Deep Interactive Image Matting

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Marco Forte

Department of Electronic and Electrical Engineering
Trinity College Dublin

This dissertation is submitted for the degree of
Doctor of Philosophy

March 2022

I would like to dedicate this thesis to my loving parents, family and friends.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Marco Forte
March 2022

Acknowledgements

It takes a village to raise a child. It took me an incredible academic, familial and fraternal support structure to raise this thesis. Prospective students reading this: do not hesitate to reach out to those around you for support, and professors: do not hesitate to offer it and encourage the student seek it everywhere they can.

First and foremost, I would like to express my deep gratitude to my advisor, Professor François Pitié for his support and patience. I was given freedom to persue the different research directions I was interested in. Many interesting technical and non-technical discussions were had throughout the years and it really helped me develop as a researcher. Additionally, Professor Anil Kokaram of Trinity College, and Dr.s Scott Cohen, Brian Price and Ning Xu of Adobe research all provided fruitful mentorship at various times throughout my PhD. This collaboration between academics is one I highly valued.

My dear friends, girlfriend, and family were essential sources of rest, play and chat between papers. Thanks to them for always coming to my side when I needed them most. Lastly, despite their inanimacy, I would be remiss not to give a nod to my surfboard, climbing gym and beatiful Irish hills.

Abstract

Image Matting for Compositing is the cutting of an object from an image for background replacement. It is an interactive process fundamental to image editing. Useful matting algorithms both reduce the amount of interaction necessary and accurately compute transparency and foreground colours in difficult cases like translucency and thin structures like hair. In 2016, the first deep-learning based matting methods were proposed with significant accuracy improvements over previous techniques. Despite excellent accuracy on difficult images, they fail to work well without copious user input even in simple cases where classic algorithms succeed. Concurrently, the first deep-learning based interactive binary-matting(segmentation) methods were proposed and successful in reducing the amount of user input necessary for a rough selection. Although these are practical for some simple images, they are not capable of real matting *i.e* finely detailed non-binary selections. In this thesis, we develop a novel deep learning-based matting algorithm which combines the strengths of interactive segmentation and matting. Specifically, we propose 'interactivity focused' training and architecture for deep interactive segmentation methods to allow for more finely detailed user edits and selections. For matting, we modify our segmentation architecture to allow for accurate joint alpha and foreground colour prediction. Finally, we propose a method to finetune our matting algorithm such that it retains its accuracy but also is capable of quality selections with far less user input. Our interactive segmentation, matting and interactive matting algorithms achieve state-of-the-art performance on various challenging benchmarks, demonstrating their effectiveness for practical applications.

Table of contents

1 Introduction **1**

1.1 Key observations and research questions . . . 3
1.2 Why deep learning? . . . 7
1.3 Contributions . . . 7
1.4 Thesis Structure . . . 9
1.5 Publications . . . 10

2 Literature Review **13**

2.1 Interactive Segmentation . . . 13
2.1.1 Scribble based Algorithms using Deep Learning . . . 15
2.1.2 Segmentation Architectures . . . 16
2.1.3 Embedding User Interactions . . . 17
2.1.4 Training Scheme . . . 19
2.1.5 Non-Scribble based Segmentation . . . 20
2.1.6 Binary Segmentation for Compositing . . . 20
2.2 Matting . . . 21
2.2.1 Trimap based Algorithms using Deep Learning . . . 22
2.2.2 Non-Trimap based Matting . . . 27
2.3 Interactive Matting . . . 29
2.4 Measuring Success . . . 31
2.4.1 Datasets . . . 31
2.4.2 Metrics . . . 34
2.5 Conclusion . . . 37

3 Deep Interactive Training and Architecture for Object Selection in Images **39**

3.1 Introduction . . . 39
3.2 Proposed Approach . . . 42
3.2.1 An Interactive Hi-Resolution Network Architecture . . . 42

3.2.2 Training . . . 45
3.3 Evaluation . . . 48
3.3.1 Benchmarks . . . 48
3.3.2 Comparison to the State of the Art . . . 48
3.3.3 Ablation Study . . . 49
3.3.4 Qualitative Evaluation . . . 50
3.4 Training on a Synthetic Dataset . . . 51
3.4.1 Synthetic Dataset Construction . . . 51
3.4.2 Synthetic Dataset Training Results . . . 52
3.5 Discussion . . . 53
3.6 Conclusion . . . 53

4 Deep joint foreground, background estimation and alpha matting 69
4.1 Introduction . . . 70
4.2 Related Works . . . 72
4.3 Proposed Approach . . . 74
4.3.1 Network Architecture . . . 74
4.3.2 Batch Normalisation vs. Group Normalisation . . . 76
4.3.3 F, B, α Losses . . . 76
4.3.4 F, B, α Fusion . . . 76
4.3.5 Training Dataset and Modifications . . . 78
4.3.6 Training Details . . . 78
4.3.7 Test Time Augmentation (TTA) . . . 78
4.4 Experimental Results . . . 79
4.4.1 Evaluating the Alpha Loss Functions . . . 79
4.4.2 Batch-Size and BatchNorm vs. GroupNorm . . . 80
4.4.3 Evaluating the Impact of Jointly Estimating F, B, α . . . 81
4.4.4 Comparison with Other Works . . . 81
4.5 Conclusion . . . 84

5 Deep Interactive Image Matting 91
5.1 Introduction . . . 91
5.1.1 The Issue with Interactivity in Deep Matting Methods . . . 91
5.1.2 Unifying Deep Interactive Segmentation and Matting to Achieve Interactive Matting . . . 93
5.1.3 Interactive Segmentation . . . 94
5.2 Proposed Interactive Matting Method . . . 95

5.2.1 An Interactive Matting Network . . . 95
5.2.2 A Multi-Task Interactive Matting Training Strategy . . . 96
5.3 Results . . . 98
5.3.1 Interactive Segmentation Benchmarks . . . 99
5.3.2 Trimap-Based Matting Benchmarks . . . 101
5.3.3 Interactive Matting Benchmark . . . 101
5.3.4 Qualitative Assessment . . . 104
5.4 Limitations . . . 104
5.4.1 The matting model . . . 104
5.4.2 The datasets and evaluation . . . 109
5.5 Conclusion . . . 109

6 Conclusion **111**
6.1 Summary of contributions . . . 111
6.2 Closing Remarks . . . 112
6.2.1 Thesis impact and contemporary works . . . 113

References **115**

Chapter 1

Introduction

We alter every photograph we take on digital cameras. Perhaps unknowingly through automated JPEG compression, but often manually and selectively to specific regions in an attempt to improve the image. What to do with a selected region is up to artistic preference, but making the selection is a classic problem in computer vision. Everyone from casual Snapchat-er to professional Photoshop-er appreciates new algorithms which facilitate higher quality selections with less effort.

The history of applications and algorithms for selecting specific image regions is long and diverse. The earliest applications came from Film in the late 1800s when cinematographers wanted to combine two shots into one [49]. Since then, greenscreen keying [110, 140] and rotoscoping have become the de facto way to separate a foreground shot to combine with a different background. The problem of automatically extracting a foreground shot from an arbitrary background is known as *matting* and has been studied since the 2000s [34]. It is crucial that the combined shot looks realistic and professional for media applications, and thus pixel-accurate mattes are required. Developed simultaneously were many applications that required automatic labeling of each pixel in an image, such as medical imaging, surveillance, satellite surveying, and robot automation. This problem is known as *segmentation*.

This thesis presents a novel state-of-the-art algorithm capable of user-assisted image matting. We build on learnings from both segmentation and matting to solve this problem. This requires a deep understanding of each and the relationship between them. Throughout this thesis, we analyse this relationship from three perspectives, 1) the objectives of segmentation and matting, 2) the algorithms and theory underlying the existing approaches, 3) the resulting behavior of tools from each approach.

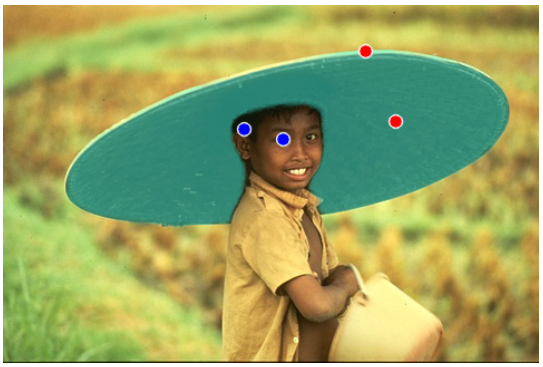


Fig. 1.1 Example of interactive segmentation with clicks.

Segmentation. *Segmentation* is a pixel labelling problem in computer vision. The ways humans recognise what forms part of the same object has been studied in psychology under the Gestalt laws of grouping [11]. In the research, people knowingly applied Gestalt theory to segmentation since the '70s [35]. The problem is formulated as a binary or multi-class classification, one for each pixel. In the early '80s, Geman and Geman [52] formalised the combination of Bayesian likelihood and priors into a single Markov Random Field (MRF) framework. The likelihood is the probability of a given pixel belonging to a class, which applies the principle of similarity, *e.g.* blue pixels form part of the sky. Priors were used to reason about the location of the boundary by encouraging adjacent pixels to have similar labelling (Continuity/Closure/Connectedness) [73]. In 2001 [16] a popular interactive segmentation algorithm using these principles was developed. The user would mark a subset of pixels as either foreground or background, these would be used to determine the likelihoods for all the other pixels, and the spatial priors were used to have a coherent solution, see Figure 1.1 In Chapter 2 we further discuss the research into user-guided interactive segmentation.

Matting. *Matting* is the problem of finding the transparency value at each pixel. Matting is a key step in *compositing* which is the combination of at least two shots to produce an integrated result [18]. Binary segmentation algorithms are not sufficient for compositing; we illustrate this in Figure 1.2. There are two reasons for this. Firstly, even minor segmentation inaccuracies around the boundary lead to an unrealistic composite. This is because segmentation algorithms are generally less concerned with boundary accuracy than rough region accuracy. Secondly, transparency is needed because parts of the object that are transparent in the original image need to also be transparent in the composite. This need for transparency is not restricted to wholly see-through materials like glass. All objects in an image contain some level of transparency along their boundary, a result of the image formation process [126]. The mathematical model

Image

Interactive Segmentation Composite

Matting Composite

Fig. 1.2 Illustration of composites onto a white background produced using interactive binary segmentation and matting. (Photo by Marco Xu on Unsplash)

of this transparency is known as the Compositing Equation [18] and is defined as follows.

$$C_i = \alpha_i F_i + (1 - \alpha_i) B_i \tag{1.1}$$

In this equation, C_i is the observed colour value at pixel site i, F_i, B_i are the pixel colours in the foreground and background layers at that same site, and α_i is the level of mixing between the two layers. An α value of 1 thus indicates a pure foreground pixel, a value of 0 indicates a pure background pixel.

A well-known early approach to matting is the bluescreen and greenscreen. Here analog technology could separate the foreground from the solid background screen. Later digital algorithms [110] were developed based on colour differences and clustering. In 2001 Chuang *et al.* [34] developed a successful Bayesian modeling approach to matting in the case of arbitrary backgrounds. Research in this area has found applications in image editing tools like Photoshop [98].

Deep learning for segmentation and matting. There has been a recent surge in accuracy for segmentation and matting due to the development of deep learning based approaches. Indeed the phenomenon is common to almost all computer vision and image processing these days. Deep learning approaches in segmentation and matting are almost exclusively discussed over classic methods, as they exceed in almost all standard benchmarks. However, we observed that some key aspects were overlooked.

1.1 Key observations and research questions

We made three key observations that sparked some initial questions which were then refined to our three research questions.

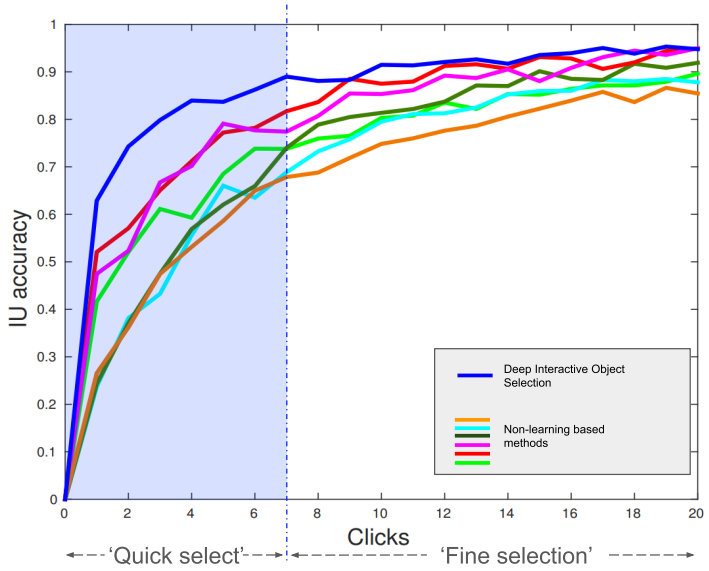


Fig. 1.3 Number of clicks vs. mIoU accuracy for *Deep Interactive Object Selection* [162], and non-learning based methods (top to bottom [10, 16, 55, 59, 59, 148]) on the GrabCut [131] dataset. Note the plateau after initial clicks and how non-learning based methods close the gap. All methods finish around 90% mIoU accuracy which is below what is acceptable for compositing.

What is the interactive behaviour of deep learning models for segmentation and matting? The first key observation we make is that deep learning based methods for interactive segmentation and for matting struggle with user interactivity. This is observed for interactive segmentation by incrementally adding user input and seeing that in most cases the algorithm reaches a plateau of accuracy. A level that is still below what is desirable if the output were to be used for compositing, see Figure 1.2. This can be observed from the metrics in the original Deep Interactive Image segmentation paper [162]. In Figure. 1.3, we highlight that after 10 user inputs, non-deep learning methods reach the same or better accuracy. Details on the mIoU accuracy metric are given in Chapter 2, for now it suffices to say that it is a measure of overlap of the predicted and ground truth mask, and the max value is 1.0.

In the case of matting, the deep learning algorithms exceed when given copious user input, yet they fail drastically when given small amounts of input, see Figure 1.5. In Chapter 5, will explore the differences between classic matting methods and deep learning based ones for sparse user input.

Fig. 1.4 Compositing a complex object with the ground-truth alpha matte, and the original image, and with the foreground colours. The image is from alphamatting.com [129].

Research Question 1: *Can we better handle all levels of user interaction in interactive segmentation networks?* Making selections in minimal time and effort is valuable in industry. It increases throughput and saves money, and improves the user experience. Typically, interactive selection tools require the user to mark some pixels as either definite *foreground* or definite *background*. Thus one formal definition of a minimal effort selection is the minimum number of pixels required to be marked for the algorithm to give a prediction that is $x\%$ accurate. The main objective of deep-learning research for interactive segmentation has been minimal effort rough selections, *i.e* $x = 90\%$. However, as alluded to above, once sufficient input is given to produce a rough selection, these algorithms fail to give improved predictions with more input. We thus measure our answer to this research question in the number of user interactions to reach a $x = 99\%$ accurate mask.

What is the importance of foreground estimation in deep learning based image matting models? Our second key observation is that estimation of the foreground colour, F, has not been thoroughly considered by deep learning algorithms for matting. They either solely estimate alpha and rely on a computationally and memory intensive post proccessing [87], or rely on a separate decoder solely for the foreground [67]. We can see in the composite inlays of Figure 1.4 how foreground colour estimation is is crucial for high-quality composites. We seek to investigate further the role of foreground prediction in deep learning based image matting.

Research Question 2: *Using deep learning can we jointly estimate the alpha and foreground and background colours?* In Chapter 4, we present a novel deep-learning based algorithm for joint foreground, background, and alpha estimation. A key measure of success here is empirical and qualitative comparison to alternative matting methods which estimate either alpha alone [160], or alpha and foreground but not background [67]. Among other contributions, we propose a post-processing method that improves the predictions when a joint estimate of alpha, foreground, and background is made.

What is the relationship between deep learning for interactive segmentation and matting? Our final key observation comes from comparing the deep learning focused literature for interactive matting and segmentation. Both fields have used very distinct network architectures, training and evaluation datasets, loss functions, and evaluation metrics. This is despite the significant overlap in these problems, and that indeed some older non deep learning based methods did take a more unified approach.

Research Question 3: *Can we develop a unified deep learning based interactive segmentation and matting algorithm?*

Deep learning approaches to interactive segmentation have sought to minimise the number of interactions necessary for a rough binary selection. Deep learning approaches to matting algorithms have sought to maximise the selection quality, including transparency, given plenty of user input.

These deep learning based methods perform far beyond classic approaches on their specific criteria. Interestingly, however, these deep algorithms do not generalise to each other, unlike classic approaches. That is to say, deep interactive segmentation algorithms are not capable of high-quality selections given more input, and deep matting algorithms cannot generate rough selections given minimal input. We give an example of this in Figure 1.5. Selection tools need this responsivity to user input, *i.e* rough selections with minimal input, then increasingly accurate selections with transparency, ideally up to a perfect selection.

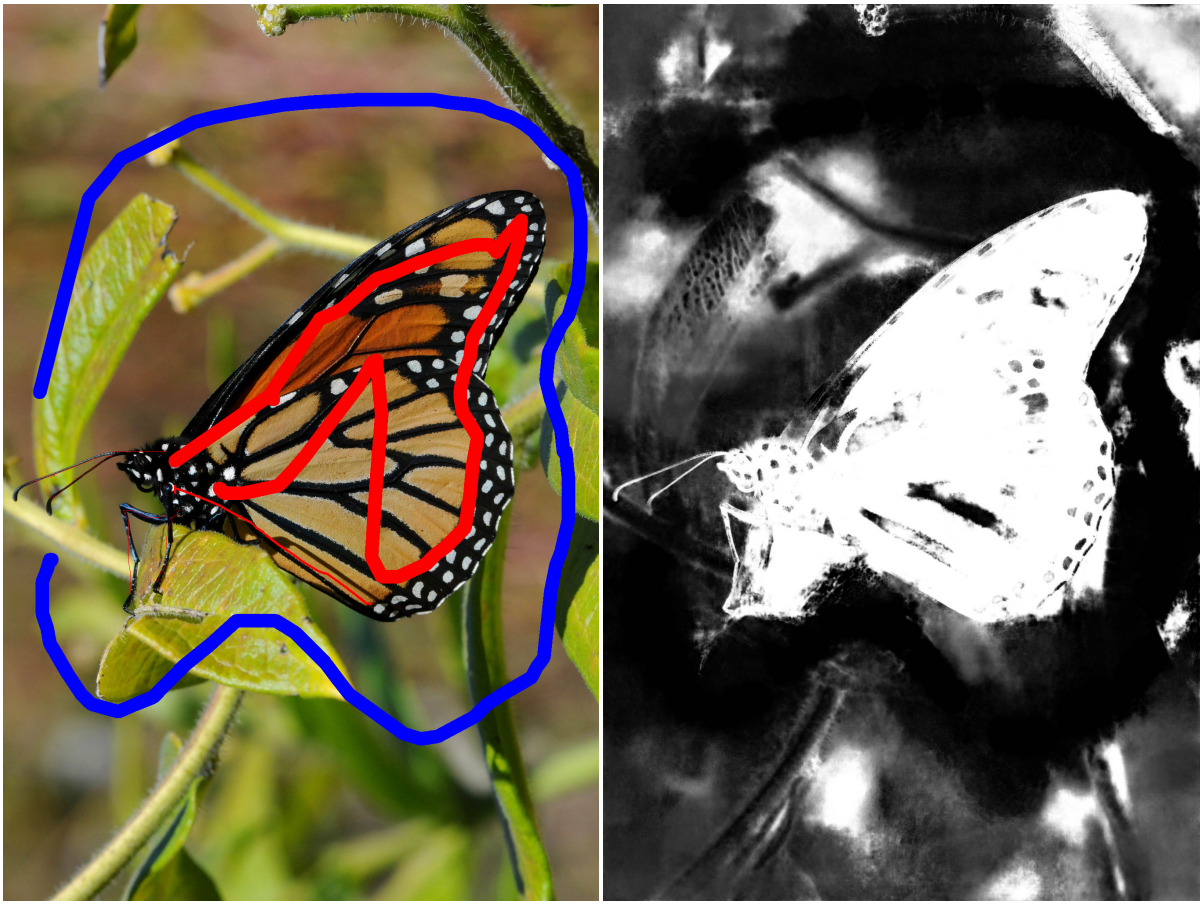
Our main contribution is a deep-learning based interactive matting algorithm that provides state-of-the-art selections for all amounts of input. This algorithm is far more suitable for image editing tools than existing methods. This is due to contributions in the training and architecture, which we designed around the interactive selection process and joint estimation of transparency and foreground colour for compositing. We take an approach of investigating both Interactive Segmentation and Matting individually, correcting and identifying fundamental shortcomings, then finally combining these learnings.

1.2 Why deep learning?

We approach interactive segmentation and matting using deep learning [54]. There are multiple reasons why choose to use deep learning to develop an interactive matting system. The first and most important is that interactive matting is a complex problem requiring significant extrapolation from limited user input on a wide range of possible images. Previous approaches on interactive matting were limited to making inferences based on colours, gradients, and textures in the image [58, 87, 128]. In practice, this approach is very brittle for real-world images and will fail on ill-defined edges, varying colours, and complex textures. Deep learning is needed due to its ability to parse images on a semantic level [136, 138].
There are some tradeoffs between deep learning and classic approaches. First, for classic approaches, their behavior is mathematically guaranteed, and deep learning approaches can fail in unpredictable ways without warning, and it is challenging to guarantee particular behavior. Take, for example, Closed-Form Matting [87], where if the image follows certain constraints, then the algorithm is guaranteed to produce the correct answer. In this situation, Closed-Form Matting is preferable over deep learning approaches if accuracy is desired. A second limitation of using deep learning algorithms for matting is the potential for bias and unfairness [109]. This is especially true if training and evaluating with a biased dataset. The work in this thesis uses mainly pre-established datasets and serves as a proof of concept. More care is needed to avoid bias and ensure fairness for production use.

1.3 Contributions

This thesis makes several contributions to address our three research questions. Below we highlight significant contributions from each chapter.



Input Scribbles | Deep Image Matting [160]

Fig. 1.5 Example of using scribbles as an input to Deep Image Matting [160]. The hand drawn scribbles are overlaid in red=definite foreground and blue=definite background.

Research Question 1 - Chapter 3. The first significant contribution of this thesis is to propose a deep-learning based method for interactive segmentation, which is modelled on the interactive process of making a selection. Sub-contributions of this method include a novel interactive training algorithm, network interactive architecture, and a new synthetic training dataset. With this method, good selections can be made by clicking on just a handful of pixels, and the algorithm incorporates a higher number of interactions as necessary to make near-perfect selections.

Research Question 2 - Chapter 4. The second major contribution of this thesis is a deep image matting algorithm that jointly estimates an object's alpha alongside the foreground and background colours. To achieve this, we make a novel modification of the network architecture to predict foreground colours without any memory or computational cost. Furthermore, we design appropriate novel loss functions for joint prediction. In addition, we make a novel observation and improvement to the training data generation to avoid mislabeling. Finally, we also design a novel post-processing technique that improves joint alpha, foreground, and background estimation results.

Research Question 3 - Chapter 5. The main contribution of this thesis is a deep interactive image matting algorithm. We are the first to tackle this problem. To achieve this, we use a novel training scheme that begins with a pre-trained matting network and introduces the interactive segmentation task. Next, we apply interactive segmentation metrics and evaluation procedures to interactive matting. Finally, we evaluate our interactive matting model on both segmentation and matting tasks to test its ability to achieve state-of-the-art results for sparse and dense user input.

1.4 Thesis Structure

Here we give a summary of each technical chapter in the thesis.

Chapter 2 - Literature Review. Here we present the existing technical material and discussions that underly the ideas we develop throughout the thesis. First, we discuss the technical history of interactive segmentation, both classic and deep learning-based methods. For deep methods, we pay particular attention to how clicks are encoded, the training regime, and accuracy along the boundary. We also discuss the technical history of matting, paying close attention to the connection with compositing, model architecture, and interactive matting. Finally, the metrics and datasets for interactive segmentation matting are detailed and motivated.

Chapter 3 - Deep interactive training and architecture for object selection in images. In this technical chapter, we investigate and answer our first research question. We do this by identifying two causes of the accuracy plateau for existing deep interactive segmentation approaches. First, we motivate and detail our solution to this plateau. Next, we evaluate our contributions across three standard benchmark datasets using a variety of metrics. Finally, we record accuracy across a range of clicks to check that our model does not plateau.

Chapter 4 - Deep joint foreground, background estimation and alpha matting. In this chapter, we address our second research question. We begin by motivating our work with more visual examples of the need for foreground estimation when compositing. We also show how the current state of the art in foreground estimation performs in these examples. Next, we examine many aspects of the problem like the loss function, network architecture, data-augmentation, post-processing, and training regime. Finally, we evaluate our model on standard benchmark datasets for both alpha matte and composite quality. We find that our model exceeds the state of the art on all metrics. To conclude, we give visual examples of the composites with our method.

Chapter 5 - Deep interactive image matting. Our final chapter answers our third research question of unifying deep interactive segmentation and matting. We begin by discussing the interactivity deficiencies in existing deep matting models, giving examples of this, and explaining why we think it is happening. Next, we show how we can combine the work from our two previous chapters to build and train an interactive matting model. We use both interactive segmentation and matting datasets and metrics to evaluate our model. For the matting datasets, we evaluate them using both sparse and dense user input, showing how our model outperforms both matting and interactive segmentation models on each of these tasks. Finally, we give visual examples of our model compared to existing approaches in segmentation and matting and compare them to our models defined in previous chapters.

1.5 Publications

Three of our publications and e-prints are publicly available. This thesis contains all technical material from these articles, as well as a more detailed discussion:

- Chapter 3: Forte, M., Price, B., Cohen, S., Xu, N., and Pitié, F. (2020). Interactive training and architecture for deep object selection. In *IEEE International Conference on Multimedia and Expo (ICME)*

- Chapter 3 (Journal version): Forte, M., Price, B., Cohen, S., Xu, N., and Pitié, F. (2020). Getting to 99% accuracy in interactive segmentation. *arXiv preprint arXiv:2003.07932*
- Chapter 4: Forte, M. and Pitié, F. (2020). F, b, alpha matting. *arXiv preprint arXiv:2003.07711*

Chapter 2

Literature Review

This chapter presents technical material underlying the ideas that we will develop in the rest of this work. We begin by recalling the model for the transparency, α, of an object in an image, $C = \alpha F + (1-\alpha)B$, where C, F, B are the colours of the image, foreground, and background. The problem is to estimate α given the observed data. We focus on algorithms that best leverage user interactions to help solve the non-linear problem for α. In particular, our work bridges the divide between the special case that α is binary, known as *Segmentation*, and the general case, known as *Matting*. This review will show how these fields have progressed in parallel and led to different applications, algorithms, and results. We conclude by examining the recurring intersections of segmentation and matting in the literature to set the stage for our unification of them for deep learning in Chapter 5.

2.1 Interactive Segmentation

Image segmentation is the partitioning of the pixels in an image into two or more classes [61]. Transparency is, of course, an issue but that is solved using alpha-matting which we discuss later. In the binary case we have two segments foreground and background. Many of the algorithms in this class can be understood from a Bayesian standpoint. Consider that we wish to manipulate the posterior $p(\alpha|C)$ to choose the best binary α. Bayes' then allows us to express the problem in terms of measurable p.d.f's as follows.

$$p(\alpha|C) \propto p(C|\alpha)p(\alpha) \tag{2.1}$$

The likelihood $p(C|\alpha)$ measures how likely it is to observe the pixel C given $\alpha = 0$ or $\alpha = 1$. While the prior $p(\alpha)$ typically injects information about the smoothness of the binary segmentation. It is in the likelihood that example foreground and background colours can be incorporated

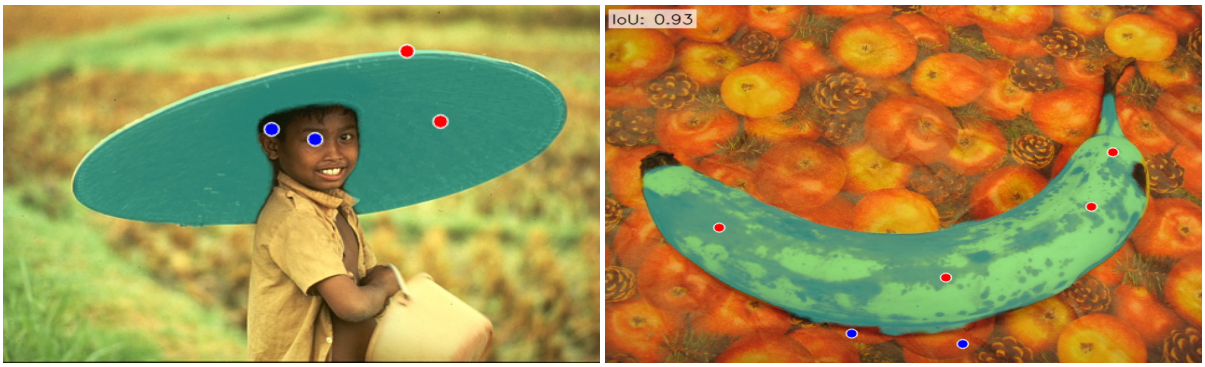


Fig. 2.1 Example interactive segmentation results.

to build p.d.f's for $p(C|\alpha = 0)$ and $p(C|\alpha = 1)$. These examples can be automatically chosen (as in video segmentation, areas that do not move are probably background) or interactively chosen.

A most famous early work in segmentation was *Otsu's method* [117](1979). It automatically selected a threshold to binarise a grayscale image based its histogram. Another early work *Segmentation of Solid Objects Using Global Local Edge Coincidence* [73] (1979) used image edges to find automatically find binary segmentations. Since the introduction of Markov Random Fields (MRFs) [52] in the '80s many popular methods for segmentation incorporate information from both image colour p.d.fs and image edges. Although at the beginning MRFs were too slow to be solved for interactive methods.

An early influential work for interactivity in segmentation was *Snakes: Active Contour Models* [81] (1988), here the user marked key points along the boundary of the object and a boundary-adherent path was automatically traced connecting them. This automatic-tracing approach was further refined in *Intelligent Scissors for Image Composition* [111] (1995) and was demonstrated as practical for colour images. A breakthrough in the fast solving of MRFs for interactive segmentation came with *Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images* [16] (2001). In this method, the user scribbles on foreground and background regions, and these scribbles are used to build p.d.fs of the foreground and background. Since then, many methods using MRFs have been developed and adopted for user image editing tools like photoshop [163].

"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts [131](2004) extended the Graph Cuts work by (among other things) modelling the foreground and background distributions by GMMs and a new more powerful iterative method of the optimisation. In 2010 two interesting methods used geodesic information and shape priors to extend the Graph Cuts work [59, 120]. The work of Liu *et al.* [98] (2009) is particularly interesting to us, as they acknowledge the progressive nature of image editing and pay particular attention to the artist

workflow. For instance, they make sure that changes to the mask stay localised to the user edit, so as not to undo progress. However as they acknowledge their work and previous ones have however poor performance on textured regions, and some images may require intensive user interaction just to get a rough selection. The is because they rely completely on the hand-crafted models proposed for likelihood and prior. Results shown by Xu *et al*. [162] indicate that these algorithms accuracy is much lower than the new DL approaches, which learn more complex image models.

2.1.1 Scribble based Algorithms using Deep Learning

While deep learning for computer vision has been around since the '80s [50], it has only recently been applied to interactive image segmentation [162] (2016). In this thesis, a high-level understanding from the reader of machine learning [15] and deep learning [54] is assumed. This thesis builds upon the great foundation of applying deep learning to semantic segmentation. So we begin by reviewing this preceding field.

The seminal application of deep learning to semantic segmentation was presented in 'Fully Convolutional Networks for Semantic Segmentation' [136] in 2014. The importance of this work is that it was the first to use a fully convolutional network architecture trained end-to-end on semantic segmentation. The convolutional network architecture, 'FCN', was a 16 layer mode (VGG16 [138]) where the final layer was replaced by one with 21 channels (one for each class) instead of the 1000 (the number of classes in ImageNet [39]), followed by a softmax activation to predict the most-likely label at each pixel. The network architecture is shown in Figure 2.2. The dataset used for training was Pascal-2011 [42], with 1112 images. Here we will cover the research on network architecture, but segmentation datasets are so important they deserve their own section 2.4.1.

It was not long before deep semantic segmentation models were adopted for interactive segmentation [162] (2016). Interactive segmentation training is different from semantic segmentation training in three key ways. Firstly, during training the target mask is just one instance from the semantic labelling of each image. Secondly, positive and negative clicks covering the object and background are generated for each instance. Finally, these clicks are fed as two additional inputs to the network. The first algorithm from Xu. *et al*. [162] showed a large reduction in the number of clicks necessary to reach 90% accuracy on benchmark datasets. Deep-learning methods train and evaluate with clicks on single pixels at a time, however when the user scribbles over multiple pixels at once this is simply interpreted by the network as multiple clicks. We refer to these as scribble based methods as clicks are a sub-case of scribbles.

In recent years, deep-learning based approaches to interactive segmentation (*e.g*. [76, 92, 162]) have superseded prior works [16, 81, 98, 111]. While the core architecture of

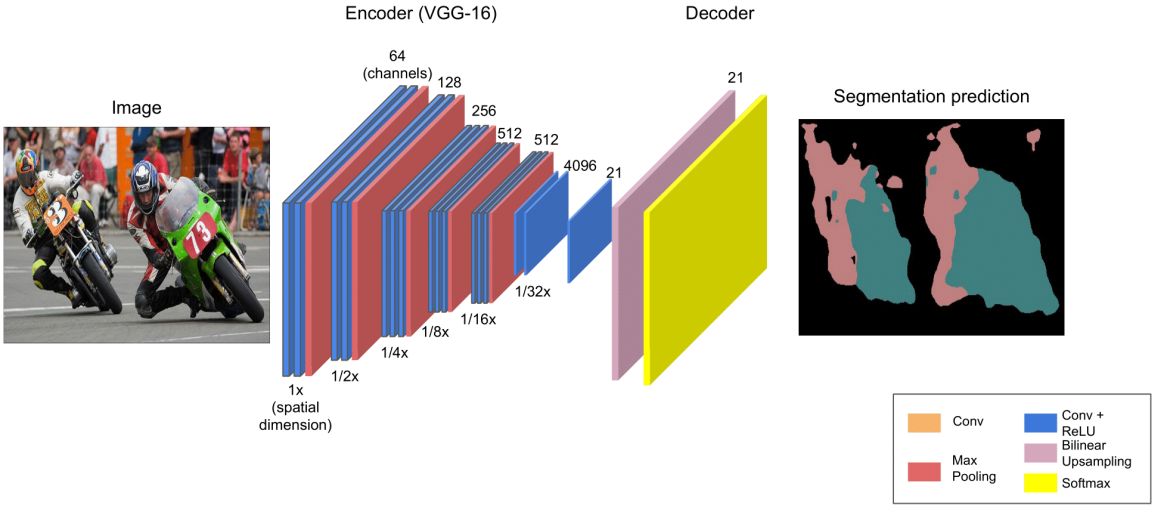


Fig. 2.2 *Fully Convolutional Networks for Semantic Segmentation* [136].

interactive segmentation networks is typically borrowed from semantic segmentation networks (*e.g.* DeepLab [22]), or general purpose image processing architectures (*e.g.* U-Net [130]), deep interactive segmentation requires addressing two specific challenges: 1) how to best encode the sequence of user interactions in the architecture and 2) how to train the network so as to best match the behaviour of artists. In the rest of this section, we look at how existing works approach the design and training of interactive segmentation networks.

2.1.2 Segmentation Architectures

Most interactive segmentation networks are either based on semantic segmentation networks, such as DeepLab [22], or general purpose image processing networks such as U-Net [130]. Semantic Segmentation networks typically follow an Encoder-Decoder architecture and use off-the-shelf encoders pretrained on ImageNet [39]. For instance, the works of Xu *et al.* [162] and Majumder *et al.* [105] are based on the FCN semantic segmentation network [136], which itself is based on the ImageNet trained VGG16 network [138].

Note that techniques based on semantic segmentation networks typically operate at ¼ or ⅛ resolution, which is a too low resolution for our aim. Full resolution output can be obtained by using Graph-Cuts (see iFCN [162], RIS-Net [92]) or by appending a dedicated upsampling networks (see FCTSFN [69] and BRS [76]). Upsampling networks need to be trained afterwards, in a two-stage process. In RIS-Net [92], fullres prediction is obtained by

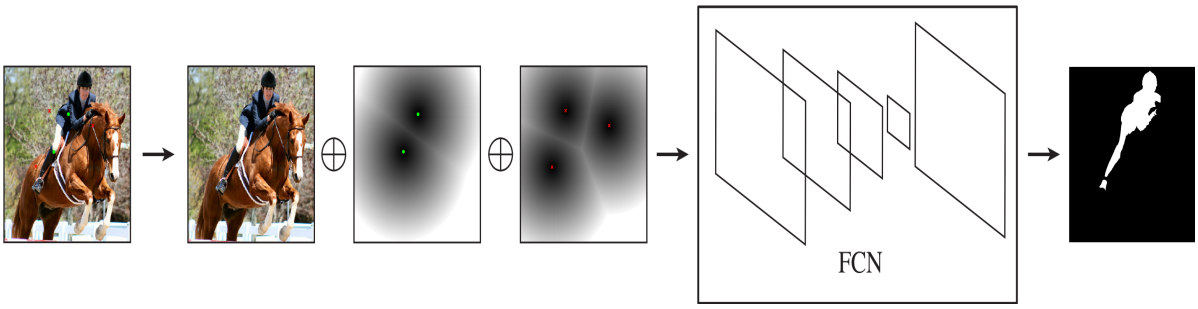


Fig. 2.3 Deep interactive segmentation pipeline, from [162]. The image is concatenated with distance transforms of the foreground and background clicks and fed to a neural network (FCN) to predict a mask.

fusing the predictions of two decoder streams: the first stream makes a global low resolution prediction and the second stream independently processes patches at click locations to give local hires predictions. In IIS-LD [91], a single full resolution network is proposed by using VGG hypercolumns. Table 2.1 gives an overview of existing interactive segmentation architectures.

2.1.3 Embedding User Interactions

Encoding Clicks to the Network. Click inputs are encoded as images either via a distance transform [17, 69, 92, 105, 162] or by fitting Gaussian masks to each of the clicks [14, 83, 91, 106]. An interesting alternative was proposed in *Content-Aware Multi-Level Guidance for Interactive Instance Segmentation* by Majumder and Yao [105]. In this work they encode the foreground and background clicks as a mask from the superpixels [1] they fall on and similarity of those to nearby superpixels. In effect this creates a roughly accurate mask which serves as guidance to the segmentation network.

Fusing clicks maps into the Network. In DIOS [162] the click maps are input to the to the network by concatinating them with the input image, as input to the first layer. Many other works adopted this approach(iFCN [162], RIS-Net [92], ITIS [104], DEXTR [106]), known as *early fusion*.
An alternative approach is to concatenate the click maps to a later stage of the network, not the first layer, this is known as *late fusion*. In IIS-LD [91], the click maps are concatenated with all the upsampled image features hypercolumn. In FCTSFN (Hu *et al*. [69]), VGG16 is applied separately to both the input image and the click maps. The lower resolution features from both inputs are then concatenated before being passed to further convolutional layers. They show that the late feature fusion of click and image streams out-performs concatenation of image

Table 2.1 Segmentation architecture in relevant previous works, detailing: the type of decoder and encoder used, the scale of the tensor after the decoder and the post-processing method used for upsampling the results to full resolution.

Name	Reference	Encoder	Decoder	Scale	Post-Processing
iFCN	Xu *et al.* [162]	VGG16 [138]	FCN [136]	1:8	graph-cuts [16]
ITIS	Mahadevan *et al.* [104]	Xception [33]	DeepLab-v3+ [24]	1:4	none
DEXTR	Maninis *et al.* [106]	ResNet-101 [65]	Deeplab-v2	1:8	none
VOS-wild	Benard *et al.* [14]	ResNet-101	Deeplab-v2 [23]	1:8	CRF [82]
FCTSFN	Hu *et al.* [69]	VGG16	CNN with bilinear upsampling	1:1	refinement network
IIS-LD	Li *et al.* [91]	VGG16	hypercolumn, CAN [27]	1:1	none
CAMLG	Majumder *et al.* [105]	VGG16	FCN	1:8	none
RIS-Net	Liew *et al.* [92]	VGG16	DeepLab-LargeFOV [23], custom CNN for local predictions around clicks	1:1	graph-cuts [16]
IITSEN	Bredell *et al.* [17]	U-Net [130]	U-Net	1:1	none
BRS	Jang and Kim [76]	DenseNet [70]	U-Net	1:1	refinement network

Table 2.2 User interactions encoding in relevant previous works.

Name	Reference	Click Embedding	Click Fusion	Feedback Fusion
iFCN	Xu *et al.* [162]	Distance Map	early	none
ITIS	Mahadevan *et al.* [104]	Gaussian	early	early
DEXTR	Maninis *et al.* [106]	Gaussian	early	none
VOS-wild	Benard *et al.* [14]	Gaussian	early	none
FCTSFN	Hu *et al.* [69]	Distance Map	late	none
IIS-LD	Li *et al.* [91]	Gaussian	late	none
CAMLG	Majumder *et al.* [105]	Superpixel based embedding [105]	early	optional
RIS-Net	Liew *et al.* [92]	Distance Map	early	none
IITSEN	Bredell *et al.* [17]	Distance Map	early	early
BRS	Jang and Kim [76]	Distance Map	early	none

Table 2.3 Loss and training in relevant previous works.

Name	Reference	Loss	Training Schedule	Training Dataset
iFCN	Xu *et al.* [162]	BCE	all clicks at once	PASCAL 2012
ITIS	Mahadevan *et al.* [104]	BCE	adding one click per epoch	SBD Training + PASCAL 2012
DEXTR	Maninis *et al.* [106]	BCE	4 boundary clicks at once	SBD Full
VOS-wild	Benard *et al.* [14]	BCE	all clicks at once [162]	SBD Full
FCTSFN	Hu *et al.* [69]	BCE	all clicks at once [162]	PASCAL 2012
IIS-LD	Li *et al.* [91]	IoU + click location	all clicks at once [162]	SBD Training
CAMLG-IIS	Majumder *et al.* [105]	BCE	all clicks at once [162]	SBD Full
RIS-Net	Liew *et al.* [92]	BCE + Click Discounting [92]	all clicks at once [162]	PASCAL 2012
IITSEN	Bredell *et al.* [17]	BCE	Separate network makes initial prediction, then adding clicks per image.	Medical [112]
BRS	Jang and Kim [76]	BCE	all clicks at once [76]	SBD Training

and clicks at the beginning of a single stream. We give an overview of fusion approaches in Table 2.2.

Mask feedback. In most prior works (including iFCN [162], RISNET [92], DEXTR [106], IIS-LD [91], FCTSFN [69]), when a user adds a new click, the previous prediction is discarded and a new prediction is make afresh. This means no state is carried on from one click to the next. We could try, however, to learn from previous clicks and predictions as they give more context to what the new click is trying to fix. The approach proposed in ITIS [104] and IITSEN [17] is to feed back the mask alongside the image and the clicks (early fusion) in order to retain some state information, see Table 2.2 for an overview. One potential issue with this early fusion approach is that it might be hard for the network to recover from a very poor previous prediction.

2.1.4 Training Scheme

User clicks need to be supplied during training. As it is impractical to provide human-generated clicks, the clicks must be simulated in some way. How best to do this is an open question.

In their early work on deep interactive segmentation Xu *et al*. [162] use a set of three strategies for generating clicks from a given segmentation label. The method goes as follows: a random number of foreground clicks are placed randomly on the object and a random number of background clicks are placed near the object. The details of the sampling strategy seek to mimic typical human input patterns and it has been adopted by many subsequent works [14, 69, 83, 91, 92, 105].

However, Mahadevan *et al*. [104] (ITIS) observe that this way of predetermining the click placements, independently of the network prediction errors, has an adverse impact on the final model accuracy. They propose to start with this scheme at first, but then, at each subsequent epoch, a new click is added on the centre of the largest incorrect region. This interactive training scheme tries to increase the correlation between the click locations and prediction errors. The issue is that the initial click samples are still randomly grouped. Also, later clicks are only updated one epoch at a time but the network weights may have changed significantly between each epoch. This means that the click sequences are not necessarily coherent.

A more direct approach for training interactive image segmentation networks was taken by Bredell *et al*. [17] for medical image segmentation. After an initial segmentation is computed by a separate network, the interactive network is trained by iteratively adding one click at a time. Each click is placed on the largest incorrect region. The approach, however, still relies on an initial segmentation, which we believe is unnecessary. See Table 2.3 for an overview of training schemes in the literature.

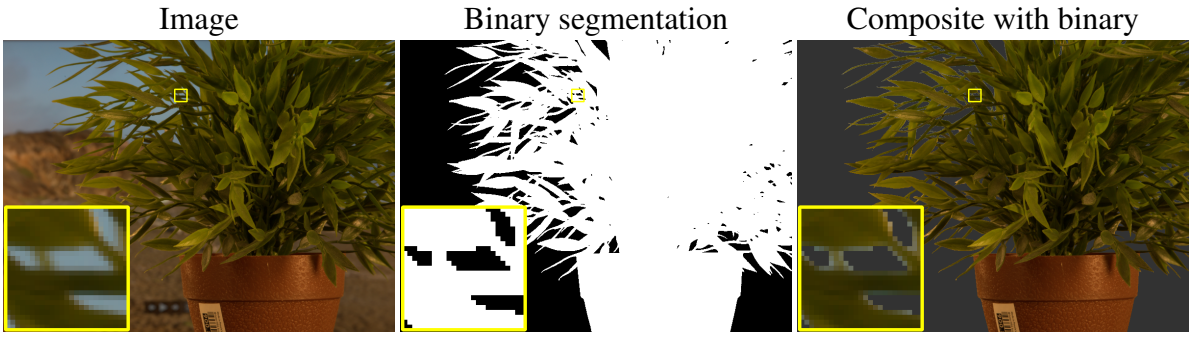


Fig. 2.4 Compositing a simple object with the perfect binary segmentation. Image from alphamattingcom [129].

The problem of selection ambiguity in training was addressed by Li *et al*. [91] which incorporated multiple-choice learning [60] into the network and training.

2.1.5 Non-Scribble based Segmentation

The focus so far has been on solving a binary segmentation problem for images using a scribble input. However, many different input methods and applications to video have been researched, so for context we give an overview of these related works. A common user input method is to envelop the object with a bounding box [86, 131, 161], and the algorithm can then determine the mask by relying on the fact that all pixels outside the box are background. Roughly tracing along the object boundary and automatically snapping to the edges was done by Barrett. & Mortensen [12]. Marking sparse boundary points [2, 83] or even extreme boundary points can be used [106] has had success with deep learning. Hu *et al*. [68] developed a method to parse natural language sentences into segmentation, *e.g*. "Segment the blue hat.".

Video segmentation from sparse user input or from marking the first frame has been a very active area of research lately [14, 36, 53, 62, 114, 118].

2.1.6 Binary Segmentation for Compositing

A binary segmentation of the object from the background could be used for selective image editing *e.g*. compositing. This naive compositing method is expressed in Eq. 2.2.

$$\text{Composite} = \begin{cases} \text{Original image} & \text{where segmention = 1} \\ \text{New background} & \text{where segmention = 0} \end{cases} \tag{2.2}$$

We show an example of this compositing method in Figure 2.4. As shown, it works somewhat well for opaque objects with simple boundaries, however some colour spill is visible along

some edges. The reason for the colour spill is that the pixels in an image of an object are actually a blend of the foreground and background colours. We can observe this in the inset of the image in Figure 2.4, the boundary pixels are a mix of the blue sky background and the plants green leaves. In this case the leaves are opaque, and the blending is due to the imaging process from a combination of defocus-blur and resolution-blur. The physical sources are discussed in detail in the Thesis of Christoph Rhemann [127], (for our purposes however understanding of the matting equation suffices, as deep-learning allows us to achieve excellent results by treating the rest as a black box).

Alpha matting is the problem of finding an objects transparency/translucency/blending/alpha matte and foreground colours. In the case above the transparency is relatively simple, as it is confined to a small band around the edges, in fact it could probably be well synthesized by a simple gaussian blur of the binary mask. In this case it is easy to think of interactively finding the alpha matte as merely an extention of interactive segmentation. The work from Rother *et al*. [131] (2004), made a strong contribution in this regard. They developed an algorithm for computing the alpha matte and foreground colours in a narrow band along the boundary of a binary segmentation. This 'border matting' approach works well in these cases, however it is not viable when the border is extremely complex and the object is partially transparent, see Figure 2.5. Nailing the matte in these difficult cases is what the field of alpha matting has revolved around.

2.2 Matting

We see a typical object for alpha matting in Figure 2.5. If we try to composite this complex hairy object using its binary segmentation we immediately encounter problems (bottom left). Moreover, using the alpha matte alone for compositing produces colour bleeding artifacts (bottom centre). Both the alpha matte and foreground colours are neccesary for a proper composite (bottom right).

The first successful model-based solution to alpha matting was *Bayesian Matting* from Chuang *et al*. [34] (2001). Here, the user marks some definite foreground and background pixels, this is known as a trimap, and the algorithm solves for matte in the unknown regions. This algorithm deployed the same ideas as in equation 2.1 but allowed for non-binary alpha. Also, rather than manipulating $p(\alpha|C)$ we solve for all α F and B. Chuang *et al*. iterated between solving for each variable, F, B, α. *Bayesian Matting* followed what is known as a sampling strategy, introduced by Ruzon *et al*. (2000) [133]. These samplings techniques try to find suitable candidate foreground and background colours for each pixel. These two could be substituted into the matting equation 1.1 to solve for α. Multiple strategies have been

developed for sampling candidates in order to improve speed and accuracy. Also multiple metrics have been developed to select the optimal foreground and background pairs from these samples [51, 79, 135, 152].

These samplings techniques typically lack spatial smoothness. Some early work was done on smoothness models for matting [141], however it was only with the 2007 landmark *Closed-Form Solution to Natural Image Matting* paper [87], that a comprehensive spatial prior model for matting was found. Levin *et al*. [87] re-expressed the compositing equation as follows in terms of the local colour line model [115]. This led to a closed-form approach. This was then widely adopted as a refinement strategy for algorithms without spatial smoothness priors [64, 80, 135].

The problem with these classic matting methods is that they struggle to overcome ambiguity in the case of similar foreground and background distributions, or when the colour line model did not hold. Deep-learning based matting methods are much more robust to these problematic cases.

2.2.1 Trimap based Algorithms using Deep Learning

Like with interactive segmentation, deep learning has hugely advanced the state of the art in matting. We first cover early approaches as the lessons learned have informed all subsequent work. The first deep-learning based approach for alpha matting came from Cho. *et al*. [32] in 2016 with *DCNN Matting*. They use a deep convolutional neural network to trained to refine mattes predicted from classic methods [26, 87]. The loss functions used in matting are so important that they deserve a separate section 2.4.2. The network was trained on the 27 images *alphamatting.com* dataset, which we discuss in more detail in Section 2.4.1. Cho. *et al*. used heavy data augmentation to expand their training set, most importantly they composited each foreground onto multiple backgrounds.

Their model was the state of the art when released, however no similar approaches have been taken since. Instead *Deep Image Matting* from Xu. *et al*. [160] set the standard. They did this by recognising and correcting three grave limitations of *DCNN Matting*. First and foremost, the training dataset is tiny; this limits learning potential and influenced the decision in opting for the other two limiting choices. Dataset size limits the model size [54]; this is partly why the DCNN matting model is only six layers deep each with 3x3 kernels. These dimensions mean the network is unable to capture semantic information since each output pixel is influenced by an input pixel at most twelve spaces away. Finally, the tiny dataset makes learning very difficult in general, and so they do not even attempt an end-to-end approach. The issue with this is that if both intermediate mattes are very inaccurate, then training is inhibited.

In addressing these limitations, *Deep Image Matting* construct a ~20x larger matting dataset, to train a larger and deeper network architecture in an end-to-end fashion. We discuss the dataset in detail in Sec. 2.4.1, but for now it is important to note that they took care when collecting the images to contain challenging cases for matting, like hair and transparency. Without this, the matting network converges to a simple model biased towards outputting mattes with hard edges. They use a deep encoder-decoder style network where the encoder is the VGG16 [138] network pretrained on ImageNet, similar to SegNet [9]. The network is trained end-to-end with patches of images and corresponding trimap patches as input. A similar data-augmentation strategy is applied. They show that their approach learns higher-level information and can resolve more ambiguities in colour and texture. For example, the network excels at separating hair from the background.

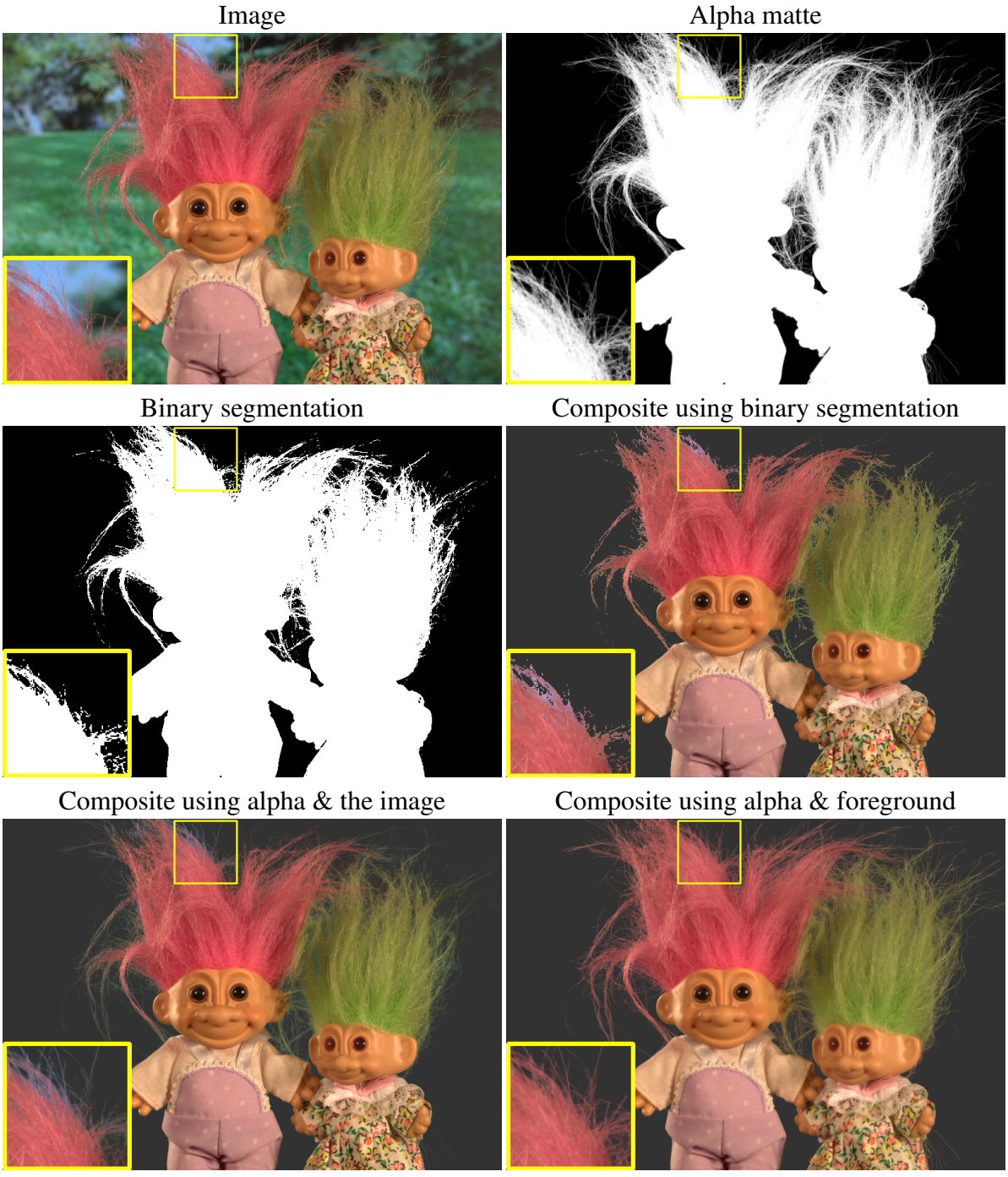
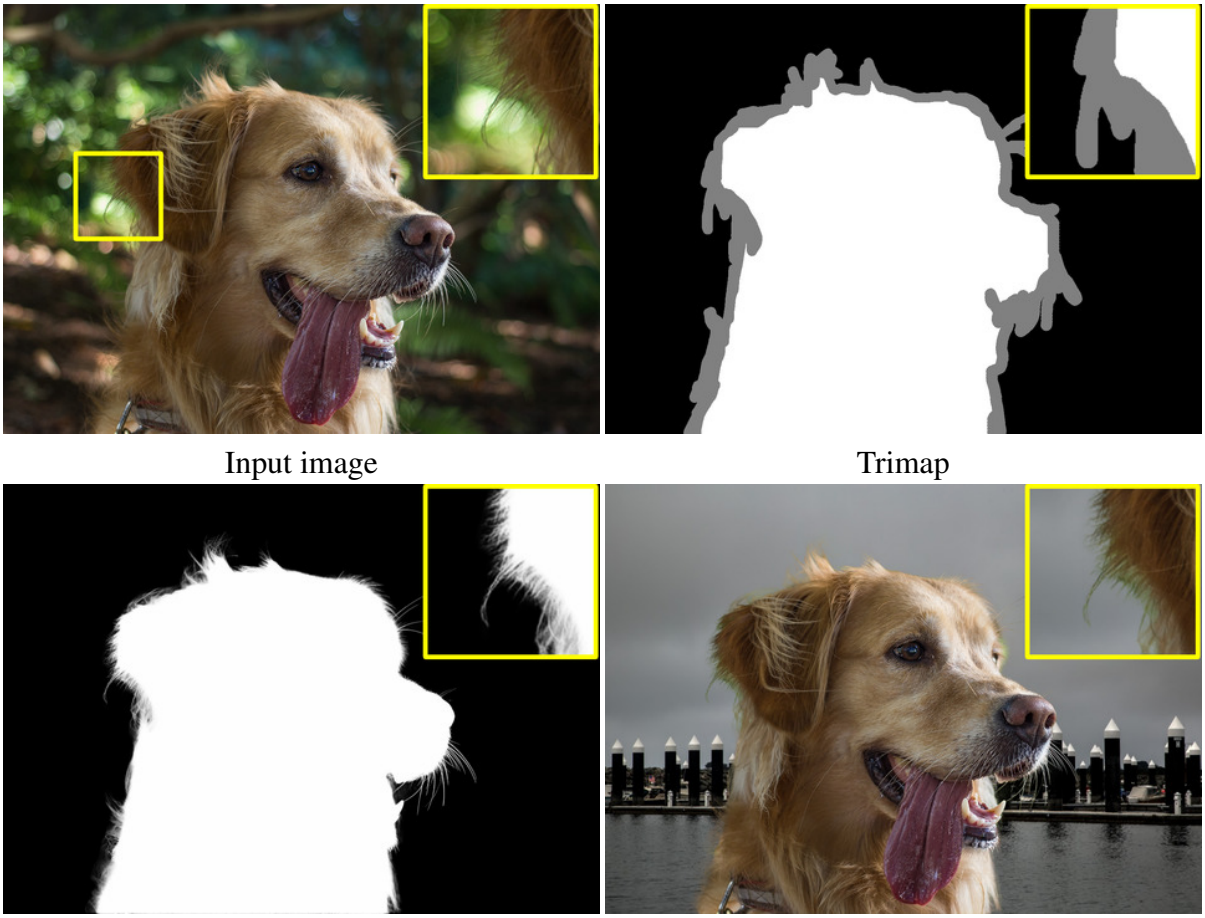


Fig. 2.5 Compositing a complex object with the perfect alpha matte and foreground colours. Image from alphamattingcom [129].



Input image | Trimap

Alpha [67] | Composite using alpha & foreground [67]

Fig. 2.6 Alpha matting and compositing with trimap input.

Table 2.4 Deep Alpha Matting Network Architectures, Input and Losses.

Name	Input	Alpha Losses	F, B colour losses	Trimap Prediction
Deep Automatic Portrait Matting [137]	Portrait photo	(custom weighted) L1	-	Yes
DCNN Matting [32]	Image + Trimap + results of CFM and KNN	L2	-	No
DIM [160]	Image + Trimap	huber + compositional	-	No
AlphaGAN [102]	Image + Trimap	huber + compositional + adversarial	-	No
IndexNet [101]	Image + Trimap	huber + compositional	-	No
VDRN [143]	Image + Trimap	huber + compositional	-	No
Learning-based Sampling [144]	Image + Trimap	huber + compositional + grad	L1 + composite using gt alpha	No
Disentangled Image Matting [19]	Image + Trimap	L1	-	Yes - Multitask learning cross entropy
Late Fusion Matting [166]	Image	L1, soft penalty	-	Yes - multiple losses
Context-Aware Matting [67]	Image + Trimap	Laplacian, vgg alpha*F_gt	L1, vgg alpha_gt*F	No
GCA [89]	Image + two binary channel trimap	L1	-	No
Boundary-Aware Portrait Segmentation [28]	Portrait image	Cross entropy, gradient	-	No
Fast Deep Matting for Portrait Animation [169]	Portrait image	huber + compositional	-	yes
INDUCTIVE GUIDED FILTER Matting [90]	Image + Binary Mask	L1, weight on boundary, gabor, adversarial	-	-
PORTRAIT [116]	Portrait image	Fixed Matting step	Predicts alpha*F	fixed dilation of mask rcnn
HAttMatting [123]	Image	L2 + Adversatial + SSIM	-	No
Semantic Human Matting [25]	Human Photo	L1 + L1 _compositional	-	yes
Boosting Semantic Human Matting [99]	Human Photo	L1	foreground rgb	No binary mask intermediate
Background Matting [134]	Human Video + bg im + motion+ binary mask	L1+grad+L1_compositional	foreground rgb	no

Deep Alpha Matting Network Architectures

Most of interest to our research are subsequent works which have explored deeper network architectures, works that better model the physical matting problem, and works that better exploit the user trimap. We present an overview of deep matting literature in terms of architectures, inputs and losses in Table 2.4.

Deeper architectures are interesting for matting as they have allowed for more finely detailed predictions and better solve difficult cases. They have been made popular and achievable by the ResNet [66], and U-Net [130] architectures which employ residual/skip-connections that add/concatenate early convolutional outputs with the inputs to later convolutional layers. The *AlphaGAN* [102] architecture uses a ResNet-50 encoder with a U-Net style decoder. In *VDRN Matting* (2019) [143] an even deeper encoder and decoder residual networks was proposed, see Figure 2.7. In *IndexNet* (2019) [101], the encoder-decoder uses a learnable index pooling to aid the upsampling in the decoder. In *Context-Aware Matting* (2019) [67], two encoders with atrous convolutions are fused together.

The matting problem is physically based in the matting equation 1.1, some works have leveraged this prior knowledge for more accurate results on both the α and F predictions. In *Learning Based Sampling Matting* [144] problem is approached from solving for each unknown in the matting equation F, B, α sequentially, see Figure 2.7. First, they use a network designed for image inpainting [164] to solve for the background, then use another network with the same architecture to solve for the foreground with the previously predicted background as input. Finally, they join the predicted foreground and background with the image and trimap as input to the *AlphaGAN* architecture. In *Context-Aware Matting* (2019) [67] two decoders are used one to solve for the foreground colours and another to solve for the alpha matte.

The user trimap indicates definite foreground and background regions, and it is hoped that the network implicitly learns to leverage it by supplying it as input during training. Some works have used architectures and custom layers designed to use the trimap explicitly. In *GCA Matting* (2020) [89], an attention mechanism is designed to use the trimap as a guide. In [19] (2019), a two-stage strategy is proposed to first refine the original trimap then proceed with alpha matting.

2.2.2 Non-Trimap based Matting

The focus so far has been on solving the matting equation for arbitrary images using a trimap. However, matting is applicable in many more situations, and different constraints can be used to solve the problem without a trimap. Here we give a wide overview of these related works to give context to our contributions, and in the next section we delve into using scribbles for

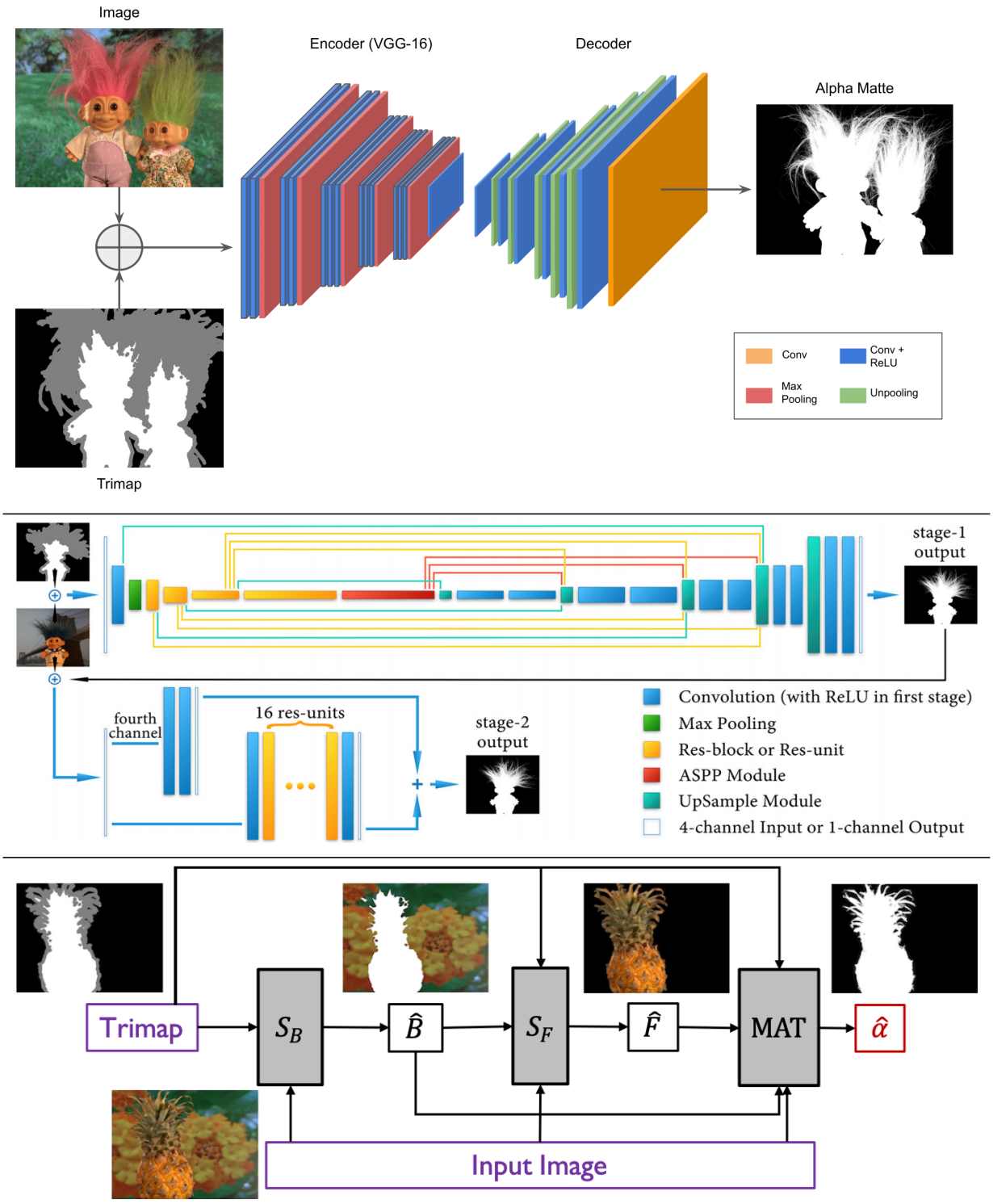


Fig. 2.7 Three convolutional network architectures for alpha matting.
1. *Deep Image Matting* [160]. Encoder-decoder architecture without skip connections. (Redrawn from [160]).
2. *Very Deep Residual Networks for Matting* [143]. Note the increased network depth with extra layers and skip connections. (Figure from [143]).
3. *Learning Based Sampling for Natural Image Matting* [144]. Note the three sequential CNNs. The first predicts the background, the second uses this to predict the foreground and the third predicts the alpha matte from these. (Figure from [144]).

matting. The most well-known approach to extracting an object matte is to use a greenscreen; this constrains the background to a roughly uniform colour. This still leaves two unknowns though, so an exact mathematical solution is not possible, but many techniques have been developed into widely used commercial products [18, 155]. From a layman's perspective, the greenscreen problem seems solved, far from it however, new research continues to push its ease of use and accuracy [5, 85]. Although the greenscreen is insufficient to solve the problem exactly, it is solvable within more constrained cases, such as photographing in front of two known backgrounds [40, 57, 129, 140] or with polarised light [13, 108]. The more general case for video with arbitrary backgrounds has been tackled with methods to propagate user input to nearby frames [8, 10, 145]. Deep learning has had much success in predicting accurate alpha mattes when trained on specific scenarios, such as cropped to the object [124, 166], with a roughly known background [77, 134], segmenting portrait pictures of people [25, 99, 116, 137]. The matting equation we use for a scene splits the image into two colours, a more general formulation known as 'soft segmentation' splits the scene into a larger number of colours, this can be useful when the part of the image we want to edit is an overlapping colour rather than an object [6, 7].

Completing the compositing story. We conclude this review of alpha matting by completing the story of the compositing workflow. We have covered how the matting equation 1.1 is the correct model for extracting an object but it does not ensure the object sits naturally in the new scene. If we look back at the composite in Figure 2.6, we somewhat subconsiously observe cues that the image is fake. Some of these cues may indeed be from colour-spill or errors in the alpha-matte, however the lighting is a standout inconsistency and it is not modelled by the matting equation. A parallel line of research has sought to create realistic composites, given the alpha mask, in various ways; modelling for matching shadows [37, 157], modelling correct lighting [155], light refraction [21, 170], removing unnatural seams in the composite [119] and more generally by approaching natural composites as a black-box deep learning problem [146]. This is clearly an interesting problem, however it is more dependent on artitic preference. In this thesis we solely focus on correctly cutting out the object.

2.3 Interactive Matting

The recent research the focus on trimap based algorithms has somewhat distracted from the importance of interactivity in matting tools. In the real world, interactivity has long been a central concern. Even in the seemingly straightforward greenscreen application, matting algorithms, also called keyers in this Visual Effects context, have always placed the user at

Fig. 2.8 Alpha matting and compositing with scribble input. Figure and algorithm used from *Easy Matting* [58].

the centre of their design, acknowledging the inherent difficulty of the task and giving plenty of control to the user. Good examples are the specific user workflows in industry tools such as Keylight and Primatte, or the work of [5] in the academic literature. We have seen how interactivity is given central focus in the related problem of interactive segmentation. The work of Liu *et al*. [98] being particularly interesting in that regard, as it acknowledges the progressive nature of image editing and pays particular attention to the artist workflow. In this thesis, we refocus on interactivity in matting research by developing novel algorithms and evaluation for interactive matting. Here we give an overview of related works.

Scribble based Matting

One well-studied form of interaction is the use of hand-drawn scribbles to sparsely indicate regions of definite foreground and background. The idea is to leave the delicate delineation of the transparent region as vague as possible. Methods that handle scribbles are numerous and include the unified image segmentation and matting approach of [151], Geodesic Matting [10], Closed Form Matting [87], Shared Matting [51], non-local-Matting [84] or KNN-Matting [26].

In Easy Matting [58], one metric of success is the actual time taken by artists to complete the entire matting process. Some of the non-deep learning methods are, however, actually too slow for real-time use and, as such, have no qualitative or quantitative evaluation of matte accuracy vs time or number of interactions. Since then, the trimap-based evaluation datasets alphamatting.com and Adobe Composition 1k have gained wide adoption by the community, thus shifting the focus solely to maximising α-matte accuracy from pre-defined trimaps. Actual evaluation of the interactive capabilities is not commonly mentioned anymore.

The use of trimaps as intermediate representation then becomes an entry point for integrating user interactions. For instance, in [128], trimaps are automatically generated from a fixed set of scribbles using Graph-Cuts. The advantage of the trimap format is that it is compatible with virtually all existing matting algorithms. The issue is that it implies a two-stages process and

any error in the trimap generation will propagate to the following matting stage. Also, what we want here is to take advantage of recent advances in semantic segmentation and harness the deep convolutional features to quickly outline semantically consistent shapes. A simple solution is to automatically generate trimaps from the results of an interactive segmentation network, as proposed in [29]. This is however not optimal as we do not know how wide the transparent region should be. In fact, we believe that we can avoid the trimap intermediate representation altogether and that a single unified approach is possible and preferable.

Notice that no end-to-end deep learning based approach has been taken for interactive matting. However, a naive combination of existing works can achieve interactive matting in simple cases. We can feed scribbles to a deep interactive segmentation algorithm to produce a binary mask. Then the mask boundary can be dilated through morphological operations to generate the unknown region of a trimap for the image. This can be fed to a deep matting algorithm. This can work in simple cases but is fundamentally limited. If the object has holes or transparency or varying width of semi-transparent hair/fur then this approach will fail. An end-to-end solution is needed. We desire a flexible matting algorithm, one that can matte all objects with sparse scribble input, and capable of refining the result to perfection with more dense trimap input.

2.4 Measuring Success

Making comparisons between the many algorithms above requires data and methodical ways of comparing predictions on that data.

2.4.1 Datasets

Creating and collecting matting and segmentation datasets is an important ongoing pursuit. Here we discuss datasets which serve one of three purposes. *Qualitative datasets*: Data without ground-truth on which to evaluate an algorithm qualitatively. *Test sets*: Data with ground-truth on which to evaluate an algorithm qualitatively. *Training sets*: Data with ground-truth to be used to tune the parameters of an algorithm. In this section, we will describe the datasets found in this thesis.

Segmentation Datasets

We evaluate our interactive segmentation work across three publicly available datasets:
GrabCut *(Test Set)* [131] is a 50 image interactive segmentation dataset with 50 ground truth masks, although not all pixels in a thin band around the object boundaries are labelled.

GrabCut [131] Berkeley [107] SBD [63]

Fig. 2.9 Example images and labels from common interactive segmentation datasets.

Typically, the images contain a single object, and the object occupies most of the scene, and the foreground and background distributions are well separated.

Berkeley *(Test Set)* [107] is a 96 image interactive segmentation dataset. The dataset contains 100 ground truth masks and all pixels are labelled. The dataset is more challenging than GrabCut. Many objects only occupy a small portion of the image, and there are times other similar objects in the background. Many masks contain fine details and long extending parts like tails and legs.

Pascal VOC 2012 *(Training & Test Sets)* [43] is 1449 image semantic segmentation dataset. There are a total of 3427 individual object masks belonging to 20 object categories. The restricted number of semantic categories means algorithms with a stronger prior on those categories excel, but this impacts generalisation ability. There are several factors that make this dataset quite challenging. The objects in the dataset are from an extensive range of scales. Some object categories, like bicycle, plant, and a chair, have thin object boundaries. Having multiple objects instances and categories per image is common, as well as often occluding each other.

The primary training dataset for segmentation is Pascal. [43] training set which has 1464 labelled images. However, this dataset is too imprecise for very high-quality segmentation. In order to train for more accurate prediction, most models are trained with the SBD dataset. It consists of 10'582 images from the PASCAL VOC dataset labelled by Bharath *et al*. [63]. The boundary labelling is more accurate than PASCAL, and there are no unannotated pixels.

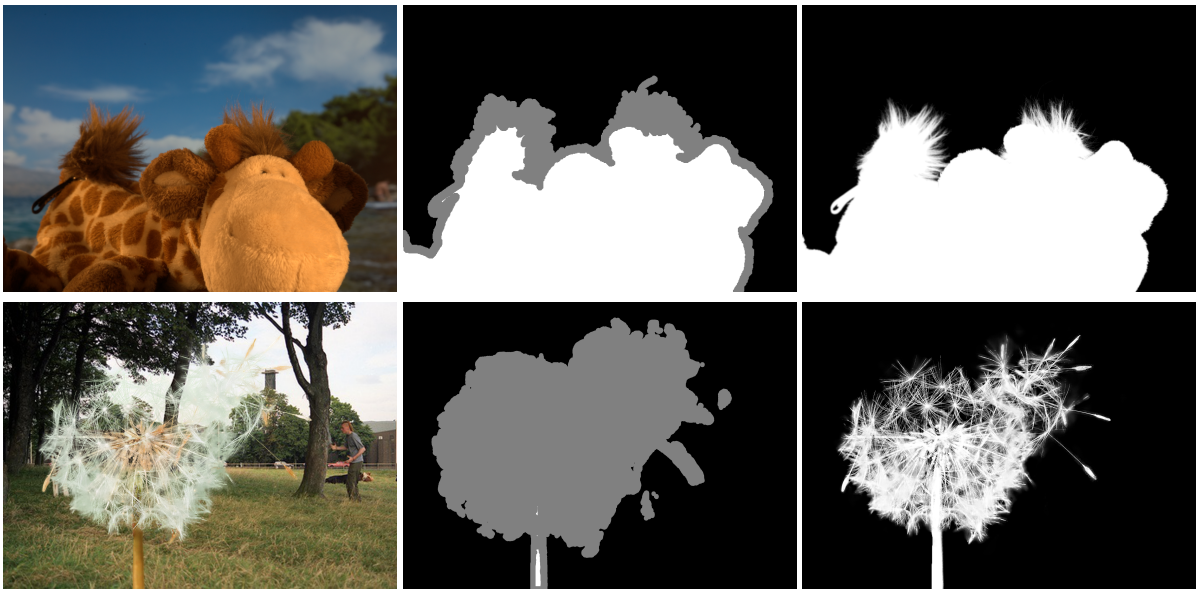


Image Trimap Ground Truth alpha

Fig. 2.10 Example images, trimaps and alpha mattes from the alphamatting.com training set [129] (top row), and the Composition-1k test set [160](bottom row).

Matting Datasets

We can view matting datasets as a generalisation of segmentation datasets, *i.e.* by setting $\alpha_i > 0.5 = 1$ we return to a binary labelling. However, in practice using matting datasets for segmentation is not helpful. This is since each field has developed its datasets in accordance with their end objectives, rough labelling for segmentation and fine details for matting. So in practice, segmentation datasets have low-resolution poor quality images, containing many simple objects sometimes overlapping. Matting datasets are high resolution, noise and artifact free images, focusing on a single complex object.

We train and evaluate our matting work across four publicly available datasets:

Alphamatting.com benchmark *(Training & Test Set)* [129]. This dataset contains eight images with three trimaps each. The user computes the alpha mattes and uploads them to the server where they are then ranked among other methods.

Composition-1k *(Test Set)* [160]. This dataset contains 1000 images together with trimap and ground truth alpha. It is made up of 50 unique objects with foreground and alpha data, the objects are composited onto 50 different backgrounds, from the Pascal VOC dataset [41]. The trimaps are generated from the ground truth alpha through dilation of the transparent region. The dataset contains six times as many unique objects, one -hundred times more images, and more challenging examples than the test set from *alphamatting.com* [129]. Though since the

objects are composited onto random backgrounds, the images are quite-unnatural.

Deep Image Matting dataset *(Training Set)* [160]. The second deep learning approach from *Xu.* et al. [160] established the much larger Deep Image Matting dataset. The set consists of 431 foreground object images with ground truth alpha mattes. These were collected by carefully annotating existing photographs with Photoshop matting tools. In their work, they constructed the (image, alpha) pairs by compositing each foreground onto 100 randomly chosen backgrounds from MS-COCO [96]. To create a 431,000 image dataset.
Real Image Matting dataset *(Qualitative Set)* [160]. This dataset is a collection of 31 real-world photographs of objects, mostly people. One manual trimap annotation is provided per image. See example image, trimap and prediction in figure 2.6. The purpose of this dataset is to qualitatively evaluate matting algorithms performance in-the-wild. This is important for deep matting algorithms as they are trained on synthetic images. Of particular practical interest is recent work by Hou. and Liu. [67] which showed how data-augmentations (blurring and re-JPEGing) of the training data can improve generalisation to real world images.

2.4.2 Metrics

We discuss two intertwined perspectives for judging interactive segmentation and matting algorithms. First is measuring user experience, and second is measuring the quality of a given matte or segmentation. User experience has been a prime focus of the literature on interactive segmentation, and matte quality has been a prime focus of the literature on trimap based matting. In both cases, the ultimate judgement lays with the artist using the algorithm, and indeed averages of variants of this subjective metric have been used to compare segmentation and matting algorithms [67, 107, 129, 160]. However, quantitative metrics of user experience and matte quality are obviously important. Our subsequent overview of the literature will prepare the reader to understand how we could evaluate *interactive* matting algorithms.

Of equal importance these days is the use of quantitative metrics to tune/train learnable parameters in segmentation and matting algorithms. Interestingly the metrics used for evaluation are not always the same as those used for training. In the case of deep learning we pay particular attention to differentiable metrics which can be used together with backpropagation and stochastic gradient descent to train deep models [54]. These metrics are referred to as 'losses' in the training context. An additional reason is that it can be beneficial to train with the sum of multiple, multiple losses, this is known as Multi-Task Learning [20, 132]. So in this section, we also cover the common combinations of losses for interactive segmentation and matting.

Binary Segmentation Metrics

The most common measure of binary segmentation quality is the **Intersection over Union (IoU)** [41],

$$\text{IoU}\left(A,\hat{A}\right) = \frac{|A \cap \hat{A}|}{|F \cup \hat{A}|} \tag{2.3}$$

which measures the area of overlap of the predicted foreground, $\hat{A}$, and the ground truth foreground, A, divided by the area of their union. IoU is particularly useful, compared to say L1 error, because it is scale-invariant

However, the IoU metric cannot be used for training as it is non-differentiable; it requires binary outputs to calculate the areas of overlap and union. For training an alternative loss is used *e.g.* cross-entropy, or the 'soft' differentiable version of IoU [91].

In interactive segmentation combining multiple losses for multi-task learning has been used many ways. Liew *et al.* [92] propose to penalise the prediction of both their global prediction and local prediction network heads. The concept of the local prediction loss around clicks was relaxed by Li *et al.* [91] who imposed an L1 penalty on the pixels where the clicks are placed. Le *et al.* [83] incorporated an L1 penalty, weighted by a Gaussian, on the predictions around the clicked pixels. They also included a loss penalising the prediction of the object boundary.

Matting Metrics

The alphamatting.com [129] benchmark introduced four perceptually motivated metrics to measure alpha matte quality. Their validity was established through measuring spearman rank correlation with human perception in user experiments. They have been used in almost all matting evaluation since.

Mean Squared Error (MSE). This is the most common measure of matte quality. The IoU is not defined for non-binary values.

$$\text{MSE}\left(\alpha,\hat{\alpha}\right) = \frac{1}{N}\sum_{i}^{N}\left(\alpha_i - \hat{\alpha}_i\right)^2 \tag{2.4}$$

Sum of Absolute Differences (SAD). This is an important metric for ensuring definite foreground and background regions are predicted as definite also. For example, mean squared error penalises a prediction of $\hat{\alpha} = 0.1$ where $\alpha = 0$ far less than SAD does.

$$\text{SAD}\left(\alpha,\hat{\alpha}\right) = \frac{1}{N}\sum_{i}^{N}\left|\alpha_i - \hat{\alpha}_i\right| \tag{2.5}$$

The next two metrics are more specific to alpha matte quality and were designed by Rhemann *et al*. [129] to model specific aspects of human quality perception.

Gradient Error (GRAD). This measures the difference between the spatial gradient of the predicted and ground truth alpha mattes. This excels in finely detailed areas like fur and hair where some algorithms tend to oversmooth predictions.

$$\mathrm{GRAD}(\alpha, \hat{\alpha}) = \frac{1}{N}\sum_{i}^{N}(\nabla\alpha_i - \nabla\hat{\alpha}_i)^2 \tag{2.6}$$

Connectivity Error (CONN). This measures the degree of connectedness by means of connectivity in binary threshold images computed from the alpha matte compared to the ground-truth. In general, this loss measures where the prediction disconnects regions that are supposed to be connected, for example, a disconnected piece of floating hair. The details are quite involved, so we refer the interested reader to follow the original specification [129].

SAD, MSE and GRAD are commonly used for training neural networks for alpha matting. Connectivity error is non-differentiable, so it cannot be used for training. Many other metrics and variations on these have been proposed, and we give an overview of this in Table. 2.7. We give a more detailed discussion and experimental analysis of losses for alpha matting in Chapter 4.

User Experience Metrics

User experience metrics aim to measure the perceived satisfaction of a person using the algorithm to generate a binary segmentation or matte quality. They aim to reflect the full interactive selection process rather than just a snapshot. A robotic program is used to simulate a human user, and the matte quality is measured over multiple interactions using the metrics previously mentioned.

Number of Clicks (NoC) *(Segmentation)* [162]. A robot simulates user scribbles or clicks as input to an interactive segmentation algorithm, algorithm details in Chapter 3. The number of interactions required to reach a baseline accuracy is measured over a segmentation dataset.

Area under the mIoU curve (AuC mIoU) *(Segmentation)* [76]. A robot simulates user input as before, and the average IoU is measured.

Trimap Dilation stability *(Matting)* [160]. Robustness to trimap dilation is measured by dilating the trimap to increasing amounts and plotting the MSE vs dilation width.

Part of our contribution in interactive matting is to measure the AuC MSE accuracy for matting algorithms on matting datasets, see Chapter 5 for details.

2.5 Conclusion

In this chapter we have given a detailed review of the interactive image segmentation and matting literature. Deep-learning approaches to these problems have significantly advanced the state of the art. However, state-of-the-art 'interactive' matting algorithms do not use deep learning and struggle in cases of textured regions and overlapping foreground and background distributions. In this thesis, we develop a novel deep-learning based approach to interactive image matting. We begin by investigating the critical components of interactivity in deep interactive image segmentation. Next, we do the same for foreground colour prediction in deep image matting. Finally, we show how to combine learnings from each of our methods to produce the first deep interactive matting algorithm.

Chapter 3

Deep Interactive Training and Architecture for Object Selection in Images

Interactive object cutout tools are the cornerstone of the image editing workflow. Recent deep-learning based cutout algorithms have made significant progress in handling complex images. They can often achieve rough binary selections with just a few clicks. Nevertheless, the accuracy tends to plateau once this rough selection has been reached. That is, finely detailed selections are not possible. In this work, we address two causes of this plateau. The first reason is that current algorithms insufficiently leverage each user interaction. The second is that their potential is limited by the current training datasets.

We propose a novel interactive architecture and training scheme for deep interactive segmentation. We introduce a synthetic training dataset with pixel-accurate complex object boundaries. Comprehensive experiments support each step of our approach, and our network achieves state-of-the-art performance.

3.1 Introduction

As covered in the previous chapter, interactive image segmentation aims at generating a binary mask to delineate a foreground object of interest from the background. Unlike semantic image segmentation, user interactions are expected and must be exploited. In this chapter, we cover interactions in the form of clicks that the user will place to fix errors in foreground or

Parts of this chapter have been published [48], and parts are under review for a journal extention [47].

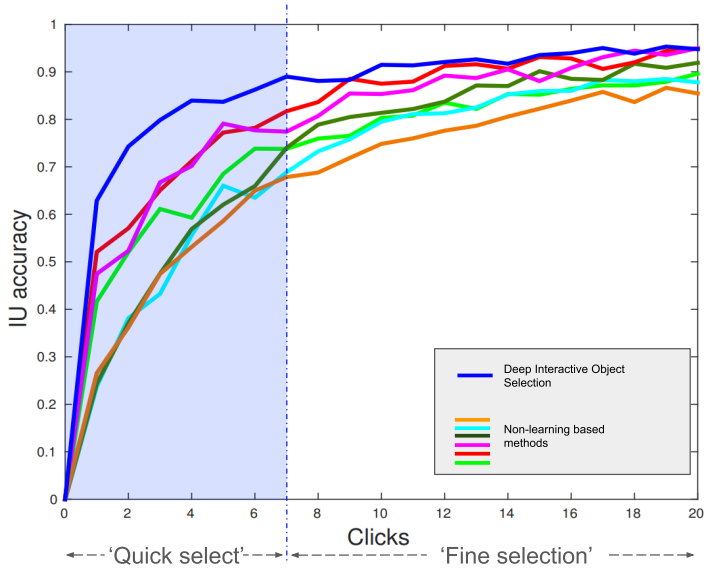


Fig. 3.1 Number of clicks vs. mIoU accuracy for *Deep Interactive Object Selection* [162], and non-learning based methods(top to bottom [10, 16, 55, 59, 59, 148]) on the GrabCut [131] dataset. Note the plateau after initial clicks and how non-learning based methods close the gap.

background areas. In an ideal interactive session, the main shape is sketched in a few clicks, then smaller local details are refined with subsequent edits. Exploring and exploiting this interactive editing loop is the objective of this chapter.

We have covered classic approaches to interactive segmentation [16, 81, 111], and pointed out how these works are held back by poor performance on textured regions. Recently, deep-learning approaches have excelled at producing rough segmentation masks with minimal user input. Object masks can now be typically be extracted to 90% accuracy(mean Intersection over Union/mIoU [43]) in under six clicks [14, 69, 91, 92, 104–106, 162], even on highly textured images. A caveat of its successes; deep-learning approaches are also observed to plateau short of 95% accuracy (See Figure 3.1), even after ten or more clicks and are unable to output very detailed masks, even after 20 clicks.

Low-resolution predictions may be sufficient for casual applications, like dataset annotation. For high-end professional applications, *e.g.* Photoshop, this is of little interest as professional artists can manually draw 90% accurate masks in no time. What they actually need is a tool to help them reach 99-100% accuracy reliably. See Figure 3.2 for a comparison of results achieving 90% [106] accuracy and our results which achieve 99%.

One reason for this plateau is that current architectures manipulate images and click interactions at low resolution, while fine details are usually only recovered through post-

Fig. 3.2 Comparison between a state-of-the-art segmentation algorithm which reaches 90% accuracy [106] and our results (right) at 99% accuracy.

processing stages (either using graph cuts [92, 162] or CNNs [69, 76]). This is partly a consequence of existing benchmarks having set low accuracy targets for the task [162]: in the Berkeley benchmark[107] the threshold is set to 90% accuracy, 90% for the GrabCut dataset [131] and only 85% for PASCAL [43]. These arbitrary targets come from a focus on quick rough selections and imperfections in the manual delineations of the objects when creating the ground-truth labelling. Any pixel within 3-5 pixels of the object boundary is typically ignored in the evaluation. As a result, it was noted in FCN [136] that working at $8\times$ lower resolution yields sufficient accuracy for the benchmark tasks of semantic segmentation, and many deep learning architectures are content working at $8\times$ or $4\times$ lower output resolution than the original image [14, 104–106, 162].

Another reason for the plateau is that current approaches treat all user interactions the same. During training, a set of foreground and background clicks are typically placed on the object and background, according to heuristic strategies [69, 91, 92, 104–106]. These clicks are placed together during training, without keeping any state or order information from one interaction to the next. The networks are thus not specifically trained to respond to corrective user clicks. Only in [17, 104] was the idea of training the network one edit at a time was explored, but only after that a good initial segmentation is given. There is thus a need to better focus on the way artists approach iterative segmentation and how to respond precisely to their input.

The primary objective of this work is to reach higher levels of segmentation accuracy by working at full resolution and modelling, from the start, the selection process as a series of interactions, whose purpose shifts over time from identifying the object globally to refining it locally. To achieve that goal, we are making a series of contributions covering all aspects of the problem. These contributions include:

1. A single architecture which allows for high resolution processing. The core architecture is based on a U-Net style decoder with skip connections to handle fine details. We further incorporate a Guided Filter layer [156] to refine the results and produce precise transparency masks.

2. Our architecture is split across two encoding streams: an Image stream and an Interaction stream. This separation of the inputs improves the propagation of the user interactions throughout the network and helps precisely respond to each click.

3. A training strategy where the corrective clicks are sequentially added from the first click, so as to match the artist workflow and thus jointly train for initial rough segmentation and detailed refinements.

4. Demonstrating that a high quality synthetic dataset, specifically designed to target fine details, can be used to further improve the quality of the segmentation.

The proposed method achieves state-of-the-art performance for any number of clicks and can reach 99% accuracy for 62% of images at 20 clicks.

The rest of the chapter is organised as follows. In Section 3.2 we detail our proposed method. In Section 3.3 we compare our method to state-of-the-art algorithms and we perform ablation experiments to quantitatively and qualitatively assess our contributions. In Section 3.4 we present our new training synthetic dataset and study how it can impact the quality of the segmentation.

3.2 Proposed Approach

3.2.1 An Interactive Hi-Resolution Network Architecture

The proposed architecture is presented in Figure 3.3. The segmentation masks are achieved through the use of an encoder-decoder network. One key aspect of our model is that we split the input image and user interactions into two streams: the Image stream and the Interaction stream. The separation into two streams is designed to allow for both early and late fusion approaches and thus better respond to user interactions.

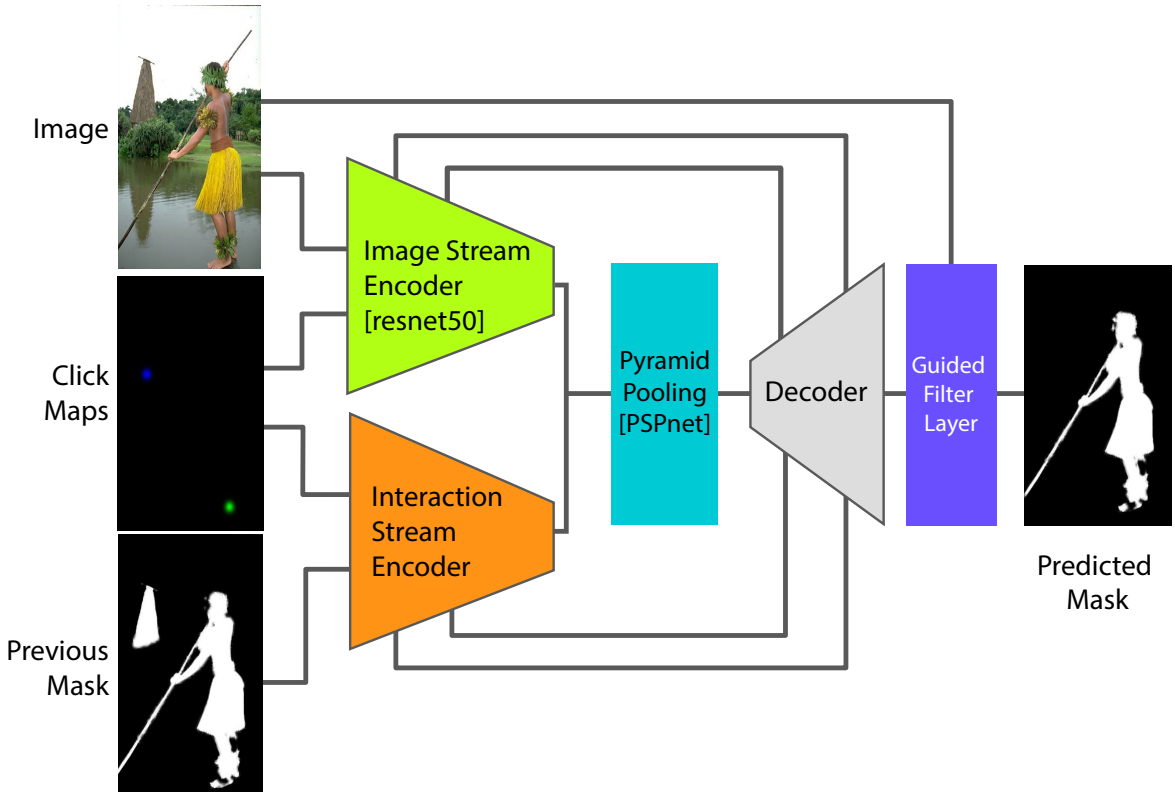


Fig. 3.3 Proposed Architecture. The *Image Stream* generates image features using ResNet-50, whereas the *Interaction Stream* embeds the previous mask estimate and the user clicks. Both streams get fused and pass through the pyramid pooling. The decoder follows a U-Net style architecture. The final layer is a 5×5 Deep Guided Filter layer, which allows for fine details extraction.

Our approach differs in some key ways from prior methods which have used the previous mask as input to the network [17, 104] and methods which separate the streams for clicks and the image [69]. Firstly, we adopt a late fusion for integrating the mask to the image stream. By using a separate branch, we prevent poor previous mask estimates from affecting the computation of the image features, while still allowing the mask information to be used. Secondly, we feed the clicks to both the image and interaction stream, which we observed to improve accuracy, as opposed to using early or late fusion alone.

The click maps are our encodings of the positive and negative click coordinates for input to the network. We encode the foreground and background clicks as in [83] using Gaussian masks at three different scales, centred on each click. Specifically they are a gaussian function, $exp(-\frac{\min_i d(p,c_i)^2}{2(\sigma L)^2})$, where $\min_i d(p,c_i)$ is the distance from pixel p to the nearest click c_i of that category. We use three values of $\sigma \in \{0.02, 0.04, 0.08\}$, this helps the network localise the clicks at different spatial scales.

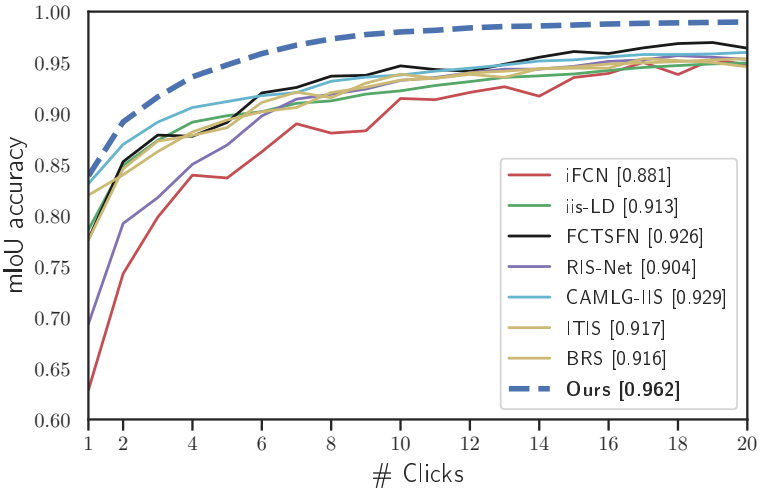


Fig. 3.4 Comparison of mean IoU scores after n clicks for the GrabCut testset [131].

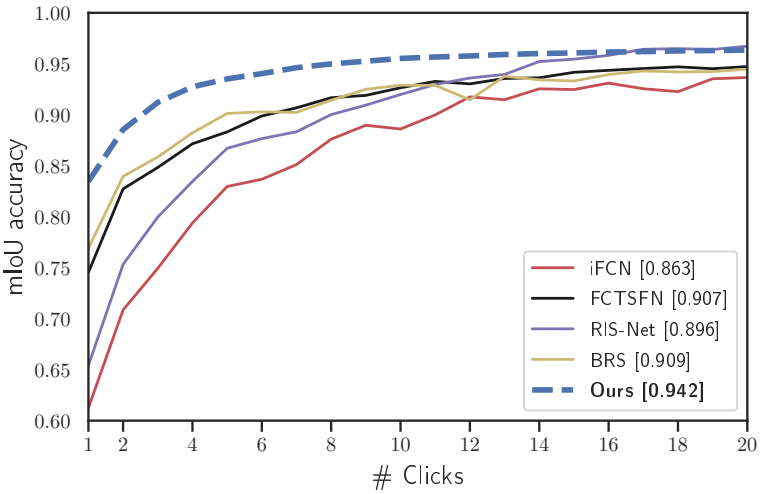


Fig. 3.5 Comparison of mean IoU scores after n clicks for the Berkeley testset [107].

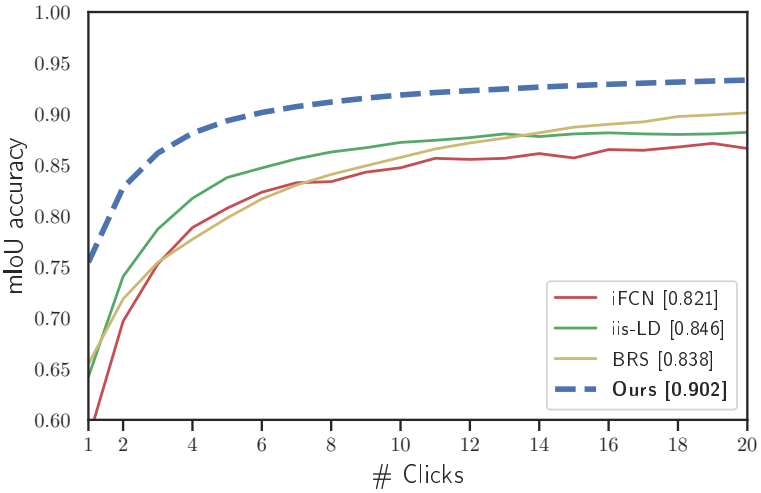


Fig. 3.6 Comparison of mean IoU scores after n clicks for the SBD testset [63].

The Image Stream architecture is based on ResNet-50 with Group Normalisation and Weight Standardisation [122]. The weights are initialised by pre-training on ImageNet for classification [122]. Two modifications are made to the network. First, we increase the number of input channels from 3 to 9 to allow for the six extra click input maps. Second, we remove the striding from 'layer 3' and 'layer 4' of ResNet-50 and increase the dilation to 2 and 4 respectively. This increases the output resolution from the Image Stream by 4 times. We experimented with replacing ResNet-50 with ResNet-101 or ResNet-101 pretrained for semantic segmentation, but we did not observe any noticeable improvement.

Because our training regime requires a batch size of 1, a crucial tweak is that we use Group Normalisation (GN) [158] (32 channels per group) with Weight Standardisation [122] instead of Batch Normalisation.

The Interaction Stream encodes the previous mask together with the click maps. It has six convolutional layers (see Table 3.1), each followed by the LeakyReLu activation function [103]. Three of the layers are strided convolutions with stride two, so the output resolution matches that of the Image Stream.

The Decoder concatenates features from both streams and passes them to a Pyramid Pooling layer [167]. The pooled features are fed into a decoder which contains eight convolutional layers interleaved with three bilinear upsampling layers and skip connections from both streams (see Table 3.2). The output of the decoder is at full resolution.

Fine Details are then extracted by appending a Guided Filter layer [156]. This further refines the mask details and allow us to obtain a transparency mask that can capture subpixel features, such as hair strands, as well as transparency, such as in motion blur. This means that the predicted mask is now a proper transparency mask that can be subsequently used for compositing the selected object onto a different background.

The full source code for testing can be found on the project webpage[1].

3.2.2 Training

Training Schedule

A fundamental deviation from previous works is that we train our network, image by image, click by click. Whereas in prior works the clicks are first bundled together, we propose to introduce the clicks sequentially, starting from a single click, and adopting the same sequential

[1] Project webpage: https://github.com/MarcoForte/DeepInteractiveSegmentation.

Table 3.1 Network architecture for the Interaction Stream Encoder. **C** is the number of output channels, **S** is the convolution stride. All convolutional layers are followed by a LeakyRelu activation [103].

Interaction Stream Encoder Layer	C	S
Concat: Clicks & Previous Mask	7	-
conv1	64	1
conv2	128	2
conv3	256	1
conv4	256	2
conv5	256	2
conv6	256	1

Table 3.2 Network architecture for the decoder. **C** is the number of output channels, **GN** indicates if the LeakyReLu was followed by a Group Normalisation [158] layer, **Up** indicates if the layer was followed by 2x bilinear upsampling. All convolutional layers except 'conv7' are followed by LeakyReLu activations.

Decoder Layer	C	GN	Up
Concat: Features from PPM & Interaction Stream 'conv6'	3328		
conv1	256	✓	
conv2	256	✓	✓
Concat: ResNet 'layer 1' & Interaction Stream 'conv4'	768		
conv3	256	✓	✓
Concat: ResNet 'conv 3' & Interaction Stream 'conv2'	384		
conv4	64	✓	✓
Concat: Image & Clicks	73		
conv5	32		
conv6	16		
conv7: (1x1, clip(0,1))	1		

scheme used to evaluate interactive segmentation algorithms [14, 69, 83, 91, 92, 105, 113, 162](See Figure 3.7, Algorithm 1).

For each click, we first look at the previously predicted map, and segment the mislabelled pixels into connected regions. We place the click at the centre of the largest incorrect region, so as to maximise the Euclidean distance to both the region boundary and the sides of the image. For the first click, the prediction map is set to zero. The loss is then computed after the placement of each click and the weights are updated through back-propagation before the next click.

By iterating through each click, we are essentially combining the loss for the entire range of clicks, hence effectively computing the area under the accuracy per clicks curve (as seen in Figures 3.4, 3.5 and 3.6). This has the advantage that our training loss matches the final metric used for evaluation. Also, this means that we are optimising for the whole range of clicks. In practice we found that using 4 clicks per image was optimal. The overall performance did not improve when placing more than 4 clicks but started to degrade when using 3 or less clicks.

Loss

The loss for a particular image is defined, as in IIS-LD [91], as the sum of the soft-IoU loss and a click location loss:

$$L(\alpha) = 1 - \frac{\sum_i \alpha_i \hat{\alpha}_i}{\sum_i \max(\alpha_i, \hat{\alpha}_i)} + \sum_{c \in \text{Clicks}} (\alpha_c - \hat{\alpha}_c)^2 \tag{3.1}$$

where α_i and $\hat{\alpha}_i$ are the mask prediction and ground truth mask values at pixel i. We found that using the soft-IoU led to visually and quantitatively better results than when using the binary cross entropy (BCE) loss. Segmentation results with BCE tend to show more spurious isolated regions than with the soft-IoU loss.

Hyper-Parameters

The ResNet-50 is initialised by pre-training on ImageNet [122]. We train our model with the RAdam optimiser [100] with a learning rate of 10^{-5}, with a single image in each batch. The learning rate is reduced by a factor of 0.1 at epochs 14, 17 and 20, and training completes after 25 epochs. We apply weight decay of $0.005, 10^{-5}$ to convolutional weights, and the GN parameters respectively. Images are cropped at a fixed size of 352×352 pixels, and we use horizontal flipping, gamma augmentations, and brightness augmentations.

3.3 Evaluation

3.3.1 Benchmarks

We evaluate our work across three publicly available segmentation datasets. GrabCut [131] is a 50 image dataset with 50 ground truth masks, although pixels in a thin band around the object boundaries are not labelled. Berkeley [107] is a 96 image dataset with 100 ground truth masks, the masks are of the highest quality of all three datasets. SBD [63] validation set contains 2,820 images with multiple labels per image. The labels span 20 semantic classes.

3.3.2 Comparison to the State of the Art

In this section, we use two metrics to demonstrate the superior performance of our algorithm over existing work.

Quick Selection. On Table 3.3 we report the mean number of clicks that it takes to reach the customarily used 85%/90% mIoU threshold, known as the Number of Clicks metric(NoC @ x%). Our algorithm reaches 90% mIoU accuracy in 2.54 clicks on GrabCut and 3.53 clicks on Berkeley. It reaches 85% mIoU in 3.90 clicks on the SBD test set. These are all significant improvements over the state of the art, at least a reduction of one click on each dataset.

We primarily use this metric in order to compare with results reported from previous methods. As seen in Figure 3.2 the segmentation quality at 90% mIoU is not sufficient for image editing, and as seen in Figures 3.4–3.6 many methods plateau in accuracy around those thresholds. So we believe it is more important to measure accuracy for a wide range of clicks.

Accuracy per Click. On Figures 3.4–3.6 is reported the average segmentation accuracy (mean IoU) across the first 20 iterated clicks for state of the art algorithms on each dataset. The click placement strategy used in the evaluation is the same used by all other methods. In the legend we also report the area under curve (AuC) score across the full range of clicks (1-20). From the graph and the AuC score we see that our algorithm exceeds or matches all others, on all datasets, for every number of clicks. On the GrabCut graph we see that for the first click our method matches the accuracy of CAMLG-IIS [105] yet their method does not make good use of the corrective clicks compared to ours. For the Berkeley dataset we observe that RIS-net [92] matches our models accuracy at 20 clicks. We believe this is due to the objects with thin details in this dataset. Here RIS-net performs well since it has a subnetwork that processes patches around each click, which allows for higher resolution processing of fine details at the cost of compute time.
We believe the AuC metric is much more relevant for future works than the number of clicks,

hence we use it for our ablation study. The graphs show there is still large room for improvement in future work for both at early clicks and for reducing the plateau at later clicks, especially on the Berkeley and SBD datasets.

3.3.3 Ablation Study

We study the effectiveness of our contributions for the following scenarios:

- A baseline model for our method is based on the Image stream, pyramid pooling layer and decoder alone and does not include the Interaction Stream nor the Guided Filter Layer. It is trained using the standard bundled click strategy from Ning *et al*. [162].
- In $(+T)$, we change the training approach to our click by click iterative training scheme.
- In $(+T, +I)$, we also include the Interaction Stream $(+I)$, without the previous mask feedback (*i.e*. it is set to zero).
- In $(+T, +I, +F)$, the previous predicted mask is fed to the Interaction Stream.
- In $(+T, +I, +F, +G)$, our full model is completed by appending a Guided Filter Layer. Unless otherwise stated all results in this chapter refer to this model.

Accuracy per Click. On Figures 3.9, 3.10 and 3.11 are reported the mean IoU across the range of 20 clicks on the SBD dataset [63]. The area under the curve, with its confidence interval, is also compiled in Table 3.4. We plot the results from the state-of-the-art method BRS [76] as a reference.

We can see that our stripped down baseline model performs about on par with the state of the art, which points to the strength of our ResNet-50 based Encoder Decoder architecture with Group Normalisation. Then, training the model with our interactive regime $(+T)$ gives a clear boost to accuracy for all clicks. Then adding the interactive stream $(+T, +I)$ gives a consistent improvement at higher numbers of clicks, where fine edits are made. Similarly, adding the feedback of the mask $(+T, +I, +F)$ improves results for later clicks on all three datasets. The Guided Filter layer gives improvements on the Grabcut [131] and Berkeley [107] datasets but not on the SBD [63] set. We believe this is because the labels on SBD do not follow the object boundaries closely (see discussion in section 3.4), but Guided Filter refines the prediction based on the image edges.

Corrective Click Accuracy. In Figure 3.8, we introduce a novel way to measure interactive segmentation performance. At each click iteration, we measure the proportion of correctly

predicted pixels within the previous incorrect region, which is where this new click was placed. In effect, this measures the networks ability to accurately respond to user inputs, something very important for the user experience. We compare ablated version of our model using this metric on the Berkeley dataset.

Interestingly, all methods perform equally for the first three clicks. However, the baseline model continuously drops in corrective accuracy after this. The more clicks are added, the less it responds to each click. Our interactive training improves this and has much higher corrective accuracy for all clicks, but it still drops over time. Our full model maintains a steady 50% corrective accuracy for each click, indicating it more consistently leverages each user input.

This novel metric must be interpreted with some nuance, and it is not a metric to measure the overall accuracy of the model like IoU. It simply evaluates the accuracy on the previous incorrect region. The important point we make is that our full model does not drastically decrease in correction accuracy compared to the baseline.
Three points are important to note. Firstly the corrective accuracy drops slightly for the initial clicks for all models; this is because the correction region gets progressively smaller, and small errors on the mask boundary have more impact on correction accuracy. Secondly, our full model has lower correction accuracy than the ours: $+T,I$ model, this is for a similar reason as the first, the full model is slightly more accurate, and so the correction regions are smaller. Thirdly the correction accuracy metric would still be 1.0 even if the mask was solely correct on the correction region and incorrect everywhere else. Our other metrics must accompany this metric to understand model accuracy and behavior thoroughly.

Reaching 99% Accuracy. On Figures 3.12, 3.13 and 3.14 we plot the proportion of images where the prediction accuracy surpasses 90-99% accuracy at 1,5,10 and 20 clicks. The higher the curve at each point the better. We see that by adding more clicks we not only raise the lower bar of accuracy but also the number of images reaching 99%. At 20 clicks, 100% of images exceed 95% accuracy, and 62% of images exceed 99% accuracy on the GrabCut dataset. The Berkeley and SBD datasets are more difficult, yet at 20 clicks our method still exceeds 95% accuracy for 81% and 55% respectively.

3.3.4 Qualitative Evaluation

In Figure 3.16 and 3.17 we compare the predictions of our baseline and full algorithms for six clicks on images from the GrabCut and Berkeley datasets. Note that the baseline approach sometimes fails to recover from poor initial guesses, whereas our iteratively trained network is better at correcting with each click.

In Figure 3.15 we see that, after 20 clicks, both models roughly select the object, but our final model is visibly more precise in these instances.

In Figure 3.18 we illustrate the behaviour of our method across the whole range of difficulties, we sampled images based on their accuracy at 20 clicks. The images are ranked from easiest (top) to hardest (bottom). We see that at 20 clicks our model is capable of good selection in all cases. For small thin objects however there is still some difficulty extracting all fine details.

3.4 Training on a Synthetic Dataset

The two most popular training sets used in interactive segmentation are PASCAL 2012 [43] and SBD [63]. They contain thousands of images labelled by annotators. However, these datasets are also quite imprecise, which is a problem when training very high-quality segmentation networks. Examples of ground-truth delineations can be found in Figure 3.19. Interestingly, the GrabCut benchmark defines a band of pixel around the object boundary that is excluded when measuring the accuracy. This is not the case with SBD and PASCAL, for which every pixel is evaluated. This partly explains why state-of-the-art techniques achieve higher levels of accuracy on GrabCut.

We can argue that whatever is drawn by the annotators should be considered as ground truth, and that there is no such thing as an incorrect labelling. After all, interactive segmentation algorithms are meant to assist the user achieve whatever arbitrary delineation they want. However, the labelled masks on these datasets would not be good enough in high-end image or video editing applications. These existing datasets are thus not representative of our final application.

We therefore propose to explore how training our network on an accurate and consistently labelled synthetic dataset would impact our overall performance.

3.4.1 Synthetic Dataset Construction

We constructed our synthetic training dataset in a similar way to the Deep Image Matting dataset from Xu *et al*. [161]. We sourced 15,000 images of objects with foreground colours and transparencies from the Internet. As illustrated in Figure 3.20, the images are synthesised on the fly during training by compositing one of these foreground objects onto a randomly sampled background. The background images come from MS-COCO [96] and ETHZ Synthesizability [38] datasets. The binary masks are obtained by thresholding the alpha matte at 0.5. A few examples of such composite training images are shown in Figure 3.21.

3.4.2 Synthetic Dataset Training Results

We propose to analyse the following three training scenarios:

1. our full model (ours:$+T,I,F,G$), trained on SBD as proposed in the previous sections
2. our full model (ours:$+T,I,F,G$ w Synth), solely trained on the synthetic dataset for 10 epochs at fixed learning rate 10^{-5}.
3. our full model (ours:$+T,I,F,G$ w Synth+FT), finetuning the previous model on SBD for 5 epochs with learning rate 5×10^{-6}.

The comparisons of the mean IoU over the GrabCut, Berkeley, and SBD datasets for these three scenarios is presented in Figures 3.22, 3.23 and 3.24.

By comparing the graphs between ours:$+T,I,F,G$ and ours:$+T,I,F,G$ w Synth we see that the former outperforms for initial clicks and the later outperforms for final clicks. We believe that ours:$+T,I,F,G$ is better for the initial clicks as it is trained on real images. The synthetic trained model ours:$+T,I,F,G$ w Synth will overfit the synthetic data's unnatural saliency, like different lighting, no shadows, different jpeg artifacts, different resolution, and different noise profile. On the other end of the IoU graph, we see the synthetic trained model outperforms the other. This is since there is sufficient click input at this level to know which object is being selected, and the differences are more on the boundary accuracy. Here the synthetic set has the advantage since the masks are perfectly pixel-accurate, and there are more masks with intricate boundaries.
We note that the difference between both training sets is most evident on the SBD benchmark, where using the synthetic dataset negatively impacts the performance. This makes sense as SBD has only 20 classes, thus training on the SBD training set gives a decisive advantage.

The initial clicks vs. final clicks difference is also visible when we look at the mask outputs of each model. In Figure 3.25 we show mask outputs at three clicks. Here the synthetic model struggles to know which object to mask, and it is more sensitive to edges in the image. We have a comparison at 20 clicks in Figure 3.26. Here the synthetic model is much more accurate, especially along the boundaries.

To get the best of both models, we train a model, ours:$+T,I,F,G$ w Synth+FT , by pre-training on the synthetic dataset and finetuning on SBD. In Table 3.5 we present the metrics of *NoC*, AuC and *>99% @ 20 clicks* for the GrabCut and Berkeley datasets. Here we see the benefits of our final model, which is best in almost all metrics.

3.5 Discussion

We have shown that our interactive segmentation method can reach levels of accuracy that can be as high as 99% mIoU. What can be done to achieve higher levels of accuracy? One limitation is that binary segmentation is not well defined for natural images. Pixels at the object boundaries are never either foreground or background but always a mix of both (*e.g.* hair, fur, motion blur, defocus, *etc.*). This mixture is what natural matting is trying to solve by defining transparency masks. Thus, instead of interactive segmentation, we should probably aim for interactive matting instead, at least on high-end applications.

One other limitation we observed in this work is that we are partly limited by the quality of existing training sets. The semantic segmentation datasets available for training and testing are based on real images but are of low quality. On the other hand, the synthetic datasets used in matting, and also in this work, are of high quality but are not based on real images. There is thus room for much better training sets. The same problem is true for the test sets used in the three popular benchmarks. Going beyond 95% accuracy on the SBD benchmark requires to match a ground truth that does not necessarily follow the actual object boundaries.

In the end, there seems to be two problems in one: rough segmentation and extracting fine details. It is difficult to solve for both in a single unified approach. We can point out to the BRS [76], or FCTSFN [69], which split their architectures into two parts: a core network that produces a rough segmentation and, appended to it, a refinement network to upsample the masks. Both networks are trained in stages. In our approach, we show that a single network can be used, but we still resort to a two-stage training when using the synthetic dataset. How to best schedule the training for rough and fine details seems to be key.

3.6 Conclusion

Existing deep-learning based interactive segmentation methods are of limited value in professional photo editing applications since they plateau in accuracy around 95% and yet artists need to achieve more than 99% accuracy. One reason for this plateau is that they insufficiently leverage user interactions. We proposed a novel single network architecture that better embeds the user interactions into two separate streams. Using a click by click training regime helps us improve the correlation between the click placements and the prediction errors. Our experiments show how each contribution improves the response to local corrections and mIoU accuracy across the full range of clicks. In comparison to existing approaches, our method achieves higher accuracy for all twenty measured clicks, across three benchmarks and we even achieve 99% accuracy for 62% of images on the GrabCut dataset within 20 clicks. We also make the

observation that the low quality of existing training sets limits the potential performance of current interactive networks. We show that introducing a more accurate synthetic training set can further improve the overall accuracy of our system, and this model reaches 99% accuracy for 74% of images on the GrabCut dataset within 20 clicks.

With this strong baseline for interactive segmentation but poor performance on hair and transparent regions turn our attention to the matting problem.

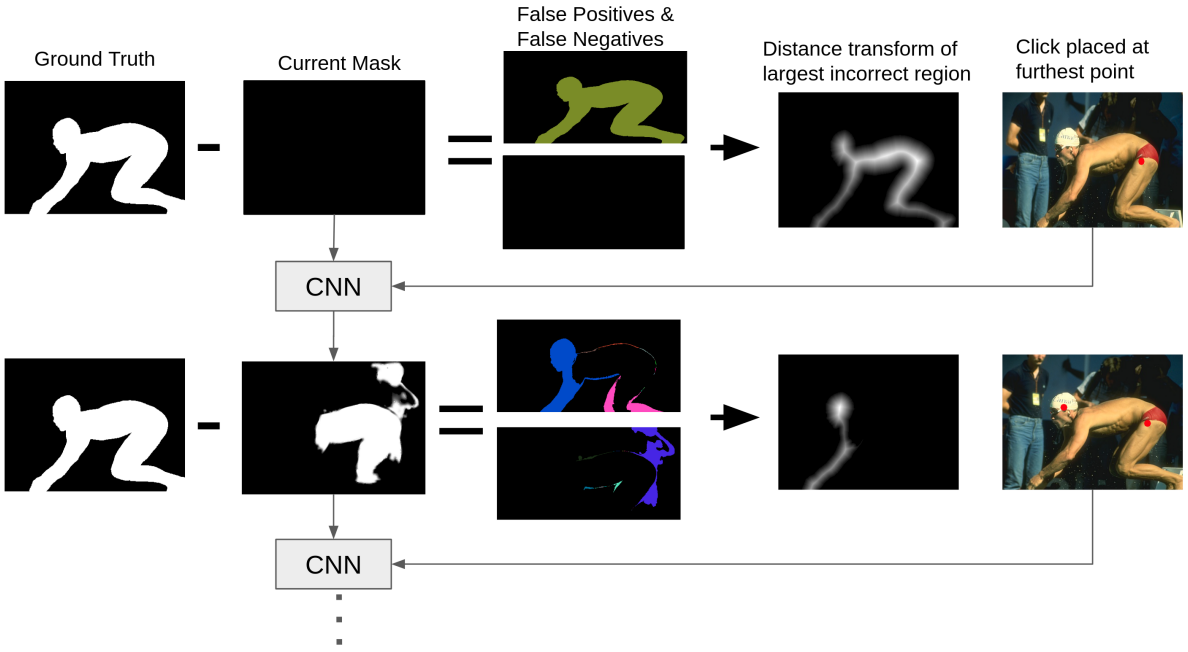


Fig. 3.7 Diagram of click placement algorithm.

Algorithm 1: Click placement algorithm

while $\#clicks < n$ **do**
  Identify pixels where the prediction is not equal to the ground truth;
  Split this into two binary masks, one for false positives and one for false negatives;
  Group the pixels in these binary masks according to connected components labelling ;
  Measure the size of each connected component ;
  Identify the largest such region;
  Measure the distance in the interior to the boundaries of this region;
  Identify the pixel furthest from boundary;
  If the region is a false positive, update the click map with a negative click. Positive click for a false negative.;
  Update prediction using new click map.
end

Table 3.3 The mean number of clicks required to achieve a certain mIoU accuracy on three datasets [63, 107, 131] by various algorithms. The best results are indicated in bold with the second best underlined.

Method	GrabCut NoC @ 90%	Berkeley NoC @ 90%	SBD NoC @ 85%
IFCN [162]	6.04	8.65	9.18
RIS-Net [92]	5.00	6.03	-
ITIS [104]	5.60	-	-
DEXTR [106]	4.00	-	-
VOS-wild [14]	3.8	-	-
FCTSFN [69]	3.76	6.49	-
IIS-LD [91]	4.79	-	7.41
CAMLG-IIS [105]	<u>3.58</u>	5.60	-
BRS [76]	3.6	<u>5.08</u>	<u>6.59</u>
Ours	**2.54**	**3.53**	**3.90**

Table 3.4 Mean IoU for the 1-20 clicks range on the different datasets [63, 107, 131] for each variant of our algorithm.

Method	GrabCut	Berkeley	SBD
ours: baseline	0.9121 ± 0.0078	0.9065 ± 0.0036	0.8416 ± 0.0008
ours:$+T$	0.9516 ± 0.0041	0.9306 ± 0.0022	0.8842 ± 0.0006
ours:$+T,I$	0.9486 ± 0.0038	0.9350 ± 0.0019	0.8941 ± 0.0005
ours:$+T,I,F$	0.9625 ± 0.0031	0.9396 ± 0.0018	0.9079 ± 0.0005
ours:$+T,I,F,G$	0.9628 ± 0.0025	0.9423 ± 0.0018	0.9026 ± 0.0005

Table 3.5 Comparison of training with SBD alone(ours:$+T,I,F,G$), with the synthetic dataset alone(ours:$+T,I,F,G$ w Synth), and with finetuning the synthetic model on the SBD set(ours:$+T,I,F,G$ w Synth+FT). We measure the proportion of images that surpass 95% & 99% accuracy at 20 clicks.

Method	GrabCut			Berkeley		
	NoC @ 90%	*AuC*	*>99% @ 20 clicks*	NoC @ 90%	AuC	>95% @ 20 clicks
ours:$+T,I,F,G$	2.54	0.963	62%	3.53	0.942	81%
ours:$+T,I,F,G$ w Synth	1.96	0.968	**74%**	3.46	0.943	82%
ours:$+T,I,F,G$ w Synth+FT	**1.8**	**0.974**	71%	**3.04**	**0.949**	**83%**

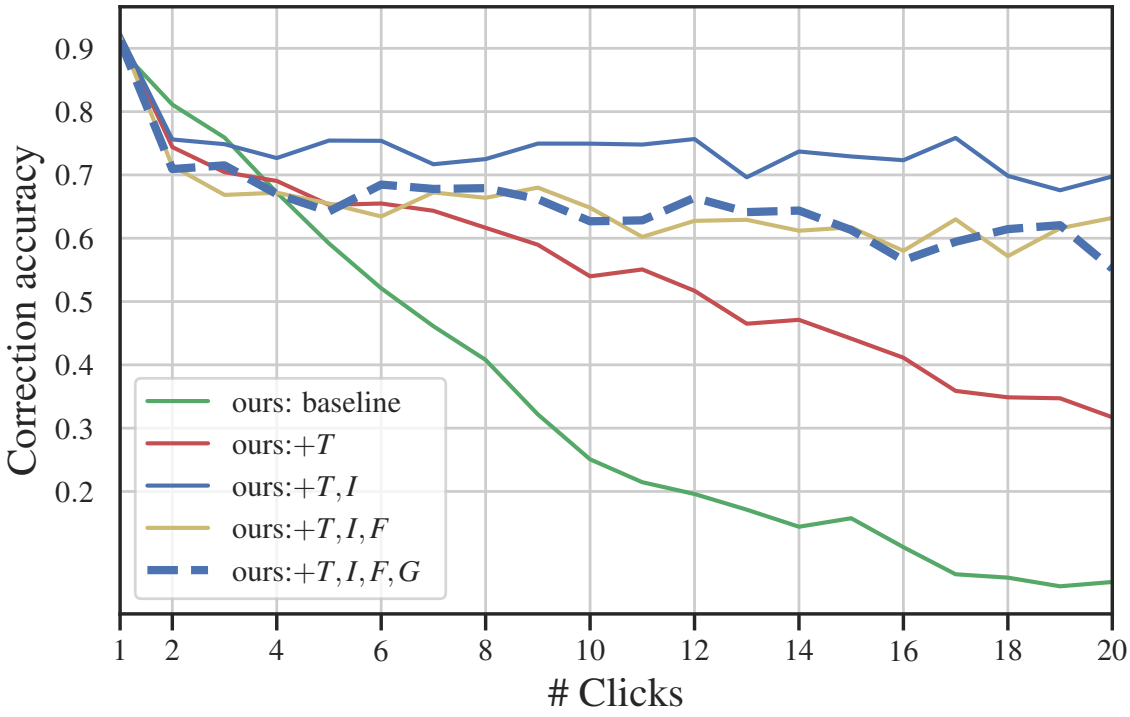


Fig. 3.8 Correction accuracy for each click. Computed on the Berkeley dataset [107]. The correction accuracy is defined as the proportion of pixels correctly predicted on the region where the click was placed (see text).

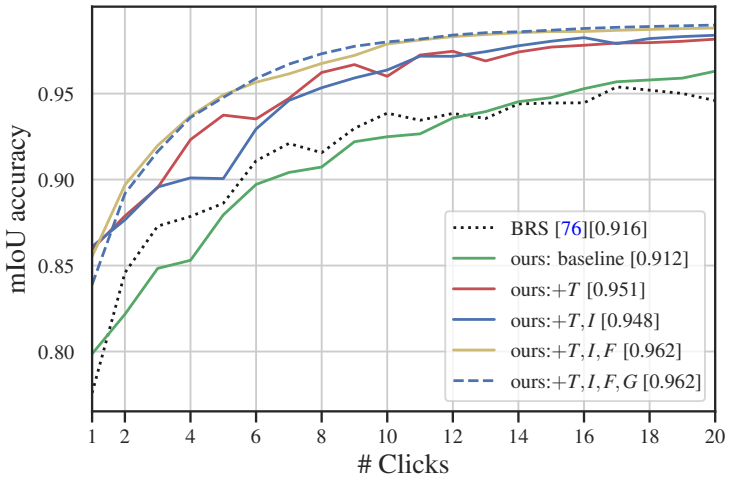


Fig. 3.9 Comparison of mean IoU scores after n clicks for the GrabCut testset [131].

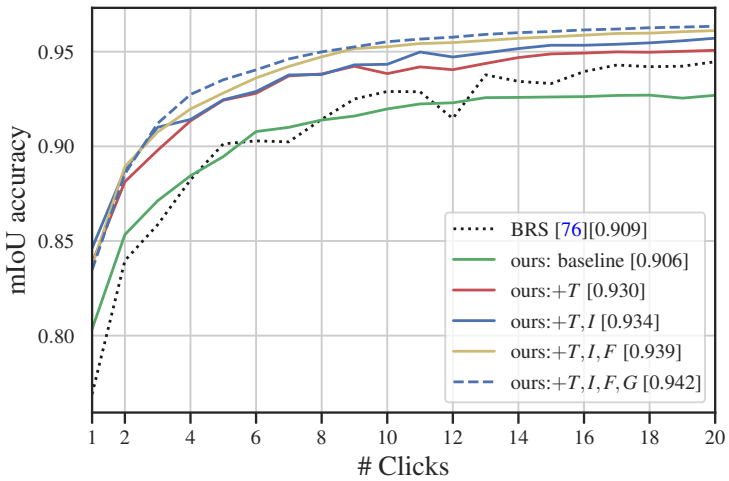


Fig. 3.10 Comparison of mean IoU scores after n clicks for the Berkeley testset [107].

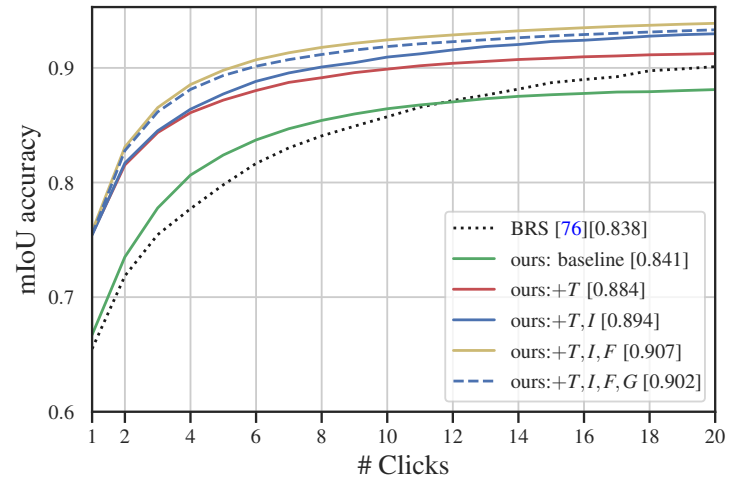


Fig. 3.11 Comparison of mean IoU scores after n clicks for the SBD testset [63].

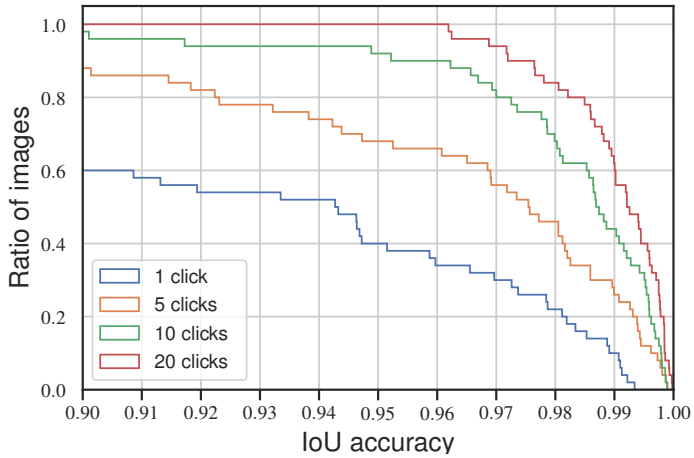


Fig. 3.12 Proportion of images surpassing *x%* mIoU accuracy after *n* clicks for the GrabCut testset [131].
62% of predictions exceed 99% mIoU accuracy after 20 clicks.

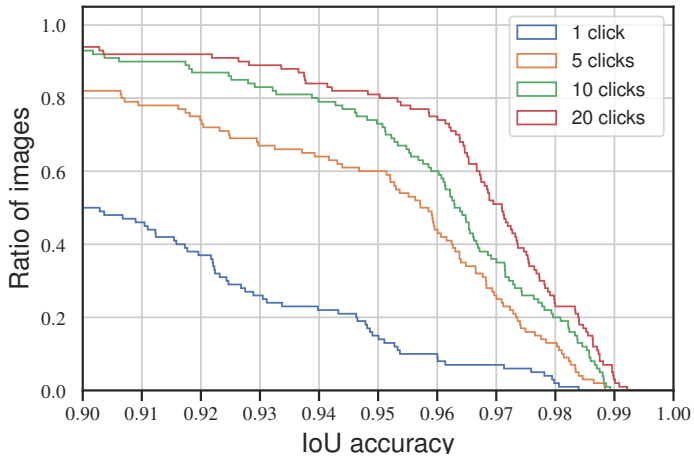


Fig. 3.13 Proportion of images surpassing *x%* mIoU accuracy after *n* clicks for the Berkeley testset [107].
81% of predictions exceed 95% mIoU accuracy after 20 clicks.

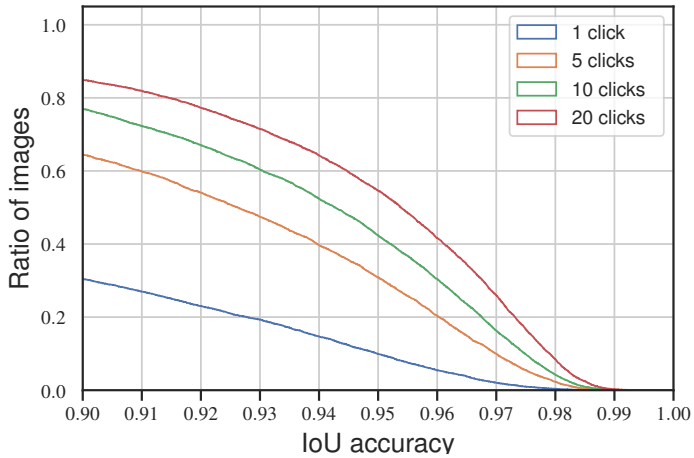


Fig. 3.14 Proportion of images surpassing *x%* mIoU accuracy after *n* clicks for the SBD testset [63].
55% of predictions exceed 95% mIoU accuracy after 20 clicks.

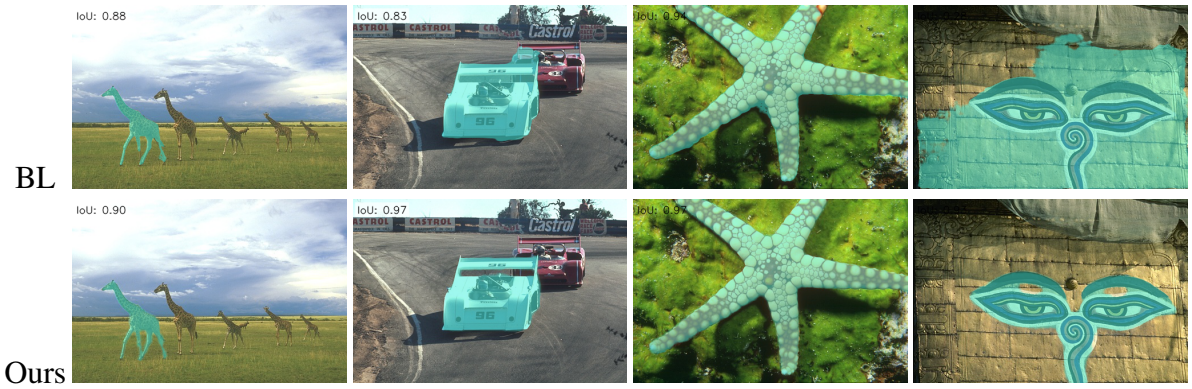


Fig. 3.15 Qualitative comparison between our baseline and our model after 20 clicks. Object masks are highlighted in cyan.

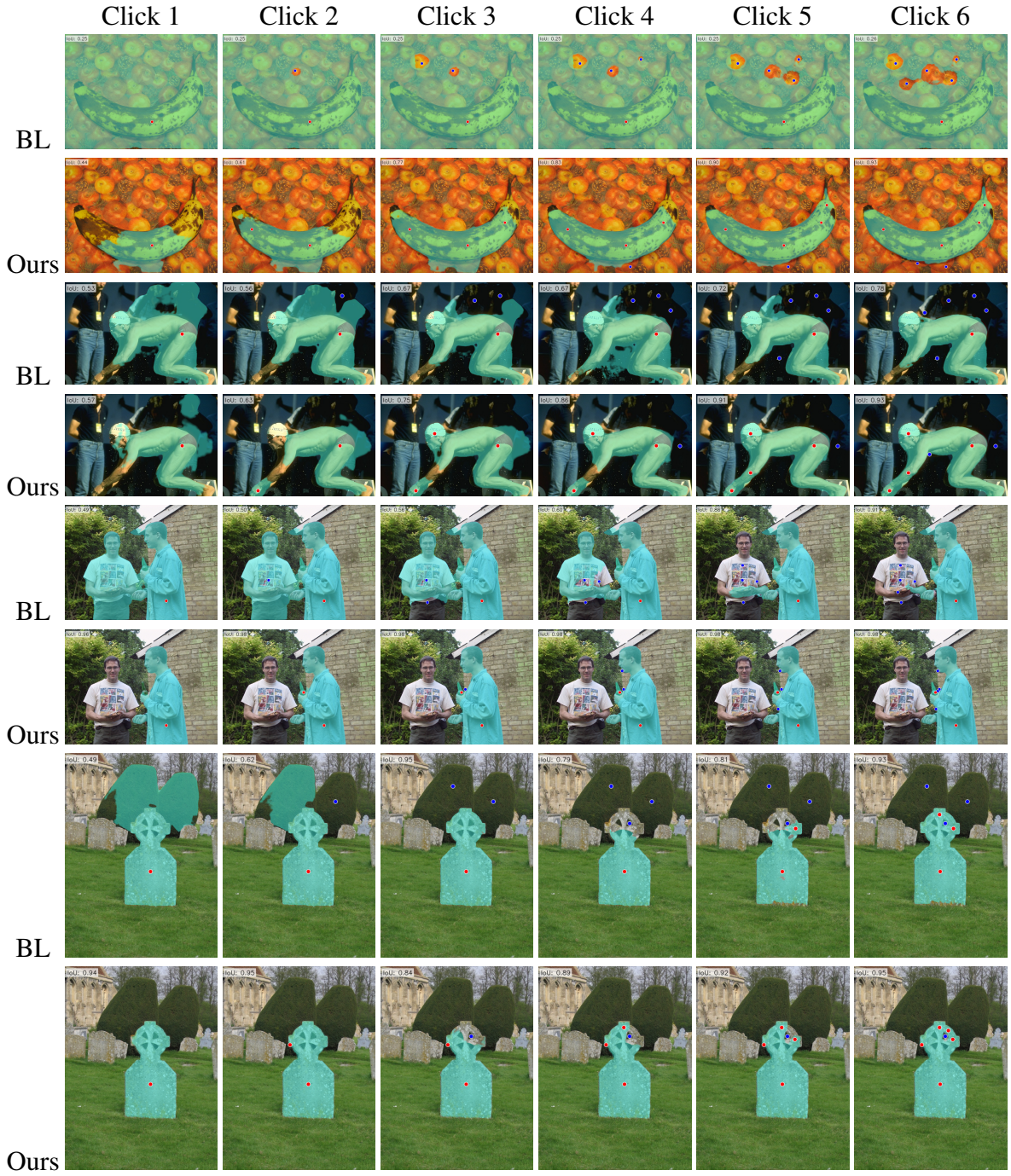


Fig. 3.16 Qualitative comparison between our baseline (BL) and our model across the first 4 clicks. Positive clicks are in red, negative in blue, object masks are highlighted in cyan. Images are from the GrabCut Dataset.

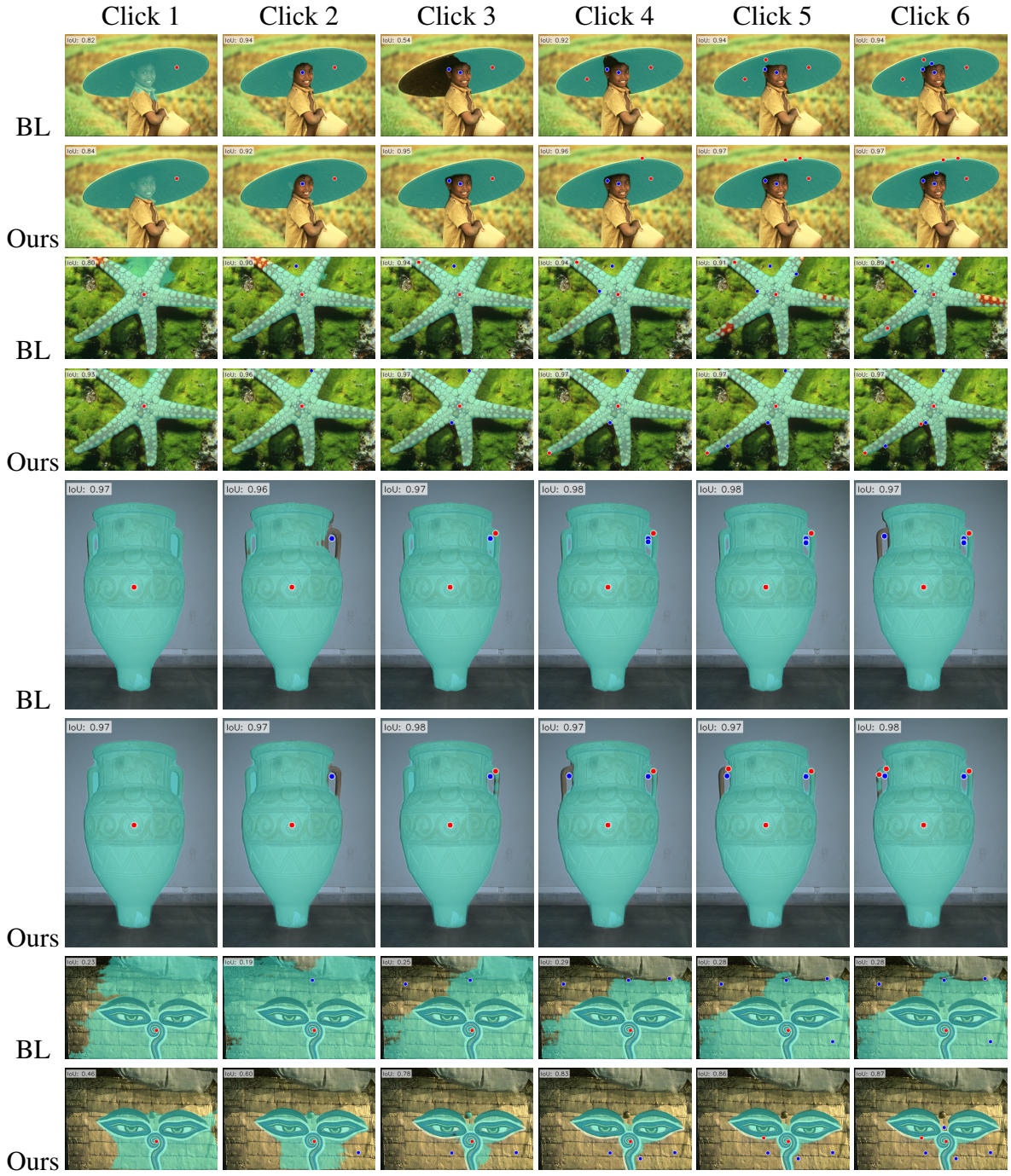


Fig. 3.17 Qualitative comparison between our baseline (BL) and our model across the first 6 clicks. Positive clicks are in red, negative in blue, object masks are highlighted in cyan. Images are from the Berkeley Dataset.

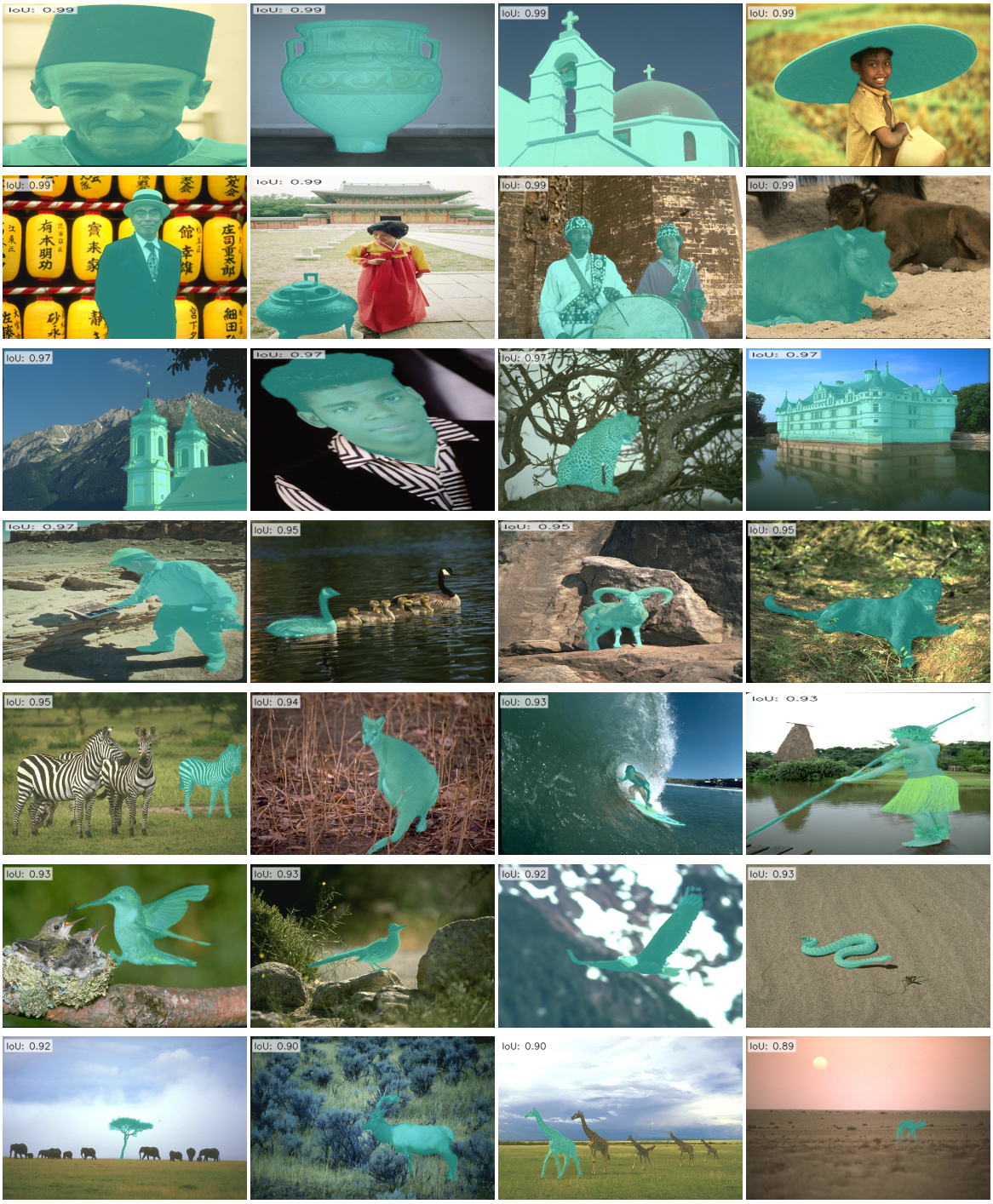


Fig. 3.18 Example predictions from our method at 20 clicks on the Berkeley dataset. Examples are ordered from the easiest (top) to the most difficult images (bottom), based on the mean IoU accuracy over the 15-20 click range. Positive clicks are in red, negative in blue, object masks are highlighted in cyan.

Fig. 3.19 Example of poor boundary and hole labelling in the SBD train and validation dataset.

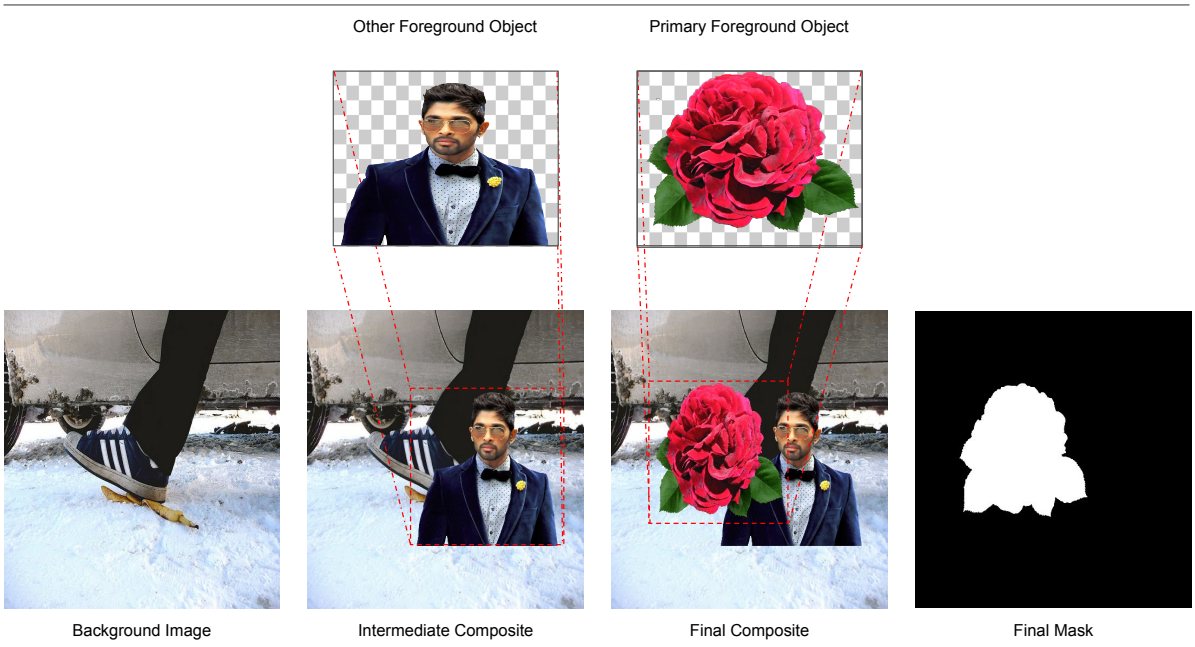


Fig. 3.20 Creating an image of the synthetic dataset.

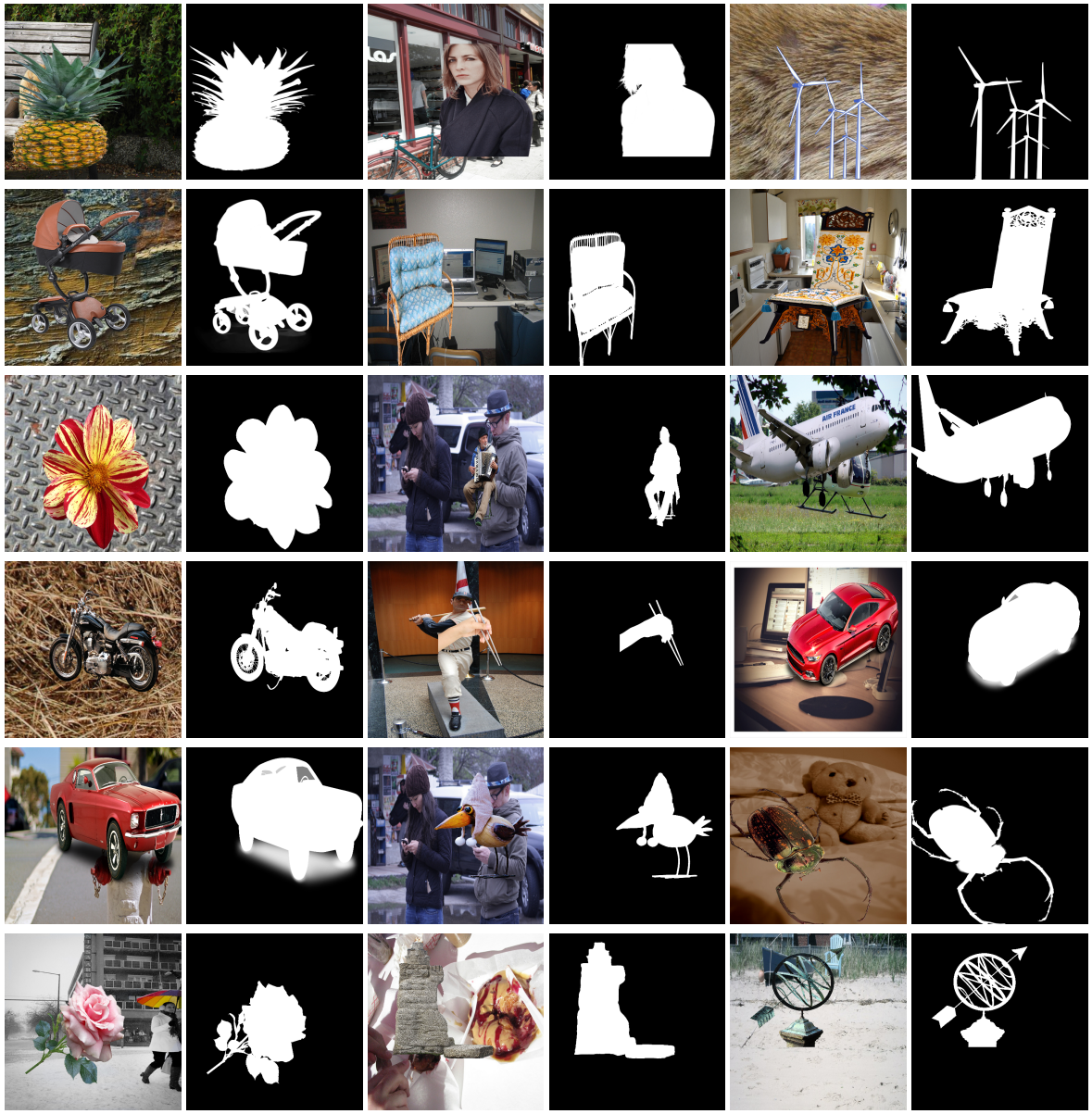


Fig. 3.21 Synthetic Dataset Examples: Variety of training images and their corresponding masks, generated from our Synthetic Dataset.

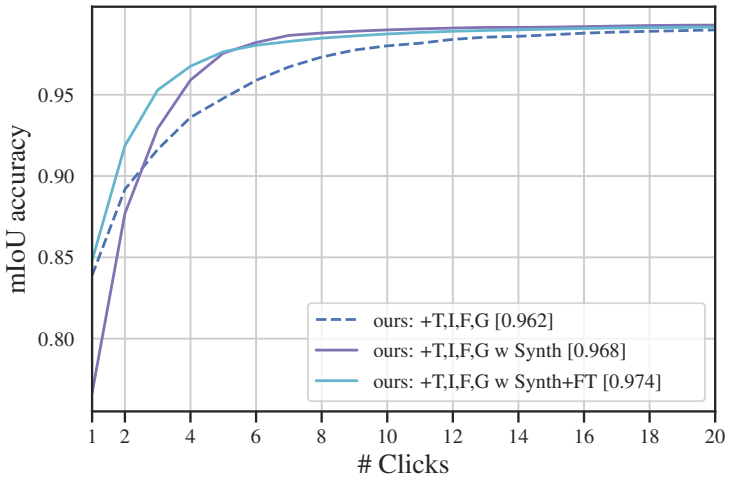


Fig. 3.22 Training with our Synthetic dataset. Mean IoU scores after n clicks for the GrabCut testset [131].

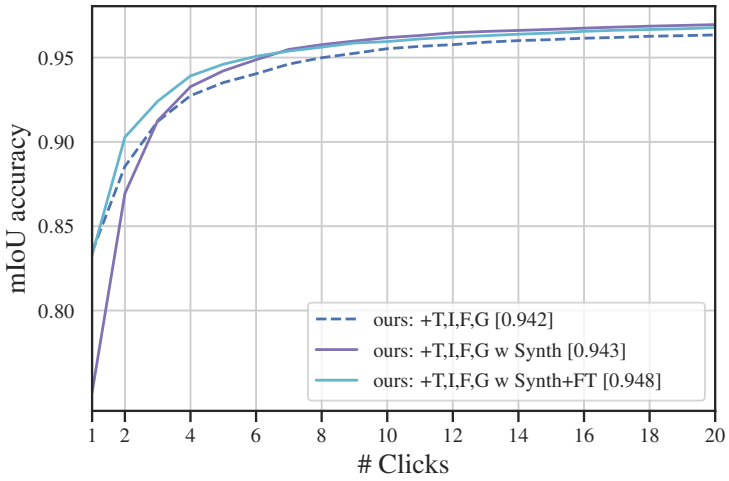


Fig. 3.23 Training with our Synthetic dataset. Comparison of mean IoU scores after n clicks for the Berkeley testset [107].

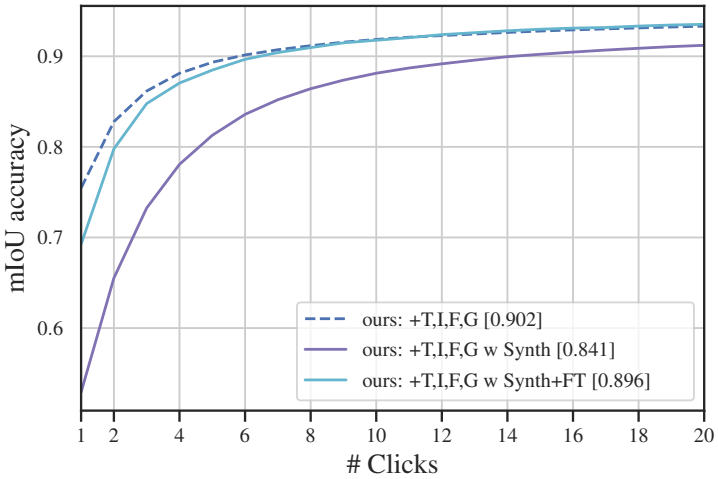


Fig. 3.24 Training with our Synthetic dataset. Comparison of mean IoU scores after n clicks for the SBD testset [63].

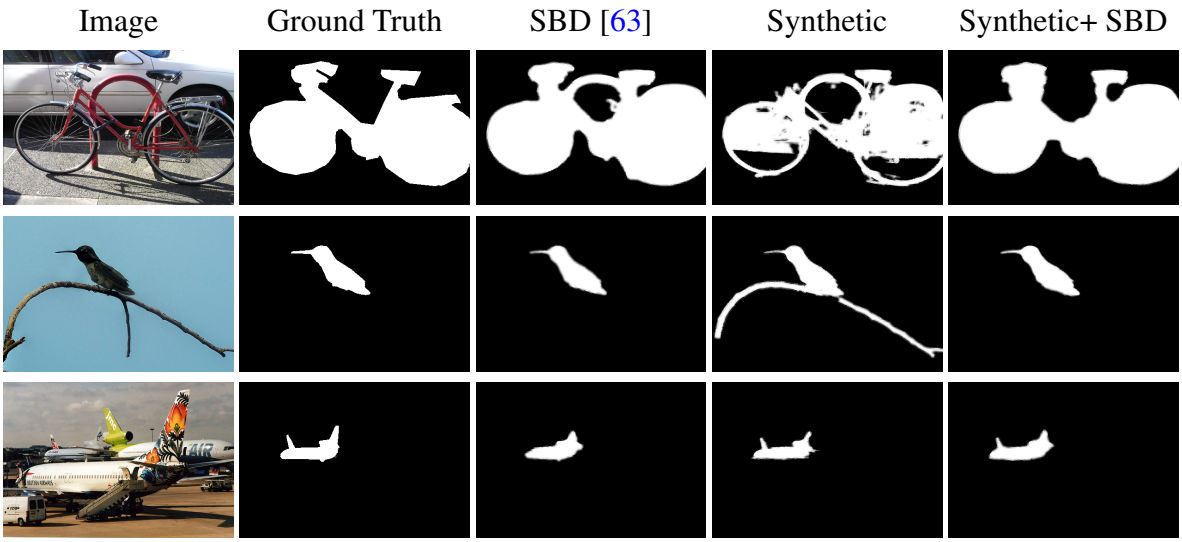


Fig. 3.25 Segmentation examples at 3 clicks, for models trained with SBD, trained with the synthetic dataset alone, and trained with the synthetic set and finetuned on SBD: Images are from the SBD validation set.

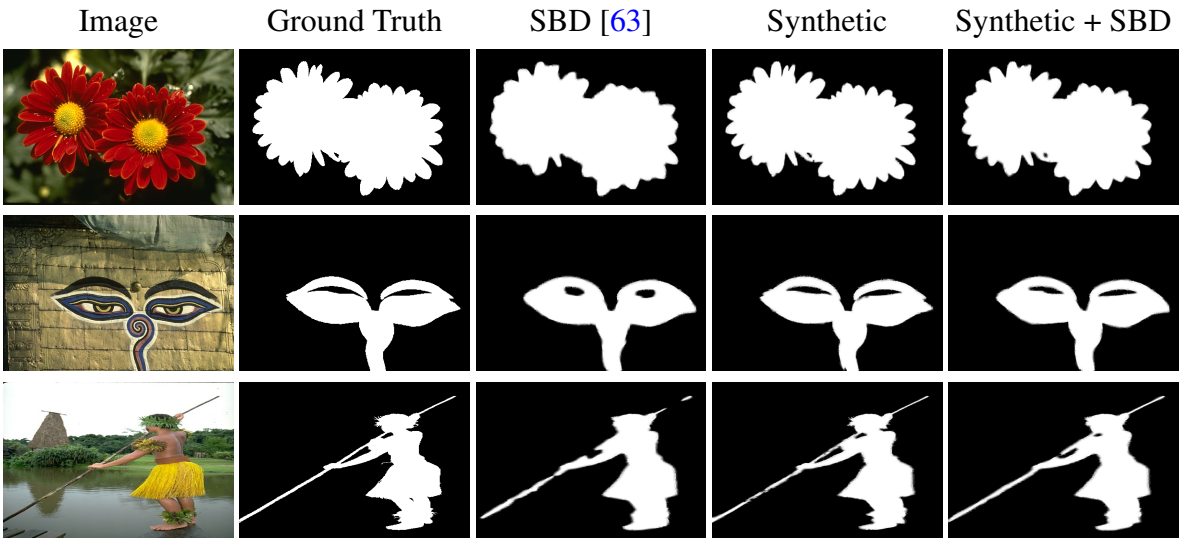


Fig. 3.26 Example segmentations at 20 clicks, of models trained with SBD, trained with the synthetic dataset alone, and trained with the synthetic set and finetuned on SBD: Images are from the Berkeley dataset.

Chapter 4

Deep joint foreground, background estimation and alpha matting

In the previous chapter, we focused on producing object mattes with minimal input. However, compelling visual effects depend on the quality of the final composite, which relies on extremely accurate alpha matte and foreground colours. In this chapter, we present changes to the approach in our previous chapter in order to optimise for composite quality. We build off learnings from the previous chapter in network architecture, training scheme and attention to the training dataset.

As discussed previously, compositing an object into a background scene B requires both its transparency α and foreground colours F to create the final composited image $C = \alpha F + (1 - \alpha)B$. Compositing artists have observed that the quality of the intermediate composite, αF, is more indicative of final composited image quality than the quality of alpha alone. This is illustrated in Fig 4.2. As observed in that example, if the foreground colours are inaccurate, then the composite can look poor even if no artefacts are visible in the matte.

Despite the importance of αF in practice, the research community has mostly focused on only predict the alpha matte and post-processing methods must then be used to recover the original foreground and background colours in the transparent regions. Recently two methods have shown improved results by also estimating the foreground colours, but at a significant computational and memory cost.

In this chapter, we propose a low-cost modification to alpha matting networks to also predict the foreground and background colours. We study variations of the training regime and explore a wide range of existing and novel loss functions for the joint prediction.

An article of this chapter has been released on arXiv.org [46]

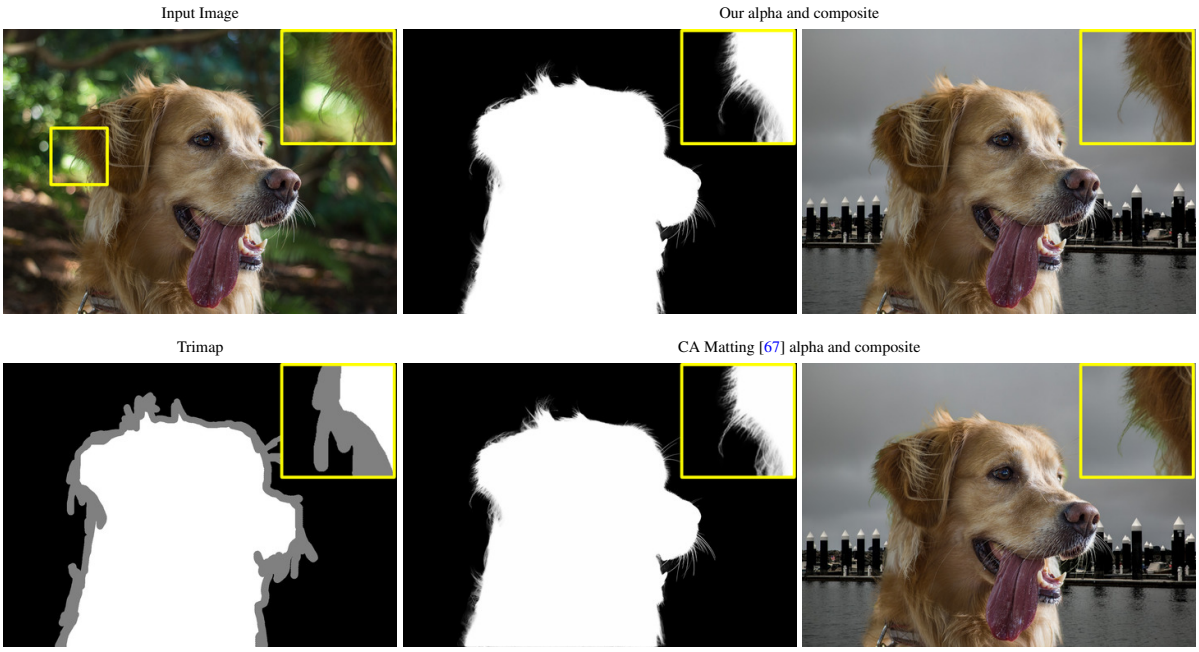


Fig. 4.1 We show our method better excludes background colours compared to the state-of-the-art-method for simultaneous prediction [67].

.

Our method achieves state-of-the-art performance on the Adobe Composition-1k dataset for alpha matte and composite colour quality. It is also the current best performing method on the alphamatting.com online evaluation.

4.1 Introduction

Estimating the α-matte is clearly an ill-posed problem as we have 3 non-linear equations for 7 unknowns, without any well defined spatial dependency to help us. The task is especially difficult when the foreground and background colours are similar or when the background is highly textured but recently, deep convolutional neural networks (CNNs) have progressed the state-of-the-art in α-matte prediction ([67, 102, 144, 160]) by better modelling natural image priors and α-mattes priors. These α-matting networks are based on the core architectures of semantic segmentation neural networks. This makes sense as α-matting can be seen as an extension of binary segmentation, where instead of estimating a binary $\{0,1\}$ label field we estimate a floating point $\alpha \in [0,1]$ field. For instance, the original *Deep Image Matting* paper of [160] adapted the FCN architecture of [136] to take as an input the original image and a trimap and as an output the α-matte instead of a binary mask.

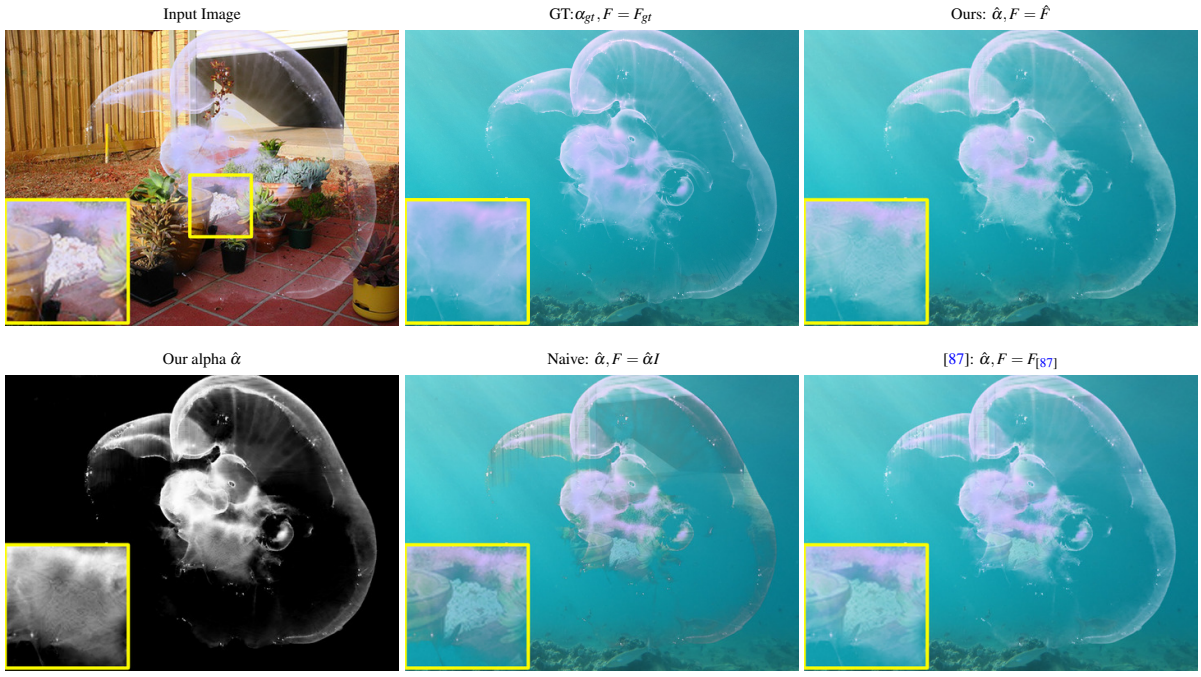


Fig. 4.2 We show the need for foreground prediction and compare our composite with a post-processing method [87]

.

As the level of details required in the output matte is much higher than in segmentation, most of the recent literature has focused on making architectural changes that can increase the resolution ability of the core encoder-decoder architecture (see [67, 101, 143, 160]). Curiously, most of these methods only estimate α, and not F or B, which are however also required by most applications. (We use α, F, B to represent the ground-truth and $\hat{\alpha}, \hat{F}, \hat{B}$ for the predicted values). Un-mixing the foreground and background colours is usually left as a post-processing step, (*e.g.* Levin *et al.* [87]). Recent works [67, 144] have started to recognise the importance of jointly estimating α, F, B, but the resulting architectures are computationally expansive. In this chapter, we propose a novel architecture for jointly estimating α, F, B and study the benefit of estimating F and B through the examination of different possible loss functions.

Whereas in binary segmentation the choice of Loss functions is relatively straightforward (*i.e.* cross-entropy or IoU), the nature of the α-matte gives us more options to consider. A few of the α losses proposed in previous works include: Huber [71], $\mathscr{L}_1$, $\mathscr{L}_1$ on the gradient, pyramid Laplacian, discriminative loss, *etc*. Also predicting F and B further extends the choice of possible loss functions and we propose to systematically study the merits of these different losses.

Parallel to these considerations on loss and architecture, interesting issues around training have also emerged. Indeed, as matting is an intrinsically harder task than segmentation, training can be long and trickier to setup correctly. Seemingly insignificant implementation details can turn out to be of critical importance. A few research groups have reported difficulties in reproducing the results of Deep Image Matting [45, 78, 102]. We discovered that simply setting the batch size to 1 in [160] had a critical impact on the performance and could, by itself, explain the reported training failures.

Contributions. We propose therefore that, of equal importance to the core architecture design, are the considerations around training and loss functions, and the main contribution of this chapter is to propose a novel architecture for jointly estimating α, F, B and to systematically explore the impact of key choices in losses and training regimes. Our study include 17 experiments that contribute to three study areas:

1. a comparison of mini-batch and stochastic gradient descent and the use of batchnorm vs. groupnorm.

2. a study of the different α-matte losses ($\mathscr{L}_1$, gradient, laplacian pyramid, compositing loss).

3. a study of the potential benefit of also predicting F and B alongside α and the possible losses associated with this ($\mathscr{L}_1$ loss and exclusion loss).

Related works are presented in section 4.2, our proposed method is proposed in section 4.3. The experimental studies and their results are discussed in section 4.4. Our resulting network achieves state-of-the-art performance on the Adobe Composition-1k dataset for alpha matte and composite colour quality. It is also the current best performing method on the alphamatting.com online evaluation.

4.2 Related Works

Losses. The exact nature of the output α-matte is a key aspect of matting. On one hand, $\alpha = \{0,1\}$ represents a binary label field similar to the one found in binary segmentation. On other hand, the α-matte is also a continuous field with values between 0 and 1, which shares some resemblance to natural images. This opens up a wide range of possible losses. In *Deep Image Matting*, Xu *et al.* [160] originally proposed a simple Huber loss on α. In later works [19, 67, 89] the $\mathscr{L}_1$ loss was preferred instead. The absolute values of α may however not be as important as the gradient of α. Indeed, errors in the reproduction of the hair strands shapes are more noticeable than slight errors in the overall opacity level. Gradient

fidelity is in fact one of the commonly used quality metric in image matting benchmarks. A basic $\mathscr{L}_1$ gradient loss has been proposed [144]. More effective, however, has been the use of the pyramid Laplacian loss L_{lap} (see Context Aware Matting [67]). This loss evaluates the gradient error at five scales, promoting global coherence. Beyond these set losses, Lutz *et al.* (2018) [102] have also proposed a discriminative loss on α as part of their GAN architecture.

The matting problem is however not limited to the α field alone as (α, F, B) are inter-dependent. In *Deep Image Matting*, Xu *et al.* [160] thus proposed to combine their Huber loss on α with a *compositing* loss $\mathscr{L}_c$, which is defined as the reconstruction error $\mathscr{L}_c(\hat{\alpha}) = \sum_i \|C_i - (\hat{\alpha}_i F_i + (1-\hat{\alpha}_i)B_i)\|_1$ when using the ground truth foreground and background colours F_i and B_i at pixel i. This loss has then been used in later works [102, 143, 144].

Foreground & Background Predictions. A few prior non-deep learning methods have proposed to predict F and B along side α [4, 51, 121] but it is only very recently that Hou and Liu (2019) [67] and Tang *et al.* (2019) [144] have started to look into jointly estimating fg/bg colours with alpha in CNNs. In [67], F and α are decoded in sequence from the same shared decoder. In [144], the estimation is also done in the sequence B, F and α, using this time three full separate encoder-decoder networks. The main drawback here is that sequential estimation depends on the success of each of the individual predictions. Also, stacking full encoder-decoders as in [144] results in a very deep and large networks.

These works have chosen the $\mathscr{L}_1$ loss on the predicted F and B. In [67], they also use a VGG16 Features Loss on the predicted pre-multiplied Foreground $\hat{\alpha}\hat{F}$.

Also predicting $\hat{B}$ allows [144] to introduce a new composition loss for the predicted $\hat{F}$ and $\hat{B}$, based on the ground-truth values of α: $\mathscr{L}_c(\hat{F}, \hat{B}) = \sum_i \|C_i - \left(\alpha_i \hat{F}_i + (1-\alpha_i)\hat{B}_i\right)\|_1$.

Other losses are however possible. One particular loss we want to investigate is the exclusion loss ($\mathscr{L}_{\text{excl}}(\hat{F}, \hat{B}) = \sum_i \|\nabla \hat{F}_i\|_1 \|\nabla \hat{B}_i\|_1$), which has been introduced in a similar form in the reflection removal literature [165] to enforce a clean separation between F and B and avoid that structures of the original image to leak into both F and B.

Novel contributions over Context-Aware Matting, Hou and Liu (2019) [67]. The Context-Aware Matting paper shares similarities with our work in the goal of joint foreground and alpha matte prediction. We have made several key novel contributions that build upon and challenge their approach. Primarily we show that a large side network for joint foreground prediction is not necessary. We show improved results with a lighter model that requires only a single extra channel for foreground prediction. We also show that joint background estimation is possible and beneficial with our FBA Fusion 4.3.4. We review their work on loss functions for alpha matting and show that the Laplacian loss is very beneficial. We investigate further the choice of

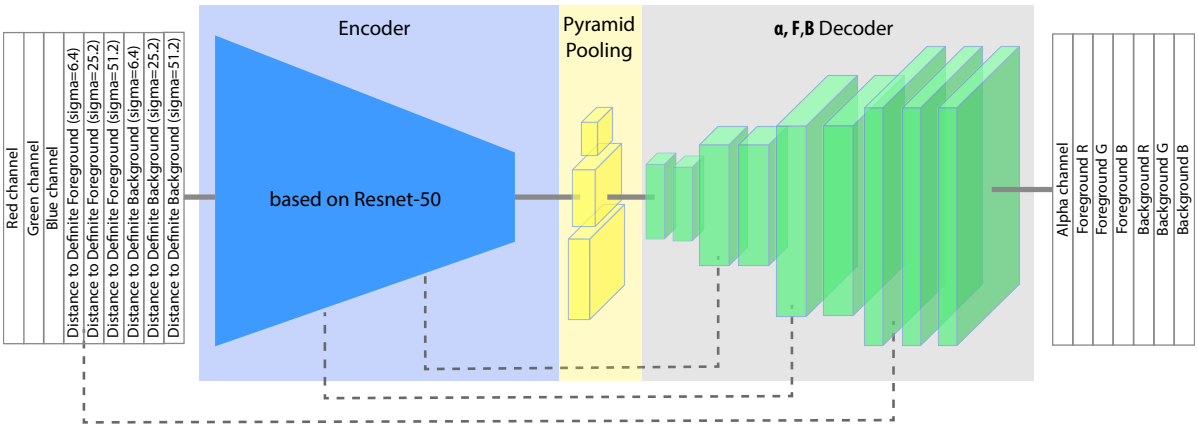


Fig. 4.3 The architecture of iMatting follows the ResNet-50 encoder/decoder of [46], with a Pyramid Pooling Module and a dedicated decoder. The input include the RGB channels of the original image, as well as the distance maps to the definite Fg/Bg user clicks. The output jointly predicts the alpha matte and the RGB channels of the Foreground and Background layers.

batch size and normalisation and find large improvements in accuracy. Lastly, we investigate the data augmentation during training and how it may be detrimental to the model. We propose a fix for this.

4.3 Proposed Approach

4.3.1 Network Architecture

Similarly to most previous works, we use an encoder-decoder with Unet [130] style architecture. The main difference is that our network also predicts F and B directly from this single encoder-decoder. Jointly estimating for α, F and B is motivated by the applications in compositing requiring an estimate for the foreground, and also by results in Multi-Task learning [132]. We choose to do this in the simplest way by extending the number of output channels from one to seven (1 for α, 3 for F and 3 for B). In contrast to previous works [67] and [144], which adopt a sequential prediction, our approach requires minimal extra parameters and avoids the delicate chaining of estimations.

The Encoder. The encoder is ResNet-50, and the weights are initialised by pre-training on ImageNet [39] for classification [122]. Two modifications are made to the encoder network.

First, we increase the number of input channels from 3 to 11 to allow for the trimap. We extend a one-hot encoding of the trimap using Gaussian blurs of the definite foreground and background masks at three different scales (in a similar way to the method of [83] in interactive

Table 4.1 Network architecture for the network decoder. **X** is the output spatial resolution, **C** is output number of channels, **GN** indicates if the LeakyReLu was followed by a Group Normalisation [158] layer, **Up** indicates if the layer was followed by 2x bilinear upsampling. Convolutional layers preceeding Group Normalisation are subject to weight standardisation [122]. All convolutional layers except 'conv7' are followed by LeakyReLu activations. All convolutional layers have 3x3 kernels with zero padding, except the pyramid pooling layers and the final layer which each have 1x1 convolutions.

Decoder Layers	**X**	**C**	**GN**	**Up**
Input: Features from Pyramid Pooling	32	3072		
conv1	32	256	✓	
conv2	32	256	✓	✓
Concat w: ResNet 'layer 1'	64	512		
conv3	64	256	✓	✓
Concat w: ResNet 'conv 3'	128	320		
conv4	128	64	✓	✓
Concat w: Image & Un-Normalized Image & Trimap Fg & Bg Mask	256	72		
conv5	256	32		
conv6	256	16		
conv7: (1x1: clip(α,0,1), sig(F), sig(B))	256	7		

segmentation). This encoding differs from existing approaches in deep image matting, as they usually encode the trimap as a single channel with value 1 if foreground, 0.5 for unknown and 0 for background.

Second, we remove the striding from 'layer 3' and 'layer 4' of ResNet-50 and increase the dilation to 2 and 4 respectively, in a similar way to what was proposed in AlphaGAN. In this way, the information can be processed at the highest scales without lowering the tensor resolution.

The Decoder. The tensor is then passed to a Pyramid Pooling layer [167], and then fed into a decoder which contains seven convolutional layers interleaved with three bilinear upsampling layers and skip connections. Full details of the decoder layers can be found in Table 4.1.

The Output Layer. The output is 7 channels for α, F, B. Experiments in section 4.4 show that clamping the values of α between 0 and 1 with a hardtanh activation as in Deep Image Matting [160] gives a small improvement over using a sigmoid function as in other previous works [19, 67, 89, 102, 143, 144]. The F, B channels also go through a sigmoid activation functions so a to also stay in the $[0, 1]$ range.

4.3.2 Batch Normalisation vs. Group Normalisation

Training Matting networks can take a long time and small implementation details can sometimes matter. One such detail that we discovered to have critical importance is the mini-batch size. All prior matting works have adopted a relatively small mini-batch size of around 6-16 [67, 89]. But through observing opensource re-implementations[72] of the *Deep Image Matting* method we found that a mini-batch size of one can greatly increase the network accuracy in the original *Deep Image Matting* [160] paper.

A mini-batch size of one is however incompatible with our ResNet-50 encoder as ResNet-50 uses Batch-Normalisation, which assumes i.i.d batches larger than size 1. We propose thus to use instead Group Normalisation (32 channels per group) with Weight Standardisation [122, 158].

4.3.3 F, B, α Losses

As discussed in Section 4.2, numerous losses have been proposed to improve the training of alpha matting networks. In this work we measure the impact of each of these losses individually, Table 4.3. By maximising the accuracy of our model for alpha matting we better validate the low-cost benefit of our modifications for joint foreground prediction. The final model is trained with a sum of all of these losses; the $\mathscr{L}_1$ loss on alpha $\mathscr{L}_1^{\alpha}$, the composition loss $\mathscr{L}_c^{\alpha}$, the gradient loss $\mathscr{L}_g^{\alpha}$, and the Laplacian pyramid loss that is computed over multiple scales s of the Laplacian pyramid L_{pyr}^{s}. For training the foreground and background we also use a sum of loss functions; $\mathscr{L}_1^{\text{FB}}$, Laplacian loss $\mathscr{L}_{\text{lap}}^{\text{FB}}$, compositional loss $\mathscr{L}_c^{\text{FB}}$, and a gradient exclusion loss $\mathscr{L}_{\text{excl}}^{\text{FB}}$. See Table 4.2 for the definition of each loss function. Our final loss function is

$$\mathscr{L}_{FB\alpha} = \mathscr{L}_1^{\alpha} + \mathscr{L}_c^{\alpha} + \mathscr{L}_g^{\alpha} + \mathscr{L}_{\text{lap}}^{\alpha} + 0.25\left(\mathscr{L}_1^{\text{FB}} + \mathscr{L}_{\text{lap}}^{\text{FB}} + \mathscr{L}_{\text{excl}}^{\text{FB}} + \mathscr{L}_c^{\text{FB}}\right) \tag{4.1}$$

4.3.4 F, B, α Fusion

One limitation of our joint prediction approach is that our predictions for $\hat{\alpha}$, $\hat{F}$ and $\hat{B}$ are decoupled and, even if they are based on the same decoder, the relationship given by the compositing Equation 1.1 is not explicitly enforced. We propose here a fusion mechanism based on the maximum likelihood estimate of $p(\alpha, F, B|\hat{\alpha}, \hat{F}, \hat{B})$. By assuming independence of the prediction errors and ignoring any spatial dependence between pixels, we can build a simplified likelihood model, derived from the individual and reconstruction losses:

$$p(\alpha, F, B|\hat{\alpha}, \hat{F}, \hat{B}) \propto p(\alpha|\hat{\alpha})p(F|\hat{F})p(B|\hat{B})p(\alpha, F, B) \tag{4.2}$$

Table 4.2 Training Loss Functions.

α Losses	F,B Losses
$\mathscr{L}_1^{\alpha} = \sum_i \|\hat{\alpha}_i - \alpha_i\|_1$	$\mathscr{L}_1^{\text{FB}} = \sum_i \left\|\hat{F}_i - F_i\right\|_1 + \left\|\hat{B}_i - B_i\right\|_1$
$\mathscr{L}_c^{\alpha} = \sum_i \|C_i - \hat{\alpha}_i F_i - (1-\hat{\alpha}_i)B_i\|_1$	$\mathscr{L}_{\text{excl}}^{\text{FB}} = \sum_i \|\nabla F_i\|_1 \|\nabla B_i\|_1$
$\mathscr{L}_{\text{lap}}^{\alpha} = \sum_{s=1}^{5} 2^{s-1} \left\|L_{\text{pyr}}^s(\hat{\alpha}) - L_{\text{pyr}}^s(\alpha)\right\|_1$	$\mathscr{L}_c^{\text{FB}} = \sum_i \left\|C_i - \alpha_i \hat{F} - (1-\alpha_i)\hat{B})\right\|_1$
$\mathscr{L}_g^{\alpha} = \sum_i \|\nabla\hat{\alpha}_i - \nabla\alpha_i\|_1$	$\mathscr{L}_{\text{lap}}^{\text{FB}} = \mathscr{L}_{\text{lap}}^{F} + \mathscr{L}_{\text{lap}}^{B}$

Assuming Gaussian distributions for the predictions and reconstruction errors:

$$p(F|\hat{F}) \propto \exp\left(-\frac{\|F-\hat{F}\|_2^2}{2\sigma_{FB}^2}\right) \qquad p(B|\hat{B}) \propto \exp\left(-\frac{\|B-\hat{B}\|_2^2}{2\sigma_{FB}^2}\right)$$
$$p(\alpha|\hat{\alpha}) \propto \exp\left(-\frac{(\alpha-\hat{\alpha})^2}{2\sigma_{\alpha}^2}\right) \qquad p(\alpha,F,B) \propto \exp\left(-\frac{\|C-\alpha F-(1-\alpha)B\|_2^2}{2\sigma_C^2}\right)$$

Essentially, we have simplified the model by ignoring the spatial losses (gradient and Laplacian pyramid) and replaced the $\mathscr{L}_1$ losses by $\mathscr{L}_2$ losses. This simplified model still yields a non-linear optimisation because of the reconstruction term is non-linear but we can adopt an iterative block solver approach. Starting at $\hat{F}^{(0)} = \hat{F}, \hat{B}^{(0)} = \hat{B}, \hat{\alpha}^{(0)} = \hat{\alpha}$, the update equations are as follows:

$$\hat{F}^{(n+1)} = \hat{F} + \frac{\sigma_F^2}{\sigma_C^2}\hat{\alpha}^{(n)}\left(C - \hat{\alpha}^{(n)}\hat{F}^{(n)} - (1-\hat{\alpha}^{(n)})\hat{B}^{(n)}\right) \tag{4.3}$$

$$\hat{B}^{(n+1)} = \hat{B} + \frac{\sigma_B^2}{\sigma_C^2}(1-\hat{\alpha}^{(n)})\left(C - \hat{\alpha}^{(n)}\hat{F}^{(n)} - (1-\hat{\alpha}^{(n)})\hat{B}^{(n)}\right) \tag{4.4}$$

$$\hat{\alpha}^{(n+1)} = \frac{\hat{\alpha}^{(n)} + \frac{\sigma_{\alpha}^2}{\sigma_C^2}(C-\hat{B}^{(n+1)})^{\top}(\hat{F}^{(n+1)} - \hat{B}^{(n+1)})}{1 + \frac{\sigma_{\alpha}^2}{\sigma_C^2}(\hat{F}^{(n+1)} - \hat{B}^{(n+1)})^{\top}(\hat{F}^{(n+1)} - \hat{B}^{(n+1)})} \tag{4.5}$$

These equations are related to the Bayesian Matting estimation scheme [34], except that the covariance matrices for F,B,C are not available. In practice, we found that 1 iteration through these block updates was enough. These equations give us a simple mechanism to fuse the 3 predictions by taking into account the matting model of Eq. 1.1. Our experiments in section 4.4, with $\sigma_C^2, \sigma_F^2, \sigma_B^2 = 1, \sigma_{\alpha}^2 = 10$, show that these updated estimates reliably produce better results.

4.3.5 Training Dataset and Modifications

The large Deep Image Matting dataset constructed by Xu et al. [160] has been instrumental for training state-of-the-art matting algorithms in recent years. The dataset is a collection of 431 foreground and alpha channel pairs. Training samples are created by using the alpha channel and the compositing equation 1.1 to composite the foreground onto a randomly chosen background from the MSCOCO dataset [96].

The issue is that the foreground images provided are only valid for non-zero alpha values and data augmentation during training can 'spill' invalid colours into these areas. For example, operations like resize and rotation re-sample pixels indiscriminately from both valid and invalid regions (see Figure 4.4). To remedy this, we re-estimated the foreground colours for all images, using Levin's F, B estimation technique [87]. This allowed us to extend the foreground estimation to the entire picture, and not only areas where $\alpha > 0$. That way, augmentation techniques become possible and makes this modified Deep Image Matting training set suitable for foreground prediction.

4.3.6 Training Details

Similarly to what can be found in the literature, our training patches of dimensions $640 \times 640, 480 \times 480, 320 \times 320$ are randomly cropped from unknown regions of the trimap. The training trimaps are generated from the ground truth α-mattes by random erosion and dilation of 3 to 25 pixels. For data augmentation, we adopt random flip, mirroring, gamma, and brightness augmentations. To further increase the dataset diversity, we randomly composite a new foreground object with 50% probability, as in [144]. The training data is shuffled after each epoch. Additionally, every second mini-batch is sampled from the $2\times$ image of the previous image, so as to increase the invariance to scale.

We use step-decay learning rate policy with the RAdam optimiser [100], with betas set to 0.9 and 0.999 respectively. The initial learning rate is set at 10^{-5} and then dropped to 10^{-6} at 40 epochs and fine-tuned for 5 more epochs. We apply weight decay of $0.005, 10^{-5}$ to convolutional weights, and the GN parameters respectively. The training process takes 16 days with a single 1080ti GPU. During inference, the full-resolution input images and transformed-trimaps are concatenated as 11-channel input and fed into the network.

4.3.7 Test Time Augmentation (TTA)

Convolutional neural networks are not invariant to flipping, rotation, zooming and cropping of the input. Randomly augmenting the training samples in these non-destructive ways has the

effect of enlarging the training set and improve the final network accuracy [89, 144]. However, training with augmented examples does not ensure total invariance to the augmentations. Predictions from augmented inputs are normally distributed, and the average of the transformations tends to the true value [149]. Test-time eight-way rotation of the input image and trimap was used by Tang *et al*. [143] but its influence on alpha matte accuracy was not published. We use a comprehensive test-time augmentation, combining rotation, flipping and scaling, and results are shown in Tables 4.5 and 4.4.

4.4 Experimental Results

In this section we report quantitative and qualitative results of our model. We perform an ablation study of our model and we also show state-of-the-art results in comparisons to existing matting methods. We use the Composition-1k [160] dataset for testing, as it contains 1000 testing images and other methods have reported their results on it for comparison. The dataset provides 50 ground truth foreground images and alpha mattes, and they are composed onto 20 different backgrounds each from Pascal [44].

We evaluate the results of both the alpha matte and also the combined $\hat{\alpha}\hat{F}$ foreground composite; supplemental results of the background prediction can be found in the supplemental material.

The alpha matte results are computed using four standard metrics [129], Sum of Absolute Differences (SAD), Mean Squared Error (MSE), Gradient Error (GRAD) and Connectivity Error (CONN). The gradient and connectivity metrics were shown at the time to be more aligned to human perception of matte quality. To measure foreground composite results we measure the MSE and SAD of the predicted $\hat{\alpha}\hat{F}$ and the ground truth αF.

Most of the ablation study is done over a 20 epochs training. The training was pushed to 45 epochs, including 5 epochs of fine-tuning at a 10^{-6} learning rate, on the most complete models.

Because of the scale of the experiments —each training taking two to four weeks to complete— the experiments have only been done on a single training run. Precise confidence intervals are thus not known but our earlier experiments seem to suggest that most of the observations made below are consistent.

4.4.1 Evaluating the Alpha Loss Functions

As discussed previously, four existing alpha matting losses emerge as reasonable options to be summed for training our network. These are the $\mathscr{L}_1$ loss, the composition loss, the laplacian loss and the gradient loss. Here we evaluate our network trained with combinations of these

Table 4.3 Ablation study of α-mattes results on the Composition-1k dataset.

Model	Norm.	Batch-Size	Loss	MSE	SAD	GRAD	CONN
Training for 20 epochs:							
(1)	BN	6	$\mathscr{L}_1^\alpha$	11.2	36.3	14.9	32.5
(2)	BN	6	$\mathscr{L}_1^\alpha + \mathscr{L}_c^\alpha$	9.1	34.5	15.0	31.3
(3)	BN	6	$\mathscr{L}_1^\alpha + \mathscr{L}_c^\alpha + \mathscr{L}_{\text{lap}}^\alpha$	7.4	33.5	12.9	28.5
(4)	BN	6	$\mathscr{L}_1^\alpha + \mathscr{L}_c^\alpha + \mathscr{L}_{\text{lap}}^\alpha + \mathscr{L}_g^\alpha$	8.1	36.3	13.8	32.0
(5)	GN	6	$\mathscr{L}_1^\alpha + \mathscr{L}_c^\alpha + \mathscr{L}_{\text{lap}}^\alpha + \mathscr{L}_g^\alpha$	10.3	36.2	15.1	32.0
(6)	GN	1	$\mathscr{L}_1^\alpha + \mathscr{L}_c^\alpha + \mathscr{L}_{\text{lap}}^\alpha + \mathscr{L}_g^\alpha$	7.2	32.8	13.3	28.6
(7)	GN	1	$\mathscr{L}_1^\alpha + \mathscr{L}_c^\alpha + \mathscr{L}_{\text{lap}}^\alpha + \mathscr{L}_g^\alpha + \text{clip}_\alpha$	6.9	31.2	12.9	27.1
Training for 45 epochs:							
Ours$_\alpha$	GN	1	$\mathscr{L}_1^\alpha + \mathscr{L}_c^\alpha + \mathscr{L}_{\text{lap}}^\alpha + \mathscr{L}_g^\alpha + \text{clip}_\alpha$	5.3	26.5	10.6	21.8

losses, models (1-4) in Table 4.3. We see that the compositing loss decreases errors for all metrics except for the gradient error. The laplacian loss proposed by Hou *et al*. [67] gives a significant reduction in errors across all metrics. We note this network, training and loss configuration is enough to achieve state-of-the-art results, see Table 4.5. The gradient loss proposed by Tang *et al*. [144] seems to increase the errors on all metrics. This was also reported in [89]. As this discovery was made late in our research, the gradient loss is included in all of our subsequent models. We leave the further examination of this loss to future work.

We also perform an experiment on the choice of alpha channel activation function (Table 4.3). A clipping activation was used in the original Deep Image Matting work [160], yet all subsequent works used a sigmoid activation, without reference to this change. However, we find that the sigmoid activation underperforms compared to the clipping activation (models (6) vs. (7)), see also Table 4.4 model (9) vs. (10). This also moves against the trend to use sigmoid for other image-to-image translation tasks [74, 164], however in these cases the rgb values are not usually 0 or 1 unlike alpha mattes, which are mostly 0 or 1. The sigmoid activation only achieves 0 and 1 for infinite valued inputs. Thus we use clipping activation for subsequent models.

4.4.2 Batch-Size and BatchNorm vs. GroupNorm

As discussed in the Proposed Approach section, we discovered that a mini-batch size of 1 greatly increases network accuracy for α-matting. We report the results of an experiment on this in Table 4.3. We use a model trained with a batch-size 6 and BatchNorm, model (4), as a baseline. BatchNorm is however, by definition, incompatible with training with mini-batch

sizes of 1, so we use Group Normalisation with Weight Standardisation (WS) [122, 158] for single image mini-batches, model (6).

We also train an intermediate model (5) with GroupNorm and WS to isolate the effects of batch-size. As expected, we see a significant reduction in error from models (4,5) to model (6) showing that a batch-size of 1 is best suited to α-matting. When comparing BN to GN with batch-size 6 it is clear that GroupNormalisation gives no hidden advantage to our hypothesis.

4.4.3 Evaluating the Impact of Jointly Estimating F, B, α

In Table 4.4 we examine the potential benefits of jointly estimating F, B, α over α alone. We measure the MSE and the SAD of the αF composite and the SAD of the alpha alone, across the Composition-1k testing set. We also record an ablation study of our method to show the benefit of each component. We see that using our foreground and background loss $\mathscr{L}^{\text{FB}}$ maintains alpha matte accuracy and allows for foreground prediction, model (6) vs. (9). Our foreground, background exclusion loss gives a minor benefit across all metrics, (8) vs. (9).

We observe an interesting trend in foreground prediction results from *Context-Aware Matting* [67], see Figure 4.6. The predicted foreground is quite poor in areas near the boundary or in highly transparent regions, and radically incorrect in nearby regions where $\alpha = 0$. This causes issues of colour bleeding artifacts when compositing onto novel backgrounds. We believe this is due to them only computing the foreground loss on the regions where $\alpha > 0$, and this leads to the network not prioritising foreground prediction in areas of very high transparency. We address this with our novel foreground dataset that has valid foreground ground truth for all pixels so we can compute the foreground prediction loss over the entire image. Our dataset also allows us to use the laplacian loss function for the foreground loss. In Table 4.4 we see that training with the $\mathscr{L}^{\text{FB}}$ throughout the entire image we improve the composite αF results, (10) vs. (11).

Our F, B, α fusion method, used at test-time, improves results for composite quality and for alpha matte quality, (11) vs. **Ours**$_{\text{FB}\alpha}$. This small improvement has been consistently observed in our experiments.

4.4.4 Comparison with Other Works

We compare our models **Ours**$_{\alpha}$ and Ours$_{\text{FB}\alpha}$ with existing state-of-the-art approaches to alpha matting.

Table 4.4 Ablation study of foreground results on the Composition-1k dataset. Here $\mathscr{L}^{FB} = \mathscr{L}_{\text{l}}^{FB} + \mathscr{L}_{\text{lap}}^{FB} + \mathscr{L}_{\text{c}}^{FB}$. In column two the * indicates that the $\mathscr{L}_{\text{l}}^{FB}, \mathscr{L}_{\text{lap}}^{FB}$ are computed over the entire image as opposed to just the unknown region of the trimap.

Model	$+\mathscr{L}_{FB}$	$+\mathscr{L}_{\text{excl}}$	output	αF		α	
				SAD	MSE	SAD	MSE
Closed-form Matting [87]				251.67	22.96	161.3	85.3
Context-Aware Matting [67]				70.00	11.49	38.1	8.9
Training at 20 epochs:							
(6)	N	N	sigmoid	-	-	32.8	7.2
(8)	Y	N	sigmoid	53.64	9.04	32.7	9.0
(9)	Y	Y	sigmoid	52.87	8.88	31.8	8.9
(7)	N	N	clip	-	-	31.2	6.9
(10)	Y	Y	clip	50.69	8.64	31.3	8.6
(11)	Y*	Y	clip	50.29	8.48	32.1	8.5
Training at 45 epochs:							
(11)	Y*	Y	clip	42.19	6.50	26.5	5.4
Ours$_{\text{FB}\alpha}$	Y*	Y	clip +fusion	39.21	6.19	26.4	5.4
Ours$_{\text{FB}\alpha}$	Y*	Y	clip +fusion +TTA	38.81	5.98	25.8	5.2

The Composition-1k Dataset

On the Composition-1k dataset, Table 4.5, each of our models significantly outperform previous approaches on all four metrics. We see that both our models **Ours**$_{\alpha}$ and **Ours**$_{\text{FB}\alpha}$ achieve state-of-the-art results. The model **Ours**$_{\text{FB}\alpha}$ which includes the foreground and background loss, with fusion, has improved connectivity and gradient measures but worse on the SAD measure. Our best model **Ours**$_{\text{FB}\alpha}$ **TTA** with test time augmentation achieves an average reduction in error of 34% over the best performing method for each metric, and the gradient error has the largest relative improvement. We give a qualitative comparison of alpha matte quality in Figure 4.5. Our method can separate very fine structures from the background even where the foreground and background colours are very similar. In Figure 4.6 we display the foreground, background and alpha outputs of our method and compare our predicted composite to that of Context-Aware matting [67]. We see that for CA Matting the foreground prediction can be quite poor in areas where alpha is close to zero, and colour spill is still visible in the $\hat{\alpha}\hat{F}$ composite. Our method, on the other hand, matches the ground truth much more closely. In the first image our background prediction successfully removes the glass from the image, which shows our method has the additional use for removing semi-transparent objects from images, *e.g.* watermarks.

Table 4.5 Alpha matting results on the Composition-1k test set [160].

Method	SAD	MSE x10³	Gradient	Connectivity
Closed-Form Matting [87]	168.1	91.0	126.9	167.9
KNN-Matting [26]	175.4	103.0	124.1	176.4
DCNN Matting [32]	161.4	87.0	115.1	161.9
Information-flow Matting [4]	75.4	66.0	63.0	-
Deep Image Matting [160]	50.4	14.0	31.0	50.8
AlphaGan-Best [102]	52.4	30.0	38.0	-
IndexNet Matting [101]	45.8	13.0	25.9	43.7
VDRN Matting [143]	45.3	11.0	30.0	45.6
AdaMatting [19]	41.7	10.2	16.9	-
Learning-Based Sampling [144]	40.4	9.9	-	-
Context-Aware Matting [67]	35.8	8.2	17.3	33.2
GCA Matting [89]	35.3	9.1	16.9	32.5
Ours$_{\alpha}$	26.5	5.3	10.6	21.8
Ours$_{\text{FB}\alpha}$	26.4	5.4	10.6	21.5
Ours$_{\text{FB}\alpha}$ **TTA**	**25.8**	**5.2**	**10.6**	**20.8**

The alphamatting.com Dataset

The alphamatting.com benchmark is an established online evaluation for natural image matting methods. It includes 27 training images and 8 testing images with 3 different kinds of trimaps, namely, "small", "large" and "user". Although there are only 8 testing images, it serves as an important tool for comparing alpha matting methods as the ground truth mattes are not released and the benchmark contains entries from most popular matting methods. In Table 4.6 we see that our method, **Ours**$_{\text{FB}\alpha}$ **TTA**, has the lowest average rank among the top four previous approaches. In particular we see that our method is a clear leader in the connectivity metric where deep-learning based matting methods have usually performed poorly. For comparison, Deep Image Matting [160] ranks best among these methods for connectivity yet is ranked 9th overall, behind Closed-Form Matting [87] which ranks 7th, our method ranks 1st. We also submitted our model **Ours**$_{\alpha}$ **TTA** for to evaluate the effect of our foreground and background loss and fusion, and we found that the **Ours**$_{\text{FB}\alpha}$ **TTA** performed best in all metrics except gradient. The full tables together with alpha matte predictions are available on alphamatting.com. In Figure 4.7 we show examples alpha matte predictions for the eight images of the alphamatting.com benchmark.

Table 4.6 Our scores in the alphamatting.com benchmark [129] together with the top-performing published methods. S, L, U denote the three trimap sizes, small, large and user, included in the benchmark. The scores denote the average rank of each method in these metrics across the 8 evaluation images.

Method	Grad.				MSE				SAD	Conn.
	overall	S	L	U	overall	S	L	U	overall	overall
Ours$_{\text{FB}\alpha}$ TTA	**1.4**	**1.4**	**1.3**	**1.6**	**1.7**	**1.6**	**1**	**2.4**	**1.9**	**6.7**
AdaMatting [19]	7.6	4.5	5.3	13	8	5.6	7.4	10.9	7	19.5
SampleNet Matting [144]	9.1	5.4	6.9	15	9	5.8	9.1	12.1	7.6	21.5
GCA Matting [89]	7.3	7.3	6.1	8.6	9.3	9.3	8.3	10.5	8.4	17.5
Deep Image Matting [160]	17.6	14.5	14.3	24	13	11.6	11.8	15.6	10	16.5

Real image Examples In Figure 4.8 we show examples of the αF composite predicted by our method and existing state-of-the-art methods. The images are from the real world image matting dataset [160].

4.5 Conclusion

This chapter presents a deep-learning method for simultaneously estimating the foreground, background and the alpha matte from a single natural image and trimap. Our primary contribution to the alpha matting training regime is to use a batch-size of one. This, combined with longer training at 45 epochs and TTA, has a bigger impact than the choices of loss functions. Finally, our proposed solution for simultaneous F, B, α prediction, can achieve state-of-the-art performance for all predictions, without increased computational or memory cost. Our reworked foreground dataset, new exclusion loss and fusion mechanism improve the final composite quality and can easily be incorporated in future works.

The previous chapter covered interactive segmentation with sparse inputs, and this chapter achieves accurate results for semi-transparent objects with dense trimap input. In the next chapter we conclude this thesis with a solution for interactive matting with sparse user input.

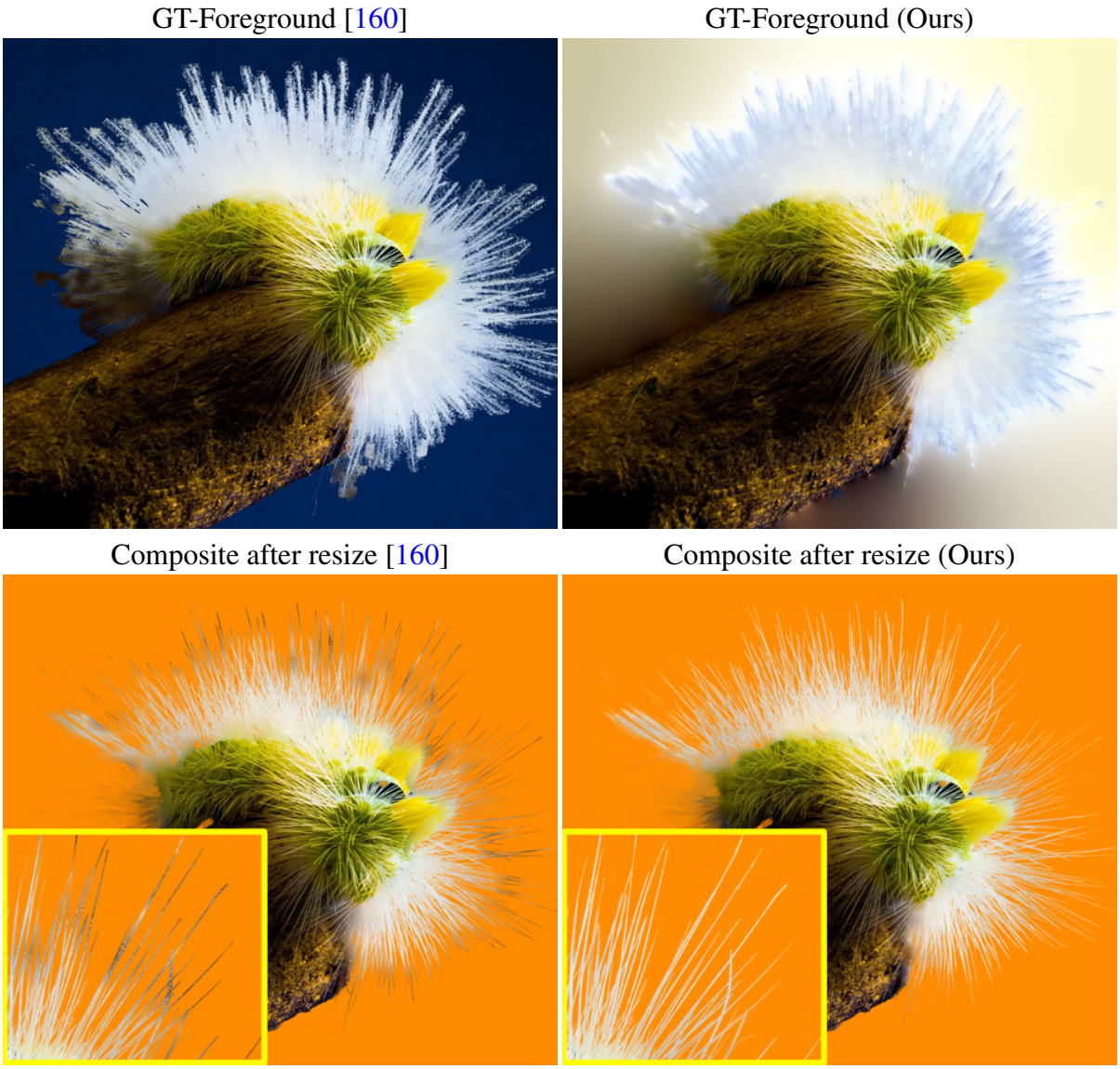


Fig. 4.4 Colour-spill from Data Augmentation. The original foreground colours provived by Xu *et al*. [160] are only defined where $\alpha > 0$. This causes issues during training when data-augmentations like resizing are used. The augmentations sample pixels from undefined regions, and the resulting foreground is contaminated. We visualise this by performing a 64x operation on the semi-transparent pixels and compositing the resulting foreground using the original alpha. Bottom left shows the result with the original foreground. Our solution is to reestimate well-defined foreground colours where $\alpha = 0$ to create a new augmentation-resistant ground truth (top left). The composite of our foreground after augmentation (bottom right) is free of colour spill. image.

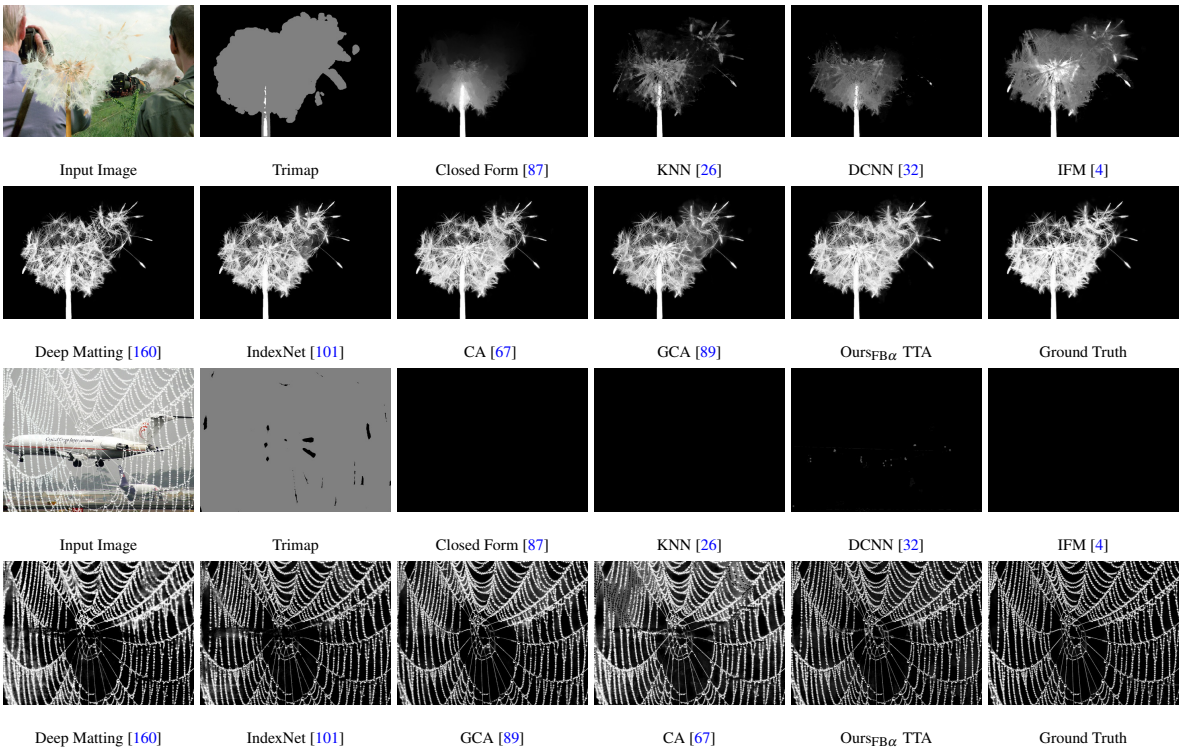


Fig. 4.5 Qualitative comparison of the alpha matte results on the Adobe Composition-1k test set [160].

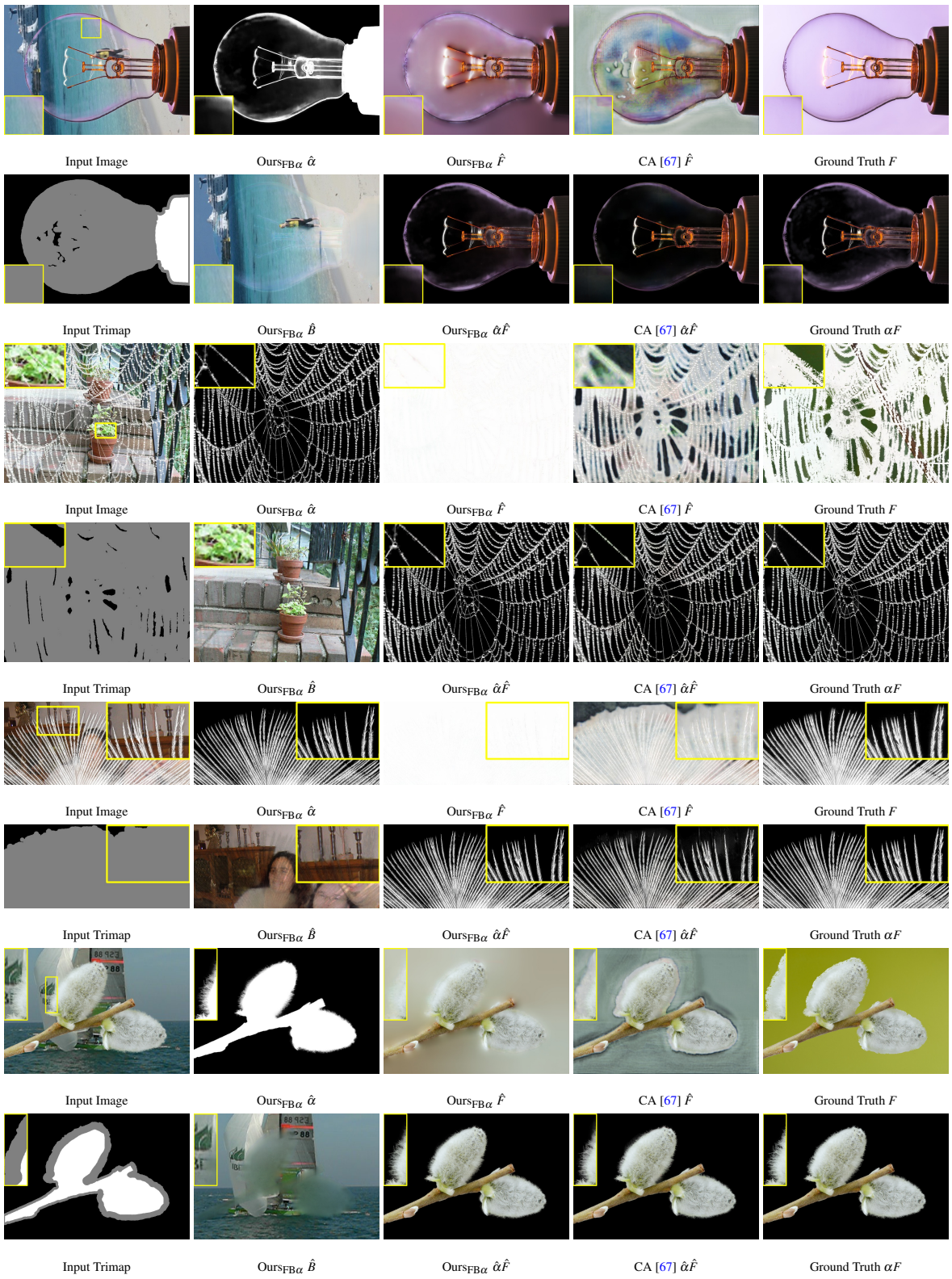


Fig. 4.6 Qualitative foreground, background and alpha matte results on the Adobe Composition-1k test set [160].

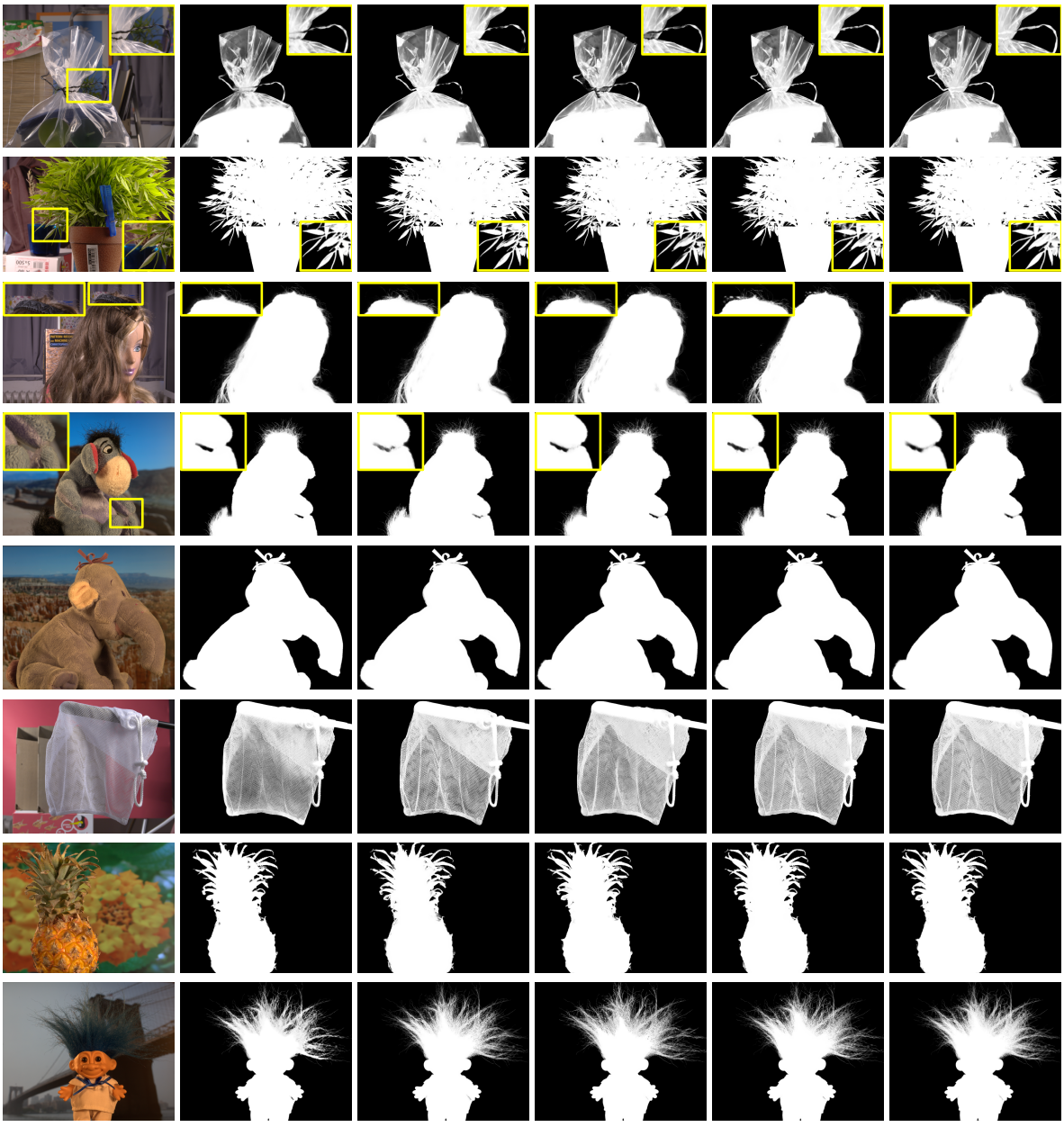


Fig. 4.7 Qualitative comparison of the alpha matte results on the alphamatting.com benchmark [129]. All eight images are included with selected examples from each trimap size.

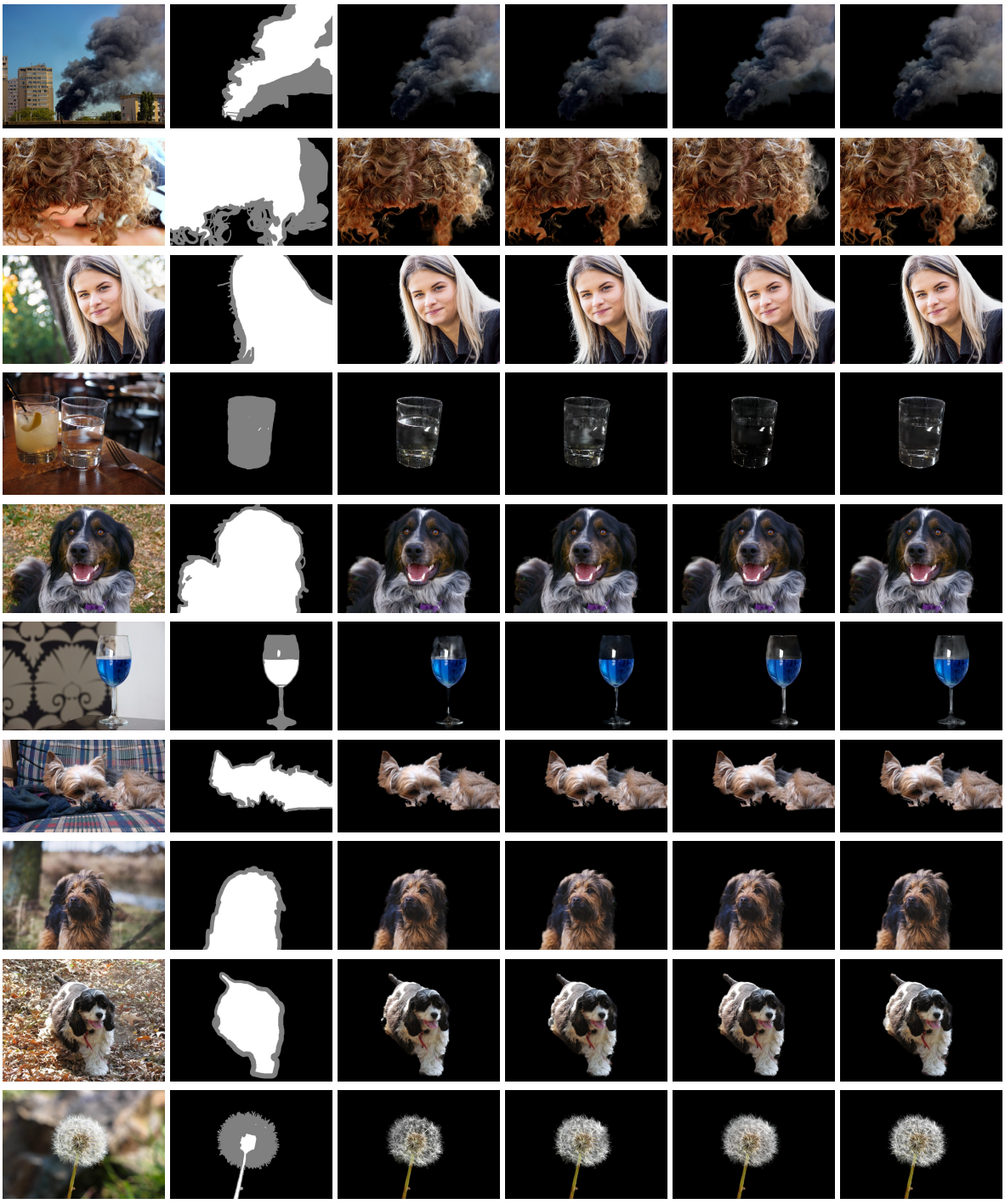


Fig. 4.8 Comparison of the results of the composites predicted from images in the real world image matting dataset [160].

Chapter 5

Deep Interactive Image Matting

In the previous two chapters, we presented methods for quick rough selections and for accurate selections at the cost of more user input. This chapter investigates a middle ground, where the matting algorithm can generate extremely accurate alpha matte selections with just a few clicks. To the best of our knowledge, our unified approach is the first attempt at offering a Deep Interactive Matting architecture and training.

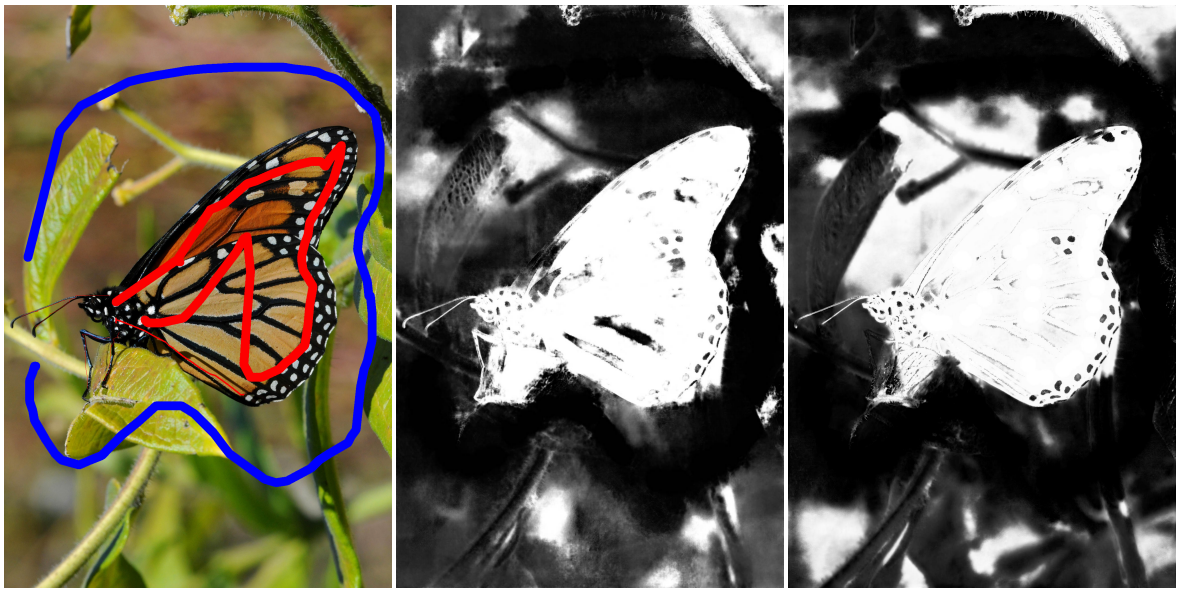
5.1 Introduction

We have seen how image matting is at the core of image editing and visual post-production. This is a critical step, yet it requires considerable effort from artists.

5.1.1 The Issue with Interactivity in Deep Matting Methods

Matting algorithms are interactive in nature and typically require the artists to define a *trimap*, that labels some regions as definite background ($\alpha = 0$), definite foreground ($\alpha = 1$) or unknown α, and then propagate that information to the entire picture to estimate the transparency at every single pixel. Matting works have long focused on the ease of use by the artists, for instance popularising the use of sparse hand-drawn scribbles to define the trimap [10, 26, 51, 84, 87, 139] or taking the time it takes for an artist to complete the entire matting process as a measure of success [58].

However, pulling alpha mattes still remains an open problem and deep convolutional neural networks (CNNs) have made notable progress in pulling more accurate mattes (*e.g.* [67, 101, 102, 144, 160]). These networks have adapted the core architectures of segmentation neural networks, such as FCN [136], to take as inputs the original RGB image and the user trimap to predict the α-matte instead of a binary mask. This progress has however come at a cost. As



Input Scribbles [160] [101]

Fig. 5.1 Example of using scribbles as an input to Deep Image Matting [160] and IndexNet Matting [101]. The hand drawn scribbles are overlaid in red=definite foreground and blue=definite background.

these networks have been optimised to operate on predefined input trimaps (see for instance the alphamatting.com benchmark [129]), the best performing networks can only respond to relatively tight trimaps, and, as acknowledged in [4], these networks are not very good at propagating mattes from very sparse inputs, such as clicks or scribbles. Figure 5.1 shows the poor quality of the predicted mattes on recent deep learning algorithms when using scribbles as input (the more complete comparison in Figure 5.7 shows that this is a systematic problem). The impracticality of using sparse trimaps means that we have lost a bit of the ease of use that characterised earlier non-deep learning approaches.

The reason deep matting algorithms are trained on relatively tight trimaps is that training directly from sparse inputs is simply too difficult. To understand this, consider that to learn how to accurately pull the transparency of individual hair strands of a person, we first need to get a good outline of the person's head. The matting loss would otherwise be overwhelmed by the larger segmentation errors. This hierarchy between the segmentation loss and the matting loss is a key problem for effectively training matting networks.

On the other hand, recent works in Interactive Segmentation [76, 92, 162] have shown that, at least for the segmentation task, networks can be adapted to successfully operate on sparse interactive user inputs and surpass prior non-deep learning methods. These methods exploit deep semantic features to get to 95% pixel accuracy (IoU) under 4 or 5 clicks. However, due

to a different application focus, these methods are unable to get much further than this 95% plateau and are still far away from being able to pull complex transparency mattes.

5.1.2 Unifying Deep Interactive Segmentation and Matting to Achieve Interactive Matting

The aim of this work is thus to bridge the gap between the prediction quality of the best matting methods and the semantic awareness and quality of user feedback of recent interactive segmentation networks to solve the combined application of *Interactive Matting*.

A simple solution to combine both approaches would be to start from an interactive segmentation network, derive a trimap by dilation/erosion of the binary mask and feed that trimap to a matting network. This is for instance the approach used in [29]. This is however not optimal as we do not know how wide the transparent region should be. The interactive segmentation step could actually be bypassed by directly proposing an interactive trimap generation tool, that would then be followed by trimap-based matting. This approach was proposed in [128] using graph-cut. This used to have a considerable computational advantage for non-deep learning techniques but, here, we would loose the benefit of the deep semantic segmentation training. We believe that we can avoid the trimap intermediate representation altogether and that a single unified approach is possible and preferable.

Our key observation comes from the fact that state-of-the-art networks in matting and interactive segmentation share very similar architectures. In particular, we recently proposed two new state-of-the-art methods for both Image Matting [46](Chapter 4) and Interactive Image Segmentation [48](Chapter 3), that are both based on an almost identical ResNet-50 encoder-decoder core architecture. Our hypothesis is therefore that, if a ResNet-50 encoder-decoder network can be trained separately for both interactive segmentation and matting tasks, it should have the capacity to resolve the combined problem of *Interactive Matting*. The main result of this work is to show how a careful training strategy can achieve this by balancing the training of both tasks. In detail, the contributions of this chapter are as follows. We propose:

1. An adaption of the matting ResNet-50 encoder-decoder architecture of [46] that can predict state-of-the-art alpha mattes, alongside with foreground and background colours, whilst also featuring the ease of use of modern interactive segmentation networks.

2. A careful training strategy that can deal with the large segmentation errors of the initial clicks, as well as the fine matting details required by matting, and also train on how to efficiently respond to user interactions.

3. A joint evaluation of the matte prediction and interactivity performance on a range of interactive segmentation, matting benchmarks, and a newly designed interactive matting benchmark.

Our results show that our unified network is relatively compact and achieves state-of-the-art performance, both in terms of alpha matte quality and interactivity, and thus offers a practical solution to the problem of quickly acquiring high quality transparency masks from images.

This chapter is organised as follows. First we detail our proposed interactive matting architecture and training regime in section 5.2. Results on matting, interactive segmentation and interactive matting benchmarks are then presented in section 5.3.

5.1.3 Interactive Segmentation

Similarly to matting networks, Deep interactive segmentation networks (*e.g.* [76, 91, 162]) have based their architecture on semantic segmentation networks (*e.g.* DeepLab [22]), or general purpose image processing architectures (*e.g.* U-Net [130]). For instance, the works of Xu et al. [162] and Majumder and Yao [105] are based on the FCN semantic segmentation network of Shelhamer et al. [136], which itself is based on the ImageNet trained VGG16 network of Simonyan and Zisserman [138]. One specific difference with matting networks is that interactive segmentation networks also need to encode the user interactions and adapt the network training to mimic these user interactions.

Interestingly the networks based on semantic segmentation operate at ¼ or ⅛ resolution, which is a too low resolution for precise object selection. Full resolution output can be obtained by using Graph-Cuts (see iFCN [162], RIS-Net [92]) or by appending dedicated upsampling networks (see FCTSFN [69] and BRS [76]), which need to be trained afterwards, in a two-stage process.

In Chapter 3 we propose a similar ResNet-50 encoder-decoder architecture to our matting network (with a shorter decoder). We also introduce a dedicated interactive branch, that includes user clicks and the previous matte results.

A key learning from our work is also that, a significant performance boost can be gained by training the network, image by image, a click at a time, and thus matching the training workflow of what a human would do. Hard coding the interactive process in training allows us to explicitly optimise for the number of clicks it takes to complete the segmentation.

5.2 Proposed Interactive Matting Method

The state of the art in image matting and interactive segmentation show us that core segmentation networks, such as the ResNet-50 U-Net style encoder-decoder, are capable of delivering best-in-class results for both matting and interactive segmentation. We propose therefore that such an architecture could be used as an end-to-end unified architecture to interactive matting and that, by adopting a specific Multi-Task training regime [132], all aspects of segmentation, matting and user interactions can be targeted at once.

5.2.1 An Interactive Matting Network

Our matting network is based on our previous work [46]. Similarly to most other existing works, we use an encoder-decoder with Unet [130] style architecture. A diagram of the architecture is presented in Figure 4.3. One notable difference to other matting works is that our network also predicts F and B directly from this single encoder-decoder. Jointly estimating for α, F and B is required to correctly re-composite the object onto a different background. The joint prediction is simply achieved by extending the output tensor channels from one to seven (1 for α, 3 for F and 3 for B).

The Encoder. The encoder is ResNet-50, and the weights are initialised by pre-training on ImageNet [39] for classification [122]. Similarly to what was proposed in AlphaGAN [102], the ResNet-50 is modified by removing the striding from 'layer 3' and 'layer 4' of ResNet-50, and instead increasing the dilation to 2 and 4 respectively. This allows processing the lower scales without downsampling.

Also, as discussed in [46], we discovered that setting the mini-batch size to 1, instead of 6-16 as it is usually done in the literature [67, 89], can greatly increase the network accuracy. This means that the ResNet-50 encoder Batch-Normalisation layers must be replaced with Group Normalisation (32 channels per group) with Weight Standardisation [122, 158].

User Interactions Embedding. Trimap, clicks or scribbles are all represented as two sparse definite foreground and background binary masks, which collate the locations where $\alpha = 0$ or $\alpha = 1$, as specified by the user. Both binary maps are transformed into distance maps, and Gaussian functions are applied at 3 different scales ($\sigma = 6.4px$, $\sigma = 25.2px$, $\sigma = 51.2px$), in a similar way to the method of [83] in interactive segmentation. This differs from other approaches in deep image matting, where the trimap is usually encoded as a single channel with value 1 for foreground, 0.5 for unknown and 0 for background. The idea here is to provide the network with an indication of the distance of each pixel to the definite foreground/background

regions. These 6 trimap channels are concatenated with the 3 RGB channels of the original image tensor and presented as input to the encoder, in an early fusion fashion.

5.2.2 A Multi-Task Interactive Matting Training Strategy

Now that we have defined the network, we need to establish a training strategy to test our initial hypothesis that this network has the capacity for interactive matting. To start the training, we have at our disposition two pre-trained networks: our interactive segmentation network from Chapter 3 and our image matting network from Chapter 4. We propose to start our training based on the pre-trained matting network as the architecture is most similar. The idea is to start from this state-of-the-art matting network, and gently introduce an interactive segmentation task alongside its matting training.

The goal of the interactive segmentation task is to outline objects in a semantic segmentation dataset, in as few as possible clicks. The difficulty will be that the low quality of the semantic dataset annotations will gradually cause a degradation of the network ability to pull fine transparent details. Therefore, we need to alternate this task with matting specific tasks, using a dedicated matting dataset, so as to reinforce the ability to pull accurate mattes. The training will then ping pong between the rough semantic segmentation task and the high precision matting tasks.

Trimap-Based Matting Pre-Training

The pre-training of our matting network is done in the same way as in Chapter 4, by providing artificial composite images, with known ground truth alpha and trimaps. The training is based on the Deep Image Matting dataset constructed by Xu *et al*. [160]. The dataset is a collection of 431 foreground and alpha channel pairs. Training samples are created by using the alpha channel and the compositing equation Eq. (1.1) to composite the foreground onto a randomly chosen background from the MSCOCO dataset [96]. Complete training details are available in Chapter 4.

Click Placement Algorithm

At the core of the interactive training regime is the automation of the user clicks. To find the click placements, we introduce here an automated procedure, which is compatible with both the segmentation and matting tasks, and which aims at mimicking, as closely as possible, the behaviour of an artist by identifying the largest prediction error in the definite foreground or background regions. The algorithm operates as follows.

For each click, we first look at the previously predicted alpha matte, $\hat{\alpha}$, and segment both the mislabelled definite foreground pixels (*i.e.* pixels for which $\hat{\alpha} < 1$ where in fact $\alpha = 1$) and definite background pixels (*i.e.* pixels for which $\hat{\alpha} > 0$ where in fact $\alpha = 0$), so as to produce two binary error masks. Computing the sum of squared errors over each of the mask connected regions allow us to select the region with the largest error. This region can correspond either to a definite foreground or definite background area. The exact click location in that region is chosen as the pixel that is the furthest away from the borders of the region (using the distance transform). For the first click, the prediction map is set to zero. This click placement algorithm is used both for training and for testing.

Task 1: Interactive Segmentation Task

The training alternates between three different tasks. Each of the 20 training epochs start with the Interactive Segmentation Task. The role of this training task is to optimise the matting network on selecting semantically coherent objects, and on real-world images. The training is done on the SBD dataset [96], which is, alongside PASCAL 2012 [43, 63], one of the most popular training sets in interactive segmentation. SBD is made of 8,548 images, with objects from 20 classes that have been manually delineated into polygonal regions. One major issue with SBD is that object boundaries are quite imprecise in comparison to what is expected for the matting task (see Figure 5.2).

The training is run over 200 pictures taken at random. A key aspect of the training is that it is done in a sequential fashion (see Chapter 3). For each picture, we place clicks in an iterative way, using our automatic click placement strategy. The loss is evaluated and the weights updated after each click placement (we stop after 4 clicks per image). For this task we reduce the loss function to the soft-IoU (Eq. 5.1) [91], instead of the full matting loss of Eq. (5.2).

$$L(\alpha) = 1 - \frac{\sum_i \alpha_i \hat{\alpha}_i}{\sum_i \max(\alpha_i, \hat{\alpha}_i)} + \sum_{c \in \text{Clicks}} (\alpha_c - \hat{\alpha}_c)^2 \tag{5.1}$$

There are three reasons for this. Firstly the ground truth labels are binary so we cannot use the compositing loss or the foreground and background losses. Secondly, the L1 and laplacian loss are more suited to pixel accurate continous labels. Thirdly when provided just a single click the IoU loss is more appropriate as the model is unlikely to get an accurate mask and we strive more for a rough approximation.

By iterating through each click, we are essentially aggregating the loss for the entire sequence of user interactions. Hence we are effectively computing the average number of clicks to reach any accuracy level. This has the advantage that our training loss matches the

interactive segmentation metric used for evaluation. Also, this means that we are optimising for the whole range of clicks.

Task 2: Interactive Matting Task

The training then moves on to the Interactive Matting Task, which mirrors the interactive segmentation, but on the 431 images of the Deep Image Matting/MSCOCO synthetic dataset instead of the binary semantic segmentation dataset. Similarly to the interactive segmentation task, the loss and weights are evaluated and updated after each new click placement. As the ground truth alpha mattes are known, the loss can be defined as the full matting loss of Eq. (5.2). The ablation study for this loss was given in Chapter 4. For each image, we randomly select a number of clicks between 5 and 8. As it is expected that the loss on the first few clicks would be still quite large, we only start updating the weights from the 4^{th} click onwards. This task is designed to best match the interactive matting application workflow.

Our matting loss is:

$$\mathscr{L} = \mathscr{L}_{\mathrm{l}}^{\alpha} + \mathscr{L}_{c}^{\alpha} + \mathscr{L}_{\mathrm{lap}}^{\alpha} + 0.25\left(\mathscr{L}_{\mathrm{l}}^{\mathrm{FB}} + \mathscr{L}_{\mathrm{lap}}^{\mathrm{FB}} + \mathscr{L}_{\mathrm{excl}}^{\mathrm{FB}} + \mathscr{L}_{c}^{\mathrm{FB}}\right) \tag{5.2}$$

Task 3: Trimap-Based Matting Task

Finally, we fine-tune the training on the original trimap-based task, as defined in the pre-training. We randomly choose 200 images and their trimaps from the Deep Image Matting dataset/MSCOCO dataset, evaluate the loss, Eq. (5.2), and update the weights, one images at a time.

There are no clicks in this training. The idea is that the two previous tasks improves the interactivity of our network, but large errors may cause the network to lose its ability to pull fine-details. Reverting to the original trimap-based training lifts the network back to its original pre-trained performance for denser trimap scenarios.

The model is trained using RAdam optimizer [100] with a learning rate 10^{-5} and weight decay 0.0005 set constantly during the training process.

5.3 Results

Recent matting works have almost exclusively focused on measuring the quality of the alpha mattes from relatively tight trimaps on two widely used benchmarks: alphamatting.com and the Composition-1k test set. In this chapter, we additionally evaluate the interactive segmentation

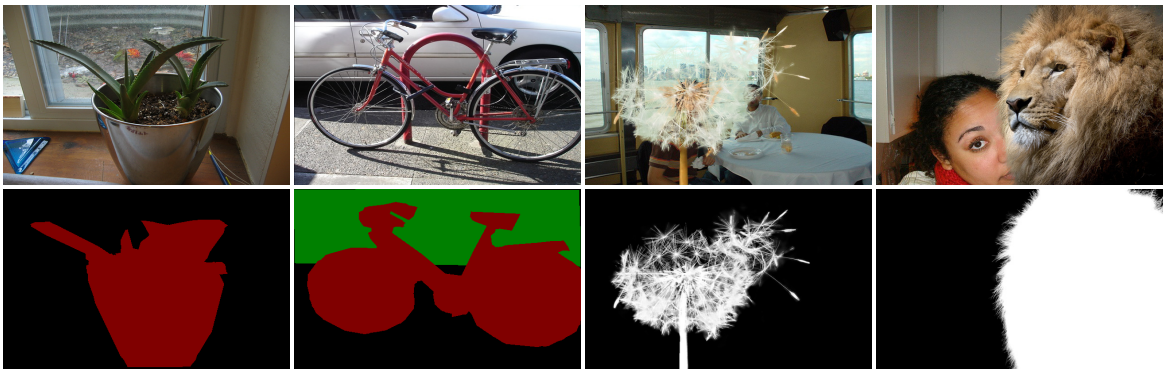


Fig. 5.2 Examples of labelling in the SBD semantic segmentation dataset [96] (left) and the synthetic composition-1k dataset for matting (right). SBD is used in training and testing interactive segmentation networks and object outlines have been hand drawn.

and matting performance of our method against others. We measure this by recording the segmentation and matting metrics over a sequence of optimal clicks.

5.3.1 Interactive Segmentation Benchmarks

Interactive Segmentation Benchmarks evaluate the Intersection over Union accuracy of the predicted masks against ground truth labels of images from the GrabCut [131], Berkeley [107] and SBD [63] test sets. The results below are obtained by binarising our predicted alpha mattes ($\alpha > 0.5$), and discarding binary regions that are not connected to a user input.

Number of Clicks to Get to 85%/90% Accuracy. On Table 5.1 we report the mean number of clicks that it takes to reach the customarily used 85%/90% IoU threshold, known as the Number of Clicks metric (NoC@85%/90%). Our algorithm reaches 90% accuracy in 3.14 clicks on GrabCut and 4.97 clicks on Berkeley. It reaches 85% in 5.75 clicks on the SBD test set. These are second best behind our method Chapter 3 and thus show that we are comparable with the state of the art in interactive segmentation.

Accuracy per Click. A more complete picture of the mean IoU accuracy at each click, across the 1-20 clicks range, is given in Figures 5.3, 5.4, 5.5. In the legend we also report the area under curve (AuC) score across the full range of clicks (1-20). The plots show that iMatting is indeed second best in that range, especially after 3 or 4 clicks.

Note that there is a natural plateau in these benchmarks. This is because the ground-truth labels are based on rough polygonal hand segmentation. It is thus probably not desirable to actually achieve 100% accuracy on these benchmarks.

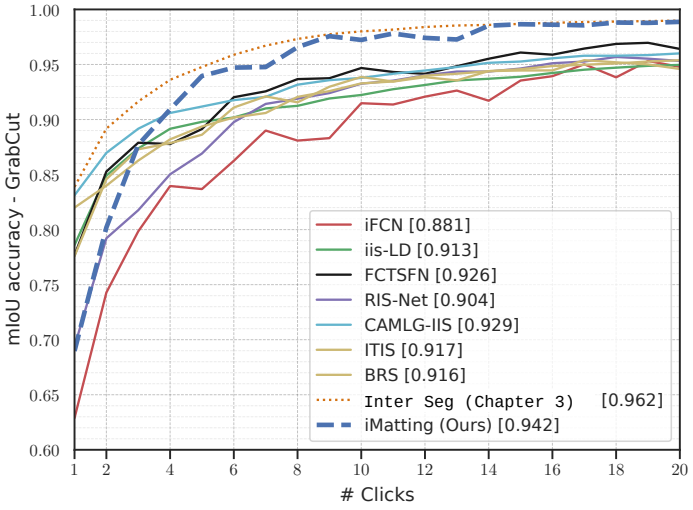


Fig. 5.3 Comparison of mean IoU scores after n clicks for the GrabCut testset [131].

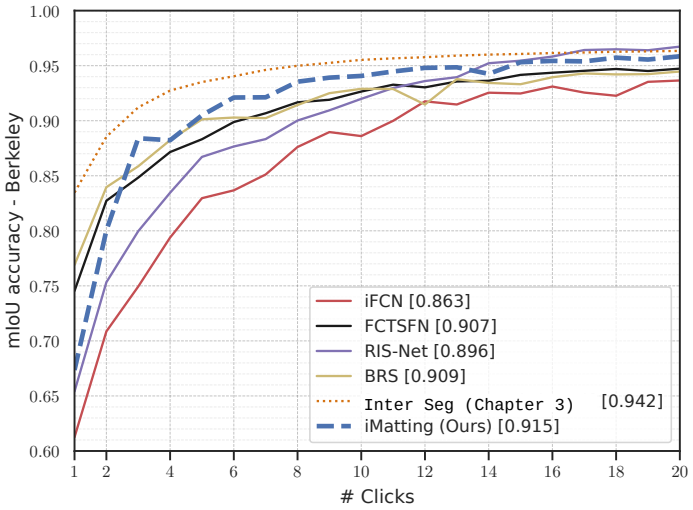


Fig. 5.4 Comparison of mean IoU scores after n clicks for the Berkeley testset [107].

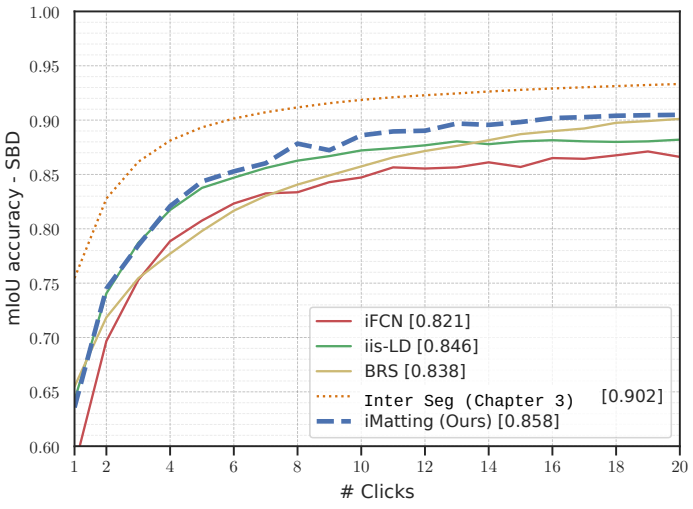


Fig. 5.5 Comparison of mean IoU scores after n clicks for the SBD testset [63].

Table 5.1 Interactive Benchmark Results: mean number of clicks required to achieve a 90% mIoU accuracy on GrabCut [131] and Berkeley [107] and 85% mIoU accuracy on SBD [63] by state-of-the-art-interactive segmentation algorithms. The best results are indicated in bold with the second best underlined.

Method	GrabCut NoC @ 90%	Berkeley NoC @ 90%	SBD NoC @ 85%
IFCN [162]	6.04	8.65	9.18
RIS-Net [92]	5.00	6.03	-
ITIS [104]	5.60	-	-
DEXTR [106]	4.00	-	-
VOS-wild [14]	3.80	-	-
FCTSFN [69]	3.76	6.49	-
IIS-LD [91]	4.79	-	7.41
CAMLG-IIS [105]	3.58	5.60	-
BRS [76]	3.60	5.08	6.59
Interactive Segmentation(Chapter 3) [48] (ours)	**2.54**	**3.53**	**3.90**
iMatting (this work)	<u>3.14</u>	<u>4.97</u>	<u>5.75</u>

5.3.2 Trimap-Based Matting Benchmarks

We then evaluate iMatting method on the more traditional Composition-1k dataset [160] for testing. It contains 1000 testing images, which have been synthesised along the same principles as the Deep Image Matting training set. Other methods have reported their results on it for comparison. The dataset provides 50 ground truth foreground images and alpha mattes, and they are composed onto 20 different backgrounds each from Pascal [44].

The alpha matte results are computed using four standard metrics [129], Sum of Absolute Differences (SAD), Mean Squared Error (MSE), Gradient Error (GRAD) and Connectivity Error (CONN). The gradient and connectivity metrics were shown at the time to be more aligned to human perception of matte quality.

The results are compiled in Table 5.2. Our results show that we are only second to base network [46].

5.3.3 Interactive Matting Benchmark

Finally, we evaluate our method on an *Interactive Composition-100* benchmark, that is made of 100 testing images from the Composition-1k testset (with 2 backgrounds per foreground image instead of 20).

The SAD, MSE, GRAD and CONN alpha metrics are computed after each click placement (according to our proposed automatic click strategy).

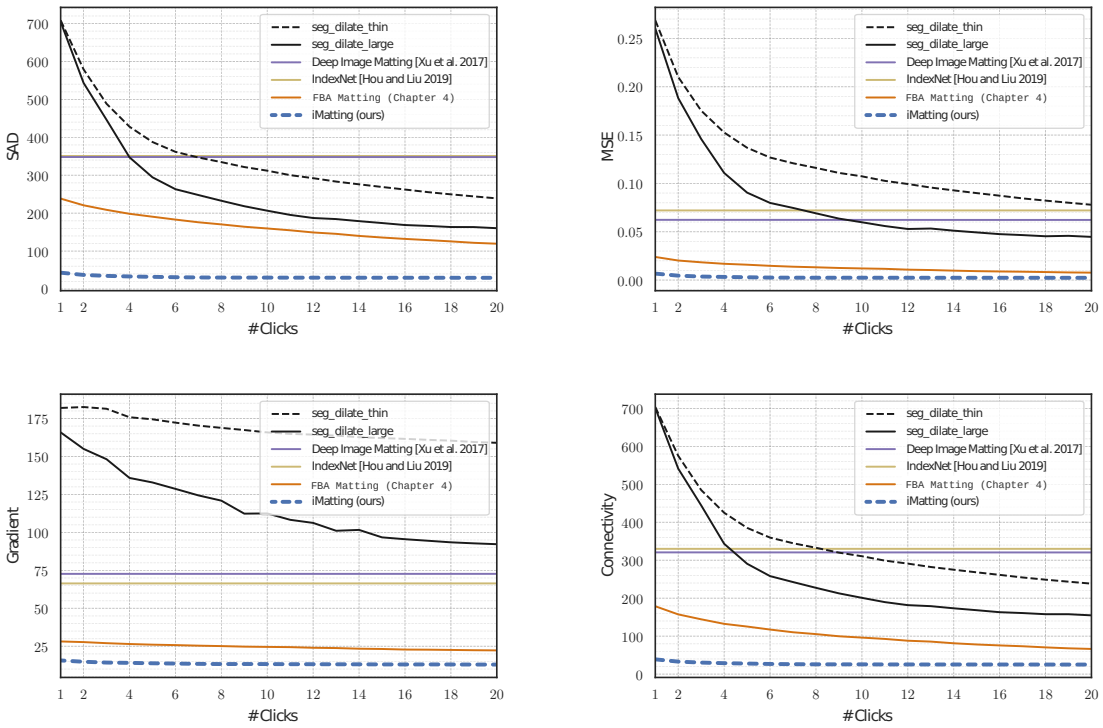


Fig. 5.6 Comparison of SAD, MSE, Gradient and Connectivity scores after n clicks for the *Interactive Composition-100* testset. Contrary to the trimap benchmark, the score is computed over the entire picture, and not only the unknown region of the trimap as there is no trimap available.

Per click results and average error over all the clicks/image are reported in Figure 5.6 and Table 5.3, respectively. We also made the evaluation for Deep Image Matting [160], IndexMatting [101], our state-of-the-art matting work [46]. We also report the ad hoc solution of chaining an interactive segmentation network Chapter 3, dilating the predicted binary boundary to create a trimap, and feeding this trimap to a trimap-based matting method [46]. The `seg_dilate_thin` method represents a 20 pixels wide unknown region and `seg_dilate_large` a 180 pixels wide unknown band.

Results clearly show that our unified approach outperforms the state of the art. As expected, Deep Image Matting [160] and IndexMatting [101] are unable to propagate mattes from such sparse inputs, and it is only after a high number of clicks that [46] starts to slowly approach the quality of iMatting.

Table 5.2 Trimap-input alpha matting results on the Composition-1k test set [160] by various state of the art matting algorithms. The test set is made of 1000 composite images. The best results are indicated in bold with the second best underlined. The results are computed over the unknown region of the trimap.

Trimap-based Method	SAD	MSE x10³	Gradient	Connectivity
Closed-Form Matting [87]	168.1	91.0	126.9	167.9
KNN-Matting [26]	175.4	103.0	124.1	176.4
DCNN Matting [32]	161.4	87.0	115.1	161.9
Information-flow Matting [4]	75.4	66.0	63.0	-
Deep Image Matting [160]	50.4	14.0	31.0	50.8
AlphaGan-Best [102]	52.4	30.0	38.0	-
IndexNet Matting [101]	45.8	13.0	25.9	43.7
VDRN Matting [143]	45.3	11.0	30.0	45.6
AdaMatting [19]	41.7	10.2	16.9	-
Learning-Based Sampling [144]	40.4	9.9	-	-
Context-Aware Matting [67]	35.8	<u>8.2</u>	17.3	33.2
GCA Matting [89]	35.3	9.1	16.9	32.5
FBA Matting (Chapter 4) [46] (ours)	**25.8**	**5.2**	**10.6**	**20.8**
iMatting (this work)	<u>33.8</u>	8.9	<u>15.3</u>	<u>31.1</u>

Table 5.3 Interactive alpha matting results on the *Interactive Composition-100* test set. The test set is made of a subset of 100 composite images from the Composition-1k test set [160]. The SAD/MSE/Gradient/Connectivity metrics are measured after each click. Reported in this table are the Area under the Curve, or average values, over the 1-20 clicks range, over the entire image. The best results are indicated in bold with the second best underlined. For a detailed comparison of the performance click, see Figure 5.6.

Interactive Method	SAD	MSE x10³	Gradient	Connectivity
	(picture average, taken over 1-20 clicks)			
Deep Image Matting [160]	348.0	62.2	72.7	320.7
IndexNet Matting [101]	350.8	72.0	66.4	330.1
[48]+small+[46]	347.0	120.9	168.0	345.2
[48]+large+[46]	262.4	81.7	115.9	257.5
[46]	<u>163.2</u>	<u>12.7</u>	<u>24.7</u>	<u>102.4</u>
iMatting (this work)	**31.5**	**2.8**	**13.5**	**27.1**

5.3.4 Qualitative Assessment

Comparison with State of the Art. Figure 5.7 illustrates the qualitative comparison of the iMatting and various matting algorithms. This comparison clearly demonstrates the superior performance of the iMatting model in response to sparse scribbles and alpha accuracy. Although the state-of-the-art models are not designed or trained for sparse trimap such as clicks/scribbles, the qualitative and quantitative results in Figure 5.7 and Table 5.2 indicate that iMatting is a generalised model capable of achieving accurate results with or without dense trimap compared to the state of the art.

Sparse and Dense Trimaps. A few samples of the alpha progression using sequences of scribbles are presented in Figure 5.10. The last column in Figure 5.10 shows the alpha predicted by iMatting given a narrow and manually generated trimap. This example clearly supports our previous statement that the proposed model can in fact estimate a visually pleasant alpha matte on natural images using a few sparse scribbles and in the absence of dense trimap.

Real World Examples. Examples of quick scribbling results are shown in Figures 5.8 and 5.9. These images come from the DIV2K dataset [3] and are in 2K resolution. Results show that just a few sparse scribbles (overlaid in red=foreground and blue=background) are enough to pull convincing mattes.

Jointly predicting α, F, allows us to directly use the network results for high quality compositing, without the need of additional post-processing to despill the foreground colour.

5.4 Limitations

In this chapter, we strove for a matting algorithm that was more practical for image editing than approaches. We have shown empirical and subjective evidence to support this claim on specific datasets, using certain metrics and simulations of user behaviour. We will now discuss some limitations of our algorithm and limitations of our evaluation.

5.4.1 The matting model

For an interactive model, latency between interaction and result is essential. The frames per second for our PyTorch model are 3.3 on an HD resolution image using an Nvidia RTX 2080TI GPU. On cheaper cards and CPU, the model is relatively slow. Furthermore, the disk size of the model weights is 138MB, which may also be prohibitive on mobile devices. There are techniques to alleviate this while maintaining model accuracy, for example, pruning and

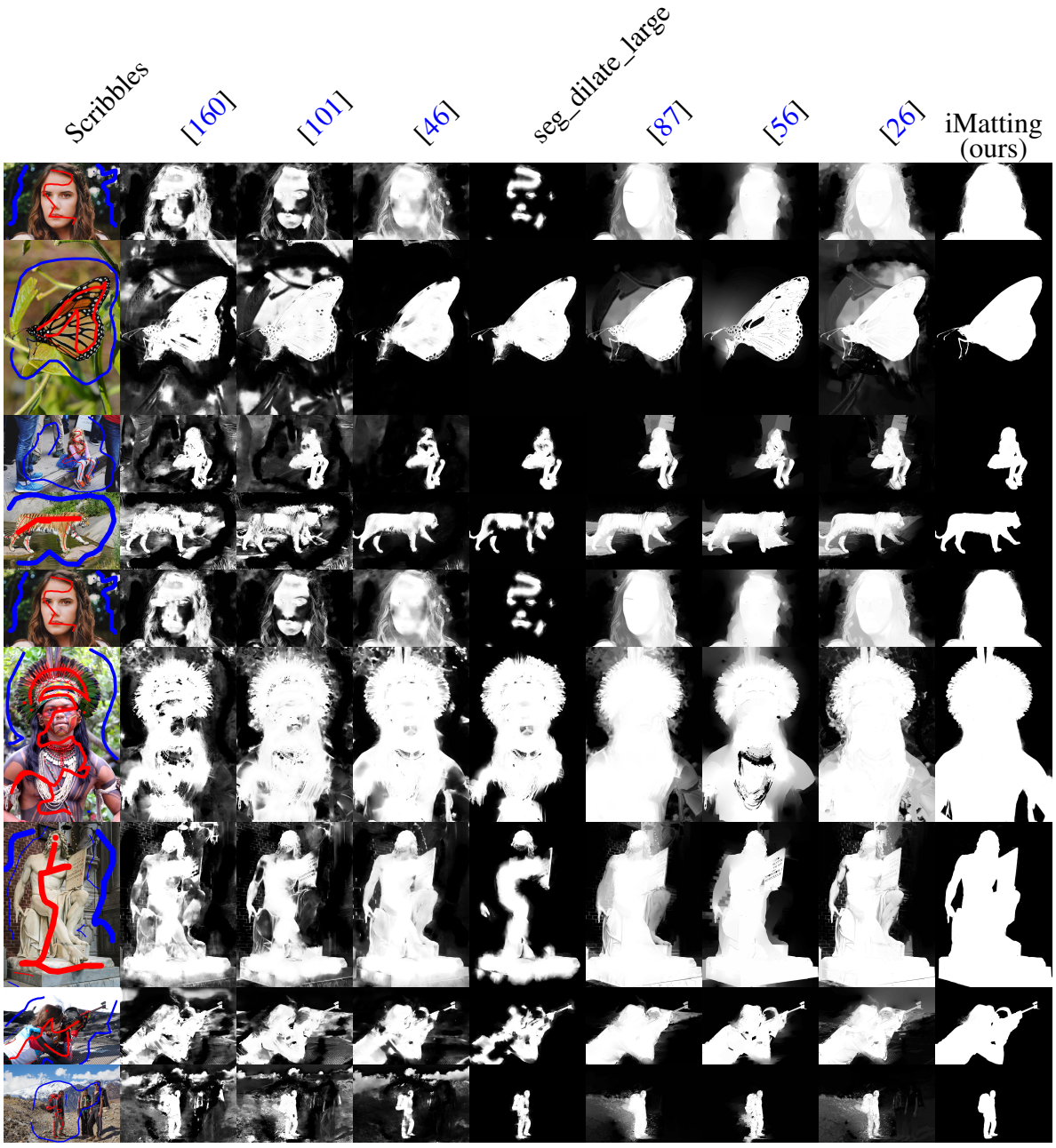


Fig. 5.7 Comparison of various matting algorithms response to sparse scribble inputs. Images are in high resolution, please zoom in for details. Deep Image Matting [160], IndexNet Matting [101], [46] are Convolutional Neural Networks, whereas Closed-Form Matting [87], Random Walk [56] and kNN-Matting [26] are not. seg_dilate_large is the sequencing of an interactive segmentation network Chapter 3 followed by the Chapter 4 matting method.

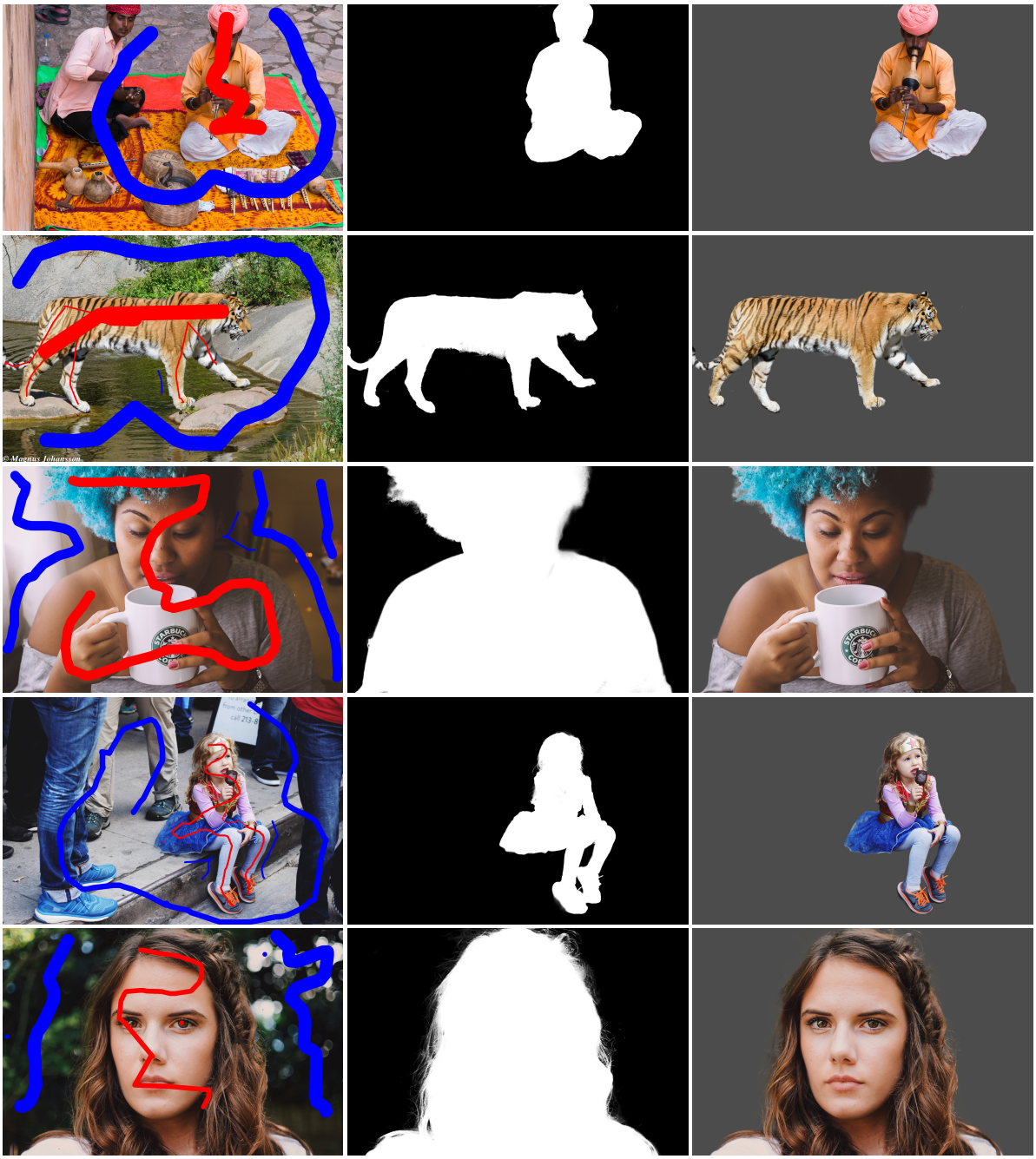


Fig. 5.8 Results of iMatting on Images from the DIV2K dataset [3]. User scribbles are overlaid in red (foreground) and blue (background). Images are in high resolution, please zoom in for details.

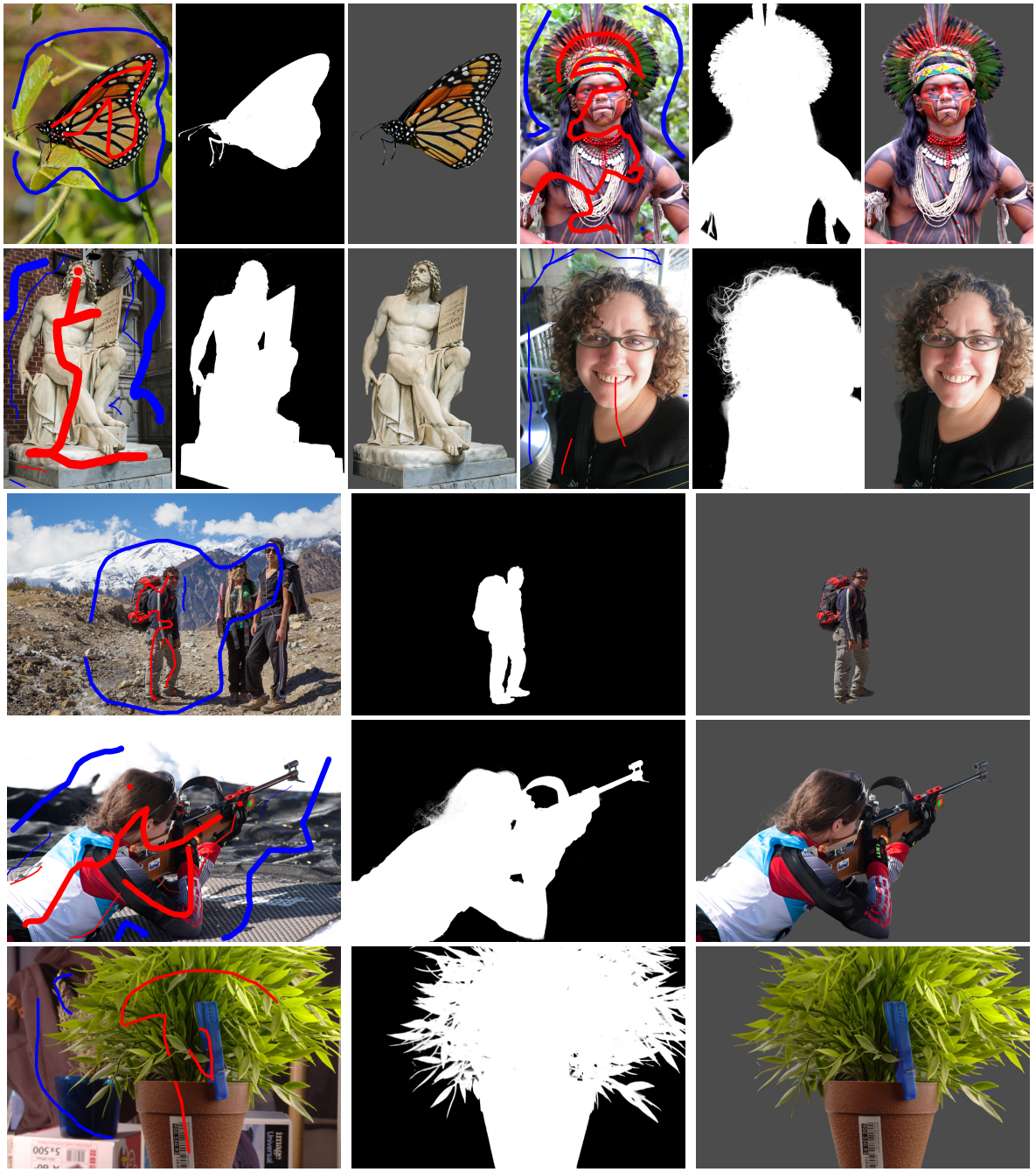


Fig. 5.9 Results of iMatting on Images from the DIV2K dataset [3], alphamatting.com test dataset [129], and Deep Image Matting real world dataset [160]. User scribbles are overlaid in red (foreground) and blue (background). Images are in high resolution, please zoom in for details.

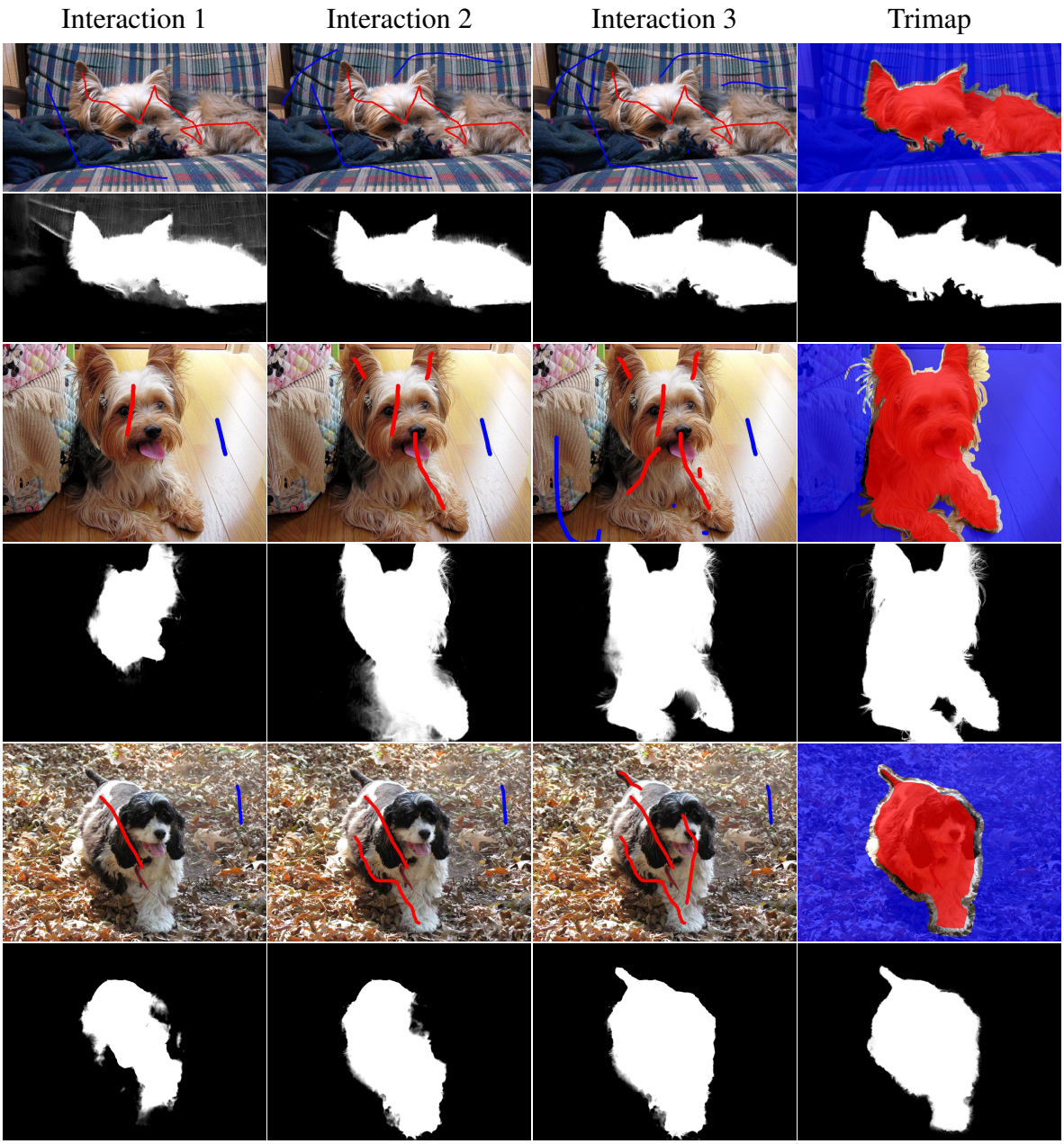


Fig. 5.10 Examples of the sequence of iterations with iMatting. The last column shows the results of iMatting when using a tight trimap. The images are from the Real Image Dataset [160].

quantization [31] or model compilation [168]. For practical application, investigating these is recommended. Our model is evaluated on images of resolution between 480x480px and 1920x1920px. Matting higher resolution images are desirable as many phones and cameras take higher resolution pictures. The current solution is to compute the matte on the downsampled image followed by bilinear upsampling. This may lead to blurry fine details and an unrealistic composite in some cases. The main limiting factor preventing the use of our algorithm in higher resolution images is inference time and memory limitations. The techniques described above could help with this, or using a second refinement model to adaptively upsample the predictions [94, 95].

5.4.2 The datasets and evaluation

Our Interactive Matting Task and Trimap-Based Matting Task are trained on synthetically generated images by compositing the foreground on random backgrounds. Our model may not perform optimally on some real-world situations in images, like shadows cast or reflections of the object onto the background, or data degradation like JPEG artifacts, blur, and noise. The interactive-segmentation task is trained on real images, which we saw in Chapter 3 Section 3.4 to improve models trained solely on synthetic data. Some contemporary and later works discuss strategies to improve generalisation to real-world images [67, 88, 94, 95]. We gave examples of our model performing well on real-world images in Figure. 5.8.

We use a click placement algorithm to evaluate our model as an interactive tool to simulate user behaviour. In future work, including a real-world user study of those using the model would give more insight. In our empirical evaluation and training, our model uses clicks. However, scribble input may be more natural on mobile devices. We show examples of our model generalising to scribble input. More empirical evaluation of accuracy for scribble input would benefit practical considerations.

5.5 Conclusion

If deep learning methods have improved the state of the art in extracting transparency masks in images, they have done so at the cost of interactivity. Recent networks require artists to provide detailed trimaps as they are not able to deal with sparse inputs such as clicks and scribbles. We have proposed that, carefully re-training a state-of-the-art matting ResNet-50 encoder/decoder architecture on three different segmentation and matting tasks could lead to a single end-to-end network that is capable of interactive matting.

Our unified network allows us to achieve comparable results to the state of the art in both interactive segmentation and matting benchmarks. The network is able to efficiently propagate sparse inputs to large regions of the pictures and produce high quality alpha mattes in a few clicks or strokes.

The network also predicts the colours of the foreground and background colours, which allows for high quality re-compositing. This and the compactness of the end-to-end unified architecture makes our proposed iMatting network a practical tool for image editing, removing the need for some of the typical steps, such as a rough interactive cutout, intermediate trimaps and foreground/background colour despill. If this all-in-one interactive matting approach is not necessarily suited to all types of scenarios, it opens up a new avenue to the the problem of natural image matting.

Chapter 6

Conclusion

This thesis introduced novel state-of-the-art deep-learning-based algorithms for interactive segmentation, trimap-based matting, and interactive matting. The result of this thesis is an interactive matting algorithm that is far more suitable for practical use in compositing than existing approaches. Furthermore, it requires less user interaction and predicts the foreground colour. This chapter summarizes our findings, especially concerning our research questions. Finally, we discuss the future of interactive matting and review the impact of this thesis.

6.1 Summary of contributions

In Chapter 5, we tackled the interactive image segmentation by maximally leveraging each user interaction. We proposed a novel neural network architecture, a novel training regime, and a novel dataset. Specifically, the network architecture was a full-resolution model with Guided Filter post-processing. We designed it to capture pixel-accurate segmentation boundaries. We also proposed the network architecture to have a separate encoding stream for the previous mask and the clicks. We trained the model using a novel click-by-click training regime. The regime and separated streams are good since they correlate the click placements to the prediction errors. We also created a novel synthetic training dataset to address the deficiencies of the standardly used training set. We proposed how to alternate between synthetic and real training data to maximise accuracy.

We performed an extensive quantitative evaluation of our method on three benchmark datasets. The results observed conclusively answer **RQ 1**. Our model outperforms existing methods for all levels of user interaction. We showed how each of our contributions improved our model. Our model does not plateau in accuracy after 20 clicks, unlike existing approaches. We design a metric to measure our model's ability to correct errors after n clicks. We find that without our contributions, the baseline model's ability to correct errors decays to zero with

increasing clicks. With our contributions, our model can correct errors just as well after ten clicks as it can after twenty.

In Chapter 4, we answered our **RQ 2.**, we presented a deep-learning method for jointly estimating the foreground, background and the alpha matte from a natural image and trimap. We proposed a modification the network from Chapter 3 for joint F, B, α estimation. Supporting this, we proposed a novel fusion mechanism that further refines the joint prediction at a negligible computational expense. For training, we found that a batch size of one is optimal, and we further investigated existing and novel loss functions. We also improve the training set to avoid mislabeling of foreground colours introduced by data augmentation. Together these changes led to state-of-the-art results for joint alpha matte and foreground estimation. We answered our research question and showed how foreground colour could be estimated with deep learning. Our changes are practical with little downside and can be adopted by future researchers and practitioners.

Chapter 5 is the final and most important chapter to our thesis. Here we answered **RQ 3.** by building upon our previous chapters to introduce the first deep-learning algorithm for interactive image matting. Key to this, we presented a novel training regime for interactive image matting and a novel evaluation scheme. The training regime alternates between interactive segmentation, trimap based matting, and interactive matting. We finetuned our network from Chapter 4 to achieve accurate results on all three tasks. We presented examples of how existing deep matting methods outright fail with scribble input. From the comparisons shown, it is clear that our method is far superior for scribbles. We presented a novel interactive matting evaluation scheme, a generalisation of the one for interactive segmentation. Using this scheme, we confirmed that our algorithm achieves state-of-the-art results. This algorithm, while not perfect, achieves our goal repeated throughout this thesis of accurate matting with sparse user input.

6.2 Closing Remarks

Image matting has been radically transformed over the past four years through the incorporation of deep learning. Excitingly, we believe that the trimap-based image matting problem is solved in practice for simple images. We can observe this through the alpha predictions on the alphamatting.com benchmark, Figure 4.7. Our method *FBA Matting* [46](and many competing methods) achieve near-perfect mattes to the eye. It is unclear how meaningful further numerical improvement on the benchmark metrics is. The possible 'retiring' of the benchmark adopted by almost every single piece of matting research leaves one wondering *"what comes next?"*.

We conclude this thesis by highlighting some ripe opportunities from both a commercial and research perspective. For example, now that image matting is so accurate it is sure to be adopted

in more image editing tools. Our research into foreground colour estimation and interactive matting is extremely practical here. Our work *Deep Interactive Image Matting* was the first of its kind. Thus there are many research possibilities in this domain, *e.g*. novel architectures, new more appropriate datasets, and metrics. Our network architecture and training also serve as a strong baseline from which to explore deep-learning based approaches to classic matting problems, such as video matting [40], image matting with additional user input [77, 142] and image matting in constrained scenarios [5, 137].
The deep learning network architectures in this thesis are convolutional neural networks. Recently Dosovitskiy *et al*. developed a transformer [147] model for image classification. Since then, many papers have adopted this vision transformer architecture for image segmentation [125, 150, 159]. Currently, these outperform all state-of-the-art CNN models. There are no interactive segmentation or matting works published using transformers, but it is a likely next step.

6.2.1 Thesis impact and contemporary works

Here we discuss the impact of the thesis, as the final writing was concluded in February 2022 while the technical work was completed before June 2020. We can look chapter by chapter at each contribution that has been referenced and expanded upon by others.

Chapter 3: Deep Interactive Training and Architecture for Object Selection

There were two publications of this work online [47, 48], which were cited seven times total. We proposed a click-by-click interactive training procedure, and this was used by Lin *et al*. [97] to train their interactive segmentation model, which extended simple clicking with a notion of click distance from the object boundary. To train our model with single image batches, we made use of Weight Standardisation [122] and Group Normalisation [158]. When Liew *et al*. [93] came to train their model for deep interactive thin object selection.

Chapter 4: Deep joint foreground, background estimation and alpha matting

We published an article of this work as an e-print on arXiv [46], it has fifteen citations. One of the primary contributions of our work was to show how joint foreground prediction was possible simply with extra channels on the output convolutional layer. Two works adopted this for video matting [94, 95] using some of our loss functions, and also our work was used as a comparison for visual and empirical evaluation. The ability of our network to produce accurate composites was a key reason in being chosen by Jampani *et al*. [75] as a component in Single Image 3D Photography. Another key contribution we made is the improvement of the training

dataset to remove colour contamination during data augmentations. An alternative solution to this problem was proposed by Wang *et al*. [153]. They empirically show the benefits of our approach and then the further benefits of their approach.

Concurrent work - Chapter 5: Deep Interactive Image Matting

The work from this chapter was not published online in any form. It was the first method of its kind at the first submission of the thesis in September 2020, since then two relevant works have tackled interactive matting and here we will discuss some of the similarities and differences.

Wei *et al*. [154] published a work titled *"Improved Image Matting via Real-time User Clicks and Uncertainty Estimation"*. In this work, they introduce user click interaction into the image matting task. There are similarities to our work in model architecture, a u-net style model. They also empirically evaluate the model using clicks on the Composition-1k dataset [161]. Unlike our model, they use a secondary refinement model that processes patches where pixels were determined by the base network to be uncertain. This allows higher resolution predictions with less memory and compute, which is an advantage over our approach. We believe however there are two clear disadvantages of their approach compared to ours. They do not train or evaluate on real image data as we do. Secondly, they do not train and evaluate using interactivity metrics. They use a fixed number of random clicks for each image, this does not model real-world user interaction.

The second work of note is from Cheng *et al*. [30] which is titled *Deep Image Matting with Flexible Guidance Input*. A core concurrently developed similarity between this work and our own is how user inputs are created during the training process. To train our interactive matting model we began from one trained using dense trimap inputs, the FBA matting model from Chapter 4. As we covered in Chapter 5 the trimap pre-training is beneficial to the final model as it is easier to learn finely detailed matte estimation when the unknown region is small. This work takes a similar approach where at the beginning epochs of training a dense trimap is used, this is then progressively shrunk over epochs until the model is training with scribble input.

Similarly to the last paper, however, this work does not do empirical evaluation using real-world data and does not use interactivity metrics to evaluate. We are thus encouraged in our belief that the ideas in this thesis are novel and can be influential in the research community.

References

[1] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282.

[2] Acuna, D., Ling, H., Kar, A., and Fidler, S. (2018). Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 859–868.

[3] Agustsson, E. and Timofte, R. (2017). Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[4] Aksoy, Y., Aydin, T. O., and Pollefeys, M. (2017a). Designing effective inter-pixel information flow for natural image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[5] Aksoy, Y., Aydın, T. O., Pollefeys, M., and Smolić, A. (2016). Interactive high-quality green-screen keying via color unmixing. *ACM Transactions on Graphics (TOG)*, 35(5):152:1–152:12.

[6] Aksoy, Y., Aydın, T. O., Smolić, A., and Pollefeys, M. (2017b). Unmixing-based soft color segmentation for image manipulation. *ACM Transactions on Graphics (TOG)*, 36(2):19:1–19:19.

[7] Aksoy, Y., Oh, T.-H., Paris, S., Pollefeys, M., and Matusik, W. (2018). Semantic soft segmentation. *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 37(4):72:1–72:13.

[8] Backes, M. H. and Oliveira, M. M. (2019). A patchmatch-based approach for matte propagation in videos. In *Computer Graphics Forum*, volume 38, pages 651–662. Wiley Online Library.

[9] Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495.

[10] Bai, X. and Sapiro, G. (2007). A geodesic framework for fast interactive image and video segmentation and matting. In *Proceedings of the International Conference on Computer Vision*.

[11] Banerjee, J. (1994). *Encyclopaedic Dictionary of Psychological Terms*. MD Publications.

[12] Barrett, W. A. and Mortensen, E. N. (1997). Interactive live-wire boundary extraction. *Medical Image Analysis*, 1(4):331 – 341.

[13] Ben-Ezra, M. (2000). Segmentation with invisible keying signal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 32–37 vol.1.

[14] Bénard, A. and Gygli, M. (2017). Interactive Video Object Segmentation in the Wild. Technical report, gifs.com.

[15] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

[16] Boykov, Y. and Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 105–112 vol.1.

[17] Bredell, G., Tanner, C., and Konukoglu, E. (2018). Iterative interaction training for segmentation editing networks. In *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 363–370.

[18] Brinkman, R. (2008). *The Art and Science of Digital Compositing: Techniques for Visual Effects, Animation and Motion Graphics (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann.

[19] Cai, S., Zhang, X., Fan, H., Huang, H., Liu, J., Liu, J., Liu, J., Wang, J., and Sun, J. (2019). Disentangled image matting. In *Proceedings of the International Conference on Computer Vision*.

[20] Caruana, R. (1997). Multitask learning. *Machine Learning*, 28(1):41–75.

[21] Chen, G., Han, K., and Wong, K.-Y. K. (2018a). Tom-net: Learning transparent object matting from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[22] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2016). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915.

[23] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017a). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848.

[24] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018b). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 801–818.

[25] Chen, Q., Ge, T., Xu, Y., Zhang, Z., Yang, X., and Gai, K. (2018c). Semantic human matting. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, page 618–626, New York, NY, USA. Association for Computing Machinery.

[26] Chen, Q., Li, D., and Tang, C. (2013). KNN matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2175–2188.

[27] Chen, Q., Xu, J., and Koltun, V. (2017b). Fast image processing with fully-convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2497–2506.

[28] Chen, X., Qi, D., and Shen, J. (2019). Boundary-aware network for fast and high-accuracy portrait segmentation. *arXiv preprint arXiv:1901.03814*.

[29] Chen, Z., Zheng, Y., Li, X., Luo, R., Jia, W., Lian, J., and Li, C. (2020). Interactive trimap generation for digital matting based on single-sample learning. *Electronics*, 9(4):659.

[30] Cheng, H., Xu, S., Jiang, X., and Wang, R. (2021). Deep image matting with flexible guidance input.

[31] Cheng, Y., Wang, D., Zhou, P., and Zhang, T. (2017). A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282*.

[32] Cho, D., Tai, Y.-W., and Kweon, I. (2016). Natural image matting using deep convolutional neural networks. In *Proceedings of the European Conference on Computer Vision*.

[33] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258.

[34] Chuang, Y.-Y., Curless, B., Salesin, D. H., and Szeliski, R. (2001). A bayesian approach to digital matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 264–271.

[35] Coleman, G. B. and Andrews, H. C. (1979). Image segmentation by clustering. *Proceedings of the IEEE*, 67(5):773–785.

[36] Criminisi, A., Cross, G., Blake, A., and Kolmogorov, V. (2006). Bilayer segmentation of live video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 53–60. IEEE.

[37] Cun, X., Pun, C.-M., and Shi, C. (2020). Towards ghost-free shadow removal via dual hierarchical aggregation network and shadow matting gan. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 10680–10687.

[38] Dai, D., Riemenschneider, H., and Gool, L. V. (2014). The synthesizability of texture examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3027–3034.

[39] Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Li, F. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[40] Erofeev, M., Gitman, Y., Vatolin, D., Fedorov, A., and Wang, J. (2015). Perceptually motivated benchmark for video matting. In *Proceedings of the British Machine Vision Conference*, pages 99–1.

[41] Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J. M., and Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338.

[42] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2011). The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html.

[43] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2012a). The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[44] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2012b). The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[45] Foamliu (2020). Deep-image-matting-pytorch. https://github.com/foamliu/Deep-Image-Matting-PyTorch.

[46] Forte, M. and Pitié, F. (2020). F, b, alpha matting. *arXiv preprint arXiv:2003.07711*.

[47] Forte, M., Price, B., Cohen, S., Xu, N., and Pitié, F. (2020). Getting to 99% accuracy in interactive segmentation. *arXiv preprint arXiv:2003.07932*.

[48] Forte, M., Price, B., Cohen, S., Xu, N., and Pitié, F. (2020). Interactive training and architecture for deep object selection. In *IEEE International Conference on Multimedia and Expo (ICME)*.

[49] Fry, R. and Fourzon, P. (1977). *The saga of special effects*. Englewood Cliffs, NJ: Prentice-Hall.

[50] Fukushima, K., Miyake, S., and Ito, T. (1983). Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE transactions on systems, man, and cybernetics*, (5):826–834.

[51] Gastal, E. S. and Oliveira, M. M. (2010). Shared sampling for real-time alpha matting. In *Computer Graphics Forum*, volume 29, pages 575–584. Wiley Online Library.

[52] Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

[53] Goel, V., Weng, J., and Poupart, P. (2018). Unsupervised video object segmentation for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 5688–5699.

[54] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[55] Grady, L. (2006). Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783.

[56] Grady, L., Schiwietz, T., Aharon, S., and Westermann, R. (2005). Random walks for interactive alpha-matting. In *Proceedings of VIIP*, volume 2005, pages 423–429.

[57] Grundhöfer, A., Kurz, D., Thiele, S., and Bimber, O. (2010). Color invariant chroma keying and color spill neutralization for dynamic scenes and cameras. *Vis. Comput.*, 26(9):1167–1176.

[58] Guan, Y., Chen, W., Liang, X., Ding, Z., and Peng, Q. (2006). Easy matting-a stroke based approach for continuous image matting. In *Computer Graphics Forum*, volume 25, pages 567–576. Wiley Online Library.

[59] Gulshan, V., Rother, C., Criminisi, A., Blake, A., and Zisserman, A. (2010). Geodesic star convexity for interactive image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3129–3136. IEEE.

[60] Guzman-Rivera, A., Batra, D., and Kohli, P. (2012). Multiple choice learning: Learning to produce multiple structured outputs. In *Advances in Neural Information Processing Systems*, pages 1799–1807.

[61] Hall, E. (1979). *Computer image processing and recognition*. Elsevier.

[62] Han, J., Yang, L., Zhang, D., Chang, X., and Liang, X. (2018). Reinforcement cutting-agent learning for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9080–9089.

[63] Hariharan, B., Arbelaez, P., Bourdev, L. D., Maji, S., and Malik, J. (2011). Semantic contours from inverse detectors. In *Proceedings of the International Conference on Computer Vision*, pages 991–998.

[64] He, K., Rhemann, C., Rother, C., Tang, X., and Sun, J. (2011). A global sampling method for alpha matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2049–2056.

[65] He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

[66] He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[67] Hou, Q. and Liu, F. (2019). Context-aware image matting for simultaneous foreground and alpha estimation. In *Proceedings of the International Conference on Computer Vision*.

[68] Hu, R., Rohrbach, M., and Darrell, T. (2016). Segmentation from natural language expressions. In *Proceedings of the European Conference on Computer Vision*, pages 108–124. Springer.

[69] Hu, Y., Soltoggio, A., Lock, R., and Carter, S. (2019). A fully convolutional two-stream fusion network for interactive image segmentation. *Neural Networks*, 109:31–42.

[70] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708.

[71] Huber, P. J. (1964). Robust estimation of a location parameter. *Ann. Math. Statist.*, 35(1):73–101.

[72] huochaitiantang (2019). Pytorch implementation of deep image matting. https://github.com/huochaitiantang/pytorch-deep-image-matting.

[73] Hwang, J., Lee, C., and Hall, E. (1979). Segmentation of solid objects using global and local edge coincidence. In *Conference on Pattern Recognition and Image Processing*, pages 114–121.

[74] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134.

[75] Jampani, V., Chang, H., Sargent, K., Kar, A., Tucker, R., Krainin, M., Kaeser, D., Freeman, W. T., Salesin, D., Curless, B., et al. (2021). Slide: Single image 3d photography with soft layering and depth-aware inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[76] Jang, W.-D. and Kim, C.-S. (2019). Interactive image segmentation via backpropagating refinement scheme. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[77] Javidnia, H. and Pitié, F. (2020). Background matting. *arXiv preprint arXiv:2002.04433*.

[78] Joker316701882 (2018). Deep-image-matting. https://github.com/Joker316701882/Deep-Image-Matting.

[79] Karacan, L., Erdem, A., and Erdem, E. (2017). Alpha matting with kl-divergence-based sparse sampling. *IEEE Transactions on Image Processing*, 26(9):4523–4536.

[80] Karacan, L., Erdem, A., and Erdem, E. (2017). Alpha matting with kl-divergence-based sparse sampling. *IEEE Transactions on Image Processing*, 26(9):4523–4536.

[81] Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

[82] Krähenbühl, P. and Koltun, V. (2011). Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, pages 109–117.

[83] Le, H., Mai, L., Price, B. L., Cohen, S., Jin, H., and Liu, F. (2018). Interactive boundary prediction for object selection. In *Proceedings of the European Conference on Computer Vision*, pages 20–36.

[84] Lee, P. and Wu, Y. (2011). Nonlocal matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2193–2200.

[85] LeGendre, C., Krissman, D., and Debevec, P. (2017). Improved chromakey of hair strands via orientation filter convolution. In *ACM SIGGRAPH 2017 Posters*, SIGGRAPH '17, New York, NY, USA. Association for Computing Machinery.

[86] Lempitsky, V., Kohli, P., Rother, C., and Sharp, T. (2009). Image segmentation with a bounding box prior. In *Proceedings of the International Conference on Computer Vision*, pages 277–284.

[87] Levin, A., Lischinski, D., and Weiss, Y. (2008). A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242.

[88] Li, J., Zhang, J., Maybank, S. J., and Tao, D. (2022). Bridging composite and real: towards end-to-end deep image matting. *International Journal of Computer Vision*, pages 1–21.

[89] Li, Y. and Lu, H. (2020). Natural image matting via guided contextual attention. *arXiv preprint arXiv:2001.04069*.

[90] Li, Y., Zhang, J., Zhao, W., Jiang, W., and Lu, H. (2020). Inductive guided filter: Real-time deep matting with weakly annotated masks on mobile devices. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

[91] Li, Z., Chen, Q., and Koltun, V. (2018). Interactive image segmentation with latent diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 577–585.

[92] Liew, J., Wei, Y., Xiong, W., Ong, S. H., and Feng, J. (2017). Regional interactive image segmentation networks. In *Proceedings of the International Conference on Computer Vision*, pages 2746–2754.

[93] Liew, J. H., Cohen, S., Price, B., Mai, L., and Feng, J. (2021). Deep interactive thin object selection. In *Proceedings of the IEEE Workshop on Applications of Computer Vision*.

[94] Lin, S., Ryabtsev, A., Sengupta, S., Curless, B., Seitz, S. M., and Kemelmacher-Shlizerman, I. (2021a). Real-time high-resolution background matting. pages 8758–8767.

[95] Lin, S., Yang, L., Saleemi, I., and Sengupta, S. (2021b). Robust high-resolution video matting with temporal guidance.

[96] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Proceedings of the European Conference on Computer Vision*, pages 740–755.

[97] Lin, Z., Zhang, Z., Chen, L.-Z., Cheng, M.-M., and Lu, S.-P. (2020). Interactive image segmentation with first click attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[98] Liu, J., Sun, J., and Shum, H. (2009). Paint selection. *ACM Transactions on Graphics (TOG)*, 28(3):69.

[99] Liu, J., Yao, Y., Hou, W., Cui, M., Xie, X., Zhang, C., and sheng Hua, X. (2020). Boosting semantic human matting with coarse annotations. *arXiv preprint arXiv:2004.04955*.

[100] Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. (2019). On the variance of the adaptive learning rate and beyond.

[101] Lu, H., Dai, Y., Shen, C., and Xu, S. (2019). Indices matter: Learning to index for deep image matting. In *Proceedings of the International Conference on Computer Vision*.

[102] Lutz, S., Amplianitis, K., and Smolic, A. (2018). Alphagan: Generative adversarial networks for natural image matting. In *Proceedings of the British Machine Vision Conference*.

[103] Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. *Proceedings of the International Conference on Machine Learning*.

[104] Mahadevan, S., Voigtlaender, P., and Leibe, B. (2018). Iteratively trained interactive segmentation. In *Proceedings of the British Machine Vision Conference*, page 212.

[105] Majumder, S. and Yao, A. (2019). Content-aware multi-level guidance for interactive instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11602–11611.

[106] Maninis, K., Caelles, S., Pont-Tuset, J., and Gool, L. V. (2018). Deep extreme cut: From extreme points to object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 616–625.

[107] McGuinness, K. and O'Connor, N. E. (2010). A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444.

[108] McGuire, M. and Matusik, W. (2006). Real-time triangulation matting using passive polarization. In *Sketches of the ACM SIGGRAPH Conference on Computer Graphics*, SIGGRAPH '06, page 88–es, New York, NY, USA. Association for Computing Machinery.

[109] Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35.

[110] Mishima, Y. (2001). Curved color separation spaces for blue screen matting. *SMPTE journal*, 110(3):131–139.

[111] Mortensen, E. N. and Barrett, W. A. (1995). Intelligent scissors for image composition. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, Proceedings of the ACM SIGGRAPH Conference on Computer Graphics, pages 191–198, New York, NY, USA. ACM.

[112] Nicholas Bloch, A. M. (2015). Nci-isbi 2013 challenge: Automated segmentation of prostate structures.

[113] Nickisch, H., Rother, C., Kohli, P., and Rhemann, C. (2010). Learning an interactive segmentation system. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, pages 274–281.

[114] Oh, S. W., Lee, J.-Y., Xu, N., and Kim, S. J. (2019). Video object segmentation using space-time memory networks. In *Proceedings of the International Conference on Computer Vision*, pages 9226–9235.

[115] Omer, I. and Werman, M. (2004). Color lines: Image specific color representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[116] Orrite, C., Varona, M. A., Estopiñán, E., and Beltrán, J. R. (2019). Portrait segmentation by deep refinement of image matting. In *Proceedings of the IEEE International Conference on Image Processing*, pages 1495–1499.

[117] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66.

[118] Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., and Sorkine-Hornung, A. (2016). A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[119] Pérez, P., Gangnet, M., and Blake, A. (2003). Poisson image editing. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pages 313–318. Association for Computing Machinery.

[120] Price, B. L., Morse, B., and Cohen, S. (2010a). Geodesic graph cut for interactive image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3161–3168. IEEE.

[121] Price, B. L., Morse, B. S., and Cohen, S. (2010b). Simultaneous foreground, background, and alpha estimation for image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[122] Qiao, S., Wang, H., Liu, C., Shen, W., and Yuille, A. (2019). Weight standardization. *arXiv preprint arXiv:1903.10520*.

[123] Qiao, Y., Liu, Y., Yang, X., Zhou, D., Xu, M., Zhang, Q., and Wei, X. (2020a). Attention-guided hierarchical structure aggregation for image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[124] Qiao, Y., Liu, Y., Yang, X., Zhou, D., Xu, M., Zhang, Q., and Wei, X. (2020b). Attention-guided hierarchical structure aggregation for image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[125] Ranftl, R., Bochkovskiy, A., and Koltun, V. (2021). Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188.

[126] Rhemann, C. (2010a). *Interactive Image Matting*. PhD thesis.

[127] Rhemann, C. (2010b). *Interactive Image Matting*. PhD thesis, TU Wien.

[128] Rhemann, C., Rother, C., Rav-Acha, A., and Sharp, T. (2008). High resolution matting via interactive trimap segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[129] Rhemann, C., Rother, C., Wang, J., Gelautz, M., Kohli, P., and Rott, P. (2009). A perceptually motivated online benchmark for image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1826–1833.

[130] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 234–241.

[131] Rother, C., Kolmogorov, V., and Blake, A. (2004). "grabcut": Interactive foreground extraction using iterated graph cuts. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, page 309–314.

[132] Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

[133] Ruzon, M. A. and Tomasi, C. (2000). Alpha estimation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 18–25.

[134] Sengupta, S., Jayaram, V., Curless, B., Seitz, S. M., and Kemelmacher-Shlizerman, I. (2020). Background matting: The world is your green screen. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[135] Shahrian, E., Rajan, D., Price, B., and Cohen, S. (2013). Improving image matting using comprehensive sampling sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 636–643.

[136] Shelhamer, E., Long, J., and Darrell, T. (2017). Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651.

[137] Shen, X., Tao, X., Gao, H., Zhou, C., and Jia, J. (2016). Deep automatic portrait matting. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Proceedings of the European Conference on Computer Vision*, pages 92–107.

[138] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

[139] Singaraju, D. and Vidal, R. (2008). Interactive image matting for multiple layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[140] Smith, A. R. and Blinn, J. F. (1996). Blue screen matting. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, volume 96, pages 259–268.

[141] Sun, J., Jia, J., Tang, C.-K., and Shum, H.-Y. (2004). Poisson matting. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, SIGGRAPH '04, page 315–321, New York, NY, USA. Association for Computing Machinery.

[142] Sun, J., Li, Y., Kang, S. B., and Shum, H.-Y. (2006). Flash matting. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pages 772–778. Association for Computing Machinery.

[143] Tang, H., Huang, Y., Jing, M., Fan, Y., and Zeng, X. (2019a). Very deep residual network for image matting. In *Proceedings of the IEEE International Conference on Image Processing*.

[144] Tang, J., Aksoy, Y., Oztireli, C., Gross, M., and Aydin, T. O. (2019b). Learning-based sampling for natural image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[145] Tang, Z., Miao, Z., Wan, Y., and Zhang, D. (2012). Video matting via opacity propagation. *The Visual Computer*, 28(1):47–61.

[146] Tsai, Y.-H., Shen, X., Lin, Z., Sunkavalli, K., Lu, X., and Yang, M.-H. (2017). Deep image harmonization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3789–3797.

[147] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[148] Vezhnevets, V. and Konouchine, V. (2005). Growcut: Interactive multi-label nd image segmentation by cellular automata. In *proc. of Graphicon*, volume 1, pages 150–156.

[149] Wang, G., Li, W., Aertsen, M., Deprest, J., Ourselin, S., and Vercauteren, T. (2019). Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *NEUROCOMPUTING*, 338:34–45.

[150] Wang, H., Zhu, Y., Adam, H., Yuille, A., and Chen, L.-C. (2021a). Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5463–5474.

[151] Wang, J. and Cohen, M. F. (2005). An iterative optimization approach for unified image segmentation and matting. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 936–943 Vol. 2.

[152] Wang, J. and Cohen, M. F. (2007). Optimized color sampling for robust matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[153] Wang, R., Xie, J., Han, J., and Qi, D. (2021b). Improving deep image matting via local smoothness assumption.

[154] Wei, T., Chen, D., Zhou, W., Liao, J., Zhao, H., Zhang, W., and Yu, N. (2021). Improved image matting via real-time user clicks and uncertainty estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[155] Wright, S. (2013). *Digital compositing for film and video*. Taylor & Francis.

[156] Wu, H., Zheng, S., Zhang, J., and Huang, K. (2018). Fast end-to-end trainable guided filter. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1838–1847.

[157] Wu, T.-P., Tang, C.-K., Brown, M. S., and Shum, H.-Y. (2007). Natural shadow matting. *ACM Transactions on Graphics (TOG)*, 26(2):8–es.

[158] Wu, Y. and He, K. (2018). Group normalization. In *Proceedings of the European Conference on Computer Vision*, pages 3–19.

[159] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. (2021). Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34.

[160] Xu, N., Price, B., Cohen, S., and Huang, T. (2017a). Deep image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[161] Xu, N., Price, B., Cohen, S., Yang, J., and Huang, T. (2017b). Deep grabcut for object selection. In *Proceedings of the British Machine Vision Conference*.

[162] Xu, N., Price, B. L., Cohen, S., Yang, J., and Huang, T. S. (2016). Deep interactive object selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 373–381.

[163] Yi, F. and Moon, I. (2012). Image segmentation: A survey of graph-cut methods. In *International Conference on Systems and Informatics*, pages 1936–1941. IEEE.

[164] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. (2019). Free-form image inpainting with gated convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4471–4480.

[165] Zhang, X., Ng, R., and Chen, Q. (2018). Single image reflection separation with perceptual losses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4786–4794.

[166] Zhang, Y., Gong, L., Fan, L., Ren, P., Huang, Q., Bao, H., and Xu, W. (2019). A late fusion cnn for digital matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[167] Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[168] Zheng, L., Jia, C., Sun, M., Wu, Z., Yu, C. H., Haj-Ali, A., Wang, Y., Yang, J., Zhuo, D., Sen, K., et al. (2020). Ansor: Generating {High-Performance} tensor programs for deep learning. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, pages 863–879.

[169] Zhu, B., Chen, Y., Wang, J., Liu, S., Zhang, B., and Tang, M. (2017). Fast deep matting for portrait animation on mobile phone. In *Proceedings of the ACM Multimedia Conference*, pages 297–305. ACM.

[170] Zongker, D. E., Werner, D. M., Curless, B., and Salesin, D. H. (1999). Environment matting and compositing. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 205–214.