THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學

Department of Electronic and Information Engineering

Final Year Project Final Report

(2019/20)

Advanced Image Processing and Machine-learning Techniques for Film Restoration

Student Name: TANG Zhiheng
Student ID: 17083046d
Programme Code: 42470
Supervisor(s): Prof. LAM Kin-Man, Kenneth
Submission Date: December 2, 2020

List of Figures

2.1 Hidden Markov Model Example . . . 3
2.2 LeNet Architecture . . . 4
2.3 AlexNet Architecture . . . 4
2.4 SegNet Architecture . . . 6
2.5 FCN Architecture . . . 6

3.1 HMM observation sequence extraction . . . 9
3.2 Original of frame 952229 . . . 10
3.3 Raw detection of frame 952229 . . . 10
3.4 Detection of frame 952229 after connectivity elimination . . . 11
3.5 Motion detection of frame 952229 . . . 12
3.6 Detection of frame 952229 after motion elimination . . . 12
3.7 Frame 952229 before(Top) and after(Bottom) restoration . . . 13
3.8 A closer inspection of the restoration performance . . . 14
3.9 Original Image(Left) and Ground Truth(Right) . . . 16
3.10 SegNet Layers . . . 18
3.11 Original image(Left) and raw detection(Right) . . . 20
3.12 Detection result of the current frame(Left) and next frame(Right) . . . 21
3.13 Original image(Left), ground truth(Middle) and CNN detection(Right) . . . 21
3.14 Original image(Left), detection result(Middle) and restored image(Right) . . . 22
3.15 Frame 952229 after restoration based on inpainting . . . 23
3.16 Original of frame 952211 with artifacts marked . . . 24
3.17 Results after 1200, 2400, 3600, 4800, 6000, and 7200 iterations . . . 25
3.18 Results after 7200 iterations with artifacts marked . . . 25

4.1 HMM Restoration of frame 952229 . . . 27
4.2 HMM Restoration of frame 952222 . . . 28
4.3 HMM Restoration of frame 952400 . . . 29
4.4 HMM Restoration of frame 952238 . . . 30
4.5 HMM Restoration of frame 952239 . . . 31
4.6 CNN Restoration Example 1 . . . 32
4.7 CNN Restoration Example 2 . . . 32
4.8 CNN Restoration Example 3 . . . 32
4.9 CNN Restoration Example 4 . . . 33
4.10 CNN Restoration Example 5 . . . 33

4.11 CNN Restoration Example 6 33

List of Tables

3.1 HMM Dataset Division 8
3.2 HMM Characteristic 8
3.3 HMM Sequence Test 10
3.4 HMM Accuracy 14
3.5 CNN Dataset Division 17
3.6 Implementation Details 18
3.7 Segnet Encoder Structure 19
3.8 Segnet Decoder Structure 20
3.9 CNN Accuracy 22

Contents

1 Introduction **1**

2 Literature review **2**
2.1 Introduction . . . 2
2.2 Filter-based Artifact Detection . . . 3
2.3 Model-based Artifact Detection . . . 3
2.4 Convolutional Neural Network . . . 3
2.5 Semantic segmentation . . . 5

3 Proposed approach **7**
3.1 Artifacts Detection Based on Hidden Markov Model . . . 7
3.1.1 Introduction . . . 7
3.1.2 Training Set Preparation . . . 7
3.1.3 Hidden Markov Model Artifacts Detection . . . 8
3.1.4 Connectivity Evaluation . . . 11
3.1.5 Lucas-Kanade Motion Tracking . . . 11
3.1.6 Final Restoration . . . 13
3.1.7 Accuracy . . . 14
3.1.8 Limitations . . . 15
3.2 Artifacts detection based on Deep Convolutional Neural Network . . . 16
3.2.1 Ground Truth Dataset Extraction . . . 16
3.2.2 Initializer . . . 17
3.2.3 Loss Function and Median frequency balancing . . . 17
3.2.4 Optimizer . . . 18
3.2.5 Segnet . . . 18
3.2.6 SegNet Simplification . . . 19
3.2.7 Model Training . . . 20
3.2.8 CNN Artifacts Detection . . . 20
3.2.9 False Alarm Elimination . . . 21
3.2.10 Final Restoration . . . 22
3.2.11 Accuracy . . . 22
3.3 Restoration Based on Inpainting . . . 23
3.3.1 Inpainting . . . 23
3.4 Restoration Based on Deep Image Prior . . . 24
3.4.1 Denoising . . . 24

4 Restoration Result **26**
4.1 Restoration based on Hidden Markov Model 26
4.2 Restoration based on CNN Model 26

5 Conclusion **34**

Chapter 1

Introduction

Before the invention of digitizing technology, images and movies have long been stored in physical films. As a kind of physical copy, films inevitably suffers from physical or chemical degradation throughout time. Especially in early history, the film quality is not good that is more venerable to degradation. At that time, the people lacked knowledge and experience for proper film storage, which accelerated this process. Meanwhile, people can only play films with physical copies without digitizing technologies, and the number of blotches and scratches might increase in every play.

The degradation of the films significantly reduces the value of movies and images. It is imaginable that the blotches, flickers, and scratches on the frame negatively affect the viewing experience and conceal many valuable details. Realizing the problem's seriousness, professionals made several trials to restore the frames in chemical ways. However, physical copies are usually scarce, especially at a time when films are costly. Directly operating on films brings too much risk and uncertainty. In the late 20 century, digitizing technologies became mature and widespread. Many movies have been scanned and archived in digital form. Digital copies are invariant of the physical and chemical condition, and it allows unlimited copies. Meanwhile, the restoration can be done using algorithms in computers, which is much more convenient than chemical ways. It also has enormous potentials because the algorithm can be improved year by year.

The degradation may appear in white noise, dirt, sparkle, blotches, and scratches, appearing in random size and position. In general, we name those kinds of degradation as "artifacts." In the industry, the artifacts are usually removed and restored by humans, using photo editing software such as "PhotoShop" of Adobe Inc. This process takes a massive amount of time and human resources. If an automatic detection and restoration system is developed to replace human effort, it will have high-value and enormous potentials in the industry.

In Chapter 2, a brief literature review will be provided on both conventional and machine-learning-based film restoration techniques. Chapter 3 will introduce and explain the artifact detection and restoration approaches I have developed this year. In Chapter 4, the restoration performance are shown to demonstrate the universality of the approaches. Chapter 5 will provide a brief conclusion on achievements, challenges encountered, and possible future works.

Chapter 2

Literature review

2.1 Introduction

The most common film restoration follows the two-step detection-restoration structure. For detection, the traditional methods focus on detecting the ”abnormality” of pixel sequences, mostly observed from the spatial or temporal characteristic. The approaches used to detect those characteristics can be further classified into two categories, filter-based and model-based. Filter-based techniques aim at examining pixels' spatial-temporal value and mark degraded pixels according to intuitive decision rules. Model-based techniques regard pixels' spatial-temporal value as an observation sequence and evaluate it using a trained model. Instead of setting up rules manually, model-based techniques let the model learns the ”natural patterns” and evaluate how much the real sequences deviate from ”natural.”

In recent years, more works based on deep convolutional networks have been developed in this field. The artifacts detection in CNN is a kind of binary semantic segmentation. The artifacts are segmented as foreground, and the natural part is segmented as background. Researchers have developed several mature methods for blotches and scratch detection. However, CNN training requires high-quality datasets, or the model will learn the artifact pattern incorrectly. It also needs a dedicated dataset and model for each artifact since they do not share the same pattern. Like traditional methods, CNN detection also raises a lot of false alarms, but they can be eliminated by connectivity elimination.

In film restoration, the artifacts region based on the detection result will be removed and replaced by the original content. The restoration methods can be further classified into two branches. The first one is only to use the information from the current frame and replace the artifacts region with neighboring pixels. The other one is to extract the content from neighboring frames for replacement. Both ways have their strength and limitations.

2.2 Filter-based Artifact Detection

The earliest digitized film restoration for digitized film restoration can be traced back to BBC's prototype equipment [1] developed in 1985, which removed film dirt and spikes on frames. The detection procedure examined whether the temporal intensity differences of a pixel are above a certain threshold. After detection, all artifacts candidates are removed and replaced by the median of local neighborhoods. The proposed detection-removal structure is still the most commonly adopted structure nowadays, providing flexibility and possibilities of combining individual methods.

Later in 1992, Kokaram and Rayner [2] proposed a new method called SDIp, which improved Storey's algorithm [1] by using motion-compensated pixel values to reduce moving interference. It also adds a rule that forward and backward luminance differences must have the same sign, or the pixel has a high probability of being on edge.

2.3 Model-based Artifact Detection

In 1989, Rabiner [3] performed a comprehensive analysis of HMM. It evaluated HMM models as "very rich in mathematical structure." It can also form the theoretical basis for applications in a wide range. However, Rabiner [3] also pointed out that one limitation of HMM is assuming the consecutive observations to be independent while ignoring any correlations. Also, in HMM, the probability of a state at time t only depends on t-1. In the real world, the dependency relationship may extend to two or more states. Despite the above limitation, HMM works well most of the time.

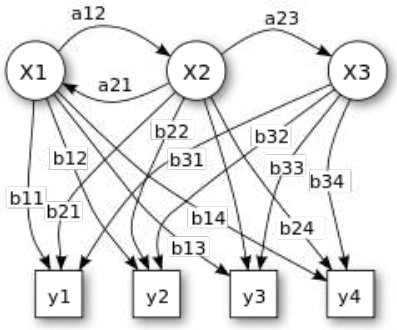

Figure 2.1: Hidden Markov Model Example

2.4 Convolutional Neural Network

One of the early pioneering work of Convolutional Neural Network is the LeNet model designed by LeCun, Bottou, Bengio, *et al.* [4] in 1994. Its target job is to recognition of handwritten zip

code digits. Several improvements have been made in the coming few years. This paper proposed a model structure consisting of three kinds of layers: convolution, pooling, non-linearity, which became the fundamentals of later works. It proves that CNN can extract spatial features through convolution, and it can be trained using a backpropagation network.

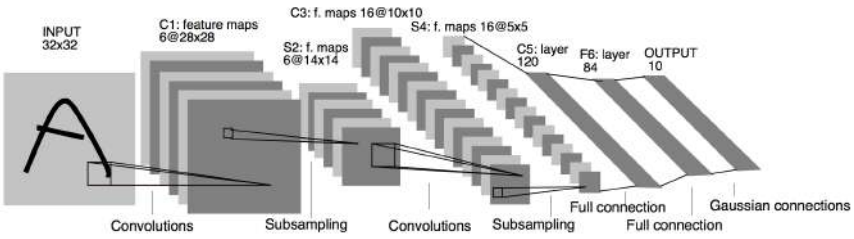

Figure 2.2: LeNet Architecture

Later in 2012, Krizhevsky, Sutskever, and Hinton [5] proposed a new network for image classification with a deeper and broader architecture than LeNet. Similar works are the VGG network [6] that further increase the number of layers to achieve better classification result.

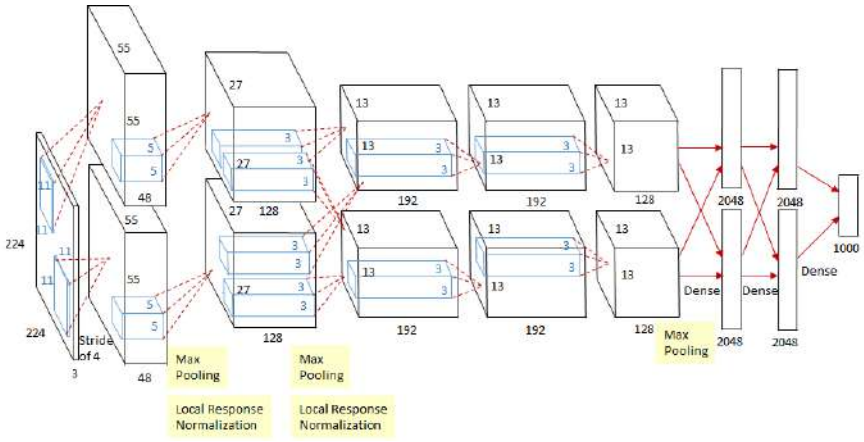

Figure 2.3: AlexNet Architecture

2.5 Semantic segmentation

Semantic segmentation, also called image segmentation, aims at clustering parts of an image together that belong to the same object class. The typical models in this field are SegNet [7] and FCN [8]. The architectures are shown below:

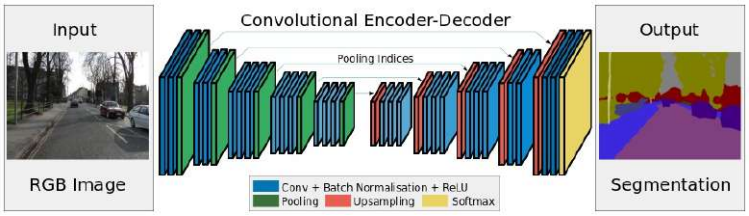

Figure 2.4: SegNet Architecture

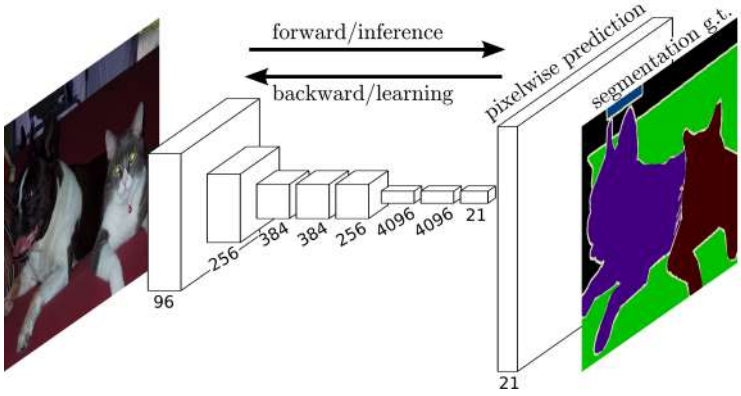

Figure 2.5: FCN Architecture

The semantic segmentation model usually has an encoder-decoder structure. As explained in the paper [7], Encoder-Decoder pairs designed for ”creating feature maps for classifications of different resolutions.” This operation prepares fundamentals for pixel-wise evaluation in the next stage and determines how the image is segmented.

Chapter 3

Proposed approach

3.1 Artifacts Detection Based on Hidden Markov Model

3.1.1 Introduction

Traditional artifact removal algorithms focus on detecting abnormal features. On contrast, HMM model attempts to evaluate how ”normal” a sequence of pixels is. To learn natural patterns, the HMM model is trained by pixel sequences from artifacts-free movie frames. Any pixel that deviates from the natural sequence can be regarded as artifacts candidates. This approach is in effect a form of novelty detection.

It is noticed that most of the defect detection approaches face the dilemma of how to improve the correct detection coverage while reducing false alarms. Like other model-based approaches, HMM provides good coverage on artifacts, but many false alarms occur as well. According to the experiment, many false alarms appear at the region of motion and varying light intensity. According to Wang and Mirmehdi [9], false alarm can be effectively eliminated by evaluating the spatial-temporal differences between frames and performing motion tracking. In this approach, I proposed a 3-step artifact detection system consisting of (1) HMM detection (2) Connectivity evaluation, and (3) Lucas-Kanade motion tracking.

3.1.2 Training Set Preparation

The training set selected is a frame sequence of Mei Ah Entertainment Group Co. Ltd. archived in the Hong Kong Polytechnique University Database. The data set consists of two subsets, the original data set and the restored data set. The frames in the two data sets have one to one correspondence, and all artifacts in the restored data set are detected and restored by professional groups of the company.

Table 3.1: HMM Dataset Division

Dataset	Frames	Size
Total	Frame 952211 to Frame 955119	2909
Training Set	Restored 955112 to Restored 955124	13
Testing Set	Original 952218 to Original 952247	30

To be representative, all training frames and testing frames selected are in the middle of a scene with moderate motion.To train the HMM model for natural sequence, 13 consecutive restored frames are used for the training set. One hundred frames other than the training frames are selected to evaluate the detection accuracy.

3.1.3 Hidden Markov Model Artifacts Detection

An HMM model is determined by its numbers of hidden states, observation states, and the length of the observation sequence. As suggested by Wang and Mirmehdi [9], the number of hidden states is set 5, since it can be interpreted as state transition between ”Background”, ”Intermediate”, ”Foreground”, ”Intermediate”, ”Background”. It is intuitively reasonable, and the performance is good in the experiment. Meanwhile, the number of observation states is set 7, and the length of the observation sequence is 13 to trade-off between representation and computational burden.

Table 3.2: HMM Characteristic

Parameter	Value
Hidden States	5
Observation States	7
Sequence Length	13

Before training, all frames are converted into gray-scale since only the pixel intensity value is considered. An observation sequence is formed by extracting the value of the same pixel location of the 13 consecutive frames. In this experiment, 10000 locations are selected randomly, which generates 10000 natural observation sequences for training, as illustrated by Figure 3.1. The observation sequence is then resized to 13 by 10000, and further scaled from (0,255) to (0,6) according to observation states. After that, the observation sequences are used to train the HMM model.

Figure 3.1: HMM observation sequence extraction

When the training is finished, the HMM model should be able to evaluate how natural a pixel sequence is, with arbitrary length and value. Denote $O_x^t(k)$ a pixel sequence extracted at pixel location x, centering at frame t. That is, $O_x^t(k),\ (1 \leq k \leq 13)$ represents the quantized pixel values at location x of frame $t-7$ to $t+7$. Then, the sequence elements are removed one at each time to create 13 sequences of length 12, denoted by $O_x^t(h), h \neq k$. For each 13 sequences, a likelihood is returned by the model $V_x^t(h) = O(h)_x^t, h \neq k, h \neq k$, representing how normal the sequence is if the element h is removed. To eliminate the effect of scene variation level, u_x^t is computed as the likelihood of the central pixel by averaging $V_x^t(h)$.

$$V_x^t(h) = P\{O(h)_x^t,\ h \neq k\} \tag{3.1}$$

$$u_x^t = \frac{V_x^t(c)}{\sum_{h=1}^{K} V_x^t(h)},\ K = 13,\ c = 7 \tag{3.2}$$

A detection test on sample sequences is performed to examine the characteristic of the HMM model, as shown in Table 3.3. The trained HMM model successfully returns a high value given a spike at the center of the sequence. Even to a gradual spike, the HMM model still return a value larger than 0.2. It can be observed that the HMM model ignores negative spike, non-central spike, and any artifact of these two nature.

Table 3.3: HMM Sequence Test

$O_x^t(k)$	Comment	u_x^t
2 2 2 2 2 2 2 2 2 2 2 2 2	Uniform Sequence	0.0552277076960331
2 2 2 2 2 2 3 2 2 2 2 2 2	Spike = 1	0.17362258367339967
2 2 2 2 2 2 4 2 2 2 2 2 2	Spike = 2	0.2652547195410352
2 2 2 2 2 2 5 2 2 2 2 2 2	Spike = 3	0.5361801561693467
3 3 3 3 3 3 0 3 3 3 3 3 3	Negative Spike	0.004094397575555158
4 4 4 4 4 5 6 5 4 4 4 4 4	Gradual Spike	0.28692186697829125
2 2 2 2 2 2 2 4 2 2 2 2 2	Non-central spike	0.045620909937678485

Figure 3.2: Original of frame 952229

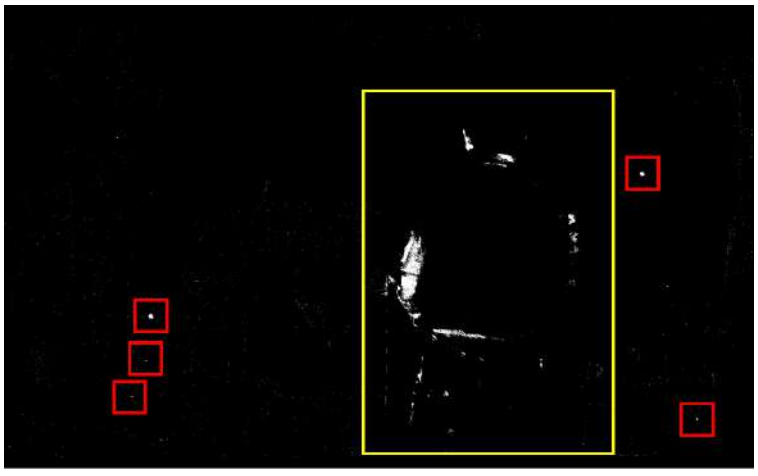

Figure 3.3: Raw detection of frame 952229

Based on the experiment, a threshold of 0.2 is suitable for covering all the artifacts while not introducing many false alarms. Frame 952229 is used as a sample frame, with both static region and motion region. In this frame, the man in the blue shirt is walking toward the inside, while all other objects remain static. The primary artifacts are small white spikes, occurring in random size and location. It can be observed from the detection result that the HMM model well covers most of the spikes, but raises a large numer of false alarms at the moving edge.

3.1.4 Connectivity Evaluation

Connectivity evaluation is performed to eliminate the false alarm. In this approach, an assumption is made that artifacts will not appear at the same location on neighboring frames. Artifacts candidates that have strong spatial-temporal connectivity is regarded as false alarm. According to Morris [10], the false alarm can be effectively identified by performing Gibbs sampling with annealing, which determines the contribution to each candidate pixel's connectivity. Due to this method's computation burden, a simplified method is adopted, that is remove the artifact candidates if there are other candidates among the eight neighboring pixels on the next frame.

Figure 3.4: Detection of frame 952229 after connectivity elimination

3.1.5 Lucas-Kanade Motion Tracking

Due to HMM detection's sensitiveness towards motion, many false alarms occur on the moving edge. As suggested by Wang and Mirmehdi [9], Lucas-Kanade motion tracker is applied to remove the false alarm at the moving region. In the experiment, artifact candidates that shows large motion intensity is removed.

Figure 3.5: Motion detection of frame 952229

Lucas-Kanade motion tracker returns more than ten times higher value to motion pixels than to static pixels in the experiment. Most of the motion pixels are covered when setting the motion threshold value as 10 while not introducing any false alarm. In frame 952229, the detection of the walking man is clean and successful. There is no static region detected falsely. Based on the motion detection, any artifact candidates within this region are removed. After motion elimination, the false alarm raised by the moving part has been eliminated. The detection result now only contains real artifacts. It is good enough and it can be used for restoration.

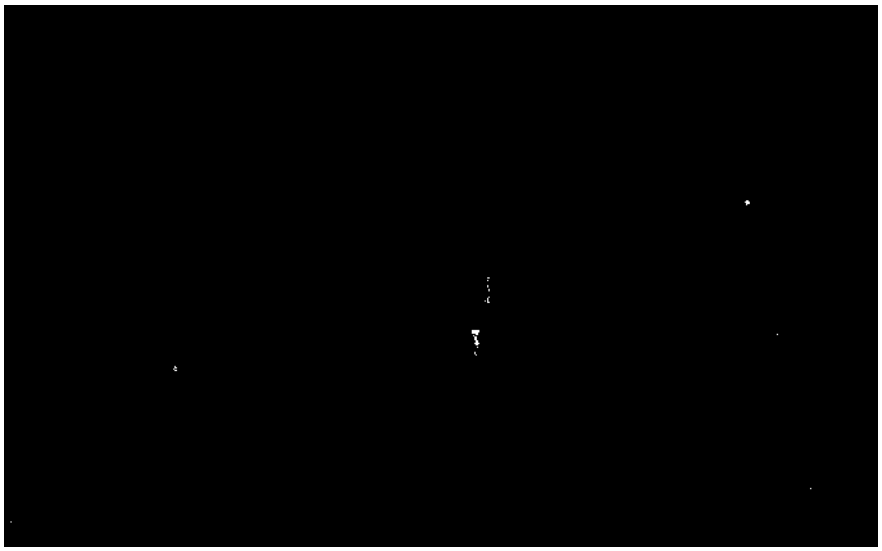

Figure 3.6: Detection of frame 952229 after motion elimination

3.1.6 Final Restoration

After detection and false alarm elimination, the defect map is used to restore the original image. The sample frame below has three noticeable artifacts, which are marked by red circles. The results are shown below.

Figure 3.7: Frame 952229 before(Top) and after(Bottom) restoration

Figure 3.8: A closer inspection of the restoration performance

3.1.7 Accuracy

To evaluate the HMM detection performance, the accuracy and false alarm number is calculated based on the testing set. In the experiment, nearly half of the artifacts can be covered by the detection result. Actually, the target dataset does not label the artifacts but only provides the original and restored images. I extract the ground truth using my own program and inevitably raise a lot of false alarms. The actual accuracy should be higher than below, and the restoration results prove that it well cover all the artifacts. Also, it can be observed that after false alarm elimination, there are still many remains. It is due to the characteristic of HMM. However, the false alarm region is usually less than 5 pixels, which does not affect inpainting results for low-resolution movies.

Table 3.4: HMM Accuracy

Name	Comment	Value
Number of Frames	—	30
Frame Size	—	1300*2100
Total Artifacts	Total artifacts pixels in the ground truth	14640
Total Detects	Total pixels HMM detect as artifacts	61772
Total Accurate Detect	Real artifacts successfully detected by HMM	7023
Total Loss	Real artifacts not detected by HMM	7623
Total False Alarm	Normal pixels incorrectly detected by HMM	47136
False Alarm Percentage	Total False Alarm/Total Detects	76.2992%
Accuracy	Total Accurate Detect/Total Artifacts	47.9508%

3.1.8 Limitations

In this approach, each pixel is examined individually, where the computational intensity is O(n) given n pixels in an image. In my implementation, the HMM forward and backward procedure is accelerated using GPU, written in PyTorch.

In the motion elimination step, the artifacts detected in the moving region are abandoned. This approach is unable to separate artifacts from the moving region, which inevitably reduces detection rate.

Since most of the artifacts in this dataset have a small region, appearing in small white dots, some false alarms are similar to real artifacts that we cannot identify. The problem caused by false alarms in the restoration process could be severe or slight based on the method used

A suitable threshold in raw detection is vital so that all artifacts are covered while not introducing many false alarms. However, the determination of the threshold remains unclear. The suitable threshold has a weak relationship with the mean luminance value and standard deviation of the image, but in the experiment, a found suitable threshold in one frame is usually suitable to other frames in the same scene.

3.2 Artifacts detection based on Deep Convolutional Neural Network

3.2.1 Ground Truth Dataset Extraction

Due to the lack of good datasets in this field, I decided to use the same dataset used by the HMM procedure and extract the ground truth myself. As mentioned above, the dataset provides original frames and artifacts-free frames, where the restored region can be regarded as ground truth. In this approach, any pixel with intensity differences between the original frame and restored frame over 10 is extracted as the ground truth.

The reason a threshold is used but not including all modified pixels is that not only the artifact pixels are modified. In the restored frame, it is common that the whole region around the artifacts has been brightened, darkened, or blurred. The modified region also has irregular shapes, that simple adjustment in luminance and contrast cannot help. In a few cases, the frames show misalignment after restoration, which raises a huge amount of false alarms.

The original image size is 1300*2100. However, the artifacts are usually very small which does not necessarily requires large image size. There are also many regions with no artifacts, which will dilute the training. Therefore, the image size is decided as 256*256.

Figure 3.9: Original Image(Left) and Ground Truth(Right)

24 images are cropped from a single frame at the fixed positions. In order to increase the qualities of the dataset, the artifacts pixel number of each cropped image will be counted. Images with artifacts pixels less than 10 is regarded as useless data, which will dilute the training. Images with artifacts pixels more than 100 usually occur from misalignment, which will mislead the model. Only images with artifacts pixels number between 10 and 100 are considered qualified and saved to the dataset source folder. At the same time, the corresponding artifact ground truth mask is saved to the label folder. In total, 7918 images is used to train the network.

Table 3.5: CNN Dataset Division

Dataset	Frames	Size
Training Set	Images cropped from Frame 955112 to 955124	7918
Testing Set	Images cropped from Frame 952218 to 952317	100

3.2.2 Initializer

During the model training process, an initializer is essential to the convergence time and model quality. With the help of an initializer, we can intialize the weights of the network so that the neuron activation functions will not start from saturated regions. In this approach, Xavier initialization is adopted as the weight initializer in Neural Networks.

In Xavier Initialization, the activation function's biases are initialized be 0, and the weights W_{ij} at each layer are initialized as:

$$W_{ij} \sim U\left[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\right]$$

U: Uniform distribution n: size of the previous layer (number of columns in W)

3.2.3 Loss Function and Median frequency balancing

Artifacts detection in CNN is a binary semantic segmentation task. In this approach, the binary cross-entropy loss is suitable for the binary. classification. The artifacts are labeled as class 1, and the normal pixels are labeled as class 0. The following is the binary cross-entropy loss function.

$$L(\theta) = -\sum_{i \in Y_1} \cdot \log P(y_i = 1|X;\theta) - \sum_{i \in Y_0} \cdot \log P(y_i = 0|X;\theta)$$

However, in our dataset, normal pixels are much more than degraded pixels. The image size is 256*256, which is 65536 pixels in total. However, most of the images have only 10-100 degraded pixels. The unbalanced class samples will lead to bad training results. According to the paper [11], we can apply median frequency balancing to increase the model's sensitiveness to artifacts. The binary cross-entropy loss is weighted as below:

$$L(\theta) = -\alpha \sum_{i \in Y_1} \cdot \log P(y_i = 1|X;\theta) - \beta \sum_{i \in Y_0} \cdot (y_i = 0|X;\theta)$$

Based on the experiment, the training result is the best when assigning weight 0.985 to artifacts and 0.015 to normal pixels. Without the median frequency balancing, the model is unable to correctly detect the artifacts.

$$L(\theta) = -0.985 \sum_{i \in Y_1} \cdot \log P(y_i = 1|X;\theta) - 0.015 \sum_{i \in Y_0} \cdot (y_i = 0|X;\theta)$$

3.2.4 Optimizer

For the optimizer, the stochastic gradient descent algorithm is chosen since it implements momentum to reduce the possibilities of falling into local minimum. SGD has two extentions - Root Mean Square Propagation (RMSProp) and Adaptive Gradient Algorithm (AdaGrad) that secure a good convergence in training. In the experiment, the performance is the best with a learning rate 0.1 and momentum 0.9.

In recent years, more and more papers are using Adam optimizer, which combines the advantages of SGD and adaptive learning rates for different parameters. However, SGD with momentum can converge better with longer training time and it shows better robustness than Adam.

Table 3.6: Implementation Details

Component	Detail
Loss function	Weighted Binary Cross-Entropy Loss
Optimizer	SGD
Learning Rate	0.1
Momentum	0.9
Input Size	$256 \times 256 \times 3$
Package	Tensorflow

3.2.5 Segnet

SegNet [7] is the most popular model in semantic segmentation. It can segment the image into several regions of different kinds of objects. For example, SegNet can segment a street photo into the sky, trees, cars, and roads, as illustrated below. SegNet consists of 13 convolution layers in both of its Encoder and Decoder Network. Each of the convolutional layer is followed by a batch normalization layer and ReLU. Also, There are 5 max-pooling layers in the encoder and 5 max-unpooling layers in the decoder.

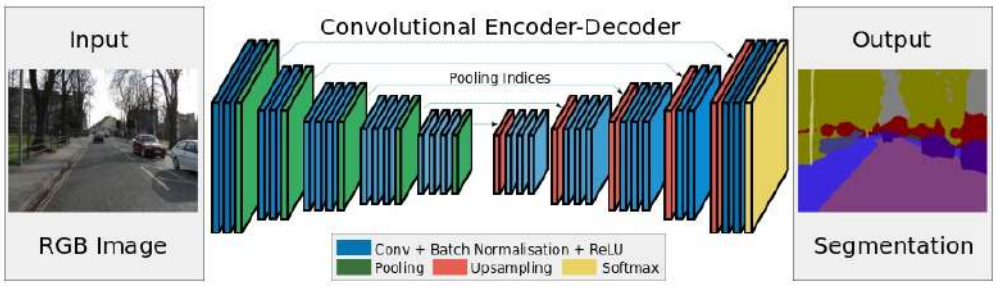

Figure 3.10: SegNet Layers

3.2.6 SegNet Simplification

As suggested by Yous, Serir, and Yous [11], a similar performance on artifacts detection can be achieved using a simplified SegNet. The model comprises only 3 layers of convolution, each of which is followed by the ReLU layer. All the batch normalization layers are removed. The number of max-pooling and max-unpooling is reduced to one. However, there is one special layer in the encoder that concatenates the three convolution layers' output.

The original SegNet is designed for segmentation that may include complex features such as car, tree, road, and humans, so a large number of convolution layers are needed to handle the high-level features. However, the artifacts are usually in the form of simple shapes, such as white blotches and scratches. Too many layers cannot improve the performance but leads a to heavy computation burden. Whether to adopt batch-normalization layers is optional. Considering the network is not complex and there is already a Xavier Initializer, the batch-normalization layers are ignored in this approach. The reduction of pooling layers is reasonable because the artifacts have small sizes and simple patterns, and the blurring effect of pooling may completely conceal them. Therefore, only 1 max-pooling and max-unpooling layer remain.

It is also noticed that a small kernel size is preferred. According to Yous, Serir, and Yous [11], small filters can leverage blotches and scratches, which can be characterized by local variations. The efficiency also increase by replacing large filters with a cascade of small filters.

In this approach, the simplified SegNet [11] is used for artifact detection. The kernel size of 3*3 is selected as suggested. The Layers of the model are shown below.

Table 3.7: Segnet Encoder Structure

Layer	Name	Input size	Kernal	Feature map
Conv	conv1	$256\times256\times3$	$3\times3\times3$	32
ReLU	relu	$256\times256\times32$	—	32
Conv	conv2	$256\times256\times32$	$3\times3\times32$	64
ReLU	relu	$256\times256\times64$	—	64
Conv	conv3	$256\times256\times64$	$3\times3\times64$	128
ReLU	relu	$256\times256\times128$	—	128
Concat	conc	—	—	224
Conv	conv11	$256\times256\times224$	$1\times1\times224$	64
MaxPool	maxpool	$256\times256\times64$	$2\times2(stride=2)$	64

Table 3.8: Segnet Decoder Structure

Layer	Name	Input size	Kernal	Feature map
MaxUnpool	maxunpool	$128 \times 128 \times 64$	—	64
Deconv	deconv2	$256 \times 256 \times 64$	$3 \times 3 \times 64$	32
ReLU	drelu	$256 \times 256 \times 32$	—	32
Deconv	deconv1	$256 \times 256 \times 32$	$3 \times 3 \times 32$	32
ReLU	derelu	$256 \times 256 \times 32$	—	32
Conv	conv	$256 \times 256 \times 32$	$3 \times 3 \times 32$	2
Sigmoid	sigmoid	$256 \times 256 \times 2$	—	—

3.2.7 Model Training

The model is trained using the dataset mentioned at the beginning of the section with 10 epochs. The dataset is shuffled at the beginning of each epoch to increase the randomness. The loss dramatically decrease at first and then gradually goes down at a relatively slow rate. In training, the model shows convergence in the first epoch, with little change in the loss. However, it needs a longer time for SGD to achieve the real global minimum.

3.2.8 CNN Artifacts Detection

Given an image in the testing set, the model returns a 2d array indicating each pixel's degraded probabilities. Usually, the value difference between artifact candidates and normal pixels reaches 100 times larger, which allows easy separation. In this procedure, I selected the pixels with top 1% largest probabilities as the artifacts candidates. By applying this rule, no matter more or fewer artifacts in the frame, they will be well detected with very few false alarms, which will be explained in false alarm elimination.

Figure 3.11: Original image(Left) and raw detection(Right)

3.2.9 False Alarm Elimination

The false alarm can be eliminated by evaluating the spatial-temporal connectivity between neighboring frames. If a pixel location is classified as degraded pixels in both frames, the pixel will be regarded as false alarms.

As mentioned above, the pixels with the top 1% most enormous degraded probabilities are marked in the defect map. The high-value region on neighboring frames is always very similar in the experiment, which means no matter the false alarm regions are large or small, they can always be canceled through comparing. The detection results also prove that point.

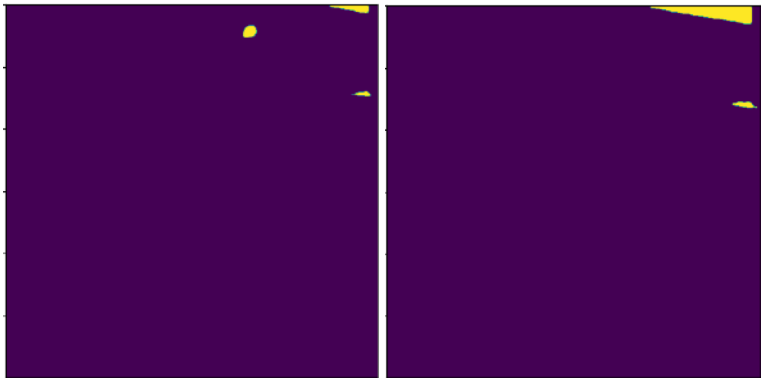

Figure 3.12: Detection result of the current frame(Left) and next frame(Right)

After false alarm elimination, the accuracy improves a bit. As shown below, the detection result is clean and precise.

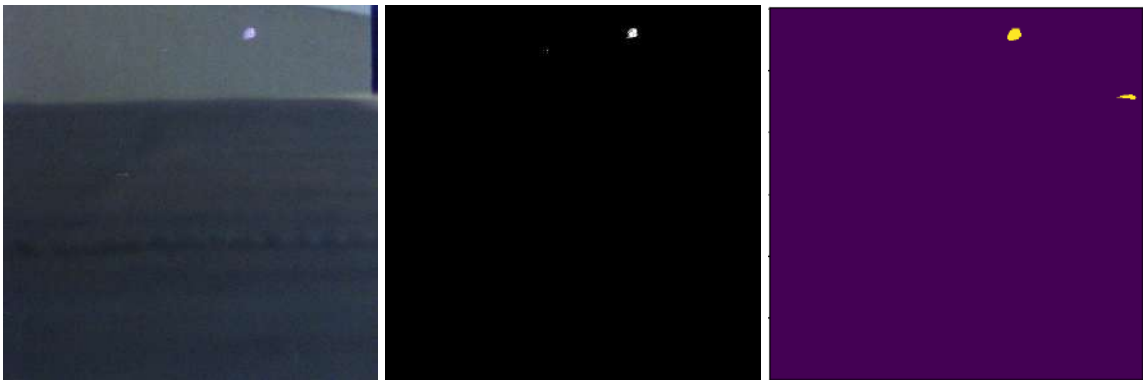

Figure 3.13: Original image(Left), ground truth(Middle) and CNN detection(Right)

3.2.10 Final Restoration

After detection and false alarm elimination, the defect map is used to restore the original image. After restoration, the artifacts disappear as expected.

Figure 3.14: Original image(Left), detection result(Middle) and restored image(Right)

3.2.11 Accuracy

To evaluate the performance of CNN detection, the accuracy and false alarm number is calculated based on the testing set. We can observed that the accuracy is higher than HMM, achieving 78.1102%. Also, there are still a lot of false alarms of CNN detection after false alarm elimination. However from the restoration result, the CNN detection covers all the artifacts and perform well in restoration.

Table 3.9: CNN Accuracy

Name	Comment	Value
Number of Frames	—	100
Frame Size	—	256*256
Total Artifacts	Total artifacts pixels in ground truth	15877
Total Detects	Total pixels CNN detect as artifacts	38654
Total Accurate Detect	Real artifacts successfully detected by CNN	12403
Total Loss	Real artifacts not detected by CNN	3474
Total False Alarm	Normal pixels incorrectly detected by CNN	26258
False Alarm Percentage	Total False Alarm/Total Detects	67.9172%
Accuracy	Total Accurate Detect/Total Artifacts	78.1102%

3.3 Restoration Based on Inpainting

3.3.1 Inpainting

Based on the detection result, restoration based on inpainting is performed to replace artifacts with neighboring pixels. To replace the artifacts region, the detection region should be well covered and a bit larger than the degraded region. However, some of the detection regions do not cover every degraded pixel, impacting the restoration performance. The defect map is first dilated and corrupted with a 5*5 kernel to fill the hollow within detection regions in this approach. The defect map is then dilated again using 5*5 kernel to expand the regions to make sure they have a larger size than artifacts.

Figure 3.15: Frame 952229 after restoration based on inpainting

After restoration by the inpainting method, the white spots and other artifacts have been successfully removed. However, due to false alarm, a few ungraded regions has been removed and restored by neighboring pixels as well, where the content is blurred slightly. For low-resolution films, the blurring effect is not identifiable for human eyes. For high-resolution films, the blurring may be apparent.

3.4 Restoration Based on Deep Image Prior

3.4.1 Denoising

Restoration based on deep image prior denoising feature is a new method for film restoration. This method does not require any defect map. The idea is to utilize the feature of the neural network that learns natural features prior to unnatural features. According to Ulyanov, Vedaldi, and Lempitsky [12], the degraded image can be restored by being fitted in the proposed network and interrupted after a certain number of iterations.

First, the initial parameter of the deep neural network f is set randomly. The overall MSE is used as the loss function. The training set consists of one single degraded image z. By iteratively training f with z, f will gradually learn the features of z. In this process, f will first learn the natural and undegraded features and then learn the noisy features of z. That is because of the invariance characteristic and layer-wise structure of the convolution. If the training process is sufficiently long, the output of f will be the same as z, since all the features are learned. However, if we interrupt it halfway, we can obtain a "restored version" of z.

Figure 3.16: Original of frame 952211 with artifacts marked

Then, the original image is fitted into the model and start training. Frame 952211 is used as an example, and all artifacts are marked. A result is extracted every 1200 iterations. From the image sequences, we can observe the feature learning order of a deep convolution network. At around 1200 iterations (top left), all the white spots and purple regions are not learned. The resolution is also low because of limited features. As the iteration number increases, the resolution becomes better, but artifacts begin to appear. At 2400 iterations (top right), the white spot on the customer's back is visible. At 3600 iterations (middle left), the white spot on the shopkeeper's neck appears. The more iterations are completed, the more artifacts are exposed.

Figure 3.17: Results after 1200, 2400, 3600, 4800, 6000, and 7200 iterations

The result after 7200 iterations is given below with artifacts marked. The purple artifacts regions are not so sharp and obvious after denoising. However, the removal of white spots performs poorly. The white spots are learned in early iterations, indicating that this kind of artifacts is not a ”noisy feature.” It is better to remove them by other methods. Also, the resolution of the result is still lower than the original image due to insufficient iterations. This method should only work well in removing artifacts with noisy features so that they are learned after reaching a satisfactory resolution.

Figure 3.18: Results after 7200 iterations with artifacts marked

Chapter 4

Restoration Result

In this chapter, a number of restoration results of methods implemented above are shown to prove their universality.

4.1 Restoration based on Hidden Markov Model

In the following 5 pages, 5 frames are selected for HMM restoration demonstration. For each frame three images are shown, which are original image, detection result, and restored image. The artifacts on the original image is circled in red.

4.2 Restoration based on CNN Model

In the 6th and 7th page, 6 images are selected for CNN restoration demonstration. For each frame three images are shown, which are original image, detection result, and restored image.

Figure 4.1: HMM Restoration of frame 952229

Figure 4.2: HMM Restoration of frame 952222

Figure 4.3: HMM Restoration of frame 952400

Figure 4.4: HMM Restoration of frame 952238

Figure 4.5: HMM Restoration of frame 952239

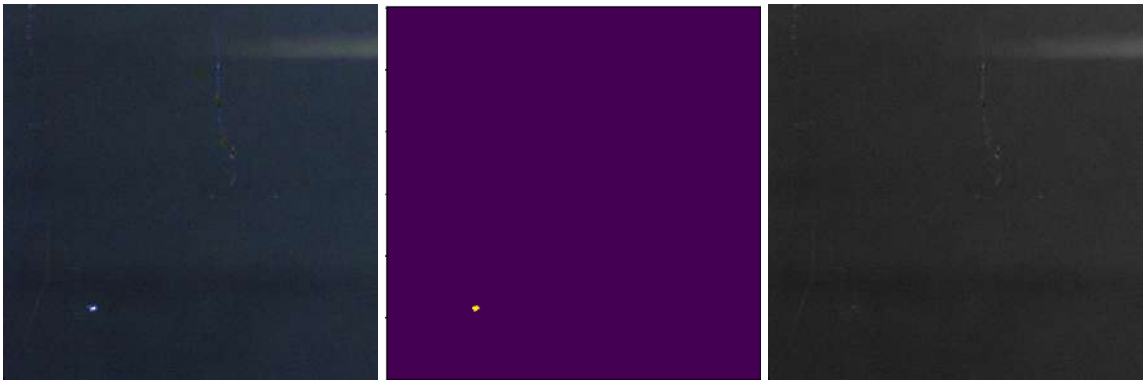

Figure 4.6: CNN Restoration Example 1

Figure 4.7: CNN Restoration Example 2

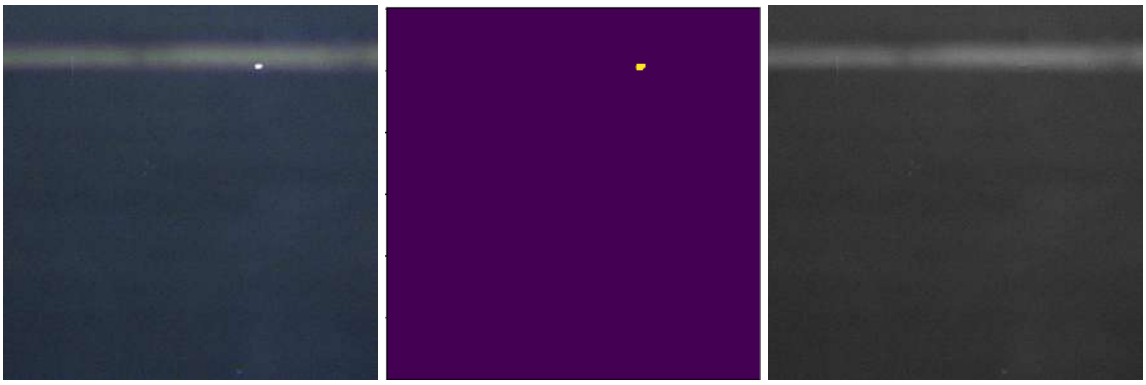

Figure 4.8: CNN Restoration Example 3

Figure 4.9: CNN Restoration Example 4

Figure 4.10: CNN Restoration Example 5

Figure 4.11: CNN Restoration Example 6

Chapter 5

Conclusion

In this final year project, I have implemented two artifacts detection systems. The first one is based on the Hidden Markov Model, followed by a two-step false alarm elimination. The work includes observation sequence extraction, model training, model examination, raw detection, connectivity evaluation, motion tracking, and accuracy test. The second one is based on convolutional neural network, followed by a one-step false alarm elimination. The work include, ground truth extraction, dataset preparation, loss function modification, adjusting learning parameters, connectivity evaluation, and accuracy test. Both methods detect the artifacts with very good performance, although the false alarm cannot be fully eliminated.

After the detection, I restore the degraded frames using inpainting method based on both HMM and CNN detection results. All the target artifacts are successfully removed and replaced by natural pixels.

Apart from the independent detection or restoration techniques. I implement the denoising feature of Deep Image Prior. This approach directly restore the images by learning natural patterns prior to unnatural patterns and being interrupted after a proper number of iterations.

Bibliography

[1] R. Storey, "Electronic detection and concealment of film dirt," *J. Soc. Motion Picture Telev. Eng.*, vol. 94, no. 6, pp. 642–647, Jun. 1985.

[2] A. Kokaram and P. Rayner, "System for the removal of impulsive noise in image sequences," *Proc. SPIE*, vol. 1818, no. 1, pp. 322–331, 1992.

[3] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

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

[7] v. badrinarayanan vijay, a. kendall alex, and r. cipolla roberto, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 2481–2495, 2017.

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[9] X. Wang and M. Mirmehdi, "Archive film defect detection and removal: An automatic restoration framework," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3757–3769, Aug 2012.

[10] R. Morris, "Image sequence restoration using gibbs distributions," *Ph.D. dissertation, Dept. Eng.*, 1995.

[11] H. Yous, A. Serir, and S. Yous, "Cnn-based method for blotches and scratches detection in archived videos," *Journal of Visual Communication and Image Representation*, vol. 59, pp. 486–500, 2019, ISSN: 1047-3203. DOI: https://doi.org/10.1016/j.jvcir.2019.02.005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1047320319300562.

[12] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition 2018*, pp. 9446–9454, 2018.