Correl-Net: Defect Segmentation in Old Films Using Correlation Networks

Arthur Renaudeau(✉), Travis Seng, Axel Carlier, and Jean-Denis Durou

IRIT, UMR CNRS 5505, Université de Toulouse, Toulouse, France
arthur.renaudeau@irit.fr

Abstract. The old film restoration process involves many operations, one of which is the ability to identify defects that altered the film. This operation can be formulated as a binary segmentation problem and solved using state-of-the-art segmentation networks such as `DeepLab v3+` or `NAS-FPN`. While being very powerful at describing the spatial characteristics of defects, these methods fail to take into account the fact that defects are also temporal anomalies. We therefore propose an architecture that builds on the correlation layer introduced in `FlowNet` to compensate for motion and eliminate potential false positives, features that look like defects but can be tracked over multiple images and are actually part of the scene. We also introduce a self-supervised pre-training process of the network, which precedes a fine-tuning phase to specifically adapt the detector to each film. Results show that our architecture, while being more compact and less resource-consuming than state-of-the-art methods, achieves higher precision and recall.

Keywords: Segmentation · Neural networks · Correlation

(a) Original frame. (b) Detected defects (in green).

Fig. 1. Detection of defects (1b) in an old film frame (1a) with `Efficient Correl-Net`. (Color figure online)

R. Gade et al. (Eds.): SCIA 2023, LNCS 13886, pp. 245–259, 2023.
https://doi.org/10.1007/978-3-031-31438-4_17

1 Introduction

After more than a century of cinema, films on reels are legion, and the question arises as to how to preserve these works for the future. Preservation and restoration are necessary for the survival of the great cinematographic classics so that their condition remains as close as possible to the original version. Inevitably, with time, human intervention and wear and tear, films deteriorate beyond repair. Fortunately, preservation and restoration techniques can reduce the damage to these physical media. Restoration can be defined, according to Usai, as "a set of technical, editorial, and intellectual procedures aimed at compensating for the loss or degradation of the moving image artifact" [26]. This definition goes beyond the simple idea of restoring the movie to its original state (assuming one can give a proper definition of the term "original"), as not all losses can, or should, be compensated for. As a practitioner himself, Busche [2] argues that "physical losses in film artifacts (a scratch, for example) are of concern only as a disturbance of the visual appearance of the image", which implies that not all defects should be corrected. The same idea is defended by Philippot [17]: "The restorer has to distinguish between lacunae that have to be reintegrated and those that should be left untouched".

Recent work have considered video restoration as a single task and proposed end-to-end solutions [6], but they do not really allow a human restorer to guide the process, which may be problematic with respect to restoration ethics. Instead, we choose to consider restoration as a two-step approach in which defects are first detected automatically, as illustrated in Fig. 1, and then proposed for validation to a restorer before being corrected using video inpainting techniques [18,28]. This allows the restorer to eliminate false positives, to identify undetected defects or to discard a subset of the true positives that he/she thinks should not be corrected.

Defect segmentation is a particular case of segmentation problems in the sense that the temporal aspect is essential. Even if blotches somehow have distinct shapes, their physical origin on the reel (dust, reel deterioration, etc.) implies these defects can not appear on consecutive frames. In other words, they can be considered as temporal anomalies, a characteristic that should guide the detection process.

In this paper, we draw inspiration from the FlowNet architecture [4] and introduce a convolutional neural network particularly suited for defect segmentation. Given three consecutive frames as input, we build an encoder in which features are extracted from the three frames separately and then compared using a correlation layer. We then reconstruct a defect mask using a simple decoder. We show that the use of these correlations leads to better performances for defect segmentation than state-of-the-art architectures. Furthermore, we propose a self-supervised pre-training process in which realistic defects are artificially generated during training, reducing the need for supervised data. A small supervised training set is required to fine-tune the network and adapt it to a particular new film. This paper is organized as follows: we begin by reviewing related work in Sect. 2, then detail our method in Sect. 3 and our experiments in Sect. 4.

2 Related Work

2.1 Defect Detection Before Deep Learning

Early video defect detectors were developed by the BBC to detect impulse noise [24]; they consist in thresholding the absolute differences between consecutive frames. However, this method achieves limited results because it does not take motion into account. In [11], this limitation is overcome by introducing the Spike Detection Index (SDI), in which motion is compensated before computing the absolute differences.

To specifically detect line scratches, [9] introduced a method where they are modeled as damped sinusoids. Detection is performed in a two-step scheme: subsampling and filtering to gather candidates, followed by Bayesian refinement to eliminate those that do not fit the damped sinusoid model. Subsequently, [16] extended [9] by adding an additional verification step, where the values of neighboring pixels around candidate scratches are examined to distinguish real scratches from simple edges. To further limit false positives, the same authors realign in [15] the different mask images using motion estimation and eliminate candidates that remain vertical, as opposed to real scratches that will distort (twist effect) with motion compensation. Another method, which was used in [7] to detect scratches, is morphological closure. The subtraction between before and after closing images reveals the scratches, which are tracked over several frames using a Kalman filter. The same idea is also present in [20], but the frame is first divided into horizontal bands in order to treat the foreground and the background separately. The detection is thus made easier in homogeneous areas.

Concerning blotches, the first MRF model of [10] has been reused in [27], together with another MRF that allows to reinforce the spatial continuity. In addition, motion compensation is applied to maintain temporal consistency. Also using motion compensation, the ROD detector [14] compares the current pixel with its temporal neighbors to detect spatio-temporal anomalies. Other methods, such as [29], require several steps to detect blotches. After identifying candidates based on their spatial features, false positives are eliminated by searching for temporal discontinuities. Median filters are used in [30] to extract candidates at spots where abrupt spatial changes occur. Then, if these gradients appear only in the current image, and not in previous and subsequent images, these candidates are considered as real blotches.

2.2 Detection and Segmentation by Deep Learning

The first application using neural networks, presented in [8], deals with the detection of line scratches using the cartoon-texture decomposition. While the detection of the shape (cartoon) is performed by filtering, the texture is classified by a neural network taking as input edge images. Another application was proposed in [22] to detect blotches using motion compensation, SROD detection [1] and classification of all outlier pixels using a convolutional neural network.

In another three-step approach, [23] creates a descriptor that contains the luminance of three consecutive images, as well as the motion-compensated images and the Lucas-Kanade optical flow amplitude. In a second step, an SDI detection is performed and finally, in a third step, its result and the descriptor are passed as input to a CNN. For both blotches and scratch detection, [31] applies a classification with an encoder-decoder CNN architecture, with concatenation in the encoder part. A spatial average is performed at the output of the network before the last convolution, followed by a thresholding to detect blotches. Scratches are detected a posteriori by morphological closing of the network output.

Defect detection can also be presented as a standard binary segmentation problem. Convolutional neural networks have been used for this task since [12]. The problem is formulated as a binary classification at each pixel location, and the network must produce a dense foreground probability map. Two main characteristics of network architectures can be found in the state of the art. The auto-encoder architecture is often at the heart of the network; it consists of an encoder that performs feature extraction like a standard classifier, as well as a decoder that performs oversampling of the latent space with the encoder intermediate feature maps. Several variants of this concept have been proposed, the most popular being `U-Net` [19] for skin lesion segmentation. A second common module in the architecture of segmentation networks is the spatial pyramid of features, introduced to combine encoder features at different scales to more efficiently recognize objects at different scales. Spatial pyramid pooling is an example of this technique, in which multiple convolutions of varying kernel size are applied simultaneously to the feature map to extract multi-scale information. This idea is used in `DeepLab v3+` [3], one of the highest performing networks, combined with an auto-encoder architecture. Some of the most recent and successful work, such as `NAS-FPN` [5], has attempted to learn how to optimally combine multiple scales of the feature map.

As is common with deep neural networks, all these architectures can be adapted to different convolutional backbones (or encoders). The original `U-Net` is based on the well-known `VGG` architecture [21], while `NAS-FPN` is built over `Efficient-Net` [25], a much more recent and performant architecture.

2.3 Motion Compensation with FlowNet

In order to efficiently detect defects in films, we take inspiration from `FlowNet` [4], a neural network originally proposed to predict optical flow from a pair of consecutive frames in a video. The authors of `FlowNet` designed a `U-Net` architecture capable of finding correspondences between the patches of the two images using a correlation layer. This correlation layer contains no parameters, and simply convolves the feature maps computed separately for the two images. In our case, the correlation layer is very useful to find regions that have no correlation with the neighboring patches of the previous and subsequent images of the video, indicating a high probability that they are defects. The correlation layer should also help to rule out features in the scene that look like defects by matching them in several consecutive frames.

3 Correl-Net

As previously stated, we formulate defect detection as a binary segmentation problem.

3.1 Network Architecture

We introduce a generic architecture for defect detection that is called `Correl-Net`, and that is depicted in Fig. 2. `Correl-Net` takes as input three consecutive frames of size 256×256 and produces a single probability map of the same dimension. Overall, `Correl-Net` is based on a simple `U-Net` architecture, to which some modifications have been made. First, the encoder consists of three separate feature extractors (one for each input frame) that share the same weights. Second, correlation layers are applied between the feature maps of frames i and $i-1$, and those of frames $i+1$ and i, the output of which is concatenated with the feature maps of frame i. Third, skip connections are used to connect the encoder and decoder, and originate from the central feature extractor in the encoder (the one for frame i). Correlation layers take the form of the one described in [4]. In our implementation, we use Tensorflow correlation layers with the following hyperparameters: kernel size equal to 3, maximum displacement of 10, input stride of size 1 and stride of patch of size 1.

Just as `U-Net` can be adapted to a different backbone, we implemented another version of `Correl-Net` using `EfficientNetB7` [25] as an encoder, also with three consecutive frames of size 256×256. We place the correlation layers just after the first convolutional layer in the network, and before the `MBConv` blocks that constitute the rest of the encoder. In the rest of the paper, we call this new version `Efficient Correl-Net`.

3.2 Loss Function and Training Details

The loss function for network training is the opposite of the linear approximation of the Dice coefficient (or F1-score), whose expression is as follows:

$$\begin{aligned}\text{Loss}\,(y,\hat{y}) &= -\frac{2\sum_{i,j} y(i,j)\hat{y}(i,j)}{\sum_{i,j} y(i,j)+\hat{y}(i,j)} \\ &\approx -\frac{2\times TP}{(TP+FP)+(TP+FN)} \in [-1,0]\end{aligned}$$

where $y(i,j) \in \{0,1\}$ and $\hat{y}(i,j) \in [0,1]$ are the values of the ground truth defect mask and the network output defect mask, respectively. The values TP, FP and FN represent, respectively, the numbers of pixels counted as true positives, false positives and false negatives. This loss function is often a good choice for segmentation, in the case where there is a significant imbalance between classes [13], which is the case in our application. The model is trained with Adam optimizer, with an initial learning rate of 10^{-5}, and reaches convergence after 150 epochs. We use Tensorflow 2 and train our model on an Nvidia RTX A6000.

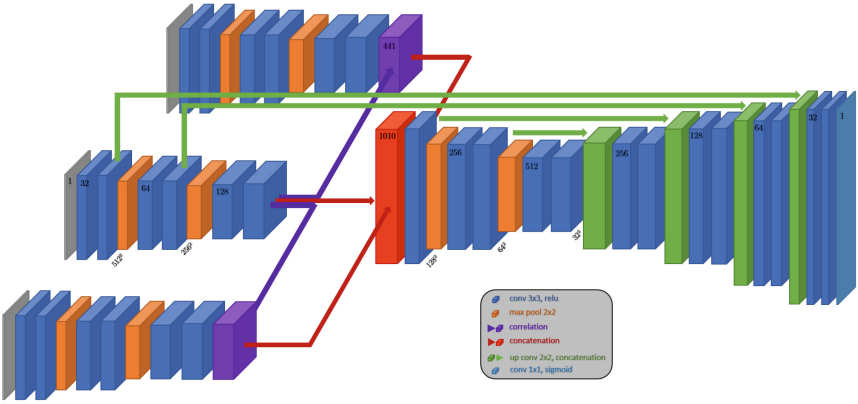


Fig. 2. `Correl-Net` architecture. The encoder is composed of three distinct branches that share the same parameters, one for each input frame (previous, current and next frames). Correlation layers are used to compare the feature maps of the current and previous frame branches, and the current and next frame branches, whose outputs are concatenated with the feature map of the current frame. The decoder takes the form of the classical `U-Net` architecture [19].

4 Experiments

4.1 Datasets

To the best of our knowledge, there is no publicly available dataset of old films annotated with defect segmentation masks. As we explained in the introduction, a restorer's expertise is a key aspect of the restoration process, which means that there can be no real ground truth in defect segmentation that would result from a consensus among restorers. This is the reason why we decided to generate artificial defects with characteristics close to those of real defects, i.e. with random sizes, shapes, colors and transparency.

All networks were trained on the DAVIS dataset, using the *trainVal* data as training set, the *test-dev 2019* dataset as validation set, and the *test-challenge 2019* dataset as test set. The defects were generated on the fly during training, using fractal noise, as described in Fig. 3. The fractal noise allows, after several treatments, to produce blotches of different sizes whose shape is very similar to the real defects.

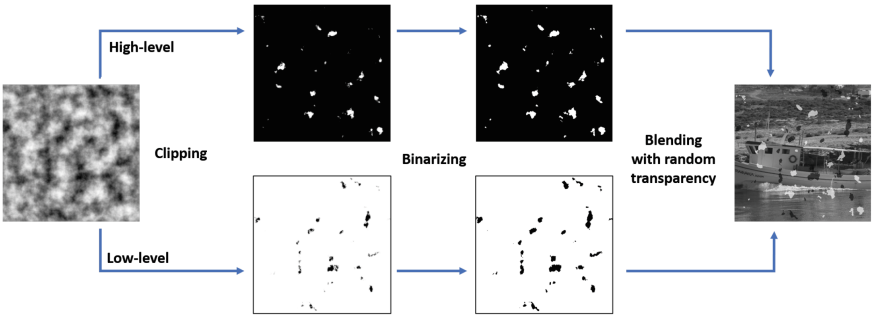


Fig. 3. Artificial defect generation. From left to right: fractal noise generation; clipping and normalization to gather the darkest and lightest defects; thresholding, then merging with random transparency.

At each learning step, we split the grayscale frames into patches of 256×256 pixels. Then, for each patch, we randomly generate a fractal noise with the following parameters: shape (256,256), number of periods (8,8), 5 octaves and a lacunarity of 2. Thus, we obtain a matrix of values between –1 and 1. From this matrix, we then obtain dark defects by selecting the interval [–0,8; –0,65] as well as light defects with the interval [0,65; 0,8]. Since real defects are usually uniform in color, we apply binarization, before merging the defects to the input image. This process is applied to the three channels of the network in order to obtain three consecutive patches with different artificial defects.

4.2 Results on Synthetic Data

In order to quantitatively evaluate the performance of our network, we first present results on the DAVIS test dataset, to which we have added synthetic defects, as described in Sect. 4.1. Table 1 presents a comparison of our method with several classical segmentation neural networks: a simple `U-Net` [19], and the more advanced architectures `Deeplab v3+` [3] and `NAS-FPN` [5]. All of them have been trained three times and evaluated on the same data as `Efficient Correl-Net`, with the three consecutive input frames clustered as a tensor.

Table 1. Comparison of metrics (in percentages) for the test dataset between four different networks. `Efficient Correl-Net` is better in all of them.

Network	**Metric**		
	F1-score	Recall	Precision
`U-Net` [19]	96.60	99.37	93.99
`Deeplab v3+` [3]	94.77	97.47	92.22
`NAS-FPN` [5]	**99.19**	**99.46**	**98.93**
`Efficient Correl-Net`	**99.54**	**99.66**	**99.41**

Table 1 shows that `Efficient Correl-Net` and `NAS-FPN` achieve better overall performance than `U-Net` and `Deeplab v3+`. It is notable that, unexpectedly, `U-Net` achieves better performance than `Deeplab v3+`, which is mainly due to the design of the latter network. Indeed, the last operation of the `Deeplab v3+` decoder consists in a $\times 4$ resizing (using bilinear interpolation) to recover an output segmentation map of the same size as the initial image, without any additional convolution. As a result, the edges of the detected defects are less precise than with other networks, which is particularly problematic for small defects.

In particular, `Efficient Correl-Net` performs slightly better than `NAS-FPN` with respect to the F1-score. This difference is mainly due to the better precision obtained by `Efficient Correl-Net` (99.41 vs. 98.93). Even though this difference is small, it is of particular interest with respect to the task since, as previously mentioned, restorers are concerned with preserving the original state of the film, which implies eliminating false positives manually. To compare the performances of the networks over the DAVIS test dataset, we also classified them for each scene. The results with the three metrics are presented in Fig. 4. Regarding every metric, `Efficient Correl-Net` is always first with one exception. It is also interesting to see that, in some cases, `NAS-FPN` is even worse than `U-Net` to detect defects correctly.

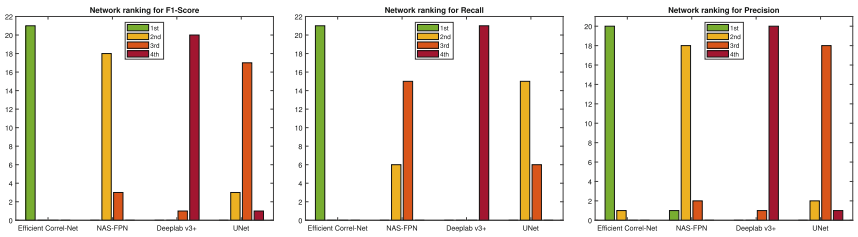


Fig. 4. Rankings of the different networks (from left to right: `Efficient Correl-Net`, `NAS-FPN`, `Deeplab v3+`, `U-Net`) for each scene of the DAVIS test dataset, regarding the metrics (from left to right: F1-Score, Recall, Precision).

In order to visualize the differences of detections of the four networks compared in Fig. 4, Fig. 5 shows the defects detected in different frames of the test set, color-coded to distinguish true positives (green), false positives (blue), and false negatives (red). A common problem shared by all networks is that background features can be detected as defects when they are located in regions that exhibit significant motion (see the example of "rodeo" sequence, in the third row) or occlusions (e.g. "skydiving-jumping" sequence, in the fourth row). Textured objects in scenes also tend to cause poor detections (e.g. the woman's torso in "mermaid" sequence, on the second line). In all of these cases, `Efficient Correl-Net` correlation layer can distinguish real defects from motion and texture artifacts that look like defects.

(a) Input frame. (b) `U-Net`. (c) `Deeplab v3+`. (d) `NAS-FPN`. (e) `Efficient Correl-Net`.

Fig. 5. Visual comparison of defect detections: true positives are highlighted in green, false positives in blue and false negatives in red. From top to bottom, zoom on frames extracted from "luggage", "mermaid", "rodeo" and "skydiving-jumping" sequences. (Color figure online)

4.3 Ablation Study

`Efficient Correl-Net` has two fundamental differences with the segmentation networks compared to it in Table 1: (*i*) the input frames are processed in separate branches of the encoder with `Efficient Correl-Net`, whereas they are stacked and processed in a single branch with `U-Net`, `Deeplab v3+`, and `NAS-FPN`, and (*ii*) the use of correlation layers is specific to `Efficient Correl-Net`.

In order to quantify the impact of these two architectural designs, we defined an intermediate network that we first mentioned in Sect. 4.2 and which is `Efficient Tri-Net`. `Efficient Tri-Net` shares the same overall architecture as `Efficient Correl-Net` (see Fig. 2) with the exception of the correlation layers which are not included in `Efficient Tri-Net`. The feature maps of the three separate branches are simply concatenated, without any correlation computation.

Table 2. Ablation study between three different networks. `Efficient Correl-Net` is better in every metric.

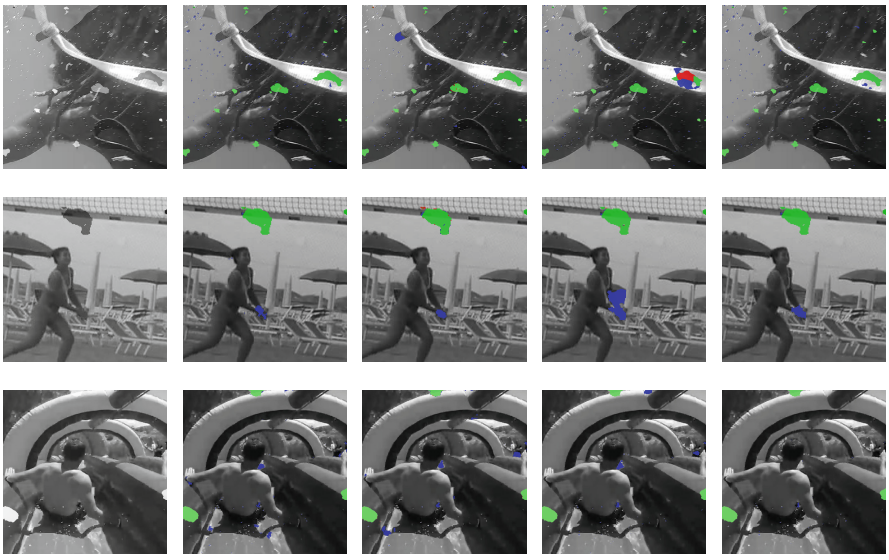
Network	**Metric**		
	F1-score	Recall	Precision
`Efficient U-Net`	99.34	99.64	99.05
`Efficient Tri-Net`	**99.41**	**99.65**	**99.17**
`Efficient Correl-Net`	**99.54**	**99.66**	**99.41**

Table 2 provides a quantitative overview of the impact of these changes on the network architectures. Interestingly, recall remains relatively stable between `Efficient U-Net` (99.64), `Efficient Tri-Net` (99.65) and `Efficient Correl-Net` (99.66). Moving from a single branch (`Efficient U-net`) to three separate branches (`Efficient Tri-Net`) already brings a improvement in precision (99.17 vs. 99.05). This is due to the fact that the comparison between the feature maps of the different frames is performed at a lower resolution, which manages to compensate motions of small objects in the scene in this case.

The use of correlation layers provides an even higher precision (99.41 vs. 99.17) by further compensating motion. The choice of the hyperparameters in the correlation layer has a significant impact on the number of floating point operations (flops) in our network. Indeed, choosing a larger patch size and maximum displacement could potentially achieve higher precision for `Efficient Correl-Net`, but at the cost of a significantly higher number of flops.

4.4 Limitations

Despite the remarkable performance of `Efficient Correl-Net`, some cases of failure remains and are illustrated in Fig. 6. The first line is composed of frames from an underwater sequence, in which the bubbles in the upper left corner are identified as defects by all networks, including `Efficient Correl-Net`. The second line is from a sports sequence in which a large region (player's hands and arms) moves quickly and is detected as a defect. The last line is a complicated case, representing fast movements in a scene with light variations and reflections on water. Many false positives are detected by all networks, including `Efficient Correl-Net`.



(a) Input frame. (b) U-Net. (c) Deeplab v3+. (d) NAS-FPN. (e) Efficient Correl-Net.

Fig. 6. Visual comparison of failing defect detections: true positives are highlighted in green, false positives in blue and false negatives in red. From top to bottom, zoom on frames extracted from "sea-turtle", "volleyball-beach" and "water-slide" sequences. (Color figure online)

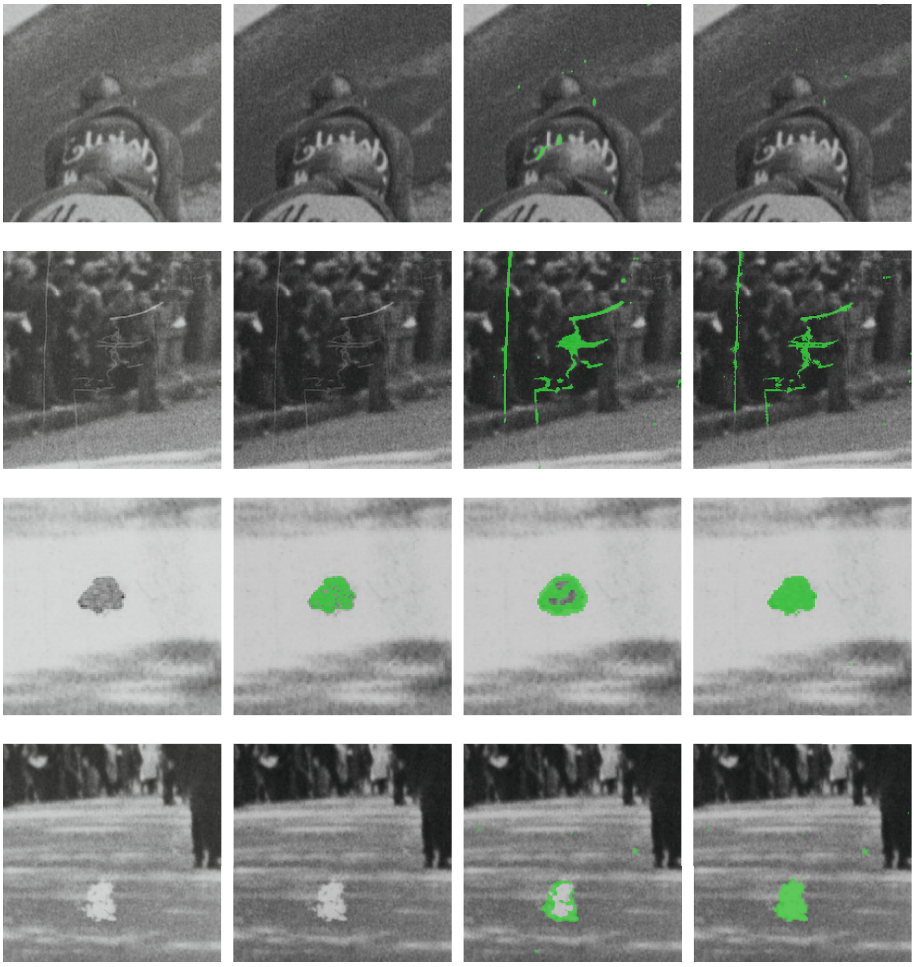
4.5 Fine-Tuning with an Old Film Dataset

In this section we consider a different dataset which is an old film that has been manually restored by an independent expert. We obtain "ground truth" defect masks by comparing the manually restored frame with the original frame. Unfortunately, these defect masks can hardly be considered as a reliable ground truth, as in many cases the restoration process altered pixels that are not in actual defects but in neighboring regions. We can nevertheless use this data for qualitative analysis of `Efficient Correl-Net` performances.

Our self-supervised pre-training on video data augmented with synthetic defects allows `Efficient Correl-Net` to learn well what a temporal anomaly is. However, when the same network is used to detect defects in old film, the results are not satisfactory since the frames are very different from those with synthetic defects (see Fig. 7b). The recall is very low, which is probably due to the following reasons: the image grain is different from the DAVIS dataset, the color and shape of the defects are distributed differently from the synthetic defects.

This old film dataset contains 3000 images, which we divided into training (80 % of the frames), validation (10 %) and test (10 %) sets. When training `Efficient Correl-Net` on these frames only (without self-supervised pre-training), the recall improves again, but the precision remains very low (see

Fig. 7c), which is due to the poor quality of the ground truth masks. On the other hand, to adapt it to a new film, we propose to refine a `Efficient Correl-Net` that has been pre-trained on synthetic data. Figure 7d shows the results we get after a single epoch of fine-tuning on this new dataset, with a learning rate of 10^{-6}. We get far fewer false positives compared to training directly on the film.



(a) Central frame of a group of three defective frames.

(b) Learning only on synthetic data.

(c) Learning only on the old film.

(d) Fine-tuning on the old film after having learnt on synthetic data.

Fig. 7. Results when fine-tuning on an old film using `Efficient Correl-Net`: predictions are applied to the central frame (7a). While the prediction after pre-training on synthetic data fails to achieve a good recall (7b), many false positives or missing detections are avoided when fine-tuning on the new dataset (7d) compared to simply training on this new dataset (7c).

The different examples in Fig. 7 show the improvements due to our two-step learning. For instance, the brand on the back of the lead runner's jersey is detected as a defect (first row) simply using the film data, as it is partially hidden by the head movements of the second runner in the adjacent images. In addition, fine-tuning allows for more accurate detection of scratches (second row), where direct learning on the film groups different parts of scratches as if they were a single blotch. Conversely, the last examples show that the direct learning on the film does not manage to detect entirely certain blotches, whether they are dark (third row) or light (fourth row), only a part of the edges, whereas the fine-tuning does.

5 Conclusion

In this paper, we present `Efficient Correl-Net`, a neural network architecture designed to efficiently detect defects in videos. `Efficient Correl-Net` uses correlation layers previously introduced in `FlowNet` [4] to accurately discriminate real defects from artifacts due to camera motion. We show that `Efficient Correl-Net` achieves slightly better results than the `NAS-FPN` segmentation network [5]. Specifically, it achieves higher precision thanks to the correlation layer, which is desirable for a restorer. Although `Efficient Correl-Net` is trained on a large video dataset augmented with synthetic defects in a self-supervised manner, we also show that it can benefit from a fine-tuning process to be more efficient when applied to a new film.

In addition to the results presented in the previous sections, it is important to note that `Efficient Correl-Net` is a much lighter network (75 million parameters) than `NAS-FPN` (485 million parameters), which means that `Efficient Correl-Net` needs less memory and fewer resources to run.

Although `Efficient Correl-Net` still has some limitations, for example with fast moving objects or light reflections, we believe that it is a significant help to the restorer's work and an important step towards efficient semi-automatic restoration of old films. In the near future, we plan to conduct a study with restoration experts to validate the quality of our defect detection on a wider variety of films, as well as to compare our defect detection to commercial tools used by restorers. We want to evaluate whether switching to `Efficient Correl-Net` would indeed help restorers, for example by decreasing the amount of manual editing required to correct detected defects.

References

1. Biemond, J., van Roosmalen, P.M.B., Lagendijk, R.L.: Improved blotch detection by postprocessing. In: Proceedings of ICASSP, vol. 6, pp. 3101–3104 (1999)
2. Busche, A.: Just another form of ideology? Ethical and methodological principles in film restoration. Moving Image **6**(2), 1–29 (2006)

3. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 833–851. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_49
4. Dosovitskiy, A., et al.: FlowNet: learning optical flow with convolutional networks. In: Proceedings of ICCV, pp. 2758–2766 (2015)
5. Ghiasi, G., Lin, T.Y., Le, Q.V.: NAS-FPN: learning scalable feature pyramid architecture for object detection. In: Proceedings of CVPR, pp. 7036–7045 (2019)
6. Iizuka, S., Simo-Serra, E.: Deepremaster: temporal source-reference attention networks for comprehensive video enhancement. ACM Trans. Graph. **38**(6), 1–13 (2019)
7. Joyeux, L., Buisson, O., Besserer, B., Boukir, S.: Detection and removal of line scratches in motion picture films. In: Proceedings of CVPR, vol. 1, pp. 548–553 (1999)
8. Kim, K.T., Kim, E.Y.: automatic film line scratch removal system based on spatial information. In: Proceedings of ISCE, pp. 1–5 (2007)
9. Kokaram, A.C.: Detection and removal of line scratches in degraded motion picture sequences. In: Proceedings of ESPC, pp. 1–4 (1996)
10. Kokaram, A.C., Morris, R.D., Fitzgerald, W.J., Rayner, P.J.: Detection of missing data in image sequences. IEEE TIP **4**(11), 1496–1508 (1995)
11. Kokaram, A.C., Rayner, P.J.: System for the removal of impulsive noise in image sequences. In: Proceedings of VCIP, vol. 1818, pp. 322–331 (1992)
12. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of CVPR, pp. 3431–3440 (2015)
13. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: fully convolutional neural networks for volumetric medical image segmentation. In: Proceedings of 3DV, pp. 565–571 (2016)
14. Nadenau, M.J., Mitra, S.K.: Blotch and scratch detection in image sequences based on rank ordered differences. In: Time-Varying Image Processing and Moving Object Recognition, vol. 4, pp. 27–35. Elsevier (1997)
15. Newson, A., Almansa, A., Gousseau, Y., Pérez, P.: Temporal filtering of line scratch detections in degraded films. In: Proceedings of ICIP, pp. 4088–4092 (2013)
16. Newson, A., Pérez, P., Almansa, A., Gousseau, Y.: Adaptive line scratch detection in degraded films. In: Proceedings of CVMP, pp. 66–74 (2012)
17. Philippot, P.: Historic preservation: philosophy, criteria, guidelines. In: Preservation and Conservation: Principles and Practices. Proceedings of the North American Regional Conference, pp. 367–382 (1976)
18. Renaudeau, A., Lauze, F., Pierre, F., Aujol, J.F., Durou, J.D.: Alternate structural-textural video inpainting for spot defects correction in movies. In: Proceedings of SSVM, pp. 104–116 (2019)
19. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Proceedings of MICCAI, pp. 234–241 (2015)
20. Shih, T.K., Lin, L.H., Lee, W.: Detection and removal of long scratch lines in aged films. In: Proceedings of ICME, pp. 477–480 (2006)
21. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
22. Sizyakin, R., Gapon, N., Shraifel, I., Tokareva, S., Bezuglov, D.: Defect detection on videos using neural network. In: Proceedings of the International Scientific-Technical Conference "Dynamic of Technical Systems". MATEC Web of Conferences, vol. 132, p. 05014 (2017)

23. Sizyakin, R., Voronin, V., Gapon, N., Pismenskova, M., Nadykto, A.: A blotch detection method for archive video restoration using a neural network. In: Proceedings of ICMV, vol. 11041, p. 110410W (2019)
24. Storey, R.: Electronic detection and concealment of film dirt. SMPTE J. **94**(6), 642–647 (1985)
25. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of ICML, pp. 6105–6114 (2019)
26. Usai, P.C.: Silent cinema: a guide to study. Bloomsbury Publishing, Research and Curatorship (2019)
27. Wang, X., Mirmehdi, M.: Archive film defect detection and removal: an automatic restoration framework. IEEE TIP **21**(8), 3757–3769 (2012)
28. Xu, R., Li, X., Zhou, B., Loy, C.C.: Deep flow-guided video inpainting. In: Proceedings of CVPR, pp. 3723–3732 (2019)
29. Xu, Z., Wu, H.R., Yu, X., Qiu, B.: Features-based spatial and temporal blotch detection for archive video restoration. J. Sig. Process. Syst. **81**(2), 213–226 (2015)
30. Yous, H., Serir, A.: Blotch detection in archived video based on regions matching. In: Proceedings of ISIVC, pp. 379–383 (2016)
31. Yous, H., Serir, A., Yous, S.: CNN-based method for blotches and scratches detection in archived videos. J. VCIR **59**, 486–500 (2019)