.0, 1.15, 1.3, 1.5\}$ for YouTube-VOS and $\{2.0, 2.15, 2.3\}$ for DAVIS.

4.3 Results

Following recent methods [48, 70], sufficient experiments are conducted on DAVIS 2017 [53] and DAVIS 2016 [52], and YouTube-VOS [81] to evaluate the effectiveness of our framework. CFBI is trained on the training split of YouTube-VOS for the YouTube-VOS experiments. For both the DAVIS-2017 and DAVIS-2016 experiments, we train CFBI on the training split of DAVIS 2017. We also evaluate the DAVIS results by training

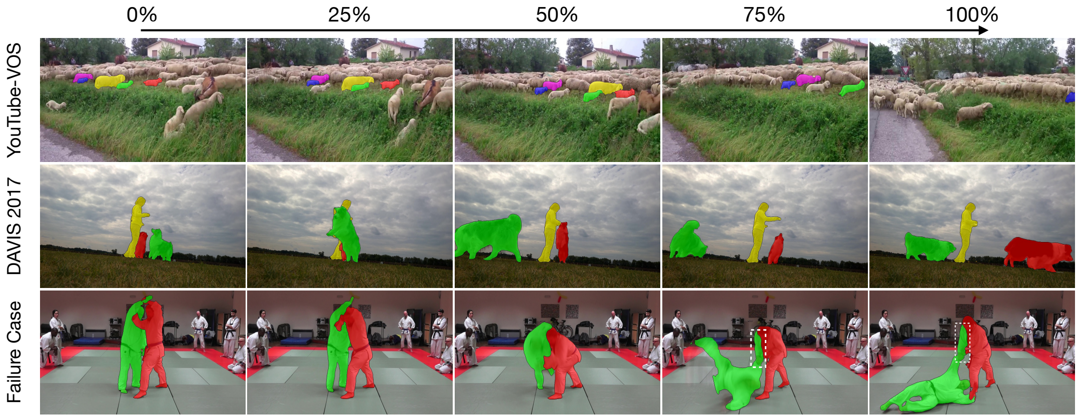


Figure 4.9: On YouTube-VOS and DAVIS 2017, we show some qualitative results of CFBI. On YouTube-VOS, CFBI succeeds in matching and segmenting all the desired sheep with a similar appearance from a flock of sheep. On DAVIS 2017, CFBI performs perfectly after the occlusion on the dogs and the person. However, when two objects have very similar textures and are very close, CFBI is still confused by a part of the object, as shown in the bottom sequence.

CFBI on both the training splits of DAVIS 2017 and YouTube-VOS.

Three popular metrics are used for evaluation, *i.e.*, $\mathcal{J}$ score, $\mathcal{F}$ score, and their mean value ($\mathcal{J}$&$\mathcal{F}$). $\mathcal{J}$ score is computed as the average Intersection-over-Union score between the ground truth mask and the predicted segmentation results. $\mathcal{F}$ measures an average boundary similarity between the mask boundary of the ground truth and the prediction. All our results are evaluated by using the official tools or the official evaluation server for a fair comparison.

4.3.1 Comparison Results

YouTube-VOS [81] is currently the largest public dataset proposed for multi-object video segmentation containing about 4500 videos. The popular DAVIS-2017 dataset is nearly 38 times smaller than YouTube-VOS. Specifically, YouTube-VOS consists of a training split with 3471 videos and 65 classes, a validation split with 507 videos and extra 26 unseen classes, and a test split with 541 videos and extra 29 unseen classes. Notably, the validation and test splits of YouTube-VOS have many unseen categories and are thus suitable for evaluating VOS frameworks' generalisation ability.

We compare CFBI with state-of-the-art network architectures on the validation 2018, validation 2019, and testing 2019 splits of YouTube-VOS. Without the use of post-processing, online fine-tuning [6, 71], or simulated data [48, 76], we significantly

Methods	FT	SD	Mean	$\mathcal{J}$	$\mathcal{F}$	t/s
OSMN [83]			-	74.0		0.14
PML [11]			77.4	75.5	79.3	0.28
VideoMatch [34]			80.9	81.0	80.8	0.32
RGMP^{-} [76]			68.8	68.6	68.9	0.14
RGMP [76]		✓	81.8	81.5	82.0	0.14
A-GAME [41] **(YT)**			82.1	82.2	82.0	**0.07**
FEELVOS [70] **(YT)**			81.7	81.1	82.2	0.45
OnAVOS [71]	✓		85.0	85.7	84.2	13
PReMVOS [46]	✓		86.8	84.9	88.6	32.8
STMVOS [48]		✓	86.5	84.8	88.1	0.16
STMVOS [48] **(YT)**		✓	**89.3**	**88.7**	89.9	0.16
CFBI			86.1	85.3	86.9	0.18
CFBI **(YT)**			**89.4**	88.3	**90.5**	0.18
CFBI^{MS} **(YT)**			**90.7**	**89.6**	**91.7**	9

Table 4.2: Compare CFBI with existing frameworks on the validation split of DAVIS 2016 [52]. **(YT)** denotes both the training splits of YouTube-VOS and DAVIS 2017 are used in the training process.

outperform all the competitors and achieve a mean score of **81**.**4**% on the validation 2018 split. More than that, we can further boost the performance to **82**.**7**% by applying a flipping & multi-scale strategy during inference. In comparison, the previous best STMVOS, which used extra training data simulated from various large image datasets [13, 21, 27, 45, 59], is 2.0% lower than our 81.4% result. Without such an elaborate simulated training procedure, the performance of STMVOS drops a lot from 79.4% to 68.2%.

To further demonstrate our generalisation ability, CFBI is compared with the winner solution, EMN [90], in the 2nd Large-scale Video Object Segmentation Challenge. On the testing 2019 split of YouTube-VOS, our single-model result exceeds the challenge winner, which utilises model ensembling, by 0.4% in the mean score.

DAVIS 2016 [52] is a single-object video segmentation dataset consisting of 20 videos with high-quality annotations in the validation split. We compare CFBI with existing methods on the validation split of DAVIS 2016 [52]. The results are shown in Table 4.2. Training with both the training splits of DAVIS 2017 and YouTube-VOS, we achieve a mean score of **89**.**4**%. This result is a bit better than STMVOS (89.3%), which uses extra simulated data. In general, it is easier to over-fit a much smaller dataset. Hence, the accuracy gap between STMVOS and CFBI is smaller on small-scale DAVIS than

Methods	FT	SD	Mean	$\mathcal{J}$	$\mathcal{F}$
		Validation Split			
OSMN [83]			54.8	52.5	57.1
VideoMatch [34]			62.4	56.5	68.2
OnAVOS [71]	✓		63.6	61.0	66.1
RGMP [76]		✓	66.7	64.8	68.6
A-GAME [41] (**YT**)			70.0	67.2	72.7
FEELVOS [70] (**YT**)			71.5	69.1	74.0
PReMVOS [46]	✓		77.8	73.9	81.7
STMVOS [48]		✓	71.6	69.2	74.0
STMVOS [48] (**YT**)		✓	**81.8**	**79.2**	84.3
CFBI			74.9	72.1	77.7
CFBI (**YT**)			**81.9**	**79.1**	**84.6**
CFBI^{MS} (**YT**)			**83.3**	**80.5**	**86.0**
		Testing Split			
OSMN [83]			41.3	37.7	44.9
OnAVOS [71]	✓		56.5	53.4	59.6
RGMP [76]		✓	52.9	51.3	54.4
FEELVOS [70] (**YT**)			57.8	55.2	60.5
PReMVOS [46]	✓		71.6	67.5	75.7
STMVOS [48] (**YT**)		✓	72.2	69.3	75.2
CFBI (**YT**)			**74.8**	**71.1**	**78.5**
CFBI^{MS} (**YT**)			**77.5**	**73.8**	**81.1**

Table 4.3: Compare CFBI with existing frameworks on the validation split and the testing split of DAVIS-2017 [53].

large-scale YouTube-VOS. Compared to FEELVOS, a more fair baseline that CFBI is based on, CFBI performs a much better result **89.4**% *vs*81.7%) while maintaining a much faster run-time (0.18*s* *vs*0.45*s*). Besides, we can boost the performance from **89.4**% to **90.1**% by applying a flipping & multi-scale augmentation during inference.

DAVIS 2017 [53] extends DAVIS 2016 to a multi-object video segmentation dataset, containing 59 objects and 30 videos in the validation split. CFBI excellently outperforms FEELVOS by 10.4% (**81.9**% *vs.* 71.5%), which is shown in Table 4.3. Moreover, using extra simulated data, STMVOS is still worse than our CFBI (81.8% *vs.* **81.9**%). Fig. 4.8 shows some qualitative results of CFBI. Similar to the previous observation, the flipping & multi-scale strategy during inference can improve the performance to be higher (from **81.9**% to **83.3**%). To further demonstrate our generalisation ability, CFBI is also

	original		+SE [33]		**+GCT (ours)**	
Backbone	top-1/5	G/P	top-1/5	GFLOPs/Param	top-1/5	G/P
ResNet-50 [29]	23.8/7.0	3.879/25.61	22.9/6.6	**3.893***/28.14	**22.7/6.3**	3.900/**25.68**
ResNeXt-50 [79]	22.4/6.3	3.795/25.10	22.0/6.1	**3.809***/27.63	**21.7/6.0**	3.821/**25.19**
Inception-v3 [66]	24.3/7.3	2.847/23.87	24.0/7.2	**2.851***/25.53	**23.7/7.1**	2.862/**23.99**
VGG-16 [60]	26.2/8.3	15.497/138.37	25.2/7.7	15.525/138.60	**25.1/7.5**	**15.516/138.38**

Table 4.4: Apply GCT to modern CNNs for evaluating the error performance (%) in image classification on ImageNet [56]. G/P denotes GFLOPS/Parameters.

Backbone	box AP	mask AP
ResNet-50	37.8	34.2
ResNet-50+SE [8]	38.2$_{(0.4)}$	34.7$_{(0.5)}$
ResNet-50+**GCT**	**39.8**$_{(2.0)}$	**36.0**$_{(1.8)}$
ResNet-101	40.1	36.1
ResNet-101+**GCT**	**42.0**$_{(1.9)}$	**37.7**$_{(1.6)}$

Table 4.5: Apply GCT to Mask R-CNN [28] in instance segmentation (mask AP) and object detection (box AP) on COCO [45].

evaluated on the testing split of DAVIS 2017. Compared to the validation split, the testing split is much more challenging. Table 4.3 shows that CFBI makes a significant improvement over STMVOS (74.8%*vs* 72.2%). The inference augmentation can boost our testing result to a much higher **77.5%**. All the above results strongly demonstrate the generalisation ability and robustness of CFBI.

Qualitative Results. More qualitative results of CFBI, on the validation split of YouTube-VOS (**81.4%**) and DAVIS 2017 (**81.9%**), are shown in Fig. 4.9. Compared to STMVOS, CFBI is more robust to challenging cases, like similar objects, blurring, occlusion, and fast motion. On YouTube-VOS, CFBI succeeds in matching and segmenting all the desired sheep with a similar appearance from a flock of sheep. However, when two objects have very similar textures and are very close, CFBI is still confused by a part of the object, as shown in the *judo* video.

Evaluating Gated Channel Transformation. To evaluate the proposed GCT unit further, we also try to apply GCT in more visual tasks, including action recognition, instance segmentation, image classification, and object detection. The related results are shown in Table 4.4, 4.5, and 4.6. Applying GCT before every convolutional layer of the modern CNN networks, all the networks get promising performance gains. Besides, com-

Backbone	NL [74]	**+GCT**
ResNet-101	75.7	$\mathbf{76.2}_{(0.5)}$
ResNet-50	74.6	$\mathbf{75.1}_{(0.5)}$

Table 4.6: Apply GCT to NL [74] in action recognition, using top-1 accuracy (%), on Kinetics.

PL	IL	Mean	$\mathcal{J}$	$\mathcal{F}$
✓	✓	74.9	72.1	77.7
✓*	✓	72.8	69.5	76.1
✓		73.0	69.9	76.0
	✓	72.3	69.1	75.4
		70.9	68.2	73.6

Table 4.7: We ablate different background information from CFBI in these experiments. *, IL, PL denote removing the foreground bias and background bias, the instance-level attention, and pixel-level matching, respectively.

pared to a commonly-used channel attention module, SE [33], the increase of parameters introduced by GCT is negligible, as shown in Table 4.4. In summary, GCT is effective, efficient, and can generalise across various visual tasks and datasets.

4.3.2 Ablation Study

We ablate each proposed component or method to evaluate its necessity on the validation split of DAVIS 2017 in this section.

Background Embedding. We first study the influence of ablating background information from the framework, and the related results are shown in Table 4.7. Without any of the background designs, the performance of CFBI seriously degrades from 74.9% to 70.9%. Such a strong result demonstrates the importance of modelling temporal information of both the foreground and background regions. It will decrease the performance to 73.0% or 72.3% by removing background information from the instance-level attention or the pixel-level matching, respectively. In other words, the background information is more important for extracting better pixel-level embedding. This is reasonable because it is easier for a foreground object to have similar background pixels than background objects. At last, the foreground and background bias, b_F and b_B, are removed from the distance function, leading to a decreased performance of 72.8%. This result proves that

	Methods	$\mathcal{F}$	$\mathcal{J}$	Mean
0	FEELVOS (baseline)	70.9	65.6	68.3
1	w/o instance-level attention	75.5	69.8	72.7
2	w/o balanced random-crop	75.8	69.8	72.8
3	w/o collaborative ensembler	76.1	70.5	73.3
4	w/o sequential training	75.7	70.8	73.3
5	w/o multi-local matching	76.8	70.8	73.8
6	CFBI (Ours)	77.7	72.1	74.9

Table 4.8: Ablation of other components.

the background distance and foreground distance should be independently considered.

Other Components. Table 4.8 shows the ablation study results of all the other proposed components and methods. We reproduce the baseline method, FEELVOS, training with the same setting of CFBI, and Line 0 is the corresponding result (68.3%). In contrast, our CFBI performances much better (74.9% in Line 6).

In line 1, we remove all the instance-level modules from the collaborative ensembler, and use only the pixel-level matching for modelling temporal information. Then, the performance of CFBI drops to the lowest 72.7 among all the components, proving the importance of the instance-level information.

In line 2, the balanced random-crop is replaced by traditional random cropping. Under this case, the framework result deteriorates to 72.8%. In other words, the proposed balanced random-crop successfully relieve the problem of biasing to background attributes.

In line 3, the collaborative ensembler is replaced by 4 depth-wise separable convolutional layers, the segmentation output network proposed in FEELVOS. Compared to our dynamic segmentation, the proposed collaborative ensembler performs 1.6% better due to larger receptive fields.

In line 4, the CFBI result drops to 73.3% without using the sequential training strategy, and thus shows the effectiveness of the strategy.

In line 5, the multi-local matching is replaced by the original matching with only a single window, leading to a performance drop of 1.1%. Our multi-local matching is more robust to various motion rates in videos than the original local matching.

In summary, we explore the necessity of each proposed component or method of CFBI.

4.4 Conclusion

In this chapter, we propose a Collaborative Foreground-Background Integration framework for modelling temporal information efficiently. It is critical to model temporal information of both the foreground and background regions. Moreover, both the pixel-level information and instance-level information are useful for improving the robustness of the framework. Besides, the proposed general transformation module, GCT, is efficient in modelling channel relations and is robust to various visual tasks and datasets. Finally, the proposed sequential training and balanced random-crop are simple and effective in learning better feature embedding.

Chapter 5

Future Works

In this thesis, I investigated efficient sequence modelling methods for both image content generation and video understanding. I demonstrated the importance of modelling the spatial sequence information in predicting large-scale scenes with high qualities for image content generation. However, the proposed framework's spatial sequence modelling is restricted in a fixed direction (the horizontal direction in our setting). It is valuable to explore how to efficiently learn sequence dependencies on both the vertical and horizontal directions with one single framework in future works. As to video understanding, we proved that the concept of foreground-background collaboration is critical in modelling more robust temporal sequence information in challenging video object segmentation. We hope such a simple yet effective concept will help promote the development of more video understanding tasks in the future. In theory, the concept of foreground-background collaboration should also be promising for some image tasks, which have the definition of foreground and background, such as few-shot segmentation and saliency detection.

BIBLIOGRAPHY

[1] M. ARJOVSKY, S. CHINTALA, AND L. BOTTOU, *Wasserstein gan*, arXiv preprint arXiv:1701.07875, (2017).

[2] N. BALLAS, L. YAO, C. PAL, AND A. COURVILLE, *Delving deeper into convolutional networks for learning video representations*, arXiv preprint arXiv:1511.06432, (2015).

[3] M. BERTALMIO, G. SAPIRO, V. CASELLES, AND C. BALLESTER, *Image inpainting*, in SIGGRAPH, ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424.

[4] L. BERTINETTO, J. VALMADRE, J. F. HENRIQUES, A. VEDALDI, AND P. H. TORR, *Fully-convolutional siamese networks for object tracking*, in ECCV, Springer, 2016, pp. 850–865.

[5] T. BLUCHE, *Joint line segmentation and transcription for end-to-end handwritten paragraph recognition*, in NIPS, 2016.

[6] S. CAELLES, K.-K. MANINIS, J. PONT-TUSET, L. LEAL-TAIXÉ, D. CREMERS, AND L. VAN GOOL, *One-shot video object segmentation*, in CVPR, 2017, pp. 221–230.

[7] C. CAO, X. LIU, Y. YANG, Y. YU, J. WANG, Z. WANG, Y. HUANG, L. WANG, C. HUANG, W. XU, ET AL., *Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks*, in ICCV, 2015.

[8] Y. CAO, J. XU, S. LIN, F. WEI, AND H. HU, *Gcnet: Non-local networks meet squeeze-excitation networks and beyond*, in ICCV Workshops, 2019, pp. 0–0.

[9] L.-C. CHEN, G. PAPANDREOU, I. KOKKINOS, K. MURPHY, AND A. L. YUILLE, *Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs*, TPAMI, 40 (2017), pp. 834–848.

[10] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, *Encoder-decoder with atrous separable convolution for semantic image segmentation*, in ECCV, 2018, pp. 801–818.

[11] Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool, *Blazingly fast video object segmentation with pixel-wise metric learning*, in CVPR, 2018, pp. 1189–1198.

[12] J. Cheng, Y.-H. Tsai, W.-C. Hung, S. Wang, and M.-H. Yang, *Fast and accurate online video object segmentation via tracking parts*, in CVPR, 2018, pp. 7415–7424.

[13] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, *Global contrast based salient region detection*, TPAMI, 37 (2014), pp. 569–582.

[14] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, *Learning phrase representations using rnn encoder-decoder for statistical machine translation*, in EMNLP, 2014.

[15] F. Chollet, *Xception: Deep learning with depthwise separable convolutions*, in CVPR, 2017, pp. 1251–1258.

[16] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, *Lip reading sentences in the wild*, in CVPR, 2017.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, *Imagenet: A large-scale hierarchical image database*, in CVPR, Ieee, 2009, pp. 248–255.

[18] E. L. Denton, S. Chintala, R. Fergus, et al., *Deep generative image models using aÔø$^{\text{o}}$ laplacian pyramid of adversarial networks*, in NIPS, 2015, pp. 1486–1494.

[19] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, *Long-term recurrent convolutional networks for visual recognition and description*, in CVPR, 2015, pp. 2625–2634.

[20] X. Dong, Y. Yan, W. Ouyang, and Y. Yang, *Style aggregated network for facial landmark detection*, in CVPR, June 2018, pp. 379–388.

[21] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, *The pascal visual object classes (voc) challenge*, IJCV, 88 (2010), pp. 303–338.

[22] C. FEICHTENHOFER, H. FAN, J. MALIK, AND K. HE, *Slowfast networks for video recognition*, in ICCV, 2019, pp. 6202–6211.

[23] I. GOODFELLOW, J. POUGET-ABADIE, M. MIRZA, B. XU, D. WARDE-FARLEY, S. OZAIR, A. COURVILLE, AND Y. BENGIO, *Generative adversarial nets*, in NIPS, 2014, pp. 2672–2680.

[24] A. GRAVES, *Generating sequences with recurrent neural networks*, arXiv preprint arXiv:1308.0850, (2013).

[25] K. GREGOR, I. DANIHELKA, A. GRAVES, D. REZENDE, AND D. WIERSTRA, *Draw: A recurrent neural network for image generation*, in ICML, PMLR, 2015, pp. 1462–1471.

[26] I. GULRAJANI, F. AHMED, M. ARJOVSKY, V. DUMOULIN, AND A. C. COURVILLE, *Improved training of wasserstein gans*, in NIPS, 2017, pp. 5767–5777.

[27] B. HARIHARAN, P. ARBELÁEZ, L. BOURDEV, S. MAJI, AND J. MALIK, *Semantic contours from inverse detectors*, in ICCV, IEEE, 2011, pp. 991–998.

[28] K. HE, G. GKIOXARI, P. DOLLÁR, AND R. GIRSHICK, *Mask r-cnn*, in ICCV, 2017.

[29] K. HE, X. ZHANG, S. REN, AND J. SUN, *Deep residual learning for image recognition*, in CVPR, 2016.

[30] ——, *Identity mappings in deep residual networks*, in ECCV, Springer, 2016, pp. 630–645.

[31] M. HEUSEL, H. RAMSAUER, T. UNTERTHINER, B. NESSLER, AND S. HOCHREITER, *Gans trained by a two time-scale update rule converge to a local nash equilibrium*, in NIPS, 2017.

[32] S. HOCHREITER AND J. SCHMIDHUBER, *Long short-term memory*, Neural computation, 9 (1997), pp. 1735–1780.

[33] J. HU, L. SHEN, AND G. SUN, *Squeeze-and-excitation networks*, in CVPR, 2018.

[34] Y.-T. HU, J.-B. HUANG, AND A. G. SCHWING, *Videomatch: Matching based video object segmentation*, in ECCV, 2018, pp. 54–70.

[35] Y. HUANG, Y. CHENG, D. CHEN, H. LEE, J. NGIAM, Q. V. LE, AND Z. CHEN, *Gpipe: Efficient training of giant neural networks using pipeline parallelism*, arXiv preprint arXiv:1811.06965, (2018).

[36] S. IIZUKA, E. SIMO-SERRA, AND H. ISHIKAWA, *Globally and locally consistent image completion*, TOG, 36 (2017), p. 107.

[37] S. IOFFE AND C. SZEGEDY, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, in ICML, 2015.

[38] P. ISOLA, J.-Y. ZHU, T. ZHOU, AND A. A. EFROS, *Image-to-image translation with conditional adversarial networks*, in CVPR, 2017.

[39] M. JADERBERG, K. SIMONYAN, A. ZISSERMAN, ET AL., *Spatial transformer networks*, in NIPS, 2015.

[40] S. JI, W. XU, M. YANG, AND K. YU, *3d convolutional neural networks for human action recognition*, TPAMI, 35 (2012), pp. 221–231.

[41] J. JOHNANDER, M. DANELLJAN, E. BRISSMAN, F. S. KHAN, AND M. FELSBERG, *A generative appearance model for end-to-end video object segmentation*, in CVPR, 2019, pp. 8953–8962.

[42] W. KAY, J. CARREIRA, K. SIMONYAN, B. ZHANG, C. HILLIER, S. VIJAYANARASIMHAN, F. VIOLA, T. GREEN, T. BACK, P. NATSEV, ET AL., *The kinetics human action video dataset*, arXiv preprint arXiv:1705.06950, (2017).

[43] D. P. KINGMA AND J. BA, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, (2014).

[44] J. KOPF, W. KIENZLE, S. DRUCKER, AND S. B. KANG, *Quality prediction for image completion*, TOG, 31 (2012), p. 131.

[45] T.-Y. LIN, M. MAIRE, S. BELONGIE, J. HAYS, P. PERONA, D. RAMANAN, P. DOLLÁR, AND C. L. ZITNICK, *Microsoft coco: Common objects in context*, in ECCV, Springer, 2014.

[46] J. LUITEN, P. VOIGTLAENDER, AND B. LEIBE, *Premvos: Proposal-generation, refinement and merging for video object segmentation*, in ACCV, 2018, pp. 565–580.

[47] K. N. NGAN AND H. LI, *Video segmentation and its applications*, Springer Science & Business Media, 2011.

[48] S. W. OH, J.-Y. LEE, N. XU, AND S. J. KIM, *Video object segmentation using space-time memory networks*, in ICCV, 2019.

[49] S. OSHER, M. BURGER, D. GOLDFARB, J. XU, AND W. YIN, *An iterative regularization method for total variation-based image restoration*, Multiscale Modeling & Simulation, 4 (2005), pp. 460–489.

[50] D. PATHAK, P. KRAHENBUHL, J. DONAHUE, T. DARRELL, AND A. A. EFROS, *Context encoders: Feature learning by inpainting*, in CVPR, 2016, pp. 2536–2544.

[51] F. PERAZZI, A. KHOREVA, R. BENENSON, B. SCHIELE, AND A. SORKINE-HORNUNG, *Learning video object segmentation from static images*, in CVPR, 2017, pp. 2663–2672.

[52] F. PERAZZI, J. PONT-TUSET, B. MCWILLIAMS, L. VAN GOOL, M. GROSS, AND A. SORKINE-HORNUNG, *A benchmark dataset and evaluation methodology for video object segmentation*, in CVPR, 2016, pp. 724–732.

[53] J. PONT-TUSET, F. PERAZZI, S. CAELLES, P. ARBELÁEZ, A. SORKINE-HORNUNG, AND L. VAN GOOL, *The 2017 davis challenge on video object segmentation*, arXiv preprint arXiv:1704.00675, (2017).

[54] A. RADFORD, L. METZ, AND S. CHINTALA, *Unsupervised representation learning with deep convolutional generative adversarial networks*, arXiv preprint arXiv:1511.06434, (2015).

[55] O. RONNEBERGER, P. FISCHER, AND T. BROX, *U-net: Convolutional networks for biomedical image segmentation*, in MICCAI, Springer, 2015, pp. 234–241.

[56] O. RUSSAKOVSKY, J. DENG, H. SU, J. KRAUSE, S. SATHEESH, S. MA, Z. HUANG, A. KARPATHY, A. KHOSLA, M. BERNSTEIN, ET AL., *Imagenet large scale visual recognition challenge*, IJCV, 115 (2015), pp. 211–252.

[57] T. SALIMANS, I. GOODFELLOW, W. ZAREMBA, V. CHEUNG, A. RADFORD, AND X. CHEN, *Improved techniques for training gans*, in NIPS, 2016.

[58] P. SANGKLOY, J. LU, C. FANG, F. YU, AND J. HAYS, *Scribbler: Controlling deep image synthesis with sketch and color*, in CVPR, vol. 2, 2017.

[59] J. SHI, Q. YAN, L. XU, AND J. JIA, *Hierarchical image saliency detection on extended cssd*, TPAMI, 38 (2015), pp. 717–729.

[60] K. SIMONYAN AND A. ZISSERMAN, *Very deep convolutional networks for large-scale image recognition*, in ICLR, 2015.

[61] B. SINGH, M. NAJIBI, AND L. S. DAVIS, *Sniper: Efficient multi-scale training*, in NIPS, 2018, pp. 9333–9343.

[62] J. SIVIC, B. KANEVA, A. TORRALBA, S. AVIDAN, AND W. T. FREEMAN, *Creating and exploring a large photorealistic virtual space*, (2008).

[63] N. SRIVASTAVA, E. MANSIMOV, AND R. SALAKHUDINOV, *Unsupervised learning of video representations using lstms*, in ICML, PMLR, 2015, pp. 843–852.

[64] I. SUTSKEVER, O. VINYALS, AND Q. V. LE, *Sequence to sequence learning with neural networks*, in NIPS, 2014.

[65] C. SZEGEDY, W. LIU, Y. JIA, P. SERMANET, S. REED, D. ANGUELOV, D. ERHAN, V. VANHOUCKE, AND A. RABINOVICH, *Going deeper with convolutions*, in ICCV, 2015.

[66] C. SZEGEDY, V. VANHOUCKE, S. IOFFE, J. SHLENS, AND Z. WOJNA, *Rethinking the inception architecture for computer vision*, in CVPR, 2016.

[67] A. TORRALBA, *Contextual priming for object detection*, IJCV, 53 (2003), pp. 169–191.

[68] D. TRAN, L. BOURDEV, R. FERGUS, L. TORRESANI, AND M. PALURI, *Learning spatiotemporal features with 3d convolutional networks*, in ICCV, 2015, pp. 4489–4497.

[69] A. VAN OORD, N. KALCHBRENNER, AND K. KAVUKCUOGLU, *Pixel recurrent neural networks*, in ICML, PMLR, 2016, pp. 1747–1756.

[70] P. VOIGTLAENDER, Y. CHAI, F. SCHROFF, H. ADAM, B. LEIBE, AND L.-C. CHEN, *Feelvos: Fast end-to-end embedding learning for video object segmentation*, in CVPR, 2019, pp. 9481–9490.

[71] P. VOIGTLAENDER AND B. LEIBE, *Online adaptation of convolutional neural networks for video object segmentation*, in BMVC, 2017.

[72] P. Voigtlaender, J. Luiten, and B. Leibe, *Boltvos: Box-level tracking for video object segmentation*, arXiv preprint arXiv:1904.04552, (2019).

[73] M. Wang, Y.-K. Lai, Y. Liang, R. R. Martin, and S.-M. Hu, *Biggerpicture: Data-driven image extrapolation using graph matching*, ACM Trans. Graph., 33 (2014), pp. 173:1–173:13.

[74] X. Wang, R. Girshick, A. Gupta, and K. He, *Non-local neural networks*, in CVPR, 2018.

[75] Y. Wu and K. He, *Group normalization*, in ECCV, 2018.

[76] S. Wug Oh, J.-Y. Lee, K. Sunkavalli, and S. Joo Kim, *Fast video object segmentation by reference-guided mask propagation*, in CVPR, 2018, pp. 7376–7385.

[77] H. Xiao, J. Feng, G. Lin, Y. Liu, and M. Zhang, *Monet: Deep motion exploitation for video object segmentation*, in CVPR, 2018, pp. 1140–1148.

[78] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, *Sun database: Large-scale scene recognition from abbey to zoo*, in CVPR, IEEE, 2010, pp. 3485–3492.

[79] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, *Aggregated residual transformations for deep neural networks*, in CVPR, 2017.

[80] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, *Show, attend and tell: Neural image caption generation with visual attention*, in ICML, 2015.

[81] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang, *Youtube-vos: A large-scale video object segmentation benchmark*, arXiv preprint arXiv:1809.03327, (2018).

[82] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, *High-resolution image inpainting using multi-scale neural patch synthesis*, in CVPR, vol. 1, 2017, p. 3.

[83] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos, *Efficient video object segmentation via network modulation*, in CVPR, 2018, pp. 6499–6507.

[84] F. YU AND V. KOLTUN, *Multi-scale context aggregation by dilated convolutions*, arXiv preprint arXiv:1511.07122, (2015).

[85] J. YU, Z. LIN, J. YANG, X. SHEN, X. LU, AND T. S. HUANG, *Generative image inpainting with contextual attention*, in CVPR, 2018, pp. 5505–5514.

[86] J. YUE-HEI NG, M. HAUSKNECHT, S. VIJAYANARASIMHAN, O. VINYALS, R. MONGA, AND G. TODERICI, *Beyond short snippets: Deep networks for video classification*, in CVPR, 2015, pp. 4694–4702.

[87] Y. ZHANG, J. XIAO, J. HAYS, AND P. TAN, *Framebreak: Dramatic image extrapolation by guided shift-maps*, in CVPR, 2013, pp. 1171–1178.

[88] Z. ZHANG, S. FIDLER, AND R. URTASUN, *Instance-level segmentation for autonomous driving with deep densely connected mrfs*, in CVPR, 2016, pp. 669–677.

[89] Q. ZHOU, Z. HUANG, L. HUANG, Y. GONG, H. SHEN, W. LIU, AND X. WANG, *Motion-guided spatial time attention for video object segmentation*, in ICCV Workshops, 2019.

[90] Z. ZHOU, L. REN, P. XIONG, Y. JI, P. WANG, H. FAN, AND S. LIU, *Enhanced memory network for video segmentation*, in ICCV Workshops, 2019.

[91] J.-Y. ZHU, T. PARK, P. ISOLA, AND A. A. EFROS, *Unpaired image-to-image translation using cycle-consistent adversarial networks*, arXiv preprint, (2017).

[92] X. ZHU, Y. XIONG, J. DAI, L. YUAN, AND Y. WEI, *Deep feature flow for video recognition*, in CVPR, 2017, pp. 2349–2358.