J. Vis. Commun. Image R. 59 (2019) 486–500

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

CNN-based method for blotches and scratches detection in archived videos ☆

Hamza Yous [a,*], Amina Serir [a], Sofiane Yous [b]

[a] USTHB, Electronic and Computer Science Faculty, U.S.T.H.B, B.P. 32 El Alia Bab Ezzouar, Algiers 16111, Algeria
[b] Intel Ireland, New Technology Group, Collinstown Industrial Park, W23 CX38 Leixlip, Kildare, Ireland

ARTICLE INFO

Article history:
Received 13 February 2018
Revised 23 January 2019
Accepted 3 February 2019
Available online 5 February 2019

Keywords:
Digital archived video restoration
Defects detection
Convolutional neural network
Deep learning

ABSTRACT

In this work, we present a fully connected convolutional encoder-decoder for defects detection in archived video. The proposed method handles the detection of two of the most common archived video-related defects, namely blotches and scratches. It consists of two stages: (1) pixel-level classification and description of each video frame into defects pixels or not, by means of a novel CNN-based encoder-decoder architecture, and (2) spatio-temporal analysis to group and fine-tune the detections. For blotch detection, the learned features, extracted from an intermediate stage of the network, are used to evaluate the dissimilarity between the pre-selected regions in consecutive frames. For scratches detection, the morphology of scratches is used to eliminate false alarms. The experiments are performed on various video sequences suffering synthetic and real scratches and blotches. The results demonstrate the effectiveness of our approach and significant improvement against the most recent detectors.

1. Introduction

Blotches and scratches detection in archive video is a common concern in image and video processing. Considerable efforts have been devoted in the last two decades to detect these types of degradation which are considered the most common defects in old videos [1]. Significant achievement has been made by exploring different image and video processing techniques, from MRF modeling [2] and HMM modeling [3] to recently spatio-temporal features [4], LBP and HOG descriptors [11].

A blotch is defined as a set of dark or bright inter-connected pixels which appear randomly in video sequences and caused essentially by the presence of particles, such as hair or dust, on the surface of the storage element of an analog storage media. Spatially, blotches have random shapes and high correlation. They are, however, weakly correlated in the time domain since the probability of appearance of the same blotch in two consecutive frames is very low [18].

Scratches have completely different characteristics compared to blotches. They are produced by the friction of a particle on the film or on the negative during the projection or duplication. Since the traces of these particles follow the unwinding direction of the film, the scratches appear most often vertically, then they are visible as persistent vertical lines on consecutive frames. In this paper, we are considering the vertical scratches as they are the most common type of scratches. The difficulty of detecting scratches relies on the fact that they may share similar characteristics and properties with natural elements of an image in a video sequence.

In this paper, we exploit the recent successes of deep learning models in different application domains. In particular, we are interested in Convolutional Neural Networks (CNN) which have been successfully applied for object recognition [6–8] and detection[9], as well as image retrieval [10]. The main motivations for using CNNs for blotches and scratches detection are:

- Archived videos defects represent a very complex phenomenon generated from various types and shapes of blotches and scratches. Hand-crafted models fail to model these levels on complexities and, therefore, data-driven approaches are more suitable for this role.
- CNNs provides a rich image representation provided by pre-trained features which is more efficient than hand-crafted features described in [11], which consider only the textural information, which is not sufficient in most cases.

We achieve the detection of blotches and scratches in two stages: (1) Pixel-wise classification and description using an encoder-decoder architecture and (2) Post-processing stage which

☆ This paper has been recommended for acceptance by Zicheng Liu.

* Corresponding author.
E-mail address: hyous@usthb.dz (H. Yous).

https://doi.org/10.1016/j.jvcir.2019.02.005
1047-3203/

aims to eliminate false alarms and refine the detections by considering the spatio-temporal artifacts properties. While both types of defects under consideration share the same first, they require two different post-processing in the second.

The main contributions in this article rely on:

- A novel supervised model provided by a fully CNN-based encoder-decoder architecture for defects detection.
- Extraction of pixel-level description from an intermediate stage of the CNN network to be used for detection refinement.
- This paper describes a unified CNN-based defect detection architecture that can be used to detect blotches and scratches simultaneously.

The remainder of this paper is organized as follows. In Section 2, we review related works of blotches and scratches detection, in the literature. Section 3 describes in detail the proposed approach. In Section 4, the evaluation results of the proposed method are presented and discussed. Finally concluding remarks and future work will be covered in Section 5.

2. Related works

2.1. Blotches detection

Various approaches have been proposed in the literature to address blotches detection in archive video. The spatio-temporal blotches properties have been taken into account to this aim and the performance is dependent on the assumptions taken into consideration as well as the efficient handling of the complexity of the blotch's properties and video content. The basic property considered in most of the previous approaches is the low temporal correlation between blotches in video frames compared to the high correlation between real objects, which have high consistency between consecutive frames of a video sequence. The different approaches differs in the way they model and account for the dissimilarity between objects in the consecutive frames.

2.1.1. Dense dissimilarity assessment

The most classical solution relies on dense pixel-based dissimilarity measure. Each pixel in a given frame is classified as blotch pixel if it is dissimilar to the corresponding pixels in the adjacent frames. SDIx [12], ROD [13] and sROD [14] detectors are three different metrics used to measure and quantify this dissimilarity. Pixels motion is estimated and compensated prior to the detection process. The main drawback of these methods is the high dependency of the performance to the accuracy of motion estimation. Multi-scale analysis using wavelet transform [15,16] were proposed to improve the accuracy of motion estimation. Furthermore, the intensity based similarity is very sensitive to the noise and lighting conditions. Therefore, many improvements were proposed by considering other spatial properties of blotches and are mostly based on morphological analysis [17].

2.1.2. Region/block based dissimilarity assessment

Region matching-based dissimilarity is proposed in [4] were spacial consistency is employed and inter-frame similarity is quantified at region level, rather than individual pixels. The authors designed a temporal feature-based detector to remove the false alarms obtained from spatial detection. The pixels labeled as blotches, from the spatial detection step, are grouped in regions to be compared to the most similar regions in the adjacent frames. The candidate regions are labeled as blotches if the dissimilarity is greater than a pre-defined threshold. Although this manner of dissimilarity assessment is more efficient than dense dissimilarity, it is still equally sensitive to the changes in lighting conditions. In order to overcome this limitations, another similarity metric is proposed in [5] by considering gradient information which is less sensitive to light conditions than intensity.

2.1.3. Texture-based dissimilarity assessment

Most recently, regions based method is proposed [11] using textural information. Blotches candidates regions are extracted using a specific spatio-temporal segmentation then dissimilarity is quantified based on textural descriptors, such as Histogram of oriented Gradient (HoG) and Local Binary Pattern (LBP). It is measured between blotches candidates in a given frame and their corresponding regions in the adjacent frames where the association is estimated by means of KLT tracker. The use of such sparse motion estimator and feature descriptors provides more robustness against motion estimation errors and light changes, and is more effective in case of complex motion were global motion compensation is not a plausible solution.

2.2. Scratches detection

Similarly to the existing methods for blotches detection, previous works on scratches detection in archive video sequences rely on their spatio-temporal properties, which are completely different compared to the blotch ones. Based on the considered properties, they could be divided into three main categories:

2.2.1. Filter based methods

The first works on scratches detection in archive videos consider only the spatial information taking into account their main spatial characteristics such as shape, color and texture. Kokaram proposes in [19] to model the vertical scratches based on an empirical observation of their shape. The model parameters are estimated in a Bayesian fashion and Hough detector is used to accelerate the detection. Bruni et al. in [20] extend the earlier work and, rather than estimating the line profile in a Bayesian way, they approximate it with a triangle for each candidate scratch as its energy. Newson et al. in [21,22] use a contrario method which is previously introduced for specific shapes to group the scratch candidate pixels into scratches segments.

2.2.2. Temporal filtering methods

In this category of methods, researchers try to analyze scratches over time. As a first work following this approach, Joyeux et al. in [23,24] use Kalman filter to track scratches. The candidate scratches are validated according to their conformity to the sinusoidal motion model. The major drawback of this method lies in the hypothesis of sinusoidal movement of scratches. Decencière in [25] applies morphological operations in space and time assuming that vertical scratches persist and appear exactly at the same position in consecutive frames. Müler et al. in [26] seek to eliminate false detections by comparing the candidate scratch's movement with that of the neighboring objects in the scene, assuming that the scratches movement is decorrelated with respect to the object's movement. Güllü et al. in [27] use local block matching error to determine whether detected pixels belong to real scratches or not. However, it is very sensitive to contrast changes. In [28], Newson et al. propose to refine the spatial detection proposed in [21] by temporal filtering, where the candidate scratches are validated or rejected depending their compliance with the global affine motion.

2.2.3. Pixel-wise classification based methods

Scratch detection methods based on classification are not numerous. We can cite only two works which have considered scratches detection as pixel-wise classification.

Kim et al. in [29,30] are based on the assumption that the pixels belonging to the scratch have a lower or higher intensity compared to the neighborhood. Indeed, Artificial Neural Network (ANN) classifier is used for pixel wise-classification. The gray-level intensity of the pixels in a rectangular window centered at a pixel being examined is the input of a neural network of 18nodes in a single hidden layer and two neurons in the output layer. Jing Luo and al [31] use the K-Nearest neighbor classifier (KNN) classifier to detect vertical scratches. They use a feature vector which consists of the association of Curvelet Transform with co-occurrence matrix features within 16 sub-blocks with 64×64 pixels.

2.3. Joint blotches and scratches detection

In the literature, the existing method for detection blotches and scratches jointly are very limited. The difficulty of designing a unified model to detect blotches and scratches simultaneously relies on the big difference between their spatial and temporal properties. In fact, line scratches have high temporal correlation and specific shape, unlike blotches that have low temporal and high spatial correlations. Li et al. in [32] proposed to use video decomposition to this aim. First, cartoon-texture decomposition is applied, where texture regions contain scratches while blotches are in cartoon. Then a second temporal decomposition is performed to separate defects from video content. This method may fail in cases of many kinds of blotches that do not fit with the used assumption, such as blotches with linear shapes or slightly textured regions.

3. Proposed approach

Despite all the efforts and achievements to date, detecting blotches and scratches in archive video is still a challenge. As explained in the previous section, hand-crafted models fail to cope with the high complexity of these defects, combined with complexity coming from the video content and its separability from the defects. On the other hand, tremendous progress has been made in the area of Deep Neural Network (DNN) in general and Convolutional Neural Network (CNN) in particular. Unmatched performance, compared to the traditional approaches, has been achieved in various application domains including ones which are very relevant to the problem under consideration in the present paper. The power of CNNs lies in their ability to model very complex pattern and their robustness to noise. Particularly, The CNN filters behave as adaptive filters with higher complexity, able to extract more relevant information hierarchically in a multi-scale fashion, in a wider receptive field than usual filters that are sensitive only to local variations.

Motivated by CNN's contribution to the relevant applications, we propose a method for blotch and scratch detection which consists of two stages: (1) pixel-level classification and description of each video frame into defects pixels or not, by means of a novel CNN-based encoder-decoder architecture depicted in Fig. 1, and (2) spatio-temporal analysis to group and fine-tune the detections. Both types of defects under consideration share same first stage, namely pixel-wise classification, but require two different post-processing.

3.1. Pixel-wise classification

The first step in the detection process is formulated as a pixel-wise classification problem, which assigns the label **1** for defect pixels, and **0** for background ones. Recently, several of CNNs-based approaches have been proposed to address pixel-wise classification problem in various applications, such semantic segmentation [33,34] or objects contour detection [35]. The typical CNN architecture often adopted for this task is what is referred to as Encoder-Decoder. As indicated by its name, it consists of two sub-networks called, respectively, Encoder and Decoder, as depicted in Fig. 1.

3.1.1. Encoder network

This part of encoder-decoder plays the role of feature extractor. In many pixel-wise classification architectures designed for semantic segmentation, it is composed of existing CNNs architecture dedicated for classification tasks, by removing the fully connected (FC) layer. Indeed, SegNet [33] and FCN [34] encoder network consists of 13 convolutional layers which correspond to the first 13 convolutional layers in the VGG16 network [6] designed for object classification, where each convolutional layers is followed by batch normalization and Rectified Linear Unit (ReLU).

In this paper, we propose a new, simple encoder architecture for artifacts detection. We prioritize simplicity and effectiveness in the design of our CNN architecture and, therefore, preferred a minimal design to fit our purpose rather than adopting an existing architecture which has been designed for applications where the features of interest are more complex and higher level of variability. To this end, we propose two architectures.

The first one is composed of 3 layers of convolution, each of which is followed by ReLU layer. The cascade of these layers is followed by Max-Pooling layer with stride 2×2, which reduces the spatial size of the tensor by a factor of 2. We have used for the convolution operation a three-dimensional kernel with the spatial size of 3×3. This could be justified by two essential reasons:

1. since the blotches and scratches could be characterized by local variations, the use of small filters can leverage this information. However, the use of large filters may lose the micro textural information, which more relevant in case of scratches or linear blotches.
2. it is more efficient to use a cascade of small filters rather than use large filters, in terms of the number of parameters. Therefore, the use of three layers of 3×3 convolution can cover the same receptive field as 7×7 convolution do, with fewer parameters and more non-linearity.

The second architecture consists of three convolution layers with outputs 32 , 64 and 128maps respectively. The three convolution outputs are iteratively concatenated, resulting a volume of 224maps mixing. The encoder architecture is depicted in Fig. 2 (a). The design of this unit is equivalent to mixing three scales in a single local descriptors, which has the potential to increase its discrimination capability. In other words, it allows capturing object level properties at the local/pixel level, which we believe make it suitable for pixel-labeling context, which is our goal. The dimensionality of the mixed layers is then reduced by adding 1×1 convolution layer which produces 64plans. Similar concepts have been employed in recent CNN architectures, such as GoogLeNet [36] and ResNet [37]. Table 1 illustrates the per-layer encoder network details.

The comparison between the two architectures is illustrated in Fig. 3. It shows that the architecture with concatenated tensors is more efficient compared to simple architecture in terms of precision. This can be explained by the fact that mixing tensors at different scales provides a richer information which allows a better decision at pixel level. Further numerical comparison between the two architectures is demonstrated in Section 4.

Due to the fact that blotches have a reduced spatial correlation and their sizes are relatively small, we have chosen to use only one Pooling layer. Other applications similar to semantic labeling use

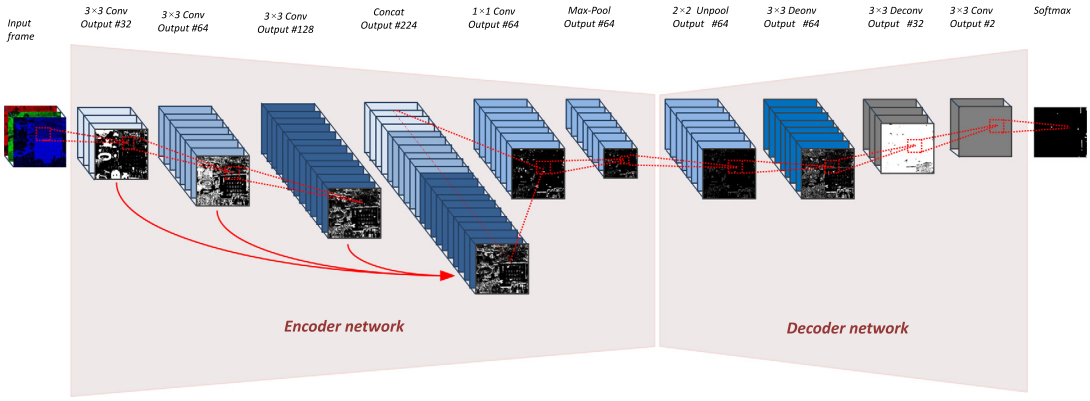

Fig. 1. The proposed encoder-decoder architecture.

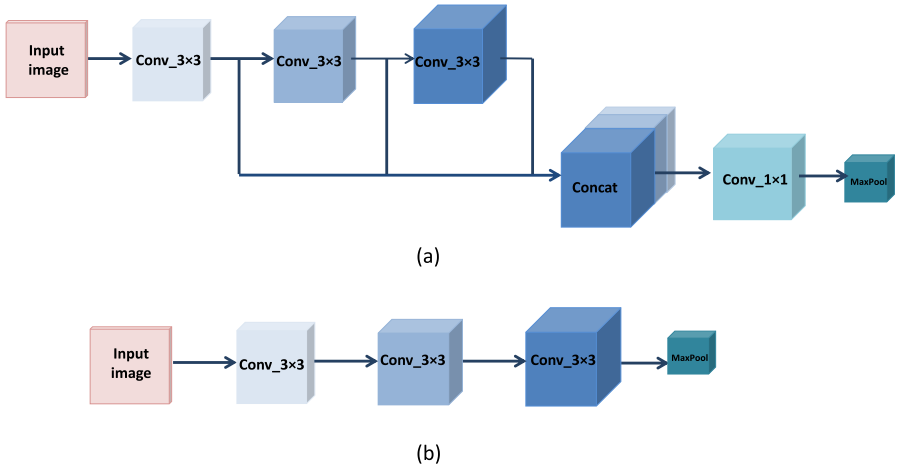

Fig. 2. The architecture of the two versions of the proposed encoder. (a) The use of the concatenation of the three convolution layers, (b) architecture without concatenation.

Table 1
Encoder network architecture details.

Layer	Name	Input size	Kernel	Feature map
Conv	conv1	$320 \times 320 \times 3$	$3 \times 3 \times 3$	32
ReLU	relu1	$320 \times 320 \times 32$	–	32
Conv	conv2	$320 \times 320 \times 32$	$3 \times 3 \times 32$	64
ReLU	relu2	$320 \times 320 \times 64$	–	64
Conv	conv3	$320 \times 320 \times 64$	$3 \times 3 \times 64$	128
ReLU	relu3	$320 \times 320 \times 128$	–	128
Concat	conc	–	–	224
Conv	conv11	$320 \times 320 \times 244$	$1 \times 1 \times 244$	64
Pool	pool1	$320 \times 320 \times 64$	2×2	64

several pooling layers because each pixel to be classified correlates with a wide neighborhood area and, in other words, relates to the global semantics of the image. This is completely different in the case of interest, as blotches are decorrelated with the context and irrelevant to the semantics of the image.

3.1.2. Decoder network

The features extracted from the encoder stage are mapped using a decoding network in order to obtain a detection map of the same size as the original image. We use a simplified version of the SegNet decoder network using two deconvolution layers inserted after a layer of an Upsampling layer. The version of Upsampling, proposed by SegNet, has the advantage that it uses the indexes of the Pooling used in the encoder network. This is more effective in terms of preservation of resolution and efficiency of computation. Moreover, this option is more suitable to our application, since it allows to obtain an accurate pixel mapping in fine structures, which is the case for scratched regions or blotches with linear shape.

Then, a convolution layer with 2 outputs, corresponding to the number of classes, is added. Finally, Softmax, which classifies each pixel independently, with two output layers, referred to as D^tmask, is added. The softmax classifier predicts the probability of each pixel to belong to the defect $y = 1$ or to the background $y = 0$, defined as:

$$P(y = j|z) = \frac{e^{z_j}}{\sum_{k=0}^{1} e^{z_k}} \quad (1)$$

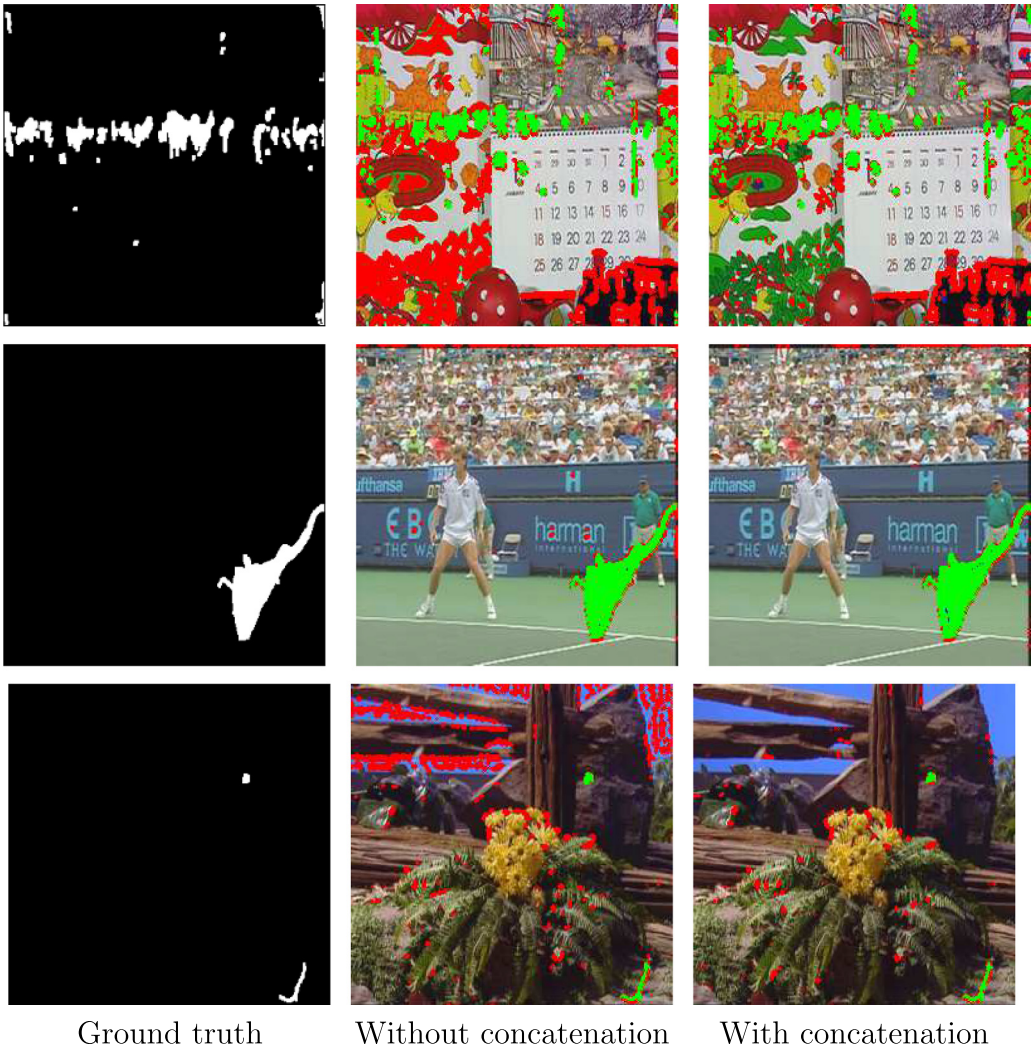

Fig. 3. The comparison between two versions of the proposed encoders. Green: true positive. Red: false positive. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where the z is the last convolution layer output of the decoder network and j is the class label.

Therefore, the predicted segmentation, referred to as D^t mask, corresponds to the class with maximum probability at each pixel, as follows:

$$D^t = \text{argmax}\ P(y|z) \tag{2}$$

Table 2 illustrates the decoder network details, where its input of size $160 \times 160 \times 64$ is the result of the pooling operation of the encoder network.

3.2. Training of the proposed network

We use the "cross-entropy loss" as an objective function for the training of the proposed network. It is calculated as the sum on all pixels of a mini-batch, formulated in [38] as:

$$L(\theta) = -\sum y_i \log P(y_i = 1|X;\theta) + (1 - y_i)\log(1 - (P(y_i = 1|X;\theta)) \tag{3}$$

where θ are the trainable parameters of the encoder-decoder, X is the input image, $y_i \in \{0, 1\}$, with $i = 1, \ldots, |X|$, is the pixel-wise bin-

Table 2
Decoder network architecture details.

Layer	Name	Input size	Kernel size	Feature map
Upsample	unpool1	$160 \times 160 \times 64$	–	64
deconv	deconv2	$320 \times 320 \times 64$	$3 \times 3 \times 64$	32
ReLU	drelu2	$320 \times 320 \times 32$	–	32
deconv	deconv1	$320 \times 320 \times 32$	$3 \times 3 \times 32$	32
ReLU	drelu2	$320 \times 320 \times 32$	–	32
conv	conv	$320 \times 320 \times 32$	$3 \times 3 \times 32$	2
Softmax	softmax	$320 \times 320 \times 2$	–	–

ary label of X. $P(.)$ is the probability of a pixel belonging to a class, obtained by applying the Softmax function to the activation of the last layer (last convolution layer).

In the case of the binary classification, the formula is reduced by the following equation:

$$L(\theta) = -\sum_{i \in Y_1} \log P(y_i = 1|X;\theta) - \sum_{i \in Y_0} \log P(y_i = 0|X;\theta) \quad (4)$$

where Y_1 and Y_0 denote the defect and non-defect ground truth label sets.

When there is a large variation in the number of pixels in each class, the loss function should be weighted differently depending on the class. This is called class balancing. The objective function then becomes:

$$L = -\alpha \sum_{i \in Y_1} \log P(y_i = 1|X) - \beta \sum_{i \in Y_0} \log P(y_i = 0|X) \quad (5)$$

with α and β represent the balancing hyper-parameters.

We use the median frequency equation [33] where the weight assigned to a class in the loss function is the ratio of the median of the class frequencies calculated over the whole training divided by the frequency of the class.

The proposed encoder-decoder has been trained on 200 video frames from 4 of artificially degraded video sequences (city, crew, flower and foreman), where blotches with different size and shape in different locations are added.

Because of the lack of datasets including video sequences which labeled scratches and blotches, we decided to train the architecture for each artifact separately.

For the training of the proposed architecture for blotches, we use videos with synthetic blotches artificially added and associated their ground truth needed for the supervised training process. This dataset consists of 200 copies of 4 video sequences, "City", "Crew", "Flower" and "Foreman", with a variety of content, speed and motion complexity. All frames are re-sized to 256×256.

The training of the proposed network for scratch detection is performed using real degraded videos, named "Knight", "Laurel and Hardey", "California", "Afgrunden" and "gate", suffering several scratched in different locations and intensities. We have divided each of these sequences in two, half for training and the other half for testing. The ground truth of these videos are obtained from [39], used in paper [28]. The size of the frames is set to 320×320.

The encoder and decoder are integrally trained and their weights have all been initialized using "Xavier" filters. Thus, we use the stochastic gradient descent algorithm (SGD) with a fixed learning rate of 0.1 and a momentum of 0.9 using SegNet [40] fork of Caffe framework. The learning was performed until the convergence of the objective function.

3.3. Detection fine-tuning

The result of the pixel-wise classification, mapped in the D^t mask using the encoder-decoder, contains pixels properly classified but also false positives and negatives.

In this section, we focus on the refinement of defects detection by grouping the pixel in regions and remove the false alarms, obtained in the pixel-wise classification stage. The two types of defects require different adapted post-processing stages driven by their relevant spatio-temporal properties.

3.3.1. Blotches detection fine-tuning

In this work, false alarms elimination are based on the main blotches spatio-temporal property, spatio-temporal decorrelation property as defined in [11], which means that the similarity of blotch regions with the neighboring regions in the same and adjacent frames is very weak. False alarms removal is possible by analyzing the consistency of the local features in the temporal axis by considering the motion information. To this end, we evaluate the dissimilarity between each blotch candidate with its neighboring regions in the same frames as well as the adjacent frames. In this proposed method, descriptors generated by the encoder-decoder, at the pixel-wise classification stage, are used to measure dissimilarity instead of using the HOG or LBP descriptor used in [11]. The use of the encoder-decoder trained features is expected to be more efficient than conventional descriptors. In fact, CNN-based

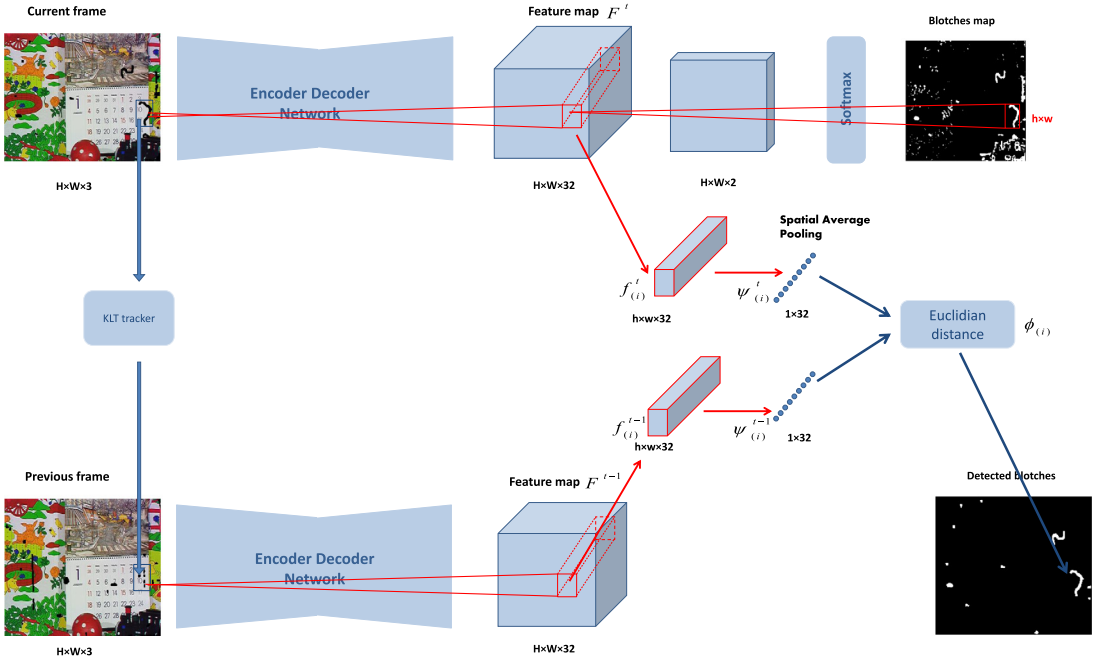

Fig. 4. Blotches detection fine-tuning: we use two streams of encoder-decoder network, one for the current frame and the second for the previous frame. The forward pass on the current frame results a blotches map containing blotches candidates. The second stage consists of using the trained features maps to fine-tune the detection.

descriptors are more appropriate to capture relevant information that can discriminate between blotches regions and other regions belonging to the video's content. Blotches detection fine-tuning process is described in Fig. 4.

The feature map of the last deconvolution layer in the encoder-decoder, shown in Fig. 5, demonstrates how their filters activate some patterns related to the presence of the blotches. Aggregating their responses across the output channels can be seen as an effective local descriptor of blotches. Thus, this descriptor can be used to discriminate them from the original video content. Therefore, we consider the feature map of the last layer of deconvolution as the descriptor map F^t. They are simply obtained by forward passing the current image I^t through the encoder-decoder network, which consists of Cfeature maps, each one have height $H$and width W. The use of the last layer of the decoder network as our feature space is based on the fact that the nearest layers to the output are less sensitive to noise, and therefore they are more discriminating.

First, the pixels classified as blotches in $D^t$are grouped into a set regions denoted as $R^t = \left\{R^t_{(1)}, R^t_{(2)}, \ldots, R^t_{(N)}\right\}$, characterized by their

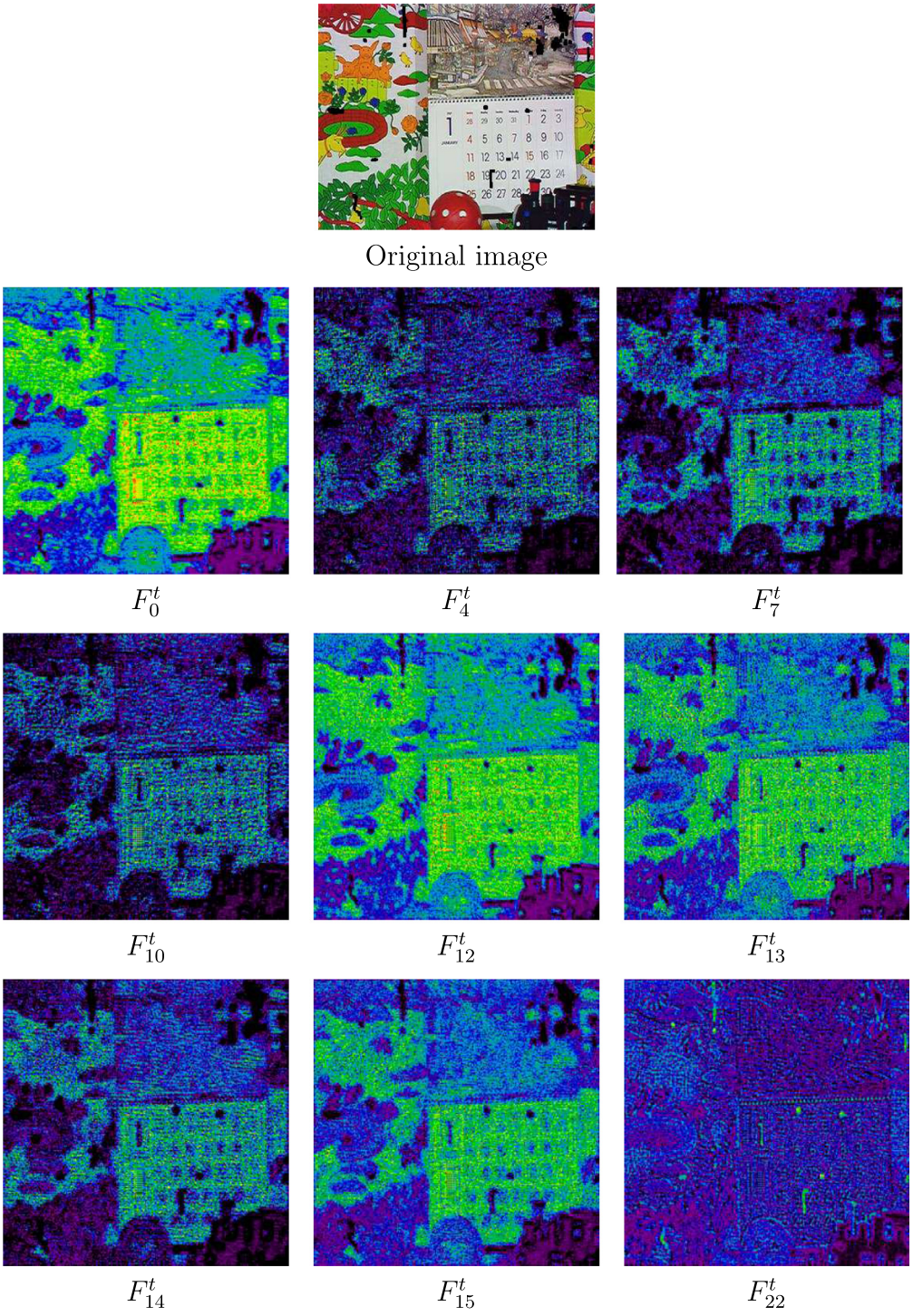

Fig. 5. An image of a degraded archive sequence with some corresponding feature maps.

bounding boxes (see Fig. 6. (c)), with N represents the total number of obtained regions. This is done by means of connected component labeling applied to the binary blotches map D^t. Each region $R^t_{(i)}$ in the current frame is associated the equivalent region in the feature space $f^t = \left\{f^t_{(1)}, f^t_{(2)}, \ldots, f^t_{(N)}\right\}$. Each $f^t_{(i)}$ is assimilated to a cuboid of size $h \times w \times C$, where h and w represent the dimensions of the bounding box of each region $R^t_{(i)}$, and C is the number of channels.

One of the main properties of a blotch is the inconsistency in time, which means it shouldn't appear in the adjacent frames. In order to evaluate this temporal consistency for a candidate region $R^t_{(i)}$, we extract the region in the previous frame, $R^{t-1}_{(i)}$, whose location is determined by means of motion analysis applied to the original images. To this end, we make use of a sparse motion estimation algorithm (Kanade-Lucas-Tomasi feature tracker), similarly to [11], where the centroids $c(R^t)$ are considered as points of interest and their estimated displacements are $d_{(t,t-1)}$. As described in [11], the use of such model-based sparse motion trajectory estimator, rather than matching-based dense motion estimator, is more efficient in terms of accuracy and temporal complexity. The centroïds of the corresponding regions in the previous frame are derived as follows:

$$c(R^{t-1}) = c(R^t) - d_{(t,t-1)} \tag{6}$$

Accordingly, their respective regions at the previous frame $(t1), R^{t-1}_{(i)}$, are extracted together with their respective feature sub-space $f^{t-1}_{(i)}$.

For each region $R^t_{(i)}$, resp. $R^{t-1}_{(i)}$, we derive a descriptor which is the vector noted ψ^t, resp. ψ^{t-1}, obtained by applying global average pooling on f^t, resp. f^{t-1}, defined for each element k as follows:

$$\psi^t_i(k) = \frac{1}{h \times w} \sum_h \sum_w f^t_i(x, y, k) \tag{7}$$

and

$$\psi^{t-1}_i(k) = \frac{1}{h \times w} \sum_h \sum_w f^{t-1}_i(x, y, k) \tag{8}$$

with x and y represent the spatial coordinate and k represents the channel index. It represents also the index of the element in the feature vector. Therefore, the size of the obtained feature vector is the number of channels C of the considered feature space, which is the last output of the last deconvolution layer which $C = 32$.

The dissimilarity for each pair $(R^t_{(i)}, R^{t-1}_{(i)})$ is evaluated using Euclidean distance between the corresponding descriptors $(\psi^t_{(i)}, \psi^{t-1}_{(i)})$, as follows:

$$\phi(R^t, R^{t-1}) = ||\psi^t - \psi^{t-1}|| \tag{9}$$

High distances, $\phi(R^t, R^{t-1})$, indicate a high dissimilarity, which corresponds to a high likelihood that the region under consideration is potentially a blotch. To this end, a threshold is set to discriminate blotches. The experiment results, by taking into account the recall and precision, show that a value around 10 of the threshold provides a good result. Fig. 7 demonstrates a comparison between detection result before and after blotches detection

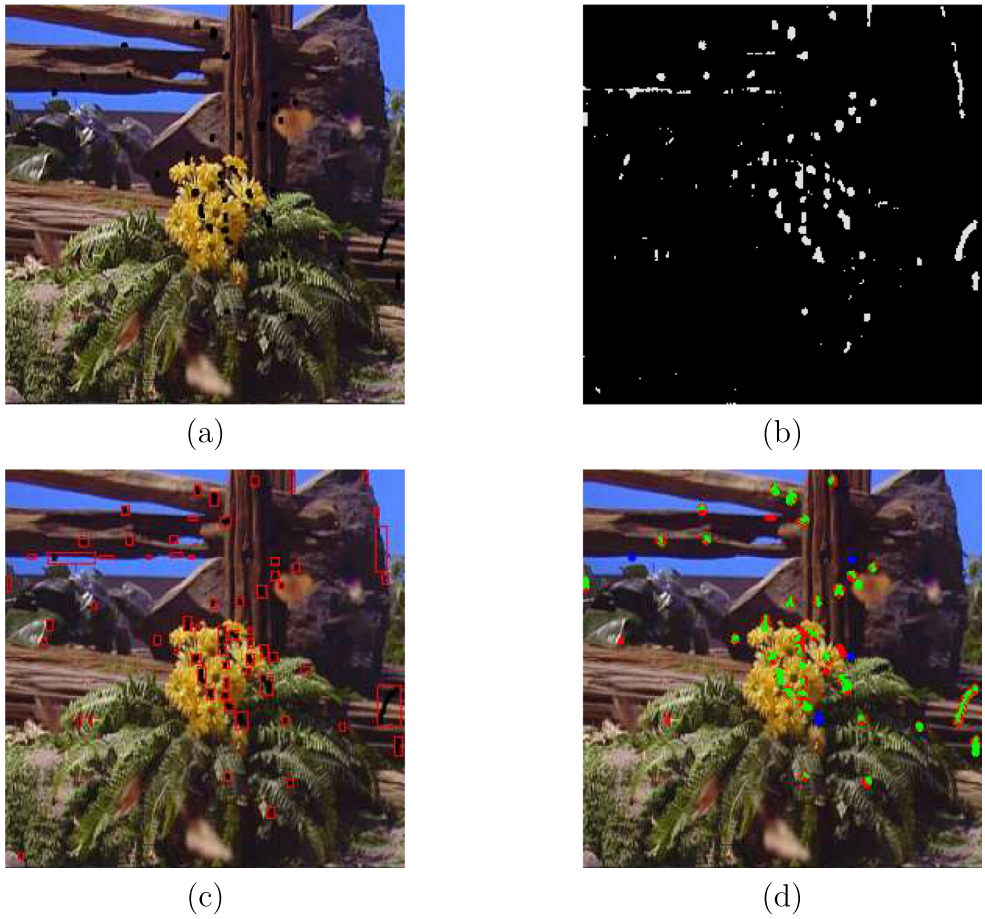

Fig. 6. Blotches detection fine-tuning. (a) Original frame, (b) pixel-wise classification result, (c) candidate regions bounding boxes, (d) final detection result.

fine-tuning. It shows its efficiency in removing false alarms in many regions, which are hardly distinguishable by considering only spatial information.

3.3.2. Scratches detection fine-tuning

False alarms in case of scratches are caused by the presence of linear objects or textures in the original video that share similar properties with scratches in terms of the discontinuity in the horizontal direction. At this stage, scratches detection fine-tuning consists to remove false alarms by filtering the detection map followed by shape analysis. Indeed, the output of pixel-wise classification will further analyzed to discriminate true detection and false alarms.

The detection map, obtained from pixel-wise classification D^t, contains many isolated points and tiny disconnected regions with circular and linear shapes. In the case of scratches, this is often due to low contrast of certain scratches pixels which prevents their detection as connected regions. In order to enhance the scratches pixels continuity, we employ morphological filtering scheme. This could be justified by the fact that the vertical shape is the main scratch property, and morphological operations are more suitable for fit the vertical shape. Hence, apply a closing operation with linear structuring element defined as:

$$D_f^t = \phi_H(D^t) \tag{10}$$

where D_f^t is the resulting filtered map.

Then, the isolated points and the regions with non-linear form in the detection map could be eliminated by shape analysis in regions level. To this aim, we extract the candidate regions, denoted as $R^t = \left\{R^t_{(1)}, R^t_{(2)}, \ldots, R^t_{(N)}\right\}$, which correspond to the connected components in the resulting map of the first step D_f^t.

Vertical scratches are such that their heights are much greater than their widths. In this stage, we use this property to discriminate scratches from the other regions in the detection map. To this end, the calculation of the ratio a ratio γ between the height and the width for each candidate region $R_{(i)}$ is defined as follows:

$$\gamma(R_{(i)}) = \frac{h}{w} \tag{11}$$

with *h*and *w*are the height and width of the candidate region $R_{(i)}$, respectively.

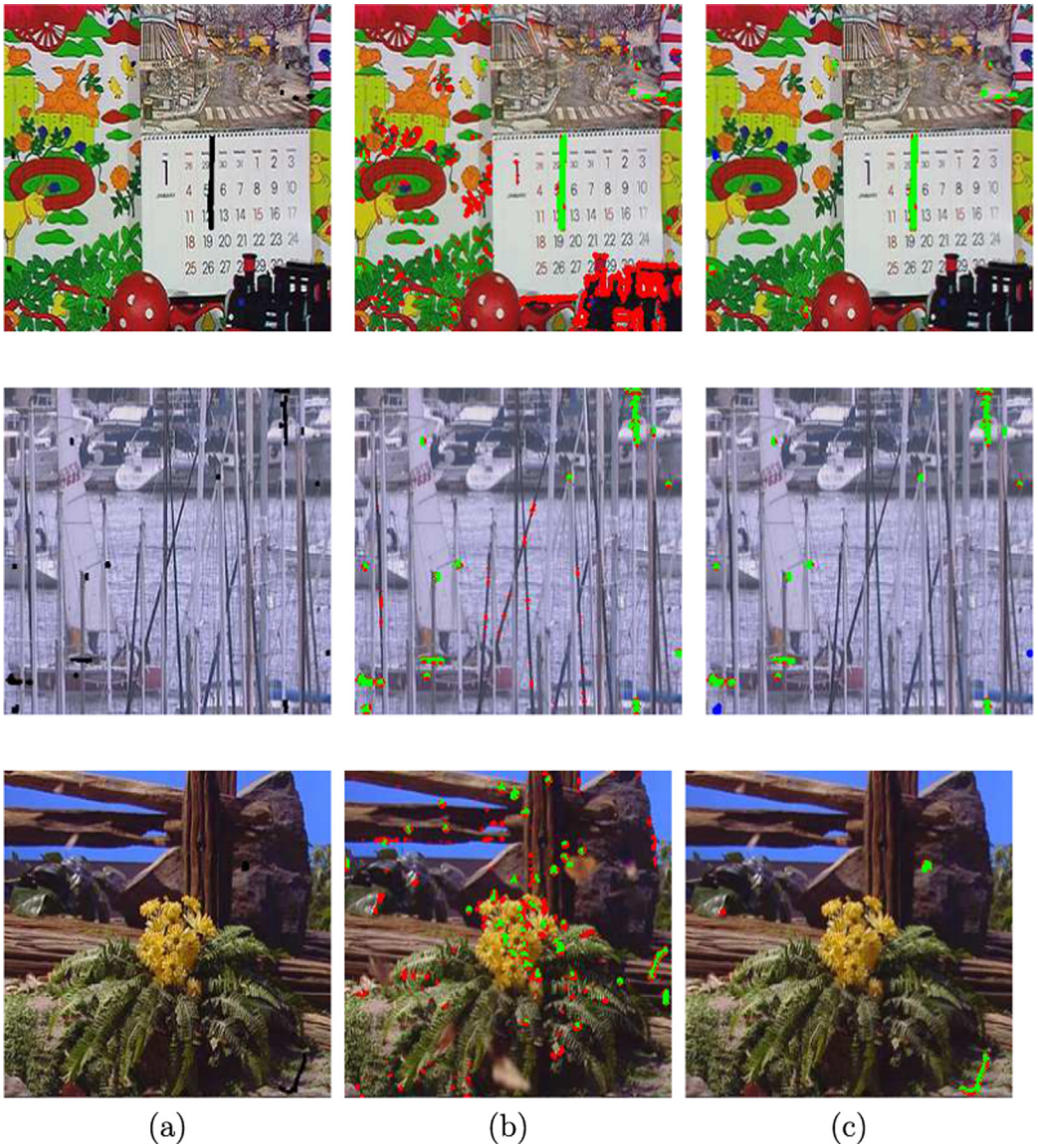

Fig. 7. Blotches detection fine-tuning results. (a) Original frame, (b) detection result before fine-tuning, (c) detection result after fine-tuning.

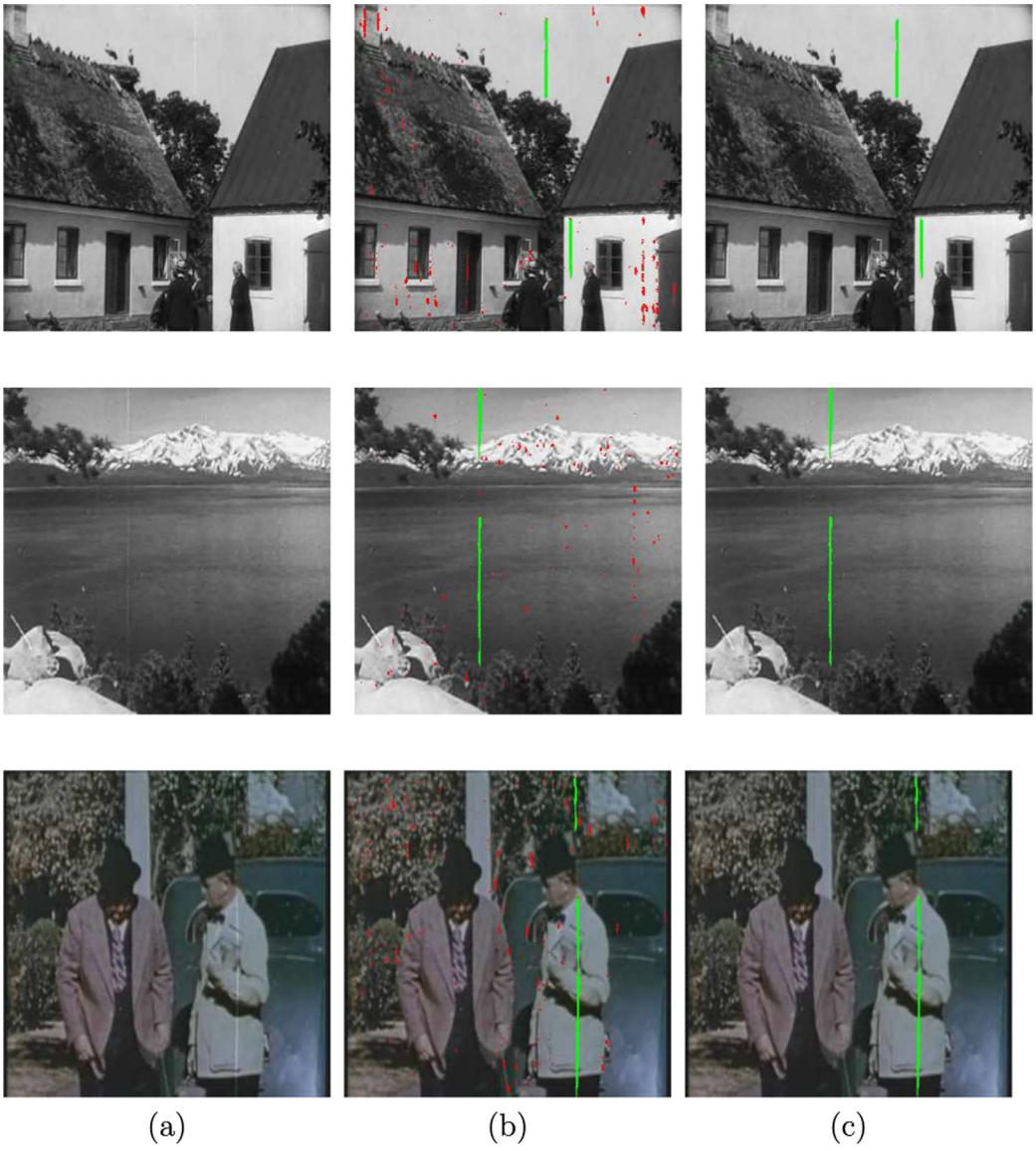

Fig. 8. Scratches detection fine-tuning results. (a) Original frame, (b) detection result before fine-tuning, (c) detection result after fine-tuning.

Table 3
Quantitative comparison between encoder network variants.

Sequence	Metrics	Without concatenation	With concatenation
Bus	Recall	0.9912	**0.9976**
	False alarm	0.0867	**0.0334**
	Precision	0.0962	**0.2093**
Harbour	Recall	0.9879	**0.9976**
	False alarm	0.0053	**0.0045**
	Precision	0.6258	**0.6581**
Soccer	Recall	0.9957	**0.9998**
	False alarm	0.0049	**0.0017**
	Precision	0.5221	**0.7403**
Stephan	Recall	0.9944	**0.9995**
	False alarm	**0.0044**	0.0048
	Precision	**0.4312**	0.3955
Tempete	Recall	0.9990	**0.9979**
	False alarm	0.0214	**0.0066**
	Precision	0.0408	**0.1031**

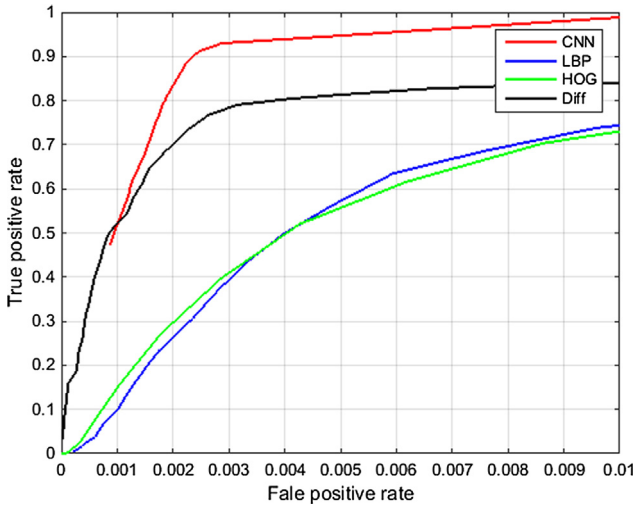

Fig. 9. Quantitative blotches detection evaluation.

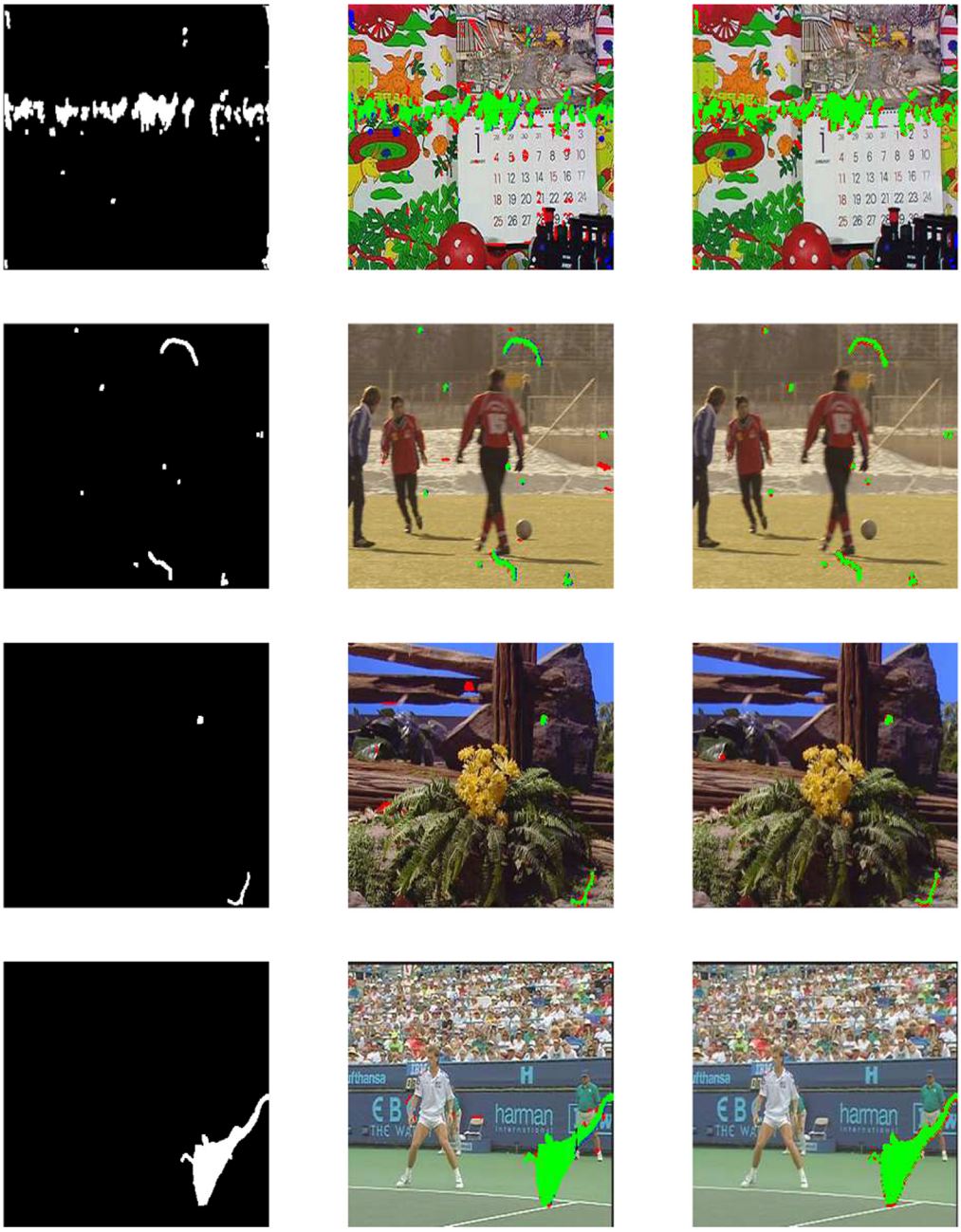

Fig. 10. Blotches detection qualitative evaluation. Left: ground truth. Middle: obtained detection with [11]. Right: obtained result with the proposed method.

Table 4
Quantitative Scratches detection evaluation.

		Afgrunden	Knight	California	Gate	Laurel and Hardy
Recall	Newsson et al. [28]	91.71	77.77	94.74	100	78.95
	Bruni et al. [20]	100	100	36.11	33.33	94.73
	Yous et al. [18]	58.34	76.47	94.87	90.90	92.87
	Proposed method	81.08	83.33	89.74	100	89.47
Precision	Newsson et al. [28]	45.33	93.33	51.31	12.90	18.52
	Bruni et al. [20]	14.44	56.00	8.84	2.98	21.42
	Yous et al. [18]	43.75	92.85	63.79	22.22	25.00
	Proposed method	55.55	100	92.27	25.53	89.47
F1-score	Newsson et al. [28]	60.71	84.84	66.56	22.85	29.96
	Bruni et al. [20]	25.23	71.79	14.20	5.47	18.48
	Yous et al. [18]	25.00	83.86	76.32	17.85	39.39
	Proposed method	**65.93**	**90.90**	**90.98**	**40.67**	**89.47**

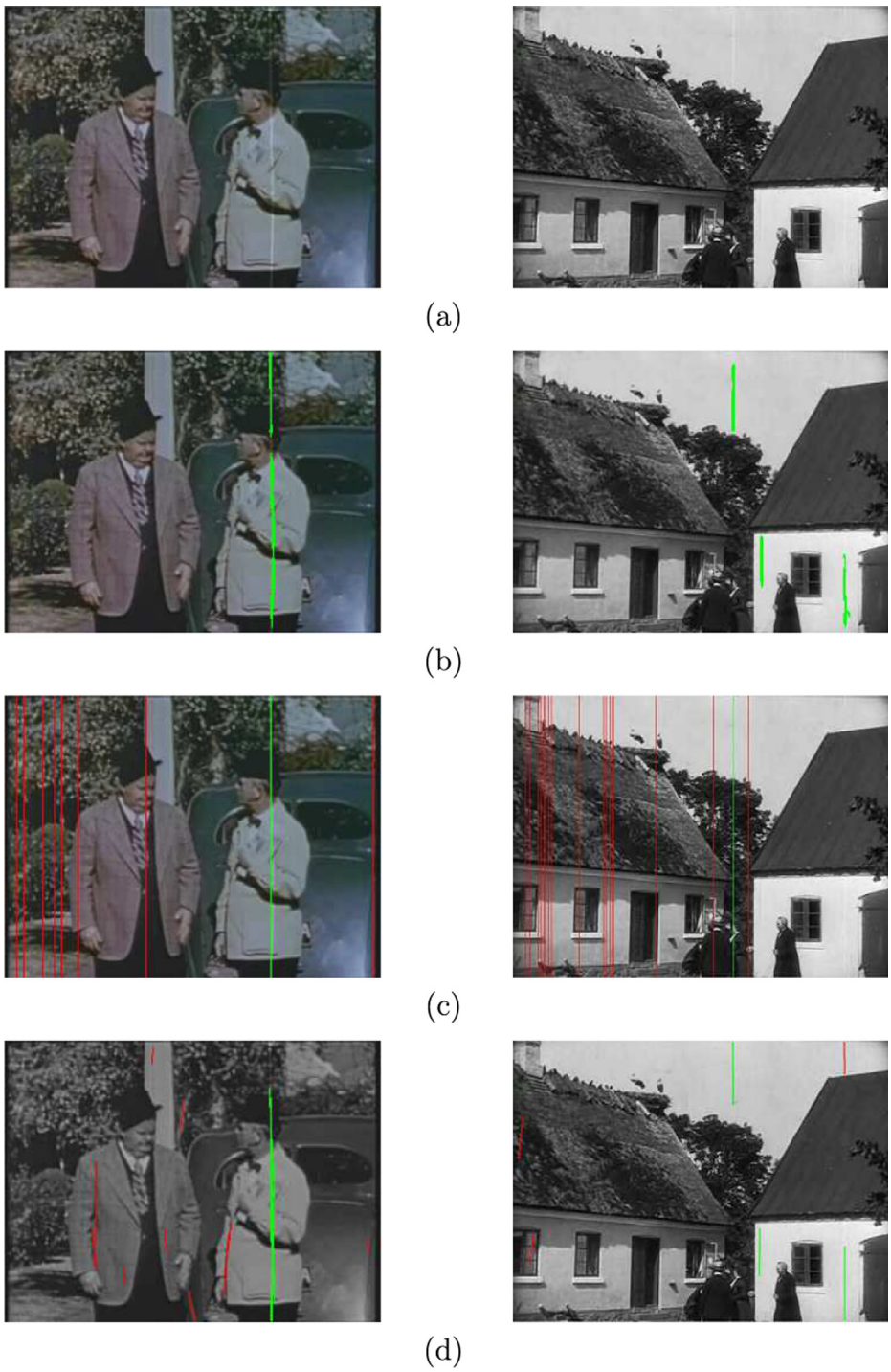

Fig. 11. Scratches detection qualitative evaluation. (a) Original image, (b) detection result with the proposed method, (c) detection result with [20], (d) detection result with [28].

A pre-defined threshold is used to select the true scratches where the ratio is high. The experiments show that a value between 9 as threshold gives a good detection performances. The comparison between the detection results before and after scratches detection fine-tuning depicted in Fig. 8 shows that the use of scratches detection fine-tuning stage enhances the detection quality by removing many isolated points and enforce the continuity in vertical scratches.

4. Experiments and discussions

The performances of the proposed detectors are compared quantitatively against other approaches with respect to four criteria: recall, precision, false alarms rate and the F1-score. Recall is defined as the number of true detection divided by the total number of true scratches or blotches present in an image. Precision is defined as the number of true detection divided by the total

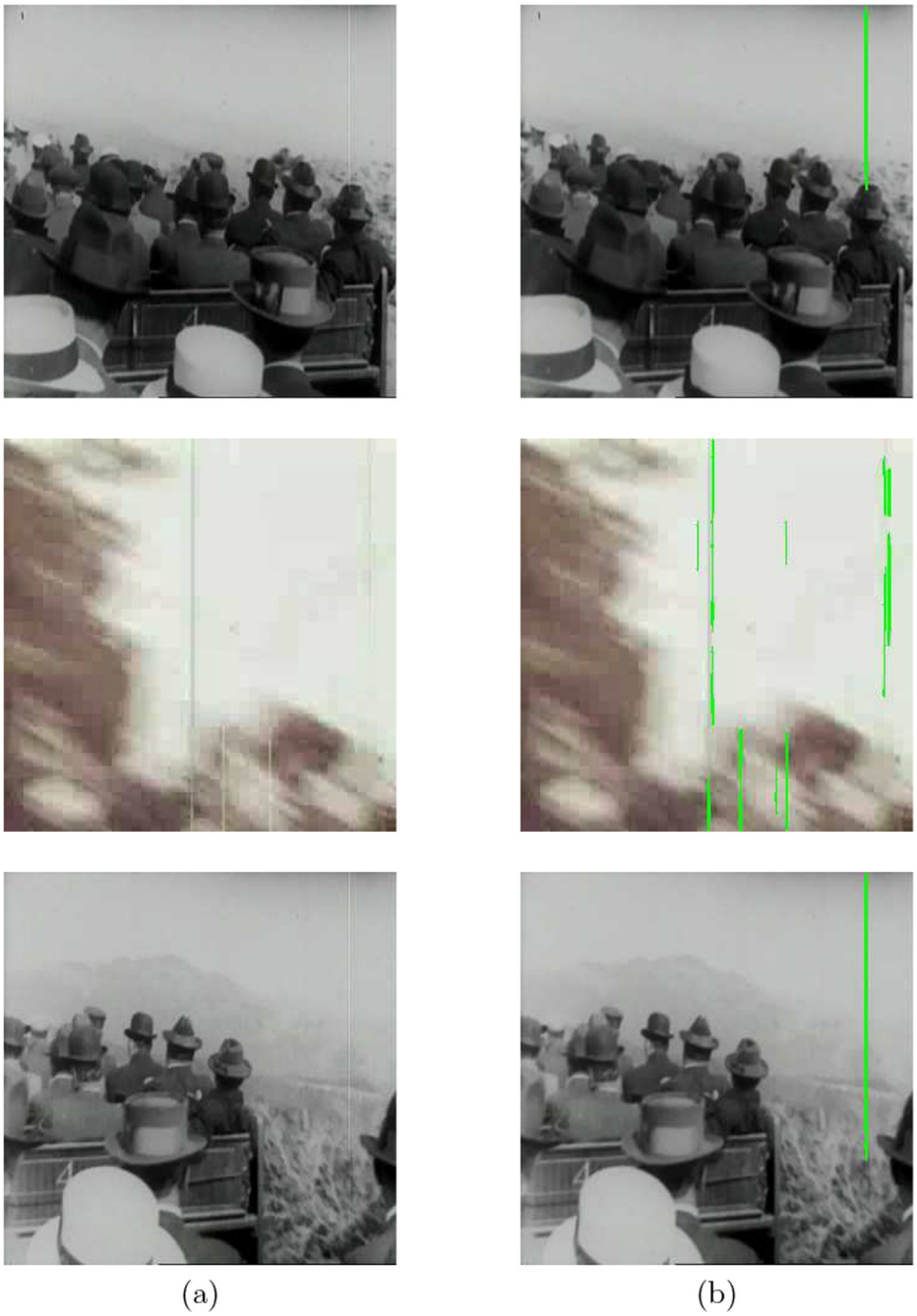

Fig. 12. Scratches detection result for unseen sequence (mount). (a) Original frames, (b) scratches detection result.

number of detection. False alarms rate is defined as the number of false positives divided by the total number of negatives. Particularly, we have defined the true positives, false positives and false negatives based on pixels for blotches detection. However, we have considered regions in case of scratches detection. Herein, we followed the same strategy adopted in the most recent works in blotches and scratches detection to be as fair as we can. The F1-score is a reflection of both recall and precision, and defined as:

$$F1 - score = \frac{2 \times recall \times precision}{recall + precision} \quad (12)$$

4.1. Encoder variants

In order to show the effectiveness of using concatenation between convolution layers, the two encoder variants are compared quantitatively. Table 3 demonstrates this comparison using five videos suffering from synthetic blotches. It shows that the architecture with concatenation is more efficient than the one where all layers are stacked. In fact, detection performances in terms of recall, precision and false alarm, highlighted in the table, are mostly higher for the architecture with concatenation than those of the regular architecture.

4.2. Blotches detection evaluation

The aim in this section is to compare the performance of our proposed CNN-based blotch detection against the ones based on handcrafted descriptors. Therefore, we compare the proposed approach against the regions based method description [11], with respect to three variants: Hog descriptor, LBP descriptor and an improved variant by taking gray level information into account. This test is performed on 200 images from the 5 videos with synthetic blotches not included training "Mobile", "Harbour", "Soccer", "Stephan" and "Tempete".

The quantitative evaluation is done by Receiver Operating Characteristic (ROC) curve in Fig. 9. The average of correct detection ratios is plotted against the average false alarm ratios computed for the 250 frames from each sequence. For each algorithm, the key parameters are varied to measure its performance by means of the ROC graph. Fig. 10 illustrates the qualitative comparison between the CNN-based and Hand-crafted features based method. The quantitative and the qualitative evaluations demonstrate clearly the superiority of the CNN-based method.

For the qualitative evaluation, the Fig. 10 depicts the comparison of the obtained results from the proposed detector against those obtained from the other detectors applied on real degraded frames. It shows that the proposed algorithm outperforms the considered detectors. Therefore, it illustrates the advantage of using the trained features over traditional descriptors.

4.3. Scratches detection evaluation

We compare the proposed scratches detector with three different methods, Bruni's method [20], Newsson method [28] and our old method [18], base on spatio-temporal morphological filtering, which are the most recent algorithms for scratches detection. The resulting detection maps of [20] are obtained from the official web site [39] and the best parameters are selected for [28].

Table 4 represents the qualitative evaluation according to recall, precision and F1-score. This later reflects the trade-off between the amount of the detection and the accuracy. Their values are still significantly higher for our approach compared to other methods, which demonstrates clearly that our method outperforms its counterpart. The Fig. 11 depicts the same conclusions in a visual way, where our approach detects scratches with less false alarms. More qualitative evaluations are demostrated in Fig. 12.

5. Conclusion and perspectives

In this paper, we presented a framework for detecting the main defects in archive videos, namely scratches and blotches, using deep convolutional neural networks. The detection is formulated as a pixel-wise classification problem and a new architecture is proposed to classify each pixel in the original frame as defectuous or not. Subsequently, post-processing is applied to group the candidate regions and eliminate the false detections by taking into account the respective properties of the two defect types under consideration. We were able to show that the use of convolutional neural network for the detection of artifacts in videos is more efficient than the conventional methods based on handcrafted features.

The same encoder-decoder architecture has been used for blotches and scratches detection, despite the fact that these defects have completely different properties. This model can be trained to detect other types of defects in videos that can be formulated as pixel classification, such as the horizontal scratches. This requires the availability of the labeled dataset to train the model to detect the other types of defects.

It is worth noting that the computation complexity of the proposed CNN encoder-decoder is very low compared to the existing CNN encoder-decoder architectures used for other applications (e.g., SegNet). This is because of the low complexity of the proposed architecture reflected by the number of the convolution layer used. Therefore, the proposed architecture could be used for other applications where the computation complexity is a relevant constraint.

The proposed technique to fine-tune the pixel-wise classification, by exploiting the feature extracted in the intermediate layers, can be exploited for other application of video segmentation. Based on the same idea, feature aggregation from the deconvolution layers could provide a more accurate image representation for many applications such as image retrieval.

References

[1] A. Kokaram, Ten years of digital visual restoration systems, in: 2007 IEEE International Conference on Image Processing, vol. 4, 2007, pp. IV – 1–IV – 4. doi:https://doi.org/10.1109/ICIP.2007.4379939.

[2] A. Kokaram, On missing data treatment for degraded video and film archives: a survey and a new bayesian approach, IEEE Trans. Image Process. 13 (3) (2004) 397–415, https://doi.org/10.1109/TIP.2004.823815.

[3] X. Wang, M. Mirmehdi, Archive film defect detection and removal: an automatic restoration framework, IEEE Trans. Image Process. 21 (8) (2012) 3757–3769. <http://dblp.uni-trier.de/db/journals/tip/tip21.html>.

[4] Z. Xu, H.R. Wu, X. Yu, B. Qiu, Features based spatial and temporal blotch detection for archive video restoration, Signal Process. Syst. 81 (2) (2015) 213–226. <http://dblp.uni-trier.de/db/journals/vlsisp/vlsisp81.html>.

[5] H. Yous, A. Serir, Spatio-temporal scratches detection in archive video, in: 3rd International Conference on Signal, Image, Vision and their Applications (SIVA'15), 2015.

[6] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR abs/1409.1556.

[7] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Computer Vision and Pattern Recognition (CVPR), 2015. <http://arxiv.org/abs/1409.4842>.

[9] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Proceedings of the 28th International Conference on Neural Information Processing Systems, NIPS'15, MIT Press, Cambridge, MA, USA, 2015, pp. 91–99. <http://dl.acm.org/citation.cfm?id=2969239.2969250>.

[10] E. Mohedano, A. Salvador, K. McGuinness, F. Marqués, N.E. O'Connor, X. Giró i Nieto, Bags of local convolutional features for scalable instance search, CoRR abs/1604.04653. <http://arxiv.org/abs/1604.04653>.

[11] H. Yous, A. Serir, Efficient region-based approach for blotch detection in archived video using texture information, J. Electron. Imaging 26 (2) (2017) 023019, https://doi.org/10.1117/1.JEI.26.2.023019.

[12] A.C. Kokaram, R.D. Morris, W.J. Fitzgerald, P.J.W. Rayner, Detection of missing data in image sequences, IEEE Trans. Image Process. 4 (11) (1995) 1496–1508. <http://dblp.uni-trier.de/db/journals/tip/tip4.html>.

[13] M.J. Nadenau, S.K. Mitra, Blotch and scratch detection in image sequences based on rank ordered differences, in: Proc. of 5 th Int. Workshop on Time-Varying Image Processing, Elsevier, 1997, pp. 27–35.

[14] P.M.B.V. Roosmalen, Restoration of archived film and video (Ph.D. thesis), Delft University of Technology, 2006.

[15] H. Ammar-Badri, A. Benazza-Benyahia, A novel spatio-temporal method for blotch detection in the wavelet transform domain, in: 2009 3rd International Conference on Signals, Circuits and Systems (SCS), 2009, pp. 1–5. doi: https://doi.org/10.1109/ICSCS.2009.5412593.

[16] H. Ammar-Badri, A. Benazza-Benyahia, Wavelet-based blotch detection in old movies exploiting interscale dependency, in: 2010 10th International Conference on Information Sciences Signal Processing and their Applications (ISSPA), 2010, pp. 708–711. doi:https://doi.org/10.1109/ISSPA.2010.5605574.

[17] S. Tilie, L. Laborelli, I. Bloch, Blotch detection for digital archives restoration based on the fusion of spatial and temporal detectors, in: FUSION, IEEE, 2006, pp. 1–8. <http://dblp.uni-trier.de/db/conf/fusion/fusion2006.html>.

[18] H. Yous, A. Serir, Blotch detection in archived video based on regions matching, in: 2016 International Symposium on Signal, Image, Video and Communications (ISIVC), 2016, pp. 379–383. doi:https://doi.org/10.1109/ISIVC.2016.7894019.

[19] A. Kokarman, Detection and removal of line scratches in degraded motion picture sequence, Signal Process. VIII (1996) 292–295.

[20] V. Bruni, D. Vitulano, A. Kokaram, Line scratches detection and restoration via light diffraction, in: Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis, 2003. ISPA 2003, vol. 1, 2003, pp. 5–10. doi:https://doi.org/10.1109/ISPA.2003.1296858.

[21] A. Newson, P. Pérez, A. Almansa, Y. Gousseau, Adaptive line scratch detection in degraded films, in: Proceedings of the 9th European Conference on Visual Media Production, CVMP '12, ACM, 2012, pp. 66–74, https://doi.org/10.1145/2414688.2414697.
[22] A. Newson, P. Pérez, A. Almansa, Y. Gousseau, Adaptive pixel-precision line scratch detection in degraded films, in: Proceedings of the 9th European Conference on Visual Media Production, CVMP '12, ACM, New York, NY, USA, 2012, pp. 66–74.
[23] L. Joyeux, S. Boukir, B. Besserer, Film line scratch removal using kalman filtering and bayesian restoration, in: 5th IEEE Workshop on Applications of Computer Vision, 2000, p. 8.
[24] L. Joyeux, O. Buisson, B. Besserer, S. Boukir, Detection and removal of line scratches in motion picture films, in: CVPR, IEEE Computer Society, 1999, pp. 1548–1553.
[25] E. Decenciere, restauration des films anciens (Ph.D. thesis), ENSMP, Paris, 1999.
[26] S. Muller, J. Buhler, S. Weitbruch, C. Thebault, I. Doser, O. Neisse, Scratch detection supported by coherency analysis of motion vector fields, 2009, 89–92. doi:https://doi.org/10.1109/ICIP.2009.5414090.
[27] M. Kemal Gullu, O. Urhan, S. Erturk, Scratch detection via temporal coherency analysis and removal using edge priority based interpolation, 2006 IEEE International Symposium on Circuits and Systems, Island of Kos, 2006, (2006) pp. 4, https://doi.org/10.1109/ISCAS.2006.1693652.
[28] A. Newson, A. Almansa, Y. Gousseau, P. Pérez, Robust automatic line scratch detection in films, IEEE Trans. Image Process. 23 (3) (2014) 1240–1254, https://doi.org/10.1109/TIP.2014.2300824.
[29] K. tai Kim, E.-Y. Kim, Automatic film line scratch removal system based on spatial information, IEEE International Symposium on Consumer Electronics, 2007. ISCE 2007 (2007) 1–5.
[30] K. tai Kim, B. Kim, E.Y. Kim, Automatic restoration of scratch in old archive, in: ICPR, IEEE, 2010, pp. 468–471.
[31] J. Luo, S. Lin, S. Li, Morphology battery scratch detection combining GLCM and CT, Journal of Computational Information Systems 7 (12) (December 2011) 4343–4350.
[32] H. Li, Z. Lu, Z. Wang, Q. Ling, W. Li, Detection of blotch and scratch in video based on video decomposition, IEEE Trans. Circuits Syst. Video Technol. 23 (11) (2013) 1887–1900, https://doi.org/10.1109/TCSVT.2013.2269016.
[33] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, CoRR abs/1511.00561. <http://arxiv.org/abs/1511.00561>.
[34] E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for semantic segmentation, CoRR abs/1605.06211. <http://arxiv.org/abs/1605.06211>.
[35] J. Yang, B.L. Price, S. Cohen, H. Lee, M. Yang, Object contour detection with a fully convolutional encoder-decoder network, CoRR abs/1603.04530. <http://arxiv.org/abs/1603.04530>.
[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Computer Vision and Pattern Recognition (CVPR), 2015. <http://arxiv.org/abs/1409.4842>.
[37] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.
[38] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, L. Van Gool, One-shot video object segmentation, in: Computer Vision and Pattern Recognition (CVPR), 2017.
[39] Adaptive pixel-precision line scratch detection in degraded films, accessed: 2018-02-02. <https://perso.telecom-paristech.fr/gousseau/scratch_detection/>.
[40] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, CoRR abs/1511.00561. <http://arxiv.org/abs/1511.00561>.