

# **DNNs for removing Dirt and Sparkle from Movies**



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

**Wei Wei**

**Supervisor: Prof. Anil Kokaram**

Department of Electronic and Electrical Engineering  
Trinity College Dublin

This dissertation is submitted in partial fulfillment for the degree of  
*MSc in Electronic Information Engineering*

I would like to dedicate this thesis to my loving parents

---

## Declaration

---

I declare that this thesis has been submitted as a partial fulfillment for the degree of MSc in Electronic Information Engineering and it is entirely my own work. It contains ten months work on project "DNNs for removing Dirt and Sparkle from Movies".

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year. I have also completed the Online Tutorial provided by the Library, Trinity College Dublin, on avoiding plagiarism 'Ready, Steady, Write', located at <http://tcd-ie.libguides.com/plagiarism/ready-steadywrite>.

I agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Wei Wei  
July 2022

---

## Acknowledgements

---

I must acknowledge my supervisor Prof. Anil Kokaram who offered me so many helps and supports while completing this project and thesis. Without him, I will not get this far. I also need to acknowledge my classmates for providing me good ideas and feasible solutions when I encountered difficulties.

Finally, I need to thank my loving family and sincere friends for giving me emotional supports when I needed the most to get through hardships.

---

## Abstract

---

There are many artefacts in archived film e.g. missing patches (blotches), grain noise, camera shake and brightness flicker. Automated restoration has become important because of the rise of Digital media entertainment which drives the demand for this material. The problem of automated treatment of blotches has been considered since the late 1980's. In this work we develop strategies that combine the classical techniques that rely on models for the video, with Neural Network approaches.

Blotches do not occur in the same place in consecutive frames. Classical techniques used this idea to detect and remove blotches. The earliest algorithm, SDIa [11] simply detected a missing patch if the brightness difference between frames was above some threshold. Then a median filter (ML3Dex [10]) combines spatial and temporal information in 3 frames to interpolate the patch. Motion compensation is key for these algorithms to work.

Neural networks have been presented in the past for blotch detection [18, 15]. While they outperform the classical techniques they do not properly incorporate motion. In this work we design and test several new architectures for blotch detection. We incorporate as input 3 frames both with and without motion compensation. Our best performing system used non-motion multi-inputs autoencoder. It resulted in 0.2326 % false alarm rate and 94.71 % correct detection rate in our dataset.

A key new contribution is the creation of a more useful dataset for training blotch detection or removal neural networks. We use 10 video clips<sup>1</sup> where each contains 100 continuous frames. Those frames were deliberately corrupted and then stored along with their binary frames that mark the positions of blotches. For training and testing neural

---

<sup>1</sup>[media.xiph.org/video/derf](http://media.xiph.org/video/derf)

networks, our dataset takes as inputs the corrupted frames and takes as ground truths the binary frames.

In this dissertation we explore the use of a highly accurate motion estimator : RAFT [18] for the purpose of Blotch removal. Blotches obviously cause errors in motion estimation and so we explore the possibility to retrain RAFT to generate more robust motion estimates. On average we improved RAFT by 27.26 % w.r.t. optical flow images when retraining.

Overall we are able to show much improved results with our final system which combined a DNN detector with ML3Dex, resulted in a PSNR improvement by 48.44 % w.r.t. restored frames.

---

## Table of contents

---

<b>List of figures</b>	<b>ix</b>
<b>List of tables</b>	<b>xii</b>
<b>Nomenclature</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Why is blotch restoration important? . . . . .	1
1.2 A rough background of blotch restoration . . . . .	2
1.3 Our methods for blotch detection and removal . . . . .	3
1.4 Outline of the dissertation . . . . .	4
<b>2 Literature review</b>	<b>5</b>
2.1 Classical Approaches . . . . .	5
2.1.1 Spike Detection Index (SDI) for blotch detection . . . . .	5
2.1.2 ML3Dex filter for blotch removal . . . . .	7
2.2 DNN Approaches . . . . .	8
2.2.1 Deep Neural Network . . . . .	8
2.2.2 Encoder-decoder architecture for blotch detection . . . . .	9
<b>3 Data preparation</b>	<b>11</b>
3.1 Artificial Dataset creation . . . . .	11
3.2 Adding motion information (RAFT) . . . . .	19

Table of contents	viii
<b>4 System for detection of dirt and sparkle</b>	<b>21</b>
4.1 Modified Autoencoder . . . . .	21
4.2 Unet . . . . .	24
<b>5 System for detection and removal</b>	<b>28</b>
5.1 A blotches restoration system . . . . .	29
5.2 The implementation of ML3Dex . . . . .	29
5.3 Evaluation of blotches restoration . . . . .	30
<b>6 Design of robust motion estimator</b>	<b>33</b>
6.1 The problem facing motion estimators on blotches . . . . .	33
6.2 Retrain RAFT and its results . . . . .	35
<b>7 Discussion and Final Comments</b>	<b>41</b>
7.1 Future work . . . . .	42
<b>References</b>	<b>44</b>

---

## List of figures

---

1.1	<i>An example of Dirt (left) and Sparkle (right). These artefacts result in spatial and temporal discontinuities in the brightness of pixels. Typically they do not occur in the same place in consecutive frames.</i>	2
2.1	<i>Sub-filter windows used in ML3Dex filter</i>	7
2.2	<i>The proposed encoder-decoder architecture</i>	9
3.1	<i>One example frame from each of ten video clips</i>	13
3.2	<i>Four examples of stretching phenomenon in the artificial blotches</i>	15
3.3	<i>Two examples of corruption explosion phenomenon</i>	15
3.4	<i>Two examples of under-controlled corruption spreading</i>	16
3.5	<i>Examples of ten corrupted frames (left) and their ground truths (right)</i>	18
3.6	<i>Examples of backward motion added frames. Top row : Frames 18, 19 from the corrupted sequence. Middle row : Motion superimposed frame (left), MC frame (right). Bottom row : Non-MC DFD frame (left), MC DFD frame (right).</i>	20
4.1	<i>The architecture of modified encoder</i>	22
4.2	<i>ROC curves of 4 autoencoders.</i>	23
4.3	<i>ROC curves of autoencoders compared with SDI detectors.</i>	23
4.4	<i>The designed architecture of our Unet</i>	24
4.5	<i>ROC curves of Unet and autoencoders.</i>	25
4.6	<i>ROC curves of SDI detectors, autoencoders and Unet.</i>	26

4.7 <i>Detection results on frame 19 of sequence Akiyo. Top row : Frames 18, 19, 20 from the corrupted sequence. Middle row : Detections using previous work Yous et al [18] (left), SDIa/SDIp (middle, right). Bottom Row : Results from three of the systems designed in this work: UNET (left), 3-Frame autoencoder no motion (middle), 3-Frame autoencoder DFD (right). The 3-Frame autoencoder with no motion shows the best performance with recall = 94.71% and false alarm = 0.2326%.</i>	27
5.1 <i>Blotch restoration system built as a cascade of detection and reconstruction modules.</i>	28
5.2 <i>Sub-filter windows of ML3Dex filter</i>	29
5.3 <i>PSNR of 98 restored frames in Sequence 1 compared with unrestored frames.</i>	30
5.4 <i>Examples of restored frame and its DFD. Top row: original frame (left), corrupted frame (middle) and restored frame (right). Bottom row: DFD frame of corrupted frame (left), DFD frame of restored frame (right).</i>	31
5.5 <i>PSNR of 98 restored frames in Sequence 1 of Unet detection and Ground Truth detection.</i>	31
6.1 <i>An example of RAFT estimation flaws on blotches. 1st column: original frame-8 (top), original frame-9 (middle) and their optical flow image (bottom). 2nd column: corrupted frame-8 (top), corrupted-frame-9 (middle) and their optical flow image (bottom).</i>	34
6.2 <i>Examples of artificially corrupted official dataset (1st row: cleanpass (left), finalpass (right)) and their clean optical flows (2nd row: backward (left), foreward (right))</i>	35
6.3 <i>Examples of two optical flows from original RAFT (1st row: clean input (left), corrupted input (right)) and two optical flows from retrained RAFT (2nd row: clean input (left), corrupted input (right)).</i>	36
6.4 <i>PSNR improvement of 99 backward optical flow images (motion vectors) in Sequence 1 after retraining.</i>	37
6.5 <i>PSNR improvement of 98 restored frames in Sequence 1 after using new motions from retrained RAFT.</i>	37
6.6 <i>Improvement of PSNR of restored frames from blotches detection and removal on blotches restoration.</i>	38

6.7 (1st) Comparison of optical flows from original RAFT and retrained RAFT models of real degraded frames. 1st row: degraded frame $n-1$ , $n$ . 2nd row: backward optical opflow images (from original RAFT (left), retrained RAFT (right)). . . . .	39
6.8 (2nd) Comparison of optical flows from original RAFT and retrained RAFT models of real degraded frames. 1st row: degraded frame $n-1$ , $n$ . 2nd row: backward optical flow images (from original RAFT (left), from retrained RAFT (right)). . . . .	39

---

## List of tables

---

3.1 <i>Basic information of 10 Image Sequences</i> . . . . .	11
--	----

---

## Nomenclature

---

### **Acronyms / Abbreviations**

BM Block Matching

CNN Convolutional Neural Network

DFD Displaced Frames Difference

DNN Deep Neural Network

DPD Displaced Pixel Difference

MC Motion Compensated

ML3Dex Extended Multi Level 3 Dimensional filtering

NLP Natural Language Processing

PSNR Peak Signal-to-Noise Ratio

RAFT Recurrent All-Pairs Field Transforms

RNN Recurrent Neural Network

ROC Receiver Operating Characteristic

ROD Rank Order Detector

SDIa Spike Detection Index - a

SDIp Spike Detection Index - p

SDI Spike Detection Index

SROD Simplified Ranked Order Difference

# CHAPTER 1

---

## Introduction

---

### 1.1 Why is blotch restoration important?

DIGITAL pictures and videos have been accessible to the general public for more than twenty years. From the time SKY and BBC started digital transmissions to now Netflix uses adaptive bitrate streaming technology for adjusting digital video and audio qualities for consumers, the demand for digital videos and audios has never been higher. The global digital video content market reached a value of US\$172 Billion in 2020<sup>1</sup>, 86% of businesses use video as a marketing tool<sup>2</sup>. Up to 2022, digital video viewers are over 3 billion worldwide<sup>3</sup> and over 40% of the global population watch streaming or downloaded video at least once per month in some capacity. The demand for digital videos and the number of digital video viewers are still increasing over years.

With the increasing demand for digital videos, there is a perceived demand for archive videos and movies which has fueled commercial interest in automatic digital restoration processes. As video quality requirement from consumers has become higher and higher, providing similar video quality for aging archive videos might be difficult given various types of artefacts exist such as missing data, noise, image flicker and telecine effects [10]. One of those artefacts is called Dirt and Sparkle, also blotches or missing data, this is

---

<sup>1</sup>[www.globenewswire.com/news-release/2021/03/03/2186030/28124/en/Worldwide-Digital-Video-Content-Industry-to-2026-by-Business-Model-Type-Device-and-Region.html](http://www.globenewswire.com/news-release/2021/03/03/2186030/28124/en/Worldwide-Digital-Video-Content-Industry-to-2026-by-Business-Model-Type-Device-and-Region.html)

<sup>2</sup>[invideo.io/blog/video-marketing-statistics](http://invideo.io/blog/video-marketing-statistics)

<sup>3</sup>[www.insiderintelligence.com/content/over-3-billion-people-worldwide-now-digital-video-viewers](http://www.insiderintelligence.com/content/over-3-billion-people-worldwide-now-digital-video-viewers)



Fig. 1.1 *An example of Dirt (left) and Sparkle (right). These artefacts result in spatial and temporal discontinuities in the brightness of pixels. Typically they do not occur in the same place in consecutive frames.*

usually caused by dust on the film when taking photos or the dust on the negative when printing.

One example of Dirt and Sparkle is shown in Fig. 1.1. As can be seen, the artefacts manifest as outliers in the local intensity of pixel patches. They often occur as patches and not isolated pixels. The artefacts also do not occur in the same place in consecutive frames.

## 1.2 A rough background of blotch restoration

Blotch detection and removal had become one of the prime issues on archive film restoration for decades. For blotch detection, there are already many useful methods, one of the simplest ones is called Spike Detection Index (SDI) [11]. All these methods exploit the unique property of spatial and temporal discontinuity of blotches to detect them. Because motions are smooth, during a very short period of time such as one frame time, scenes and objects in one frame mostly remain the same, which means most of pixels in one frame can be alternatively found in previous and next frame. When phenomena such as occlusion and uncovering occur, pixels can only be alternatively found in either previous or next frame. So in the end, without blotches, all pixels can be alternatively found in either previous or next frame. However, because of the unique property of spatial and temporal discontinuities of blotches, they only appear in one specific frame, and no others. And this is what all those blotches detectors can use for blotches detection.

SDI detectors is one of those detectors which exploit the unique property of blotches to detect blotches. SDIp (Spike Detection Index - p) detectors [10] also modify the original SDI model by adding a sign judgement condition for regarding one pixel as a blotch, it reduces false alarm because its requirement of detecting a blotch becomes more strict. There

are also some other classical blotch detection methods including Rank Order Detector (ROD) [14] and two-stage Simplified Ranked Order Difference (SROD) [8].

In recent years, deep neural networks had been proved efficient in many different areas. Some people also apply deep neural networks for blotch detection. You et al propose a novel supervised model created by a fully CNN-based encoder-decoder architecture [18]. Sizyakin et al introduce a new three-stage method which includes motion compensation, LBP calculations and data classification using deep neural networks [15], they prove their method more efficient than traditional methods SDI, ROD and SROD.

For blotches removal, typical idea is to take similar patches found in other frames to alternatively replace blotches in the current frame. This process is called interpolation. Normally median filters are applied in interpolation process. There are many literatures focused on median filters such as 3D median filter introduced by Alp [1] and Arce [3, 2], 3 tap motion compensated median operation presented by Huang [9] and Martinez [12]. However all those interpolation methods need a good motion estimator to tell them where to find the patches they need for blotches replacement. Motion estimators can estimate displacement on each pixel site between two consecutive frames and tell us where to find the pixel closest to the pixel we need to use for blotches replacement. Over decades, many motion estimators had been introduced and proven efficient, including classical methods such as block matching (BM) [6, 7], and gradient-based approaches [12, 5, 17], or DNN-based methods such as Recurrent All-Pairs Field Transforms (RAFT) [16].

### 1.3 Our methods for blotch detection and removal

We modify the autoencoder architecture [18] to let it receive more inputs, so not only one picture but three pictures can be sent to neural network for blotches detection. The original autoencoder takes only current frame for blotches detection, but blotches in one frame can only be determined as blotches when they compared with other frames. A black hole shown on a wall in one frame might actually exist in that actual wall in reality, we do not know until we see it in the flesh or compare it with other frames. If a suspicious blotch exists in one frame alone, then it can be determined as a blotch. So based on this idea, we believe sending more frames to neural network can help it detect blotches. Our results prove this, showing a higher accuracy is obtained after sending two more frames to neural network. Those two frames can be any of the following three pairs, previous and next frames, backward and forward motion compensated frames, or backward and forward displaced frames difference (DFD). The last two pairs are created by adding motion information to the previous and next frames.

In addition to modify the autoencoder model, we also try out an Unet model for blotches detection. This idea comes from the similarity between blotch detection problem and image segmentation problem where Unet is well-performed. We build an Unet with 10 convolution layers, 4 maxPooling layers and 4 deconvolution layers and train it taking current frame alone as input. The results show the model is even better than most of modified autoencoders.

After blotch detection, a ML3Dex median filter is implemented for interpolation process where blotches are removed by interpolating similar patches found in other frames in blotch areas. Here the motion estimator used for finding similar patches is Recurrent All-Pairs Field Transforms (RAFT) [16]. Therefore, at this point, a complete blotch detection and removal system using DNN detectors and ML3Dex median filter is established.

We also improve the motion estimator RAFT on blotch areas by retraining it taking corrupted frames as inputs and clean motion files as ground truths. The results show its robustness on blotch areas is improved and therefore the blotch removal performance of ML3Dex is improved with it.

## 1.4 Outline of the dissertation

The remainder of this dissertation is organized as follows. In Chapter 2, we review some of the classical and DNN approaches for blotches detection and removal, including classical SDI detectors, ML3Dex for blotches interpolation and a newly DNN architecture designed for blotches detection. In Chapter 3, we artificially corrupt 1,000 clean images from 10 video clips and create our own dataset. In Chapter 4, a modified architecture based on an encoder-decoder [18] and a Unet are introduced for blotches detection. The layout of our blotches restoration system and the implementation of ML3Dex are introduced in Chapter 5. In Chapter 6, we painted out the problem RAFT would encounter when dealing with blotches and mitigate it by retraining RAFT. Finally concluding our works and future work will be covered in Chapter 7.

# CHAPTER 2

---

## Literature review

---

### 2.1 Classical Approaches

#### 2.1.1 Spike Detection Index (SDI) for blotch detection

Spike Detection Index (SDI) detectors are firstly introduced by Kokaram et al [11]. All pixels in one frame except blotches can either be found in the previous frame or be found in the next frame, only blotches will suddenly appear in one particular frame with outlying intensities on the surrounding local areas. This unique property of blotches enables SDI detectors to separate them from all the other areas in the frame.

To actually understand how SDI detectors work, we firstly assume changes of intensity on each pixel site during one frame time are due only to motions, not light effects, so the relations between pixel intensities of two continuous frames can be expressed as

$$I_n(x) = I_{n-1}(x + d_{n,n-1}) \quad (2.1)$$

where  $I_n(x)$  is intensity of the pixel at the location  $x$  in frame n, and  $d_{n,n-1}$  is the displacement mapping pixels in the current frame n into the previous frame n-1. In this model, pixel intensity in the current frame can be predicted by pixel intensity in the previous frame. Motion vector  $d$  is the only parameter we need in the model, it is obtained by motion estimator.

Therefore, the crucial clue for blotch detection is blotches spatial and temporal discontinuity on pixel intensities along motion trajectories. If motion is correctly estimated, the Displaced Pixel Difference (DPD)  $I_n(x) - I_{n-1}(x + d_{n,n-1})$  at each pixel should generally be small. But when a location is corrupted, the DPD would be high since the corruption destructure the model with extreme intensity value. However, this also happens when occlusion and uncovering occur (motion discontinuities). But because objects that are occluded must come from previous frames and objects that are uncovered must go to somewhere in following frames, the temporal discontinuity of occlusion and uncovering happens only in one direction. And for corruption it happens in both two directions so that this difference can help tell them apart. To express this in a mathematical form, the equation 2.2 is shown as followed

$$\begin{aligned} E_b &= I_n(x) - I_{n-1}(x + d_{n,n-1}) \\ E_f &= I_n(x) - I_{n+1}(x + d_{n,n+1}) \end{aligned} \quad (2.2)$$

For occlusion and uncovering, either  $E_b$  or  $E_f$  is high, for corruption, both  $E_b$  and  $E_f$  are high. This difference helps invent a simple blotches detector, the SDIa (Spike Detection Index - a) detector [11], which is defined as follows

$$b_{SDIa}(x) = \begin{cases} 1 & \text{for } (|E_b| > E_t) \text{ AND } (|E_f| > E_t) \\ 0 & \text{Otherwise} \end{cases} \quad (2.3)$$

where  $b_{SDIa}(x)$  is 1 if a blotch is detected and is 0 otherwise,  $E_t$  is a threshold defined by users.

In addition, one also observe a phenomenon that when blotches occur, their intensities are in the way outside intensity range of the surrounding area. This implies not only DPDs of a corruption are large, also signs of  $E_b$  and  $E_f$  should be the same. This discovery is added as an extra condition for SDIa model to detect a blotch and hence result in a new SDI model, the SDIp (Spike Detection Index - p) [10], it is defined in equation 2.4

$$b_{SDIp}(x) = \begin{cases} 1 & \text{for } (|E_b| > E_t) \text{ AND } (|E_f| > E_t) \text{ AND } sign(E_f) = sign(E_b) \\ 0 & \text{Otherwise} \end{cases} \quad (2.4)$$

After adding the extra sign judgment condition, SDIp detectors become more accurate than SDIa because it tightens requirements for detecting a blotch and hence reduces the possibility of wrongly detecting a blotch. The possibility is also called false alarm.

### 2.1.2 ML3Dex filter for blotch removal

After blotch detection, we need interpolation process to replace the intensity of blotch pixels with the best estimated values we found in other frames. To obtain the best intensity values for blotches removal, a median filter is needed. It helps exploit the information motion estimators give us and find the best values from it. One median filter that is broadly used and implemented is called ML3Dex. To well define ML3Dex median filter, a figure of 5 sub-filter windows is shown in Fig. 2.1.

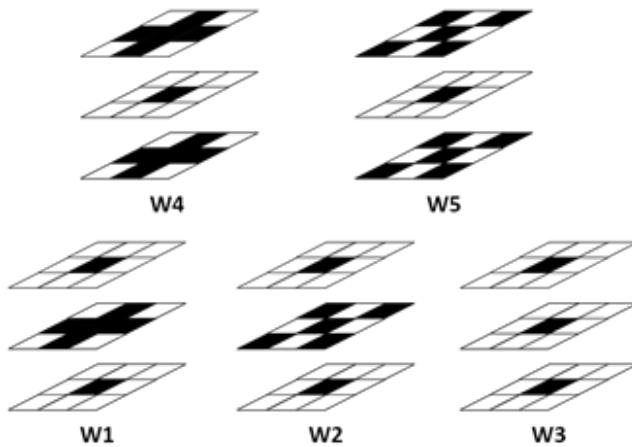


Fig. 2.1 *Sub-filter windows used in ML3Dex filter*

In Fig. 2.1, for each set of 3 windows, centre one is a  $3 \times 3$  window centered at the blotch pixel we focus on and needs to be replaced in the current frame. Above one is a  $3 \times 3$  window centered at the pixel where we predict the blotch pixel comes from in the previous frame. The below one is a  $3 \times 3$  window centered at the pixel where we predict the blotch pixel will be going to in the next frame. With the help of these sub-filter windows, ML3Dex filter can now be defined as follows

$$\begin{aligned} z_l &= \text{median}[W_l], 1 \leq l \leq 5 \\ \text{ML3Dex Filter output} &= \text{median}[z_1, z_2, z_3, z_4, z_5] \end{aligned} \quad (2.5)$$

where  $z_l$  is the median of intensities of all the black pixels in the  $W_l$  set of 3 windows. And ML3Dex Filter output is the median of all the  $z$  values. Because the intensity value of a blotch is always at the way outside the intensity range in the surrounding regions and also too far away from the intensity value it would have been, the intensity values of blotches will be at sides when median filtering. Therefore the final output we obtain from ML3Dex is very likely an intensity value of an intact, undamaged pixel which will soon be used for blotch removal. Even when dealing with large blotches, windows  $W_1$  and  $W_2$  will output blotch values, but the other three windows  $W_3$ ,  $W_4$  and  $W_5$  still output intensity values of intact pixels. Therefore the ML3Dex Filter output, the median value of  $z_1$ ,  $z_2$ ,  $z_3$ ,  $z_4$  and  $z_5$ , will still be an intensity value of an intact pixel which is appropriate for blotch removal.

## 2.2 DNN Approaches

### 2.2.1 Deep Neural Network

The history of deep neural network can be traced back to 1943, when Walter Pitts and Warren McCulloch created a computer model based on the neural networks of the human brain. They used a combination of algorithms and mathematics they called “threshold logic” to mimic the thought process. Now this method has been passed down to the present and explored by millions of people worldwide for developing new features and making more progresses. Over the decades, Deep Neural Network has made great achievements in many different areas including computer vision, speech recognition, natural language processing, machine translation and drug design. Plenty of DNN approaches had been proven more efficient than many of the classical approaches.

Deep Neural Network is an artificial neural network that imitates neural networks in human brain. It consists of plentiful different layers where the first is input layer, and the last is output layer. Those layers are made of neurons where each of them has its own properties such as its value and weights. To apply Deep Neural Network for solving problems, usually a target is set and the network will adjust its output to let it as close as it can to the target by changing the weights of neurons step by step. This process is called training. By this way artificial neural networks can learn as human neural networks to solve problems we throw to them.

Deep Neural Network also developed different types of architectures for optimally solving problems in different areas, such as Convolutional Neural Networks (CNN) in computer vision area, Recurrent Neural Networks (RNN) in Natural Language Processing

(NLP). DNN has been so successful in vision tasks, such as image classification, object detection, semantic segmentation, optical flow estimation and video classification.

### 2.2.2 Encoder-decoder architecture for blotch detection

A CNN-based encoder-decoder architecture is introduced [18] for blotches and scratches detection in archived videos. The encoder helps extract features from input images. It consists of three convolution layers that results in three outputs of 32, 64, 128 maps respectively. Those three outputs are then concatenated and results in a 224 maps mixing. The dimensionality of the mixing is then reduced by an upcoming 1x1 convolution layer and outputs a 64 maps. Finally, a Pooling layer is added with kernel size of 2x2, it reduces into half size the height and width of the image. The decoder helps turn features extracted by encoder into a detection map at the same size with the original image. It consists of an Upsampling layer followed by two deconvolution layers both with the spatial size of 3 x 3. In the end a convolution layer with 2 outputs followed by a Softmax layer is added to gain the final detection map in binary form. The whole architecture of the autoencoder is shown in Fig. 2.2.

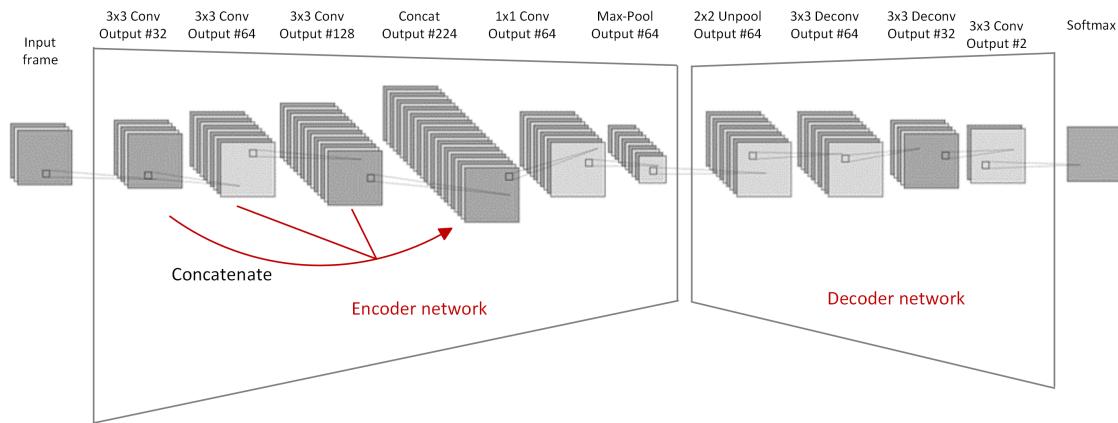


Fig. 2.2 *The proposed encoder-decoder architecture*

The whole network only contains one pooling layer due to the fact that blotches have a reduced spatial correlation and their sizes are relatively small. Yous et al also eliminate false alarms in post-processing stage to refine defects detection by considering the unique spatio-temporal properties of blotches and scratches. They fine-tune the encoder-decoder neural network by extracting pixel-level description from an intermediate stage of the network. Results [18] show the efficiency of fine-tuning on removing false alarms in many regions.

The DNN detectors are presented to be more accurate than classical detectors SDI, ROD and SROD [18, 15]. But it still has one problem that it does not take into account other frames except the current frame when training neural network. A blotch in one frame can be certainly defined as a blotch only when it compared with other frames. If a black hole appears in the same position with the same shape and size in all frames of one scene, then this hole is not a blotch, it is a real hole that exists in the real scene in the reality. By this logic, sending one current frame alone and hope neural network output the correct blotch detection results are unreasonable. It is not reliable even when we manually select blotches because there is no clue of showing if a blotch is a real blotch or a real hole that exists in the reality until we compare it with other frames. Therefore, to deal with this issue, we modify the architecture of the autoencoder [18] in Chapter 4 and finally achieve better results.

# CHAPTER 3

---

## Data preparation

---

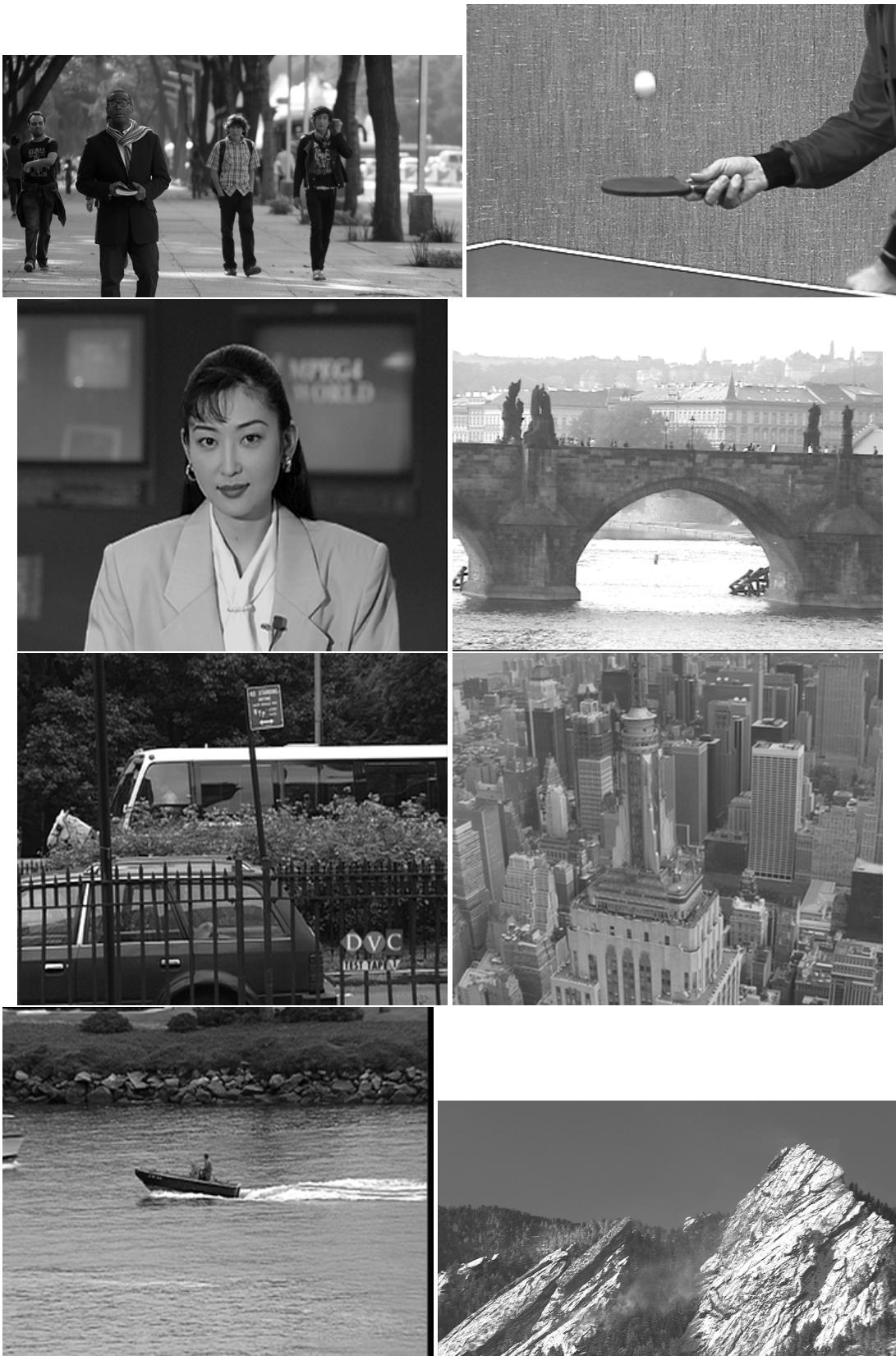
### 3.1 Artificial Dataset creation

For training and evaluating our DNN detectors, we create our own dataset by manually corrupting 1,000 clean frames from 10 video clips, and take those corrupted frames as inputs and their corresponding blotches position as ground truths.

For simplicity, all frames are converted into grayscale at intensities of 8 bits, and resized into resolutions smaller than 1024x540. After conversion, the basic information of these 10 image sequences is shown in Table 3.1. Frames from each of the 10 sequences are shown in Fig. 3.1.

Table 3.1 *Basic information of 10 Image Sequences*

No.	Sequence	Color Space	Resolution
1		Grayscale	1024x540
2		Grayscale	352x240
3		Grayscale	352x288
4		Grayscale	352x288
5		Grayscale	352x288
6		Grayscale	352x288
7		Grayscale	352x288
8		Grayscale	960x540
9		Grayscale	352x288
10		Grayscale	960x540



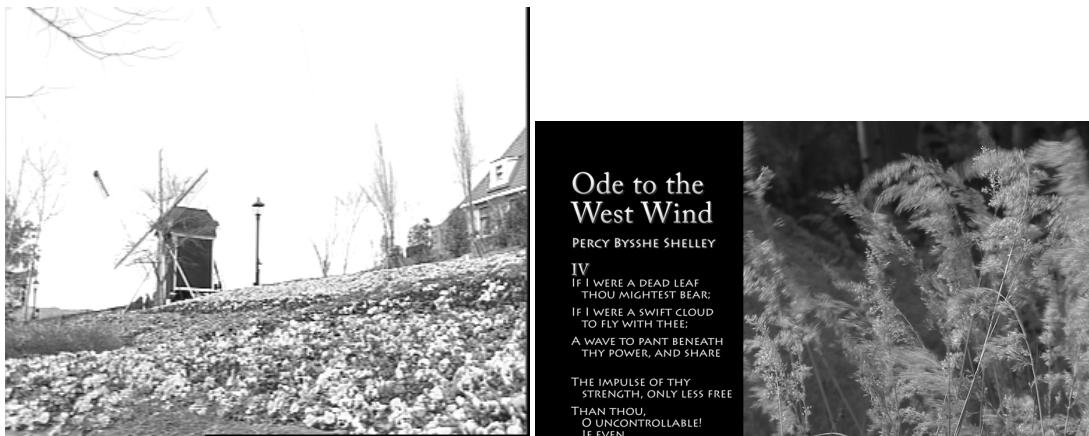


Fig. 3.1 One example frame from each of ten video clips

Those clean frames come from website<sup>1</sup>. The way we corrupt them is shown as the following Matlab code snippet

```
% import original frame
original_frame = imread('original_frame.png');
[rows, cols] = size(original_frame);
% generate corruption matrix (1 with blotch, 0 without)
corrupt_position = rand(rows, cols);
% 5*e-05 rate for each pixel to be corrupted
corrupt_position = corrupt_position * 10000 < 5;
% loop each blotch position
for i = 1 : rows
    for j = 1 : cols
        % find corrupted position
        if corrupt_position(i, j) == 1
            c = rand;
            if (c > 0.5)
                r = 1; % radius of corruption circle
            else
                r = 0;
            end;
            % corrupt circle by loop a square (x, y axis)
            % and avoid edges of the pic
            for square_x = max(1, j - r) : min(cols, j + r)
```

<sup>1</sup>[media.xiph.org/video/derf](http://media.xiph.org/video/derf)

```

        for square_y = max(1, i - r) : min(rows, i + r)
            % whether the pixel is within the circle
            if sqrt((j - square_x) ^ 2 + (i - square_y) ^ 2) <= r
                corrupt_position(square_y, square_x) = 1;
            end
        . . .
    end

    corrupt_frame = original_frame;
    % paint blotches on corrupt frame
    area = bwlabel(corrupt_position);
    num = max(area(:));
    for i = 1 : num
        index = find((area == i));
        % assign to each blotch an intensity value ranged from 0 to 255
        corrupt_frame(index) = (rand * 255);
    end;

```

Here we firstly create a corruption matrix at the same resolution with the original frame which indicates the initial pixel sites for corruption, those initial sites are chosen randomly with a probability which we can see from the code snippet is set to 0.05%. For each initial corruption site, we want to create a circle blotch with a random radian centered at the site, so inside the loop of each initial corruption site we assign other pixels within the circle blotch area as corruption sites by assigning 1 on those sites in the corruption matrix. Here comes an interesting phenomenon, we do not copy the corruption matrix, we add corruption sites in the corruption matrix within the loop of itself. Hence as more pixels within the circle are assigned as corruptions in the corruption matrix, more initial corruption sites are created and results in more loops for creating blotches hence widely spreads the corruptions.

This contagiously spreading tend to make blotches in the shape of a disorganized line bar streetching along from top-left to bottom-right. Some examples of this phenomenon is shown in red circles in Fig. 3.2.

In this corruption creation manner, if we set the radian of circle blotch for each initial corruption site in the corruption matrix as equal randomly from a range of 0, 1, 2. Corruptions will quiet widespread and in the end almost occupy the whole picture as shown in Fig. 3.3. If we set the radian as equally 0 or 1 like what we assign in the code snippet, the spreading will be under control. Some corrupted frames created in this manner are



Fig. 3.2 Four examples of stretching phenomenon in the artificial blotches

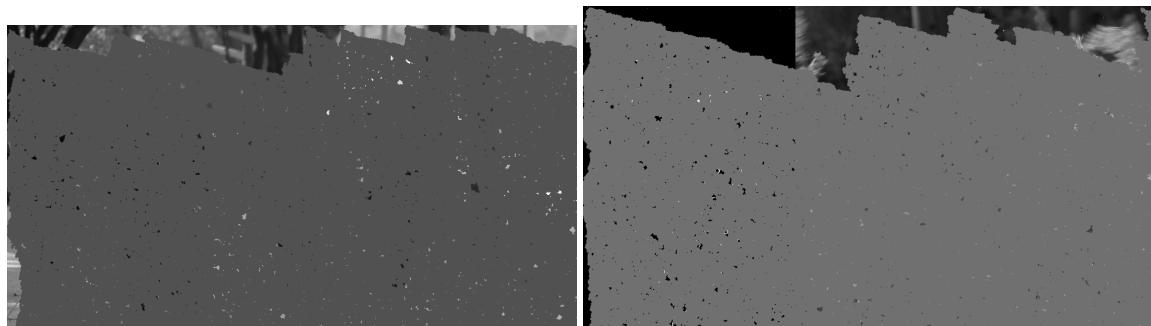


Fig. 3.3 Two examples of corruption explosion phenomenon

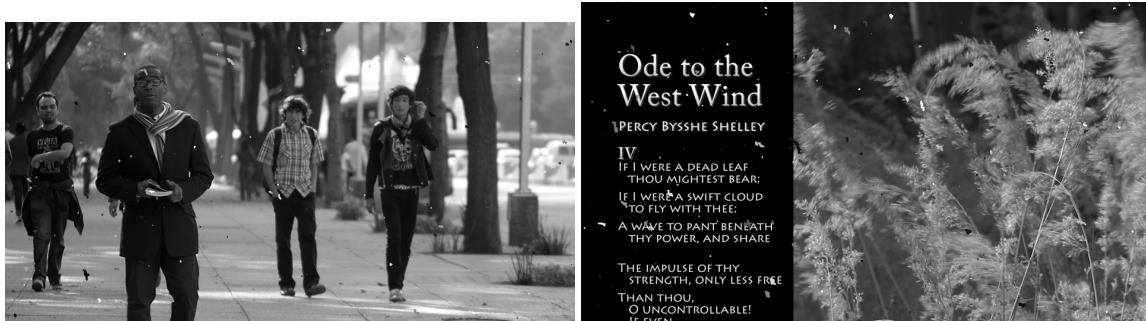


Fig. 3.4 *Two examples of under-controlled corruption spreading*

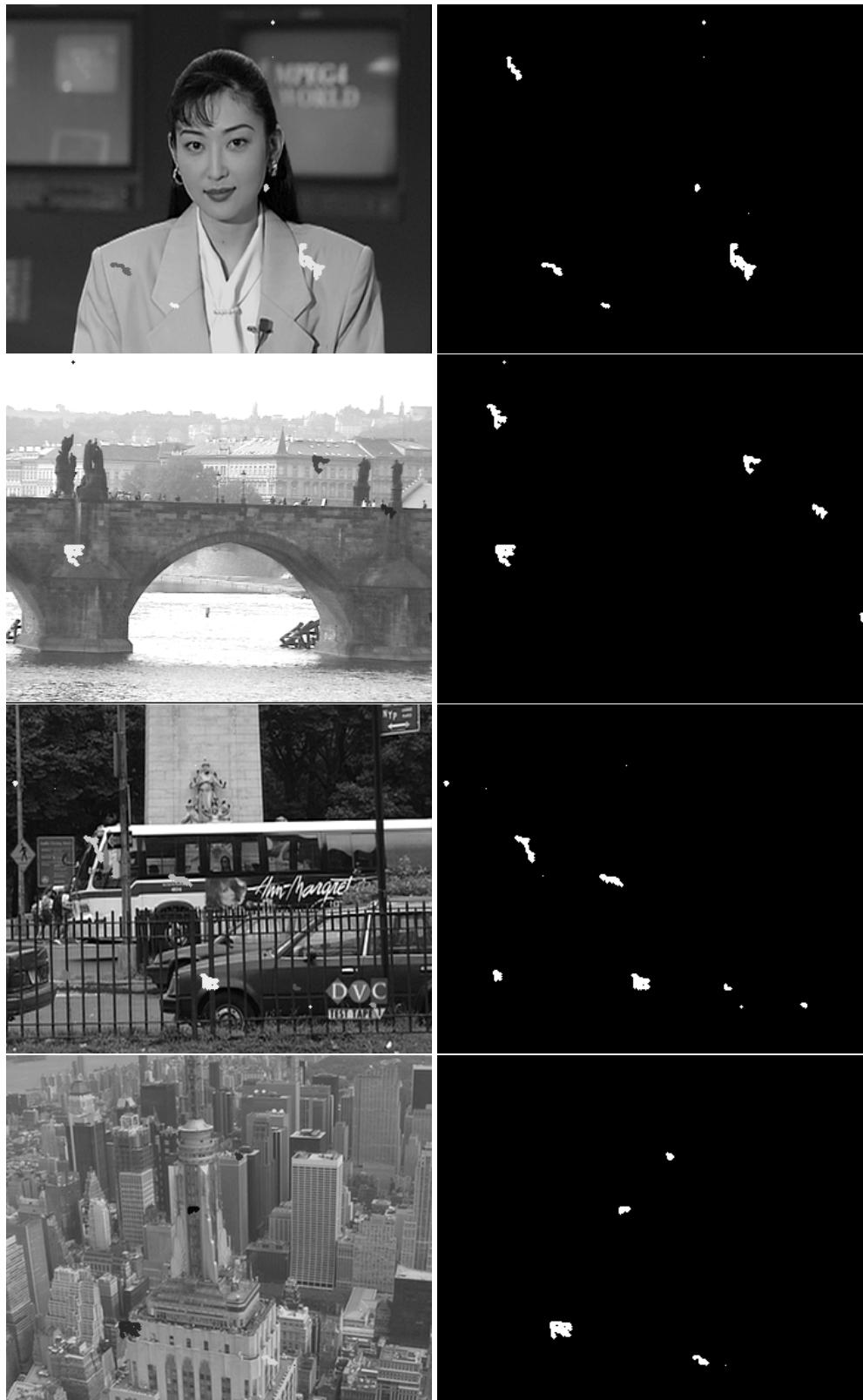
shown in Fig. 3.4. We also assign a random value ranged from 0 to 255 as the intensity of each blotch.

In Fig. 3.4, we can see the shape of blotches seems to be disorganized and random, they are different from the circle blotches we firstly want to implement. But these random shaped blotches are out of questions closer to the blotches in the reality. Therefore they are applied on blotch creation process. We also limit the created blotch size no less than 17 pixels because normally real blotches will not be too small. For a high resolution image, one pixel or five pixels blotches might not even be noticed.

In the end by the blotch creation method we introduced above, 1,000 clean frames are corrupted and then are stored. Examples of their corrupted frames and corresponding ground truths are shown in Fig. 3.5.

We can also change the probability of generating an initial pixel site to change blotches density in an image.





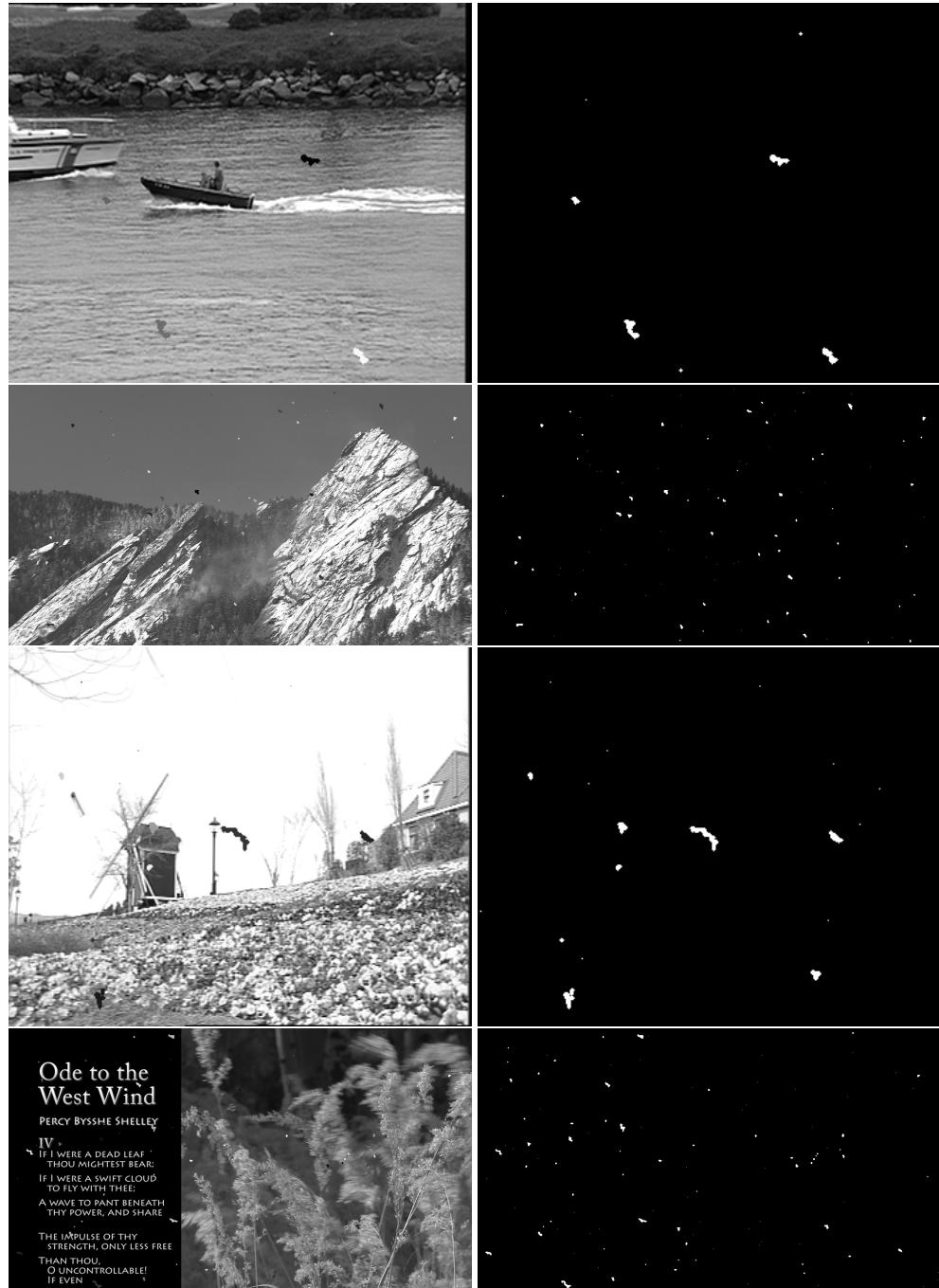


Fig. 3.5 Examples of ten corrupted frames (left) and their ground truths (right)

## 3.2 Adding motion information (RAFT)

We also prepare other corrupted frames with motion information added, those different types of corrupted frames are used for different training datasets. Through the code provided in the website<sup>2</sup>, we implement RAFT, run it in our own dataset and obtain estimated motions, or precisely displacements, on each of the pixel sites between two consecutive frames. We then compensate the displacements by relocating all the pixels in one frame and hence reconstruct the other frame. The reconstructed frame is also called Motion Compensated (MC) frame. If we backwardly estimate the motions from frame n to frame n-1, and use those motions as well as frame n-1 to reconstruct frame n, then this reconstructed frame is called backward MC frame. The same logic goes with forward MC frame.

Another type of corrupted frames added with motion information is called MC Displaced Frame Difference (DFD). Those MC DFDs are difference frames between current frame and its motion compensated frames, backward MC Displaced Frame Difference (DFD) is obtained using backward MC frame, forward MC Displaced Frame Difference (DFD) is obtained using forward MC frame. There are, also, non-MC DFD frames which are obtained when we directly subtract other frames from current frame. For backward non-MC DFD frame, we subtract previous frame from current frame.

Examples of these two types of motion added frames are shown in Fig. 3.6. As can be seen, compared to the non-MC DFD frame in the bottom-left, the MC DFD frame in the bottom-right shows us the great reduction of DFD motion compensation attains.

---

<sup>2</sup>[github.com/princeton-vl/RAFT](https://github.com/princeton-vl/RAFT)



Fig. 3.6 Examples of backward motion added frames. Top row : Frames 18, 19 from the corrupted sequence. Middle row : Motion superimposed frame (left), MC frame (right). Bottom row : Non-MC DFD frame (left), MC DFD frame (right).

# CHAPTER 4

---

## System for detection of dirt and sparkle

---

### 4.1 Modified Autoencoder

A neural network with CNN-based encoder-decoder architecture for blotch detection is introduced [18], it had achieved better results than traditional SDI, ROD and SROD detectors which proves the advantages for applying neural networks on blotch detection area. However, the neural network still has flaws, the main one is it only takes one current degraded frame as input. But according to what we discussed in Chapter 2, one blotch can only be deemed as a blotch when it compared with other frames. A blotch-like patch in an image might actually exist in the reality, we do not know until we compare it with other frames. If the patch only exist in one frame, then it is a blotch. If it shows in all continuous frames of the same scene, then it is not a blotch but an actual blotch-like patch that exists in the reality.

Therefore, it is unreasonable for the approach [18] to take one current frame alone as input for detecting blotches. At this end, we modify its architecture from accepting one input to accepting three inputs. The decoder remains the same, and the encoder is modified into the one shown in Fig. 4.1.

In Fig. 4.1, the encoder is similar to the one in Fig. 2.2. The only difference is we repeat parts from the input layer to the 4th convolution layer twice, each for accepting one more input frame. We then concatenate those 3 output features and do the following convolution and max-pooling operations like what is done in the previous autoencoder. After doing this, the autoencoder can accept two more frames as inputs for blotches

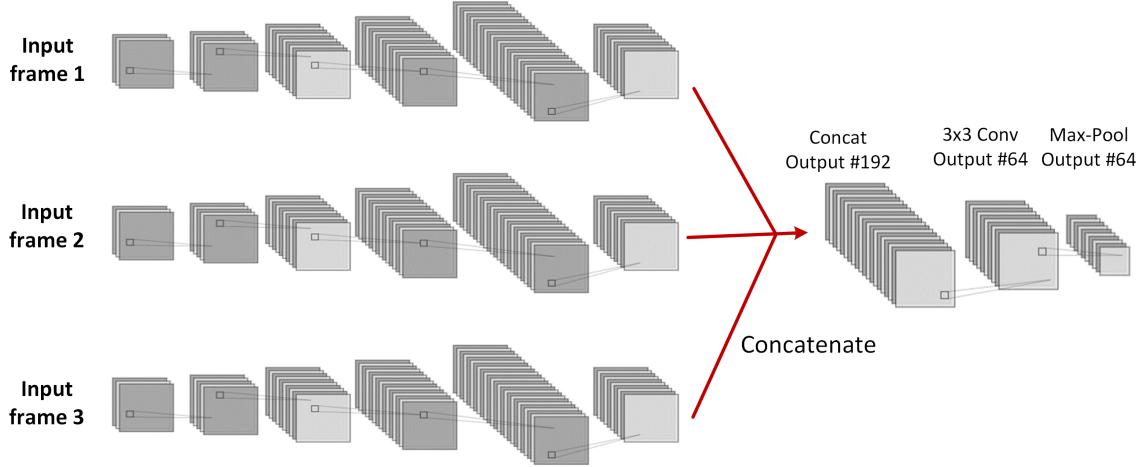


Fig. 4.1 *The architecture of modified encoder*

detection. The input of 3-frames, according to the datasets we prepared, can be any of the following 3 triplets:

1. Previous corrupted frame  
Current corrupted frame  
Next corrupted frame
2. Backward MC corrupted frame  
Current corrupted frame  
Foreward MC corrupted frame
3. Backward DFD corrupted frame  
Current corrupted frame  
Foreward DFD corrupted frame

Training modified autoencoder using different datasets will result in different models. We name those three models, in turns, as 3-frame Autoencoder (no motion), 3-frame Autoencoder (MC) and 3-frame Autoencoder (DFD), where MC means motion compensated, DFD means displaced frame difference. Modified autoencoders are trained under the same configurations when training the original autoencoder. We then obtain models and calculate their recalls and false alarms based on our own dataset. The comparison between modified and original autoencoders is shown in the form of ROC curves in Fig. 4.2.

In Fig. 4.2, DFD and non-motion information models perform better than original autoencoder, where recalls are higher and false alarms are lower. This means neural

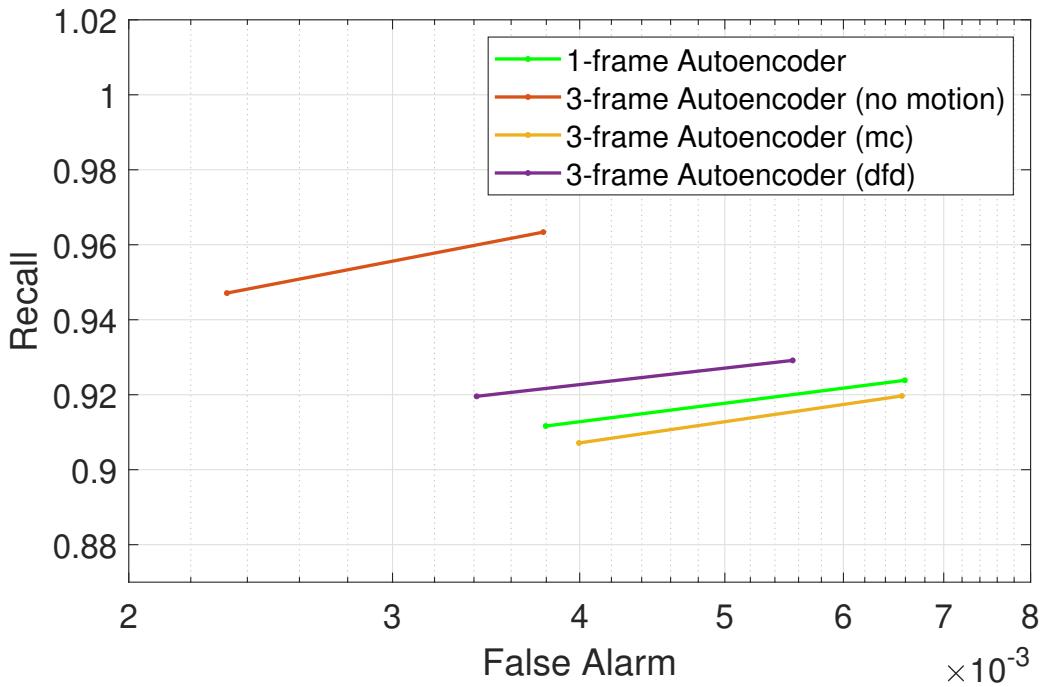


Fig. 4.2 ROC curves of 4 autoencoders.

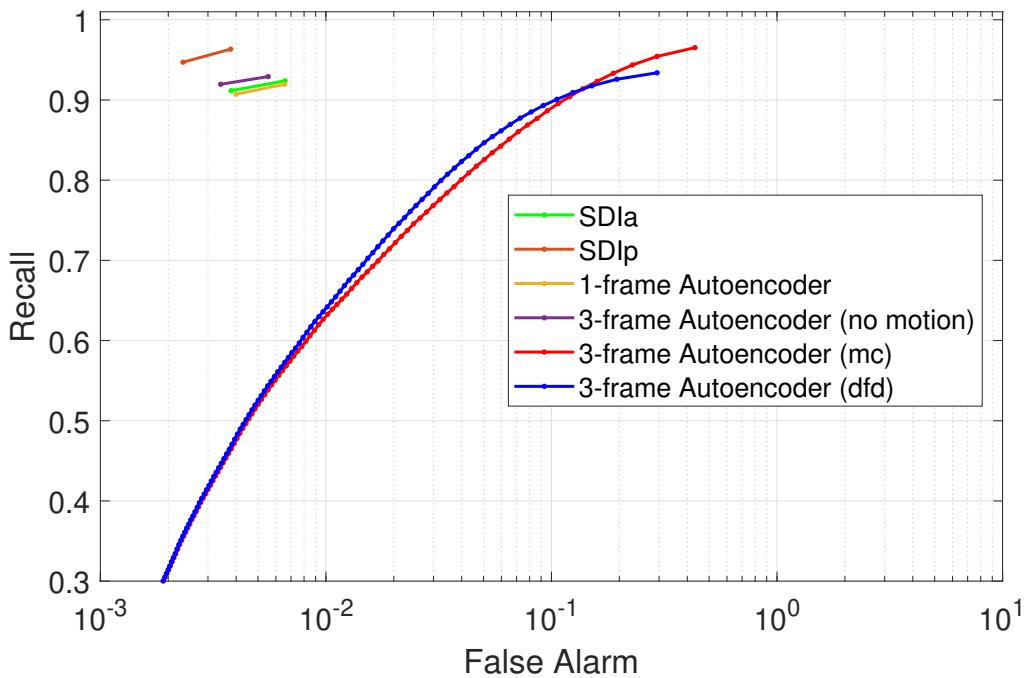


Fig. 4.3 ROC curves of autoencoders compared with SDI detectors.

networks discriminate blotches better given more information exploited. But MC model performs worse than the original one. This is because in motion compensated frames errors caused by inaccurate motions confuse neural networks, lead them to make wrong decisions and finally present a worse performance. Because adding motion information is the only difference between MC model and non-motion information model that makes MC model perform worse than non-motion information model.

We also implement classical detectors SDIa and SDIp and compare their results with the autoencoders using ROC curves. The comparisons can be seen in Fig. 4.3. It is easy to discover DNN models perform way better than classical SDI detectors.

## 4.2 Unet

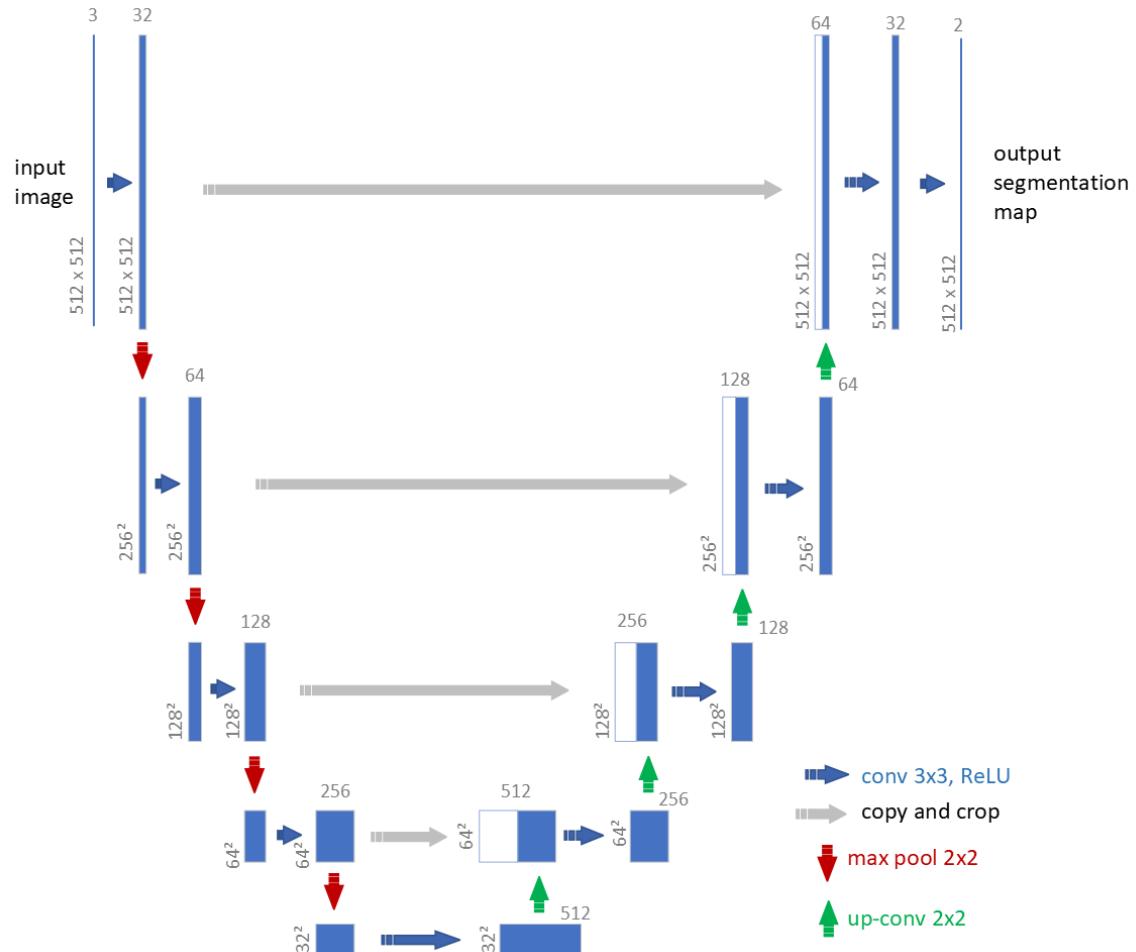


Fig. 4.4 The designed architecture of our Unet

Blotches detection is actually a type of image segmentation problems. Encoder-decoder is one of the most common architecture for tackling image segmentation problems. It has been proved by us to be efficient as well on blotches detection. Therefore, it is worthy to try out Unet, an improved version of autoencoders for tackling image segmentation problems, on blotches detection problem. Inspiring by the architecture of Unet presented in [13], we design our own Unet as shown in Fig. 4.4.

Our Unet takes as input a corrupted color image and outputs a binary segmentation map showing the positions of estimated blotches. It consists of 10 convolution layers, 4 maxpooling layers, 4 deconvolution layers and 4 copy, crop operations. The Unet is trained and evaluated under the same configurations we set for autoencoders. Its ROC curve is obtained and compared with other DNN models as shown in Fig. 4.5.

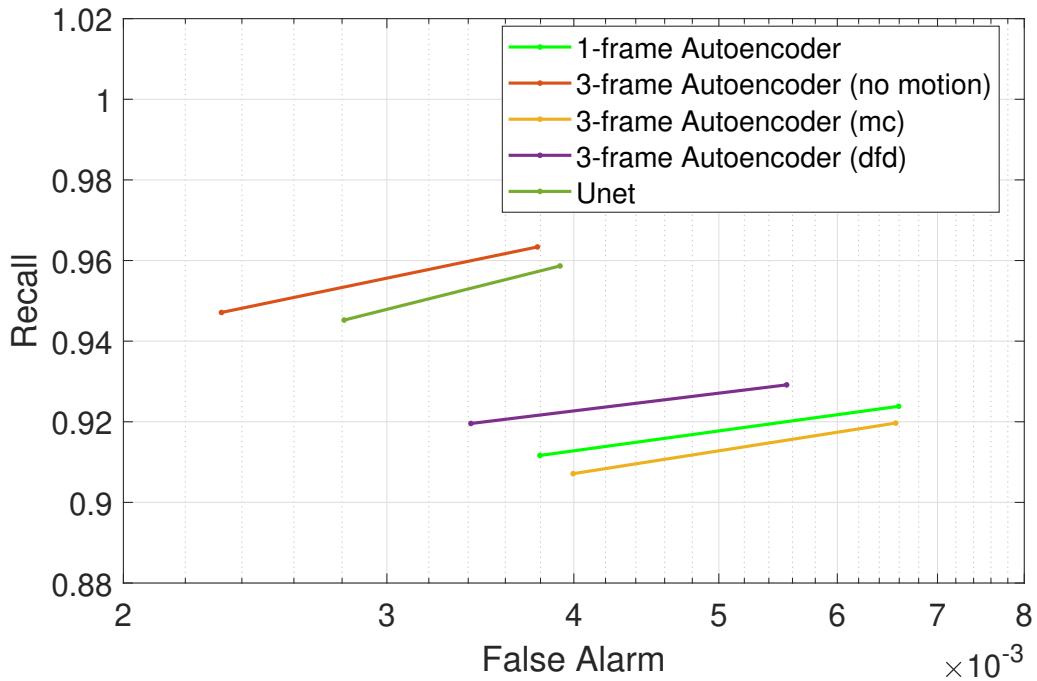


Fig. 4.5 ROC curves of Unet and autoencoders.

In Fig. 4.5, the Unet is found to be better than most of autoencoders except non-motion one. It proves the advantages of Unet on tackling not only image segmentation but also blotches detection problem. In addition, during the process, we find that Unet is trained more than twice faster than autoencoders. For example, training one epoch of 800 images at the batch size of 8 for the Unet normally needs 550 seconds while training one epoch under the same configurations for autoencoders roughly needs more than 1200 seconds. A graph that shows ROC curves of all models we built including SDIa, SDIp, four

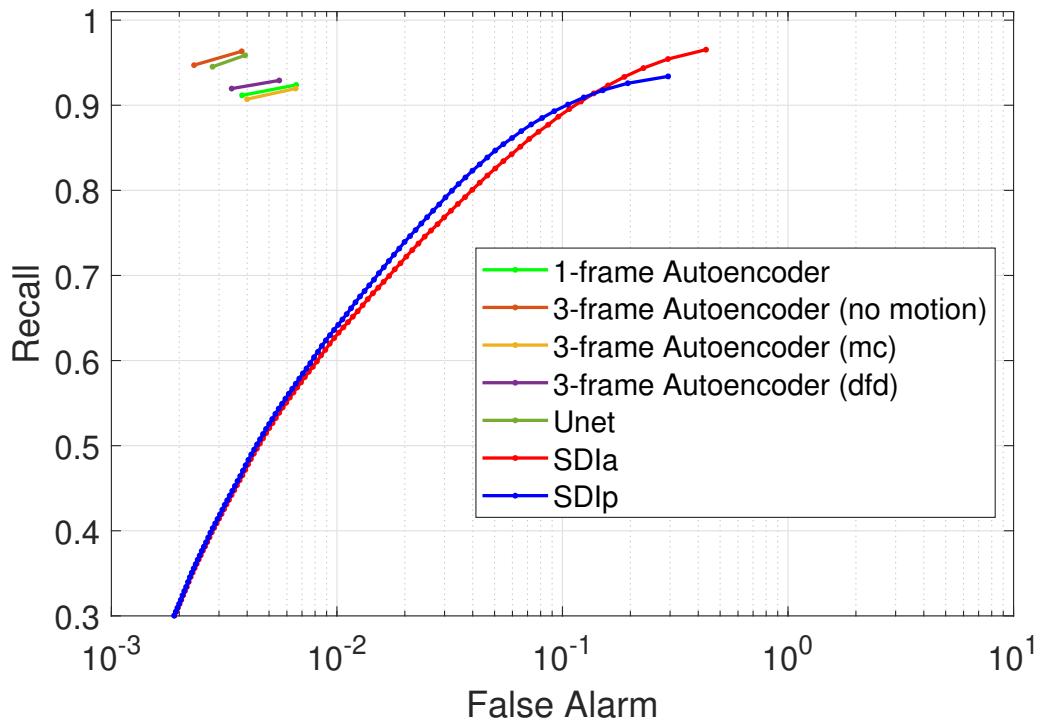


Fig. 4.6 ROC curves of SDI detectors, autoencoders and Unet.

autoencoders and Unet is shown in Fig. 4.6. And examples of 6 detection results from 6 different models are shown in Fig. 4.7.



Fig. 4.7 *Detection results on frame 19 of sequence Akiyo. Top row : Frames 18, 19, 20 from the corrupted sequence. Middle row : Detections using previous work Yous et al [18] (left), SDIa/SDIp (middle, right). Bottom Row : Results from three of the systems designed in this work: UNET (left), 3-Frame autoencoder no motion (middle), 3-Frame autoencoder DFD (right). The 3-Frame autoencoder with no motion shows the best performance with recall = 94.71% and false alarm = 0.2326%.*

# CHAPTER 5

## System for detection and removal

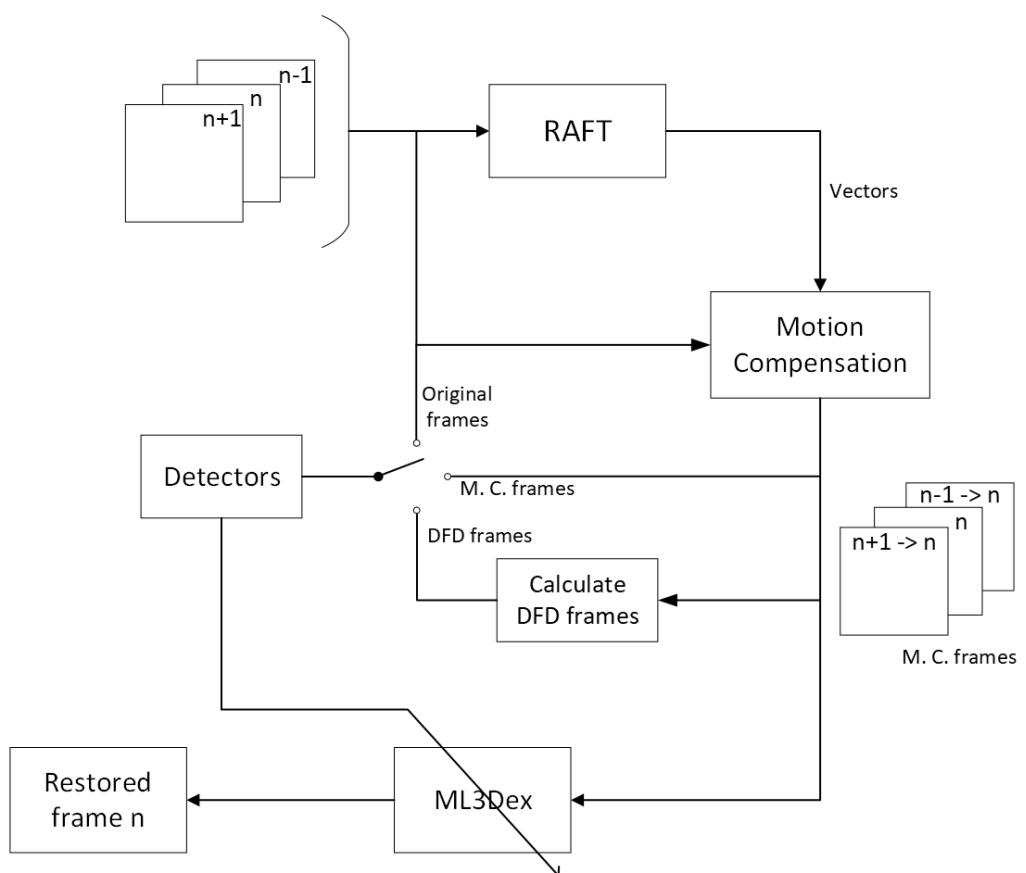


Fig. 5.1 Blotch restoration system built as a cascade of detection and reconstruction modules.

## 5.1 A blotches restoration system

In this chapter we present a complete blotch restoration system built as a cascade of detection and reconstruction modules. The system is shown in Fig. 5.1. In this work the reconstruction module is implemented as a 3D median filter which is switched on and off at location detected as blotches by the Detector. We investigate a number of different modules for detection including classical and deep neural network approaches.

The idea is that one of the main challenges in blotch removal is the detection step, and having built a confident detector we can reduce the problems associated with incorrect detection. And now we switch our focus on blotch removal where a 3D median filter ML3Dex is used. The ML3Dex is very similar to the early ideas in blotch removal proposed by Kokaram et al [10].

## 5.2 The implementation of ML3Dex

For blotch removal, or blotch reconstruction, ML3Dex is introduced and implemented. For better explaining the implementation, five sub-filter windows used for illustrating ML3Dex working principle in Chapter 2 are shown here again in Fig. 5.2.

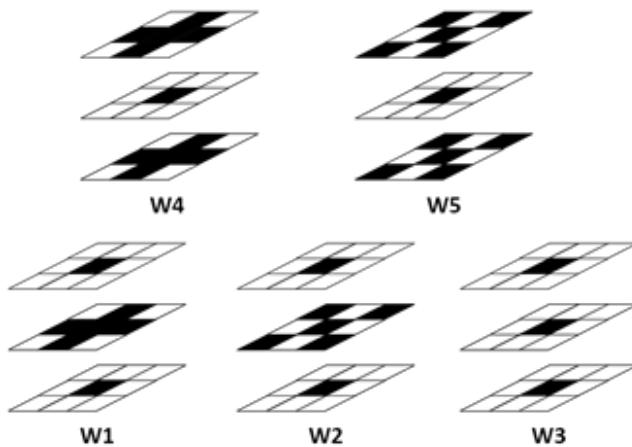


Fig. 5.2 Sub-filter windows of ML3Dex filter

In Fig. 5.2, on each of 3 windows, the above and bottom one can be found respectively in previous and next frames given motion information attained through RAFT. In Matlab, interp2() function is used to compensate backward motions in previous frame to reconstruct current frame, it results in backward motion compensated frame. And it is easy to discover that each pixel in backward motion compensated frame is the center of the above

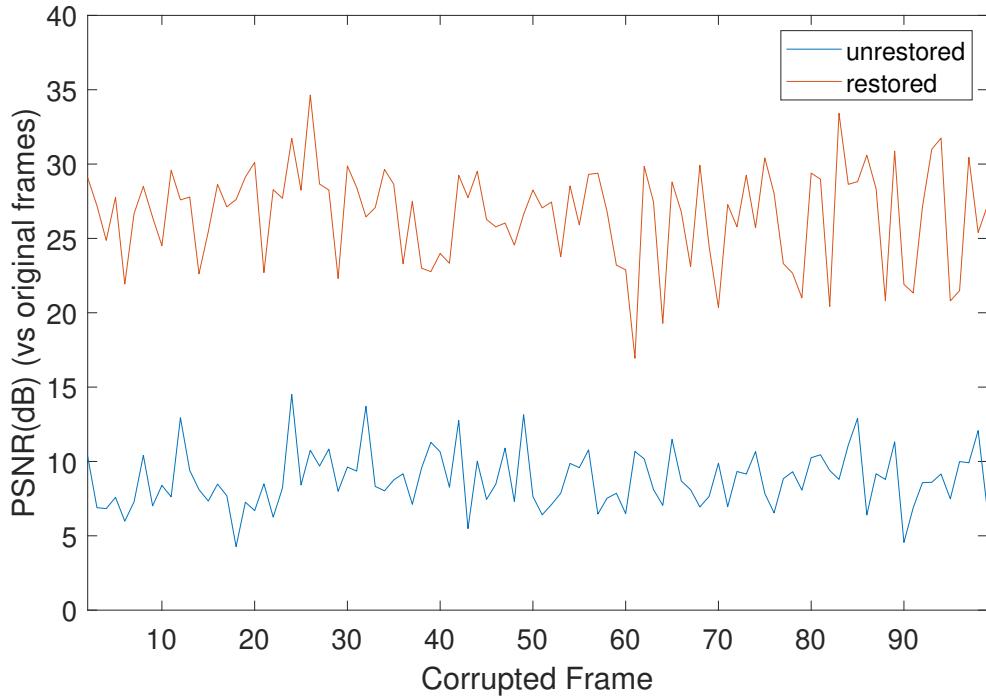


Fig. 5.3 PSNR of 98 restored frames in Sequence 1 compared with unrestored frames.

3x3 window to the blotch pixel at exact same coordinate in current frame. The same logic goes with forward motion compensated frame. Therefore, by this way we can find all the five sets of three 3x3 windows and do median filtering as what we introduced in Chapter 2. And we finally obtain the best intensity value and use it for replacing the corresponding blotch pixel site then restore the blotch.

### 5.3 Evaluation of blotches restoration

The effectiveness of our restoration system shown in Fig. 5.1 can be generally estimated in Fig. 5.3. It shows the PSNR improvement of 98 frames in first sequence after blotches restoration. The restoration applies Unet for detection and ML3Dex for blotches removal. Here PSNRs are only measured on blotches area, not the whole picture. This is why PSNRs of unrestored frame are so low. By comparing two PSNR curves in Fig. 5.3, we notice restored frames are much closer to the original frames than before. One example of the effort of blotches restoration can be seen in Fig. 5.4 which shows the restored frame and the DFD frame before and after restoration.

In Fig. 5.1, different blotches detectors are used for blotch detection. We know the errors of detectors will affect the performance of blotch restoration because it tells ML3Dex

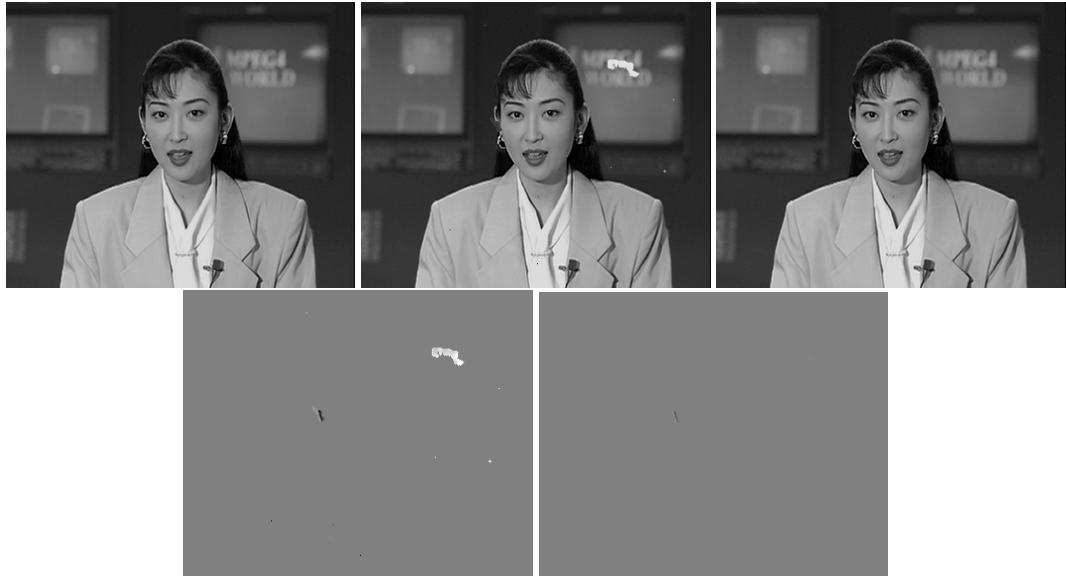


Fig. 5.4 Examples of restored frame and its DFD. Top row: original frame (left), corrupted frame (middle) and restored frame (right). Bottom row: DFD frame of corrupted frame (left), DFD frame of restored frame (right).

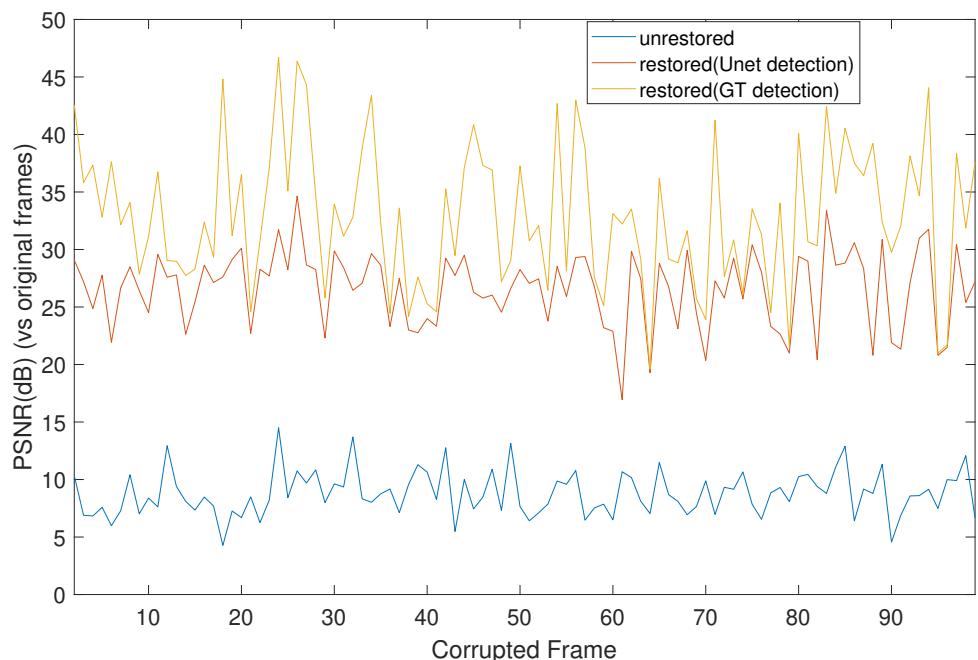


Fig. 5.5 PSNR of 98 restored frames in Sequence 1 of Unet detection and Ground Truth detection.

where to restore blotches and where not to. To quantify the flaws brought by Detectors on blotches restoration, we restore corrupted frames on both perfect detections and detections obtained by Unet. And we compare those two sets of restored frames with original frames, measure both of their PSNRs and show the result in Fig. 5.5. Here the perfect detections we used are ground truths on training detection neural networks.

As can be seen, the gap between two curves in Fig. 5.5 shows the improvement can be made from blotches detection. But not just blotch detection, blotches removal process can be better as well. ML3Dex chooses the best intensity value from motion information provided by estimator and do median filtering. The way ML3Dex interpolates does not seem to be having problems, but the motion information it uses is not so. Motion estimators always have problems on dealing with blotches area which can be seen in next Chapter that even now one of the most state of the art estimator RAFT still has many flaws when encountering blotches. But ML3Dex needs motion information for analysis, which makes its performance greatly depend on accuracy of estimators. So here for improving blotches removal process, we focus on the enhancement of robustness of motion estimators on blotches area. This work will be done in next Chapter.

# CHAPTER 6

---

## Design of robust motion estimator

---

### 6.1 The problem facing motion estimators on blotches

As we discussed in Chapter 5, the performance of ML3Dex greatly depends on the accuracy of motion estimator on blotches area. However the estimator RAFT which we used in this project still has many flaws when tackling blotches, those issues can be observed in Fig. 6.1, which shows one example of two backward optical flow images obtained respectively from corrupted inputs and clean inputs.

The differences between two optical flows in Fig. 6.1 show us damages made by blotches. For example the area in the blue circle in the left optical flow image that is detected as almost no motion by RAFT can be mistakenly detected as having large motion when a blotch appears at the site. The mistake is shown in the red circle in the right optical flow image.

The difference between two optical flow tells us the places we can improve. Because the two images should be the same if the blotches do not affect anything. Our aim is to make optical flows (motion vectors) from corrupted inputs as close as possible to the ones from clean inputs. And we will approach this by retraining RAFT in the following content.



Fig. 6.1 *An example of RAFT estimation flaws on blotches. 1st column: original frame-8 (top), original frame-9 (middle) and their optical flow image (bottom). 2nd column: corrupted frame-8 (top), corrupted frame-9 (middle) and their optical flow image (bottom).*

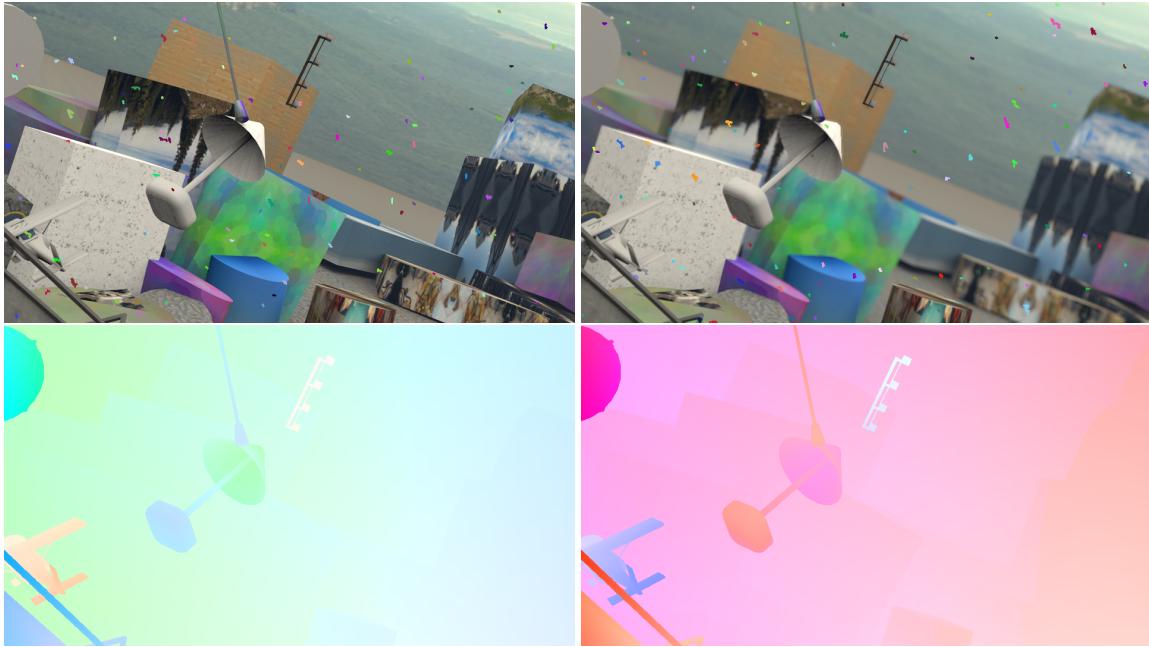


Fig. 6.2 Examples of artificially corrupted official dataset (1st row: cleanpass (left), finalpass (right)) and their clean optical flows (2nd row: backward (left), forward (right))

## 6.2 Retrain RAFT and its results

To increase the robustness of RAFT on blotches, we retrain RAFT by taking as inputs the corrupted frames and as ground truths the clean motion files. We firstly download official FlyingThings3D dataset from website<sup>1</sup> and randomly choose a thousand triplets in the dataset that includes cleanpass, finalpass images and their shared motion files. We then artificially corrupt those cleanpass, finalpass images in the way we did in Chapter 3 and take those corrupted frames as inputs, their former motion files as ground truths for retraining. Examples of a corrupted cleanpass, finalpass frames and their corresponding optical flow images converted from motion files are shown in Fig. 6.2.

In Fig. 6.2, blotches are similar to the blotches we created in Fig. 3.5 in Chapter 3. The official raft-things.pth model downloaded from website is retrained by us using our retraining dataset under the configurations set in train\_mixed.sh. The modified raft-things model is obtained and the optical flow outputs from modified model are acquired. Examples of two backward optical flow images from the original and retrained RAFT models of clean inputs and two backward optical images from two RAFT models of corrupted inputs are shown in Fig. 6.3

<sup>1</sup>[academictorrents.com/userdetails.php?id=9551](http://academictorrents.com/userdetails.php?id=9551)



Fig. 6.3 Examples of two optical flows from original RAFT (1st row: clean input (left), corrupted input (right)) and two optical flows from retrained RAFT (2nd row: clean input (left), corrupted input (right)).

When dealing with corrupted input, optical flows are initially not desirable from the original RAFT as shown in the top-right in Fig. 6.3. The estimated motion is largely affected by blotches as showing huge difference between optical flow from clean input (top-left) and corrupted input (top-right), especially in the red circles. But after retraining, optical flow image of corrupted input improves a lot as shown in bottom-right in Fig. 6.3. The effect of blotches becomes not that significant since lots of abnormal patches caused by blotches disappear in optical flow as shown in green circles in the bottom-right. Although there are still some patches that have not been removed like the one in the brown circle in the bottom-right, there is still space for RAFT to improve here. Even the non-blotch areas become more accurate because they are closer to the top-left optical flow compared to the top-right optical flow. Though they are not our focuses here. All those indicate our RAFT becomes more robust towards blotches after retraining.

To quantify it, we take as ground truths the optical flows of clean inputs from the original RAFT and measure the closeness (PSNR) to them of two types of optical flows of corrupted inputs from the original and retrained RAFT. We assume that the closer the optical flows to ground truths, the more accurate the motion vectors. The PSNR improvement of optical flows of corrupted input can be seen in Fig. 6.4. Here the result shows in 99 backward optical flows (No.2 to No.100) in Sequence 1 with average PSNR arrived at 32.7866dB in retrained RAFT and 25.7641dB in original RAFT.

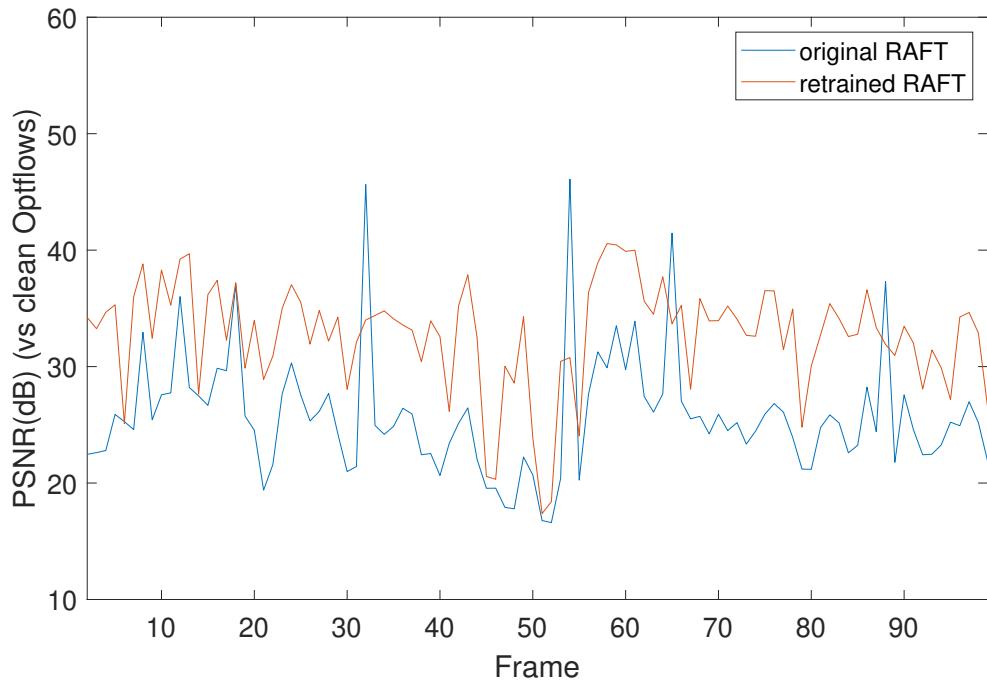


Fig. 6.4 PSNR improvement of 99 backward optical flow images (motion vectors) in Sequence 1 after retraining.

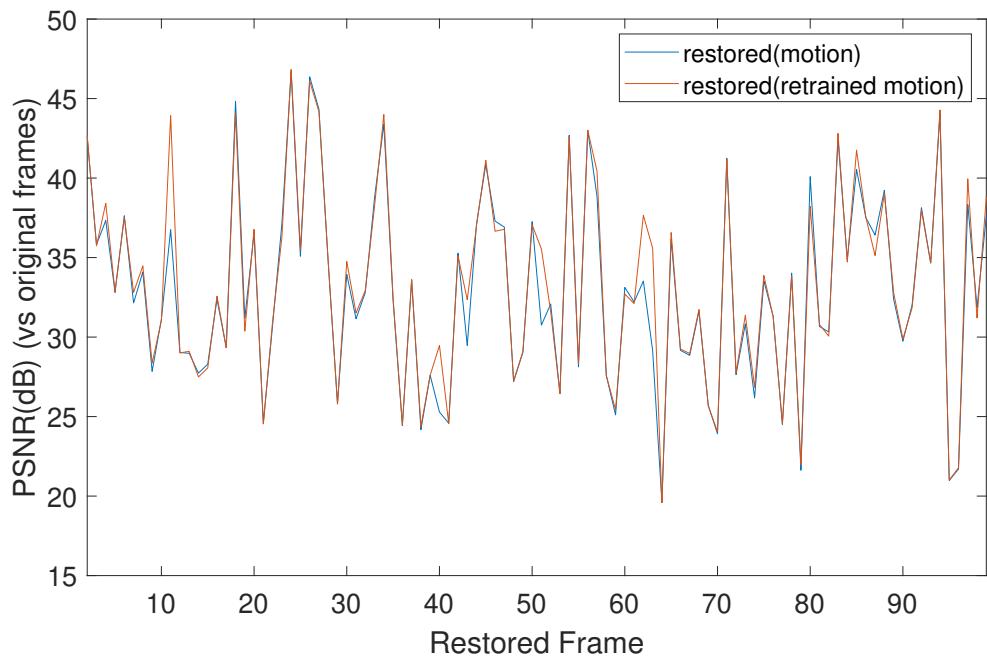


Fig. 6.5 PSNR improvement of 98 restored frames in Sequence 1 after using new motions from retrained RAFT.

We also restore corrupted frames with motions estimated by retrained RAFT. Compared to the original frames, we calculate PSNRs of restored frames using motions respectively from the original and retrained RAFT, PSNRs are shown in Fig. 6.5.

In Fig. 6.5, the gap between two curves illustrates the improvement we make on blotches restoration by retraining RAFT. We can see, for some frames, retrained motion help improve restorations, with higher PSNRs. On average, mean of PSNRs from retrained motion is 33.2835dB while unretrained motion results in a value of 32.9237dB. As a reminder, here we show the efforts that can be done in blotches detection and that are done in removal process for blotches restoration. Those are presented as PSNR improvements of restored frames in Fig. 6.6.

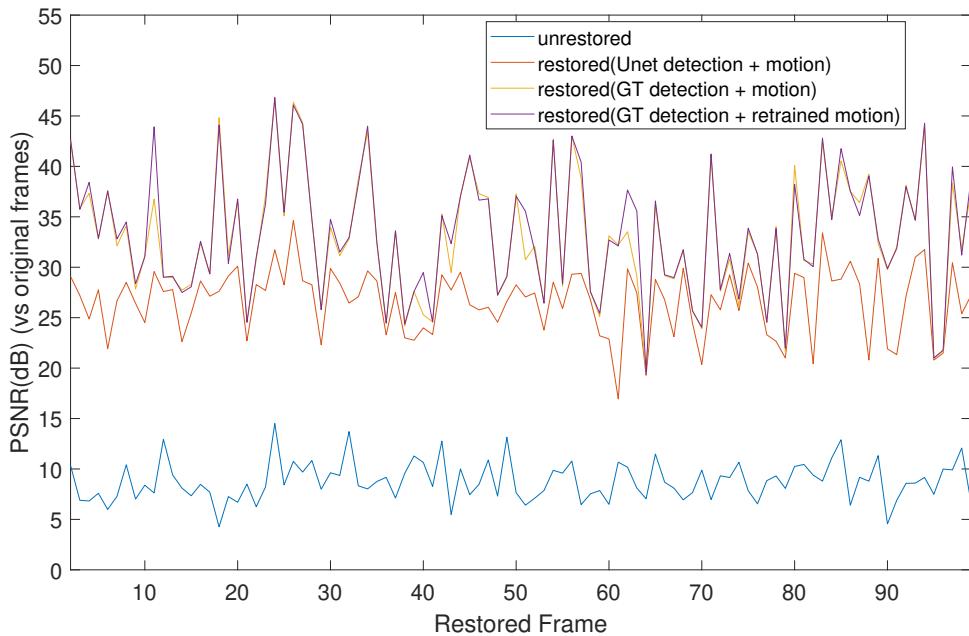


Fig. 6.6 *Improvement of PSNR of restored frames from blotches detection and removal on blotches restoration.*

We also test our retrained model on real degraded images. Several output examples from real degraded frames are shown in Fig. 6.7 and Fig. 6.8. Each of the two shows corresponding degraded frame n-1, n and their backward optical flow images from original and retrained RAFT models.

In Fig. 6.7 and Fig. 6.8, we observe that the retrained RAFT model does not improve its robustness in real blotches as much as what it did in our artificial blotches. Some incorrect estimated motions in blotch areas still show incorrectly after using retrained RAFT model. We attribute the invalidity of the retrained model towards real blotch areas



Fig. 6.7 (1st) Comparison of optical flows from original RAFT and retrained RAFT models of real degraded frames. 1st row: degraded frame  $n-1$ ,  $n$ . 2nd row: backward optical opflow images (from original RAFT (left), retrained RAFT (right)).



Fig. 6.8 (2nd) Comparison of optical flows from original RAFT and retrained RAFT models of real degraded frames. 1st row: degraded frame  $n-1$ ,  $n$ . 2nd row: backward optical flow images (from original RAFT (left), from retrained RAFT (right)).

to the dissimilarity between real blotches and our artificial blotches. One difference is intensity, each of the blotches we create has constant intensities while each of the real blotches has an intensity varied within a certain range. Another difference is shape, our blotches tend to be an irregular patch going southeast while real blotches can be any shape with no rules at all.

Therefore our retrained RAFT model can recognize our artificial blotches and modify their motions as we expect, but cannot recognize some real blotches which are essentially different from the artificial ones. For those blotches retrained RAFT processes them the same way before retraining. However it is promising that if we can model those real blotches well enough, we will be able to deal with real blotches by retraining RAFT.

# CHAPTER 7

---

## Discussion and Final Comments

---

Our aim in this project is to remove blotches from archive movies. And in the end we build a complete blotches restoration system which handles the task well. It consists of blotches detection section and blotches removal section where each has been improved by either introducing new methods or improving old methods.

The project began by reviewing previous work in the area. In this dissertation we briefly introduced previous work on blotches detection and removal area. The work was separated into Classical and Neural Network approaches. We reviewed the classical SDI methods for detection and ML3Dex median filter for interpolation of the patches. There are relatively few neural networks or other recent technologies introduced specifically for blotch detection [18, 4]. Those methods used CNN-based encoder-decoder architecture and visual saliency map and did not incorporate motion at the input stage.

We firstly prepare and corrupt clean frames downloaded from website and build our own dataset which helps training of detection DNNs and testing of our blotches restoration system. It also can be taken as a reference of database in relevant areas.

We then start to build our blotches restoration system. Firstly we focus on blotches detection area and implement some of previous works including classical SDI detectors, autoencoder detector [18]. Those previous works are trained and evaluated in our own dataset. We also create two detectors of our own, one is multi-inputs autoencoders obtained by modifying the number of inputs of the autoencoder [18], the other is a noval Unet detector. Both of those two achieve good results, our modified autoencoders are proven to be better performed than the original autoencoder in our dataset, our Unet

detector is the best-performed detector over all single-input detectors we presented in this work. The ROC curves of all the above detectors evaluated in our own dataset can be seen in Fig. 4.6, with best-performed 3-frame autoencoder (no motion) resulting in a recall of 94.71% and a false alarm of 0.2326%.

After detection, we shifted our attention to blotches removal area. A classical 3D median filter ML3Dex is introduced and implemented, it interpolates blotch areas with other patches and hence remove them. We also employ RAFT, one of the most state-of-the-art motion estimators, to provide the motion vectors ML3Dex needs.

Therefore, a complete blotches restoration system is established with DNN and many other detectors for blotches detection and ML3Dex for blotches removal. The system is presented in Fig. 5.1. It is evaluated in our dataset as shown in Fig. 5.3 where the closeness (PSNRs) between restored, unrestored frames and original frames are shown and compared.

The blotch removal performance of ML3Dex is also improved by ameliorating motion vectors it uses. The flawed motion vectors in blotches estimated by original RAFT are significantly improved after retraining. It means the retraining process makes RAFT more robust in blotches. When motion vectors get improved, ML3Dex can access better motion information for its analysis, which makes blotch removal performance better. The improvement of optical flow images (reflect motion vectors) and restoration after retraining can be seen in Fig. 6.4 and Fig. 6.5.

## 7.1 Future work

By discovering the necessity of multi-frames comparison for blotches existence, we invent 3-frame autoencoders in wishes of providing necessary and more sufficient information resulting in higher accuracy. We prove the effectiveness of providing more frames for blotches detection as the 3-frame autoencoder (no motion) is the best-perform detector over all the detectors we implement, including Unet detector. But motion information which we use as expression of MC frames and DFD frames is not as useful as we expect in the way we did for training DNNs for 3-frame autoencoder (MC) and 3-frame autoencoder (DFD) perform worse than 3-frame autoencoder (no motion) according to the ROC curves as shown in Fig. 4.2. Reflecting on classical detectors SDIa and SDIp, they perform better when they are provided with motion information presented as motion compensated (MC) frames they use. Therefore it is a natural thought for exploiting motion information, or precisely motion vectors, can improve detection performance for as well DNN detectors. Our failure on processing motion vectors in a good sense may reveal the improper opera-

tion we conduct during the procedure. DNN detectors are different from SDI detectors, which suggests they might need to be treated in a different way. But precisely how to use motion vectors properly for DNN detectors is remained as a research area for following researchers.

In Chapter 6, we retrain RAFT for making more accurate the motion vectors on blotches and hence improve blotches removal performance of ML3Dex. It does work in our dataset as shown in Fig. 6.4 and Fig. 6.5 where each shows the improvement of optical flows (motion vectors) and of blotches restoration. But it does not necessarily work well in real degraded frames, in some cases it works like in Fig. 6.8, in others it does not like in Fig. 6.7. We think it is because some real degraded blotches are not like the ones we create, our blotches are of single color, single intensity and have tendency to be a long patch going southeast, the real degraded blotches in Fig. 6.8 are rather similar to the blotches we create hence can trigger retrained RAFT to make a modification. But the blotches in Fig. 6.7 are too different to be recognized by retrained RAFT so it treats them the same way before retraining, which still has flaws. To make retrained RAFT more robust to real blotches, one can strive to model the real blotches more properly to help RAFT recognize those real blotches better. This is an interesting avenue for future work.

---

## References

---

- [1] Alp, B., Haavisto, P., Jarske, T., Oistamo, K., and Neuvo, Y. A. (1990). Median-based algorithms for image sequence processing. In *Visual Communications and Image Processing'90: Fifth in a Series*, volume 1360, pages 122–134. International Society for Optics and Photonics.
- [2] Arce, G. R. (1991). Multistage order statistic filters for image sequence processing. *IEEE Transactions on Signal Processing*, 39(5):1146–1163.
- [3] Arce, G. R. and Malaret, E. (1989). Motion-preserving ranked-order filters for image sequence processing. In *IEEE International Symposium on Circuits and Systems,,* pages 983–986. IEEE.
- [4] Aydin, Y. and Dizdaroglu, B. (2020). Blotch detection in archive films based on visual saliency map. *Complexity*, 2020.
- [5] Biemond, J., Looijenga, L., Boekee, D., and Plomp, R. (1987). A pel-recursive wiener-based displacement estimation algorithm. *Signal Processing*, 13(4):399–412.
- [6] Boyce, J. M. (1992). Noise reduction of image sequences using adaptive motion compensated frame averaging. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, volume 3, pages 461–464. IEEE Computer Society.
- [7] Ghanbari, M. (1990). The cross-search algorithm for motion estimation (image coding). *IEEE Transactions on Communications*, 38(7):950–953.
- [8] Gullu, M. K., Urhan, O., and Erturk, S. (2008). Blotch detection and removal for archive film restoration. *AEU-International Journal of Electronics and Communications*, 62(7):534–543.
- [9] Huang, T. S. and Tsai, R. (1981). Image sequence analysis: Motion estimation. In *Image sequence analysis*, pages 1–18. Springer.
- [10] Kokaram, A. C. (1998). Motion picture restoration. Springer.

- [11] Kokaram, A. C., Morris, R. D., Fitzgerald, W. J., and Rayner, P. J. (1995). Detection of missing data in image sequences. *IEEE Transactions on Image Processing*, 4(11):1496–1508.
- [12] Martinez, D. M. (1987). Model-based motion estimation and its application to restoration and interpolation of motion pictures. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE RESEARCH LAB OF ELECTRONICS.
- [13] Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., and Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*.
- [14] Nadenau, M. J. and Mitra, S. K. (1997). Blotch and scratch detection in image sequences based on rank ordered differences. In *Time-Varying Image Processing and Moving Object Recognition*, 4, pages 27–35. Elsevier.
- [15] Sizyakin, R., Voronin, V., Gapon, N., Pismenskova, M., and Nadykto, A. (2019). A blotch detection method for archive video restoration using a neural network. In *Eleventh International Conference on Machine Vision (ICMV 2018)*, volume 11041, pages 230–237. SPIE.
- [16] Teed, Z. and Deng, J. (2020). Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer.
- [17] Walker, D. and Rao, K. (1984). Improved pel-recursive motion compensation. *IEEE Transactions on communications*, 32(10):1128–1134.
- [18] Yous, H., Serir, A., and Yous, S. (2019). Cnn-based method for blotches and scratches detection in archived videos. *Journal of Visual Communication and Image Representation*, 59:486–500.