MACHINE LEARNING AND DEEP LEARNING APPROACHES TO PRINT DEFECT DETECTION, FACE SET RECOGNITION, FACE ALIGNMENT, AND VISUAL ENHANCEMENT IN SPACE AND TIME

by

Xiaoyu Xiang

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

School of Electrical and Computer Engineering

West Lafayette, Indiana

August 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Jan P. Allebach, Co-Chair

School of Electrical and Computer Engineering

Dr. Qian Lin, Co-Chair

HP Personal System Software, Palo Alto, CA

Dr. Edward J. Delp

School of Electrical and Computer Engineering

Dr. Michael D. Zoltowski

School of Electrical and Computer Engineering

Approved by:

Dr. Dimitrios Peroulis

To my beloved family.

ACKNOWLEDGMENTS

Four years ago, I returned to university after two years of work experience as an engineer, which started my challenging journey of doctoral study. Along with my young and brilliant peers, I struggled to pick up and update my skills and knowledge in computer engineering. Therefore, I know that I could not have been able to make this far without the help and patience from so many incredible individuals I have known along this journey. At the last milestone in my Ph.D. journey, I would like to express my sincere gratitude to all of them.

First, I want to express my deepest appreciation to my advisor, Professor Jan P. Allebach. I first met him in Fall 2017, when I was without direction and goal. I have been extremely lucky to join his group and spending my four years with many nice students. Professor Allebach has been providing us the most precious things for Ph.D. students: supports and freedom. This makes me start to envision more possibilities for the future and the courage to grasp them. I want to let him know that he is the best mentor I have ever met and that I wish to become a great researcher like him.

I am deeply grateful to my co-chair Dr. Qian Lin, my mentor at HP since Summer 2018. She guides my path to deep learning and image enhancement, which becomes my research interest and career direction. She always provides us with rich resources and sharp feedbacks. I would also like to thank my committee members, Professor Edward J. Delp and Professor Michael D. Zoltowski, for their guidance and insightful discussions on my research. It is a great honor to have my dissertation work vetted by such prestigious scholars in the fields of signal and image processing.

I would like to express my sincere gratitude to my collaborators Yapeng Tian, Yulun Zhang, and Ruofan Zhou, who provide me countless hands-on guidance and discussion on issues of my research and life. They motivate me to be a better version of myself. I would also like to acknowledge the support and guidance I have received from my mentors and peers during my internship: Ding Liu, Yiheng Zhu, Xiao Yang, Xiaohui Shen, Jon Morton, Fitsum A. Reda, Lucas D. Young, Federico Perazzi, Amit Kumar, Rakesh Ranjan, and Andrea Colaco; the help from my labmates Yang Cheng, Jianhang Chen, Shaoyuan Xu, Tongyang Liu, Weijuan Xi, Ruiyi Mao, Fan Bu, Tianqi Guo, Qiulin Chen, Wan-Eih Huang;

and my friends Xuesi Shen, Changlin Wan, Xueyan Zou, He Huang, Qingyuan Zheng, and Srikanth Kuthuru. Thanks to everyone for supporting me through hard times and cheering for me during the pinnacle moments in my life.

Last but not least, I would like to express my deepest love and gratitude to my parents, Jian Xiang and Dazhi Zhang. They have been giving me dedication and unconditional love that shape my personality and aspiration. I have been missing them every single day when I am thousands of miles away.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

ΔE CIE L*a*b* Color Difference

ABBREVIATIONS

DPI	Dot Per Inch
RGB	Red Green Blue
PQ	Print Quality
EP	Electrophotographic
OPC	Organic Photoconductor
ITB	Intermediate Transfer Belt
PNG	Portable Network Graphs
CIE	International Commission on Illumination
ROI	Region of Interest
FN	False Negative
TP	True Positive
FP	False Positive
TN	True Negative
ROC	Receiver Operating Characteristic
CNN	Convolutional Neural Networks
VGG	Visual Geometry Group
ResNet	Residual Neural Network
IJB	IARPA-Janus Benchmark
MTCNN	Multitask Convolutional Neural Network
TAR	True Acceptance Rate
FAR	False Acceptance Rate
LFW	Labeled Face in the Wild
SGD	Stochastic Gradient Descent
CMC	Cumulative Match Characteristic
FPIR	False-Positive Identification-error Rate
AFLW	Annotated Facial Landmarks in the Wild
WFLW	Wider Facial Landmarks in-the-wild
AR	Augmented Reality

AOM	Active Orientation Model
NME	Normalized Mean Error
MSE	Mean Square Error
SR	Super-Resolution
SISR	Single Image Super-Resolution
PPI	Pixels Per Inch
CAR	Compression Artifact Removal
CARSR	Compression Artifacts Reduction and Super-Resolution
JPEG	Joint Photographic Experts Group
H.264/AVC	Advanced Video Coding
H.265/HEVC	High Efficiency Video Coding
DCT	Discrete Cosine Transform
ASPP	Atrous Spatial Pyramid Pooling
GPU	Graphics Processing Unit
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index
SOTA	State-Of-The-Art
GT	Ground Truth
QF	Quality Factor
RefSR	Reference-based Super-Resolution
AdaIN	Adaptive Instance Normalization
LPIPS	Learned Perceptual Image Patch Similarity
GMACs	Giga Multiply–Accumulate Operations
VFI	Video Frame Interpolation
VSR	video super-resolution
STVSR	Space-Time Video Super-Resolution
LSTM	Long Short-Term Memory
S2S	Sequence-to-Sequence
RNN	Recurrent Neural Network

ABSTRACT

The research includes machine Learning and Deep Learning Approaches to Print Defect Detection, Face Set Recognition and Face Alignment, and Visual-Enhancement in Space and Time. This thesis consists of six parts which are related to 6 projects:

In Chapter 1, the first project focuses on detection of local printing defects including gray spots and solid spots. We propose a coarse-to-fine method to detect local defects in a block-wise manner and aggregate the blockwise attributes to generate the feature vector of the whole test page for a further ranking task. In the detection part, we first select candidate regions by thresholding a single feature. Then more detailed features of candidate blocks are calculated and sent to a decision tree that is previously trained on our training dataset. The final result is given by the decision tree model to control the false alarm rate while maintaining the required miss rate.

Chapter 2 introduces face set recognition and Chapter 3 is about face alignment. In order to reduce the computational complexity of comparing face sets, we propose a deep neural network that can compute and aggregate the face feature vectors with different weights. As for face alignment, our goal is to solve the jittering of landmark locations when applied on video. We propose metrics and corresponding methods around this goal.

In recent years, mobile photography has become increasingly prevalent in our lives with social media due to its high portability and convenience. However, many challenges still exist in distributing high-quality mobile images and videos under the limit of data capacity, hardware storage, and network bandwidth. Therefore, we have been exploring enhancement techniques to improve the image and video qualities, considering both effectiveness and efficiency for a wide variety of applications, including WhatsApp, Portal, TikTok, even the printing industry.

Chapter 4 introduces single image super-resolution to handle real-world images with various degradations, and its influence on several downstream high-level computer vision tasks. Next, Chapter 5 studies on headshot image restoration with multiple references, which is an application of visual enhancement under more specific scenarios. Finally, as a

step towards the temporal domain enhancement, the Zooming SlowMo framework for fast and accurate space-time video super-resolution will be introduced in Chapter 6.

1. BLOCKWISE BASED DETECTION OF LOCAL DEFECTS

1.1 Introduction

Print quality is an important criterion for a printer's performance. The detection, classification, and assessment of printing defects can reflect the printer's working status and help to locate mechanical problems inside. To handle all these questions, an efficient algorithm is needed to replace the traditional visual checking method. In this project, we focus on pages with local defects, including gray spots and solid spots. We propose a coarse-to-fine method to detect local defects in a blockwise manner, and aggregate the blockwise attributes to generate the feature vector of the whole test page for a further ranking task. In the detection part, we first select candidate regions by thresholding a single feature. Then, more detailed features of candidate blocks are calculated and sent to a decision tree that is previously trained on our training dataset. The final result is given by the decision tree model to control the false alarm rate while maintaining the required miss rate. Our algorithm is proved to be effective in detecting and classifying local defects compared with previous methods.

Laser electrophotographic (EP) printers have been widely used in past decades. As one of the most important criteria in evaluating the performance of a printer, print quality is not only of concern to customers, but also designers of printers. For the reason that print quality can reflect the current working status and reveal hidden mechanical problems inside of a printer, the assessment of print quality has continued to be an important topic in printer-related research.

The traditional way of print quality diagnosis relies on the manual examination of a printed page, which is specially designed for testing purposes. The assessment work that is usually conducted by well-trained experts, includes marking exact areas with local defects and rating the overall page. Each test page can be rated as "A" "B" "C" or "D" four ranks, in which "A" and "B" mean the page passes the print quality assessment, while "C" and "D" mean the page fails the assessment. However, given the large number of pages to be evaluated, manually examining all pages is too costly and time-consuming. To solve this problem, a print defect detection algorithm is highly desired for building a smart print quality diagnosis system.

The local defect is one of the print defects of most critical concern. Typical local defects include gray spots and solid spots. The gray spot (also called carrier spot) is a phenomenon of low density around the agglomerates, which usually happens when the toner transfer from the Organic Photoconductor (OPC) to the Intermediate Transfer Belt (ITB) is blocked by some developed carriers or toner agglomerates on the OPC. Thus, a poor transfer of toners occurs around the agglomerates. An obvious visual feature of gray spots is that their color is lighter than that of the surrounding content (as shown in Figure 1.1).

The solid spot is another type of local defect. Different from gray spots, the color of solid spots is usually darker than nearby contents (as shown in Figure 1.1)). This phenomenon of high density around is due to the agglomerates of toner or carriers. The solid spot is a defect that occurs during the retransfer process, and is often observed in halftone patterns. Generally, the cause of solid spots is that the toner retransferred from the ITB to the OPC is blocked by some developed carriers or toner agglomerates on the ITB. When the after-image transferred on the ITB moves to the next Pod (T1 or T2), an air gap occurs by the carrier on non-image area or contamination. Since the retransfer quantity is lower in the air gap area, the area appears as a solid spot.

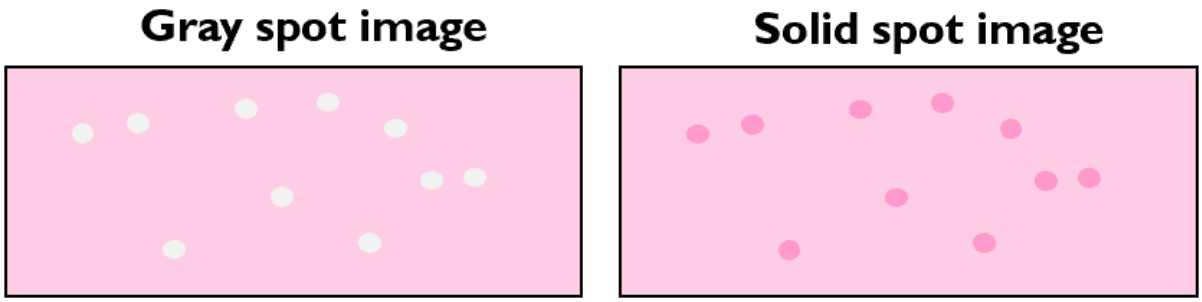


Figure 1.1. Comparison of simulated gray spots and solid spots.

There are some previous works on the automatic detection of print defects. Jing *et al.* [1] borrowed a metric from the image quality area for print defect assessment. Ju *et al.* [2], Yan *et al.* [3] and Xiao *et al.* [4] proposed new algorithms to predict the visibility of fading defects. Zhang *et al.* [5][6] modeled periodic and aperiodic bands, and applied a histogram-based method to detect them. Nguyen *et al.* [7][8][9] designed a complete framework for print

defect prediction based on defined intra-block and inter-block features for local and global characteristics, respectively. For local defects, Wang *et al.* [10] developed an algorithm to detect them and predict overall print quality with a trained support vector machine (SVM). In previous papers, we have a limited understanding of the cause, type, and severity of local defects. Different from streaks [11] or banding [12], we still lack a model describing common local defects.

In this project, we develop a blockwise algorithm to detect and characterize local defects. This method involves a coarse-to-fine strategy in defect areas detection: first select possible regions by simple thresholding, and then apply a decision tree to exclude false alarms. In addition, our algorithm can classify different local defects according to their perceptual attributes, including size, brightness, and other aspects.

1.2 Methodology

The overall workflow of our method is shown in Figure 1.2. The detection of local defects can be roughly divided into two stages: finding candidate areas, and verify the defect features inside each candidate block to give the final results. The two-stage detection pipeline can greatly reduce the runtime while ensuring the required miss rate.

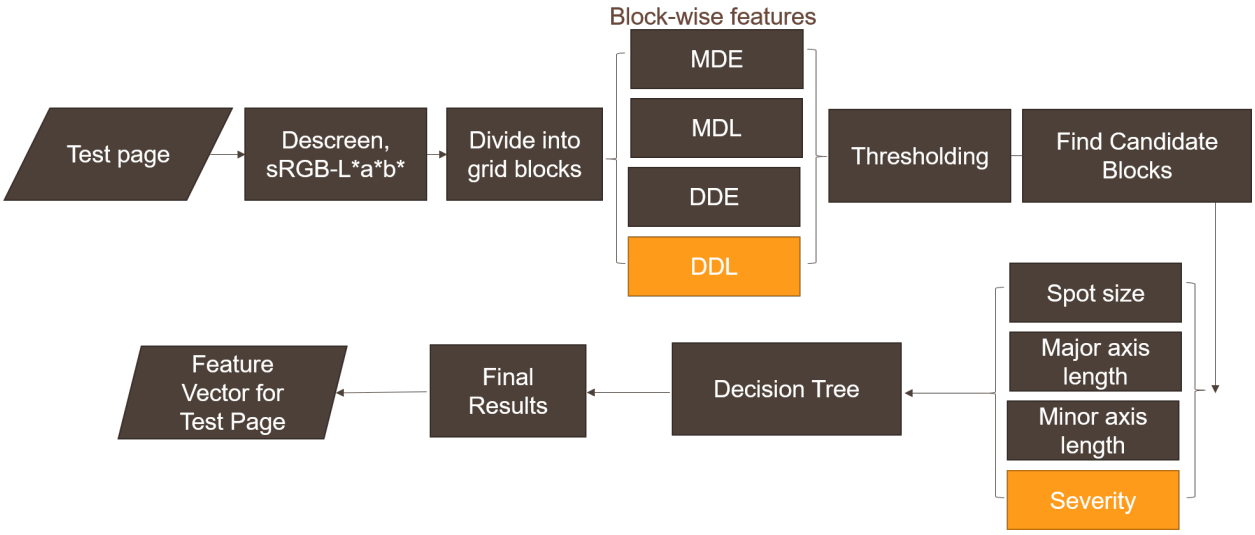


Figure 1.2. The pipeline of our method.

1.2.1 Preprocessing Test Page

Masking

The test pages are letter-size pages with at least one constant-tint area that is printed with one solid color (cyan, magenta, red, *etc.*). These test pages are scanned at 600 dpi and stored in Portable Network Graphs (PNG format) that includes an alpha channel as a mask, which is a binary channel where 0 (black) stands for the masked part while 1 (white) is for the content. The mask channel is used to tell our algorithm which part of the test page should be processed. Because besides those constant-tone areas that our defect detection algorithm focuses on, each test page also has some other contents such as fiducial marks, and a barcode that records metadata about the master file and the original printer. Also, the unprinted areas, e.g., white edge, should also be excluded from our area of interest. Figure 1.3 shows an original image, and the masked image of one test page.

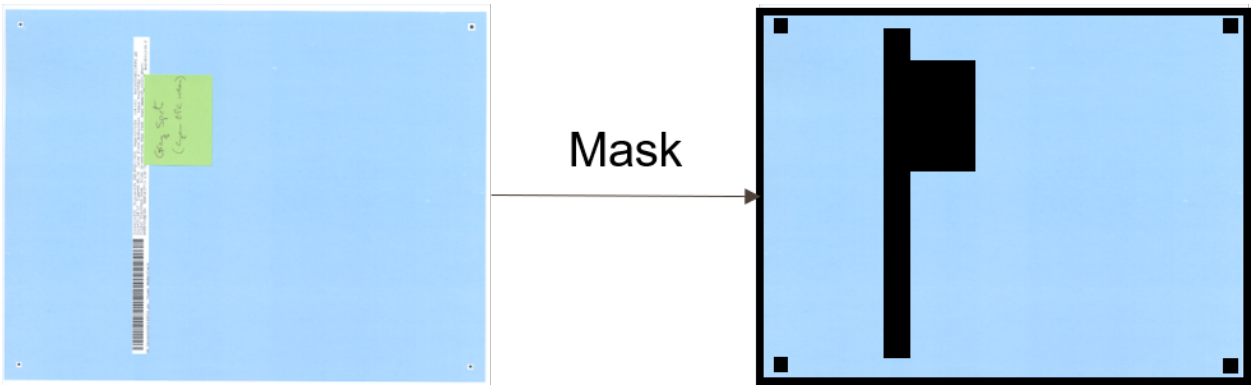


Figure 1.3. Test page sample (Original Page and masked page).

Descreening

Since all the pages are printed as halftones, high-resolution scanning would show the halftone patterns in our test pages. This might cause abnormal false alarms for local defect detection. In order to remove the halftone patterns without ruining the local defects, we apply a 12×12 Gaussian filter with a standard deviation of 2. Figure 1.4 shows a test area

before and after descreening. It is clear to see that the processed region is smoothed with no visible halftone patterns while maintaining the gray spots in which we are interested.

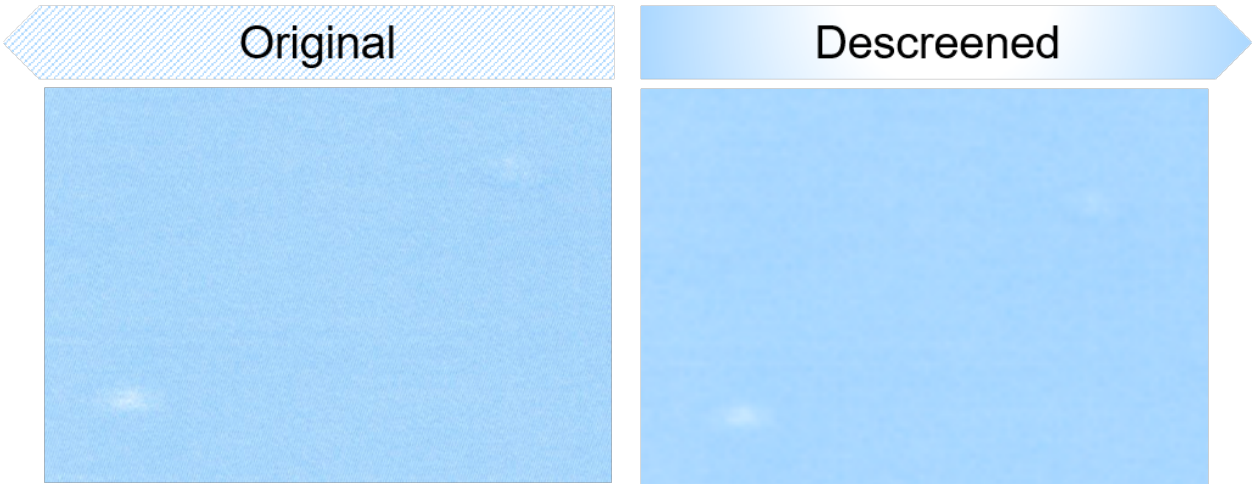


Figure 1.4. Comparison of the original page and descreened page.

Color Space Conversion

After descreening the halftoned pages, we need to transfer the pixels from the RGB color space to the CIE L*a*b* color space, where $L*$ is the lightness, and $a*$ and $b*$ are the green-red and blue-yellow color components, respectively. Compared with the RGB color space, L*a*b* is designed to be perceptually uniform according to human color vision [13]. Thus, the L*a*b* color space is widely used in color comparisons.

1.2.2 Blockwise Detection of Local Defects

Select Candidate Blocks

In order to detect local defects, we first divide the big region into relatively small blocks and detect local defects in each block. The advantage of blockwise detection is to lower memory consumption. We choose 75×75 as the block size according to the scale of common local defects (see Figure 1.5), so that a block can be big enough to contain a complete local defect, but not too big that it contains multiple defects that might interrupt the following computations. The pages are scanned at 600 dpi.

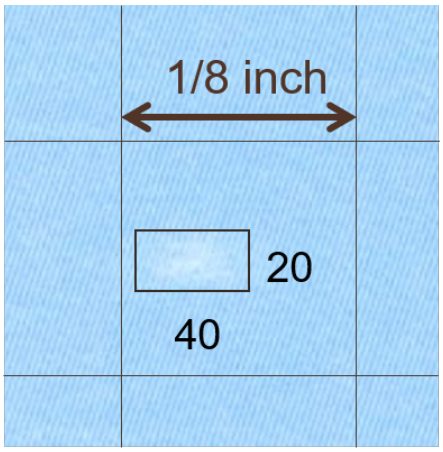


Figure 1.5. Choosing block size based on defect size.

Since the local defects are randomly located on our test page, it is very likely that a local defect falls on the boundary or even a vertex of the grid. If such a case happens, these local defects might be hard to detect. So it is necessary to search a second time for the missing defects. In the second detection, we move the grid by 35 pixels in both the x and y directions from its initial location. Figure 1.6 shows the difference between the two grids. The local defects that cannot be detected the first time would fall in the middle of a block. So we need to combine the two detection results to get all the local defects. However, there can be overlaps between the two detection results. The connected components algorithm is applied to count the local defects accurately. The combined output is our initial estimation of areas with local defects, or the Region of Interest (ROI).

For each block, we use the metrics of graininess on the local scale that were first defined by Nguyen *et al.* [9]. These metrics have proved to be effective in the following works [7][10]. In general, Nguyen *et al.*quantized the intra-block fluctuation by computing the RMS (root mean square) difference from the mean for each pixel in a block.

In the pre-processing step, the input page is converted to L*a*b* color space. So for each block, we can compute the average L*, a*, and b* according to every pixel's L*a*b* value. Then we can compute the difference between the block average and a pixel i inside of block j:

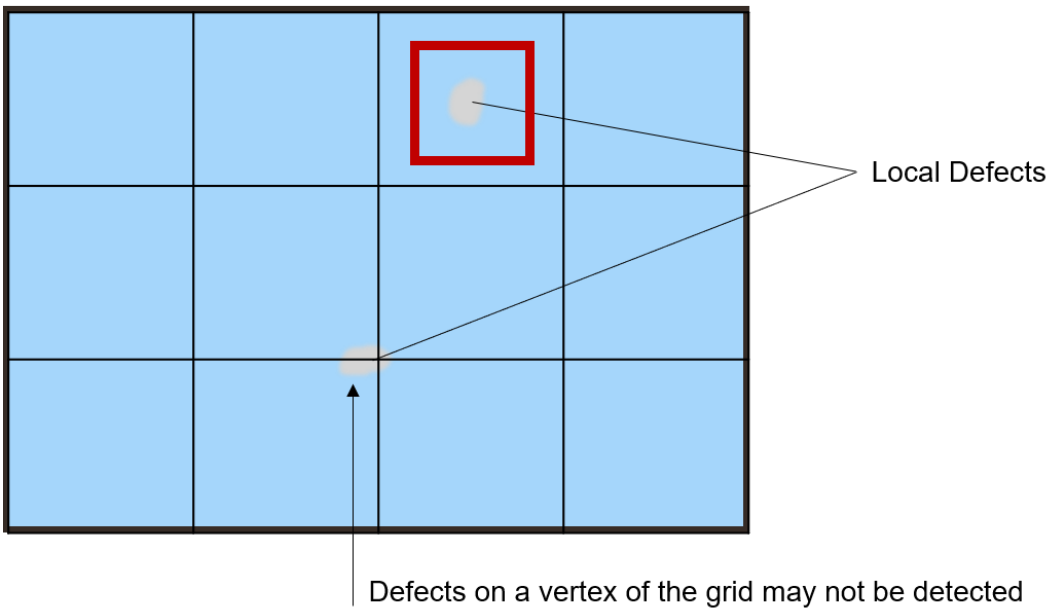
$$\Delta E_{\mathrm{ij}} = \sqrt{(L^*_{\mathrm{ij}} - L^*_{block_{\mathrm{j}}})^2 + (a^*_{\mathrm{ij}} - a^*_{block_{\mathrm{j}}})^2 + (b^*_{\mathrm{ij}} - b^*_{block_{\mathrm{j}}})^2} \tag{1.1}$$

where $L^*_{block_{\mathrm{j}}}$, $a^*_{block_{\mathrm{j}}}$, $b^*_{block_{\mathrm{j}}}$ denote average values within the block j, and L^*_{ij}, a^*_{ij}, b^*_{ij} stand for the pixel values. After finishing the calculations for all the pixels, the mean ΔE (MDE) for a block j can be computed:

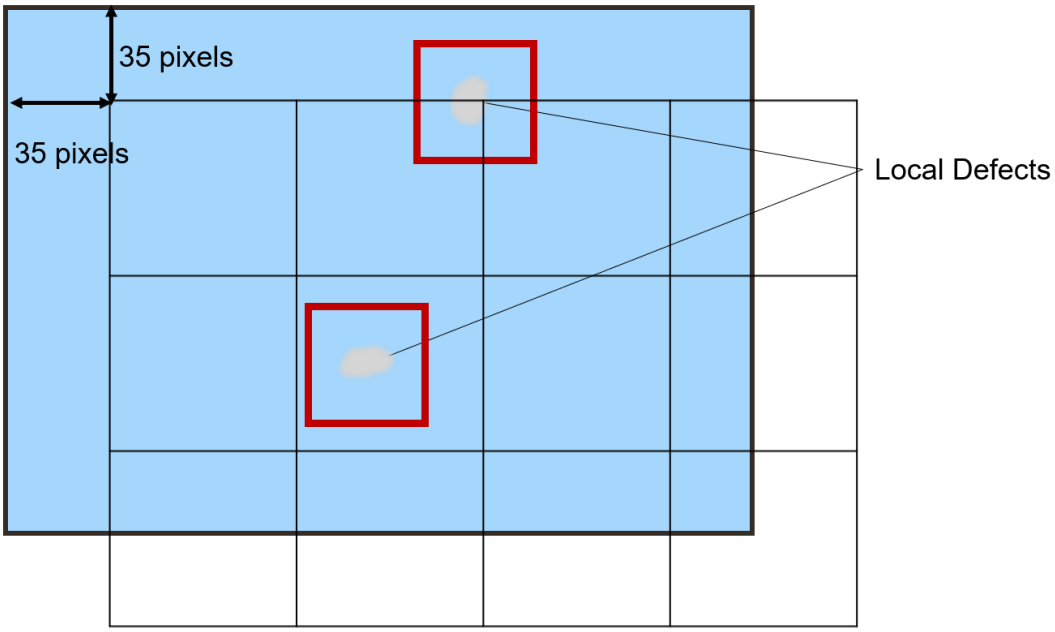
$$MDE_{\mathrm{j}} = \frac{1}{75^2} \sum_{\mathrm{i}=1}^{75^2} \Delta E_{\mathrm{ij}} \tag{1.2}$$

The standard deviation of ΔE (DDE) is given by:

$$DDE_{\mathrm{j}} = \sqrt{\frac{1}{75^2 - 1} \sum_{\mathrm{i}=1}^{75^2} (\Delta E_{\mathrm{ij}} - MDE_{\mathrm{j}})^2} \tag{1.3}$$



(a) Initial grid



(b) Move the grid by 35 pixels in both directions

Figure 1.6. Move the grid to detect defects in all possible positions. After performing the detection twice, we combine the detected blocks as the Region of Interest (ROI)

In a similar manner, we can define $L*$ related metrics ΔL, MDL, and DDL:

$$\Delta L^*_{\mathrm{ij}} = |L^*_{\mathrm{ij}} - L^*_{block_{\mathrm{j}}}| \tag{1.4}$$

$$MDL_{\mathrm{j}} = \frac{1}{75^2} \sum_{\mathrm{i}=1}^{75^2} \Delta L^*_{\mathrm{ij}} \tag{1.5}$$

$$DDL_{\mathrm{j}} = \sqrt{\frac{1}{75^2 - 1} \sum_{\mathrm{i}=1}^{75^2} (\Delta L^*_{\mathrm{ij}} - MDL_{\mathrm{j}})^2} \tag{1.6}$$

We repeat the calculations above until we go over all blocks. Higher DDE is related to a block with more fluctuations, and thus it is more likely to have local defects. We take the DDE as the metric to get the candidate ROI. As shown in Figure 1.7, first, we remove the baseline of DDE to make our algorithm less sensitive to local noise. Then we need to filter out the blocks with less fluctuation by thresholding [1]. The peaks remaining in the last graph identify the blocks that comprise the ROI.

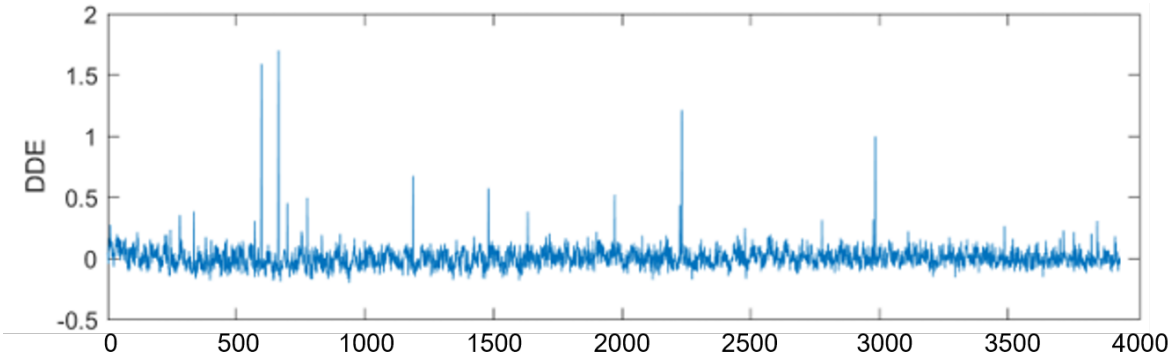
Features of Local Defects

In this section, we will introduce how to find the visible defects in the candidate blocks. Usually, local defects are small spots that are lighter or darker than the background. So we adopted the valley-emphasis algorithm to mark the distinctive pixels in a candidate block. Otsu's algorithm [14] is a popular method, where the preferred threshold t is chosen automatically by maximizing the between-class variance:

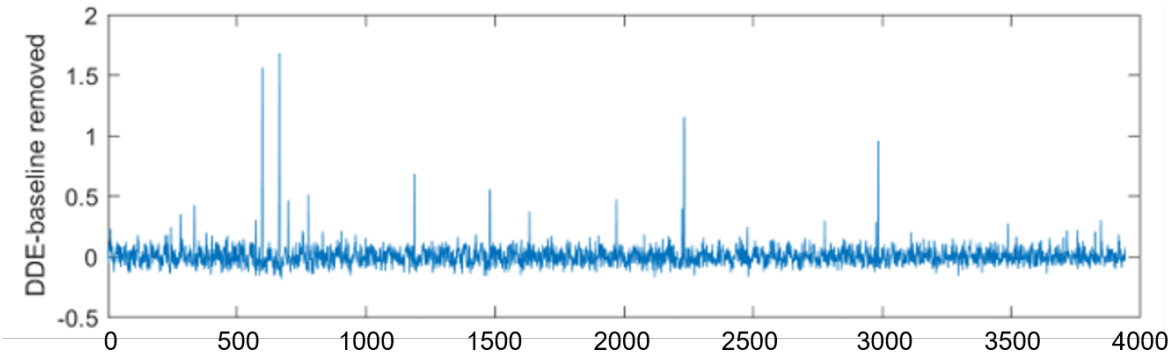
$$t^* = \arg\max_{0 \le t \le L} [\omega_1(t)\mu_1(t)^2 + \omega^2(t)\mu_2(t)^2], \tag{1.7}$$

where t denotes the input ($e.g. \Delta E$, L^*) value, L is the number of distinct gray levels, ω_1, ω_2 are percentages of pixels that belong to the background and defects, respectively, and μ_1 and μ_2 are the average values of background and defect pixels.

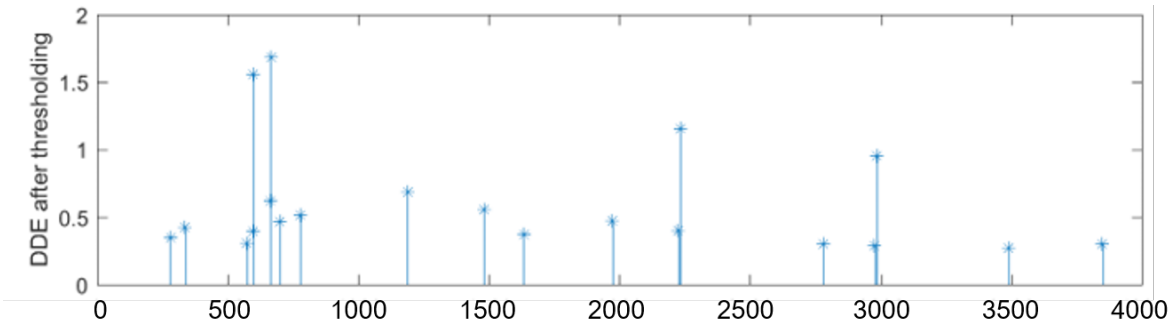
[1] ↑Generally, we set the lower threshold to be 0.3 and the upper threshold to be 2 to filter out the undesired regions coarsely. These threshold values are chosen empirically. Here, the image DDE values are on a scale from 0 to 5



(a) Plot DDE of each block



(b) Remove baseline



(c) Select candidate blocks according to DDE

Figure 1.7. Select candidate blocks according to their DDE.

Ng *et al.* [15] proposed a new form of valley-emphasis algorithm based on Otsu's method by adding a new term in the maximization to emphasize the "valley" in the histogram. It is based on the assumption that the correct threshold should be located at the "valley" of histogram; so the foreground and background can be separated properly. Ng *et al.*'s modification can be expressed as:

$$t^* = \arg\max_{0 \le t \le L} [(1 - p(t))(\omega_1(t)\mu_1(t)^2 + \omega^2(t)\mu_2(t)^2)] \tag{1.8}$$

The additional term $p(t)$ is the percentage of pixels at level t. It serves as a weighting term, so that the lower the percentage is, the higher the weight is.

In Figure 1.8, we compare the results of applying the valley-emphasis algorithm followed by thresholding, to the descreened image expressed in either ΔE or L^* units. Figure 1.8 shows the results of different inputs after thresholding. The left-most image is the descreened input block, which includes a gray spot on the upper-left corner and some dispersed dark agglomerates. The result of L^* seems to be more focused on the gray spot region, while the ΔE result marks out both the gray spot and the dark agglomerates. For comparison, we also show the result that is directly acquired from the FWHM (Full width at half maximum) of ΔL, which is more close to the ΔE results. The reason for this difference lies in operation to get the "Δ values", which is computing the absolute difference from average. In this manner, both dark and light regions strongly deviate from average, so that they are both marked out by the valley-emphasis algorithm. According to more comparisons on our test pages, the L^* inputs tend to give more gray spot results while fewer dark spots. Depending on our actual needs, we can choose either input to maximize accuracy.

After we get the mask of defects within a block, we can conduct the analysis for its attributes:

- Size (pixels): number of pixels in the defect area selected by valley-emphasis algorithm;
- Light / Dark: if the defect area is lighter or darker than the block average. This attribute can help us identify the type of local defect;

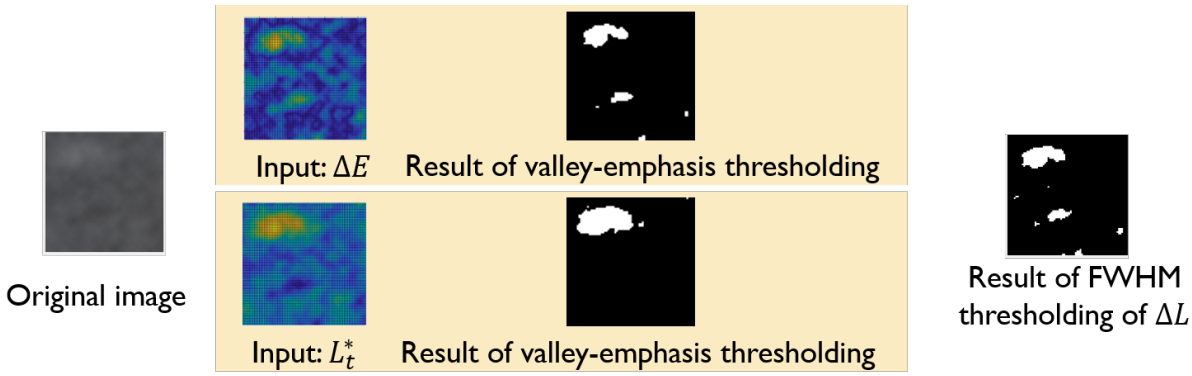


Figure 1.8. Results of Valley-emphasis Algorithm.

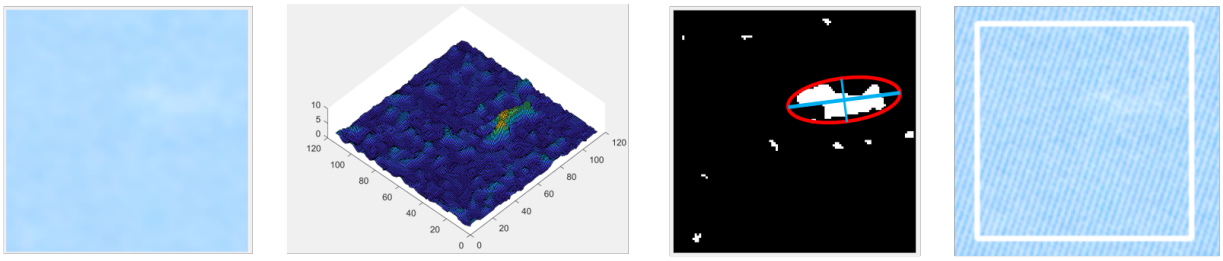
- Major and minor axis lengths: major and minor axis lengths of the equivalent eclipse of the defect area;

- Severity: the contrast of the defect area versus the background, defined by:

$$\frac{\sum_{defect} \Delta E}{\sum_{background} \Delta E},$$

  which can also be regarded as the ratio of "defect volume" to the "background volume".

By inspecting each ROI according to the process shown in Figure 1.9, we can get the attributes above. Along with DDE, MDL, DDL, these features will be used in the following steps to exclude false positives (invisible defects). To better visualize the detected visible defects, we draw a white bounding box around the block with gray spots, and a black box around the block with dark spots. We use the Connected Components algorithm to combine bounding boxes of adjacent blocks with the same type of defect into one bigger box.



(a) Candidate area (b) Valley-emphasis method (c) Defect mask (d) Bounding box

Figure 1.9. The processing of a candidate area in each step.

If we mark blocks with visible local defects as 1 and the rest as 0, we can plot their distribution versus each feature as shown in Figure 1.10. According to these plots, we can tell that MDL, DDL, and DDE are all good metrics for choosing candidate blocks in the initial step. The severity can exclude abnormal cases (*e.g.*infinite values). With defect size, we can exclude cases that are too small to see. Similarly, the major and minor axis lengths can help us exclude thin lines that are imperceptible and non-local. Although all these features obviously correlate with the existence of visible local defects, there is no single threshold that separates blocks with/without defects.

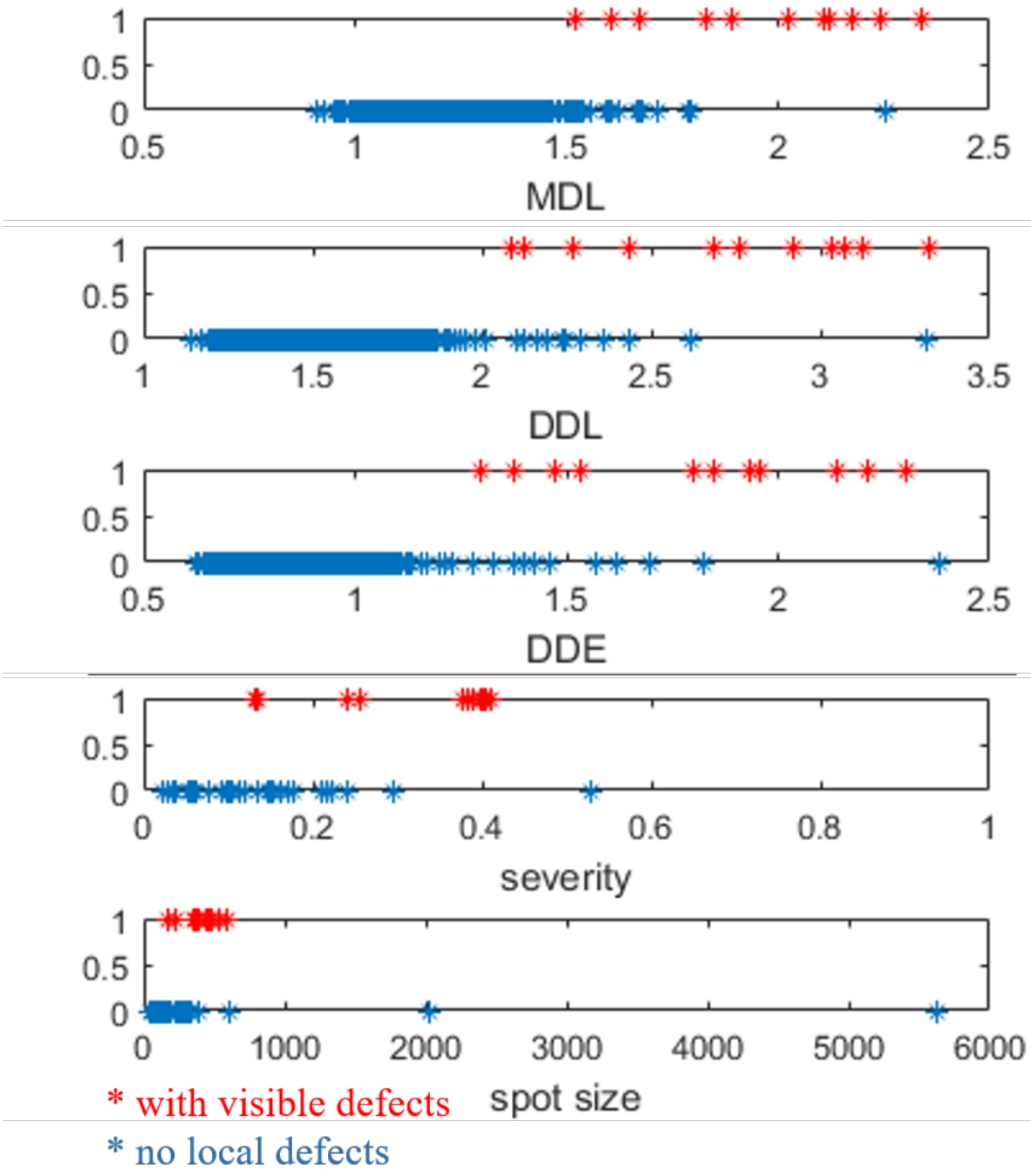


Figure 1.10. Distribution of each feature. The blocks with defects are marked as 1, and the other blocks as 0, so different blocks appear on the top and bottom of the graph, respectively.

Blockwise Dataset

We create a blockwise dataset that includes several types of local defects, including gray spots, pinholes, *etc.*. for further refinement models. This dataset is from 15 test pages with 66 uniform color regions including 12 colors. These test pages are in A4 size, scanned at 600 dpi. Taking out margins and barcode areas, 67,465 blocks are sampled from all test pages, among which 1,502 blocks are marked as blocks with local defects. After initial computation, 5,043 blocks are selected as ROIs by our algorithm.

Each block sample contains the following three types of metadata: 1) global features, including filename, block index, and color, which are related to the whole page; 2) blockwise features, including the block's x and y coordinate ranges, average L^*, a^*, b^* values, DDE, MDL, DDL, and the ground truth of local defects. These features can help us locate the block among the raw pages and are only related to the block itself; 3) local defect features: light/dark, defect size, equivalent eclipse's major and minor axis length, and severity. Only the 5,043 blocks that passed initial selection have the third type of feature.

Based on our blockwise dataset, we trained a decision tree model with 7 blockwise features as decision nodes: MDL, DDL, DDE, defect size, major axis length, minor axis length, and severity. We calculate the information gain (IG) of these features a to decide which feature to use and the its proper split:

$$IG(T, a) = H(T) - H(T|a), \tag{1.9}$$

where T denotes our training set. The entropy is:

$$H(T) = \sum_{\mathrm{i} \in labels} -p_{\mathrm{i}} \log_2 p_{\mathrm{i}}. \tag{1.10}$$

For a value v taken by feature a, let $S_a(v) = \{x \in T | x_a = v\}$ to be the set of training data of T for which feature a is equal to v, the conditional entropy $H(T|a)$ is:

$$H(T|a) = \sum_{v \in vals(a)} \frac{|S_a(v)|}{T} \cdot H(S_a(v)) \tag{1.11}$$

Since in our training set, the number of blocks with defects is smaller compared to normal blocks, we train the decision tree model with a 2×2 cost matrix, where element $C(\mathrm{i}, \mathrm{j})$ of this matrix is the cost of classifying an observation into class j if the true class is i. By changing the miss-classification cost, we can get models with different miss rates and false alarms that are calculated by:

$$\text{Miss Rate} = \frac{FN}{TP + FN} \tag{1.12}$$

$$\text{False Alarm} = \frac{FP}{FP + TN} \tag{1.13}$$

where FN denotes "False Negative", TP is "True Positive", FP is "False Positive", and TN is "True Negative". The performance of our model is shown in Figure 1.11. Considering the balance of the false alarm and the miss rate, the best result of our model is obtained when $cost = 2$, the False Alarm rate is 0.088, while the Miss rate is 0.266.

1.3 Results

Taking the final output blocks of the decision tree, we can get the final detection output as shown in Figure 1.12. By aggregating their information, we can generate a feature vector for the whole test page. There are several features that we care about: number of all types of defects on the test page, number of each type of defect, local defects' average size, and maximal and minimal size, and standard deviation of the size, average severity, maximal and minimal severity. In addition, we also want to know the average location of all local defects to determine whether or not their distribution is random. The output feature vector for a test page is listed in Table 1.1.

Our algorithm can not only be applied on print quality assessment, but also on detecting the scratches and contamination in the manufacturing of glass touchpads. Figure 1.13 shows that our method is robust to background noise that is generated from the matte surface and uneven lighting.

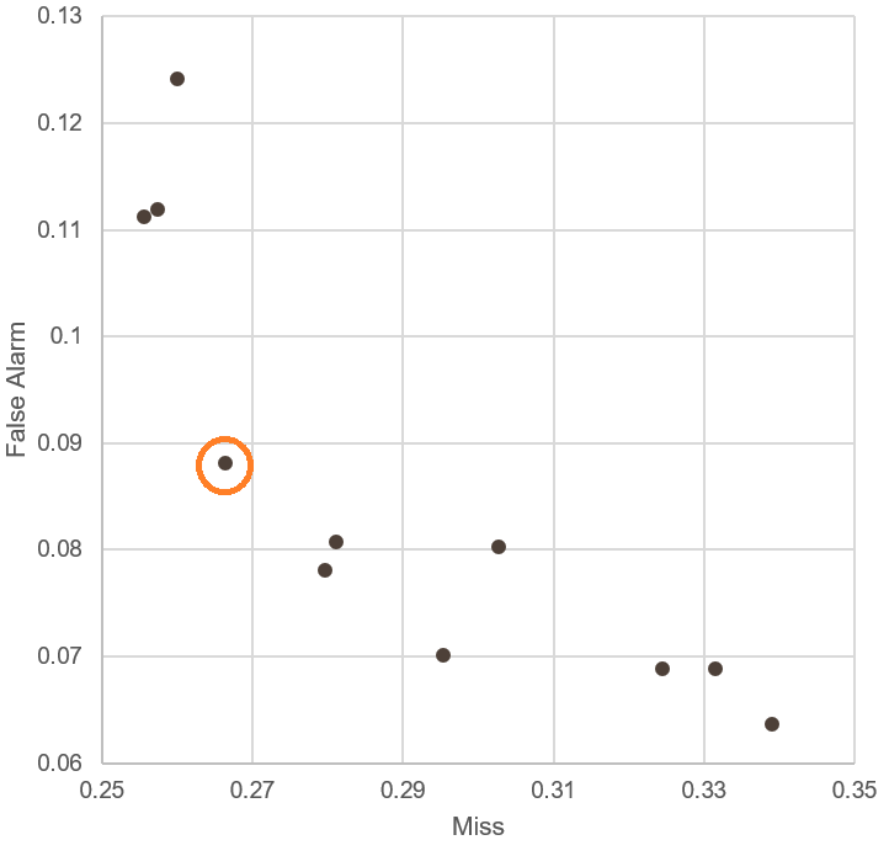


Figure 1.11. ROC (Receiver operating characteristic) plot. The best operating point is circled.

Table 1.1. Table 1: The feature vector for a test page

Number of defects on this page	19
Number of Gray Spots	15
Number of Solid Spots	4
Average size (mm^2)	0.53
Max size (mm^2)	1.24
Min size (mm^2)	0.08
The standard deviation of size (mm^2)	0.30
Average severity	0.16
Max severity	0.16
Min severity	0.16
Average y coordinate from the center of the page (mm)	-5.95
Average x coordinate from the center of the page (mm)	24.27

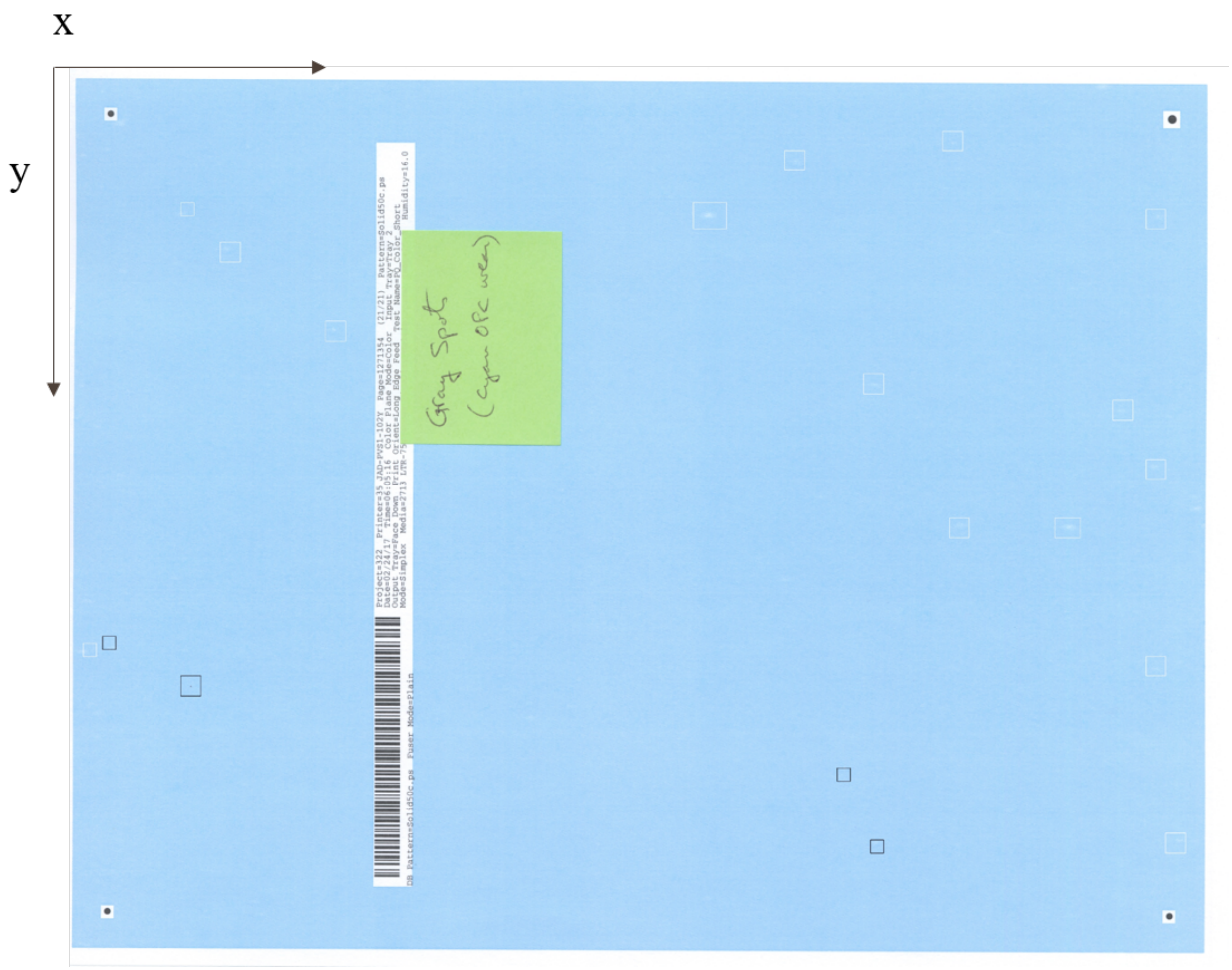


Figure 1.12. Detection output of a test page. The boxes indicate the defects that have been identified (White boxes stand for gray spots, black boxes represent solid spots).

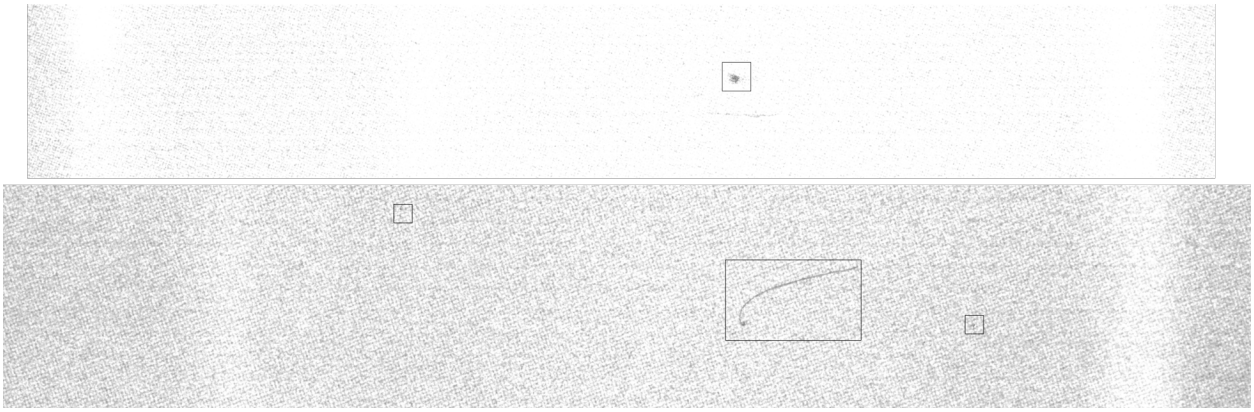


Figure 1.13. The detection output of touchpad products. The boxes indicate the defects that have been identified.

1.4 Conclusion

In this project, we develop a coarse-to-fine method to automatically detect local defects, including the initial detection by thresholding, and the secondary refinement by a trained model. Different from previous works, we propose blockwise features to describe attributes of visible defects in the candidate area, which can help us determine the exact defect type. With these proposed features, we build a blockwise dataset of local defects for future training. A decision tree model is applied to produce more accurate results for visible local defects. Finally, we agglomerate blockwise results to generate a feature vector for each test page, which can be used for further assignment of page rank.

2. FACE SET RECOGNITION WITH MULTI-COLUMN NETWORK

2.1 Introduction

Face recognition is widely applied in areas including video surveillance, mobile payment, *etc.*. In practical applications, we are often faced with the problem of comparing two sets of face images. Face set recognition is designed to deal with such cases. Different from single face identification and verification, a face set may include images with various quality, head angle, illumination conditions, and an unknown number of images. Besides face feature extraction and similarity comparison, the key issue in face set recognition is to find an appropriate representation for the target set to increase the accuracy.

Due to the great success of deep neural networks on image classification [16][17][18], the approach to face set recognition follows the similar manner: first train a deep convolutional neural network on face image classification [19][20][21][22][23], and then acquire a set-wise descriptor by aggregating the features of single images [24][25][26][27][28][29]. The identification and verification process is conducted on the set descriptor domain via cosine distance. With the help of large-scale datasets [30][31][32][33][34], methods following the procedures above have achieved impressive results on challenging benchmarks including IJB-A [35], IJB-B [36], IJB-C [37], etc.

In this project, we apply a multi-column network to aggregate the face features into a unified fixed-length feature vector to efficiently conduct the set-wise comparison. We apply SphereFace [38] as the backbone network in face feature extraction, and use a multi-column network for face feature aggregation.

2.2 Experimental Setup

The workflow consists of the following steps: face alignment, face feature extraction, and set feature aggregation as plotted in Figure 2.1. This process will be introduced in detail in the following sections.

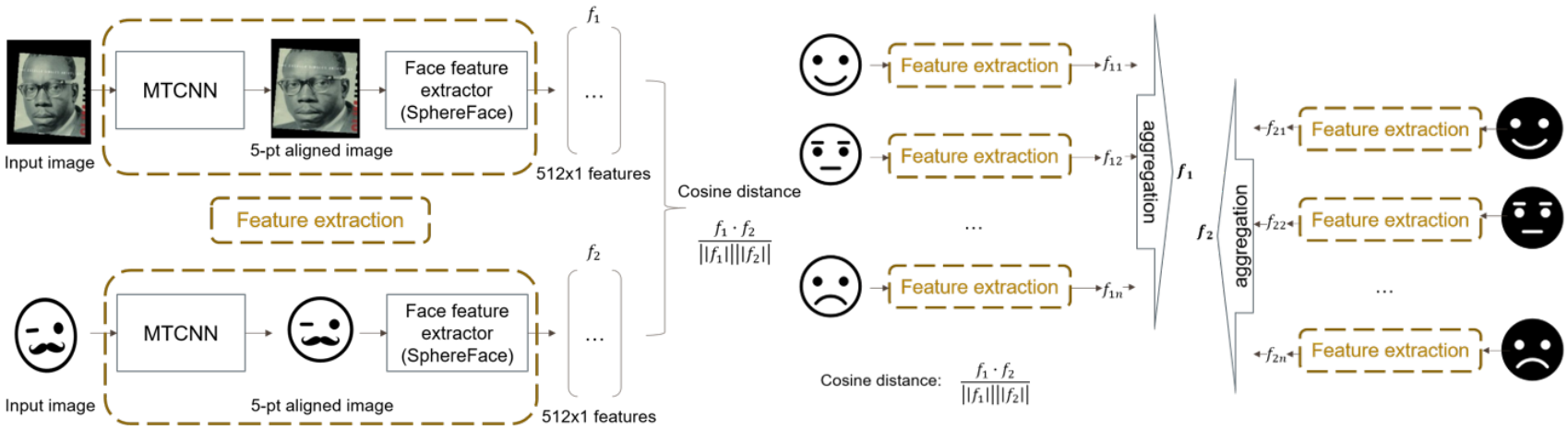


Figure 2.1. Workflow of Face Set Recognition.

2.2.1 Face Alignment

Since the input images are of different sizes and face regions, directly sending the whole image into the face feature extractor would also include a lot of unwanted information from the background. In order to improve the accuracy of the face feature extractor, face alignment is needed as a pre-processing step: locating and cropping the face region, and aligning the face to a normalized location.

We apply MTCNN [39] to detect the 5-point facial landmarks of the face. Then the face images are cropped using a similarity transformation. Each pixel in the original RGB images(in the range $[0, 255]$) is normalized to $[0, 1]$ for the following steps.

However, since there are many images with occlusion, blurring, rare-angle, and bad illumination issues, MTCNN cannot detect the face or facial landmarks in every image set. On the complete IJB-C dataset, MTCNN can only successfully find 15,928/22,257 still images (missing rate: 28.4%), and 80,510/118,484 for video frames (missing rate: 32.0%). Taking Set 4 as an example, MTCNN failed on 7 images out of 145 inputs as shown in Figure 2.2.

Figure 2.2. Images that are failed to be detected by MTCNN.

So here comes the question: what should we do with the images without landmarks? In order to address the problem, we build a small subset out of IJB-C with 15,000 pairs of faces and conduct the 1:1 verification experiment on them. Note that the 15,000 matching pairs could include a face image with landmarks, and a face image without landmarks. We can conduct a similarity transformation to align the images with landmarks, and for the remaining images, we have no way to align them.

Here, I conduct the following four experiments:

1. Only make the comparison between aligned faces and ignore the remaining pairs (ideal case: only aligned-aligned pairs);

2. Compare all pairs of aligned faces and un-aligned faces (possible cases: aligned-aligned, aligned-unaligned, and unaligned-unaligned pairs);

3. Compare all pairs, but when one of the faces fails to be detected, treat both of the images as unaligned (possible cases: aligned-aligned, unaligned-unaligned);

4. Don't align any images and conduct comparisons on them (base case: unaligned - unaligned).

Table 2.1 shows the results of all four experiments in TAR (True Acceptance Rate) under different FARs (False Acceptance Rates) using the workflow shown in Fig. 2.1. According to these results, we can conclude that the alignment of faces can greatly improve the performance of face feature extractors. Hence we can acquire high accuracy on face identification. On the other hand, the existence of unaligned images will decrease the matching accuracy. A very interesting fact is that comparing the aligned face and unaligned face in the same pair will actually decrease the matching accuracy, which could be because the extracted features are not from the same domain. As a result, our following experiments apply the settings from Experiment 3.

Table 2.1. Influence of Face Alignment on Face Pair Identification

Setting Type	TAR@FAR 0.001	TAR@FAR 0.01	TAR@FAR 0.1
1	0.7154	0.8495	0.9430
2	0.4457	0.5651	0.7309
3	0.4589	0.5939	0.7489
4	0.2055	0.3473	0.5766

2.2.2 Single Face Feature Embedding

To extract the feature vector of each input image, we apply SphereFace [38] trained on the CASIA-WebFace [31] as the extractor.

We fine-tune the feature extractor on the CASIA-WebFace dataset [31]. We test the final model on LFW [40][41] and our subset of IJB-C [37] to measure and compare the performance of the face extractor. The TAR@FAR curves are shown separately in Figures 2.3 and 2.4.

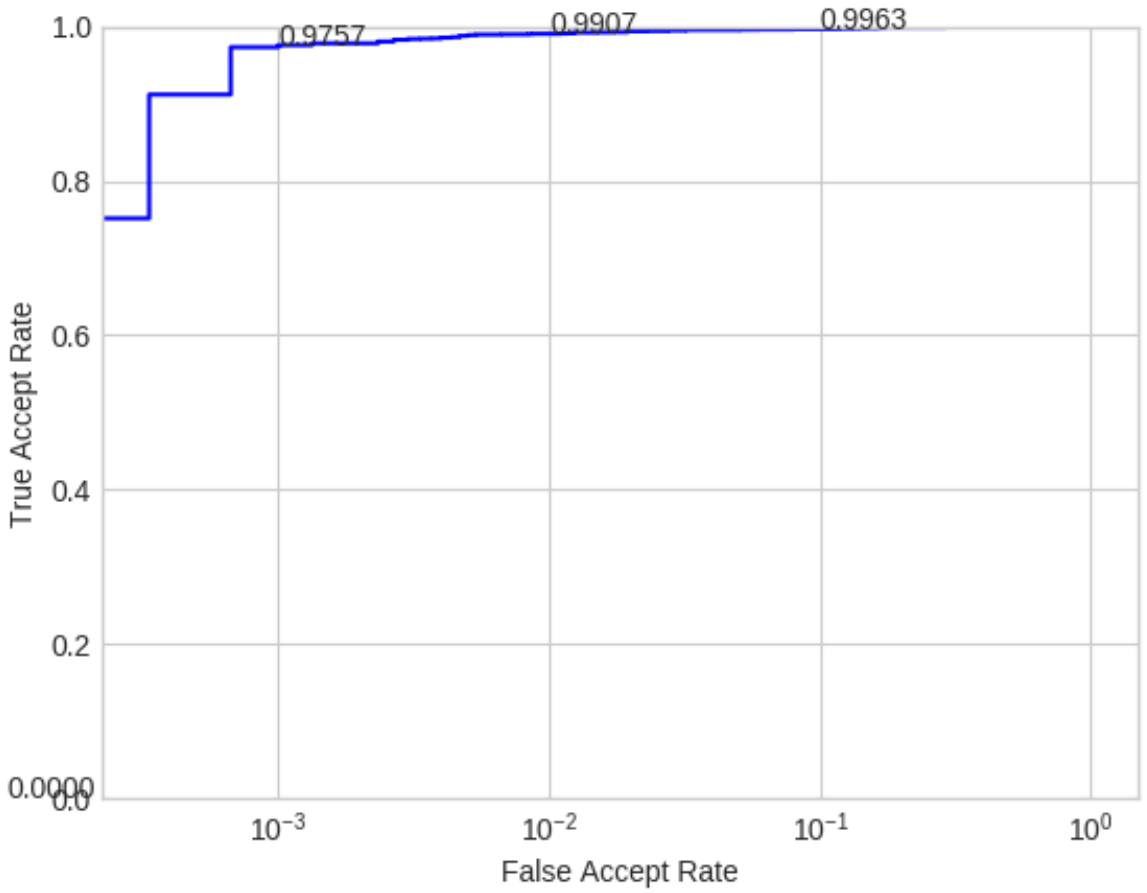


Figure 2.3. TAR@FAR curve of the SphereFace model on the LFW dataset.

Compared with the public released SphereFace model's performance (shown in Table 2.2), our fine-tuned model shows even higher accuracy on the LFW dataset.

2.2.3 Feature Aggregation

We consider the following two factors for face feature vector aggregation: the quality and the content of the input images. The multi-column network is attached at the end of

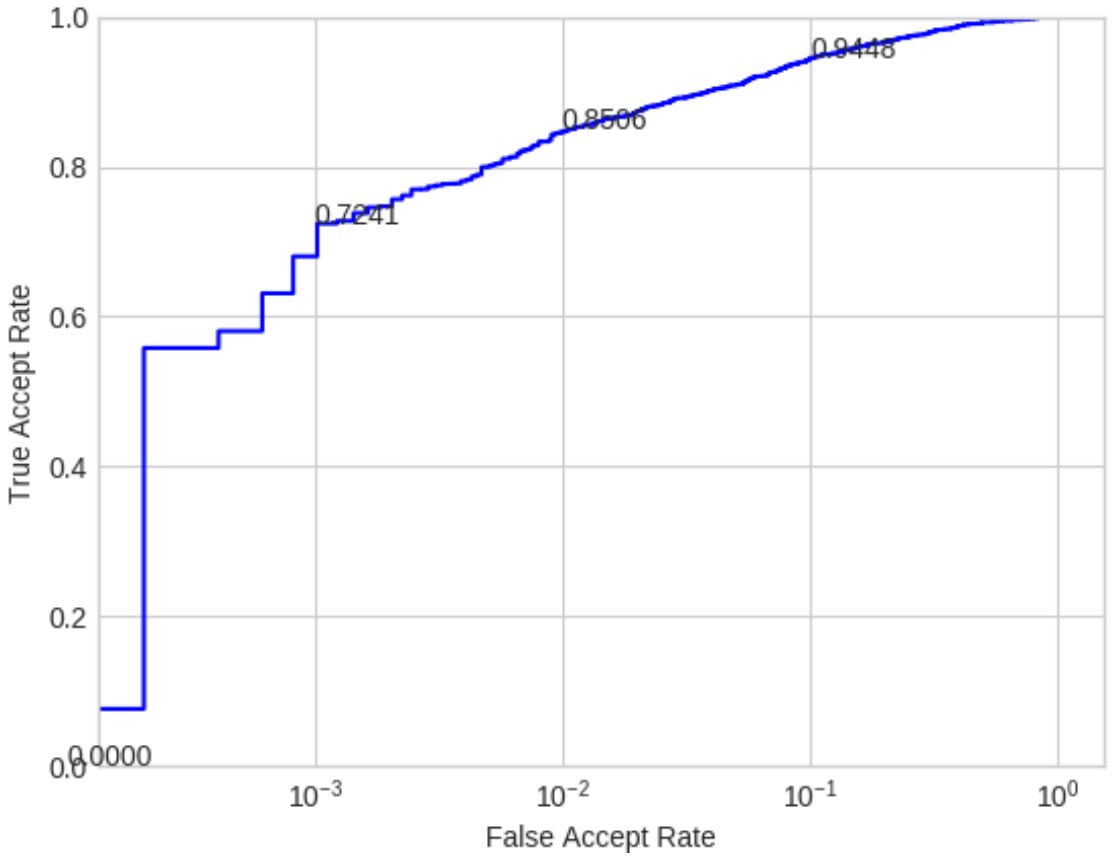


Figure 2.4. TAR@FAR curve of the SphereFace model on the IJB-C subset.

Table 2.2. Influence of Face Alignment on Face Pair Identification. We compare the accurary of different methods on the LFW [40] and YTF [30] datasets. For a fair comparison, we also denote the number of models used in each method and the source of their training data. For the methods using private datasets, we include the number of training images instead.

Method	Model number	Data	LFW	YTF
DeepFace [42]	3	4M	97.35	91.4
FaceNet [43]	1	200M	**99.65**	95.1
Deep FR [44]	1	2.6M	98.95	**97.3**
DeepID2+ [45]	1	300K	98.70	N/A
DeepID2+ [45]	25	300K	99.47	93.2
Baidu [46]	1	1.3M	99.13	N/A
Center Face [47]	1	0.7M	99.28	94.9
Yi *et al.* [31]	1	WebFace [31]	97.73	92.2
Ding *et al.* [48]	1	WebFace [31]	98.43	N/A
Liu *et al.* [49]	1	WebFace [31]	98.71	N/A
Softmax Loss	1	WebFace [31]	97.88	93.1
Softmax+Contrastive [50]	1	WebFace [31]	98.78	93.5
Tripet Loss [43]	1	WebFace [31]	98.70	93.4
L-Softmax Loss [49]	1	WebFace [31]	99.10	94.0
Softmax+Center Loss [47]	1	WebFace [31]	99.05	94.4
SphereFace [38] (Adopted by us)	1	WebFace [31]	**99.42**	**95.0**

the SphereFace and generates the quality factor and content factor with two fully connected layers. The structure of the multi-column network is shown in Figure 2.5.

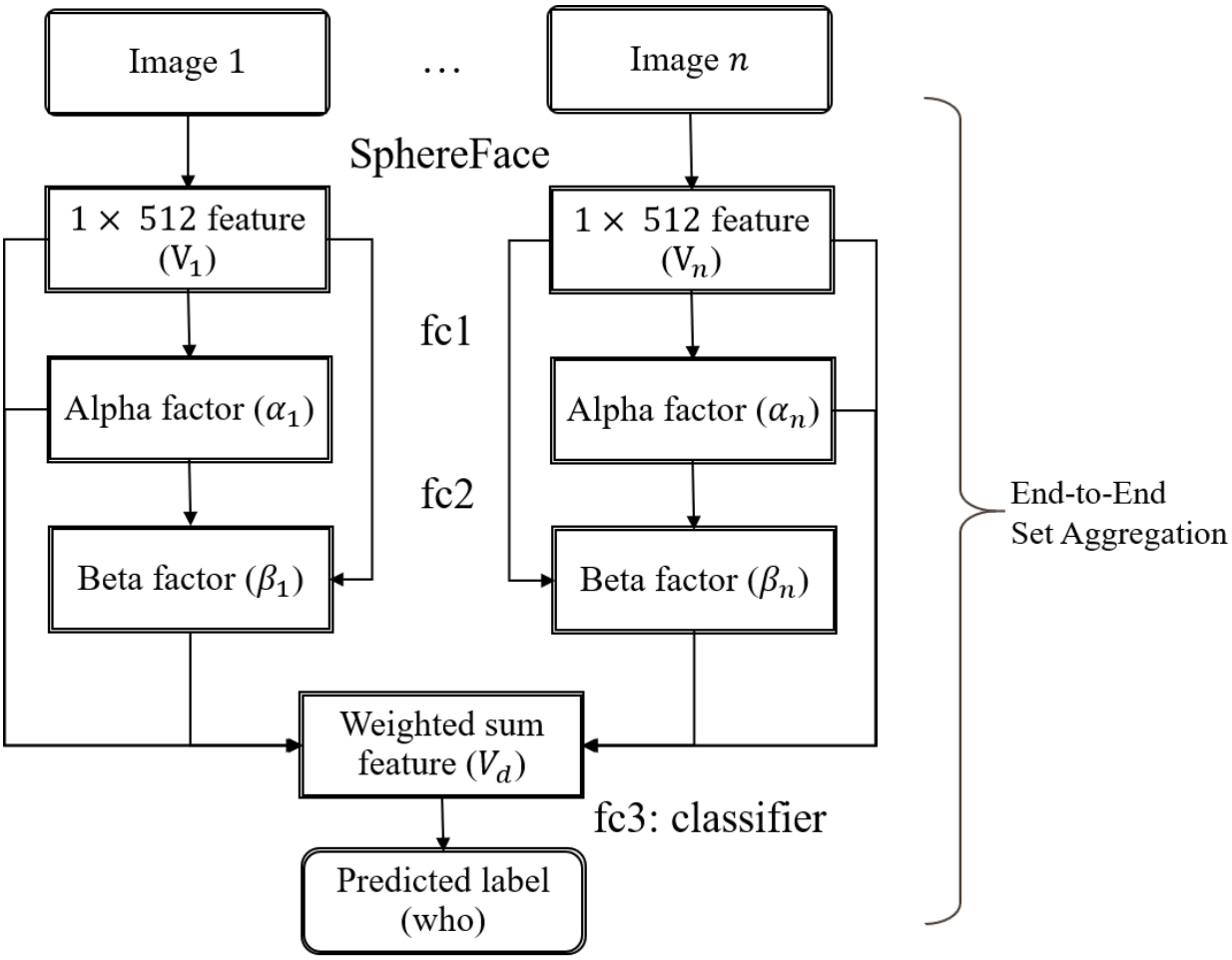


Figure 2.5. multi-column network structure.

The α and β factors generated by the two columns serve as the weighting factor for each face. The final set descriptor can be expressed by

$$V_d = \frac{\sum_i \alpha_i \beta_i V_i}{\sum_i \alpha_i \beta_i} \tag{2.1}$$

where i refers to the i-th input image, and V refers to the feature vector with 1×512 dimensions.

2.3 Experiment Results

2.3.1 Training Details

The multi-column network is trained on MsCeleb-20K [32]. In order to make the network implicitly learn the weight factors, we train the network with three images in a group and use the angular softmax loss as the supervision to maximize the difference between different faces while minimizing the distance of the faces from the same identity. Since the first face feature extractor network SphereFace is already fully trained, we freeze the weights and only train the multi-column layers with the SGD [51] optimizer. The training loss and verification accuracy over the epochs are shown in Figure 2.6.

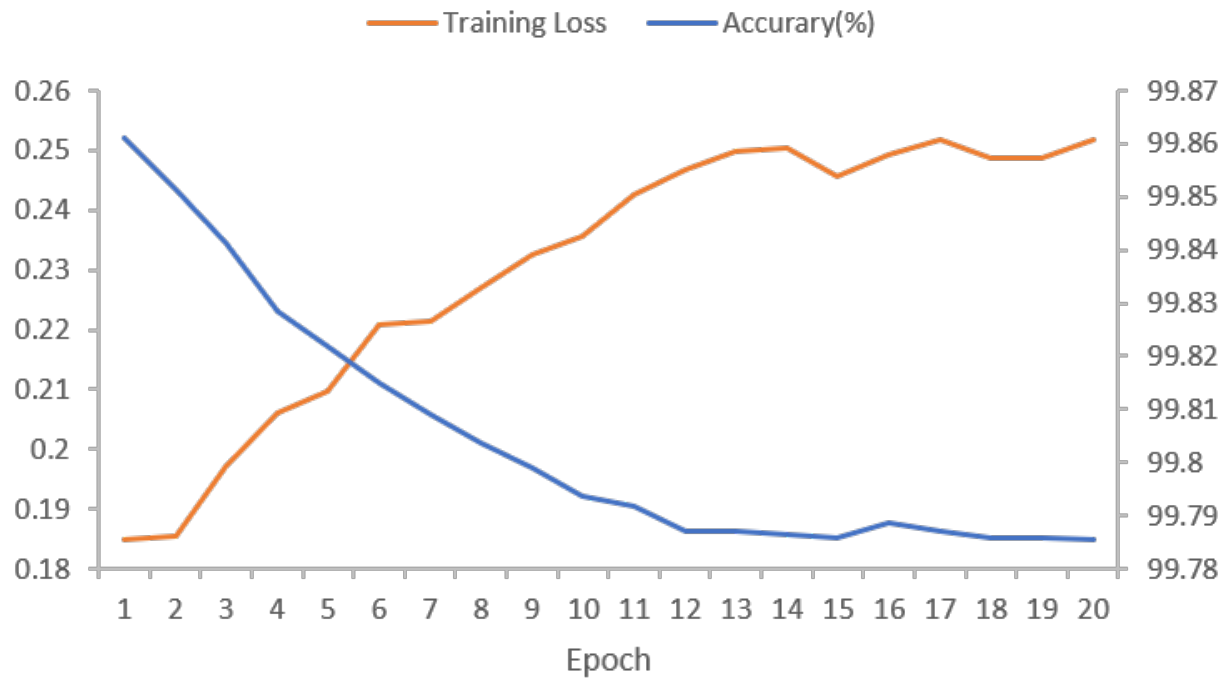


Figure 2.6. Plot of the loss and accuracy in the training process of the multi-column network shown in Fig. 2.5.

2.3.2 Results on the IJB-C Benchmark

The IARPA Janus Benchmark-C face challenge (IJB-C) [37] is related to unconstrained in-the-wild face images. It defines eight challenges addressing verification, identification, *etc.* of full-motion videos. This challenge is supported by the IJB-C set of 138000 face images, 11000 face videos, and 10000 non-face images. It includes three different tasks:

1. 1:1 verification: the goal is to compare the probe templates with gallery templates (G1/G2). The verification list includes around 16 million matching pairs, with $10^0 \approx 10^1$ faces in each template;

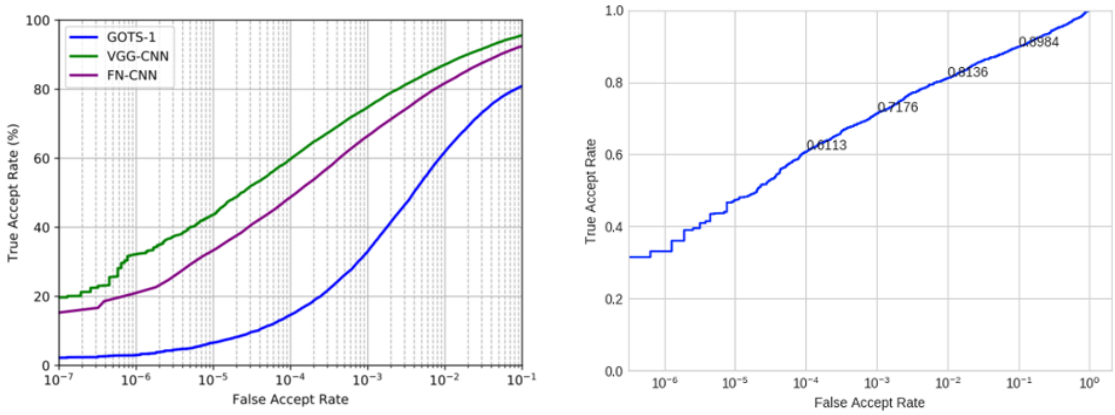
2. 1:N close-set identification: given the probe templates, the goal is to find the closest identities in G1/G2;

3. 1:N open set identification: it is similar to 1:N close-set identification, but the goal is more than finding the correct identities in G1/G2. It also includes rejecting the ones that are not in G1/G2.

The 1:1 verification results for the NIST benchmark and our model are plotted in Figure 2.7. Our results for the 1:N identification (Task 3 above) is plotted in two graphs: the ROC graph in Figure 2.8, and the CMC graph in Figure 2.9. The quantitative comparison between our results and the NIST benchmarks is shown in Table 2.3, where we show the FPIR (False Positive Identification Rate) and the accuracy of the 1, 5, and 10 top-ranked predictions. Our result outperforms the NIST benchmarks when the FPIR is low.

Table 2.3. Comparison of our result with NIST benchmark on IJB-C :N identification

Name	FPIR0.001	FPIR0.01	FPIR0.1	Rank1	Rank5	Rank10
GOTS	0.0266	0.0578	0.156	0.3785	N/A	0.6024
FaceNet [43]	0.2058	0.3240	0.5098	0.6922	N/A	0.8136
VGGFace [44]	0.2618	0.4506	**0.6275**	**0.786**	N/A	0.892
Ours	**0.3253**	**0.4728**	0.6197	0.712	0.7805	0.8065



NIST benchmark. Our results.

Figure 2.7. Results of the IJB-C 1:1 verification task.

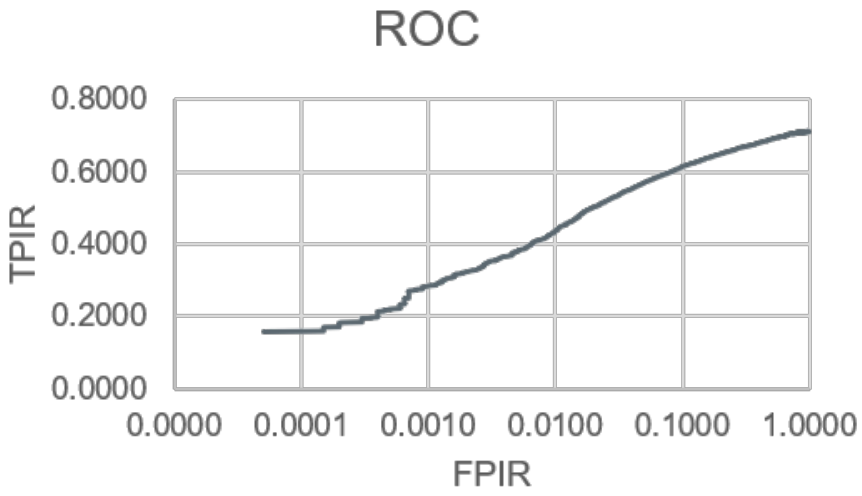


Figure 2.8. ROC of IJB-C 1:N identification task for our method.

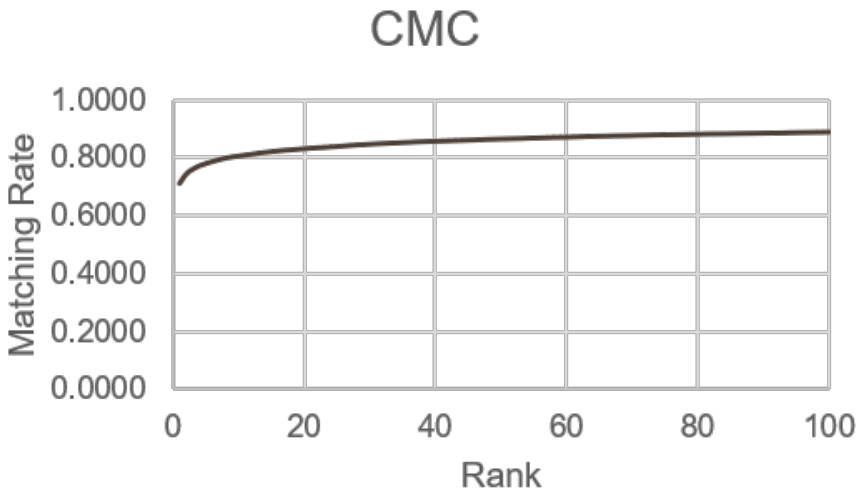


Figure 2.9. CMC of IJB-C 1:1 verification task for our method.

2.4 Conclusion

In this project, we have presented a multi-column network for face set representation and recognition. It takes all the images in the set as input and extracts their features and weights based on the quality and content factors, and aggregates them to a compact representation of the face set. This method only adds an extra $\approx 6k$ parameters to the face feature extraction models, and can be widely used in scenarios including video surveillance, mobile payment, and other tasks.

3. FACE ALIGNMENT

3.1 Introduction

The evolving algorithms for 2D facial landmark detection empower people to recognize faces, analyze facial expressions, *etc.*. However, existing methods still encounter problems of unstable facial landmarks when applied to videos. Because previous research shows that the instability of facial landmarks is caused by the inconsistency of labeling quality among the public datasets, we want to have a better understanding of the influence of annotation noise in them. In this project, we make the following contributions: 1) we propose two metrics that quantitatively measure the stability of detected facial landmarks, 2) we model the annotation noise in an existing public dataset, 3) we investigate the influence of different types of noise in training face alignment neural networks, and propose corresponding solutions. Our results demonstrate improvements in both the accuracy and the stability of detected facial landmarks.

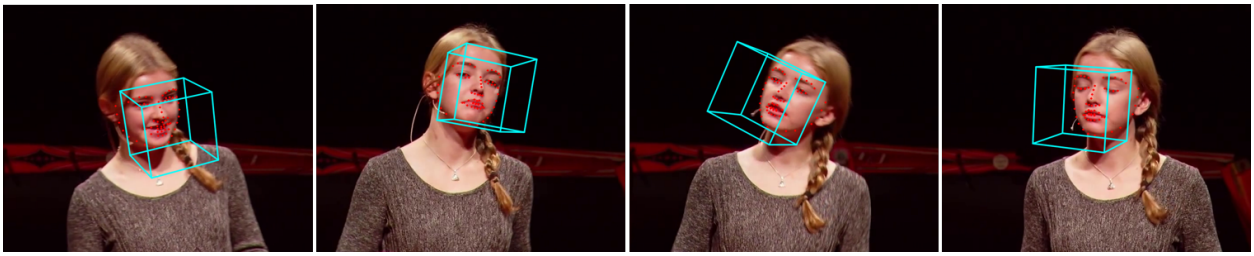


Figure 3.1. Video frames from the 300-VW dataset with the detected 3D bounding box and 68 landmark points.

2D facial landmark detection, as shown in Figure 3.1, is a fundamental technology behind face recognition [52], expression recognition [53], augmented reality 3D mask rendering, *etc.*. Besides precise and accurate localization, several problems have attracted more and more attention in recent years: landmark stability, facial landmark detection under extreme conditions such as occlusion, rare angle faces, low luminance, large movement, *etc.*. Datasets play a pivotal role in addressing these problems in landmarks localization, including the early AFLW dataset [54] with 21 point markup and the more recent 300-W [55], 300-VW [56][57][58] with 68 point annotations, and WFLW [59] with 98 manually annotated

landmarks. These datasets have not only grown larger in scale, but also have become more diverse in attributes, including occlusion, pose, make-up, illumination, motion, and facial expressions.

Although the scale of facial landmark datasets is growing, it is still far from being comparable with the tremendous size of face recognition datasets such as Ms-Celeb-1M [32], which consists of 10M images of 100K celebrities. These large-scale datasets, along with research on data cleaning [60], drive the development of new methods to achieve better results in face recognition. The main factor that constrains the scale of facial landmark datasets is that, in the current stage, the labeling of landmarks heavily relies on manual annotation and verification. Different from establishing face recognition datasets whose labels can be cleaned automatically, building a facial landmark dataset is tedious and time-consuming. Besides, [61] noticed that human annotations inherently have flaws in precision and consistency: the positions of the same landmark point annotated by different people vary a lot, even those of the points with clear features (e.g., corner of the mouth). The variance in the training data would degrade the stability of landmark detectors, thus leading to perceptually unpleasant jitters when the detector is applied to videos. Traditional landmark detection frameworks treat each frame of a video as an individual input and pay little attention to temporal consistency; as a result, it is hard for the output points to maintain a consistent visual presence in consecutive frames.

Considering the fact that the inconsistency of landmark annotations widely exists in popular facial landmarks datasets, such as 300-W and 300-VW, we regard the unwanted jitters as a type of noise and design relevant experiments around this concept.

The first goal of our work is to understand the noise in facial landmarks and how it will influence the training of deep convolutional neural networks (CNN). In other words, we concern ourselves with the relationship between the training set's noise and the model's performance, the extent that the noise can be reduced, and the best method to get a clean output. To achieve this goal, we propose two plausible metrics that measure landmarks' stability. Although we have to admit that currently it is almost impossible to get noise-free landmark points in either the training or the detection stage, a better understanding of the

concerns expressed above would be beneficial for the design of more robust algorithms for real-world applications.

The second goal of this project is to propose a complete workflow to help decrease the inconsistency of facial landmark annotations that exists in a wide range of popular public datasets. In this project, we show that training on a corrected dataset can actually improve the detector's performance. Our proposed workflow can also boost the performance of existing methods. The corrected dataset used in training our facial landmark detector consists of approximately 4,000 still images and 300 videos of 200 identities. Due to the nature of the dataset source, these images exhibit great variations in scale, pose, lighting, and occlusion. For a better comparison, we also carried out experiments on corrupted datasets by injecting noise on landmark annotations following previous research [61]. By controlling the amount of additive noise, this study helps us understand the noise's influence on detection accuracy quantitatively.

The third goal of this project is to explore practical solutions for real-world applications, including dense facial landmarks, augmented reality (AR) mask rendering, *etc.*. Such application environments usually require real-time and temporally consistent landmark detection on mobile devices. Based on the understanding of noise, post-processing methods are introduced to generate the required results with small additional computation costs. Our experimental results show that these methods can significantly reduce perceptually annoying jitters as a complement to deep CNN.

3.2 Related Work

Semi-automatic Facial Landmarks Annotation: To aid the manual annotation work, Christos *et al.* [62] proposed a semi-automatic methodology for facial landmark annotation in creating massive datasets. They used the annotated subset to train an Active Orientation Model (AOM) that provides an initialization to non-annotated subsets, and then classifies the results to "good" and "bad" manually. However, this kind of method can only reach a relatively accurate annotation by cleaning out the obviously "bad" examples, but cannot avoid the jitters among different annotations.

Face Alignment and Tracking: To improve face alignment in the video, Peng *et al.*. [63] proposed an incremental learning method for sequential face alignment. To better make use of the temporal coherency in image sequences, Peng *et al.* [64] designed a recurrent encoder-decoder network model for video-based face alignment, where the encoding module projects the input image into a low-dimensional feature space, whereas the decoding module maps the features to 2D facial point maps. The recurrent module demonstrates improvements to the mean and standard deviation of errors by taking previous observations into consideration. However, this module unavoidably increases the network complexity and training difficulty. Besides, many researchers applied tracking as an extension of face alignment, though it always results in drifting and loss of accuracy of facial landmarks. Conducting face tracking usually involves generic facial landmark detection, the combination of model-free tracking and re-initialization [65]. Khan *et al.* [66] proposed a synergistic approach to eliminate tracking drifts, in other words, to apply face alignment when drifting happens. Still, stepping happens when shifting between tracking and detection results. Barros *et al.* [67] applied a Kalman filter to fuse the tracking and detection.

3.3 Methodology

3.3.1 Noise Modelling

Noise in Public Datasets

Naturally, people assume that the ground truth in popular public datasets is accurate and precise. However, Dong *et al.* [61] has noticed the existence of hand annotation noise. Here, we will look into two most commonly used datasets, 300-W [55] and 300-VW [56][57][58]:

300-W provides 68 2D landmark annotations for 3,837 face images. These images were split into four sets: training, common testing, challenging testing, and full testing. In this project, our base detector is trained on the 300-W training set. In addition, we evaluated the detector's performance and modeled the output noise on all three 300-W test sets.

300-VW [56] is a video dataset consisting of 50 training videos with 95,192 frames. Its test set contains three subsets (1, 2, and 3) with 62,135, 32,805, and 26,338 frames, respectively. Among the three categories, Subset-3 is the most challenging. We apply the

proposed method to correct landmarks of the 300-VW full set and report results on all three subsets.

Figure 3.2 shows some examples of inaccurate annotations in the 300-W and 300-VW datasets. Images in the dataset can be categorized into four categories: **accurate subset**: data samples with acceptable ground truth; **inaccurate eyes**, **inaccurate mouth**, and **inaccurate contour**. Besides, there are also images with more than one inaccurate facial component.

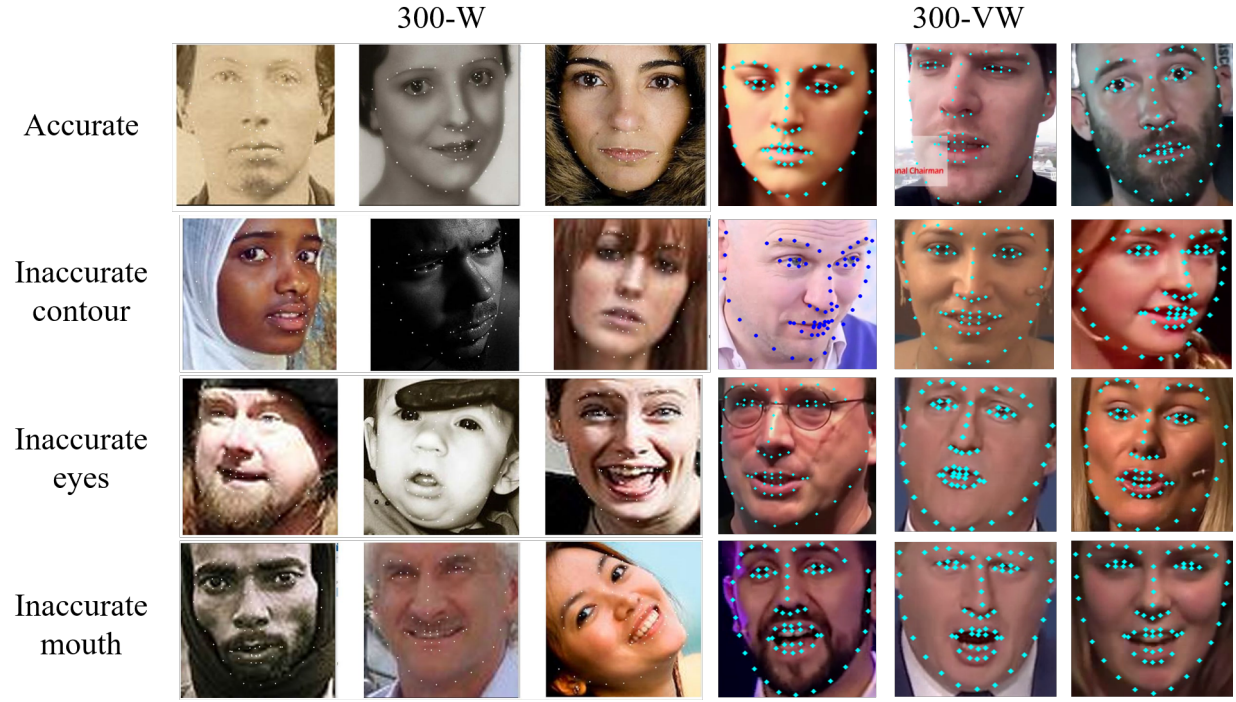


Figure 3.2. Noisy annotations in public datasets. The images in the left 3 columns are from 300-W, and the images in the right 3 columns are from 300-VW. The quality of the annotations is not consistent among these two well-known datasets. The reader is advised to zoom in to see the annotations.

Metrics

In this section, we will introduce the widely used metrics for facial landmark accuracy, and our proposed methods to measure the stability of detected facial landmarks quantitatively.

Accuracy reflects the difference between the predicted result and the ground truth. A good facial landmark detector should produce results of low prediction error for any given inputs, including still images and video frames, which corresponds to a small prediction error.

The normalized mean error (NME) is widely used as the evaluation metric for accuracy. It is defined as follows:

$$\text{NME} = \frac{\frac{1}{N}\sum_{\text{i}=1}^{N} L_2(f_\theta(\hat{x}_\text{i}), y_\text{i})}{d}, \tag{3.1}$$

where y and $f_\theta(\hat{x})$ denote the ground truth and predicted points, respectively. N is the number of landmarks on a face (in this project, $N = 68$), and d represents the outer ocular distance (the distance between outer corners of eyes) for normalization. It is worth noting that in some previous papers, the distance between centers of eyes (inter pupil distance) was also used for normalization.

Precision reflects the robustness of a model when given inputs with different kinds of noise, *e.g.*pixel-wise noise including camera shot noise, Gaussian blur, or re-centering, *etc.*, which can exist in video frames. These kinds of noise will not change the spatial distance among different facial components of the ground truth, but may cause jitters in detection outputs. Since the absolute locations of facial landmarks should be unchanged, we can define the variance of the detected points for precision as follows:

- Standard Deviation (STD). This metric does not require annotated ground truth but a test set of still images, *e.g.* video frames of an unmoved face from a fixed camera. In this case, each frame is naturally injected with different camera shot noise. The normalized standard deviation of the landmark locations in the same video can be used as a metric for stability:

$$STD = \frac{1}{d}\sqrt{\frac{\sum_{\text{i}=1}^{n}(f_\theta(\hat{x}_\text{i}) - \overline{f_\theta(x)})^2}{n-1}}, \tag{3.2}$$

  where i refers the index of frames: $1, ..., \text{i}, ...n$. This output of this equation is an array composed of the standard deviation of each landmark coordinate. Larger standard deviation indicates more jitters.

- Standard Deviation of Difference (SDD). This metric requires annotated ground truth as a reference: Firstly, the difference between the ground truth and detection results for each frame is calculated as $\Delta y_{\mathrm{i}} = f_\theta(\hat{x}_{\mathrm{i}}) - y_{\mathrm{i}}$. In the ideal case where the detection results exactly follow the ground truth, the variance of the difference should be low no matter how big the difference is. Larger SDD values indicate more jitters. For a given video of frames $i \in \{1, \ldots, n\}$, the formula for this metric is given by:

$$SDD = STD(\Delta y) = \frac{1}{d}\sqrt{\frac{\sum_{\mathrm{i}=1}^{n}(\Delta y_{\mathrm{i}} - \overline{\Delta y})^2}{n-1}} \tag{3.3}$$

We borrow the idea from similar definitions of noise made in other fields that require assessment over variant application backgrounds [12][68]. In a similar manner, we can then define the detection noise as the deviation between the detection results and the target values, as mentioned in Section 3.3.1. We can use the distribution of the detection noise as a tool to visualize the detector's robustness and stability. Figure 3.3 shows an example of the noise of each landmark point in both the X and Y coordinates that is calculated from a "pseudo" video generated by the augmentation method mentioned later in Section 3.4.1. In this figure, the detection noise is plotted in bar graphs with error bars, where blue and red bars correspond to the mean value of the detection results' difference from ground truth in the X and Y directions, respectively. The error bars are calculated from the standard deviation of these differences among a group of test images, as defined previously in this project. We can get some interesting information from this plot: the most unstable points are the eyebrows' ends and then the lowest point of the lips, which are actually the most deformable points on our face. Besides, the color fading of the eyebrows' ends also throws challenges to the landmark detector.

Figure 3.4 shows the detection noise plotted as a 2d histogram with both X and Y directions, which serves as another way to visualize the spatial distribution of the noise. In previous research papers, for simplicity, all noises are assumed to be of a Gaussian distribution. In this graph, the scattered map of points without clear peaks usually indicates that the noise severely deviates from a Gaussian distribution. This graph can be used to select

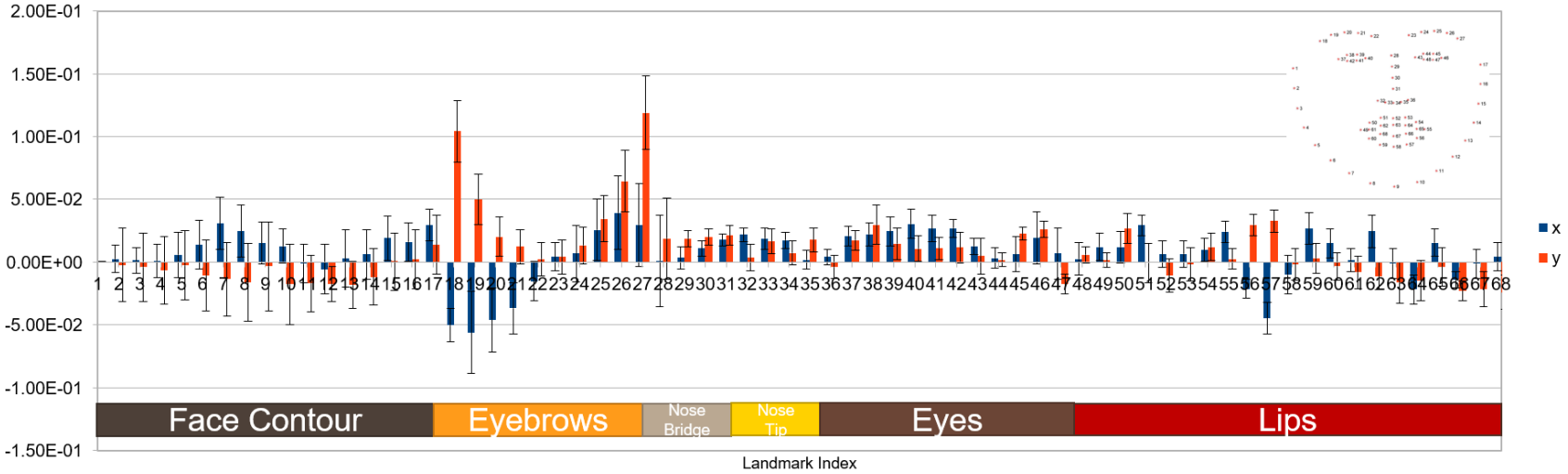


Figure 3.3. Example of the detection noise(SDD (Equation 3.3)) in X (blue bar) and Y (red bar) coordinates. The X-axis denotes the index number of the facial landmark, *e.g.* points $1 \sim 17$ represent face contour points, 18∼27 stand for eyebrow points, *etc.*. The solid bars are the mean values of the detection results' difference, and the whiskers represent the standard deviation of the difference as defined before. Bigger error bars of points indicate that these points are unstable in that direction, while mean values can tell us the prediction error, or "bias" from ground truth. Ideally, we wish the model's prediction result to be an unbiased estimate of the ground truth, which corresponds to zero mean values in this graph.

stable landmarks as reference points for further face alignment in Augmented Reality (AR) applications.

The noise in the detection result is connected with inaccurate locations and jarring visual effects. Thus, methods including tracking and temporal filtering are applied in post-processing to reduce the noise as discussed in Section 3.3.2. Although the physical noise cannot be eliminated due to the limitation of the input data, it is possible to attain better results using prior knowledge about the noise distribution.

Theoretical Assumptions

Assume we have a set of unreliable annotations $(\hat{y_1}, \hat{y_2}, ...)$ of facial landmarks. Denoting the "true and only" landmark coordinate as y, we describe their relationship by the following equation:

$$\hat{y} = y + \delta(y), \tag{3.4}$$

where $\delta(x)$ is the difference between the annotated data and the true value, or to say, the human annotation noise. Usually, the noise distribution is assumed to have zero mean and finite variance [69]:

$$E[\delta(y)] = \sum_{\mathrm{i}=1}^{n} \delta(y_{\mathrm{i}}) p_{\mathrm{i}} = 0, n \to \infty \tag{3.5}$$

Based on this assumption, we can get

$$E[\hat{y}] = E[y + \delta(y)] = y + E[\delta(y)] = y, \tag{3.6}$$

indicating that the expected value of the human-annotated value is the true unknown y. In practice, it is possible to approach y through a learned mapping function with a sufficient number of randomly distributed training samples. Most facial landmark models are trained with L2 loss (MSE Loss): The influence of noise in the target data is further discussed below.

$$L2Loss = \frac{1}{N} \sum_{\mathrm{i}=1}^{N} ||f_\theta(x_{\mathrm{i}}) - \hat{y}_{\mathrm{i}}||_2^2, \tag{3.7}$$

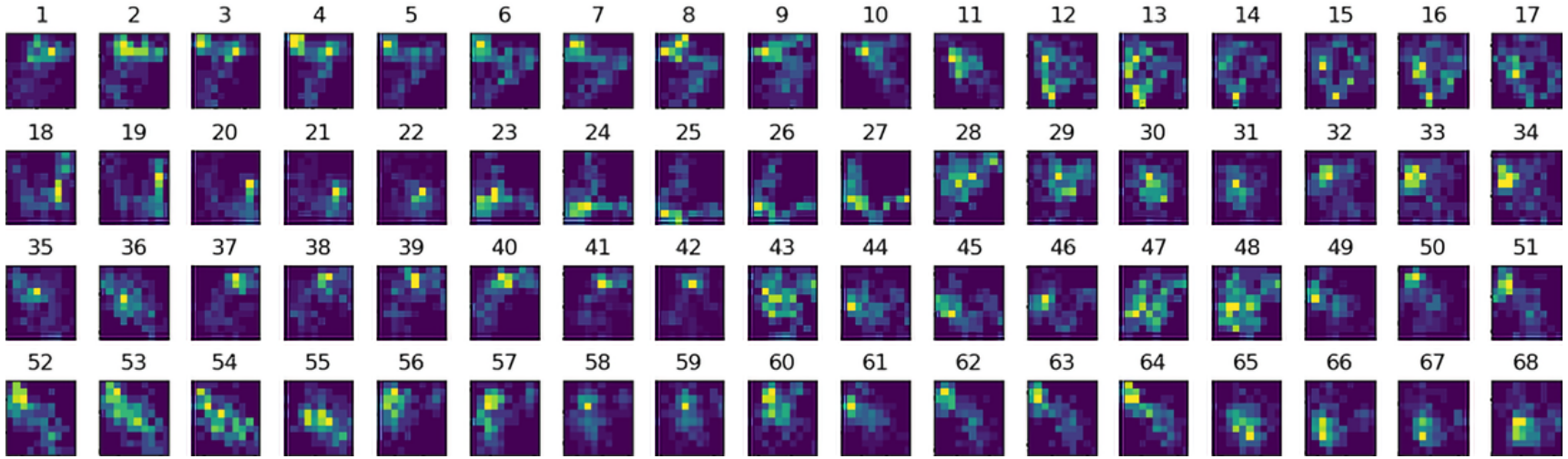


Figure 3.4. Example of the noise (SDD (Equation 3.3)) plotted as a 2d histogram. Each histogram represents the noise distribution of every facial landmark point in the X and Y coordinates. Ideally, a stable point should have a Gaussian distribution with a clear peak. If the prediction result is an unbiased estimate of the ground truth, this peak should be located at the zero point, i.e. the center, of the histogram in both X and Y.

where x_i denotes the input data, and f_θ is the mapping function. Thus, $f_\theta(x_i)$ is the network's output array, $\hat{y}_i$ is the value of the ground truth, and N refers to the number of elements in the network's output array. For the common case of 68-point facial landmarks, $N = 136$.

The optimization goal of L2 Loss is given by:

$$\begin{aligned} &argmin_\theta E_{\hat{y}_i}[(f_\theta(x_i) - \hat{y}_i)^2] = \\ &argmin_\theta E_{f_\theta(x_i)}\left[[E_{\hat{y}_i|f_\theta(x_i)}[(f_\theta(x_i) - \hat{y}_i)^2]\right] \end{aligned} \tag{3.8}$$

As mentioned before, the noise in landmark positions is assumed to be zero-mean with finite variance for all images in our dataset. A network trained with L2 loss, from a statistical point of view, will produce outputs that approximate the conditional mean of the target values in the training set:

$$E[\hat{y}_i|f_\theta(x_i)] = E[\hat{y}_i] \tag{3.9}$$

As defined in Equation 3.6, the expectation of $\hat{y}_i$ can approach the unobserved true value y_i if there are enough random inputs.

However, if $\delta(y)$ belongs to other types of noise, L2 loss may not guarantee that the destination of optimization is $E[\hat{y}_i] = y_i$. For example, if injected with salt-and-pepper noise (impulse noise), the conditional median would be better at approaching the true value, which can be learned by least-absolute-value training with L1 loss. Following a derivation similar to that above, we can get the correspondence between loss function and noise type shown in Table 3.1.

Table 3.1. Loss function for each type of noise.

Noise Type	Loss Function
Additive Gaussian noise Poisson noise	L2 loss
Bernoulli noise (binominal noise)	Modified L2 loss [70]
Salt-and-pepper noise	L1 loss
Random-valued impulse noise	L0 loss

Researchers from the image denoising area have already noticed the connection between noise type and training loss type [70]. In [71], the model learns to restore images by only looking at corrupted data with synthetic noise. Previous research [72] also compared training facial landmark detectors using L1 and L2 loss, and a custom loss function. The different performances of acquired models, along with our noise modeling results, indicate that the actual noise in public training data is more complex than that based on theoretical assumptions.

3.3.2 Methods to Reduce Noise

Based on previous assumptions about noise, we propose a method to reduce the spatial noise with the temporal information from adjacent frames. For each image, we assume the relationship between the ground truth and the prediction is:

$$p_{\mathrm{i}} = gt_{\mathrm{i}} + err_{\mathrm{i}}, \tag{3.10}$$

where gt denotes the ground truth, p stands for the prediction(or annotation), and err is the error for view i. The landmark points of different frames can be regarded as individual samples around the ground truth. The difficulty lies in calculating the errors from different frames under the same view. We choose optical flow to find the corresponding points between two images; this process can be expressed as follows:

$$F_t^k(p_{\mathrm{i}}) = F_t^k(gt_{\mathrm{i}} + err_{\mathrm{i}}) = p_{\mathrm{i},k}, \tag{3.11}$$

where F_t represents the point registration process, and $p_{\mathrm{i},k}$ stands for the information in sample p_{i} under view k. Note that the acquired $p_{\mathrm{i},k}$ includes $err_{\mathrm{i},k}$ that is cast to the new view. In this way, we can apply the previous equation in Section 3.3.1 to reduce noise.

Until now, optical flow has been successfully used as a tracking method in many applications. Optical flow assumes that the brightness of a point that moves slightly from frame to frame does not vary as the time varies; the movement of the neighbors follows in the same way. In our method (Figure 3.5), the 68 facial landmarks detected from the first frame are considered the initial points of interest to be tracked by the optical flow, which gives the

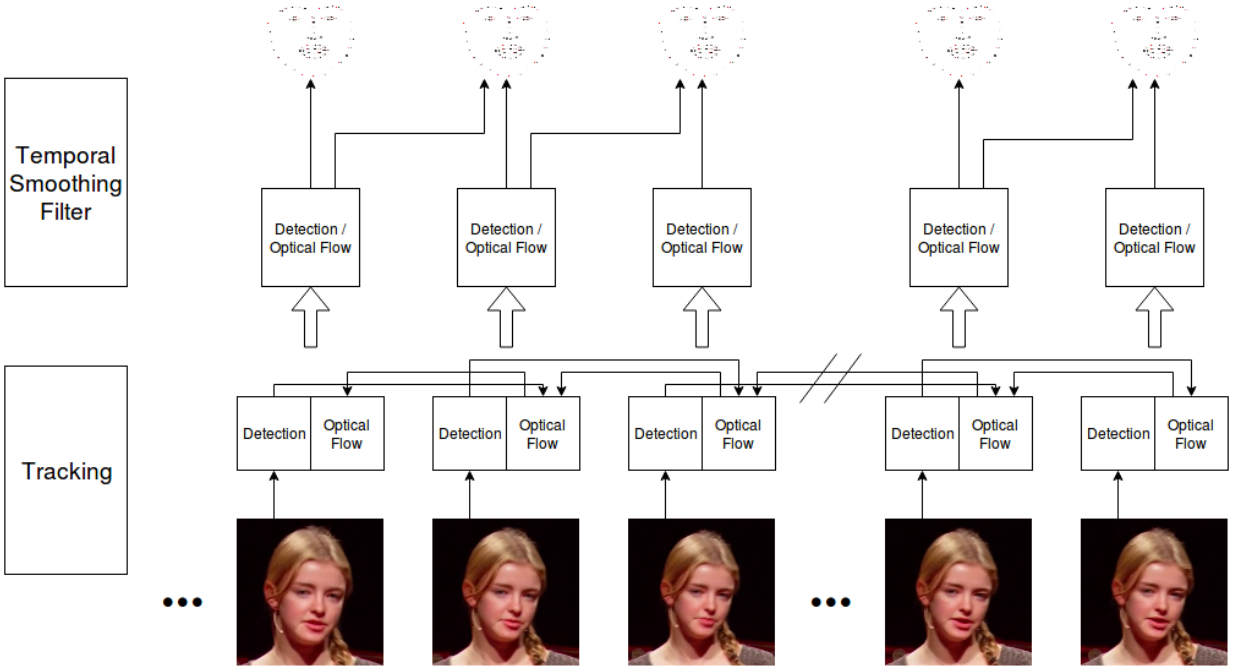


Figure 3.5. Overview of our framework. Each frame of the video sequence is fed into the cascaded detection network to obtain the facial landmarks prediction. Then the optical flow algorithm is applied to the facial landmarks of each frame to predict the landmarks of neighboring frames. Each frame now has two sets of facial landmarks, which are then fused together by assigning different weights to the landmarks of these two sets. For each landmark, if the prediction from the optical flow is close to that from the detection network, a higher weight is assigned to the prediction from optical flow, and vice versa.

predicted positions of these points in the second frame. Then, the optical flow is used again on these predicted points but in a reverse way to predict their positions in the first frame. Finally, the detection model gives another set of 68 facial landmarks in the second frame. Therefore, in both frames, there exist two sets of 68 points, predicted by the detection model and the optical flow. To supervise the correctness of optical flow, the tracking result is chosen over the detection result in the second frame if the tracking result is close to the detection result, and vice versa.

Usually, the optical flow performs well in tracking, but there exist some cases that easily cause the failure of optical flow. For example, the point on the upper eyelid has been tracked stably by optical flow, but when the person blinks their eyes, the point may stick to the lower

eyelid instead of going up with the upper eyelid. Therefore, we assume that the tracking result from optical flow should be close to the detection result. Also, the reason for applying optical flow again to the tracking points in the second frame is to give their positions in the first frame is to check the trustworthiness of the optical flow. This is based on the assumption we made on the optical flow algorithm that the results of a two-frame scenario should be consistent, regardless of the sequence of two frames.

Although applying optical flow to the detection results improves the stability of the facial landmarks, it suffers from the failure case in which the detection result is not close to the tracking result, so it is hard to decide which result should be chosen as the final landmark of the frame. To resolve this issue, we developed a simple but efficient way to smartly leverage the detection result and the tracking result. From a high-level view, the final facial landmarks are given by

$$P_{final} = \alpha P_{detection} + \beta P_{tracking}, \tag{3.12}$$

where α and $\beta = 1-\alpha$ denote the weights assigned to the detection result and tracking result, respectively. To determine the value of α, we made two assumptions: 1) forward projection similarity: if we use optical flow to track the landmarks detected in the first frame to the second frame, the acquired landmarks $F_t^2(p_1)$ should be close to the detected landmarks p_2 in the second frame, and 2) forward-backward projection consistency: if we project $F_t^2(p_1)$ back to the first frame, the obtained $F_t^1(F_t^2(p_1))$ should be close to the original detection results p_1 in the first frame. By measuring these two distances, the value of α can be calculated since the larger the distances, the smaller the value of α. In other words, the less trustworthy the tracking result is.

3.4 Experiment and Analysis

3.4.1 Experimental Setup

Baseline. In [73], Mao *et al.* proposed a cascaded VGG-style network, which demonstrated a strong ability to detect facial landmarks accurately and performed extremely well on the 300-W test dataset [55]. This cascaded network is designed as a two-level network, where the first level outputs initial 68 facial landmarks, and the second level further refines

the prediction results for each component, e.g., eyes, by fusing the global information obtained by the first level network and the features extracted by the second-level network. Compared to the conventional single-level network, this cascaded network outputs facial landmarks of higher accuracy. Figure 3.6 shows an example comparing the results obtained from the first-level network and the cascaded network. In this project, we adopt the first level of Mao's cascaded network as the basic facial landmark detector.

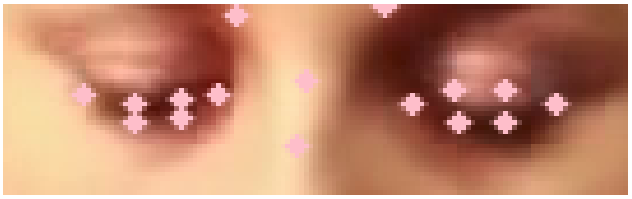


Figure 3.6. Comparison between the results obtained from the single-level network and the cascaded network. The landmarks on the left-side eye are predicted by the cascaded network, while the landmarks on the right-side eye are predicted by the single-level network. This indicates that the cascaded network performs better on the components since it only focuses on them.

Fusion By jointly detecting and tracking the landmarks using optical flow, we are able to reduce the instability of the landmarks that appeared in 300-VW. In each video, the ground truth landmarks can be treated as the detection result. Starting from the second frame, the tracking result is obtained by applying optical flow on the detected landmarks in the previous frame. Then these tracked landmarks in the second frame are used by the optical flow to predict the landmarks back to the first frame. As previously mentioned in Section 3.3.2, the value of α and the weight assigned to each detected landmark can be calculated to obtain the final facial landmarks in the current frame.

Data Augmentation. We design a data augmentation method especially for the video task. The goal of our augmentation is to turn a still image into a "pseudo" video, with continuous changes in the pixel-wise noise, motion blur, brightness, scale, projective distortions, *etc.*. The key point of doing this data augmentation is to acquire a training video without intra-frame noise in landmark locations. To generate diversified videos with these augmentation methods, we borrow the idea of a "storyboard" from designers to assign the

Figure 3.7. Comparison between the image from the original 300-VW dataset and our corrected 300-VW dataset. The annotations of the original 300-VW dataset are not temporally consistent, causing the original dataset to be too noisy to be used. The image on the left side shows one of the examples: the landmarks on the contour are not closely attached to the face contour. However, this is not the case in the image on the right side, which is obtained from our corrected 300-VW dataset.

start status and the end status of the generated video, where the intermediate steps are set to adapt to the fps (frame-per-second) in order to simulate real-world changes.

Figure 3.8. Data augmentation. This sequence of images continuously changes in brightness, Gaussian noise, scale, and projective distortion. In this way we can augment a single image into a "pseudo" video.

3.4.2 Results on Public Datasets

The comparison of our result with previous methods on the three test sets of 300-VW dataset is shown in Table 3.2. Both the base model [73] (as described in Section 3.4.1) and the new model, which is retrained using L2 loss with our proposed method, are compared in this table. The better performance of the new model shows that noise reduction can improve the performance on all three subsets. Compared with the baseline model [73], the noise reduction training provides our network with a better comprehension of the data distribution in hyperspace.

We also report the NME on the three test sets of 300-W for several different methods in Table 3.3. Table 3.3 shows that the best result on 300-W so far is reported by LAB [59] along with a new dataset WFLW [59] including 10,000 images with different environments, poses, occlusions, *etc.*. In addition, SBR [61] and SAN [78] also include private training sets. Our model is able to reach comparable results with only 300-W's training set.

Table 3.2. Comparison of NME on the 300-VW test set. The results of the three subsets are showed in different columns, respectively.

Method	Training Set	Subset-1	Subset-2	Subset-3
Inter-pupil distance				
SDM [74]	/	7.41	6.18	13.04
TCDCN [75]	300-W	7.66	6.77	14.98
CFSS [76]	300-W	7.68	6.42	13.67
DRSN [77]	AFLW, 300-W, CelebA, MAFL, 300-VW	5.33	4.92	8.85
Inter-ocular distance				
Ours-base	300-W	5.13	5.94	8.81
Ours-new	300-W	**4.60**	**4.04**	**8.49**

Table 3.3. Comparison of NME on the 300-W test set.

Method	Training Data	Common	Challenging	Full Set
Inter-pupil distance				
SDM [74]	/	5.57	15.40	7.52
LBF [79]	300-W	4.95	11.98	6.32
MDM [58]	300-W	4.83	10.14	5.88
TCDCN [75]	MAFL	4.90	8.60	5.54
CFSS [76]	300-W	4.73	9.98	5.76
DRA-STR [80]	300-W, AFLW	4.36	7.56	4.99
DRSN [77]	AFLW, 300-W, CelebA, MAFL, 300-VW	4.12	9.68	5.21
3DALBF [81]	300-W, 300W-LP	3.69	10.03	4.93
LAB [59]	WFLW	3.42	6.98	4.12
Inter-ocular distance				
SBR [61]	300-W, AFLW, 300-VW	3.28	7.58	4.10
SAN [78]	300-W, AFLW	3.34	6.60	3.98
LAB [59]	WFLW	**2.98**	**5.19**	**3.49**
Ours	300-W	3.62	5.41	3.97

3.4.3 Effect of Noise on the Accuracy and the Precision

As discussed in Section 3.3.1, we can estimate the mean of target landmark values to any desired degree of accuracy if given a sufficiently large and representative training set [82]. L2 loss can guarantee a high accuracy towards the "true and only" target as long as the noise is zero-mean. In order to verify this, we add different amounts of Gaussian noise (from 0% to 3%) following the settings in [61] to the training set and train another three models using the same approach as the base model. The results in Figure 3.9 show that even if these models have never seen clean data, they are able to reach the same level of accuracy as the base model. Similar results are also discussed in previous papers [61][71].

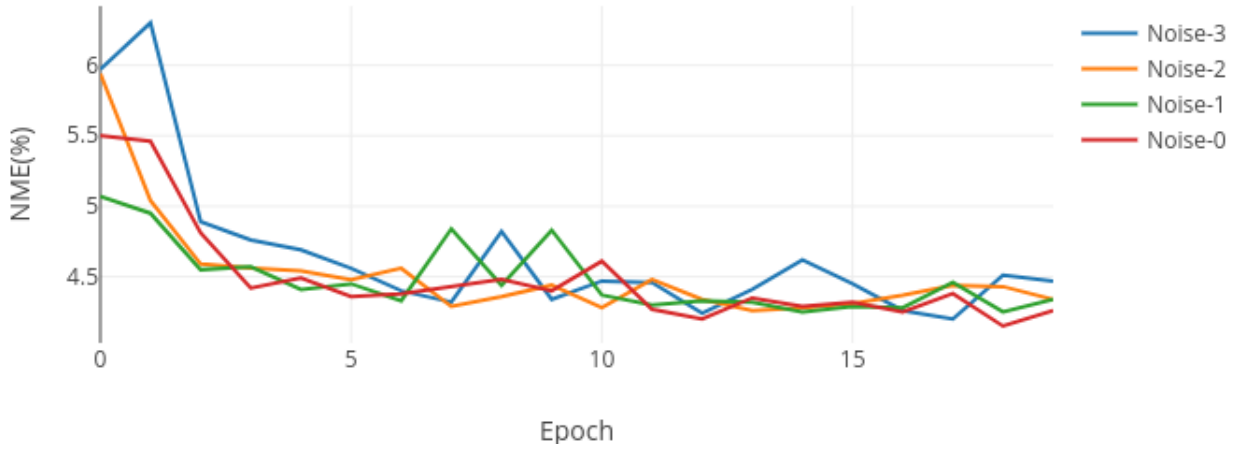


Figure 3.9. NME of models trained with different amounts of injected noise. The X-axis is the epoch of training. Each line represents a series of models acquired by injecting a fixed amount of Gaussian noise to facial landmark locations in the training set. All models are tested on the same 300-W test set. As the epoch increases, all models would converge to a similar level of accuracy regardless of how much injected noise in training data.

When we tried to fine-tune our base model on a small subset of challenging images, we also discovered a possible "blessing" of the noise. Since the challenging images are of a small number and severely deviate from the main set, the fine-tuning can lead to a generally worse result on the 300-VW test set because of over-fitting on the challenging set. We compare the fine-tuning results from original data, augmented data (higher probability of over-fitting),

and data with 3% additive Gaussian noise in Table 3.4. Surprisingly, the model trained with additive noise performs better than the other two. This result may imply the positive effects of known noise in overcoming the trend of over-fitting, which could be the blessing of noise in the facial landmark problem.

Table 3.4. Comparison of NME acquired by fine-tuned models on 300-VW test set.

Model	Subset-1	Subset-2	Subset-3
Original	5.62	4.63	9.97
Aug	5.88	4.76	9.98
Noise	5.55	4.54	9.87

The annotation noise's curse to the detection accuracy has already been discussed in earlier sections. Besides, our experimental results also reveal that the noise in the training set can degrade the model's precision and lead to more jitters in video results. We compare the facial landmark outputs of three models trained with different data: corrected data, noisy data without noise reduction processing, and augmented data. As shown in Figure 3.10, compared with the corrected-data model, the model trained with noisy data is higher in SDD. It indicates that the injected noise would increase the spatial instability of output landmarks. Besides, the augmented-data model shows the best performance among all three models by a large margin, which indicates our 2d-augmentation method can reduce the output jitters without extra modifications to the model.

3.5 Conclusion

In this project, we investigate the effect of noise on the facial landmark detection task by modeling the noise in both the training set and detection output, and comparing models trained with different noise and training strategies. Our results show the great potential of getting a better landmark detector trained on public datasets with our proposed noise reduction method. Our method is capable of handling multiple types of noise in both annotation and detection processes. Besides, we also discuss the relationship between the loss function

Figure 3.10. Comparison of SDD of different models. In this graph, the X-axis denotes the landmark point's index (from 1 to 68), and the Y-axis is the standard deviation of difference (SDD) as defined in Section 3.3.1. The X direction and Y direction results are plotted in the top and bottom graphs, where different color bars are results of three different models: corrected (trained on corrected data), noise (trained on corrected data with injected noise), and aug (trained on augmented corrected data).

and different types of noise. Our further experiments suggest that noise injection could be a good method to avoid over-fitting.

4. SUPER-RESOLUTION ON COMPRESSED IMAGES

4.1 Introduction

Image down-scaling and compression techniques are widely used to meet the limits of hardware storage and data capacity, which sometimes sacrifice the visual effects as well as bringing troubles to visual detection and recognition. Compression artifact reduction (CAR) [83] and single image super-resolution (SISR) [84] have been used in manifold applications, *e.g.* digital zoom on smartphones [85], video streaming [86] and print quality enhancement [68], [87] to restore a high-quality and high-resolution image. With a higher resolution, the target image has more pixels per inch (PPI), resulting in rich information and creating a crisp image. Since Dong [88] first proposed SRCNN that applied a three-layer convolutional neural network (CNN) for the SISR task, more and more works [89]–[91] have explored how to make use of the deep neural networks (DNN) to achieve better image quality as measured by PSNR and SSIM [92], or better visual quality [93], [94] as measured by other perceptual metrics [95], [96].

Conventional methods adopt a two-stage pipeline to leverage the quality and resolution of real-world images: first preprocess the user's photos with a compression artifacts reduction (CAR) algorithm [98]–[102], and then conduct a super-resolution (SR) algorithm [84], [88]–[91], [103]–[106]. However, the CAR step often causes loss of high-frequency information, which results in a lack of detail in reconstructed SR images. Besides, the computation and data transmission between the two models is time-consuming. To deal with these issues, a single-stage method that jointly solves the Compression Artifact Reduction and Super-Resolution (CARSR) problems is needed to reach a balance between reducing artifacts while retaining most details for the upscale step with a short run-time.

Both CAR and SR aim to learn the high-frequency information for reconstruction. Thus, instead of simply concatenating two networks together, we design two functional modules in a single-stage network that reduces the model size by simplifying the two reconstruction processes into one, and can directly obtain high-quality SR output without reconstructing the intermediate artifact-free LR images. Towards this end, we propose a context-aware joint CAR and SR neural network (CAJNN) that can make use of the locally related features in

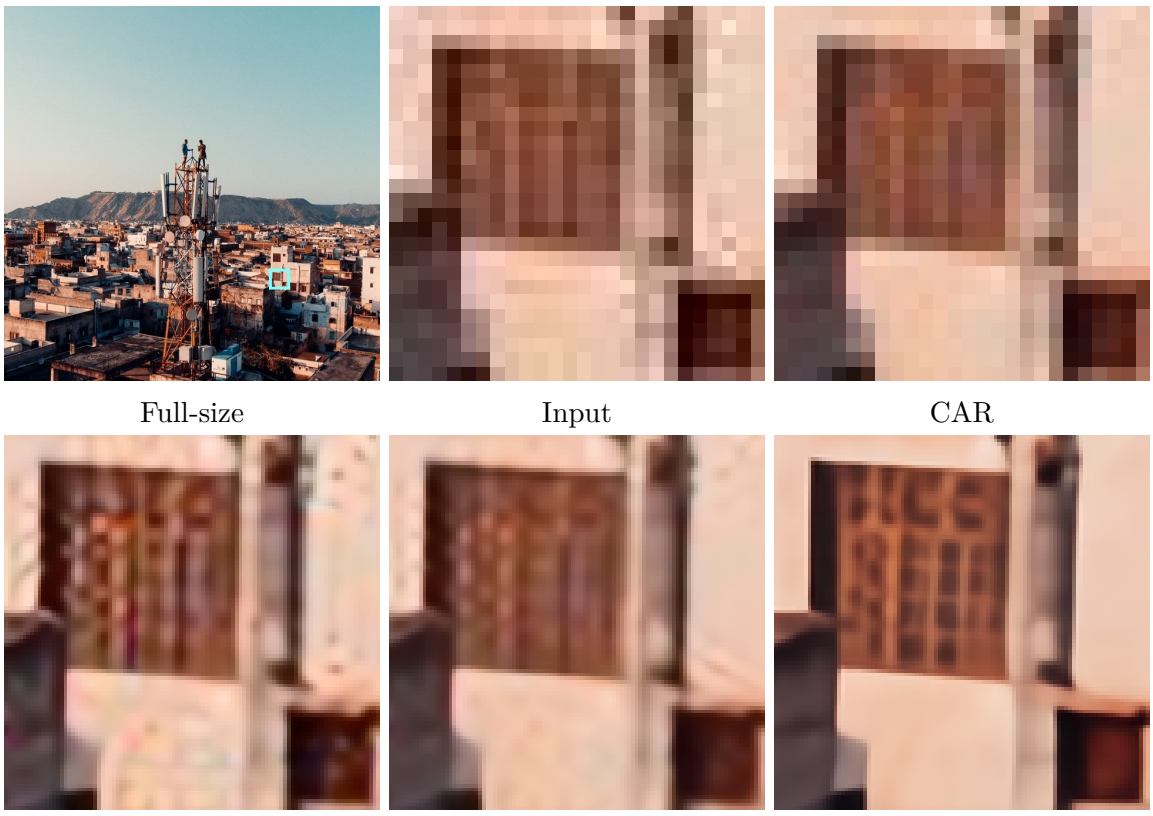


Figure 4.1. Demonstration of the joint CAR and SR task. For a user's image without ground truth, our joint CAR and SR model can generate a more visually appealing output with sharper edges and significantly fewer artifacts compared with either CAR (DnCNN [97]), SR (RCAN [89]), or a two-stage CAR+SR method with above models.

low-quality, low-resolution images to reconstruct high-quality, high-resolution images. To train this network, we construct a paired LR-HR training dataset based on modeling the degradation kernels of web images. Our model turns out to be able to reconstruct high-resolution and artifact-free images with high stability for users' images from a garden variety of web apps (*e.g.* Facebook, Instagram, WeChat). Figure 4.1 illustrates the performance of our proposed algorithm and the benefits of the single-stage joint CARSR method compared with previous two-stage methods: our result can reconstruct a more visually appealing output with accurate structures, sharp edges, and significantly fewer compression artifacts. These output images are not only more recognizable for human viewers, but also for off-the-shelf computer vision algorithms. In this paper, we demonstrate that our proposed CAJNN can enhance the detection and recognition accuracy of high-level vision tasks by reducing the compression artifacts and increasing the resolution of input images.

To summarize, our contributions are mainly three-fold: (1) We propose a novel CAJNN framework that jointly solves the CAR and SR problems for real-world images that are from unknown devices with unknown quality factors. Here, we explore ways to represent and combine both local and non-local information to enforce image reconstruction performance without knowing the input quality factor. (2) Our experiments show that CAJNN achieves the new state-of-the-art performance on multiple datasets *e.g.* Set5 [107], Set14 [108], BSD100 [109], Urban100 [110], *etc.* as measured by the PSNR and SSIM [92] metrics. Compared with the prior art, it generates more stable and reliable outputs for any level of compression quality factors. (3) We provide a new idea for enhancing high-level computer vision tasks like real-scene text recognition and extremely tiny face detection: by preprocessing the input data with our pretrained model, we can improve the performance of existing detectors. Our model demonstrates its effectiveness on the WIDER face dataset [111], and the ICDAR2013 Focused Scene Text dataset [112].

4.2 Related Work

4.2.1 CNN-based Single Image Super Resolution

Convolutional Neural Network (CNN) methods have demonstrated a remarkable capability to recover LR images with known kernels after the pioneering work of Dong *et al.* [88] that adopted a 3-layer CNN to learn an end-to-end mapping from LR images to HR images. The follow-up work FSRCNN [105] established the general structure of most SR networks until today, which conducts most computations in the low-resolution domain and upsamples the image to the required scale at the end of the network. After 2016, more and more works began to explore how to make the network go deeper. EDSR [89] reduces the number of parameters by removing the batch normalization layer, and shares the parameters between the low-scale and high-scale models to achieve better training results. RDN [90], [91] and RRDB [94] employ densely-connected residual groups as the major reconstruction block to reach large depth and to allow sufficient low-frequency information to be bypassed. In the mean time, some useful structures have been introduced to enhance the processing speed or output quality. Shi *et al.* [106] designed a sub-pixel upscaling mechanism. RCAN [103] introduces a channel attention mechanism to rescale channel-wise features adaptively, and SAN [113] exploits a more powerful feature expression with second-order channel attention.

4.2.2 Compression Artifact Reduction

Lossy compression methods [114], [115] are widely applied in web image transmission due to their higher compression rates. Traditional methods for the CAR problem generally fall into two categories: unsupervised methods, which include removing noise and increasing sharpness [100], and supervised methods like dictionary-based algorithms [116]. After the success of SRCNN on the super-resolution task, Yu *et al.* [117] directly applied its architecture to compression artifacts suppression. Similar to the development of SR, CNN-based CAR networks also go deeper with the introduction of residual blocks and skip connections [97], [118], [119]. Besides, SSIM loss is employed [99] as a supervision method to obtain better performance than MSE loss. JPEG-related priors are also considered in the

network structure design, *e.g.* DDCN [120] adds a Discrete Cosine Transform (DCT)-domain before the dual networks, and the D^3 [121] takes a further step in the practice of dual-domain approaches [116] by converting sparse-encoding approaches into a one-step sparse inference module.

Unlike the above approaches that require reconstructing the intermediate clean LR images, our joint CARSR framework directly obtains artifact-free HR images without prior information of quality factors or explicit CAR supervision in the LR domain.

4.3 Joint Compression Artifacts Reduction and Super-Resolution

Given an LR JPEG image I^{LRLQ}, our goal is to reconstruct the high resolution, high-quality image $G(I^{LRLQ})$ that approaches the high-resolution, high-quality ground truth I^{HRHQ} with a generator G. The CARSR task can be expressed as:

$$\arg\min_{\theta} l(I^{HRHQ}, G(I^{LRLQ}, \theta_g)), \tag{4.1}$$

where l is any designated loss function (*e.g.* MSE, L1, Charbonnier, *etc.*). G is the function representing our deep neural network with parameters θ, for which we wish $G_\theta \approx F^{-1}((I^{LRLQ} \otimes k) \uparrow_s, q)$, where $\otimes$ stands for the convolution operation, k is the degradation kernel of downsampling method, and s is the downscaling factor. To effectively handle the CARSR task, we propose a single-stage framework, CAJNN. Our proposed model is end-to-end trainable with I^{HRHQ} and I^{LRLQ} pairs according to the function above.

The CAJNN framework mainly consists of three modules (see Figure 4.2): the *context-aware feature extractor*, the *reconstruction module*, and the *upsampling and enhancement module*. The context-aware feature extractor captures and assembles both intra- and inter-block information from different receptive fields. The reconstruction module further refines the extracted feature maps. Finally, after the processing of the upsampling and enhancement module, these feature maps are converted to high-resolution outputs.

4.3.1 Model

Here we discuss the CAJNN structure in detail. The majority of our network operates in the feature domain. Given I^{LRLQ} ($c \cdot h \cdot w$ in size), a feature extraction layer first turns the image into feature maps ($n_f \cdot h \cdot w$ in size, where n_f denotes the number of feature channels) in the domain for the following process. The feature map will be converted to a high-resolution image ($c \cdot H \cdot W$ in size) after passing the upsampling and enhancement module. To achieve a balance between GPU capacity and output quality, we apply $n_f = 64$ channels to ensure enough information for the reconstruction. We adopt a 3×3 convolution layer that serves as the initial feature extractor. After this module, the input image I^{LRLQ} is turned into a $64 \cdot h \cdot w$ tensor $f^{L'}$.

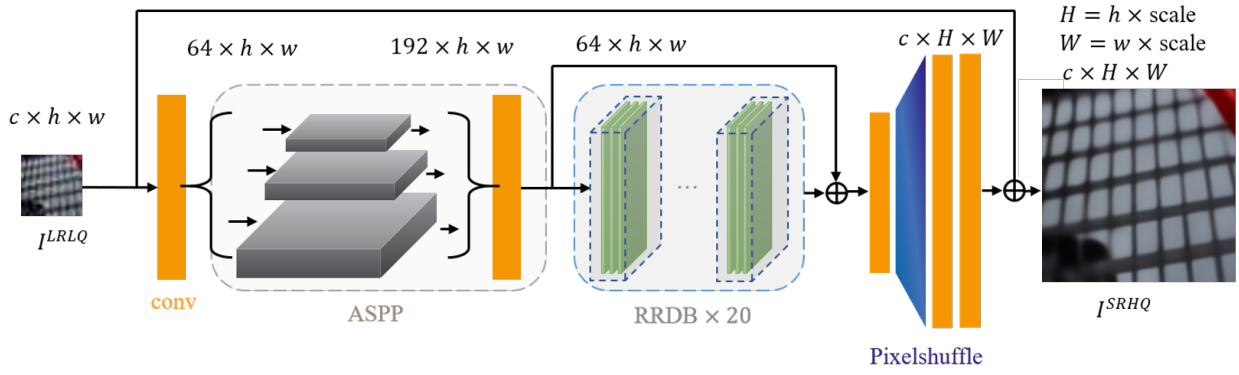


Figure 4.2. The network architecture of our proposed CAJNN. It directly reconstructs artifact-free HR images from the LR low-quality images I^{LRLQ}. Atrous Spatial Pyramid Pooling (ASPP) is adopted to utilize the inter-block features and intra-block contexts for the joint CARSR task. The reconstruction module turns the features into a deep feature map, which is converted to a high-quality SR output I^{SRHQ} by the upsampling and enhancement module.

Context-aware Feature Extractor

The pipeline of JPEG compression involves the following steps: color space transformation (*e.g.* JPEG, H.264/AVC, H.265/HEVC), downsampling, block splitting, discrete cosine transform (DCT), quantization, and entropy encoding. Some previous research assumed that the quality factors of input images are known, and the original images are well-aligned

by 8×8 patches with respect to the JPEG block boundaries. However, real-world inputs do not always meet such assumptions. In the worst case, the input images might be compressed multiple times and contain sub-blocks or larger blocks, which requires the model to be insensitive, or even blind to the encoding block alignment. Thus, the spatial context information of both intra- and inter- JPEG blocks is essential for designing a CARSR network.

We adopt the ASPP module to extract and integrate multi-scale features with an atrous spatial pooling pyramid (ASPP) [122]. We adjust the dilation rates of each layer in the pyramid to extend the filter's perceptive field for extracting different ranges of context information, in which the largest field-of-view should cover the 8×8 block. Besides, we should avoid sampling overlap in different levels of the 3×3 convolutions. Considering the factors above, we choose 1, 3, 4 as the dilation groups to find a good balance between accurately retrieving local details and assimilating context information between adjacent blocks. The input tensors are sent to 3 layers of the pyramid: a 3×3 convolutional layer with dilation rate $= 1$, a 3×3 convolutional layer with dilation rate $= 3$, and a 3×3 convolutional layer with dilation rate $= 4$. The outputs of these three layers are concatenated and aggregated by a 1×1 convolution that converts the $192-$dimension feature into a 64-dimension one. The process in ASPP can be described by:

$$f^{L'} = [C_{3\times3,1} \otimes f^{L} | C_{3\times3,3} \otimes f^{L} | C_{3\times3,4} \otimes f^{L}] \otimes C_{1\times1,1}, \tag{4.2}$$

where $f^{L'}$ denotes the output feature ($64 \cdot h \cdot w$ in size), $C_{a\times a,r}$ represents the parameters of $a \times a$ convolution with dilation rate r, and $|$ is a concatenation operation.

Reconstruction

RRDB (residual-in-residual dense block) [94] is applied as the basic block for the reconstruction trunk. Compared with residual blocks, it densely connects the convolution layers to local groups while removing the batch normalization layer. In our network, the reconstruction module includes 20 RRDBs.

Upsampling and Enhancement

After the reconstruction module, the image feature is preprocessed by a 3×3 convolution layer before the PixelShuffle layer [106] for upsampling. The Pixelshuffle layer produces an HR image from LR feature maps directly with one upscaling filter for each feature map. Compared with the upconvolution, the PixelShuffle layer is $\log_2 s^2$ times faster in theory because of applying sub-pixel activation to convert most of the computations from the HR to the LR domain. The feature $f^{L''}$ is turned into a $c \cdot sh \cdot sw$ HR image by the PixelShuffle layer, which can be described by:

$$I^{SR} = PS(W_L \otimes f^{L''} + b_L), \tag{4.3}$$

where W_L denotes the convolution weights and b_L the bias in the LR domain, PS is a periodic shuffling operator for re-arranging the input LR feature tensor $f^{L''}$ $(c \cdot s^2 \cdot h \cdot w)$ to an HR tensor of shape $c \cdot sh \cdot sw$:

$$PS(T)_{x,y,c} = T_{\lfloor x/s \rfloor, \lfloor y/s \rfloor, c \cdot s \cdot \mathrm{mod}(y,s) + c \cdot s \cdot \mathrm{mod}(x,s)}. \tag{4.4}$$

Instead of directly outputting the high-resolution image, we process it through two 3×3 convolution layers for further enhancement, and get $I^{SR} = C_{3\times3,1} \otimes (C_{3\times3,1} \otimes I^{SR})$.

To make the major network focus on learning the high-frequency information in the input image, we bilinearly upsample the input LR image I^{LRLQ} and add it to form the final output $G(I^{LRLQ}, \theta_g)$:

$$G(I^{LRLQ}, \theta_g) = I^{LRLQ} \uparrow_s + I^{SR}. \tag{4.5}$$

This long-range skip connection changes the target of our major network from directly reconstructing a high-resolution image to reconstructing its residual. By letting the low-frequency information of the input bypass the major network, it lowers the difficulty of reconstruction and increases the convergence speed of the network.

4.4 Experiments and Analysis

4.4.1 Experimental Setup

Training Dataset

In this paper, we choose the DIV2K dataset [123] (800 RGB images of 2k resolution) and Flick2K dataset [124] (2650 RGB images of 2k resolution) as our training set. Here, "resolution" refers to the number of pixels, not the spatial pixel dimensions in the displayed image. To get the training pairs that approach the degradation kernels of web images, we first model the downsampling and compression types of popular web platforms. We also discover that adding severely compressed samples to the training set can improve the output quality in terms of PSNR, even for input images compressed with high-quality factors. Based on these pre-experimental results, the images of the training set are downsampled with a scaling factor $s = 4$ and compressed by MATLAB [125] with random quality factors from 10 to 100. Besides, we perform data augmentation on these images by randomly cropping, randomly rotating by 90°, 180°, and 270°, and randomly horizontal-flipping. As a result, each cropped image patch can have eight augmentation types at maximum.

Test Datasets

We compare the performance of our model and previous methods on Set5 [107], Set14 [108], BSD100 [109], Urban100 [110] and Manga109 [126]. Each image is downscaled $\times 4$ and compressed with quality factors of 10, 20, and 40 to be consistent with previous works.

Implementation Details

Our network is trained on one Nvidia Titan Xp graphics card. The batch size is 36, and the patch size is 128 for ground truth and 32 for the low-resolution input. We use Adam [127] as the optimizer with a cosine annealing learning rate, in which the initial learning rate is $2e-4$, and the minimum learning rate is $1e-7$. The scheduler restarts every 2.5e5 iterations. The network is trained for 1e6 iterations in total.

4.4.2 Results for Image Quality Assessment

Comparison with SOTA on Standard Test Sets

We compare the performance of CAJNN to the previous state-of-the-art (SOTA) methods on the standard test sets as mentioned above. We report the PSNR and SSIM [92] of the Y channels in the test sets to be consistent with previous works. We also show the number of parameters and the inference time on Set5 in Table 4.1. Depending on the workflow for solving the CAR and SR problem, these methods can be categorized into the following three types: (1) *SR:* directly use pretrained SR models. (2) *CAR+SR:* the aforementioned two-stage method, which first removes the compression artifacts and then sends the output images to the SR model. (3) *Joint CAR & SR:* the single-stage method that jointly handles CAR and SR with one model. We report both the direct output and the self-ensembled [90] output of our network.

According to Table 4.1, CAJNN significantly outperforms the existing methods for all QFs and yields the highest overall PSNR across all five datasets with Set5. The improvement is consistently observed on SSIM, as well. Moreover, our model is more light-weight than most of the current models, including one-stage and two-stage in summation, which results in faster inference speed on the same hardware (all the tests are conducted on one Nvidia Titan Xp graphics card).

Figure 4.3 gives a qualitative example of the results of our model, where the input image is *woman* from the Set5 [107] that is downsampled and compressed by a wide range of quality factors from 10 to 100. It is worth noting that compression with very low quality factors causes a significant color shift to the hue and spatial distribution of the original image, which can be seen in the leftmost LR image (QF = 10). Our model is able to correctly restore the color aberrations of RGB images with a high consistency among different QFs.

Results on User Images

Besides the above experiments on standard test images, we also conduct experiments on real user images to demonstrate the effectiveness of our model. We mainly focus on the

Table 4.1. Quantitative comparison of applying SOTA SR methods, two-stage SR and CAR methods, and our CAJNN for three different quality factors. The best two results are highlighted in red and blue colors, respectively. Our method greatly outperforms all two-stage methods in terms of PSNR and SSIM, while having a relatively small model size and shorter runtime. The runtime (inference only) is measured on the entire Set5.

QF	Method	Network	Runtime (s)	Parameters (Million)	Set5		Set14		BSD100		Urban100		Manga109	
					PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
10	SR	Bicubic	-	-	23.99	0.6329	22.94	0.5513	23.33	0.5303	20.95	0.5182	21.94	0.6383
		EDSR	1.94	43.1	23.41	0.6019	22.48	0.5272	22.96	0.5098	20.57	0.5006	21.53	0.6151
		RCAN	2.04	16	23.14	0.5733	22.29	0.5064	22.78	0.4984	20.36	0.4819	21.21	0.5878
		RRDB	0.65	16.7	22.43	0.5223	22.86	0.5051	20.43	0.4940	20.43	0.4940	21.34	0.6075
	CAR+SR	ARCNN+RRDB	3.20+0.65	0.56+16.7	24.21	0.6699	23.38	0.5774	23.63	0.5474	21.28	0.5466	22.36	0.6856
		DnCNN+RRDB	0.38+0.65	0.06+16.7	24.07	0.6434	23.13	0.5582	23.37	0.5324	21.04	0.5305	22.10	0.6532
	Joint CAR&SR	CAJNN (ours)	0.48	14.8	25.04	0.7169	23.95	0.6028	23.84	0.5598	21.97	0.5977	23.29	0.7333
		CAJNN (ours, self-ensembled)	2.50	14.8	25.14	0.7202	24.03	0.6052	23.88	0.5610	22.18	0.6051	23.44	0.7377
20	SR	Bicubic	-	-	25.32	0.6761	23.85	0.5870	24.14	0.5611	21.66	0.5526	22.84	0.6724
		EDSR	1.94	43.1	24.76	0.6490	23.59	0.5707	23.88	0.5482	21.38	0.5427	22.58	0.6549
		RCAN	2.04	16	24.44	0.6226	23.40	0.5502	23.65	0.5351	21.12	0.5234	22.14	0.6253
		RRDB	0.65	16.7	24.65	0.6450	23.57	0.5661	23.79	0.5442	21.25	0.5365	22.38	0.6474
	CAR+SR	ARCNN+RRDB	3.20+0.65	0.56+16.7	25.40	0.7082	24.30	0.6091	24.39	0.5755	22.02	0.5811	23.52	0.7172
		DnCNN+RRDB	0.38+0.65	0.06+16.7	25.55	0.6946	24.24	0.6001	24.28	0.5679	21.90	0.5732	23.24	0.6961
	Joint CAR&SR	CAJNN (ours)	0.48	14.8	26.59	0.7604	25.03	0.6391	24.70	0.5924	23.06	0.6482	24.81	0.7783
		CAJNN (ours, self-ensembled)	2.50	14.8	26.65	0.7633	25.10	0.6404	24.74	0.5936	23.28	0.6550	24.98	0.7820
40	SR	Bicubic	-	-	26.38	0.7154	24.55	0.6201	24.77	0.5898	22.26	0.5877	23.66	0.7081
		EDSR	1.94	43.1	26.01	0.6972	24.48	0.6120	24.62	0.5836	22.18	0.5893	23.73	0.7003
		RCAN	2.04	16	25.70	0.6726	24.30	0.5936	24.36	0.5704	21.86	0.5690	23.13	0.6673
		RRDB	0.65	16.7	25.99	0.6958	24.50	0.6079	24.54	0.5804	22.10	0.5851	23.50	0.6918
	CAR+SR	ARCNN+RRDB	3.20+0.65	0.56+16.7	26.65	0.7495	25.16	0.6424	25.06	0.6053	22.82	0.6235	24.68	0.7578
		DnCNN+RRDB	0.38+0.65	0.06+16.7	26.87	0.7403	25.15	0.6373	25.00	0.5995	22.78	0.6194	24.42	0.7404
	Joint CAR&SR	CAJNN (ours)	0.48	14.8	28.05	0.7981	25.96	0.6729	25.43	0.6240	24.09	0.6962	26.25	0.8177
		CAJNN (ours, self-ensembled)	2.50	14.8	28.16	0.7993	26.03	0.6742	25.46	0.6251	24.31	0.7011	26.44	0.8211

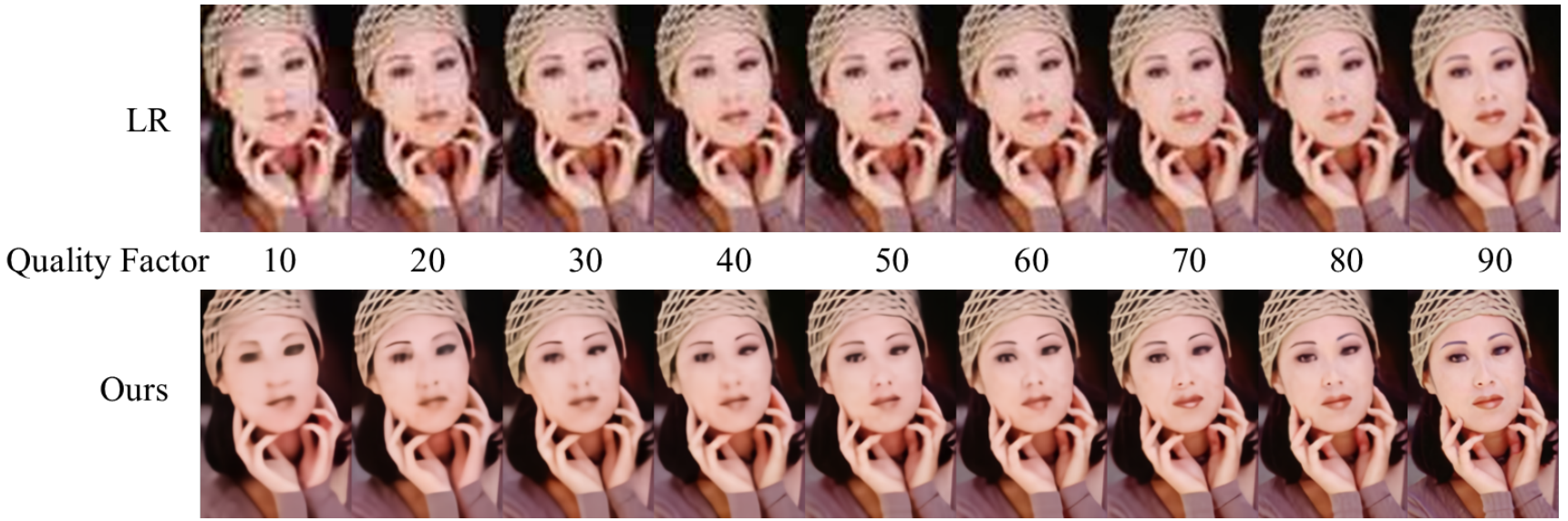


Figure 4.3. The qualitative result of our network from compressed images with different quality factors (zoom in for a better view). Our model is able to reconstruct reasonable SR images, even at extremely low quality factors. Besides, our results are free of color jittering and other inconsistencies for such a wide range of compression ratios. The image is the "woman" image from Set5 [107].

perceptual effect since there are no ground-truth images. Figure 4.4 shows the CAJNN results on real-world image from the WIDER face dataset [111]. For comparison, RCAN [103] and RRDB [94] are used as representative SR method, ARCNN [98] and DnCNN [97] are used as the representative CAR methods. The real-world images have unknown downsampling kernels and compression mechanisms, depending on the platforms. According to Figure 4.4, the SR methods generate images with obvious color shift and ringing artifacts. These artifacts are alleviated with two-stage methods. Still, the results are blurry. Compared with the two-stage methods, our CAJNN can provide SR outputs with sharp edges and rich details, which demonstrates the superiority of our proposed single-stage method when applied to real-world CARSR problems.

4.4.3 Results for Low-Resolution Text Recognition

Comparing the input LR image and our output in Figure 4.4, the texts become more readable after being processed by our model. Inspired by this observation, we conducted the following experiments to explore our model's potential to leverage a real-scene text recognition task from low-resolution characters.

We compare the total accuracy of generic text detection on the ICDAR2013 Focused Scene Text dataset [112] with TPS-ResNet-BiLSTM-Attn [128] as the text recognition method. The baseline result is acquired by directly recognizing the original input images. As a comparison with the baseline, we use the CAJNN model as described in previous sections to generate artifact-free SR images from the original images and the down-sampled images and conduct recognition on the outputs.

As can be seen in Table 4.2, the preprocessing of CAJNN improves the recognition accuracy from 85.30% to 85.75%, which indicates that the outputs of our model are not only visually appealing to human viewers, but also include more distinct information for the text recognition network as shown in Figure 4.5. It is worth noting that our output image is 4× the size compared with the baseline inputs, and the average detection time is increased from 31.22s to 41.56s. Although the improvement in accuracy demonstrates the positive effect yielded by our model, the rise in computation is hard to ignore. Therefore, we disentangle the

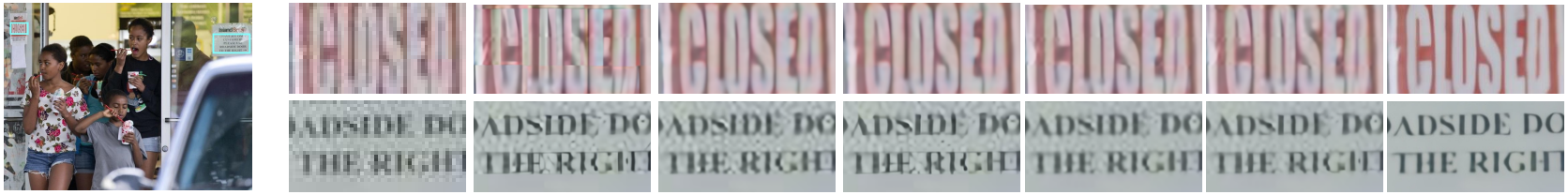


Figure 4.4. CAR & SR performance comparison of different methods on a user's image from the WIDER face dataset [111]. Compared with previous methods, our model can generate artifact-free high-resolution images with sharp edges.

influence of SR and CAR by bicubicly downsampling the CARSR output images and acquire the third recognition result. Since the image size remains the same as that of the original image, the detection time is identical to the baseline. Compared with the downsampled image, our method greatly improves the accuracy by 14.34%. Even compared with the baseline, the recognition accuracy still improves 0.27% due to the reduction of compression artifacts, which indicates that our model is capable of extracting and maintaining critical features of input images. This experimental result points out a plausible direction for future text recognition research: image quality plays a vital role in the recognition accuracy, which can be improved by utilizing the learned priors from a pretrained CARSR model.

Table 4.2. Text recognition accuracy on the ICDAR 2013 Focused Scene Text dataset [112]. Compared with the baseline method, the introduction of our CARSR method improves the detection performance by 0.45% (without downsampling). For the downsampled one, our method improves by 14.34% (with downsampling).

Method	Accuracy	Detection Time (s)
Baseline [128]	85.30%	31.22
Ours + Baseline [128]	**85.75**%	41.56
Downsample + Baseline [128]	71.23%	2.65
Ours + Downsample + Baseline [128]	85.57%	31.22

4.4.4 Results for Extremely Tiny Face Detection

Extremely tiny face detection is another practical, yet challenging task in high-level computer vision. Most of the state-of-the-art (SOTA) face detectors [129], [130] for in-the-wild images have already taken various scales and distortions into consideration to achieve impressive detection performance. [131] proposed a solution to tackle tiny face detection by explicitly restoring an HR face from a small blurry one using a Generative Adversarial Network (GAN) [132].

We experimentally validate the effect achieved by our CAJNN on tiny face images in the WIDER FACE dataset [111] by comparing the detection results from the following three types of data: original HR (serves as the baseline), 4× downsampled LR (serves as the

GT Ours (×4) Ours (downsampled)

Figure 4.5. Test samples of ICDAR2013 dataset (*word 161*, *word 836*). The first column shows original input images, the second column is the CARSR output generated by our method, and the third column is acquired by downsampling the second column. By comparing the detection results in the first and second columns, our method can serve as a supportive method for the recognition of low-resolution texts. Besides, the artifact-free image in the third column can also provide more recognizable features for the baseline model without increasing the image size.

extremely tiny face inputs) and CARSR outputs from our model. [133] is applied as the backbone face detector (We use an unofficial PyTorch [134] implementation provided by https://github.com/varunagrawal/tiny-faces-pytorch).

Table 4.3 shows the Average Precision (AP) of the downsampled tiny images and our enhanced ones on all the three validation sets (easy, medium, and hard) of WIDER FACE [111]. From Table 4.3, we observe that the data processed by CAJNN dramatically improves the detection of LR inputs from 0.317 to 0.611 in AP on the hard set. The reason is that the baseline detector performs downsampling operations by large strides on the tiny faces. Considering the fact that the tiny faces themselves contain less information than average, the detailed information of face structure is lost after several downsampling convolutions. In contrast, our CAJNN provides an artifact-free SR image, which can boost the detection performance by better utilizing the information of small faces. In Figure 4.6, the precision-recall curve of our reconstructed image (green line) is close to the ground truth (red line) on the easy and medium subsets. In the hard subset, our CAJNN yields a significant improvement compared to the LR curve. The gap between our output and the GT is due to the irreversible loss of information in extremely tiny faces that happens more frequently in the hard set during the downsampling process.

Table 4.3. Average precision of three data types in the WIDER FACE validation set [111] with the same face detector [133]. We downsample the original images by scale factor= 4 to acquire the LR images. The application of our CARSR method greatly improves the detection performance with LR images on all three subsets.

Input Data	Easy	Medium	Hard
GT	0.900	0.887	0.792
LR	0.824	0.692	0.317
LR + Ours	0.893	0.857	0.611

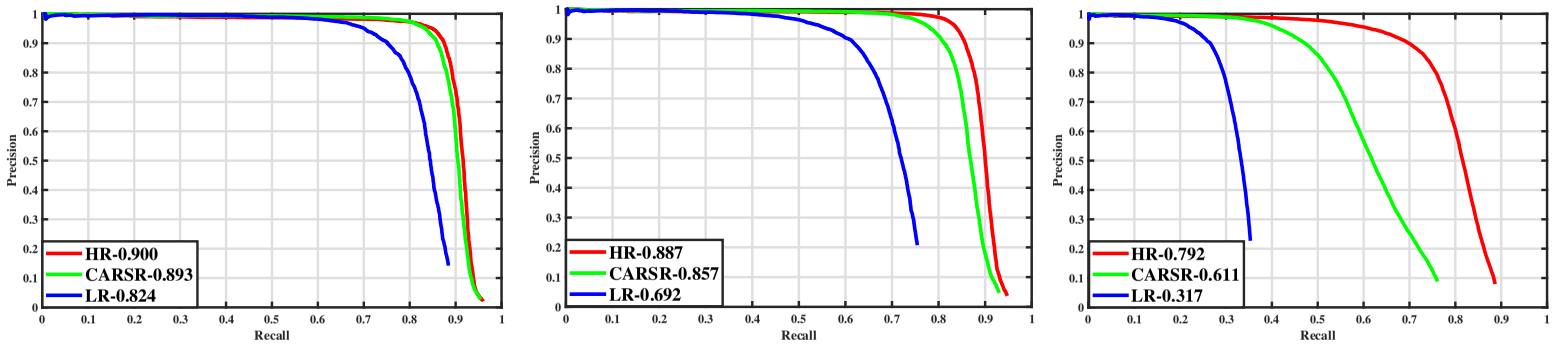


Figure 4.6. The precision-recall curve of three subsets in WIDER FACE [111]. The AOC (area under curve) reflects the detector's performance on each type of data (GT, LR that is bicubicly downsampled from the GT, and CARSR that takes the LR images back to the original resolution). With preprocessing by our model, the detection performance of tiny images can be improved close to that achieved with GT. (Zoom in for a better view.)

4.4.5 Ablation Study

Effect of Multi-scale Information

As discussed in previous sections, both intra- and inter-block context information is important for designing a CARSR network. In other low-level vision tasks, context information at different scales has already been proved to be effective in improving network performance. Inspired by the first convolution layer of the ResNet [16], previous researchers [135] applied 7×7 convolution to extract the context features for the video frame interpolation task. However, such big kernels bring a tremendous number of parameters to the network, especially when embedded in the feature domain, resulting in higher computational costs. Another way of enlarging the filter's receptive field is to use a non-local module [136], [137], where the input images are downsampled by convolutional strides and processed at different scales. The non-local module has a rather complex structure and also a large number of parameters. In order to use the context information in a much simpler and lighter representation, our method adopts atrous convolution. By adjusting the dilation rate r, the filter can incorporate the context information from a larger receptive field without dramatically increasing the number of parameters compared to the above methods.

Table 4.4. Ablation Study on the validation set (Set5). We report the performance of CAJNN without the long-range skip connection and ASPP as the baseline. Rows 1-3 show the influence of different ways to extract contextual information by replacing ASPP with other network structures. Rows 4-5 compare the effect of two different upsampling methods on PSNR. The combination of the ASPP and Pixelshuffle modules yields the best performance, and thus is adopted in our network architecture.

Model	Base	1	2	3	4
Non-local module		√			
ASPP			√	√	
Seqeuntial atrous pooling					√
Upconvolution	√	√	√		
Pixelshuffle				√	√
PSNR (dB)	27.868	28.274	28.276	**28.292**	28.262

We conduct an ablation study to illustrate the effect of different ways of representing contextual information in Table 4.4. In Rows 1-3, we compare the performance of the non-local module, ASPP, and sequential atrous pooling. Comparing the base model to Column 1 in Table 4.4, we can conclude that the introduction of multi-scale information via a non-local module can significantly improve the PSNR by 0.406 dB. This result validates the superiority of aggregating both intra- and inter-block features rather than using a purely local representation for the CARSR task. Furthermore, as seen by comparing Columns 1 and 2, replacing the non-local module with our well-designed ASPP can improve the PSNR by 0.002 dB. Although the improvement is rather small, it is worth noting that the ASPP has fewer convolution layers and parameters, which results in smaller model size and fewer FLOPs. Remarkably, it can achieve results that are comparable, or even better than those yielded by models with more parameters. By comparing Columns 3 and 4, we also note that the PSNR of ASPP is higher than that of sequential atrous pooling by 0.03 dB, which means that the pyramid-fusion structure is more efficient in representing the multi-scale information. Finally, by comparing Columns 2 and 3 of Table 4.4, we can observe that the PixelShuffle layer brings a 0.16 dB improvement to PSNR.

End-to-End Supervision by Joint CAR and SR

Another ablation study on supervising the CARSR task is conducted to illustrate the effect of joint end-to-end training. Instead of supervising with I^{HRHQ}, we attempt to disentangle the CAR and SR by introducing a reconstruction loss according to the definition in Equation 4.6, where we can generate an artifact-free LR image I^{LRHQ} from the ground truth I^{HRHQ}:

$$I^{LRHQ} = (k \otimes I^{HRHQ}) \downarrow_s, \tag{4.6}$$

and use it to explicitly supervise the intermediate CAR output $\hat{G}(f^{L'})$ after the context-aware module:

$$l^{LR} = l(I^{LQHQ}, \hat{G}(f^{L'})). \tag{4.7}$$

Denoting the pixel-wise loss of the final output and ground truth (shown in Equation 4.1) as l_{HR}, the overall training loss becomes:

$$l = l^{HR} + \lambda l^{LR}, \tag{4.8}$$

by increasing the weight λ, we can acquire models trained with higher disentanglement levels. We train three models with $\lambda = 0, 1, 16$ while keeping all the other factors the same. The performance of these models on our validation set is shown in Table 4.5. The trend is obvious: the PSNR increases as the entanglement increases, which demonstrates the effectiveness of the joint CARSR method with a single-stage network.

Table 4.5. Ablation study on joint end-to-end supervision. We introduce the explicit reconstruction loss as a disentanglement mechanism of CAR and SR. By changing the weight of this loss term, we can study the effect of different levels of joint-supervision. Among all the settings, the model trained without the reconstruction loss performs best on our validation set.

Model	a	b	c
Weight of reconstruction loss λ	16	1	0
PSNR (dB)	27.507	27.627	**27.672**

4.5 Conclusion

In this paper, we propose a single-stage network for the joint CARSR task to directly reconstruct an artifact-free high-resolution image from a compressed low-resolution input. To address the CARSR problem, we make use of the contextual information by introducing a specially designed ASPP that integrates both intra- and inter-block features. Our experiments illustrate the effectiveness and efficiency of our method with both standard test images and real-world images. Moreover, the extensive experimental results reveal a high potential for enhancing the performance of current methods for various high-level computer vision tasks, *e.g.* real-scene resolution text recognition, and extremely tiny face detection.

5. HEADSHOT IMAGE SUPER-RESOLUTION WITH MULTIPLE EXEMPLARS

5.1 Introduction

It has been shown by numerous psychological and cognitive studies that face perception is one of the most important and specialized aspects of social cognition [138], [139]. The facial regions of a picture tend to draw the attention and interest of observers immediately. Moreover, humans are susceptible to minor changes in familiar faces [140]. Thus, increasing the quality of the face region in images and videos has the potential to significantly enhance the user experience of many social communication applications, *e.g.* real-time video chat, mobile photo booth, *etc.*.

For the above reasons, the machine learning community has widely explored face hallucination [141]–[146] as a domain-specific problem of single image super-resolution(SISR) [84], [104], [147], which aims to restore realistic details from a low-resolution (LR) face image to a high-resolution (HR) one. Benefiting from the integration of face structure and identity priors and recent progress in deep neural network designs, it is now possible to generate visually pleasing results even for extremely tiny faces. When the input LR headshot does not contain enough attribute or identity information, using additional references can help

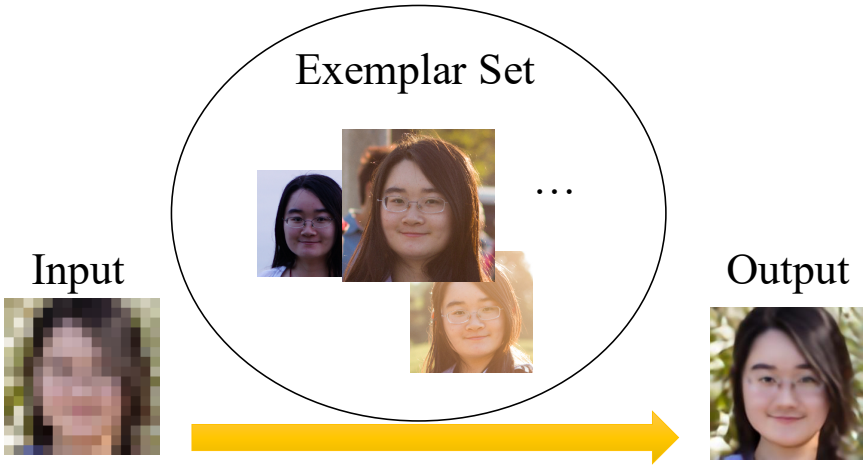


Figure 5.1. Headshot super-resolution that recovers the lost information in the input using a set of exemplars, i.e. reference images.

to achieve a more faithful reconstruction result. In this paper, we explore a novel method that makes full use of an arbitrarily-sized set of exemplar images to increase the fidelity of headshot image super-resolution.

In order to extract the correct information for reconstruction, we need to search the matched regions and transfer the corresponding features to the output. Previous methods choose to conduct the global context matching with registration [148], optical flow [149]–[152] combined with a warping process [153]. Still, these works assume that the exemplars share a similar viewpoint with the LR input [153], which cannot always be guaranteed. Besides, their performance depends on accurate motion estimation and may poorly capture long-range correlations. Other methods [154]–[157] solve this problem by conducting an exhaustive patch-wise comparison of extracted features from the LR content and the references, which require a large amount of computation, especially when the reference resolution is high[1]. In addition, these methods cannot handle inter-patch misalignment or non-rigid deformations. To better utilize the information of faces from different poses or views, we propose a Reference Feature Alignment module (RFA) to find the corresponding information in reference features and align them with the LR content. The acquired set of aligned reference feature maps can be fed to the following modules for feature transferring and reconstruction.

In practical applications like smart home cameras or mobile photography, it is possible to acquire many high-resolution images of different views when the user is close to the camera. These images can naturally serve as good exemplars to enhance far-away tiny faces. However, most previous works focus on reference-based super-resolution (RefSR) with one exemplar [151]–[153], [155], [156], which is a simplified assumption. To handle a set of exemplars, these methods require an extra step to select the most similar image as the reference according to SIFT [158], [159] or facial landmarks points [160], which is a poor representation of the whole set. [161] devises a framework to process and combine multi-exemplars with a weighted pixel average. Still, it is not robust to the displacement or distortions in reference images as is our method. To utilize the reference set effectively and efficiently, we propose a Content-conditioned Feature Aggregation module (CoFA) that

[1]↑Note that throughout this chapter, we use the terms “exemplars” and “references” to refer to the same thing.

simplifies the set-to-image RefSR problem to a point-to-point RefSR by aggregating feature maps in a set into a single representation.

Benefiting from the module designs above, our network is end-to-end trainable without requiring other face-specific meta-information. Aiming to generate an SR output with highly-detailed textures, we propose a novel correlation loss inspired by the correlation layer in FlowNet2 [162], [163] to supervise the reconstruction of texture patterns. We compute the pixel-wise correlation across the channel dimensions to represent the local textures within a certain window size.

In summary, our contribution is four-fold:

(1) we propose a novel headshot super-resolution network that takes advantage of multiple exemplars. Our method is more effective than previous approaches by thoroughly integrating the corresponding information in the exemplar set. It is also more computationally efficient since we make the matching and transferring process in the LR space with careful design;

(2) we propose a novel reference feature alignment network to search and align corresponding reference features to the LR content based on deformable sampling. We devise a feature aggregation module conditioned on the LR content to explicitly improve the set representation by favoring features that are high in quality and similarity to the LR content;

(3) we propose a novel correlation loss that helps represent the local texture and reconstruct more realistic details;

(4) compared with previous approaches, our method achieves state-of-the-art face hallucination performance on the CelebAMask-HQ testset. It has significantly fewer parameters and computational costs than recent exemplar-guided methods.

5.2 Related Works

5.2.1 Reference-based Super-Resolution

Reference-based SR (RefSR) [164] can reconstruct more accurate structures and details benefiting from the reference HR image. The general solution of RefSR includes two steps: searching the matched textures between LR inputs and HR references, and transferring the textures. Some of the previous RefSR approaches choose to align the LR and Ref images

with either global registration [148] or optical flow [149], [150]. Other methods choose to match by patches with gradient features [154], or deep features extracted by the CNN [155]–[157]. Shim *et al.* [153] changes the feature matching to LR space to reduce computation. Yang *et al.* [156] introduces the transformer architecture in a cross-scale manner to improve the accuracy of searching and transferring relevant textures. The above works usually include pixel-wise reconstruction loss, perceptual loss [18] and adversarial loss as the objective functions. Zhang *et al.* [159] introduce a Haar wavelet loss and a degradation loss to avoid over-smoothing in the final results. Besides, CMSR [165] further expands the reference source from a single image to a pre-built image pool and searches the $k-$nearest patches from the pool. Since these patch-based methods exhaustively conduct a patch-wise comparison of LR and reference feature maps, they usually have a high computational cost.

5.2.2 Face Hallucination

Face hallucination methods can be roughly divided into two categories: blind face hallucination and exemplar-guided restoration. The first category focuses more on integrating face priors in designing the reconstruction network and loss functions: some works include sub-branches for facial landmarks or face structures [143], [166]–[170], or a face parsing map [145], [171]. Using face structure priors may bring advantages, including the better recovery of the face shape, as reflected by fewer errors on face alignment and parsing. However, the reconstruction results might not look like the same person, especially when the input images contain barely any identifying information. To solve this problem, [172]–[174] employ identity information to supervised the training of the reconstruction network. However, these blind reconstruction methods are heavily influenced by the bias within the distribution of training data, and usually fail to generate satisfying results for minority groups.

The second category, exemplar-guided restoration, aims to use another HR image of the same person to improve the visual content quality of the generated images. [151], [152] include a warping sub-network for using the HR guidance, which increases the training steps as well as the overall number of parameters of the network. [160] uses moving least-squares to align the input and guidance images in the feature space and applies AdaIN for feature

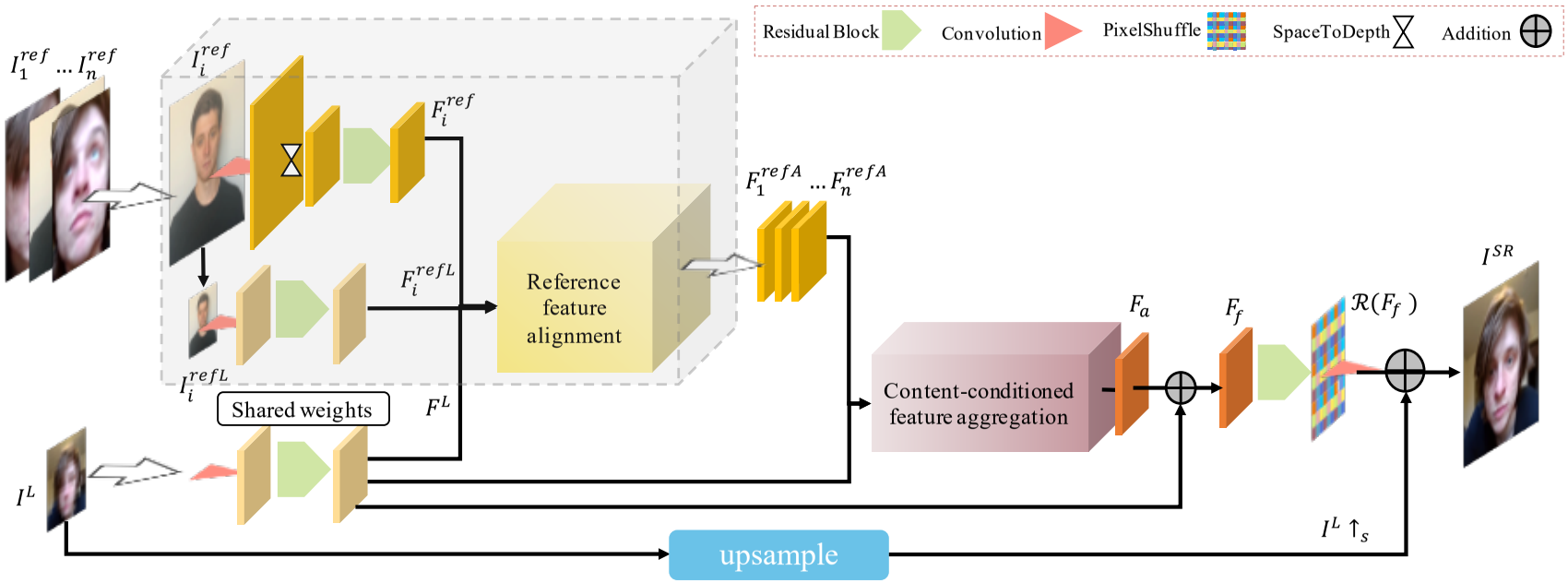


Figure 5.2. Overview of our **H**eadshot **I**mage Super-Resolution with **M**ultiple **E**xemplars (HIME) framework. Given an input LR image and any number of exemplars, it matches, aligns, and aggregates the features of the reference images conditioned on the input content to reconstruct the SR output.

transfer. It selects a single exemplar from the guidance images, thus it cannot fully use the rich information in the guidance face sets. [161] takes a step forward by using multiple exemplars with a weighted pixel average module in the network. But, it cannot handle the large deformation between unaligned faces.

Compared with the approaches above, our method can take full advantage of an unaligned exemplar set as a reference in headshot reconstruction, and our network is end-to-end trainable without requiring face-specific metadata.

5.3 HIME Framework

Given a low-resolution input face I^L and a set of high-resolution headshot images $\mathcal{I}^{ref} = \{I_{\mathrm{i}}^{ref}\}, \mathrm{i} = 1, 2, \ldots$ from the same identity, our goal is to generate the corresponding high-resolution image I^{SR}. To efficiently and accurately transfer the matched information from the unaligned reference sets of arbitrary length, we propose the *HIME* framework as illustrated in Figure 5.2. This framework consists of four main components: *feature extractor*, *reference feature alignment module (RFA)*, *content-conditioned feature aggregation module (CoFA)*, and *HR reconstructor*. The detailed structures of these modules will be introduced in Sections 5.3.1, 5.3.2, 5.3.3 and 5.3.4, respectively.

We first utilize an HR feature extractor to extract feature maps $\{F_{\mathrm{i}}^{ref}\}_{\mathrm{i}=1}^{n}$ from the reference set with n HR images. For efficient feature matching and transfer, we downsample these references to the LR space and use an LR feature extractor to extract the feature maps F^L from the input LR image, and $\{F_{\mathrm{i}}^{refL}\}_{\mathrm{i}=1}^{n}$ from the downsampled reference set, respectively. Then we refine the reference feature maps with the proposed RFA module. Furthermore, to better utilize the face set information, we use a CoFA module to aggregate the refined features into one. Finally, we reconstruct the HR face image from the aggregated feature map.

5.3.1 Feature Extractors

We adopt an HR feature extractor and an LR feature extractor to acquire the feature maps in HR space and LR space, respectively. The HR feature extractor turns the HR

reference images into a set of feature maps: $\{F_{\mathrm{i}}^{ref}\}_{\mathrm{i}=1}^{n}$. The RGB images are first converted into a mono-channel feature map as shown in Fig. 5.3, since the color information of the reference images is not important for the final output. Then, we adopt a space-to-depth operation to convert the HR feature maps into the same spatial resolution as the input without discarding any information. Next, we apply a convolution layer and k_h residual blocks [16] to extract the HR reference feature maps. The LR feature extractor generates feature maps for both the input image and the downsampled reference set with a convolutional layer and k_l residual blocks [16].

Image Mono-channel feature map

Figure 5.3. Visualization of the mono-channel feature map after the conversion. The image is 128×128-resolution.

5.3.2 Reference Feature Alignment

Given extracted feature maps F^L from the input LR image, and $\{F_{\mathrm{i}}^{ref}\}_{\mathrm{i}=1}^{n}$ and $\{F_{\mathrm{i}}^{refL}\}_{\mathrm{i}=1}^{n}$ from the reference images, we want to acquire guiding features that are well-aligned with the contents of the LR image to mitigate any mismatches in view or pose. To achieve this goal, previous reference-based face hallucination methods perform warping on RGB images [151], [152], which require a pre-trained dense warping model and thus a two-stage training scheme. We instead propose learning a feature alignment function $f(\cdot)$ to directly align the reference feature maps F_{i}^{ref} as shown in Figure 5.4. A general form of the alignment function can be formulated as:

$$F_{\mathrm{i}}^{refA} = f(F_{\mathrm{i}}^{ref}, F_{\mathrm{i}}^{refL}, F^L) = T(F_{\mathrm{i}}^{ref}, \Phi_{\mathrm{i}}), \tag{5.1}$$

where F_{i}^{refA} denotes the i-th aligned reference feature, $T(\cdot)$ is the sampling function, and Φ_{i} is the corresponding sampling parameters. Inspired by the deformable alignment [175], [176] in [86], [177], [178] for spatial and temporal super-resolution, we propose to use deformable sampling functions to implicitly capture the similarities between the LR content and the reference images. With the deformable sampling, our RFA module can refine the reference feature maps into an aligned form without any flow or landmark supervision.

The offset for the deformable sampling function should be learned based on the similarity between the reference image and the input LR image. To make these features comparable, we take F_{i}^{refL} and F^{L}, which are both in the LR space, to predict the offset Δp_{i} for sampling the F_{i}^{ref}:

$$\Delta p_{\mathrm{i}} = g([F_{\mathrm{i}}^{refL}, F^{L}]), \tag{5.2}$$

where Δp_{i} also refers to the sampling parameter Φ_{i}; $g(\cdot)$ denotes a general operation of convolution layers for the offset estimation; and $[\cdot,\cdot]$ denotes channel-wise concatenation. With the learned offset, the sampling function in Equation 5.1 can be performed with a deformable convolution [175], [176]:

$$F_{\mathrm{i}}^{refA} = T(F_{\mathrm{i}}^{ref}, \Phi_{\mathrm{i}}) = DConv(F_{\mathrm{i}}^{ref}, \Delta p_{\mathrm{i}}). \tag{5.3}$$

5.3.3 Content-conditioned Feature Aggregation

Now we have a set of aligned reference feature maps: $\{F_{\mathrm{i}}^{refA}\}_{\mathrm{i}=1}^{n}$ for the following feature transferring and reconstruction steps. As shown in Figure 5.5, the CoFA module aims to map this feature map set to a representation with fixed dimension. In this way, the reference image set with a different number of images can be represented in a unified manner. The representation is determined by all items in the set and conditioned on the LR content. Therefore it can be denoted as:$F_a = \mathcal{F}(F_1^{refA}, F_2^{refA}, \ldots, F_n^{refA} | F^{L})$, where $\mathcal{F}(\cdot)$ is the aggregation function that maps an arbitrary-sized set to a representation of fixed dimension.

It is challenging to find a proper $\mathcal{F}(\cdot)$ that aggregates features from the whole reference set to obtain an optimized representation. Based on the intuition that references with

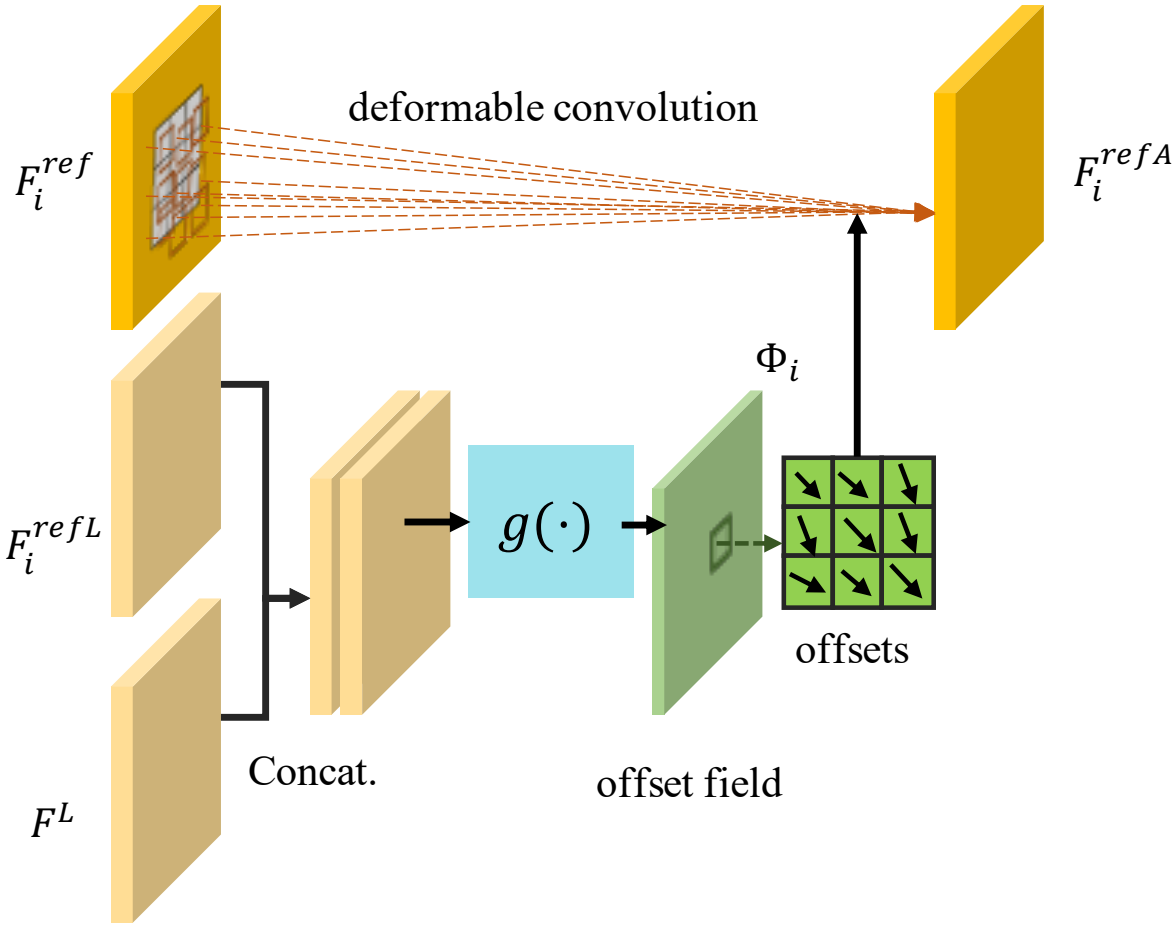


Figure 5.4. Reference feature alignment (RFA) based on deformable sampling. Since the aligned F_{i}^{refA} will be used for transferring the feature to the corresponding LR content, it will implicitly enforce the learned offset to match the similarities between the LR and reference images.

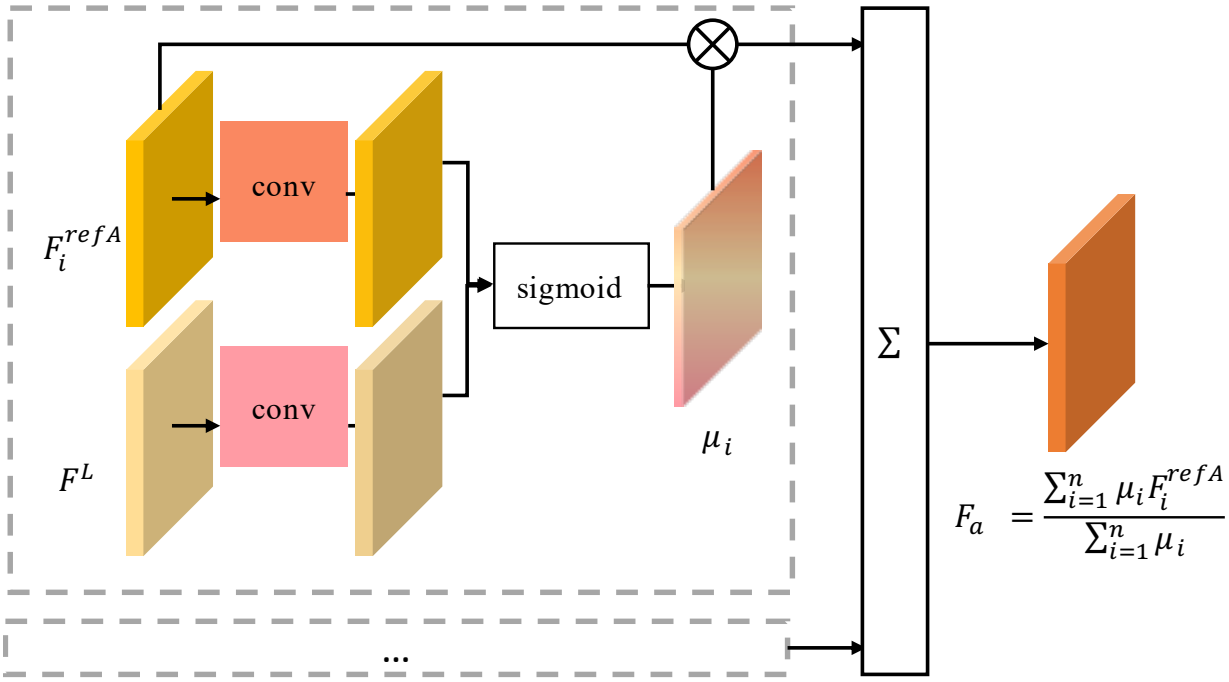


Figure 5.5. Illustration of the content-conditioned feature aggregation module. For each aligned reference feature, we compute a similarity score μ and then aggregate all the features in the set with a weighted average.

higher similarity and quality should contribute more to feature transfer, while faces with mismatched features and low-quality features should have less effect on the set representation, we denote $\mathcal{F}(\cdot)$ as:

$$\mathcal{F}(F_1^{refA}, \ldots, F_n^{refA} | F^L) = \frac{\sum_{\mathrm{i}=1}^{n} \mu_\mathrm{i} F_\mathrm{i}^{refA}}{\sum_{\mathrm{i}}^{n} \mu_\mathrm{i}}, \tag{5.4}$$

$$\mu_\mathrm{i} = \mathcal{S}(F_\mathrm{i}^{refA}, F^L), \tag{5.5}$$

where $\mathcal{S}(\cdot)$ generates a similarity score μ_i for the aligned reference feature map F_i^{refA} that is acquired in the same manner as shown by Equation 5.3. Therefore, the final representation of the set is a fusion of each feature weighted by its similarity score. For each aligned reference feature F_i^{refA}, the pixel-wise similarity score is calculated as:

$$\mathcal{S}(F_\mathrm{i}^{refA}, F^L) = \sigma(g_1(F^{refA})^T g_2(F^L)), \tag{5.6}$$

where $\sigma(\cdot)$ is sigmoid function that is used for bounding the outputs to the range $[0, 1]$ and stabilizing the gradient propagation; and $g_1(\cdot)$ and $g_2(\cdot)$ denote general convolution layers. The similarity score can also be regarded as an attention mask conditioned on the input content.

Finally, the summation F_a and LR feature map is sent to HR image reconstruction: $F_f = F_a + F^L$. The similarity computation and weighted aggregation steps are parameter-free. Thus, the CoFA module is light-weight by design.

5.3.4 High-Resolution Image Reconstruction

The HR reconstruction module takes the fused feature F_f as input and generates the residual of our target HR output. It is composed of k_r stacked residual blocks [16] for learning deep features and a sub-pixel upsampling module with PixelShuffle [179] initialized using the ICNR method as in [147], [180]. To encourage the network to focus on learning high-frequency information that is not present in the LR input, we introduce a long-range skip connection to form the final SR output: $I^{SR} = I^L \uparrow_s + \mathcal{R}(F_f)$, where $\uparrow$ denotes the bicubic upscaling operation and s denotes the scale factor; $\mathcal{R}(\cdot)$ denotes the reconstruction

operations as described above. Allowing the low-frequency information in the LR input to bypass the reconstruction network lowers the difficulty of reconstruction learning and accelerates the convergence of the optimization process. We use the Charbonnier penalty function [181] as the loss term for pixel-wise reconstruction to optimize our framework: $L_{rec} = \sqrt{||I^{HR} - I^{SR}||^2 + \epsilon^2}$, where I^{HR} denotes the ground-truth HR frame, and ϵ is empirically set to 1×10^{-3}.

Since the input and reference images are highly related in the face domain, our model can simultaneously learn the feature alignment and similarity score with only supervision from the HR ground truths through the end-to-end training.

5.4 Correlation Loss

5.4.1 Motivation

The commonly used pixel-wise reconstruction losses inevitably lead to over-smoothing of outputs and don't match the human visual perception of natural images [181], since they fail to capture the underlying local relationships between pixels. While the perceptual loss [18] and style loss [182] have been introduced to provide more perception-oriented supervision, they require a pretrained network from another high-level vision task, and are not versatile for representing textures of very high-resolution images due to the limits of training data. To effectively represent the local texture patterns at different scales in a controllable manner, we devise the correlation loss. It first builds a correlation map from the correlation between the center pixel and its neighbors to represent the spatial patterns. Thus, matching the correlation map can help the network reconstruct more realistic details and improve the perceptual quality of the output images.

5.4.2 Design of Correlation Loss

As shown in Figure 5.6, each image I can be represented by a 3D tensor of size (C, H, W), where C is the number of channels and (H, W) denotes the spatial resolution. We first subtract the mean of each channel to center the data around 0. For a given pixel $I(x, y)$, we

calculate its inner product with the neighboring pixels $I(x-\mathrm{i}, y-\mathrm{j})$ as well as itself within a $k \times k$ window:

$$cor(\mathrm{i}, \mathrm{j}, x, y) = \frac{1}{k^2}\langle I(x,y), I(x-\mathrm{i}, y-\mathrm{j})\rangle, \tag{5.7}$$

where $\mathrm{i}, \mathrm{j} \in \lfloor -\frac{k+1}{2}, \frac{k+1}{2} \rfloor$, and $\frac{1}{k^2}$ is for normalization. k is the maximal displacement for computing the local correlation. As a result, we can acquire a local correlation map M_{cor} of size $(k \times k, H, W)$. The correlation loss is defined as the distance between the correlation maps resulting from the ground truth HR and the generated SR images:

$$L_{cor} = ||M_{cor}^{HR} - M_{cor}^{SR}||. \tag{5.8}$$

In our implementation, we adopt the $L1-$distance for this loss term. A larger window size k can encode more information while quadratically increasing the computational cost. Thus, we define the dilated correlation following the same manner as the dilated convolution [183]. By increasing the dilation factor d, we can enlarge the correlation window from $k \times k$ to $(kd - d + 1) \times (kd - d + 1)$.

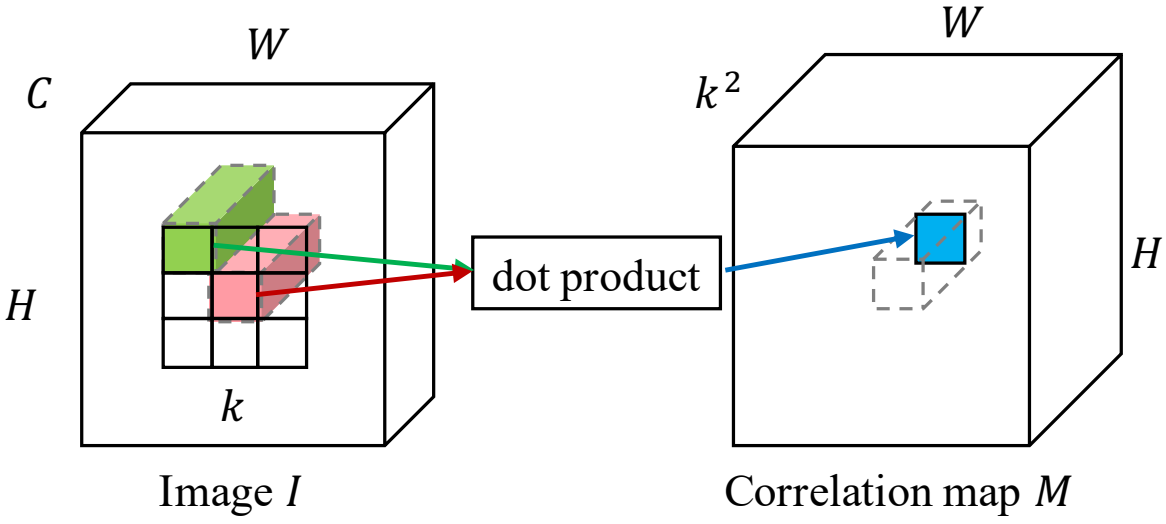


Figure 5.6. Illustration of the proposed correlation loss. The correlation operator is used for both generated and ground-truth images. Then we take the corresponding output correlation maps to calculate the correlation loss.

5.4.3 Visualizing Correlation Maps

To better understand the correlation operation, we visualize the correlation maps of the HR image with different correlation kernel window sizes $k \in \{3, 5, 7\}$. The correlation map is plotted in a heatmap using the maximum value of each channel. In Figure 5.7, we observe that the correlation map encodes the original image based on the local textures. In each correlation map, the blue areas correspond to the regions with more high-frequency features, like fur and the background, regardless of the color difference. While the red regions are more smooth, *e.g.*, the brightest and darkest part of the fur. With the increase of window size k, the correlation operator perceives and encodes features within a broader area, and thus looks more coarse-grained in the visualized results.

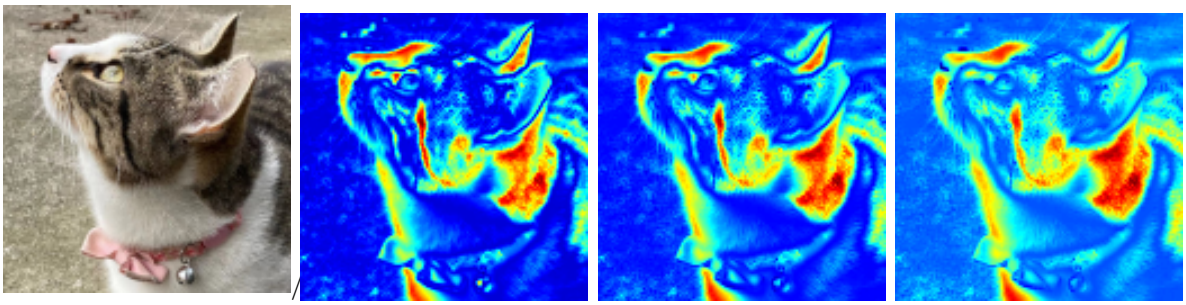


Figure 5.7. Visualization of correlation maps of different window sizes. The image is 128×128-resolution.

5.5 Experiment

5.5.1 Implementation Details

In our implementation, $k_l = 5$, $k_h = 3$, and $k_r = 20$ residual blocks are used in the LR feature extraction, HR feature extraction, and HR image reconstruction modules, respectively. For each LR input, we randomly select three different HR images to build the reference set during training and testing.

We use the Adam [127] optimizer, decaying the learning rate with a cosine annealing schedule for each batch [184] starting from 1×10^{-4}. For 16×16 LR inputs, we set the batch

size as 128 and train the network on 1 Nvidia P100 GPU for 8×10^4 iterations. Our network is implemented with PyTorch [134].

Architecture

HIME The overall structure of this network has been described previously. Here we plot the detailed structure and profile the number of parameters of each module in Figure 5.8: the LR feature extraction, HR feature extraction, RFA, CoFA, and HR reconstruction modules.
Discriminators We use the discriminator in StyleGANv2 [185] as the architecture for the discriminator in our framework. It includes a convolutional layer and several residual blocks that downsample the input feature into different scales and turns it into an output tensor as the final prediction.

Objective Function

For a fair comparison with previous methods, we train two types of models: reconstruction-oriented models HIME$_{rec}$ with L_{rec} only; and perception-oriented models HIME$_P$ including the pixel-wise reconstruction loss L_{rec}, the adversarial loss L_{adv}, the perceptual loss L_{per}, and our proposed correlation loss L_{cor}:

$$\mathcal{L}_P = \lambda_{rec} L_{rec} + \lambda_{adv} L_{adv} + \lambda_{per} L_{per} + \lambda_{cor} L_{cor}, \tag{5.9}$$

where λs are the weights for each loss term. In our implementation, $\lambda_{rec} = 1.0, \lambda_{adv} = 0.1, \lambda_{per} = 0.01, \lambda_{cor} = 0.1$. The pixel-wise reconstruction loss and the correlation loss have already been described. For the perceptual loss, we adopt the structure of VGG-19 [18] and extract the features Fea before the ReLU layer. The perceptual loss is measured by L_1 distance:

$$L_{per} = ||Fea^{HR} - Fea^{SR}||_1. \tag{5.10}$$

We adopt the relativistic GAN [186] for the L_{adv}:

$$L_{adv} = -\mathbb{E}_{HR}[\log(1 - D_{Ra}(I^{HR}, I^{SR}))] - \mathbb{E}_{SR}[\log(D_{Ra}(I^{SR}, I^{HR}))], \tag{5.11}$$

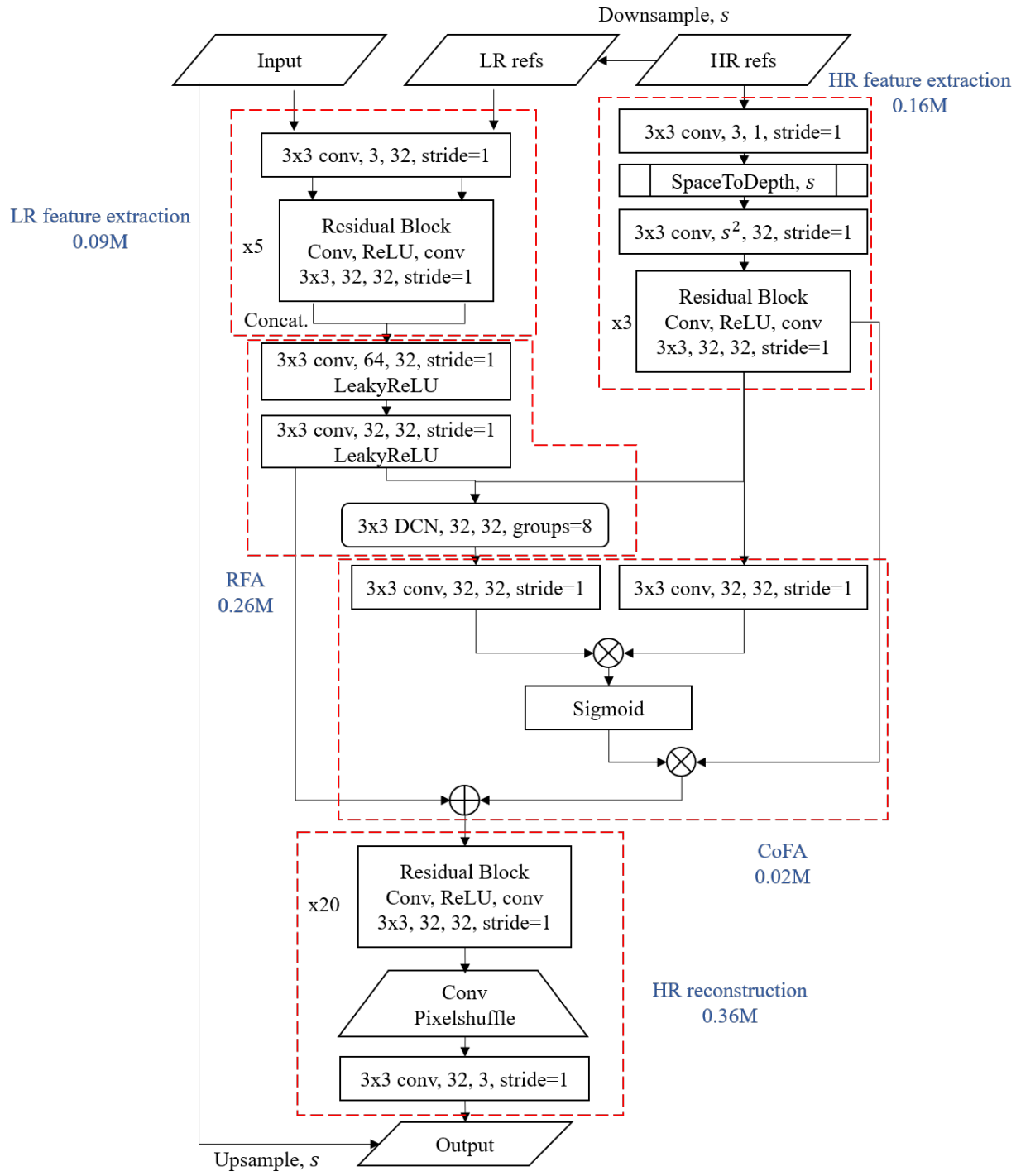


Figure 5.8. The architecture of our HIME. We profile the number of parameters for each module.

where I^{HR} and I^{SR} stand for the ground-truth and generated images, respectively. D_{Ra} denotes the relativistic average discriminator, which can be formulated as:

$$D_{Ra}(I^{HR}, I^{SR}) = \sigma(C(I^{HR}) - \mathbb{E}_{SR}[C(I^{SR})]), \tag{5.12}$$

where $C(\cdot)$ is the discriminator output, and σ is the Sigmoid function, $\mathbb{E}_{SR}[\,\cdot\,]$ stands for averaging all I^{SR} in a minibatch. The discriminator loss is defined as:

$$L_D = -\mathbb{E}_{HR}[\log(D_{Ra}(I^{HR}, I^{SR}))] - \mathbb{E}_{SR}[\log(1 - D_{Ra}(I^{SR}, I^{HR}))]. \tag{5.13}$$

Datasets

CelebAMask-HQ is used as the training and evaluation datasets [187]. It is a large-scale face image dataset including over 30,000 high-resolution (1024×1024) headshots selected from the CelebA dataset [188]. However, it doesn't have identity info. So we acquire the identity information from the original CelebA dataset. Still, each identity might have a very unbalanced number of images as shown in Figure 5.9. To make sure each identity has enough exemplars during training, we remove 3,300 out of 6,217 identities with < 4 images, which are not enough to construct a set of multiple references. The remaining identities are randomly split into a training set and an evaluation set, including 2,600 and 287 identities, respectively. We generate LR inputs by bicubic downsampling with factor= s. For each LR input, we randomly select three different HR images to build the reference set during training. The corresponding HR image of $s\times$ size is used for supervision. To evaluate our model's capability to conduct RefSR with different spatial resolutions and scale factors, we set $s = 2, 4, 8, 16, 32$ and 64 to generate a wide range of LR faces from 16×16 to 512×512.

Evaluation Metrics

We adopt the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [92] metrics to evaluate the reconstruction performance on all RGB channels. We also compare the perceptual quality of the generated images with LPIPS (Learned Perceptual Image Patch

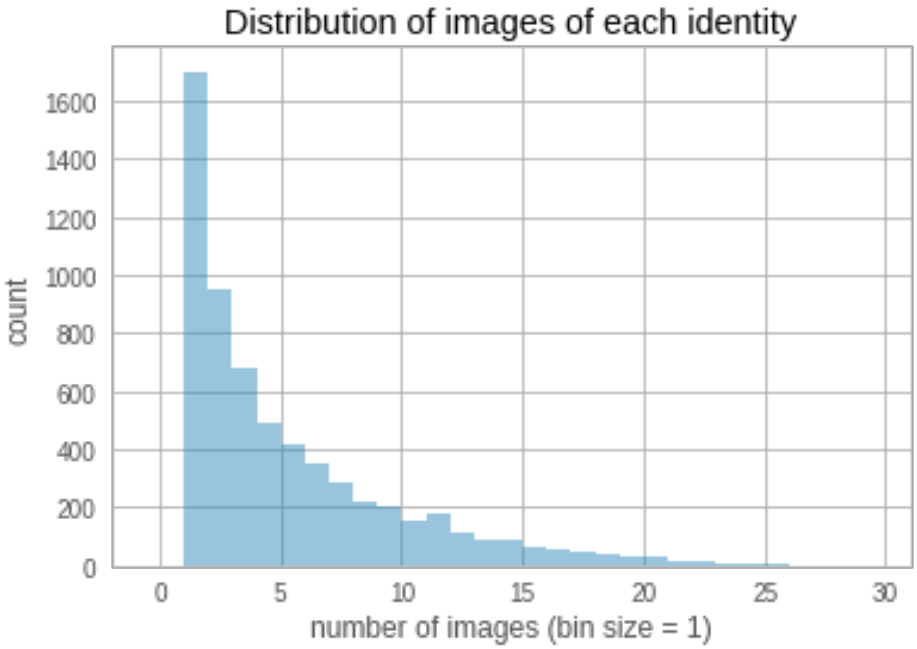


Figure 5.9. Histogram of number of images for each identity in the CelebAMask-HQ dataset.

Similarity) [189]. To measure the efficiency of the different methods, we also compare the model parameters and computational cost for each setting.

Besides, we also compare the model size with the number of parameters, and the computational cost with GMACs (Giga Multiply–ACcumulate operationS), which is computed using Turing Profiler under standard CPU rules. It should be noted that this metric may yield accurate estimates of computational cost for networks implemented on GPUs or hardwares with architectures especially designed for AI algorithms.

5.5.2 Comparison to the State of the Art

We evaluate the performance of our HIME network under the 4× and 8× upsampling setting following the previous approaches. For 4× upscale, we compare two SOTA RefSR methods: SRNTT [155][2] and TTSR [156], and three recent face restoration methods SPARNet [190], PSFR-GAN [171] and DFDNet [191]. We did not test DFDNet [191] on the 32×4 setting since its face and landmark detectors cannot handle such tiny faces. For 8× upsampling, we compare our method with five face hallucination methods: PFSR [146], FSRNet [145][3], GWAInet [152], SPARNet [190] and PSFR-GAN [171]. Quantitative results are shown in Table 5.1.

From Table 5.1, we can learn the following facts: (1) reference-based SR methods, like SRNTT, TTSR and our HIME, demonstrate better performance than other non-reference approaches on both distortion-oriented metrics and perception-oriented metrics, which validates that using references can improve the SR fidelity. Our network outperforms the second-best result by 1.09 dB on $(32, 4)$, and 0.83 dB on $(64, 4)$; (2) Although SRNTT and TTSR have fewer parameters than other comparison methods, their computational costs are relatively high due to the exhaustive search during feature matching. With the learnable feature extractors, our model is over 7× smaller than SRNTT and TTSR. The small model size and the reference feature alignment in LR space make our network have 14.3× fewer GMACs than TTSR. For the $(16, 8)$ setting, we can observe that our method performs well

[2]↑We adopt a third-party PyTorch implementation: https://github.com/S-aiueo32/srntt-pytorch
[3]↑We adopt a third-party Pytorch implementation: https://github.com/cydiachen/FSRNET-pytorch.

Table 5.1. Quantitative comparison of our results and other SOTA methods. The best results are shown in **bold**.

(LR, s)	Methods	PSNR	SSIM	LPIPS	Params (M)	GMACs
$(32, 4)$	Bicubic	25.64	0.7752	0.3229	-	-
	SRNTT [155]	28.02	0.8434	0.0682	6.30	36.47
	TTSR [156]	27.31	0.8346	0.0633	6.73	26.62
	SPARNet [190]	20.50	0.6118	0.1617	85.73	45.25
	PSFR-GAN [171]	25.47	0.7709	0.0981	67.05	117.84
	HIME_{rec}	**29.11**	**0.8794**	0.1136	**0.87**	**1.86**
	HIME_{P}	27.16	0.8269	**0.0464**	**0.87**	**1.86**
$(64, 4)$	Bicubic	28.40	0.8169	0.2860	-	-
	SRNTT [155]	30.41	0.8552	0.0906	6.30	145.89
	TTSR [156]	29.87	0.8484	0.0851	6.73	106.48
	SPARNet [190]	23.26	0.6990	0.1341	85.73	180.99
	PSFR-GAN [171]	26.62	0.7685	0.1039	67.05	161.89
	DFDNet [191]	21.55	0.6587	0.1581	133.34	601.04
	HIME_{rec}	**31.24**	**0.8785**	0.1611	**0.87**	**7.48**
	HIME_{P}	29.06	0.8262	**0.0633**	**0.87**	**7.48**
$(16, 8)$	Bicubic	21.83	0.5929	0.5247	-	-
	PFSR[146]	21.44	0.5778	0.2065	10.08	8.97
	FSRNet [145]	20.03	0.5749	0.2865	15.52	3.20
	GWAINet [152]	21.96	0.5844	0.2056	4.29	6.55
	SPARNet [190]	19.00	0.5022	0.2576	85.73	45.25
	PSFR-GAN [171]	22.05	0.6102	0.2062	67.05	117.84
	HIME_{rec}	**24.54**	**0.7411**	0.2433	**0.90**	**0.49**
	HIME_{P}	22.45	0.6338	**0.1297**	**0.90**	**0.49**

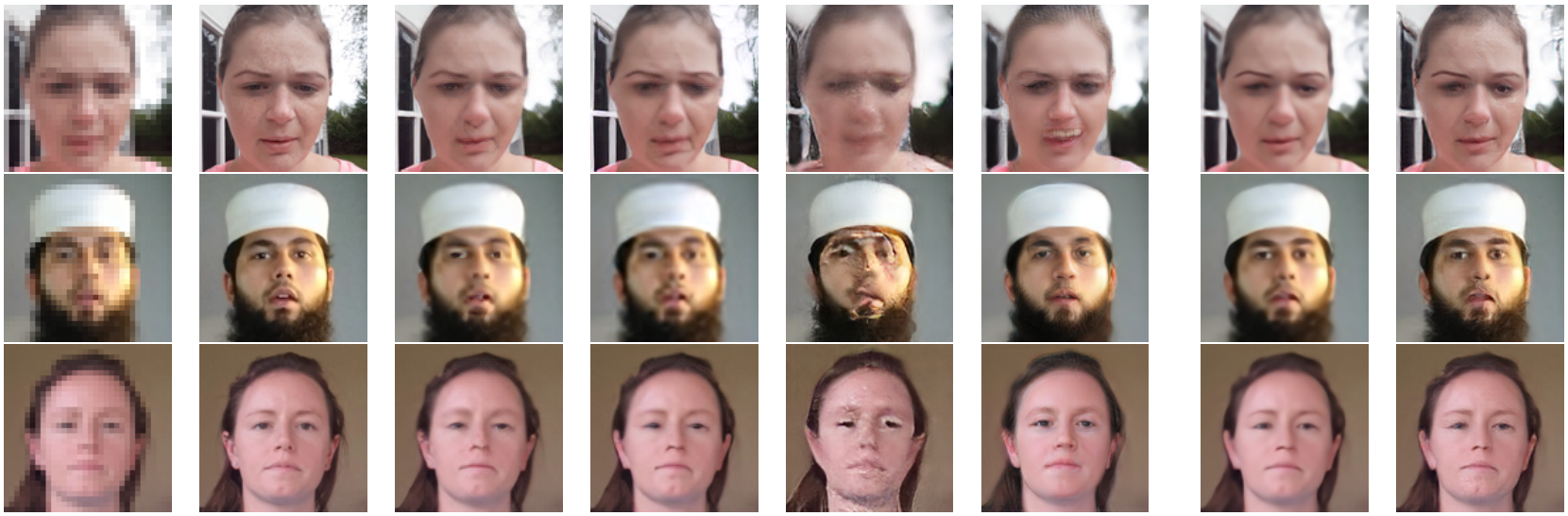


Figure 5.10. Qualitative comparison with SOTA methods for 4× upscale setting. Input resolution: 32×32.

even under the very challenging 8× upsampling setting. Our model is almost 5× smaller than the GWAINet while 6.5× fewer in GMACs than FSRNet.

The visual results on the DFDC dataset [192] are shown in Figure 5.10, which validates our observations above. RefSR methods like SRNTT, TTSR and ours can generate more robust and visually pleasing results. For the GAN-based face enhancement methods SPARNet and PSFR-GAN, while their results are rich in details, sometimes they fail on tiny faces with deformations.

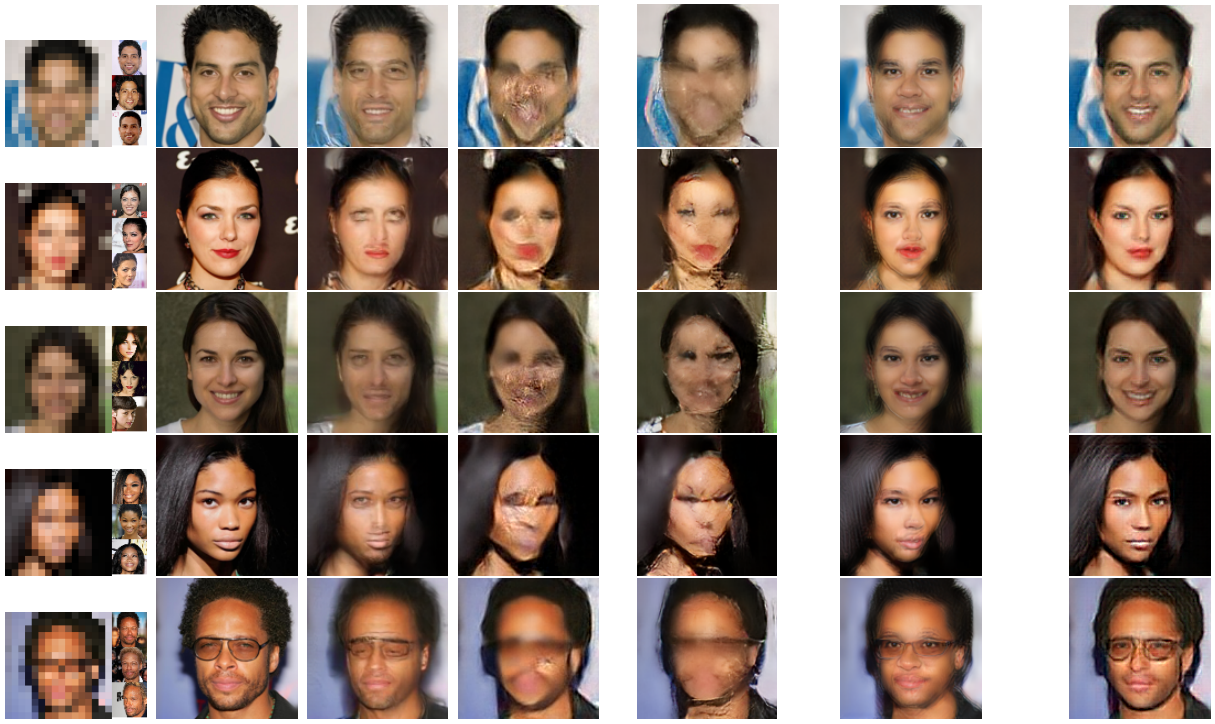


Figure 5.11. Qualitative comparison with SOTA methods for 8× upscale setting. Input resolution: 16 × 16. Zoom in for a better view of the reference images.

The 8× upscale results of 16 × 16 input images are shown in Figure 5.11, we compare our method with other face-hallucination methods: PFSR [146], GWAInet [152], SPARNet [190], and PSFR-GAN [171]. We show the three reference images side by side with the LR input. Comparing these results with the input and the ground truths, we can observe that PFSR generates rich face details as well as artifacts: all the generated eyes are similar, and the

shapes of generated lips are incorrect. For GWAInet and SPARNet, they fail to handle such tiny faces as input. The PSFR-GAN can generate photo-realistic results on some components, *e.g.* eyes, nose. However, the overall face structure is not well preserved in the generated outputs. Compared with the above methods, our model can synthesize better results that preserve the correct face structure. Besides, the facial components are realistic and close to the ground-truth.

The 4× upscale results of 32×32 input images in addition to those shown in Figure 5.12 are shown in Figure 5.12, we compare our method with other RefSR methods: SRNTT [155] and TTSR [156], and face hallucination methods SPARNet [190] and PSFR-GAN [171]. For SRNTT and TTSR, the textures of hairs and backgrounds are pretty good, while the eye regions are not as good. SPARNet cannot handle such tiny faces. PSFR-GAN can generate more realistic facial components. However, these results are not similar to the ground truth. Compared with these methods, our outputs have better fidelity and can construct vivid details, even for non-symmetric faces, profile faces, and occluded faces.

The 4× upscale results of 64×64 input images are shown in Figure 5.10. We add DFDNet [191] into the comparison. Compared with other methods, our method can generate sharper outputs that have higher fidelity. Our method is also robust to extreme poses and facial expressions, and complex textures of hairs. The other RefSR methods, SRNTT and TTSR, cannot generate sharp enough results due to the misalignment and distortions between the LR input and the reference image. The facial components generated by SPARNet are not similar to the ground truth, *e.g.* shape of eyes. The outputs of PSFR-GAN are not as sharp as ours. For DFDNet, while the results have vivid details, the overall image has a color shift, making the outputs look as though the images were captured in soft and warm light. Due to this reason, DFDNet doesn't perform as well as other methods in terms of PSNR and SSIM.

5.5.3 More Results of Our Method

Here we show more results of our method on these challenging settings: ×8 upscale for input resolution 32 in Figure 5.14, 64×64 in Figure 5.15, and 128×128 in Figure 5.16 and

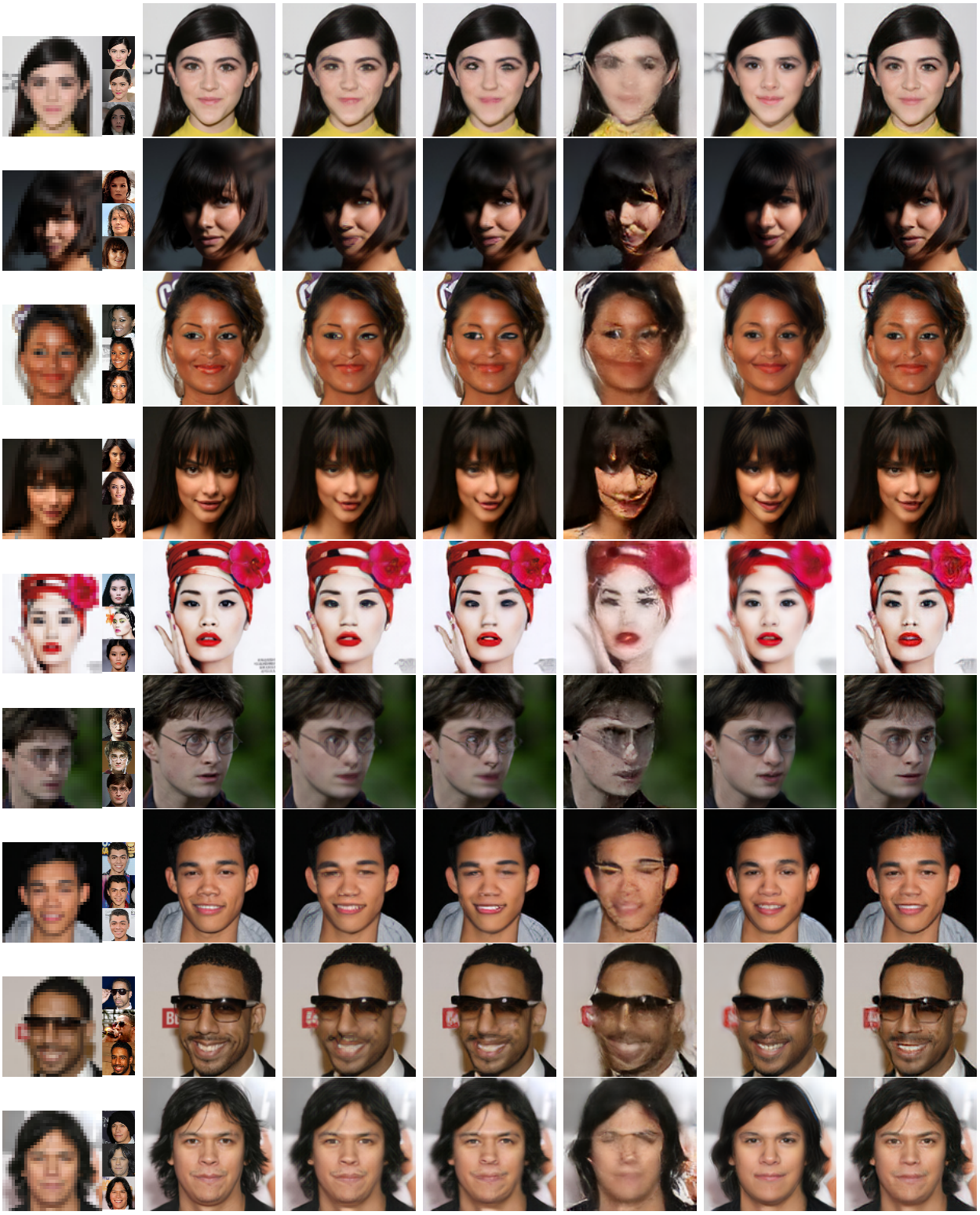


Figure 5.12. Qualitative comparison with SOTA methods for 4× upscale setting. Input resolution: 32×32.

Figure 5.13. Qualitative comparison with SOTA methods for 4× upscale setting. Input resolution: 64×64. Zoom in for a better view of the reference images.

5.17. Our method works well across different input resolutions and can generate visually-pleasing results while maintaining a good fidelity of the input identity.

5.5.4 Failure Cases

In our current framework, the proposed RFA module can implicitly enforce the matching and alignment of reference images according to LR contexts. However, when the input resolution is too low to maintain some high-frequency contents that are not so common in other face images, the results from our model still suffer from the following issues as shown in Figure 5.18: incorrect eye-gazing and facial component reconstruction, squinted eyes turned into open eyes, incorrect headwear, and failure to reconstruct hands on the face. To alleviate this problem, we should consider collecting more diverse data with such challenging cases to increase the robustness of the model.

5.5.5 Ablation Study

In previous sections, we have already illustrated the superiority of our proposed methods. In this section, we perform a comprehensive ablation study to further demonstrate the effectiveness of the correlation loss and different modules in our network. We also discuss the influence of face chirality under different data augmentation strategies. All experiments below are conducted under the same setting: $8\times$ upscale with input size 16×16 headshot images.

Effectiveness of Deformable Feature Alignment

To investigate the proposed RFA module, we compare three models: (a), (b), and (c), where (a) replaces the deformable convolution in the RFA module with common convolution that does not have the capability of feature alignment, and (b) utilizes the deformable convolution as illustrated in Section 5.3.2, (c) changes the way of computing Δp_{i} into $g(F_{\mathrm{i}}^{ref}, F^{L})$ which is across the LR and HR spaces.

From Table 5.2, we can see that adopting the deformable feature alignment brings up the performance on all metrics compared with using the common convolution of the same

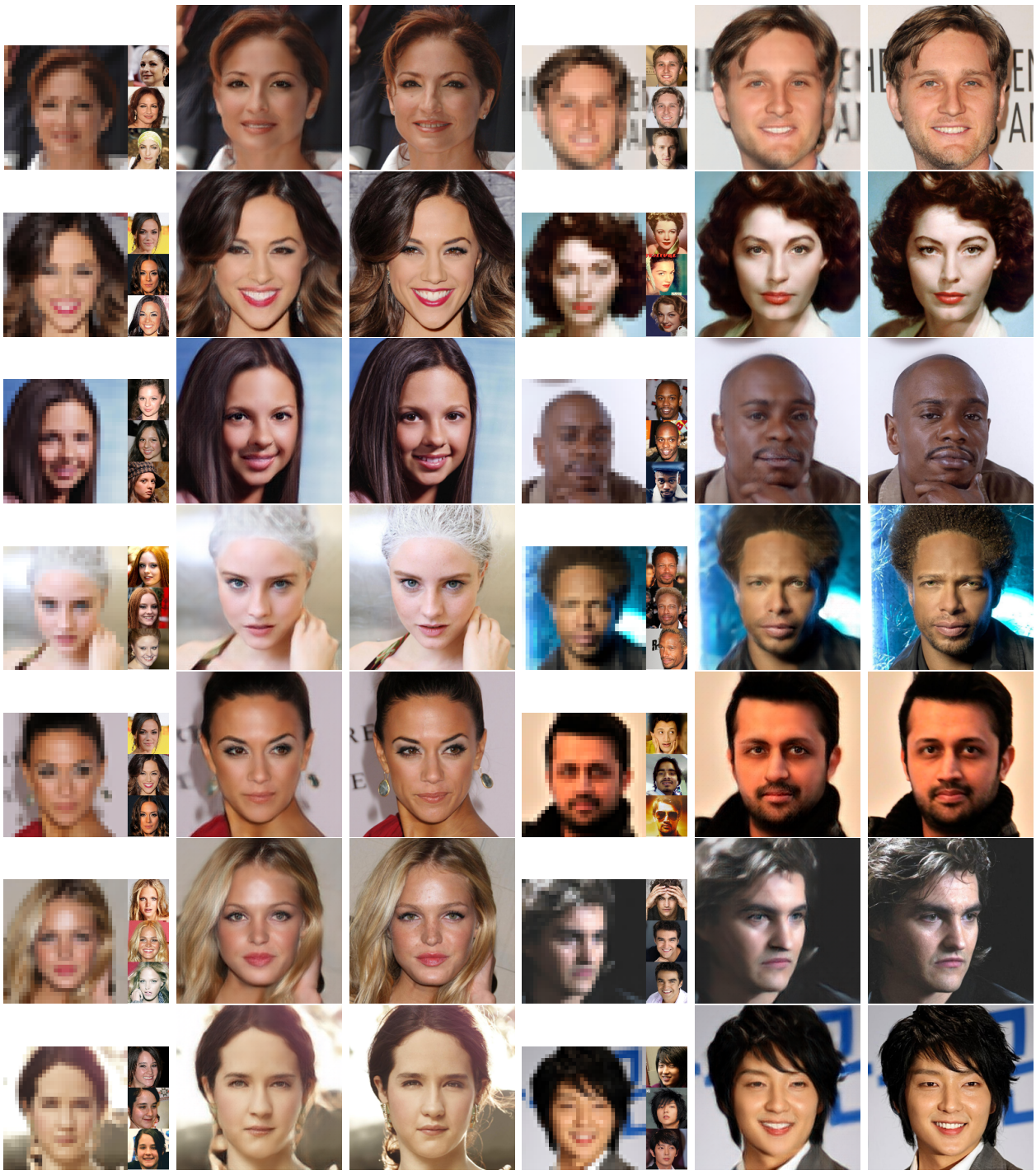


Figure 5.14. Qualitative result of our method for 8× upscale setting. Input resolution: 32×32. Zoom in for a better view of the reference images.

Figure 5.15. Qualitative result of our method for 8× upscale setting. Input resolution: 64×64. Zoom in for a better view of the reference images.

Figure 5.16. Qualitative result of our method for 8× upscale setting. Input resolution: 128 × 128. Zoom in for a better view of the reference images.

Figure 5.17. Qualitative result of our method for 8× upscale setting. Input resolution: 128 × 128. Zoom in for a better view of the reference images.

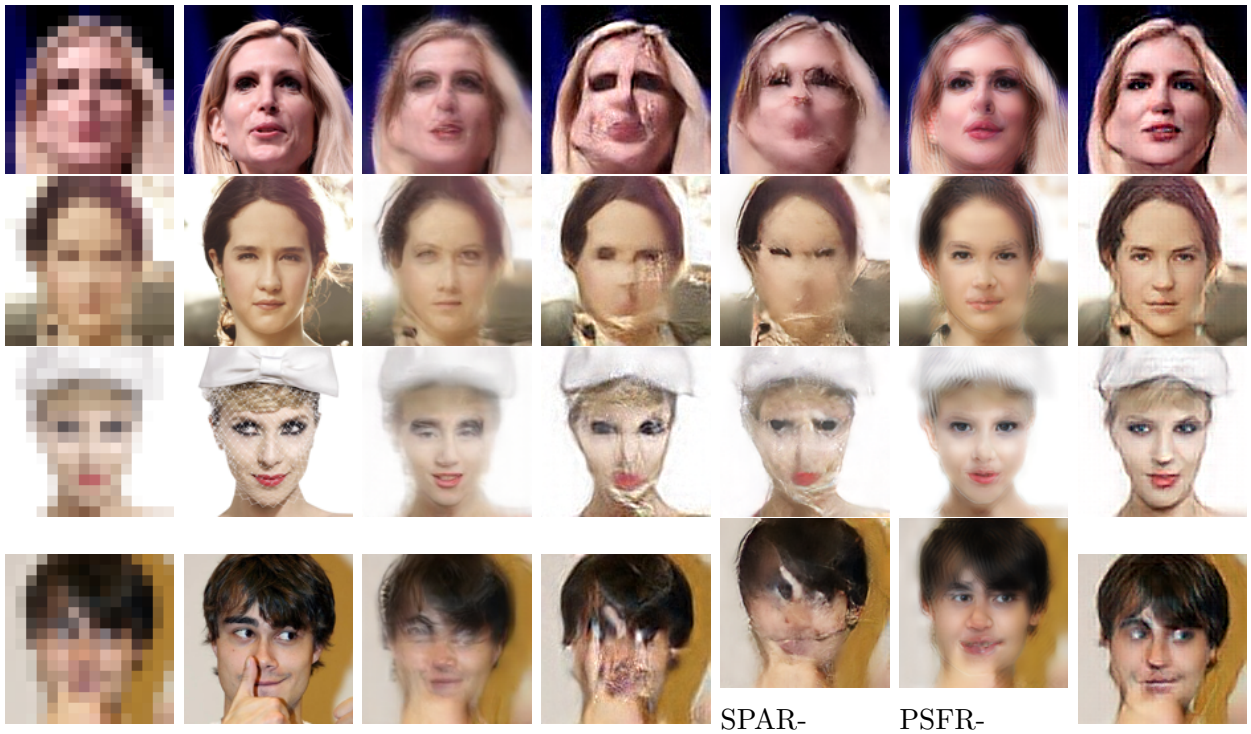


Figure 5.18. Failure cases: facial component, squinted eyes, headwear, and hands, Input resolution: 16×16 for $8\times$ upscale.

Table 5.2. Ablation study of feature alignment methods.

Methods	PSNR↑	SSIM↑	LPIPS↓
Conv	24.33	0.7311	0.2605
Dconv-LR ref	24.38	0.7339	0.2533
Dconv-ref	24.36	0.7333	0.2537

kernel size. The results demonstrate that our deformable feature alignment module can better match the features between the LR content and the references and is more robust to the misalignment and distortion. Furthermore, we observe that model (b) is slightly better than (c), which indicates that matching the features in the same space is better than cross-space. Our network conducts the offset computation and feature matching in the LR space, achieving a better performance while reducing the computational cost.

Set Feature Aggregation

To validate the effect of our proposed feature aggregation mechanism in the CoFA module, we compare three different models: (a) averages the features without content conditioning, (b) aggregates the features by max-pooling across the set, and (c) is our proposed aggregation method weighted by the learned content similarity. The quantitative results are shown in Table 5.3.

Table 5.3. Ablation study of feature aggregation methods.

Methods	PSNR↑	SSIM↑	LPIPS↓
Average	22.120	0.6350	0.4332
Max-pool	22.118	0.6349	0.4331
CoFA	24.381	0.7339	0.2533

From Table 5.3, we can see that the model with our content-conditioned feature aggregation module outperforms the average and max pooling by over 2 dB in terms of PSNR. Adopting the CoFA module greatly improves performance on all metrics, which indicates that our designed module can extract a better set representation, helping to restore the LR information and enhance the output quality.

Effect of Multiple Exemplars

To validate whether using an exemplar set can improve the face super-resolution result, we conduct the following experiments: (a) non-ref: a baseline SR network without references and removing the HR matching and aggregation modules, (b) training and testing with one

reference image and (c) with three reference images. From the results in Table 5.4, we can observe that using references significantly increases the PSNR by 0.49 dB while using multiple references further improves it by 0.19 dB. Such improvements also apply to the SSIM and LPIPS. These results verify that our model can benefit from the rich information in the exemplar set, and can effectively utilize the corresponding features to improve the output quality.

Table 5.4. Ablation study of multiple exemplars by changing the number of references during training and testing.

Num of Ref	PSNR↑	SSIM↑	LPIPS↓
0	23.84	0.7088	0.3440
1	24.35	0.7318	0.2572
3	24.54	0.7409	0.2433

Effect of Correlation Loss

To justify the effectiveness of correlation loss, we experimentally compare different configurations of HIME in Table 5.5. We consider the following models: (a) reconstruction loss only; (b) reconstruction loss + correlation loss; (c) multiple losses in GAN training (without correlation loss); (d) correlation loss + (c).

Table 5.5. Effectiveness of our proposed correlation loss.

Methods	PSNR↑	SSIM↑	LPIPS↓
L_{rec}	24.38	0.7339	0.2533
$L_{rec} + L_{cor}$	24.35	0.7346	0.2437
L_P w/o L_{cor}	22.44	0.6204	0.1543
L_P w/ L_{cor}	23.28	0.6673	0.1389

From Table 5.5, by comparing the first two rows, we can observe that introducing the correlation loss slightly decreases the PSNR. However, it improves the structural and perceptual metrics SSIM and LPIPS, which demonstrates that the proposed correlation loss benefits the reconstruction of local textures. Comparing the last two rows, training with the

correlation loss greatly leverages the perception-oriented model's performance on all metrics, which further validates the effectiveness of the correlation loss as perception-oriented supervision.

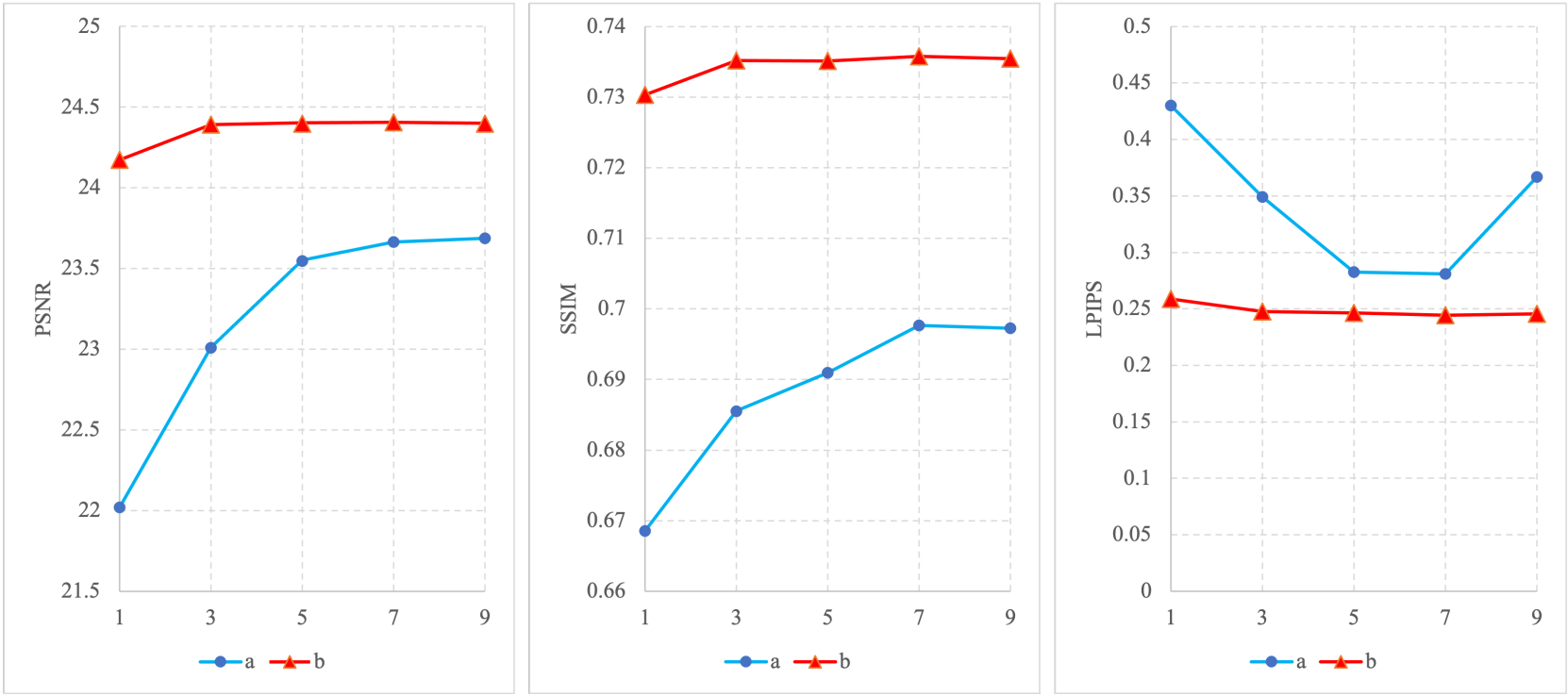
Figure 5.19 shows the performance of HIME for different correlation window sizes $k \in \{1, 3, 5, 7, 9\}$, where $k = 1$ degrades to the common $L1$ loss of the squared pixel values. We conduct two types of experiments: (a) training with correlation-loss only (plotted in blue), (b) fine-tuning with both L_{rec} and L_{cor} (plotted in red). Viewing the blue plots, we can observe that with the growth of k, the model performs better in terms of PSNR and SSIM. These results demonstrate that the correlation map itself is a good representation of the RGB image. With a larger window size, the correlation map can encode more information. Still, such improvement becomes more marginal when k is large enough. When $k = 9$, the LPIPS even increases. As for the red plots, we can see a similar trend: when $k \geq 3$, the improvement on PSNR and SSIM is very trivial. These results indicate that for a certain scale, there exists a range of k that work best in representing the local patterns. Within this range, the LPIPS scores keep decreasing with the increase of k. It implies that the correlation loss is more like perception-oriented supervision, which validates our description in Section 5.4.

Face Chirality

Typical human faces contain a variety of asymmetries. [193] brings up the visual chirality in faces and the distribution bias in public face datasets. Here we compare the influence of such asymmetries in headshot RefSR to answer the following questions: 1) Does the mismatch between input LR and references matter? 2) Does the bias in the training set influence the reconstruction performance?

Table 5.6. Influence of face chirality.

Models	No-aug	Uneven h-flip	Even h-flip	PSNR↑	SSIM↑
(a)	√			22.60	0.660
(b)		√		22.52	0.654
(c)			√	23.63	0.662



PSNR SSIM LPIPS

Figure 5.19. Effect of correlation window size k on output quality in terms of PSNR, SSIM, and LPIPS: (a) training with L_{cor} only, (b) fine-tuning with both L_{rec} and L_{cor}.

Tabel 5.6 shows our experimental results of changing the augmentation: (a) no augmentation; (b) randomly horizontal-flip the LR or Ref images, but not both for a given pair, which introduces the face view mismatch; (c) randomly horizontal-flip both the LR and reference images, which balances the number of left and right faces without introducing mismatches. By comparing (a) and (b), we can observe that the PNSR drops by 0.08 and the SSIM drops by 0.006, respectively, which indicates that training with mismatched views of faces would impair the model's performance. Compared with (a), (c) performs better in terms of the PSNR and SSIM by a small margin, which demonstrates that the proper augmentation improves the performance by mitigating the distribution bias in the dataset.

5.6 Conclusion and Future Work

In this paper, we propose an efficient framework for headshot image super-resolution with multiple exemplars without face structure priors or pretrained dense warper. To achieve this, we introduce a reference feature alignment module to search and align corresponding features to the LR content. To construct an optimized set representation, we propose a feature aggregation network conditioned on the input content. With such a design, our network can learn to fully utilize the rich information in the exemplar set and to be robust to misalignment and deformations. Furthermore, we propose a correlation loss that supervises the reconstruction of local textures with correlation maps. We believe that our new **H**eadshot **I**mage Super-Resolution with **M**ultiple **E**xemplars network (HIME) provides a novel idea to efficiently utilize a set of data for the reference-based super-resolution and face hallucination task. In future works, we will explore other aggregation methods to generate a better set representation with the aid of face priors. In addition, we will further validate the effectiveness of the correlation loss as generic supervision for other low-level tasks, *e.g.* image denoising, video frame interpolation, style transfer, *etc.*.

6. SPACE-TIME VIDEO SUPER-RESOLUTION

6.1 Introduction

Space-Time Video Super-Resolution (STVSR) [194] aims to automatically generate a photo-realistic video sequence with high space-time resolution from a low-resolution and low frame rate input video. Since HR high frame rate (HFR) videos are more visually appealing containing fine image details and clear motion dynamics, they are desired in rich applications, such as film making and high-definition television.

To tackle the problem, most existing works in previous literatures [194]–[199] usually adopt hand-crafted regularization and make strong assumptions. For example, space-time directional smoothness prior is adopted in [194], and [195] assumes that there is no significant change in illumination for the static pixels. However, these strong constraints make the methods have limited capacity in modeling various and diverse space-time visual patterns. Besides, the optimization for these methods is usually computationally expensive (*e.g.*, ~ 1 hour for 60 frames in [195]).

In recent years, deep convolutional neural networks have shown promising efficiency and effectiveness in various video restoration tasks, such as video frame interpolation (VFI) [200], video super-resolution (VSR) [201], and video deblurring [202]. To design an STVSR network, one straightforward way is by directly combining a video frame interpolation method (*e.g.*, SepConv [203], ToFlow [204], DAIN [205] *etc.*) and a video super-resolution method (*e.g.*, DUF [206], RBPN [207], EDVR [177] *etc.*) in a two-stage manner. It firstly interpolates missing intermediate LR video frames with VFI and then reconstructs all HR frames with VSR. However, temporal interpolation and spatial super-resolution in STVSR are intra-related. The two-stage methods splitting them into two individual procedures cannot make full use of this natural property. In addition, to predict high-quality video frames, both state-of-the-art VFI and VSR networks require a big frame reconstruction network. Therefore, the composed two-stage STVSR model will contain a large number of parameters and is computationally expensive.

To alleviate the above issues, we propose a unified one-stage STVSR framework to learn temporal interpolation and spatial super-resolution simultaneously. We propose to adap-

tively learn a deformable feature interpolation function for temporally interpolating intermediate LR frame features rather than synthesizing pixel-wise LR frames as in two-stage methods. The learnable offsets in the interpolation function can aggregate useful local temporal contexts and help the temporal interpolation handle complex visual motions. In addition, we introduce a new deformable ConvLSTM model to effectively leverage global contexts with simultaneous temporal alignment and aggregation. HR video frames can be reconstructed from the aggregated LR features with a deep SR reconstruction network. To this end, the one-stage network can learn end-to-end to map an LR, LFR video sequence to its HR, HFR space in a sequence-to-sequence manner. Experimental results show that the proposed one-stage STVSR framework outperforms state-of-the-art two-stage methods even with many fewer parameters. An example is illustrated in Figure 1.

The contributions of this chapter are six-fold:

- We propose a one-stage space-time super-resolution network that can address temporal interpolation and spatial SR simultaneously in a unified framework. Our one-stage method is more effective than two-stage methods taking advantage of the intra-relatedness between the two sub-problems. It is also computationally more efficient since only one frame reconstruction network is required rather than two large networks as in state-of-the-art two-stage approaches.

- We propose a frame feature temporal interpolation network leveraging local temporal contexts based on deformable sampling for intermediate LR frames. We devise a novel deformable ConvLSTM to explicitly enhance temporal alignment capacity and exploit global temporal contexts for handling large motions in videos.

- Our one-stage method achieves state-of-the-art STVSR performance on both Vid4 [208] and Vimeo [204]. It is 3 times faster than the two-stage network: DAIN [205] + EDVR [177] while having a nearly 4× reduction in model size.

- We improve model performance via integrating guided feature interpolation learning into our one-stage framework.

- We investigate space-time video super-resolution under even noisy conditions, in which random noises or JPEG compression artifacts corrupt the input LR video frames. Such applications allow us to explore the flexibility and potential breath of our Zooming SlowMo (ZSM) method.
- Additional and extensive experimental results demonstrate the effectiveness of the proposed guided interpolation learning, and further show the superiority of our one-stage network on tackling more challenging noisy STVSR tasks.

6.2 Related Work

In this section, we discuss works on three related topics: video frame interpolation (VFI), video super-resolution (VSR), and space-time video super-resolution (STVSR).

6.2.1 Video Frame Interpolation

The target of video frame interpolation is to synthesize non-existent intermediate frames in between the original frames. Meyer *et al.* [209] introduced a phase-based frame interpolation method, which generates intermediate frames through per-pixel phase modification. Long *et al.*[210] predicted intermediate frames directly with an encoder-decoder CNN. Niklaus *et al.* [200], [203] regarded the frame interpolation as a local convolution over the two input frames, and used a CNN to learn a spatially-adaptive convolution kernel for each pixel for high-quality frame synthesis. To explicitly handle motions, there are also many flow-based video interpolation approaches [135], [205], [211]–[213]. These methods usually have inherent issues with inaccuracies and missing information from the optical flow results. In our one-stage STVSR framework, rather than synthesizing the intermediate LR frames as current VFI methods do, we interpolate features from two neighboring LR frames to directly synthesize LR feature maps for missing frames without requiring explicit supervision.

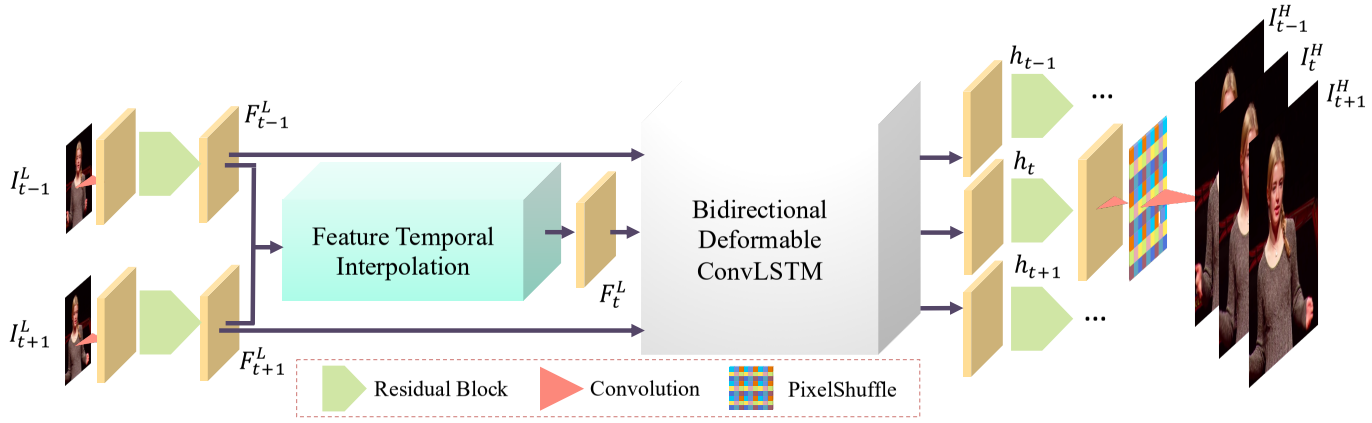


Figure 6.1. Overview of our one-stage STVSR framework. It directly reconstructs consecutive HR video frames without synthesizing LR intermediate frames I^L_t. Feature temporal interpolation and bidirectional deformable ConvLSTM are utilized to leverage local and global temporal contexts for better exploiting temporal information and handling large motions. Note that we only show two input LR frames from a long sequence in this figure for a better illustration.

6.2.2 Video Super-Resolution

Video super-resolution aims to reconstruct an HR video frame from the corresponding LR frame (reference frame) and its neighboring LR frames (supporting frames). One key problem for VSR is how to temporally align the LR supporting frames with the reference frame. Several VSR methods [201], [204], [214]–[216] use optical flow for explicit temporal alignment, which first estimates motions between the reference frame and each supporting frame with optical flow and then warps the supporting frame using the predicted motion map. Recently, RBPN proposes to incorporate the single image and multi-frame SR for VSR in which flow maps are directly concatenated with LR video frames. However, it is difficult to obtain accurate flow; and flow warping also introduces artifacts into the aligned frames. To avoid this problem, DUF [206] with dynamic filters and TDAN [217] with deformable alignment were proposed for implicit temporal alignment without motion estimation. EDVR [177] extends the deformable alignment in TDAN by exploring multiscale information. However, most of the above methods are many-to-one architectures, and they need to process a batch of LR frames to predict only one HR frame, which makes the methods computationally inefficient. Recurrent neural networks, such as convolutional LSTMs [218] (ConvLSTM), can ease sequence-to-sequence (S2S) learning; and they are adopted in VSR methods [219], [220] for leveraging temporal information. However, without explicit temporal alignment, the RNN-based VSR networks have limited capability in handling large and complex motions within videos. To achieve efficient yet effective modeling, unlike existing methods, we propose a novel ConvLSTM structure embedded with an explicit state updating cell for space-time video super-resolution.

Rather than simply combining a VFI network and a VSR network to solve STVSR, we propose a more efficient and effective one-stage framework that simultaneously learns temporal feature interpolation and spatial SR without accessing LR intermediate frames as supervision.

6.2.3 Space-Time Video Super-Resolution

Shechtman *et al.* [221] firstly proposed to extend SR to the space-time domain. Since pixels are missing in LR frames and even several entire LR frames are unavailable, STVSR is a highly ill-posed inverse problem. To increase video resolution both in time and space, [221] combines information from multiple video sequences of dynamic scenes obtained at sub-pixel and sub-frame misalignments with a directional space-time smoothness regularization to constrain the ill-posed problem. Mudenagudi [195] posed STVSR as a reconstruction problem using a Maximum a posteriori-Markov Random Field [222] with graph-cuts [223] as the solver. Takeda *et al.* [196] exploited local orientation and local motion to steer spatio-temporal regression kernels. Shahar *et al.* [197] proposed to exploit a space-time patch recurrence prior within natural videos for STVSR. However, these methods have limited capacity to model rich and complex space-time visual patterns, and the optimization for these methods is usually computationally expensive. To address these issues, we propose a one-stage network to directly learn the mapping between partial LR observations and HR video frames and to achieve fast and accurate STVSR.

6.3 Space-Time Video Super-Resolution

In this section, we first give an overview of the proposed framework in Sec. 6.3.1. Built upon this framework, we then propose a novel frame feature temporal interpolation network in Sec. 6.3.2; deformable ConvLSTM in Sec. 6.3.3; frame reconstruction module in Sec. 6.3.4; guided feature interpolation learning in Sec. 6.3.5. Finally, we provide details about our implementation in Sec. 6.3.6.

6.3.1 Overview

Given an LR, LFR video sequence: $\mathcal{I}^L = \{I^L_{2t-1}\}_{t=1}^{n+1}$, our goal is to generate the corresponding high-resolution slow-motion video sequence: $\mathcal{I}^H = \{I^H_t\}_{t=1}^{2n+1}$. To intermediate HR frames $\{I^H_{2t}\}_{t=1}^{n}$, there are no corresponding LR counterparts in the input sequence. To fast and accurately increase resolution in both space and time domains, we propose

a one-stage space-time super-resolution framework: **Zooming Slow-Mo** as illustrated in Figure 6.1. The framework mainly consists of five parts: *feature extractor*, *frame feature temporal interpolation module*, *deformable ConvLSTM*, *HR frame reconstructor*, and *guided feature interpolation learning module*.

We first use a feature extractor with a convolutional layer and k_1 residual blocks to extract feature maps: $\{F^L_{2t-1}\}^{n+1}_{t=1}$ from input video frames. Taking the feature maps as input, we then synthesize the LR feature maps: $\{F^L_{2t}\}^{n}_{t=1}$ of intermediate frames with the proposed frame feature interpolation module. Furthermore, to better leverage temporal information, we use a deformable ConvLSTM to process the consecutive feature maps: $\{F^L_t\}^{2n+1}_{t=1}$. Unlike vanilla ConvLSTM, the proposed deformable ConvLSTM can simultaneously perform temporal alignment and aggregation. Finally, we reconstruct the HR slow-mo video sequence from the aggregated feature maps. Since the features for reconstructing intermediate frames are synthesized, there will be feature synthesis errors that will be propagated into restored HR frames, making the predicted video sequence suffer from jitters. To alleviate this problem, we further propose a guided feature interpolation learning mechanism.

6.3.2 Frame Feature Temporal Interpolation

Given extracted feature maps: F^L_1 and F^L_3 from input LR video frames: I^L_1 and I^L_3, we want to synthesize the feature map F^L_2 corresponding to the missing intermediate LR frame I^L_2. Traditional video frame interpolation networks usually perform temporal interpolation on pixel-wise video frames, which will lead to a two-stage STVSR design. Unlike previous methods, we propose to learn a feature temporal interpolation function $f(\cdot)$ to directly synthesize the intermediate feature map F^L_2 (see Fig. 6.2). A general form of the interpolation function can be formulated as:

$$F^L_2 = f(F^L_1, F^L_3) = H(T_1(F^L_1, \Phi_1), T_3(F^L_3, \Phi_3)) \ , \tag{6.1}$$

where $T_1(\cdot)$ and $T_3(\cdot)$ are two sampling functions and Φ_1 and Φ_3 are the corresponding sampling parameters; $H(\cdot)$ is a blending function to aggregate sampled features.

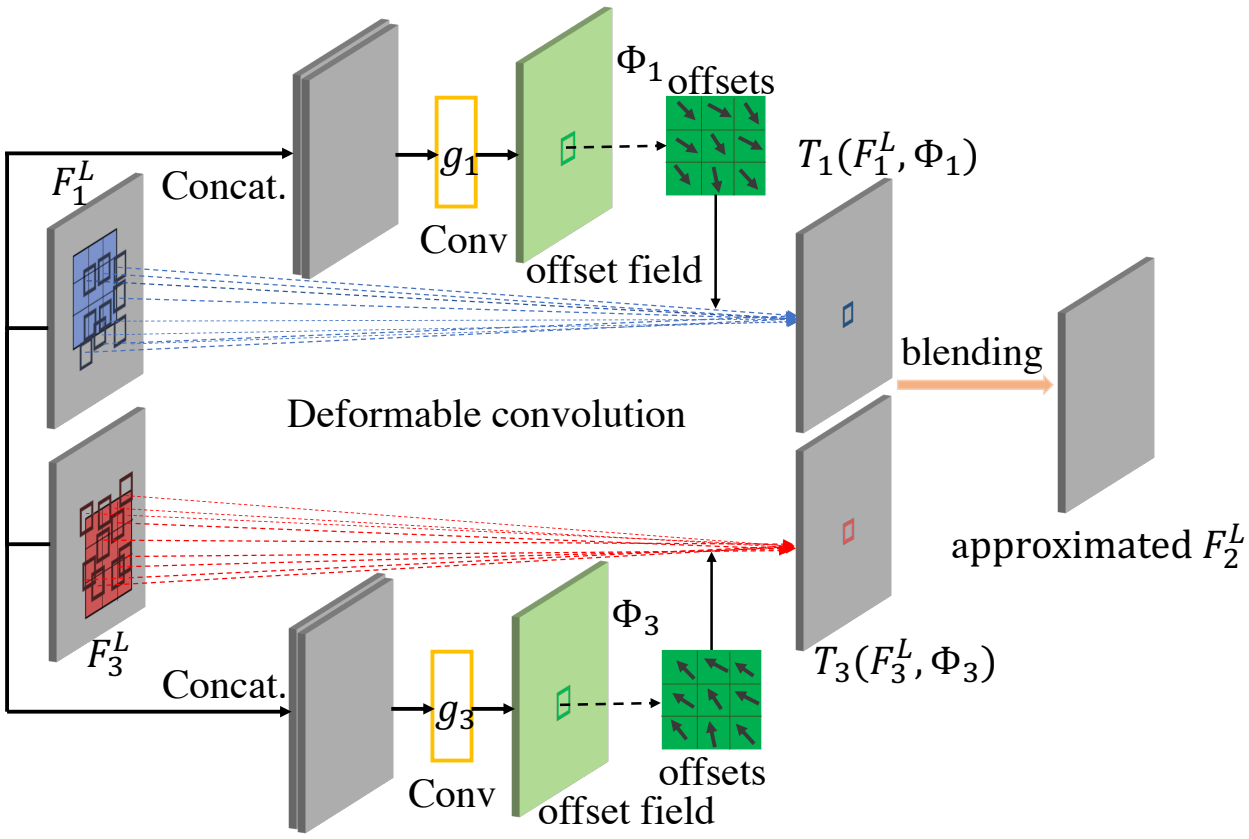


Figure 6.2. Frame feature temporal interpolation based on deformable sampling. Since the approximated F_2^L will be used to predict the corresponding HR frame, it will implicitly enforce the learnable offsets to capture accurate local temporal contexts and be motion-aware.

For generating accurate F_2^L, the $T_1(\cdot)$ should capture forward motion information between F_1^L and F_2^L, and the $T_3(\cdot)$ should capture backward motion information between F_3^L and F_2^L. However, the F_2^L is not available for computing forward and backward motion information in this task.

To alleviate this problem, we use motion information between F_1^L and F_3^L to approximate forward and backward motion information. Inspired by recent deformable alignment in [217] for VSR, we propose to use deformable sampling functions to implicitly capture motion information for frame feature temporal interpolation. While exploring rich local temporal contexts by deformable convolutions in sampling functions, our feature temporal interpolation can even handle very large motions in videos.

The two sampling functions share the same network design but have different weights. For simplicity, we use the $T_1(\cdot)$ as an example. It takes LR frame feature maps F_1^L and F_3^L as input to predict an offset for sampling the F_1^L:

$$\Delta p_1 = g_1([F_1^L, F_3^L]) \ , \tag{6.2}$$

where Δp_1 is a learnable offset and also refers to the sampling parameter: Φ_1; g_1 denotes a general function of several convolution layers; $[,]$ denotes the channel-wise concatenation. With the learned offset, the sampling function can be performed with a deformable convolution [175], [176]:

$$T_1(F_1^L, \Phi_1) = DConv(F_1^L, \Delta p_1) \ . \tag{6.3}$$

Similarly, we can learn an offset $\Delta p_3 = g_3([F_3^L, F_1^L])$ as the sampling parameter: Φ_3 and then obtain sampled features $T_3(F_3^L, \Phi_3)$ with a deformable convolution.

To blend the two sampled features, we use a simple linear blending function:

$$F_2^L = \alpha * T_1(F_1^L, \Phi_1) + \beta * T_3(F_3^L, \Phi_3) \ , \tag{6.4}$$

where α and β are two learnable 1×1 convolution kernels and $*$ is a convolution operator. Since the synthesized LR feature map F_2^L will be used to predict the intermediate HR frame I_2^H, it will enforce the synthesized LR feature map to be close to the real intermediate LR

feature map. Therefore, the two offsets Δp_1 and Δp_3 will implicitly learn to capture the forward and backward motion information, respectively.

Applying the designed deformable temporal interpolation function to $\{F^L_{2t-1}\}^{n+1}_{t=1}$, we can obtain intermediate frame feature maps $\{F^L_{2t}\}^{n}_{t=1}$.

6.3.3 Deformable ConvLSTM

Now we have consecutive frame feature maps: $\{F^L_t\}^{2n+1}_{t=1}$ for generating the corresponding HR video frames, which will be a sequence-to-sequence mapping. It has been proven in previous video restoration tasks [177], [204], [214] that temporal information is vital. Therefore, rather than reconstructing HR frames from the corresponding individual feature maps, we aggregate temporal contexts from neighboring frames. ConvLSTM [218] is a popular 2D sequence data modeling method; and we can adopt it to perform temporal aggregation. At the time step t, the ConvLSTM updates hidden state h_t and cell state c_t with:

$$h_t, c_t = ConvLSTM(h_{t-1}, c_{t-1}, F^L_t) \ . \tag{6.5}$$

From its state updating mechanism [218], we learn that the ConvLSTM can only implicitly capture motions between previous states: h_{t-1} and c_{t-1} and the current input feature map with small convolution receptive fields. Therefore, ConvLSTM has limited ability to handle large motions in natural videos. If a video has large motions, there will be a severe temporal mismatch between previous states and F^L_t. Then, h_{t-1} and c_{t-1} will propagate mismatched "noisy" content rather than useful global temporal contexts into h_t. Consequently, the reconstructed HR frame I^H_t from h_t will suffer from annoying artifacts.

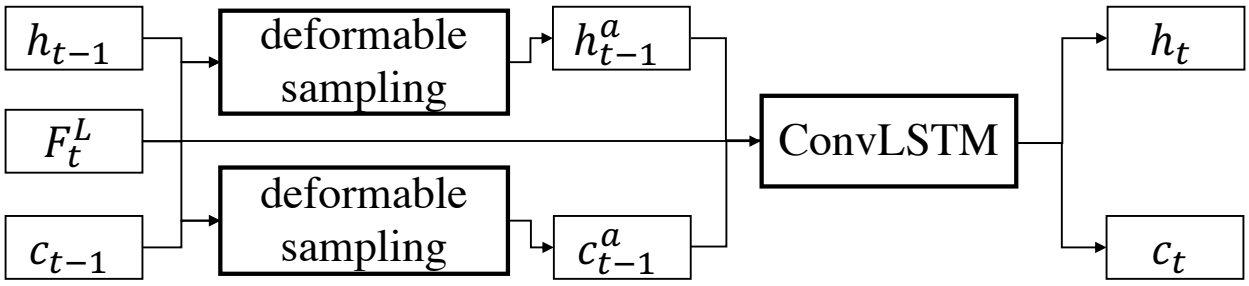


Figure 6.3. Deformable ConvLSTM for better exploiting global temporal contexts and handling fast motion videos. At time step t, we introduce state updating cells to learn deformable sampling to adaptively align hidden state h_{t-1} and cell state c_{t-1} with current input feature map: F^L_t.

To tackle the large motion problem and effectively exploit global temporal contexts, we explicitly embed a state-updating cell with deformable alignment into ConvLSTM (see Fig. 6.3):

$$\begin{aligned}
\Delta p^h_t &= g^h([h_{t-1}, F^L_t]) \ , \\
\Delta p^c_t &= g^c([c_{t-1}, F^L_t]) \ , \\
h^a_{t-1} &= DConv(h_{t-1}, \Delta p^h_t) \ , \\
c^a_{t-1} &= DConv(c_{t-1}, \Delta p^c_t) \ , \\
h_t, c_t &= ConvLSTM(h^a_{t-1}, c^a_{t-1}, F^L_t) \ ,
\end{aligned} \tag{6.6}$$

where g^h and g^c are general functions of several convolution layers, Δp^h_t and Δp^c_t are predicted offsets, and h^a_{t-1} and c^a_{t-1} are aligned hidden and cell states, respectively. Compared with vanilla ConvLSTM, we explicitly enforce the hidden state h_{t-1} and cell state c_{t-1} to align with the current input feature map F^L_t in our deformable ConvLSTM, which makes it more capable of handling motions in videos. Besides, to fully explore temporal information, we use the Deformable ConvLSTM in a bidirectional manner [224]. We feed temporally reversed feature maps into the same Deformable ConvLSTM and concatenate hidden states from forward pass and backward pass as the final hidden state h_t[1] for HR frame reconstruction.

[1]↑We use h_t to denote the final hidden state, but it will refer to a concatenated hidden state in the Bidirectional Deformable ConvLSTM.

6.3.4 Frame Reconstruction

To reconstruct HR video frames, we use a temporally shared synthesis network, which takes the individual hidden state h_t as input and outputs the corresponding HR frame. It has k_2 stacked residual blocks [89] for learning deep features and utilizes a sub-pixel upscaling module with PixelShuffle as in [106] to reconstruct HR frames $\{I_t^H\}_{t=1}^{2n+1}$. To optimize our network, we use a reconstruction loss function:

$$l_{rec} = \sqrt{||I_t^{GT} - I_t^H||^2 + \epsilon^2} \ , \tag{6.7}$$

where I_t^{GT} refers to the t-th ground-truth HR video frame, the Charbonnier penalty function [181] is used as the loss term, and ϵ is empirically set to 1×10^{-3}. Since the space and time SR problems are intra-related in STVSR, our model is end-to-end trainable and can simultaneously learn this spatio-temporal interpolation with only supervision from HR video frames.

6.3.5 Guided Feature Interpolation Learning

In addition, we employ a cyclic interpolation loss to guide the learning of frame feature interpolation with LR frames. It utilizes the inherent temporal coherence in natural video (see Figure 6.4).

Given a sequence of LR, LFR inputs $\{I_{2t-1}^L\}_{t=1}^{n+1}$, we can obtain the extracted frame feature maps $\{F_{2t-1}^L\}_{t=1}^{n+1}$, and the interpolated intermediate frame feature maps $\{F_{2t}^L\}_{t=1}^{n}$. During the training phase, we have a a set of LR ground truth $\{I_t^{LGT}\}_{t=1}^{2n+1}$. The first-order interpolation loss is defined as:

$$l_{\mathrm{i}}^1 = ||I_{2t}^{LGT} - \rho(F_{2t}^L)||_c, \tag{6.8}$$

where $||\cdot||_c$ stands for the Charbonnier penalty function as defined in Equation (6.3.4), and ρ represents the LR synthesis module that turns feature maps into the corresponding LR

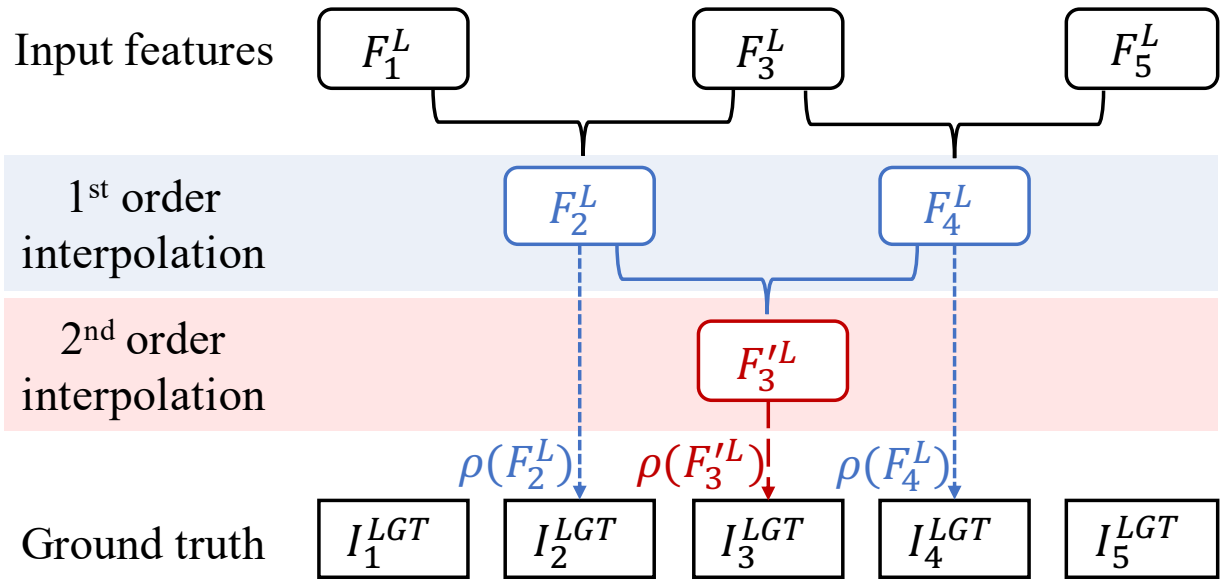


Figure 6.4. Feature interpolation learning guided by LR frames. The cyclic interpolation loss is computed between the ground truth LR frames and the 1st-order and 2nd-order interpolated LR frames. By minimizing the difference of LR frames and their corresponding interpolation acquired at each order, our temporal interpolation module can be self-supervised with the natural temporal coherence.

frames. We apply k_3 stacked residual blocks [89] and a convolution layer, which is similar to the design in Section 6.3.4, to predict the LR frames.

If we conduct feature interpolation on the acquired intermediate frame feature maps $\{F_{2t}^L\}_{t=1}^n$, we can get a sequence of re-interpolated feature maps $\{F_{2t+1}^L\}_{t=1}^{n-1}$. Similar to Equation (6.8), the second-order cyclic interpolation loss is defined as:

$$l_{\mathrm{i}}^2 = ||I_{2t+1}^{LGT} - \rho(F_{2t+1}^L)||_c. \tag{6.9}$$

The overall training loss is the weighted summation of the reconstruction loss, and the 1st- and 2nd-order cyclic interpolation losses:

$$L = \lambda_1 l_{rec} + \lambda_2 l_{\mathrm{i}}^1 + \lambda_3 l_{\mathrm{i}}^2. \tag{6.10}$$

6.3.6 Implementation Details

In our implementation, $k_1 = 5$ and $k_2 = 40$, and $k_3 = 5$ residual blocks are used in the feature extraction and HR frame reconstruction modules, respectively. We randomly crop a sequence of down-sampled image patches with the size of 32×32 and take out the odd-indexed 4 frames as LFR and LR inputs, and the corresponding consecutive 7-frame sequence of $4\times$[2] size as supervision. During the inference stage, we also take 4 frames as input and synthesize a 7-frame sequence. Besides, we perform data augmentation by randomly rotating 90°, 180° and 270°, and horizontal-flipping. We adopt a Pyramid, Cascading and Deformable (PCD) structure in [177] to employ deformable alignment and apply the Adam [127] optimizer, where we decay the learning rate with a cosine annealing for each batch [184] from $4\mathrm{e}-4$ to $1\mathrm{e}-7$. The batch size is set to be 24 and trained on 2 Nvidia Titan XP GPUs.

[2]↑Considering that recent state-of-the-art methods (*e.g.*, EDVR [177] and RBPN [207]) use only 4 as the upscaling factor, we adopt the same practice.

Network Architecture

We further illustrate the feature temporal interpolation network in Figure 6.5 and the proposed STVSR framework in Figure 6.6 to help readers better understand the overall structure of our proposed network.

To make this chapter be concise and easy to follow, we use a simple version of deformable sampling to introduce the proposed feature temporal interpolation and deformable ConvLSTM. However, in our implementation, as stated in Section 3.4 of this chapter, we adopt a Pyramid, Cascading and Deformable (PCD) structure as in [177] to implement the deformable sampling, which can exploit multi-scale contexts with a feature pyramid. The official PyTorch implementation of the PCD can be found in https://github.com/xinntao/EDVR.

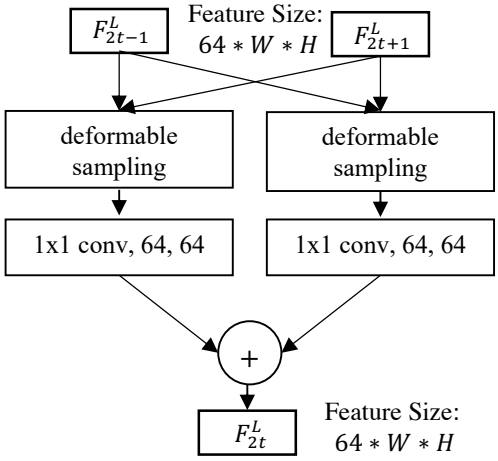


Figure 6.5. Feature temporal interpolation for intermediate LR frames. It will predict an intermediate LR frame feature map F^L_{2t} from two neighboring feature maps: F^L_{2t-1} and F^L_{2t+1}, where $t = 1, 2, ..., n$. Note that the deformable sampling module on the left samples features from F^L_{2t-1} with generated sampling parameters from both F^L_{2t-1} and F^L_{2t+1}; on the contrary, the deformable sampling module on the right samples features from F^L_{2t+1}.

6.4 Experiments and Analysis

6.4.1 Experimental Setup

Datasets Vimeo-90K is used as the training set [204], including over 60,000 7-frame training video sequences. Vimeo-90K is widely used in previous VFI and VSR works [177], [205], [207],

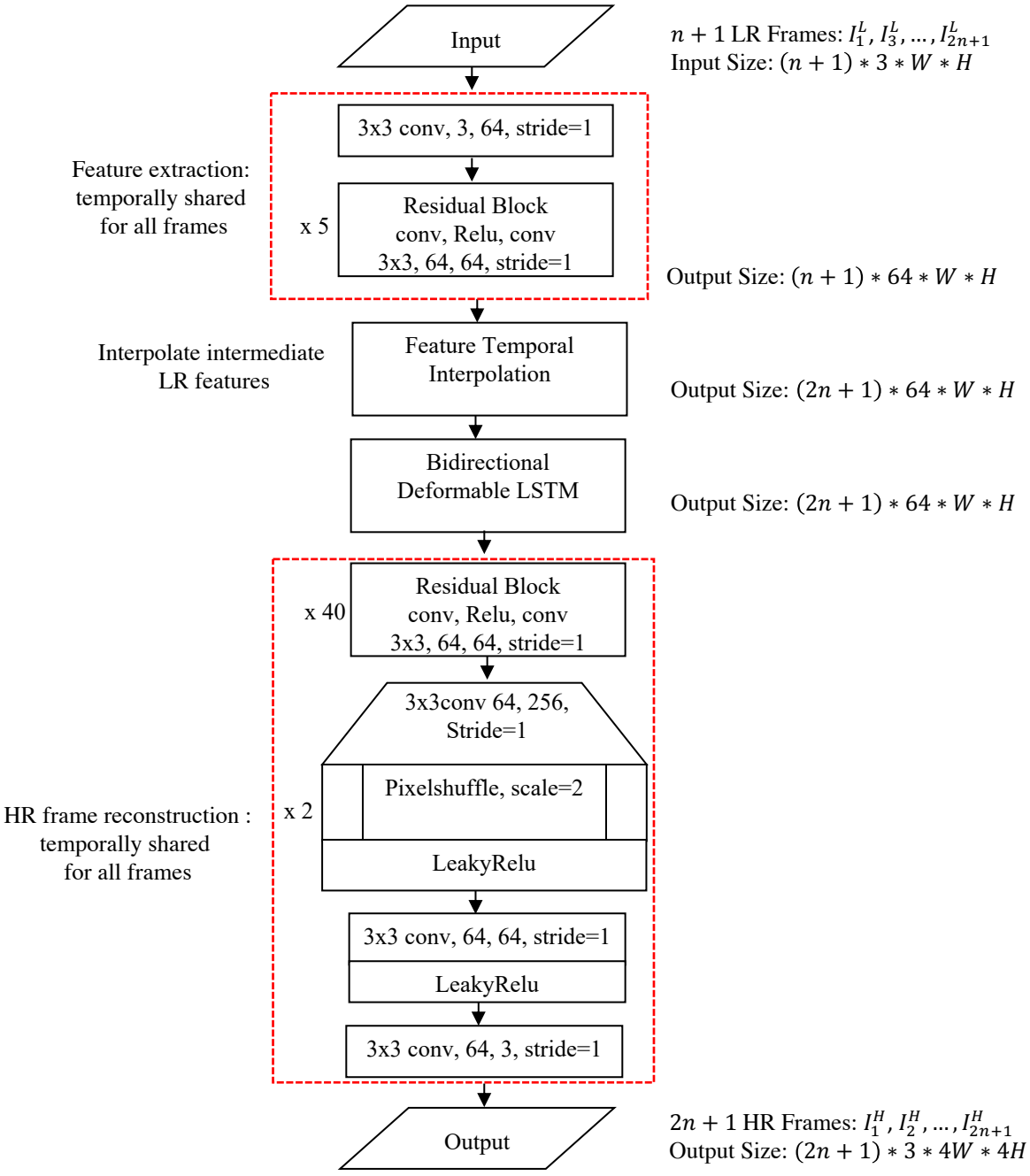


Figure 6.6. Flowchart of the proposed one-stage STVSR framework. The feature extraction and HR frame reconstruction networks are temporally shared for all frames, in which different frames are processed independently.

[213], [217]. Besides, Vid4 [208] and Vimeo testset [204] are used as the evaluation datasets. To compare the performance of different methods under different motion conditions, we split the Vimeo testset into fast motion, medium motion, and slow motion sets as in [207], including 1225, 4977 and 1613 video clips, respectively. We remove 5 videos from the original medium motion set and 3 videos from the slow motion set, which include consecutively all-black frames that will lead to infinite values when calculating PSNR. We generate LR frames by bicubic downsampling with factor= 4 and use odd-indexed LR frames as inputs for predicting the corresponding consecutive HR and HFR video frames.

Evaluation Metrics We adopt Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [92] to evaluate STVSR performance. To measure the efficiency of different methods, we also compare the model parameters and inference time on the entire Vid4 [208] dataset using an Nvidia Titan XP GPU.

6.4.2 Space-Time Video Super-resolution

We compare the performance of our one-stage Zooming SlowMo (ZSM) network to other two-stage methods that are composed of state-of-the-art (SOTA) VFI and VSR networks. Three recent SOTA VFI approaches, SepConv[3] [203], Super-SloMo[4] [211], and DAIN[5] [205], are compared. Besides, three SOTA SR models, including a single-image SR model, RCAN[6] [103], and two recent VSR models, RBPN[7] [207] and EDVR[8] [177], are adopted to generate HR frames from both original and interpolated LR frames.

Quantitative results on Vid4 and Vimeo testsets are shown in Tables 6.1 and 6.2. From these tables, we can observe the following facts: (1) DAIN+EDVR is the best performing two-stage method among the compared 12 approaches; (2) the VFI model matters, especially for videos with fast motion. RBPN and EDVR perform much better than RCAN for SR.

[3]↑https://github.com/sniklaus/sepconv-slomo
[4]↑Since there is no official code released, we used an unofficial PyTorch [225] implementation from https://github.com/avinashpaliwal/Super-SloMo.
[5]↑https://github.com/baowenbo/DAIN
[6]↑https://github.com/yulunzhang/RCAN
[7]↑https://github.com/alterzero/RBPN-PyTorch
[8]↑https://github.com/xinntao/EDVR

Table 6.1. Quantitative comparison of two-stage VFI and VSR methods and our results on the Vid4 [208] dataset. The best two results are highlighted in red and blue colors, respectively. We measure the total run time on the entire Vid4 dataset [208]. Note that we omit the baseline methods with Bicubic when comparing in terms of run time.

VFI Method	SR Method	Vid4 PSNR	Vid4 SSIM	Parameters (Million)	Runtime-VFI (s)	Runtime-SR (s)	Total Run time (s)	Average Run time (s/frame)
SuperSloMo [211]	Bicubic	22.84	0.5772	19.8	0.28	-	-	-
	RCAN [103]	23.80	0.6397	19.8+16.0	0.28	68.15	68.43	0.4002
	RBPN [207]	23.76	0.6362	19.8+12.7	0.28	82.62	82.90	0.4848
	EDVR [177]	24.40	0.6706	19.8+20.7	0.28	24.65	24.93	0.1458
SepConv [203]	Bicubic	23.51	0.6273	21.7	2.24	-	-	-
	RCAN [103]	24.92	0.7236	21.7+16.0	2.24	68.15	70.39	0.4116
	RBPN [207]	26.08	0.7751	21.7+12.7	2.24	82.62	84.86	0.4963
	EDVR [177]	25.93	0.7792	21.7+20.7	2.24	24.65	26.89	0.1572
DAIN [205]	Bicubic	23.55	0.6268	24.0	8.23	-	-	-
	RCAN [103]	25.03	0.7261	24.0+16.0	8.23	68.15	76.38	0.4467
	RBPN [207]	25.96	0.7784	24.0+12.7	8.23	82.62	90.85	0.5313
	EDVR [177]	26.12	0.7836	24.0+20.7	8.23	24.65	32.88	0.1923
ZSM (Ours)		26.49	0.8028	11.10	-	-	10.36	0.0606

Table 6.2. Quantitative comparison of our one-stage ZSM and two-stage VFI and VSR methods on the Vimeo-90K [204] testset. The best two results are highlighted in red and blue colors, respectively.

VFI Method	SR Method	Vimeo-Fast PSNR	Vimeo-Fast SSIM	Vimeo-Medium PSNR	Vimeo-Medium SSIM	Vimeo-Slow PSNR	Vimeo-Slow SSIM
SuperSloMo [211]	Bicubic	31.88	0.8793	29.94	0.8477	28.37	0.8102
	RCAN [103]	34.52	0.9076	32.50	0.8884	30.69	0.8624
	RBPN [207]	34.73	0.9108	32.79	0.8930	30.48	0.8584
	EDVR [177]	35.05	0.9136	33.85	0.8967	30.99	0.8673
SepConv [203]	Bicubic	32.27	0.8890	30.61	0.8633	29.04	0.8290
	RCAN [103]	34.97	0.9195	33.59	0.9125	32.13	0.8967
	RBPN [207]	35.07	0.9238	34.09	0.9229	32.77	0.9090
	EDVR [177]	35.23	0.9252	34.22	0.9240	32.96	0.9112
DAIN [205]	Bicubic	32.41	0.8910	30.67	0.8636	29.06	0.8289
	RCAN [103]	35.27	0.9242	33.82	0.9146	32.26	0.8974
	RBPN [207]	35.55	0.9300	34.45	0.9262	32.92	0.9097
	EDVR [177]	35.81	0.9323	34.66	0.9281	33.11	0.9119
ZSM (Ours)		36.96	0.9444	35.56	0.9385	33.50	0.9166

However, when equipped with the more recent SOTA VFI network DAIN, DAIN+RCAN can achieve a comparable or even better performance than SepConv+RBPN and SepConv+EDVR on the Vimeo-Fast testset; (3) the VSR model also matters. For example, equipped with the same VFI network DAIN, EDVR keeps achieving better STVSR performance than other SR methods. Moreover, we can observe that our ZSM outperforms the DAIN+EDVR by 0.19 dB on Vid4, 0.25 dB on Vimeo-Slow, 0.75 dB on Vimeo-Medium, and 1.00 dB on Vimeo-Fast in terms of PSNR. Such significant improvements obtained on fast-motion videos demonstrate that our one-stage approach with simultaneously leveraging local and global temporal contexts can better handle diverse space-time patterns, including challenging large motions, than two-stage methods.

Furthermore, we also compare the efficiency of different networks and show their model sizes and run times in Table 6.1. To synthesize high-quality frames, SOTA VFI and VSR networks usually come with very large frame reconstruction modules. As a result, the composed two-stage SOTA STVSR networks will have a large number of parameters. Our one-stage model has only one frame reconstruction module. Thus, it has many fewer parameters than the SOTA two-stage networks. Table 6.1 shows that our ZSM is more than 4× and 3× smaller than DAIN+EDVR and DAIN+RBPN, respectively. In terms of run time, the smaller model size makes our network more than 8× faster than DAIN+RBPN and 3× faster than DAIN+EDVR. Our method is still more than 2× faster compared to two-stage methods with a fast VFI network: SuperSlowMo. These results can validate the superiority of our one-stage ZSM model in terms of efficiency.

Qualitative results of these different methods are illustrated in Figure 6.7. Our method demonstrates noticeable perceptual improvements over the other compared two-stage methods. Clearly, our proposed network can synthesize visually appealing HR video frames with more accurate structures, more fine details, and fewer blurry artifacts, even for challenging video sequences with large motions. We can also observe that the SOTA VFI methods: SepConv and DAIN fail to handle fast motions. Consequently, the composed two-stage methods tend to generate severe motion blurs in output frames. In the proposed one-stage framework, we simultaneously learn temporal and spatial SR by exploring the intra-relatedness within

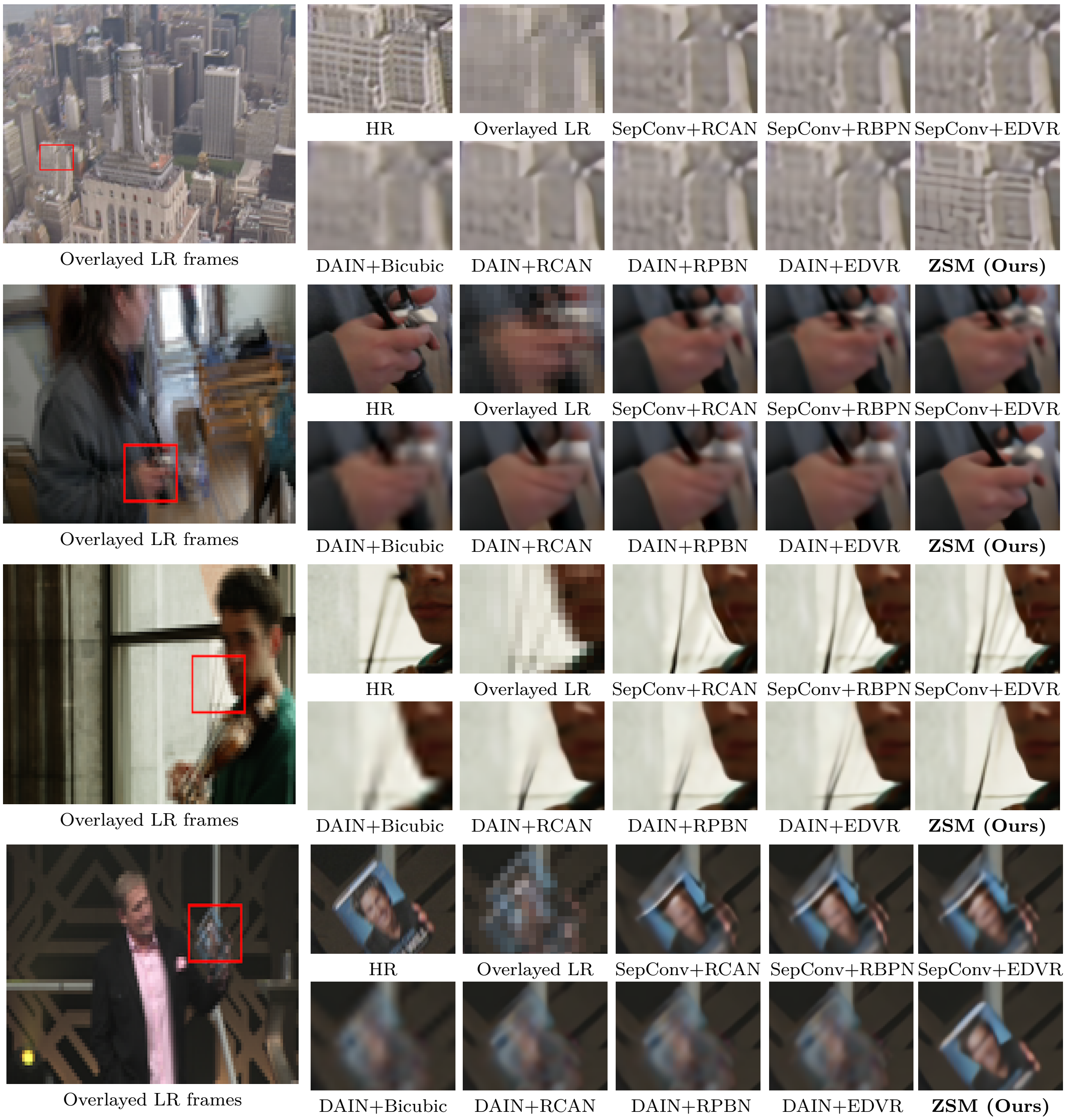


Figure 6.7. Visual comparisons of different methods on Vid4 and Vimeo datasets. Our one-stage Zooming SlowMo model (ZSM) can generate more visually appealing HR video frames with fewer blurring artifacts and more accurate image structures.

natural videos. Thus, our proposed framework: ZSM can well address the large motion issue in temporal SR even with a much smaller model.

6.4.3 Noisy Space-Time Video Super-resolution

Real-world videos are usually compressed and come with complicated noise. To further validate the robustness of the proposed space-time video super-resolution method on noisy data, we add random noise and JPEG compression artifacts into LR video frames [147], respectively.

Random Noise

In our experiments, we train our model on the Vimeo-90K dataset with a mixture of Gaussian noise and 10% salt-and-pepper noise added to the input LR frames as in ToFlow [204]. We compare our method with four models: ToFlow + SepConv + RBPN, Toflow + SepConv + EDVR, Toflow + DAIN + RBPN, and Toflow + DAIN + EDVR on the Vid4, Vimeo-fast, Vimeo-Medium, and Vimeo-Slow datasets. Here, Toflow is used for denoising in the compared methods. The quantitative and qualitative results are shown in Table 6.3 and Figure 6.8, respectively.

From Table 6.3, we can see that our one-stage Zooming SlowMo (ZSM) achieves the best performance among all compared approaches in terms of the four evaluation datasets. Our method outperforms the best three-stage method by 1.11 dB on Vid4, 2.41 dB on Vimeo-Fast, 2.00 dB on Vimeo-Medium, and 1.81 dB on Vimeo-Slow in terms of PSNR. This trend is more obvious in the visual results shown in Figure 6.8. From Figure 6.8, we observe that all three methods can effectively remove severe noises in input LR frames while the HR frames reconstructed by our model have more visual details and fewer artifacts. The results demonstrate that our one-stage network is capable of handling STVSR even for noisy inputs.

JPEG Compression Artifact

For this setting, we train our model on the Vimeo-90K dataset with JPEG compression of different quality factors (QF = 10, 20, 30, and 40). We compare our model with two com-

Figure 6.8. Visual comparisons of different methods on noisy input video frames. Our one-stage Zooming SlowMo model (ZSM) can effectively restore clean missing HR frames from noisy LR frames.

Table 6.3. Quantitative comparisons of our results and three-stage denoising, VFI and VSR methods on video frames with noise. The best two results are highlighted in red and blue colors, respectively.

DN Method	VFI Method	SR Method	Vid4		Vimeo-Fast		Vimeo-Medium		Vimeo-Slow	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
ToFlow [204]	SepConv [203]	RBPN [207]	22.78	0.5692	29.03	0.8376	28.46	0.8140	27.54	0.7826
	SepConv [203]	EDVR [177]	22.79	0.5692	29.02	0.8370	28.44	0.8131	27.52	0.7819
	DAIN [205]	RBPN [207]	22.79	0.5687	29.08	0.8388	28.50	0.8144	27.56	0.7830
	DAIN [205]	EDVR [177]	22.80	0.5693	29.08	0.8387	28.49	0.8144	27.55	0.7825
	ZSM (Ours)		23.91	0.6514	31.49	0.8446	30.49	0.8594	29.36	0.8321

pression artifact reduction (CAR) methods: DNCNN [102] and RNAN [226] +DAIN+RBPN on Vid4 [208] and three subsets of Vimeo90K's testset. The quantitative results and visual comparisons for each quality factor are shown in Table 6.4 and Figure 6.9, respectively.

It is shown in Table 6.4 that our one-stage Zooming SlowMo (ZSM) outperforms the existing three-stage methods for all QFs significantly, and yields the highest overall PNSR and SSIM across all testsets, especially on Vimeo-fast that has large motions: our method exceeds the second-best result by 1.60 dB for QF=10 test data, 1.54 dB for QF=20, 1.61 dB for QF=30 and 1.76 dB for QF=40. Figure 6.9 provides some qualitative examples. We can assess the quality of the output images from the following perspectives: compression artifact reduction (*e.g.* blocking and ringing artifacts, color shift), and the reconstruction of details. According to Figure 6.9, even though the other methods can reduce the compression artifacts in most cases, they suffer from over-smoothing and lack of details. Compared with them, our output has fewer ringing and blocking artifacts, and sharper edges under different QFs. These results demonstrate that our one-stage framework can reach a better balance between artifact reduction and high-fidelity reconstruction. It is worth noting that our method is capable of restoring the color aberrations in the images compressed by low quality factors (see the top 2 rows of Figure 6.9).

6.4.4 Ablation Study

In previous sections, we have already illustrated the superiority of our proposed one-stage framework. In this section, we make a comprehensive ablation study to further demonstrate the effectiveness of different modules in our network.

Effectiveness of Deformable ConvLSTM

To investigate the proposed Deformable ConvLSTM (DConvLSTM) module, we take four different models for comparison: (b), (c), (d), and (e), where (c) adds a vanilla ConvLSTM into (b), (d) adds the proposed DConvLSTM, and (e) utilizes the DConvLSTM module in a bidirectional manner.

Table 6.4. Quantitative comparison of our results and three-stage CAR, VFI and VSR methods on compressed testsets (QF = 10, 20, 30, and 40). The best two results for each QF are highlighted in red and blue colors, respectively.

QF	CAR Method	VFI Method	SR Method	Vid4		Vimeo-Fast		Vimeo-Medium		Vimeo-Slow	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
10	DNCNN [102]	SepConv [203]	RBPN [207]	21.33	0.4659	27.84	0.7917	26.59	0.7475	25.46	0.7010
		SepConv [203]	EDVR [177]	21.31	0.4647	27.83	0.7909	26.58	0.7466	25.45	0.7001
		DAIN [205]	RBPN [207]	21.37	0.4688	27.92	0.7958	26.65	0.7510	25.49	0.7028
		DAIN [205]	EDVR [177]	21.35	0.4679	27.92	0.7955	26.64	0.7504	25.48	0.7020
	RNAN [226]	SepConv [203]	RBPN [207]	20.41	0.4864	23.80	0.7926	23.61	0.7607	22.83	0.7171
		SepConv [203]	EDVR [177]	20.40	0.4858	23.80	0.7925	23.60	0.7604	22.82	0.7165
		DAIN [205]	RBPN [207]	20.41	0.4859	23.81	0.7936	23.60	0.7610	22.82	0.7172
		DAIN [205]	EDVR [177]	20.41	0.4856	23.81	0.7936	23.59	0.7609	22.81	0.7167
		ZSM (Ours)		22.03	0.5216	29.52	0.8367	28.13	0.8009	26.64	0.7532
20	DNCNN [102]	SepConv [203]	RBPN [207]	22.10	0.5102	29.36	0.8232	28.03	0.7834	26.76	0.7396
		SepConv [203]	EDVR [177]	22.09	0.5089	29.36	0.8225	28.02	0.7825	26.75	0.7386
		DAIN [205]	RBPN [207]	22.14	0.5132	29.46	0.8266	28.10	0.7865	26.79	0.7411
		DAIN [205]	EDVR [177]	22.13	0.5121	29.46	0.8263	28.09	0.7859	26.78	0.7403
	RNAN [226]	SepConv [203]	RBPN [207]	20.91	0.5292	24.93	0.8197	24.76	0.7920	24.01	0.7527
		SepConv [203]	EDVR [177]	20.90	0.5285	24.92	0.8196	24.76	0.7917	24.01	0.7524
		DAIN [205]	RBPN [207]	20.93	0.5293	24.95	0.8213	24.77	0.7927	24.00	0.7530
		DAIN [205]	EDVR [177]	20.93	0.5290	24.95	0.8214	24.76	0.7927	24.00	0.7528
		ZSM (Ours)		22.78	0.5707	31.00	0.8607	29.54	0.8300	27.95	0.7864
30	DNCNN [102]	SepConv [203]	RBPN [207]	22.52	0.5361	30.08	0.8377	28.73	0.8009	27.43	0.7593
		SepConv [203]	EDVR [177]	22.51	0.5354	30.09	0.8372	28.72	0.8000	27.41	0.7583
		DAIN [205]	RBPN [207]	22.57	0.5393	30.19	0.8410	28.81	0.8039	27.46	0.7608
		DAIN [205]	EDVR [177]	22.56	0.5388	30.20	0.8401	28.80	0.8033	27.44	0.7598
	RNAN [226]	SepConv [203]	RBPN [207]	21.41	0.5569	25.54	0.8345	25.35	0.8084	24.64	0.7717
		SepConv [203]	EDVR [177]	21.41	0.5567	25.54	0.8344	25.34	0.8082	24.63	0.7715
		DAIN [205]	RBPN [207]	21.45	0.5576	25.58	0.8365	25.36	0.8095	24.63	0.7721
		DAIN [205]	EDVR [177]	21.44	0.5577	25.58	0.8367	25.36	0.8095	24.63	0.7720
		ZSM (Ours)		23.25	0.6013	31.81	0.8733	30.30	0.8082	28.65	0.8042
40	DNCNN [102]	SepConv [203]	RBPN [207]	22.83	0.5559	30.51	0.8464	29.17	0.8120	27.84	0.7720
		SepConv [203]	EDVR [177]	22.83	0.5560	30.51	0.8458	29.16	0.8110	27.83	0.7710
		DAIN [205]	RBPN [207]	22.88	0.5590	30.63	0.8497	29.26	0.8149	27.88	0.7734
		DAIN [205]	EDVR [177]	22.88	0.5592	30.64	0.8496	29.25	0.8144	27.86	0.7725
	RNAN [226]	SepConv [203]	RBPN [207]	21.63	0.5764	25.87	0.8426	25.71	0.8190	25.01	0.7837
		SepConv [203]	EDVR [177]	21.63	0.5764	25.87	0.8425	25.70	0.8187	25.01	0.7836
		DAIN [205]	RBPN [207]	21.67	0.5774	25.92	0.8449	25.73	0.8202	25.01	0.7842
		DAIN [205]	EDVR [177]	21.67	0.5779	25.92	0.8451	25.73	0.8203	25.01	0.7842
		ZSM (Ours)		23.59	0.6226	32.40	0.8818	30.85	0.8560	29.15	0.8164

Figure 6.9. Visual comparisons of different methods on compressed LR frames. The first, second, third, and fourth rows are results for QR = 10, 20, 30, and 40, respectively.

Table 6.5. Ablation study on the proposed modules in our ZSM method. While ConvLSTM performs worse for fast-motion videos, our deformable feature interpolation and deformable ConvLSTM can effectively handle motions and improve overall STVSR performance.

Method	(a)	(b)	(c)	(d)	(e)	(f)
Naive feature interpolation	√					
DFI		√	√	√	√	√
ConvLSTM			√			
DConvLSTM				√		
Bidirectional DConvLSTM					√	√
GFI						√
Vid4 (slow motion)	25.18	25.34	25.68	26.18	26.31	26.49
Vimeo-Fast (fast motion)	34.93	35.66	35.39	36.56	36.81	36.96

From Table 6.5, we can see that (c) outperforms (b) on Vid4 with slow motion while performing worse than (b) on Vimeo-Fast with fast motion videos. The results indicate that vanilla ConvLSTM can utilize global temporal contexts for slow motion videos, but cannot handle sequences with large motion. Furthermore, we observe that model (d) is significantly better than both (b) and (c), which demonstrates that our DConvLSTM can learn the temporal alignment between previous states and the current feature map. Therefore, it can better exploit global contexts for reconstructing visually appealing frames with more vivid details. Our findings are supported by qualitative results in Figure 6.10.

In addition, we verify the bidirectional mechanism in DConvLSTM by comparing (e) and (d) in Table 6.5 and Figure 6.11. From Table 6.5, we observe that (e) with bidirectional DConvLSTM can further improve STVSR performance over (d) on both slow motion and fast motion testing sets. The visual results in Figure 6.11 show that our full model with a bidirectional mechanism can restore more details by making better use of global temporal information for all input frames.

Effectiveness of Deformable Feature Interpolation

To validate our deformable feature interpolation (DFI) module, we introduce two baselines for comparison: (a) and (b), where the model (a) only uses convolutions to blend LR features instead of deformable sampling functions as in model (b). Note that neither model (a) nor model (b) has ConvLSTM or DConvLSTM.

From Table 6.5, we observe that (b) outperforms (a) by 0.16 dB on Vid4 and 0.73 dB on the Vimeo-Fast dataset with fast motions in terms of PSNR. Figure 6.12 shows a qualitative comparison, where we can see that model (a) generates a face with severe motion blur, while the proposed deformable feature interpolation module can effectively address the large motion issue by exploiting local temporal contexts and help the model (b) generate clearer face structures and details. The superiority of the proposed DFI module demonstrates that the learned offsets in the deformable sampling functions can better exploit local temporal contexts and capture forward and backward motions in natural video sequences, even without any explicit supervision.

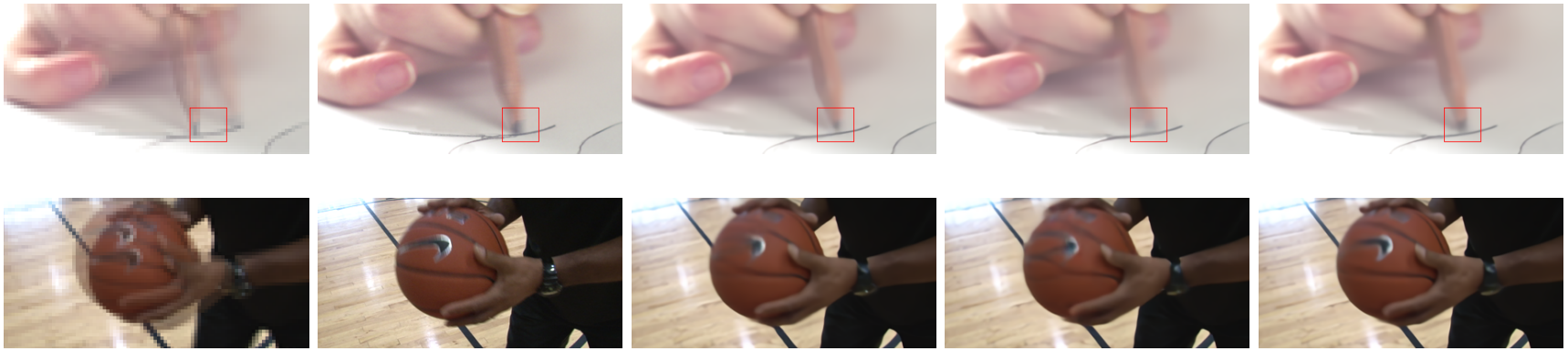


Figure 6.10. Ablation study on Deformable ConvLSTM (DConvLSTM). Vanilla ConvLSTM will fail on videos with fast motions. Embedded with state updating cells, the proposed DConvLSTM is more capable of leveraging global temporal contexts for reconstructing more accurate content, even for fast-motion videos. The red box in each image in the upper row is intended to highlight the reproduction of a particular detail.

HR w/o bidirectional w/ bidirectional

Figure 6.11. Ablation study on the bidirectional mechanism in DConvLSTM. By adding the bidirectional mechanism into DConvLSTM, our model can utilize both previous and future contexts, and therefore can reconstruct more visually appealing frames with finer details, especially for video frames at the first step, which cannot access any temporal information from preceding frames.

Overlayed LR HR

w/o DFI@model (a) w/ DFI@model (b)

Figure 6.12. Ablation study on feature interpolation. The naive feature interpolation model without deformable sampling will obtain overly smooth results for videos with fast motions. With the proposed deformable feature interpolation (DFI), our model can well exploit local contexts in adjacent frames, thus is more effective in handling large motions.

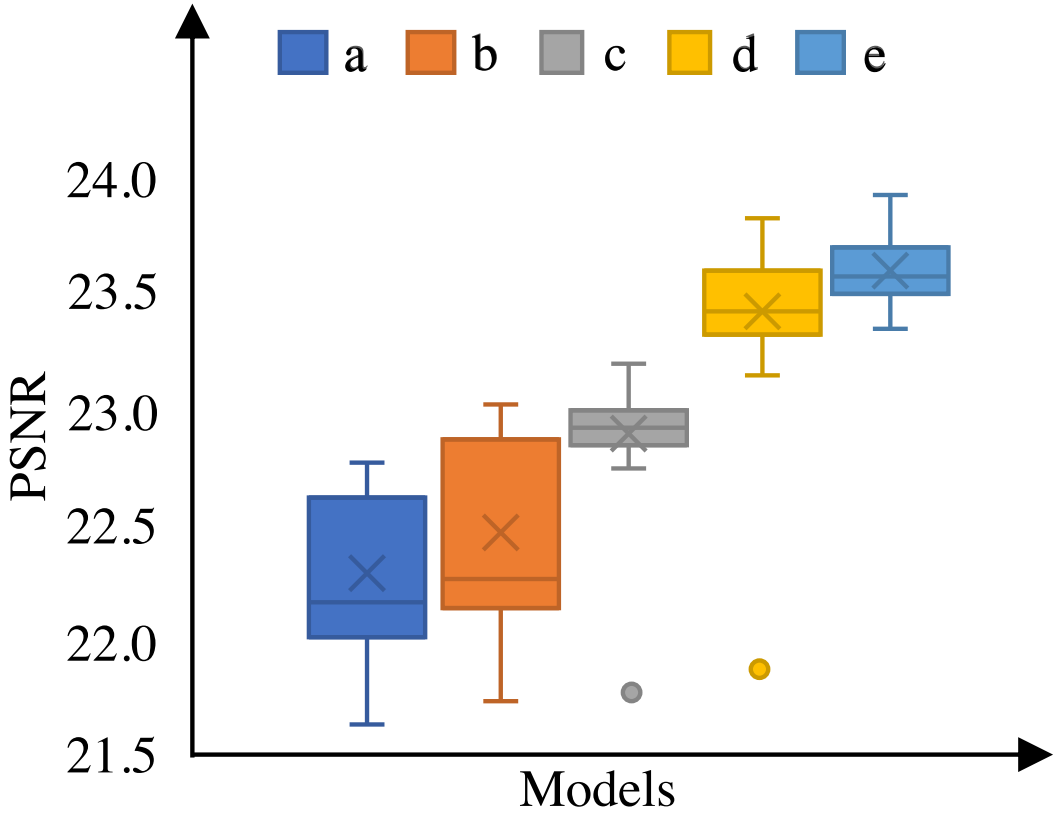


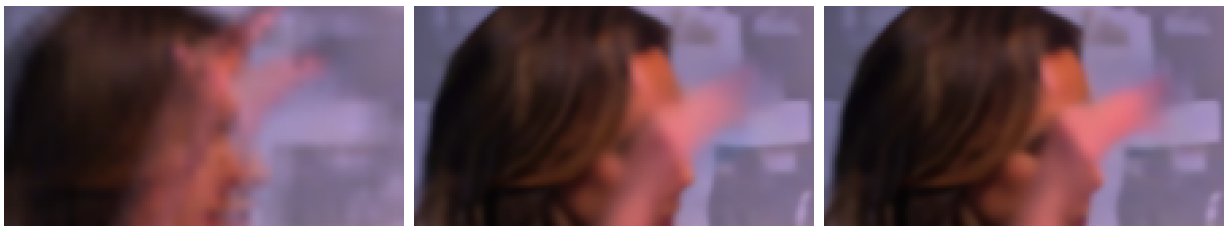
Figure 6.13. Statistics of different models computed based on 31 frames from the Vid4's Calendar sequence. The box-whisker plot reflects the distribution of PSNR for each model by five numbers: minimum, first quartile, median, third quartile, and maximum. The box is drawn from the first quartile to the third quartile, a horizontal line goes through the box at the median, and the whiskers go from each quartile to the minimum or maximum. The "X" is the average. At the first time step, ConvLSTM cannot leverage temporal information, so the results for the first frame from models with ConvLSTM are much worse than other frames. These outliers are plotted as dots. As shown in the plot, the average and median of PSNR become higher from model *a* to e.

In Figure 6.13 it is obvious to see that the entire distribution of PSNR moves up, which indicates that deformable ConvLSTM actually brings an improvement on both existent and synthetic frames. Comparing models d and e in Figure 6.13, we can observe that the introduction of the bi-directional mechanism solves the problem of outlier frames (the first frame of the sequence), and further enhances the quality of all frames.

Guided Feature Interpolation Critic

We first validate the strength of the guided feature interpolation (GFI) learning on the STVSR task by comparing the model trained with it (model f) to the model without its supervision (model e) in Table 6.5. Since the final performance is evaluated on the overall video sequence, optimizing the intermediate reward provided by LR frames can improve temporal consistency across frames. Visual results illustrated in Figure 6.14 can further demonstrate the superiority of the proposed GFI.

Furthermore, we demonstrate the influence of the guided feature interpolation on noisy STVSR in Table 6.6. We compare PSNR and SSIM of the Y channel of models trained with/without the guided loss on the Vid4 dataset [208]. For fair comparisons, both models are trained for the same number of iterations. We observe that the intermediate features guide actually decreases the performance in most cases when the input data is noisy.



Overlayed LR | w/o guidance | w/ guidance

Figure 6.14. Ablation study on guided feature interpolation module. The additional guidance can help to strengthen the ability of the temporal feature interpolation network on handling motions.

Table 6.6. Ablation study of guided feature learning on noisy STVSR. We compare the PSNR and SSIM of the Y channel of models trained with/without the guided loss on the Vid4 dataset [208].

Type		w/o GFI		w/GFI	
		PSNR	SSIM	PSNR	SSIM
Noise		23.91	0.6514	23.89	0.6510
Compression	QF=10	22.03	0.5216	21.99	0.5204
	QF=20	22.78	0.5696	22.78	0.5707
	QF=30	23.25	0.6000	23.25	0.6013
	QF=40	23.59	0.6226	23.55	0.6220

6.5 Discussion

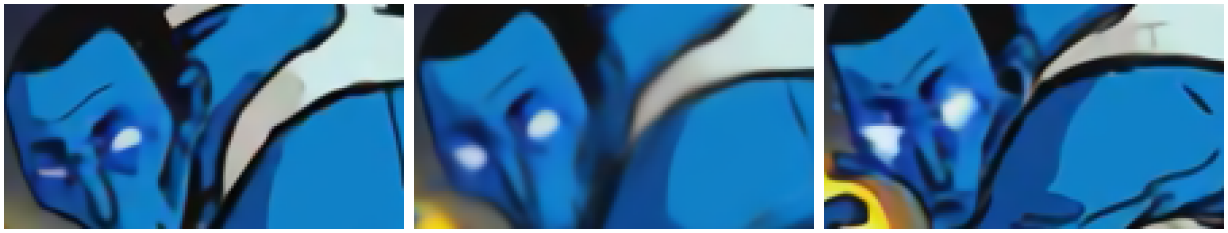
For space-time video super-resolution, the goal is to reconstruct HR frames for both missing intermediate and input LR frames. For example, given four input LR frames: $\{I_1^L, I_3^L, I_5^L, I_7^L\}$, we want to obtain the corresponding seven consecutive HR frames: $\{I_1^H, I_2^H, ..., I_7^H\}$. Since I_2^L, I_4^L, and I_6^L are unavailable, we need to hallucinate missing information for the frames. Therefore, it is more challenging to reconstruct $\{I_2^H, I_4^H, I_6^H\}$ than $\{I_1^H, I_3^H, I_5^H, I_7^H\}$. The unbalanced difficulty can lead to temporally inconsistent video results; and the potential video jittering becomes one of the most crucial issues that we need to consider when designing a space-time video super-resolution method. In our current framework, the proposed deformable ConvLSTM can implicitly enforce temporal coherence by handling visual motions and aggregating temporal contexts. However, results from our model still suffer from the temporal inconsistency issue due to the essential unbalanced difficulty (see Figure 6.15). To further alleviate the problem, we could consider devising new temporal consistency constraints to explicitly guide HR video frame reconstruction.

Videos might contain dramatically changing scenes or objects due to the existence of temporal motions and geometric deformations. Although our Zooming SlowMo is capable of handling large-motion videos, it might fail for certain dynamic objects with severe deformations. Multiple parts in a video object across temporal frames might have different motion patterns, which leads to complex object deformations and introduces additional difficulties for space-time video super-resolution. We illustrate one failure example in Figure 6.16. Although the synthesized frame from our model contains fewer artifacts, the reconstructed *hands* are pretty blurry due to large deformations in the animated character. To mitigate the issue, we desire a more deformation-robust temporal alignment approach. We believe it would be a promising direction.

6.6 Conclusion

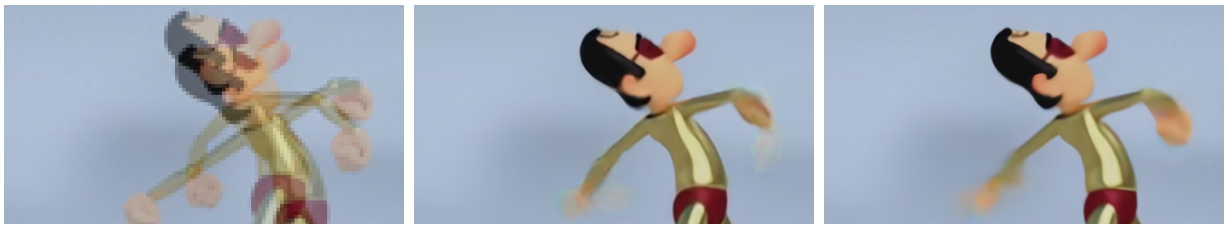
In this chapter, we propose a one-stage framework for space-time video super-resolution to directly reconstruct high-resolution and high frame rate videos without synthesizing intermediate low-resolution frames. To achieve this, we introduce a deformable feature interpola-

DAIN+EDVR: $t-1$ DAIN+EDVR: t DAIN+EDVR: $t+1$



ZSM: $t-1$ ZSM: t ZSM: $t+1$

Figure 6.15. Temporal inconsistency issue in STVSR. It is more difficult to synthesis HR frame t than HR frames: $t-1$ and $t+1$, since LR frames: $t-1$ and $t+1$ are available and LR frame t is missing during testing. Synthesized HR frame at the time step t is more blurry with fewer visual details than results at $t-1$ and $t+1$.



Overlayed LR DAIN+EDVR ZSM

Figure 6.16. Failure example. Our model might fail to handle dynamic video objects with severe geometric deformations.

tion network for feature-level temporal interpolation. Furthermore, we propose a deformable ConvLSTM for aggregating temporal information and handling motions. With such a one-stage design, our network can well explore intra-relatedness between temporal interpolation and spatial super-resolution in the task. It enforces our model to adaptively learn to leverage useful local and global temporal contexts for alleviating large motion issues. Extensive experiments show that our one-stage framework is more effective, yet more efficient than existing two-stage networks, and the proposed feature temporal interpolation network and deformable ConvLSTM are capable of handling very challenging fast motion videos.

7. Summary and Contributions

Blockwise Based Detection of Local Defects

In Chapter 1, we present a coarse-to-fine method to automatically detect local defects. This method includes the initial detection by thresholding, and the secondary refinement by a trained model. Our major contribution include:

- Different from previous works, we propose blockwise features to describe attributes of visible defects in the candidate area, which can help us to determine the exact defect type.
- With these proposed features, we build a blockwise dataset of local defects for future training.
- A decision tree model is applied to produce more accurate results for visible local defects. Our algorithm can classify different local defects according to their perceptual attributes, including size, brightness, and other aspects. Finally, we agglomerate blockwise results to generate a feature vector for each test page, which can be used for further assignment of page rank.

Face Set Recognition

In Chapter 2, we present a multi-column network for face set representation and recognition. It takes all the images in the set as input and extracts their features and weights based on the quality and content factors, and aggregates them to a compact representation of the face set. Our contribution include:

- We propose an efficient solution for face set recognition. This method only adds an extra $\approx 6k$ parameters to the face feature extraction model, and can be widely used in a wide variety of scenarios including video surveillance, mobile payment, and other tasks.

Face Alignment

In Chapter 3, we investigate the effect of noise on the facial landmark detection task by modeling the noise in both the training set and detection output, and comparing models trained with different noise and training strategies. Our contribution include:

- We propose two metrics that quantitatively measure the stability of detected facial landmarks.
- We model the annotation noise in an existing public dataset.
- We investigate the influence of different types of noise in training face alignment neural networks, and propose corresponding solutions.
- Our results demonstrate improvements in both the accuracy and the stability of detected facial landmarks. Our further experiments suggest that noise injection could be a good method to avoid over-fitting.

Super-Resolution on Compressed Images

In Chapter 4, we propose a single-stage network for the joint compression artifact reduction task to directly reconstruct an artifact-free high-resolution image from a compressed low-resolution input. To summarize, our contributions are mainly three-fold:

- We propose a novel CAJNN framework that jointly solves the CAR and SR problems for real-world images that are from unknown devices with unknown quality factors. Here, we explore ways to represent and combine both local and non-local information to enforce image reconstruction performance without knowing the input quality factor.
- Our experiments show that CAJNN achieves the new state-of-the-art performance on multiple datasets as measured by the PSNR and SSIM metrics. Compared with the prior art, it generates more stable and reliable outputs for any level of compression quality factors. Our experiments illustrate the effectiveness and efficiency of our method with both standard test images and real-world images.

- We provide a new idea for enhancing high-level computer vision tasks like real-scene text recognition and extremely tiny face detection: by preprocessing the input data with our pretrained model, we can improve the performance of existing detectors. Our model demonstrates its effectiveness on the WIDER face dataset, and the ICDAR2013 Focused Scene Text dataset.

Headshot Image Super-Resolution with Multiple Exemplars

Chapter 5 presents a solution to a more specifc scenario: headshot image super-resolution with multiple exemplars. In summary, our contribution is four-fold:

- We propose a novel headshot super-resolution network that takes advantage of multiple exemplars. Our method is more effective than previous approaches by thoroughly integrating the corresponding information in the exemplar set. It is also more computationally efficient since we from the matching and transferring process in the LR space with careful design.
- We propose a novel reference feature alignment network to search and align corresponding reference features to the LR content based on deformable sampling. We devise a feature aggregation module conditioned on the LR content to explicitly improve the set representation by favoring features that are high in quality and similarity to the LR content.
- We propose a novel correlation loss that helps to represent the local texture and reconstruct more realistic details.
- Compared with previous approaches, our method achieves state-of-the-art face hallucination performance on the CelebAMask-HQ testset. It has significantly fewer parameters and computational costs than recent exemplar-guided methods.

In future works, we will explore other aggregation methods to generate a better set representation with the aid of face priors. In addition, we will further validate the effectiveness of the correlation loss as generic supervision for other low-level tasks, *e.g.* image denoising, video frame interpolation, style transfer, *etc.*.

Space-Time Video Super-Resolution

In Chapter 6, we propose a one-stage framework for space-time video super-resolution to directly reconstruct high-resolution and high frame rate videos without synthesizing intermediate low-resolution frames. The contributions of this chapter are six-fold:

- We propose a one-stage space-time super-resolution network that can address temporal interpolation and spatial SR simultaneously in a unified framework. Our one-stage method is more effective than two-stage methods by taking advantage of the intra-relatedness between the two sub-problems. It is also computationally more efficient since only one frame reconstruction network is required rather than two large networks as in state-of-the-art two-stage approaches.

- We propose a frame feature temporal interpolation network leveraging local temporal contexts based on deformable sampling for intermediate LR frames. We devise a novel deformable ConvLSTM to explicitly enhance temporal alignment capacity and exploit global temporal contexts for handling large motions in videos.

- Our one-stage method achieves state-of-the-art STVSR performance on both the Vid4 and Vimeo datasets. It is 3 times faster than the two-stage SOTA networks while having a nearly 4× reduction in model size.

- We improve model performance by integrating guided feature interpolation learning into our one-stage framework.

- We investigate space-time video super-resolution under even noisy conditions, in which random noises or JPEG compression artifacts corrupt the input LR video frames. Such applications allow us to explore the flexibility and potential breath of our Zooming SlowMo (ZSM) method.

- Additional and extensive experimental results demonstrate the effectiveness of the proposed guided interpolation learning, and further show the superiority of our one-stage network on tackling more challenging noisy STVSR tasks.

REFERENCES

[1] X. Jing, S. Astling, R. Jessome, E. Maggard, T. Nelson, M. Shaw, and J. P. Allebach, "A general approach for assessment of print quality," in *Image Quality and System Performance X*, International Society for Optics and Photonics, vol. 8653, 2013, p. 86530L. DOI: https://doi.org/10.1117/12.2008819.

[2] Y. Ju, E. Maggard, R. Jessome, and J. Allebach, "Autonomous detection of text fade point with color laser printers," in *Image Quality and System Performance XII*, International Society for Optics and Photonics, vol. 9396, 2015, 93960G. DOI: https://doi.org/10.1117/12.2081238.

[3] N. Yan, E. Maggard, R. Fothergill, R. J. Jessome, and J. P. Allebach, "Autonomous detection of iso fade point with color laser printers," in *Image Quality and System Performance XII*, International Society for Optics and Photonics, vol. 9396, 2015, 93960F. DOI: https://doi.org/10.1117/12.2078324.

[4] Z. Xiao, S. Xu, E. Maggard, K. Morse, M. Shaw, and J. Allebach, "Detection of color fading in printed customer content," in *Color Imaging XXIII: Displaying, Processing, Hardcopy, and Applications*, Society for Imaging Science and Technology, 2018, pp. 297-1–297-6. DOI: https://doi.org/10.2352/issn.2470-1173.2018.16.color-297.

[5] J. Zhang and J. P. Allebach, "Estimation of repetitive interval of periodic bands in laser electrophotographic printer output," in *Image Quality and System Performance XII*, International Society for Optics and Photonics, vol. 9396, 2015, 93960J. DOI: https://doi.org/10.1117/12.2083547.

[6] J. Zhang, S. Astling, R. Jessome, E. Maggard, T. Nelson, M. Shaw, and J. P. Allebach, "Assessment of presence of isolated periodic and aperiodic bands in laser electrophotographic printer output," in *Image Quality and System Performance X*, International Society for Optics and Photonics, vol. 8653, 2013, 86530N. DOI: https://doi.org/10.1117/12.2008818.

[7] M. Q. Nguyen and J. P. Allebach, "Controlling misses and false alarms in a machine learning framework for predicting uniformity of printed pages," in *Image Quality and System Performance XII*, International Society for Optics and Photonics, vol. 9396, 2015, p. 93960I. DOI: https://doi.org/10.1117/12.2083162.

[8] M. Q. Nguyen and J. P. Allebach, "Feature ranking and selection used in a machine learning framework for predicting uniformity of printed pages," in *Image Quality and System Performance XIV*, Society for Imaging Science and Technology, 2017, pp. 166–173. DOI: https://doi.org/10.2352/ISSN.2470-1173.2017.12.IQSP-238.

[9] M. Q. Nguyen, R. Jessome, S. Astling, E. Maggard, T. Nelson, M. Shaw, and J. P. Allebach, "Perceptual metrics and visualization tools for evaluation of page uniformity," in *Image Quality and System Performance XI*, International Society for Optics and Photonics, vol. 9016, 2014, p. 901 608. DOI: https://doi.org/10.1117/12.2038752.

[10] J. Wang, T. Nelson, R. Jessome, S. Astling, E. Maggard, M. Shaw, and J. P. Allebach, "Local defect detection and print quality assessment," in *Image Quality and System Performance XIII*, Society for Imaging Science and Technology, 2016, pp. 1–7. DOI: https://doi.org/10.2352/ISSN.2470-1173.2016.13.IQSP-207.

[11] Z. Runzhe, M. Eric, J. Renee, B. Yousun, C. Minki, and A. Jan, "Detection of streaks on printed pages," in *Image Quality and System Performance XVI*, Society for Imaging Science and Technology, 2019.

[12] H. Wan-Eih, M. Eric, J. Renee, B. Yousun, C. Minki, and A. Jan, "Banding estimation for print quality," in *Image Quality and System Performance XVI*, Society for Imaging Science and Technology, 2019.

[13] M. D. Fairchild, *Color appearance models*. John Wiley & Sons, 2013, p. 340. DOI: https://doi.org/10.1002/9781118653128.

[14] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979. DOI: https://doi.org/10.1109/TSMC.1979.4310076.

[15] H.-F. Ng, "Automatic thresholding for defect detection," *Pattern recognition letters*, vol. 27, no. 14, pp. 1644–1649, 2006. DOI: https://doi.org/10.1109/ICIG.2004.43.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. DOI: https://doi.org/10.1109/cvpr.2016.90.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105. DOI: https://doi.org/10.1145/3065386.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[19] Y. Zhong, J. Chen, and B. Huang, "Toward end-to-end face recognition through alignment learning," *IEEE signal processing letters*, vol. 24, no. 8, pp. 1213–1217, 2017. DOI: https://doi.org/10.1109/lsp.2017.2715076.

[20] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L 2 hypersphere embedding for face verification," in *Proceedings of the 25th ACM International Conference on Multimedia*, ACM, 2017, pp. 1041–1049. DOI: https://doi.org/10.1145/3123266.3123359.

[21] X. Qi and L. Zhang, "Face recognition via centralized coordinate learning," *arXiv preprint arXiv:1801.05678*, 2018.

[22] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699. DOI: https://doi.org/10.1109/cvpr.2019.00482.

[23] Y. Guo and L. Zhang, "One-shot face recognition by promoting underrepresented classes," *arXiv preprint arXiv:1707.05574*, 2017.

[24] R. Ranjan, A. Bansal, H. Xu, S. Sankaranarayanan, J.-C. Chen, C. D. Castillo, and R. Chellappa, "Crystal loss and quality pooling for unconstrained face verification and recognition," *arXiv preprint arXiv:1804.01159*, 2018.

[25] W. Xie and A. Zisserman, "Multicolumn networks for face recognition," *arXiv preprint arXiv:1807.09192*, 2018.

[26] L. Xiong, J. Karlekar, J. Zhao, Y. Cheng, Y. Xu, J. Feng, S. Pranata, and S. Shen, "A good practice towards top performance of face recognition: Transferred deep feature fusion," *arXiv preprint arXiv:1704.00438*, 2017.

[27] M. Gunther, S. Cruz, E. M. Rudd, and T. E. Boult, "Toward open-set face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 71–80. DOI: https://doi.org/10.1109/cvprw.2017.85.

[28] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and G. Hua, "Neural aggregation network for video face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4362–4371. DOI: https://doi.org/10.1109/cvpr.2017.554.

[29] Y. Liu, J. Yan, and W. Ouyang, "Quality aware network for set to set recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5790–5799. DOI: https://doi.org/10.1109/cvpr.2017.499.

[30] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, 2011, pp. 529–534. DOI: https://doi.org/10.1109/cvpr.2011.5995566.

[31] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[32] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *Proceedings of the European Conference on Computer Vision*, Springer, 2016, pp. 87–102. DOI:
.

[33] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, IEEE, 2018, pp. 67–74. DOI: https://doi.org/10.1109/fg.2018.00020.

[34] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4873–4882. DOI: https://doi.org/10.1109/cvpr.2016.527.

[35] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain, "Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1931–1939. DOI: https://doi.org/10.1109/cvpr.2015.7298803.

[36] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen, *et al.*, "Iarpa janus benchmark-b face dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 90–98. DOI: https://doi.org/10.1109/cvprw.2017.87.

[37] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, *et al.*, "Iarpa janus benchmark-c: Face dataset and protocol," in *2018 International Conference on Biometrics (ICB)*, IEEE, 2018, pp. 158–165. DOI: https://doi.org/10.1109/icb2018.2018.00033.

[38] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 212–220. DOI: https://doi.org/10.1109/cvpr.2017.713.

[39] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016. DOI: https://doi.org/10.1109/lsp.2016.2603342.

[40] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

[41] G. B. Huang and E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," *Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep*, vol. 14, no. 003, 2014.

[42] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708. DOI: https://doi.org/10.1109/cvpr.2014.220.

[43] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823. DOI: https://doi.org/10.1109/cvpr.2015.7298682.

[44] O. M. Parkhi, A. Vedaldi, A. Zisserman, *et al.*, "Deep face recognition," in *The British Machine Vision Conference (BMVC)*, vol. 1, 2015, p. 6. DOI: https://doi.org/10.5244/c.29.41.

[45] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2892–2900. DOI: https://doi.org/10.1109/cvpr.2015.7298907.

[46] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang, "Targeting ultimate accuracy: Face recognition via deep embedding," *arXiv preprint arXiv:1506.07310*, 2015.

[47] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proceedings of the European Conference on Computer Vision*, Springer, 2016, pp. 499–515. DOI:
.

[48] C. Ding and D. Tao, "Robust face recognition via multimodal deep face representation," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2049–2058, 2015. DOI: https://doi.org/10.1109/tmm.2015.2477042.

[49] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *ICML*, vol. 2, 2016, p. 7. DOI: https://doi.org/10.5555/3045390.3045445.

[50] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1891–1898. DOI: https://doi.org/10.1109/cvpr.2014.244.

[51] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

[52] L. Tongyang, X. Xiaoyu, L. Qian, and A. Jan, "Face set recognition," in *Imaging and Multimedia Analytics in a Web and Mobile World*, vol. 2019, Society for Imaging Science and Technology, 2019. DOI: https://doi.org/10.2352/issn.2470-1173.2019.8.imawm-400.

[53] X. Shaoyuan, L. Qian, and A. Jan, "Real-time facial expression recognition using deep learning," in *Imaging and Multimedia Analytics in a Web and Mobile World*, vol. 2019, Society for Imaging Science and Technology, 2019. DOI: https://doi.org/10.2139/ssrn.3421486.

[54] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, IEEE, 2011, pp. 2144–2151. DOI: https://doi.org/10.1109/iccvw.2011.6130513.

[55] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 397–403. DOI: https://doi.org/10.1109/iccvw.2013.59.

[56] G. G. Chrysos, E. Antonakos, S. Zafeiriou, and P. Snape, "Offline deformable face tracking in arbitrary videos," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 1–9. DOI: https://doi.org/10.1109/iccvw.2015.126.

[57] H. Shen, S.-I. Yu, Y. Yang, D. Meng, and A. Hauptmann, "Unsupervised video adaptation for parsing human motion," in *Proceedings of the European Conference on Computer Vision*, Springer, 2014, pp. 347–360. DOI:
.

[58] G. Trigeorgis, P. Snape, M. A. Nicolaou, E. Antonakos, and S. Zafeiriou, "Mnemonic descent method: A recurrent process applied for end-to-end face alignment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4177–4187. DOI: https://doi.org/10.1109/cvpr.2016.453.

[59] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou, "Look at boundary: A boundary-aware face alignment algorithm," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2129–2138. DOI: https://doi.org/10.1109/cvpr.2018.00227.

[60] F. Wang, L. Chen, C. Li, S. Huang, Y. Chen, C. Qian, and C. Change Loy, "The devil of face recognition is in the noise," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 765–780. DOI:
.

[61] X. Dong, S.-I. Yu, X. Weng, S.-E. Wei, Y. Yang, and Y. Sheikh, "Supervision-by-registration: An unsupervised approach to improve the precision of facial landmark detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 360–368. DOI: https://doi.org/10.1109/cvpr.2018.00045.

[62] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "A semi-automatic methodology for facial landmark annotation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 896–903. DOI: https://doi.org/10.1109/cvprw.2013.132.

[63] X. Peng, S. Zhang, Y. Yang, and D. N. Metaxas, "PIEFA: Personalized incremental and ensemble face alignment," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3880–3888. DOI: https://doi.org/10.1109/iccv.2015.442.

[64] X. Peng, R. S. Feris, X. Wang, and D. N. Metaxas, "A recurrent encoder-decoder network for sequential face alignment," in *Proceedings of the European Conference on Computer Vision*, Springer, 2016, pp. 38–56. DOI:
.

[65] G. G. Chrysos, E. Antonakos, P. Snape, A. Asthana, and S. Zafeiriou, "A comprehensive performance evaluation of deformable face tracking "in-the-wild"," *International Journal of Computer Vision*, vol. 126, no. 2-4, pp. 198–232, 2018. DOI: https://doi.org/10.1007/s11263-017-0999-5.

[66] M. H. Khan, J. McDonagh, and G. Tzimiropoulos, "Synergy between face alignment and tracking via discriminative global consensus optimization," in *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, 2017, pp. 3811–3819. DOI: https://doi.org/10.1109/iccv.2017.409.

[67] J. M. D. Barros, B. Mirbach, F. Garcia, K. Varanasi, and D. Stricker, "Fusion of keypoint tracking and facial landmark detection for real-time head pose estimation," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2018, pp. 2028–2037. DOI: https://doi.org/10.1109/wacv.2018.00224.

[68] X. Xiang, R. Jessome, E. Maggard, Y. Bang, M. Cho, and J. Allebach, "Blockwise based detection of local defects," in *Image Quality and System Performance XVI (Part of IS&T Electronic Imaging 2019)*, Society for Imaging Science and Technology, 2019, pp. 303–1. DOI: https://doi.org/10.2352/issn.2470-1173.2019.10.iqsp-303.

[69] P. McCullagh and J. Nelder, "Generalized linear models., 2nd edn.(chapman and hall: London)," *Standard Book on Generalized Linear Models*, 1989. DOI: https://doi.org/10.2307/2347392.

[70] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454. DOI: https://doi.org/10.1109/cvpr.2018.00984.

[71] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2noise: Learning image restoration without clean data," *arXiv preprint arXiv:1803.04189*, 2018.

[72] Z.-H. Feng, J. Kittler, M. Awais, P. Huber, and X.-J. Wu, "Wing loss for robust facial landmark localisation with convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2235–2245. DOI: https://doi.org/10.1109/cvpr.2018.00238.

[73] R. Mao, Q. Lin, and J. P. Allebach, "Robust convolutional neural network cascade for facial landmark localization exploiting training data augmentation," in *Imaging and Multimedia Analytics in a Web and Mobile World*, Society for Imaging Science and Technology, 2018, pp. 374–1. DOI: https://doi.org/10.2352/issn.2470-1173.2018.10.imawm-374.

[74] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2013, pp. 532–539. DOI: https://doi.org/10.1109/cvpr.2013.75.

[75] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Learning deep representation for face alignment with auxiliary attributes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 5, pp. 918–930, 2016. DOI: https://doi.org/10.1109/tpami.2015.2469286.

[76] S. Zhu, C. Li, C. Change Loy, and X. Tang, "Face alignment by coarse-to-fine shape searching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4998–5006. DOI: https://doi.org/10.1109/cvpr.2015.7299134.

[77] X. Miao, X. Zhen, X. Liu, C. Deng, V. Athitsos, and H. Huang, "Direct shape regression networks for end-to-end face alignment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5040–5049. DOI: https://doi.org/10.1109/cvpr.2018.00529.

[78] X. Dong, Y. Yan, W. Ouyang, and Y. Yang, "Style aggregated network for facial landmark detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 379–388. DOI: https://doi.org/10.1109/cvpr.2018.00047.

[79] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face alignment at 3000 fps via regressing local binary features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1685–1692. DOI: https://doi.org/10.1109/cvpr.2014.218.

[80] J. Lv, X. Shao, J. Xing, C. Cheng, and X. Zhou, "A deep regression architecture with two-stage re-initialization for high performance facial landmark detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3317–3326. DOI: https://doi.org/10.1109/cvpr.2017.393.

[81] S. Guo, F. Li, H. Nada, H. Uchida, T. Matsunami, and N. Abe, "Face alignment via 3d-assisted features," in *Imaging and Multimedia Analytics in a Web and Mobile World*, Society for Imaging Science and Technology, 2019, pp. 403–1. DOI: https://doi.org/10.2352/issn.2470-1173.2019.8.imawm-403.

[82] H. White, "Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings," *Neural networks*, vol. 3, no. 5, pp. 535–549, 1990. DOI: https://doi.org/10.1016/0893-6080(90)90004-5.

[83] M.-Y. Shen and C.-C. J. Kuo, "Review of postprocessing techniques for compression artifact removal," *Journal of Visual Communication and Image Representation*, vol. 9, no. 1, pp. 2–14, 1998. DOI: https://doi.org/10.1006/jvci.1997.0378.

[84] J. Allebach and P. W. Wong, "Edge-directed interpolation," in *Proceedings of 3rd IEEE International Conference on Image Processing*, IEEE, vol. 3, 1996, pp. 707–710. DOI: https://doi.org/10.1002/9781119119623.ch7.

[85] B. Wronski, I. Garcia-Dorado, M. Ernst, D. Kelly, M. Krainin, C.-K. Liang, M. Levoy, and P. Milanfar, "Handheld multi-frame super-resolution," *arXiv preprint arXiv:1905.03277*, 2019. DOI: https://doi.org/10.1145/3306346.3323024.

[86] X. Xiang, Y. Tian, Y. Zhang, Y. Fu, J. P. Allebach, and C. Xu, "Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3370–3379. DOI: https://doi.org/10.1109/cvpr42600.2020.00343.

[87] Z. Xiao, M. Nguyen, E. Maggard, M. Shaw, J. Allebach, and A. Reibman, "Real-time print quality diagnostics," in *Image Quality and System Performance XIV (Part of IS&T Electronic Imaging 2017)*, Society for Imaging Science and Technology, 2017, pp. 174–179. DOI: https://doi.org/10.2352/issn.2470-1173.2017.12.iqsp-239.

[88] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proceedings of the European Conference on Computer Vision*, Springer, 2014, pp. 184–199. DOI:
.

[89] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 136–144. DOI: https://doi.org/10.1109/cvprw.2017.151.

[90] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2472–2481. DOI: https://doi.org/10.1109/cvpr.2018.00262.

[91] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. DOI: https://doi.org/10.1109/TPAMI.2020.2968521.

[92] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *et al.*, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. DOI: https://doi.org/10.1109/tip.2003.819861.

[93] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4681–4690. DOI: https://doi.org/10.1109/cvpr.2017.19.

[94] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *The European Conference on Computer Vision Workshops*, 2018. DOI:
.

[95] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proceedings of the European Conference on Computer Vision*, Springer, 2016, pp. 694–711. DOI: .

[96] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang, "Learning a no-reference quality metric for single-image super-resolution," *Computer Vision and Image Understanding*, vol. 158, pp. 1–16, 2017. DOI: https://doi.org/10.1016/j.cviu.2016.12.009.

[97] P. Svoboda, M. Hradis, D. Barina, and P. Zemcik, "Compression artifacts removal using convolutional neural networks," *arXiv preprint arXiv:1605.00366*, 2016.

[98] C. Dong, Y. Deng, C. Change Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 576–584. DOI: https://doi.org/10.1109/iccv.2015.73.

[99] L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo, "Deep generative adversarial compression artifact removal," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4826–4835. DOI: https://doi.org/10.1109/iccv.2017.517.

[100] B. Zhang and J. P. Allebach, "Adaptive bilateral filter for sharpness enhancement and noise removal," *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 664–678, 2008. DOI: https://doi.org/10.1109/tip.2008.919949.

[101] B. Zhang, J. Gu, C. Chen, J. Han, X. Su, X. Cao, and J. Liu, "One-two-one networks for compression artifacts reduction in remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 145, pp. 184–196, 2018. DOI: https://doi.org/10.1016/j.isprsjprs.2018.01.003.

[102] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017. DOI: https://doi.org/10.1109/tip.2017.2662206.

[103] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 286–301. DOI: .

[104] C. B. Atkins, C. A. Bouman, and J. P. Allebach, "Optimal image scaling using pixel classification," in *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, IEEE, vol. 3, 2001, pp. 864–867. DOI: https://doi.org/10.1109/icip.2001.958257.

[105] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proceedings of the European Conference on Computer Vision*, Springer, 2016, pp. 391–407. DOI: .

[106] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient subpixel convolutional neural network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883. DOI: https://doi.org/10.1109/cvpr.2016.207.

[107] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *The British Machine Vision Conference (BMVC)*, 2012. DOI: https://doi.org/10.5244/c.26.135.

[108] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International Conference on Curves and Surfaces*, Springer, 2010, pp. 711–730. DOI: .

[109] D. Martin, C. Fowlkes, D. Tal, J. Malik, *et al.*, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of the Eighth IEEE International Conference on Computer Vision*, ICCV 2001, Vancouver, vol. 2, 2001, pp. 416–423. DOI: https://doi.org/10.1109/iccv.2001.937655.

[110] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5197–5206. DOI: https://doi.org/10.1109/cvpr.2015.7299156.

[111] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A face detection benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. DOI: https://doi.org/10.1109/CVPR.2016.596.

[112] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. De Las Heras, "ICDAR 2013 robust reading competition," in *2013 12th International Conference on Document Analysis and Recognition*, IEEE, 2013, pp. 1484–1493. DOI: https://doi.org/10.1109/icdar.2013.221.

[113] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 065–11 074. DOI: https://doi.org/10.1109/cvpr.2019.01132.

[114] N. Doulamis, A. Doulamis, D. Kalogeras, and S. Kollias, "Low bit-rate coding of image sequences using adaptive regions of interest," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 8, pp. 928–934, 1998. DOI: https://doi.org/10.1016/b978-044482587-2/50123-3.

[115] H. R. Wu and K. R. Rao, *Digital video image quality and perceptual coding*. CRC Press, 2017. DOI: https://doi.org/10.1201/9781420027822.

[116] X. Liu, X. Wu, J. Zhou, and D. Zhao, "Data-driven sparsity-based restoration of JPEG-compressed images in dual transform-pixel domain," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5171–5178. DOI: https://doi.org/10.1109/cvpr.2015.7299153.

[117] K. Yu, C. Dong, C. C. Loy, and X. Tang, "Deep convolution networks for compression artifacts reduction," *arXiv preprint arXiv:1608.02778*, 2016.

[118] H. Chen, X. He, C. Ren, L. Qing, and Q. Teng, "Cisrdcnn: Super-resolution of compressed images using deep convolutional neural networks," *Neurocomputing*, vol. 285, pp. 204–219, 2018. DOI: https://doi.org/10.1016/j.neucom.2018.01.043.

[119] S. Zini, S. Bianco, and R. Schettini, "Deep residual autoencoder for quality independent JPEG restoration," *arXiv preprint arXiv:1903.06117*, 2019. DOI: https://doi.org/10.1109/access.2020.2984387.

[120] J. Guo and H. Chao, "Building dual-domain representations for compression artifacts reduction," in *Proceedings of the European Conference on Computer Vision*, Springer, 2016, pp. 628–644. DOI: .

[121] Z. Wang, D. Liu, S. Chang, Q. Ling, Y. Yang, and T. S. Huang, "D3: Deep dual-domain based fast restoration of JPEG-compressed images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2764–2772. DOI: https://doi.org/10.1109/cvpr.2016.302.

[122] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017. DOI: https://doi.org/10.1109/tpami.2017.2699184.

[123] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 126–135. DOI: https://doi.org/10.1109/cvprw.2017.150.

[124] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *The IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 114–125. DOI: https://doi.org/10.1109/CVPRW.2017.149.

[125] MATLAB, *9.5.0.944444 (R2018b)*. Natick, Massachusetts: The MathWorks Inc., 2018.

[126] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, "Sketch-based manga retrieval using manga109 dataset," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21 811–21 838, 2017. DOI: https://doi.org/10.1007/s11042-016-4020-z.

[127] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[128] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, "What is wrong with scene text recognition model comparisons? dataset and model analysis," *arXiv preprint arXiv:1904.01906*, 2019.

[129] C. Zhu, R. Tao, K. Luu, and M. Savvides, "Seeing small faces from robust anchor's perspective," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5127–5136. DOI: https://doi.org/10.1109/cvpr.2018.00538.

[130] Y. Yoo, D. Han, and S. Yun, "Extd: Extremely tiny face detector via iterative filter reuse," *arXiv preprint arXiv:1906.06579*, 2019.

[131] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem, "Finding tiny faces in the wild with generative adversarial network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 21–30. DOI: https://doi.org/10.1109/cvpr.2018.00010.

[132] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680. DOI: https://doi.org/10.5555/2969033.2969125.

[133] P. Hu and D. Ramanan, "Finding tiny faces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 951–959. DOI: https://doi.org/10.1109/cvpr.2017.166.

[134] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, pp. 8026–8037, 2019.

[135] S. Niklaus and F. Liu, "Context-aware synthesis for video frame interpolation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1701–1710. DOI: https://doi.org/10.1109/cvpr.2018.00183.

[136] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, "Non-local recurrent network for image restoration," in *Advances in Neural Information Processing Systems*, 2018, pp. 1673–1682. DOI: https://doi.org/10.1109/10.5555/3326943.3327097.

[137] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803. DOI: https://doi.org/10.1109/cvpr.2018.00813.

[138] M. J. Farah, K. D. Wilson, M. Drain, and J. N. Tanaka, "What is "special" about face perception?" *Psychological Review*, vol. 105, no. 3, p. 482, 1998. DOI: https://doi.org/10.1037/0033-295x.105.3.482.

[139] K. A. Quinn and C. N. Macrae, "The face and person perception: Insights from social cognition," *British Journal of Psychology*, vol. 102, no. 4, pp. 849–867, 2011. DOI: https://doi.org/10.1111/j.2044-8295.2011.02030.x.

[140] G. Rhodes and J. Haxby, *Oxford Handbook of Face Perception*. Oxford University Press, 2011. DOI: https://doi.org/10.1093/oxfordhb/9780199559053.001.0001.

[141] J.-S. Park and S.-W. Lee, "An example-based face hallucination method for single-frame, low-resolution facial images," *IEEE Transactions on Image Processing*, vol. 17, pp. 1806–1816, 2008. DOI: https://doi.org/10.1109/tip.2008.2001394.

[142] C.-Y. Yang, S. Liu, and M.-H. Yang, "Structured face hallucination," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1099–1106. DOI: https://doi.org/10.1109/cvpr.2013.146.

[143] Y. Song, J. Zhang, S. He, L. Bao, and Q. Yang, "Learning to hallucinate face images via component generation and enhancement," *arXiv preprint arXiv:1708.00223*, 2017. DOI: https://doi.org/10.24963/ijcai.2017/633.

[144] Q. Cao, L. Lin, Y. Shi, X. Liang, and G. Li, "Attention-aware face hallucination via deep reinforcement learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 690–698. DOI: https://doi.org/10.1109/cvpr.2017.180.

[145] Y. Chen, Y. Tai, X. Liu, C. Shen, and J. Yang, "Fsrnet: End-to-end learning face super-resolution with facial priors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2492–2501. DOI: https://doi.org/10.1109/cvpr.2018.00264.

[146] A. Bulat, J. Yang, and G. Tzimiropoulos, "To learn image super-resolution, use a GAN to learn how to do image degradation first," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 185–200. DOI: .

[147] X. Xiang, Q. Lin, and J. P. Allebach, "Boosting high-level vision with joint compression artifacts reduction and super-resolution," *arXiv preprint arXiv:2010.08919*, 2020. DOI: https://doi.org/10.1109/icpr48806.2021.9413154.

[148] H. Yue, X. Sun, J. Yang, and F. Wu, "Landmark image super-resolution by retrieving web images," *IEEE Transactions on Image Processing*, vol. 22, pp. 4865–4878, 2013. DOI: https://doi.org/10.1109/tip.2013.2279315.

[149] Y. Wang, Y. Liu, W. Heidrich, and Q. Dai, "The light field attachment: Turning a DSLR into a light field camera using a low budget camera ring," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, pp. 2357–2364, 2016. DOI: https://doi.org/10.1109/tvcg.2016.2628743.

[150] H. Zheng, M. Ji, H. Wang, Y. Liu, and L. Fang, "Crossnet: An end-to-end reference-based super resolution network using cross-scale warping," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 88–104. DOI: .

[151] X. Li, M. Liu, Y. Ye, W. Zuo, L. Lin, and R. Yang, "Learning warped guidance for blind face restoration," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 272–289. DOI: .

[152] B. Dogan, S. Gu, and R. Timofte, "Exemplar guided face image super-resolution without facial landmarks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019. DOI: https://doi.org/10.1109/cvprw.2019.00232.

[153] G. Shim, J. Park, and I. S. Kweon, "Robust reference-based super-resolution with similarity-aware deformable convolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8425–8434. DOI: https://doi.org/10.1109/cvpr42600.2020.00845.

[154] V. Boominathan, K. Mitra, and A. Veeraraghavan, "Improving resolution and depth-of-field of light field cameras using a hybrid imaging system," in *2014 IEEE International Conference on Computational Photography (ICCP)*, IEEE, 2014, pp. 1–10. DOI: https://doi.org/10.1109/iccphot.2014.6831814.

[155] Z. Zhang, Z. Wang, Z. Lin, and H. Qi, "Image super-resolution by neural texture transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7982–7991. DOI: https://doi.org/10.1109/cvpr.2019.00817.

[156] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, "Learning texture transformer network for image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5791–5800. DOI: https://doi.org/10.1109/cvpr42600.2020.00583.

[157] Y. Xie, J. Xiao, M. Sun, C. Yao, and K. Huang, "Feature representation matters: End-to-end learning for reference-based image super-resolution," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 230–245. DOI: .

[158] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, IEEE, vol. 2, 1999, pp. 1150–1157. DOI: https://doi.org/10.1109/iccv.1999.790410.

[159] Y. Zhang, Z. Zhang, S. DiVerdi, Z. Wang, J. Echevarria, and Y. Fu, "Texture hallucination for large-factor painting super-resolution," *arXiv preprint arXiv:1912.00515*, 2019. DOI: .

[160] X. Li, W. Li, D. Ren, H. Zhang, M. Wang, and W. Zuo, "Enhanced blind face restoration with multi-exemplar images and adaptive spatial feature fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2706–2715. DOI: https://doi.org/10.1109/cvpr42600.2020.00278.

[161] K. Wang, J. Oramas, and T. Tuytelaars, "Multiple exemplars-based hallucination for face super-resolution and editing," in *Proceedings of the Asian Conference on Computer Vision*, 2020. DOI: .

[162] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2462–2470. DOI: https://doi.org/10.1109/CVPR.2017.179.

[163] F. Reda, R. Pottorff, J. Barker, and B. Catanzaro, *Flownet2-pytorch: Pytorch implementation of flownet 2.0: Evolution of optical flow estimation with deep networks*, https://github.com/NVIDIA/flownet2-pytorch, 2017.

[164] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002. DOI: https://doi.org/10.1109/38.988747.

[165] S. Cui, "Towards content-independent multi-reference super-resolution: Adaptive pattern matching and feature aggregation," 2020. DOI: .

[166] S. Zhu, S. Liu, C. C. Loy, and X. Tang, "Deep cascaded bi-network for face hallucination," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 614–630. DOI: .

[167] A. Bulat and G. Tzimiropoulos, "Super-fan: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with gans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 109–117. DOI: https://doi.org/10.1109/cvpr.2018.00019.

[168] X. Yu, B. Fernando, B. Ghanem, F. Porikli, and R. Hartley, "Face super-resolution guided by facial component heatmaps," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 217–233. DOI: .

[169] D. Kim, M. Kim, G. Kwon, and D.-S. Kim, "Progressive face super-resolution via attention to facial landmark," *arXiv preprint arXiv:1908.08239*, 2019.

[170] Y. Yin, J. Robinson, Y. Zhang, and Y. Fu, "Joint super-resolution and alignment of tiny faces," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 12 693–12 700. DOI: https://doi.org/10.1609/aaai.v34i07.6962.

[171] C. Chen, X. Li, L. Yang, X. Lin, L. Zhang, and K.-Y. K. Wong, "Progressive semantic-aware style transformation for blind face restoration," *arXiv preprint arXiv:2009.08709*, 2020.

[172] K. Zhang, Z. Zhang, C.-W. Cheng, W. H. Hsu, Y. Qiao, W. Liu, and T. Zhang, "Super-identity convolutional neural network for face hallucination," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 183–198. DOI: .

[173] C.-C. Hsu, C.-W. Lin, W.-T. Su, and G. Cheung, "Sigan: Siamese generative adversarial network for identity-preserving face hallucination," *IEEE Transactions on Image Processing*, vol. 28, pp. 6225–6236, 2019. DOI: https://doi.org/10.1109/tip.2019.2924554.

[174] K. Grm, W. J. Scheirer, and V. Štruc, "Face hallucination using cascaded super-resolution and identity priors," *IEEE Transactions on Image Processing*, vol. 29, pp. 2150–2165, 2019. DOI: https://doi.org/10.1109/tip.2019.2945835.

[175] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773. DOI: https://doi.org/10.1109/iccv.2017.89.

[176] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9308–9316. DOI: https://doi.org/10.1109/cvpr.2019.00953.

[177] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. Change Loy, "EDVR: Video restoration with enhanced deformable convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019. DOI: https://doi.org/10.1109/cvprw.2019.00247.

[178] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, "Tdan: Temporally-deformable alignment network for video super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3360–3369. DOI: https://doi.org/10.1109/cvpr42600.2020.00342.

[179] M. Guizar-Sicairos, S. T. Thurman, and J. R. Fienup, "Efficient subpixel image registration algorithms," *Optics Letters*, vol. 33, no. 2, pp. 156–158, 2008. DOI: https://doi.org/10.1364/ol.33.000156.

[180] A. Aitken, C. Ledig, L. Theis, J. Caballero, Z. Wang, and W. Shi, "Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize," *arXiv preprint arXiv:1707.02937*, 2017.

[181] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 624–632. DOI: https://doi.org/10.1109/cvpr.2017.618.

[182] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2414–2423. DOI: https://doi.org/10.1109/cvpr.2016.265.

[183] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[184] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[185] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8110–8119. DOI: https://doi.org/10.1109/cvpr42600.2020.00813.

[186] A. Jolicoeur-Martineau, "The relativistic discriminator: A key element missing from standard gan," *arXiv preprint arXiv:1807.00734*, 2018.

[187] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. DOI: https://doi.org/10.1109/cvpr42600.2020.00559.

[188] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision*, 2015. DOI: https://doi.org/10.1109/iccv.2015.425.

[189] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595. DOI: https://doi.org/10.1109/cvpr.2018.00068.

[190] C. Chen, D. Gong, H. Wang, Z. Li, and K.-Y. K. Wong, "Learning spatial attention for face super-resolution," *IEEE Transactions on Image Processing*, vol. 30, pp. 1219–1231, 2020. DOI: https://doi.org/10.1109/tip.2020.3043093.

[191] X. Li, C. Chen, S. Zhou, X. Lin, W. Zuo, and L. Zhang, "Blind face restoration via deep multi-scale component dictionaries," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 399–415. DOI:
.

[192] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The deepfake detection challenge dataset," *arXiv preprint arXiv:2006.07397*, 2020.

[193] Z. Lin, J. Sun, A. Davis, and N. Snavely, "Visual chirality," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 295–12 303. DOI: https://doi.org/10.1109/cvpr42600.2020.01231.

[194] E. Shechtman, Y. Caspi, and M. Irani, "Space-time super-resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 4, pp. 531–545, 2005. DOI: https://doi.org/10.1109/TPAMI.2005.85.

[195] U. Mudenagudi, S. Banerjee, and P. K. Kalra, "Space-time super-resolution using graph-cut optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 995–1008, 2010. DOI: https://doi.org/10.1109/tpami.2010.167.

[196] H. Takeda, P. Van Beek, and P. Milanfar, "Spatiotemporal video upscaling using motion-assisted steering kernel (mask) regression," in *High-Quality Visual Experience*, Springer, 2010, pp. 245–274. DOI: .

[197] O. Shahar, A. Faktor, and M. Irani, *Space-time super-resolution from a single video*. IEEE, 2011. DOI: https://doi.org/10.1109/cvpr.2011.5995360.

[198] E. Faramarzi, D. Rajan, and M. P. Christensen, "Space-time super-resolution from multiple-videos," in *International Conference on Information Science, Signal Processing and their Applications*, IEEE, 2012, pp. 23–28. DOI: https://doi.org/10.1109/ISSPA.2012.6310553.

[199] T. Li, X. He, Q. Teng, Z. Wang, and C. Ren, "Space–time super-resolution with patch group cuts prior," *Signal Processing: Image Communication*, vol. 30, pp. 147–165, 2015. DOI: https://doi.org/10.1016/j.image.2014.10.007.

[200] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive convolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 670–679. DOI: https://doi.org/10.1109/cvpr.2017.244.

[201] J. Caballero, C. Ledig, A. Aitken, A. Acosta, J. Totz, Z. Wang, and W. Shi, "Real-time video super-resolution with spatio-temporal networks and motion compensation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4778–4787. DOI: https://doi.org/10.1109/cvpr.2017.304.

[202] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep video deblurring for hand-held cameras," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1279–1288. DOI: https://doi.org/10.1109/cvpr.2017.33.

[203] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive separable convolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 261–270. DOI: https://doi.org/10.1109/iccv.2017.37.

[204] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1106–1125, 2019. DOI: https://doi.org/10.1007/s11263-018-01144-2.

[205] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, "Depth-aware video frame interpolation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3703–3712. DOI: https://doi.org/10.1109/cvpr.2019.00382.

[206] Y. Jo, S. Wug Oh, J. Kang, and S. Joo Kim, "Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3224–3232. DOI: https://doi.org/10.1109/cvpr.2018.00340.

[207] M. Haris, G. Shakhnarovich, and N. Ukita, "Recurrent back-projection network for video super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3897–3906. DOI: https://doi.org/10.1109/cvpr.2019.00402.

[208] C. Liu and D. Sun, "A bayesian approach to adaptive video super resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, 2011, pp. 209–216. DOI: https://doi.org/10.1109/cvpr.2011.5995614.

[209] S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. Sorkine-Hornung, "Phase-based frame interpolation for video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1410–1418. DOI: https://doi.org/10.1109/cvpr.2015.7298747.

[210] G. Long, L. Kneip, J. M. Alvarez, H. Li, X. Zhang, and Q. Yu, "Learning image matching by simply watching video," in *Proceedings of the European Conference on Computer Vision*, Springer, 2016, pp. 434–450. DOI:
.

[211] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz, "Super slomo: High quality estimation of multiple intermediate frames for video interpolation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9000–9008. DOI: https://doi.org/10.1109/cvpr.2018.00938.

[212] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4463–4471. DOI: https://doi.org/10.1109/iccv.2017.478.

[213] W. Bao, W.-S. Lai, X. Zhang, Z. Gao, and M.-H. Yang, "Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. DOI: https://doi.org/10.1109/tpami.2019.2941941.

[214] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, "Detail-revealing deep video super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4472–4480. DOI: https://doi.org/10.1109/iccv.2017.479.

[215] M. S. Sajjadi, R. Vemulapalli, and M. Brown, "Frame-recurrent video super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6626–6634. DOI: https://doi.org/10.1109/cvpr.2018.00693.

[216] L. Wang, Y. Guo, Z. Lin, X. Deng, and W. An, "Learning for video super-resolution through hr optical flow estimation," in *Asian Conference on Computer Vision*, Springer, 2018, pp. 514–529. DOI: .

[217] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, "Tdan: Temporally-deformable alignment network for video super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3360–3369. DOI: https://doi.org/10.1109/cvpr42600.2020.00342.

[218] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems*, 2015, pp. 802–810. DOI: https://doi.org/10.5555/2969239.2969329.

[219] B. Lim and K. M. Lee, "Deep recurrent resnet for video super-resolution," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2017, pp. 1452–1455. DOI: https://doi.org/10.1109/apsipa.2017.8282261.

[220] Y. Huang, W. Wang, and L. Wang, "Video super-resolution via bidirectional recurrent convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 1015–1028, 2017. DOI: https://doi.org/10.1109/tpami.2017.2701380.

[221] E. Shechtman, Y. Caspi, and M. Irani, "Increasing space-time resolution in video," in *Proceedings of the European Conference on Computer Vision*, Springer, 2002, pp. 753–768. DOI: .

[222] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 721–741, 1984. DOI: https://doi.org/10.1016/b978-0-08-051581-6.50057-x.

[223] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001. DOI: https://doi.org/10.1109/iccv.1999.791245.

[224] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997. DOI: https://doi.org/10.1109/78.650093.

[225] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[226] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," *arXiv preprint arXiv:1903.10082*, 2019.

VITA

Xiaoyu Xiang received the B.E. degree in engineering physics from Tsinghua University in 2015; and a doctoral degree in 2021 in Electrical and Computer Engineering at Purdue University. Her primary areas of research include image and video restoration and enhancement.

PUBLICATION(S)

Technical Reports

- **Xiaoyu Xiang**, Jon Morton, Federico Perazzi, Fitsum A. Reda, Rakesh Ranjan, "Efficient Headshot Image Super-Resolution with Multi-References", *submitted to ICCV 2021*

- Lucas Young, Fitsum A. Reda, Rakesh Ranjan, Jon Morton, Jun Hu, Yazhu Ling, **Xiaoyu Xiang**, David Liu, Vikas Chandra, "Feature-Align Network and Knowledge Distillation for Efficient Denoising", *submitted to ICCV 2021*

- **Xiaoyu Xiang**, Ding Liu, Xiaohui Shen, Yiheng Zhu, Xiao Yang, Jan Allebach, "Adversarial Open-Domain Adaption for Sketch-to-Photo Synthesis", *submitted to ICCV 2021*

- **Xiaoyu Xiang**, Yapeng Tian, Yulun Zhang, Yun Fu, Jan Allebach, Chenliang Xu, "Zooming SlowMo: An Efficient One-Stage Framework for Space-Time Video Super-Resolution", *submitted to T-PAMI.*

Conference Publications

- **Xiaoyu Xiang**, Ding Liu, Xiaohui Shen, Yiheng Zhu, Xiao Yang, Jan Allebach, "Adversarial Open-Domain Adaption for Sketch-to-Photo Synthesis", *CVPR Workshop 2021*

- **Xiaoyu Xiang**, Qian Lin, Jan Allebach, "Boosting High-Level Vision with Joint Compression Artifacts Reduction and Super-Resolution", *ICPR 2020*

- **Xiaoyu Xiang**, Yapeng Tian, Yulun Zhang, Yun Fu, Jan Allebach, Chenliang Xu, "Zooming Slow-Mo: Fast and Accurate One-Stage Space-Time Video Super-Resolution", *CVPR 2020.*

- **Xiaoyu Xiang**, Yang Cheng, Shaoyuan Xu, Qian Lin, and Jan Allebach, "The Blessing and The Curse of the Noise Behind Facial Landmark Annotations", *Imaging*

and Multimedia Analytics in a Web and Mobile World (IS&T Electronic Imaging 2020 Symposium). **(Oral)**

- **Xiaoyu Xiang**, Yang Cheng, Jianhang Chen, Qian Lin, and Jan Allebach, "Semi-supervised Multi-task Network for Image Aesthetic Assessment", *Imaging and Multimedia Analytics in a Web and Mobile World (IS&T Electronic Imaging 2020 Symposium).* **(Oral)**

- **Xiaoyu Xiang**, Eric Maggard, Renee Jessome, Yousun Bang, Minki Cho, and Jan Allebach, "Blockwise Detection of Local Defects on Printed Pages", *Image Quality and System Performance XVI (IS&T Electronic Imaging 2019 Symposium)*, N. Bonnier and S. Perry, Eds., Burlingame, CA, 13 January - 17 January 2019. **(Oral)**

- Tongyang Liu, **Xiaoyu Xiang**, Qian Lin, and Jan Allebach, "Face Set Recognition", *Imaging and Multimedia Analytics in a Web and Mobile World (IS&T Electronic Imaging 2019 Symposium)* J. Allebach and Z. Fan, and Q. Lin, Eds., Burlingame, CA, 13 January - 17 January 2019. **(Oral)**

Patents

- **Xiaoyu Xiang**, Ding Liu, Xiaohui Shen, Yiheng Zhu, Xiao Yang, "Adversarial Open-Domain Adaption for Sketch-to-Photo and Photo-to-Sketch Generation", *US Patent & CN Patent* application filed

- **Xiaoyu Xiang**, Tianqi Guo, Qian Lin, Jan Allebach, "Generating Super-Resolution Images with Optimized Face Rendering", *US Patent* application filed

- **Xiaoyu Xiang**, Qian Lin, Jan Allebach, "Deep Neural Network with Atrous Spatial Pooling Pyramid Module for Jointly Super Resolution And Compression Artifacts Reduction", *US Patent* application filed

- Yang Cheng, **Xiaoyu Xiang**, Shaoyuan Xu, Qian Lin, Jan Allebach, "Real-time Temporal Smoothing Methods for 2D Facial Landmark Stabilizing in Video Streams for Augmented Reality Applications Using Tracking Methods", *US Patent* application filed