UNIVERSITÀ
DI TRENTO

Department of Information Engineering and Computer Science

Bachelor's Degree in Computer Science

Final Dissertation

Detection and Localization of Deep Video Inpainting

Supervisor	Co-Supervisor	Student
Giulia Boato	Andrea Montibeller	Letizia Girardi

Academic year 2021/2022

I would like to express my gratitude to my supervisor, Mr.s Giulia Boato, who gave me the possibility to carry out this traineeship.

To my co-supervisor, Mr. Andrea Montibeller, for helping me during this long phase of traineeship and pushing me to improve myself.

To my family, for having always loved, supported and believed in me.

To my friends Martina, Anna, Daria, Elisa for never leaving and always encouraging me.

To my parents, Massimiliano and Serena, who are the reason for what I have become.

To my sister Nicole for helping me when things get a bit discouraging.

To my loved Paolo, who has always been close to me on this long journey, helping me, encouraging me and showing me how much he cares about me.

And, finally, to me and all my sacrifices made. After all, as dad always says
No ambitious goal can be reached without taking long, uphill roads

Contents

Summary

This dissertation describes in detail the activity performed during my thesis project at the University of Trento - Department of Information Engineering and Computer Science, which was supervised by Mrs. Giulia Boato and Andrea Montibeller, members of the MMLAB research group.

This project was created to obstruct the deep-fake phenomenon and consists in the analysis, detection, and localization of inpainting over a manipulated video. The purpose of this thesis project was to make possible asserting whether a video has been manipulated and which type of deep inpainting technique has been used during the falsification of the analyzed video. The available knowledge in the detection of deep video inpainting fields is still at an early stage. That is why our study tries to tackle this problem by using a well-known method employing a fully convolutional network based on high-pass filtered image residuals, High-Pass Fully Convolutional Network (HPFCN) [6] and formulating the problem as a deep learning task.

The intuition of our work is that by detecting the inpainting technique used during the falsification process, we can make the localization task more efficient by addressing the learning process of the HPFCN network on that specific technique. We wanted to recognize the inpainting technique used during the tampering process and, for this purpose, we implemented a statistical classifier. To localize the deleted objects, instead, we trained the network over our dataset on the previously detected inpainting technique.

Since the video inpainting problem is still at the beginning of its research, the datasets available are few and so, for this project, we preferred to build a new dataset composed of some of the most recent inpainting techniques. The selected networks are Generative Multi-column Convolutional Neural Networks [11], Onion-Peel Network [5], and Spatial-Temporal Transformations Network [13].

Once the dataset has been generated, we analyzed the inpainting techniques trying to define learning common features which allow a plausible localization of the removed object. In particular, we focused on OPN network since it is the combination of two different networks: OPN and Temporary Consistency Network (TCN). We formulated the localization task as a deep learning problem so that we could use HPFCN to localize the removed object and rank the three techniques in terms of better localization capability. Briefly, we observed that HPFCN turned out to be a good choice for the localization tasks in deep video inpainting, except in presence of post-processed video frames where it cannot extract the best features in the learning process. Thus it struggles in generalizing them during the recognition of the removed object since when TCN removes flickering, it consequently leaves noise in the image decreasing the localization accuracy. Specifically, if we train the HPFCN on data manipulated with OPN and post-processed with TCN, the network is more robust in localizing a post-processed video frame and therefore more accurate location maps can be achieved. In the figure below we can see the behavior of HPFCN network in localizing an airplane in case of manipulation of OPN. The Figure 1 illustrates a perfect recognition of the removed object obtained by training the network on data manipulated with OPN and testing it on data manipulated wit OPN; the Figure 2, instead, shows how TCN worsen the localization. In this case we have trained the network on data manipulated with OPN and post-processed with TCN but tested it on data manipulated with OPN. The Figure 3 shows how a correct training and testing of the network can increase the performance of it. Indeed, in this case we trained and tested the nertwork on data manipulated with OPN and post-processed with TCN.

Figure 1: Qualitative visualization of localization maps obtained by training and testing HPFCN on data manipulated with OPN with and without post-processing with TCN.

Since the network HPFCN is not able to generalize when dealing with video post-processed with TCN, we proposed some solutions: firstly we trained the network on a higher number of epochs; secondly we designed new residuals by isolating the high frequencies by applying a circular mask of radii equal to 50 pixels to the Fast Fourier Domain of each video frame and computing the inverse Discrete Fourier Transform. We can see the designed residuals in **Figure 2**.

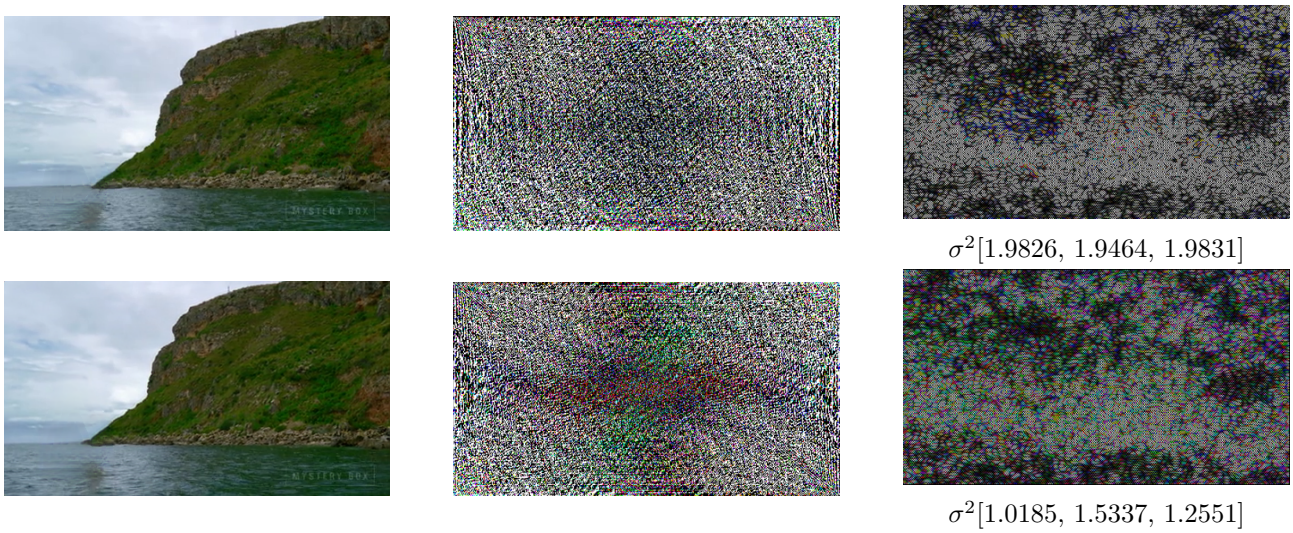


Figure 2: Original video frame manipulated with OPN, FFT transform on it and the obtained new residual (first colum). Original video frame manipulated with OPN and post-processed with TCN, FFT transform on it and the obtained new residual (first colum)

Hence, by studying each video in the Fast Fourier Domain, we computed a valid score for performing a preliminary classification of post-processed and not post-processed frames. Precisely, we noticed that the variances computed with the Fast Fourier Transform on the three channels RGB of post-processed frames are extremely different from each other while, in the case of the not-processed frames, the variances are imperceptibly different. We can notice this by looking at the **Figure 2** . If we consider the last column we can observe that the variances of the second frame are extremely different from each other while, in the case of the first one, the variances are imperceptibly different.

At this point, the focus of the research shift to the detection problem. We have implemented a statistical classifier capable of determining the inpainting technique used in manipulating the video and, in the case of the inpainting with OPN, whether it has been post-processed with TCN or not. In order to complete the research, we re-designed the classifier so that it could distinguish a pristine video from a manipulated one. Firstly, we computed a score for recognizing a which technique has been used during the tampering process. The pristine-score has been computed considering the localization maps of each video frames. It consists of the difference between the medium value of localization map pixels with value greater or equal than 0.5 and the medium value of pixels with value lower than 0.5. By computing the score on each video of our validation data-set we modelled three statistical distributions, each one representing the probability of classifying a video as inpainted with the respective technique. We can observe the three statistical distribution in **Figure 3**.

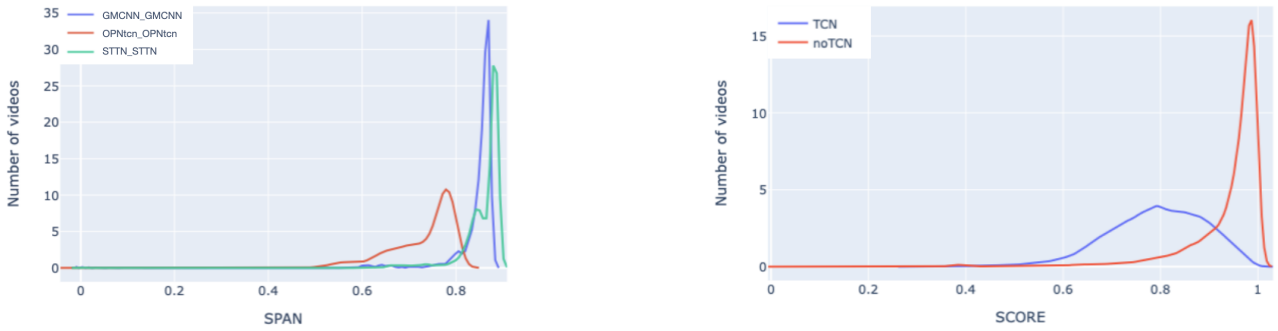


Figure 3: On the left the probability distribution of representing a video as the respective inpainting technique. On the right TCN and noTCN scores distributions

Secondly, we performed a preliminary classification of post-processed frames from not post-processed ones looking at the localization maps of each video frame and, accordingly, defining a score s

$$s = \frac{\min(\sigma^2(\mathbf{I}))}{\max(\sigma^2(\mathbf{I}))}$$

By testing it on video frames belonging to our validation data-set, we are able to represent the TCN-score in two different distributions, one for video frames post-processed with TCN and one for video frames not post-processed. The classifier will compare the TCN-score with the threshold tau_{TCN} returned by the respectively ROC and those frames which score is lower than the threshold, are evaluated as "post-processed".

Finally, we updated the classifier in order to distinguish a pristine video from a forged one. For this reason we evaluated a third score by looking at each localization map of each video frame. We have considered a pristine video frame as a frame where any pixel has not been forged by inpainting techniques and so, where the localization map is untouched, any ghost regions should be recognized. Thus, if the average of pixels coming from the localization map with a value grater than 0.5 is lower than a computed threshold tau_{prist}, the video is labeled as "pristine".

By implementing a majority voting criterion by looking at each score evaluated on each frame, the classifier is able to recognize a manipulated video and detect the inpainting technique used. The confusion matrix, obtained on double compressed inpainted frames in **Figure 4** as proof of the accuracy of our detector.

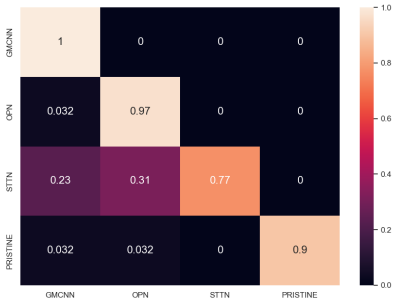


Figure 4: Confusion matrix obtained on double compressed inpainted frames. When inpainted with OPN, the frames are post processed with TCN; while with GMCNN and STTN the frames are not post processed

To summarize, the proposed framework is capable of recognizing a manipulated video from a pristine

one and, after recognizing the inpainting technique used for the inpainting of the content, it is able to localize the removed object by addressing the learning process of HPFCN network on that specific technique. Despite our proposed framework performs well both in detection and localization, more sophisticated methods need to be developed, for instance, deep-learning based.

1 Introduction

In recent years the diffusion of many digital video and image editing software has made the accurate authentication of multimedia content challenging. The current manipulation technique made it possible even for a novice to easily manipulate each type of media, starting from an image, altering, for instance, their colors and brightness, or even an entire video sequence, deleting or adding objects.

With the technological progress the popularization of digitization and informatization has increased. Internet, also due to the significant diffusion of social media, became the most important information source where people can publish their digital media at any time. These media convey important information to people but ,due to the diffusion of digital video and digital image editing software, more and more people modify their contents before posting them, distorting the reality. In the last few years, fake multimedia has become a central problem for society. Through the usage of easy-to-use and powerful image/video editing tools, such as Photoshop or Adobe After Effects, more and more contents have been manipulated. [10] Harmlessly, most of the time people modify their body imperfections (e.g passage of time or acne), or the main features of an image, such as resizing, rotation or color adjustment in order to improve the visual appearance of the media. But, unfortunately, such type of processing can also be used to manipulate a photograph, or a video, for illegal purposes. Among the most common illegal activities we can underline the deletion of objects in an image to fabricate a fake scene, for instance erasing visible copyright watermarks, creating fake porn videos to blackmail people or building fake-news campaigns to manipulate the public opinion.
Research in multimedia forensics has been going on for at least 15 years and nowadays they are increasingly required. Internet, in fact, has became, among the rest, a way of disseminating fake news. It favors the dissemination of information from any source (eg unreliable websites, fake profiles), that allows anyone to publish and share information, even false ones, which through these channels can spread sometimes to the point of becoming viral. **Figure 1.1** shows some popular deepfakes circulating on the internet.

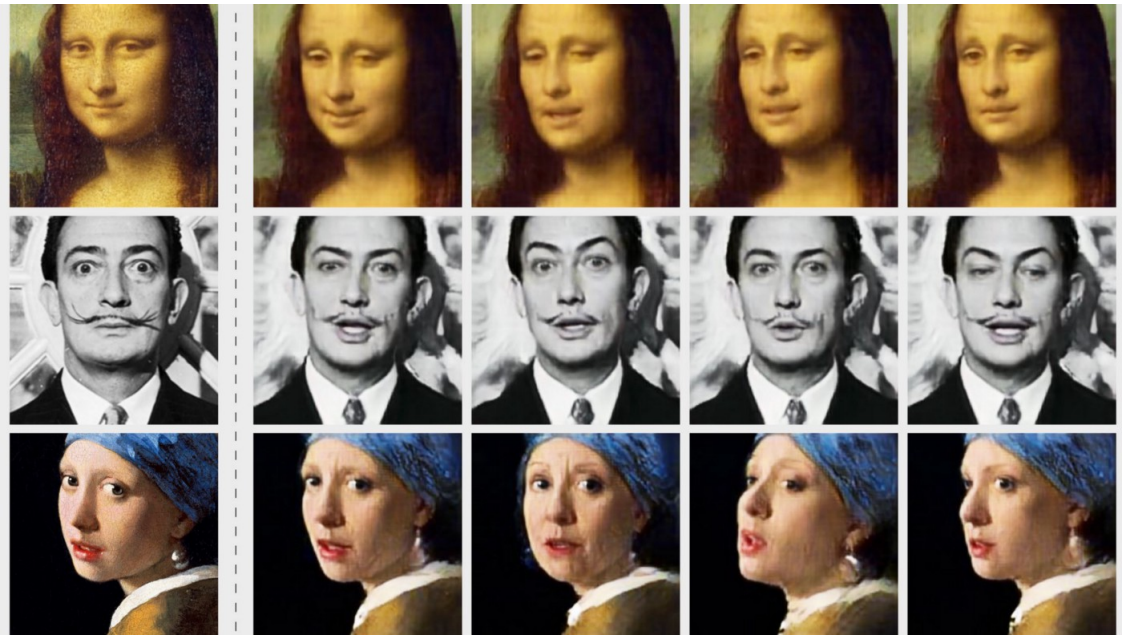


Figure 1.1: Examples of deepfake

There are many ways to manipulate visual content. Specifically, in this thesis project we will study deep video inpainting. Inpainting is a digital processing that allows you to modify an image or in our case, a video, beyond recognition. It is the task of completing an image or in our case, a video, that has empty pixels by filling the corrupted regions with visually plausible pixels. The reconstruction of missing or corrupted regions was already used in the past to restore works of art, then it was extended to photographs to eliminate, for example, defects due to the passage of time in order to improve their appearance.

In detail, video inpainting aims to fill spatio-temporal holes with plausible content in a video. This technique can help numerous video editing and restoration tasks such as undesired object removal, scratch and damage restoration. Nowadays video inpainting is widely used in combination with Augmented Reality (AR) for a greater visual experience. Removing existing items permits the users to focus their attention on specific elements in the scene. Moreover, there are several semi-online streaming scenarios such as automatic content filtering which use this type of post-processed technique.
Despite tremendous progress on deep learning-based inpainting of a single image, in the last five years many deep learning-based video inpainting techniques have been studied, but they all remains at an early stage. [12] A straightforward way to perform video inpainting is to apply image inpainting on each frame individually of the video. However, the complex motions and the requirement of temporal consistency make the video inpainting a challenging problem. In the field of video completion, in fact, the missing regions have to be filled with content consistent over time: when an object has been removed in a video, the region occluded by that object may be visible not in the same frame but in the following ones. Therefore, filling the corrupted region without considering the original content in the other frames will break the temporal coherence.[5]

For these reasons, it was developed an algorithm for obstructing the deepfake phenomenon by detecting and localizing inpainting over a manipulated video. Nevertheless, since the state of the art on detection and localization of deep video inpainting is at an early stage we tried to tackle this problem by designing and implementing a statistical classifier capable of recognize the technique used during the completion process and then, by using the deep image inpainting technique High Pass Fully Convolutional Network (HPFCN) [6] formulating the problem as a deep learning task, the localization of the deleted object has been addressed. To do so, three different inpainting techniques have been used for completion task: Generative Multi-column Convolutional Neural Networks [11] (GMCNN), Onion-Peel Network [5] (OPN) and Spatial-Temporal Transformations Network [13] (STTN) and a fully convolutional network based on high-pass filtered image residuals has been employed to locate the regions manipulated by deep inpainting.
This project will allow us to reduce the many current cyber criminal that use deep-fake by enabling the distinction of a true media from a tampered one. And also by restoring the real content of the media.
The rest of this dissertation is organized as follows. Chapter 2 briefly introduces the deep video inpainting state-of-the-art by presenting the most recent deep inpainting forensic methods. Chapter 3 describes in details the custom-made dataset by presenting the related inpainting techniques. Chapter 4 describes how localization and detection has been achieved. Chapter 5 reports the experimental results we developed and, finally, the concluding remarks are drawn in Chapter 6.

2 State-of-the-art in video inpainting

This chapter present the most recent video inpainted detection and localization algorithms and the most recent techniques in video inpainting.

2.1 Approaches to video inpainting

With the advance of recent image inpainting approaches, more recent studies have been conducted in the video inpainting field. So far, two different types of approaches of video inpainting problems have been studied: patch-based and learning-based approach. [12] The conventional video inpainting techniques usually modify the video in order to fill missing regions of the various frames in a temporal-extended form of the patch-based image inpainting. During this first phase of research, the main focus is on two conventional approaches: the first was based on the temporal extension of patch-based image inpainting and the second on spatial-temporal optimization-based video inpainting. Despite some positive results, the temporal extension of patch-based image inpainting suffers from severe flickering and ghosting artifacts, while the spatial-temporal optimization-based video inpainting approach cannot handle videos with complex motions, complex structures, or objects that do not appear in the given videos. To tackle the weaknesses of the conventional video inpainting approaches, there have been introduced several deep video inpainting approaches yielding a great improvement in video inpainting. By using 3D temporal convolution, optical flow, and attention mechanisms, existing state-of-the-art video inpainting methods can obtain useful information about the temporal aspect of the video from neighboring frames and produce contents that are spatial-temporally consistent. [12] These existing deep video inpainting methods can be summed up as two principal modules: a temporal feature aggregation and single-frame inpainting for achieving temporal consistency. [3] The first one aims at finding corresponding valid pixels in neighboring frames within a temporal radius for missing or removed regions through context matching; the second one creates visually plausible frames in a spatial-temporally consistent way, which learns the information from previous frames and the current one. From there, the deep generative models for video inpainting cannot only show impressive restoration results on complex scenes, but also produce novel objects temporally consistent. Moreover, with the deep video inpainting approaches, we can recover targeted object regions with photo-realistic contents and preserver temporal consistency.[12]

2.2 The problem of spatial and temporal coherence in video completion

Even in our modern days the problem about spatial and temporal coherence in video completion is still being studied . When we need to fill missing regions of a manipulated video, we need to guarantee coherent contents through time because the occluded region by the object could be visible in other frames so filling such regions without taking account of the content in the original frame could result in bad temporal consistency. Indeed, the soundness of a video inpainting algorithm depends on the insurance of temporal coherence. Specifically, for our project, this consistency is obtained by introducing a suitable post-processing module that aims to eliminate the resulting flickering. We will take into account an inpainting technique that works as described above. The technique in question [5] is often combined with a post-processing technique [4] whose deep recurrent network structure guarantees an enforcement in temporal consistency in a manipulated video.

Currently, the most recent video inpainting algorithm deep learning based is summarized by Suraj *and al.* [1]. The network is based on the regeneration process using reference frames: a set of reference frames regenerates the holes in a given target frame one layer at the time ensuring coherent contents in the video. Once there is enough information about the generated hole, the network computes the similarities between the pixels in the target and the non-hole pixels in the references and completes the occluded region. The perk of this method is the unlimited spatial-temporal window which guarantees global coherence in the inpainting process.

2.3 Deep video inpainting Localization and Detection state-of-the-art

Unfortunately, the ability to produce realistic videos by removing objects can also be used maliciously. Tampered videos could result in serious legal and social implications including swaying a jury

and spread of misinformation on the internet. Over the years, it has become necessary to authenticate digital videos. In order to verify the trustworthiness of digital videos, video forensics has studied this field for more than a decade by developing many different algorithms with the goal of localizing and detecting the tampered region.

One recent proposed framework to localize the deep inpainting regions, Spatiotemporal Convolution and Refinement Network [3], designs a spatial-temporal convolution-based refinement network to enhance the tampering traces for image forensics. By the concatenation of four ResNet blocks, and two up-sampling layers, a detection module has been designed in order to learn recognizable features and achieve a rough location map. Hence, the pixel-wise localization map is obtained by the development of a refinement module, which effectively localizes the inpainted regions.
Additionally, the recent end-to-end framework to locate the inpainted regions in a manipulated video presented in "Depp Video in-painting localization using Spatial and Temporal Traces" paper [12], relies on the spatial-temporal traces left by the inpainting. In Figure 2.1 we can see the spatial and temporal inconsistencies caused by extracting intra-frame and inter-frame residuals guided by optical flow. Hence, the two different residuals are fused and encoded into deep discriminative features by designing a dual-stream network. Finally, in order to produce satisfactory pixel-wise localization results, where the temporal correlation among sequential frames is guaranteed, bidirectional convolutional LSTMs (Bi-ConvLSTMs) has been embedded into the decoder network.

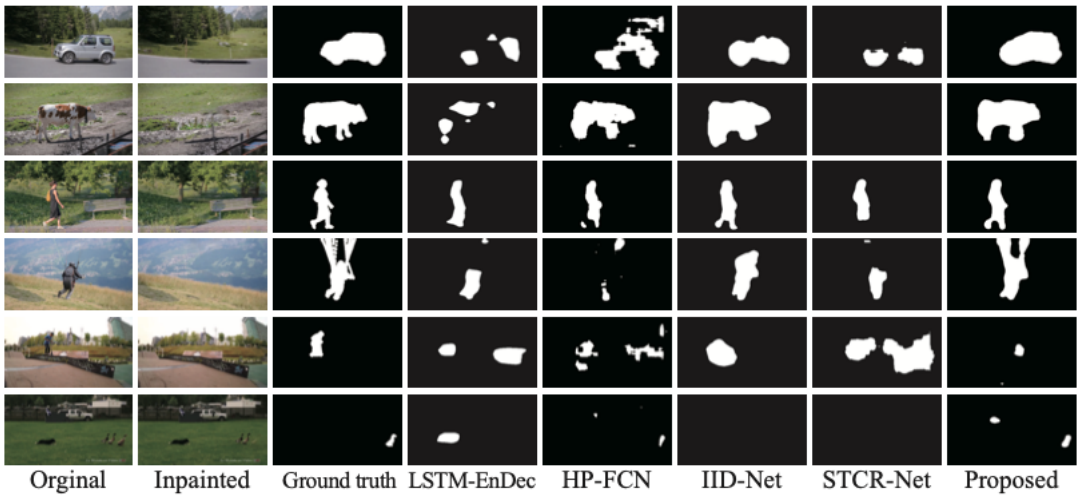


Figure 2.1: Examples of localization results. The proposed method is referred to [12]

Although there are satisfactory studies on localizing tampered regions both in manipulated images and videos, very limited effort has been devoted to video inpainting detection. Actually, most of them are designed specifically for deep image inpainting detection.
More recently, most of the approaches rely on the application and re-elaboration version of image inpainting detection methods to deep video inpainting problems. One example is the network presented by Zhou *and al.* [14] who proposed a framework whose goal is the detection of an inpainted region based on an encoder-decoder architecture named VIDNet, Video inpainting Detection Network. The intuition performed in the creation of the network is the fact that inpainting methods borrow information from neighboring pixels of the region to be inpainted. Hence, a multi-head local attention module that uses adjacent pixels to discover inpainting traces has been designed. The final prediction has been evaluated by modeling the temporal relations among different frames with a ConvLSTM.

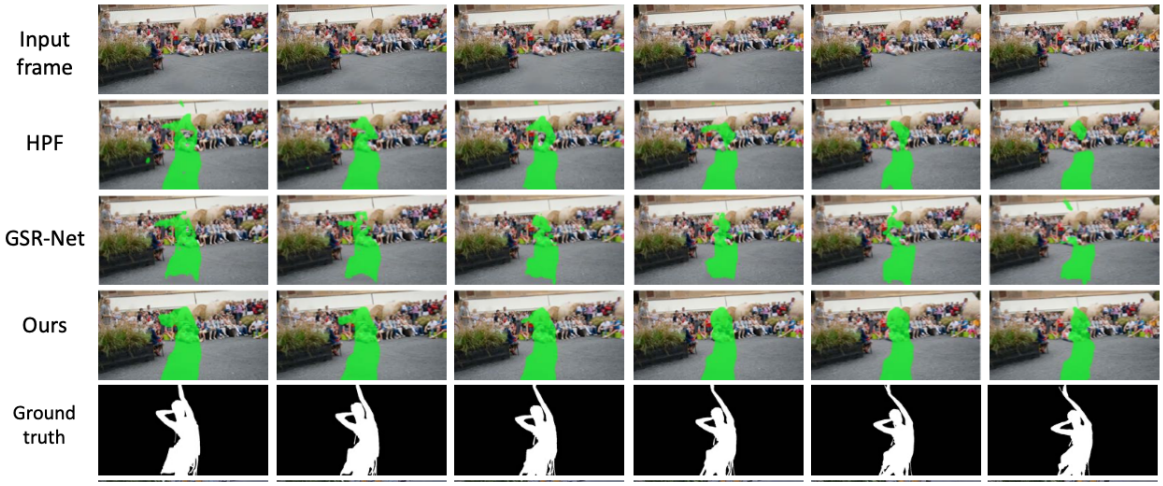


Figure 2.2: Qualitative visualization. The first row shows the inpainted video frame. The second to fourth row indicates the final predictions from different methods. Ours is referring to the framework [14]. The fifth row is the ground truth.

In this project, we approached to localization task by educating the deep image inpainting technique High Pass Fully Convolutional Network (HPFCN) because, since the state-of-the-art of image-inpainting is rich, the application of image inpainting techniques on each frame individually of the video allows us to have access to a considerable amount of information, which is fundamental for this kind of research. Moreover, we are interested in HPFCN as it is a valid network that many networks are confronted with and as starting point to make it more robust in terms of deep video inpainting detection and localization.

3 Dataset

In this chapter we present the dataset used during the conducted experiments to detect and localize the inpainting over the manipulated videos.

3.0.1 Dataset structure

In order to get as accurate as possible results from the testing of the network HPFCN, we need to train it on a large and diverse training data-set.

Since the video inpainting problem is still at the beginning of its research, the data-sets available are few. For this reason, we preferred to build a new data-set composed of some of the most recent inpainting techniques. The selected networks are:

- Generative Multi-column Convolutional Neural Network (GMCNN) [11];
- Spatial-Temporal Transformer Network (STTN) [13];
- Onion Peel Network (OPN) [5].

The data-set created counts 312 pristine videos and 3744 inpainted videos whose resolution is equal to 432×240. The gathered data are divided as follows:

Branch	# videos
pristine	312
double compression	624
double compression + TCN	312
TOT.	936

Table 3.1: Numerosity of the created dataset

The pristine videos have been collected from Youtube and other datasets such as Vision, Socrates, Youtube-8m and Youtube-vos. Each video contains unique objects and we are interested in being able to learn some common features from these in order to detect a particular object removed from the scene. The inpainted frames generated with the inpainting techniques GMCNN, OPN, and STTN, have been initially stored as single compression, and therefore stored as double compressed frames. Double compression frames, as their name implies, are obtained by two compressions of the original video; the first consists of the transition from video in frames which are inpainted and eventually, in case of the manipulation with OPN, post-processed with TCN. At this point, the frames are encoded back to the video. Single compression frames, instead, consist of just one compression: frames directly extracted from original videos are stored without being encoded.

Generative Multi-column Convolutional Neural Network (GMCNN)

Generative Multi-column Convolutional Neural Network (GMCNN) [[11]] is an image deep inpainting technique based on a multi-column structure. Despite its principal application, we have chosen this technique to show how its application can work in an appreciable way even for video inpainting problems. The network is composed of three parallel encoder-decoder which directly aim at modeling different image components and decomposing them into different multi-level features. Given a set of images and the co-respective masks, GMCNN extracts different level features from the original resolution and concatenates them into a feature map which, finally, is decoded back to an image. GMCNN can work with random size inpainting areas and images. However, this method still has difficulties dealing with large-scale data-sets with thousands of different object and scene categories.

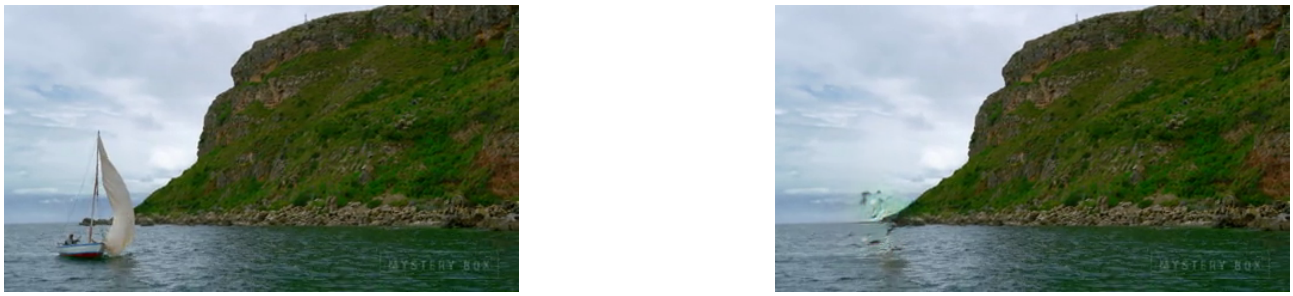


Figure 3.1: Qualitative visualization of inpainting with GMCNN. On the left the original video frame. On the right the inpainted video frame.

Spatial-Temporal Transformer Network (STTN)

Spatial-Temporal Transformer Network (STTN) [13] is a video deep inpainting technique, which takes both neighboring and distant frames as input and simultaneously fills missing regions in all input frames, formulating the video inpainting problem as a multi-to-multi task. The intuition is that an occluded region in a current frame would probably be revealed in a region from a distant frame. To simultaneously complete all the input frames in a single feed-forward process, the transformer needs to search for coherent contents from all the frames along both spatial and temporal dimensions. The multi-scale patches extracted allow us to overcome the problem of complex motions in a video: by

comparing the patches extract from the input frames and aggregating the most relevant ones, we can complete the corrupted region in the best manner.


Figure 3.2: Qualitative visualization of inpainting with STTN. On the left the original video frame. On the right the inpainted video frame.

Onion Peel Network (OPN)

Onion-Peel Network (OPN) [5] is the second video deep inpainting technique present in our data-set. This technique works differently from the two proposed: using a set of sampled frames as the reference, the network fills the hole of the target frame by looking at the contents of reference frames for the right pixels. The network inpaints the hole region one layer at a time (one "peel" at the time) by progressively eroding the hole from its boundaries towards the center exploiting similarities with the non-hole pixels. By doing like this, OPN can exploit richer contextual information for the missing regions at every step which, if numerous enough, successfully allows the inpainting of scenes with occlusions and large holes. One peculiarity of this technique is the attention to the minimization of visible artifacts on the manipulated area. The inpainting process, indeed, may cause flickering artifacts on the inpainted area, and the proposed technique, in order to remove these artifacts, post-processes the inpainted frame with a Temporal Consistency Network (TCN).

Temporal Consistency Network (TCN)

Temporal Consistency Network (TCN) [4] is an efficient approach for enforcing temporal consistency in a video by minimizing the short-term and long-term temporal loss. During the inpainting process, some flickering artifacts could occur and TCN could be useful. The network takes an original video $\{I_t|t = 1...T\}$ and the corresponding processed video $\{P_t|t = 1...T\}$ as input and generates the respective temporally stable output video $\{O_t|t = 1...T\}$. The approach is formulated as a learning task: both original and compromised videos are transformed into images and after setting the first output frame $O_1 = P_1$, the network generates the output frames sequentially from t = 1 to T. At each step, the network learns to generate an output frame O_t that is temporally consistent with respect to the O_{t-1} frames. Hence, the current output frame consists of the input at the next time step.

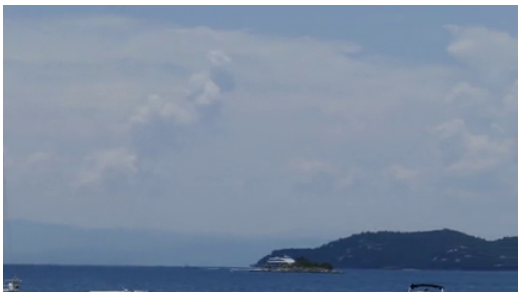



Figure 3.3: Qualitative visualization of inpainting with OPN and post-processed with TCN. On the left the original video frame. On the right the inpainted video frame.

4 Localization and detection of deep inpainting

This chapter describes the studied method for localization and detection of inpainting in a manipulated video. We are going to present the advantages and weaknesses of the proposed method. In concluding, we will introduce some solutions studied in order to improve the effectiveness of our method.

4.1 Localization of deep inpainting

Given a manipulated video, we want to specify where the video has been manipulated. For this task, we educate a common Convolution Neural Network, HPFCN [14], to recognize the corrupted region of video forged by the application of one of the already discussed inpainting techniques.

High-Pass Fully Convolutional Neural Network (HPFCN)

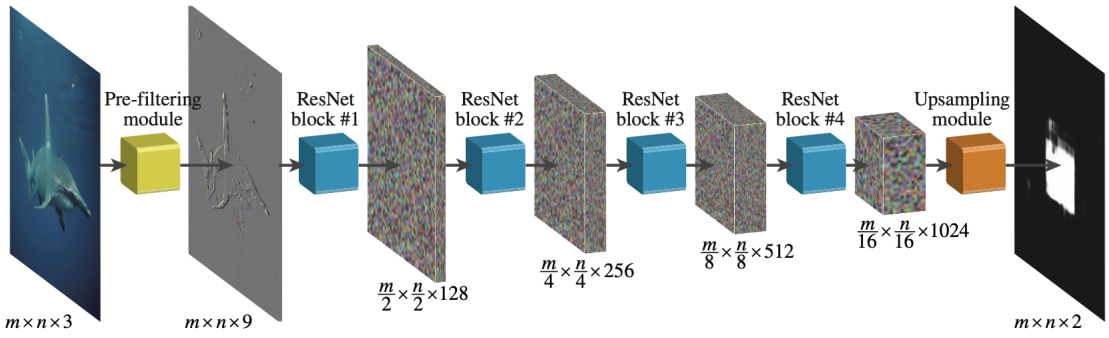


Figure 4.1: HPFCN pipeline [6]

High-Pass Fully Convolutional Neural Network (HPFCN) [14] is a method to locate the regions of an image manipulated by deep inpainting. The proposed method employs a fully convolutional network that is based on high-pass filtered image residuals. In order to correctly implement the network, it is necessary a pre-filtered module, as illustrated in **Figure 4.1**. A common practice for highlighting tampering traces is to perform high-pass filtering on an image. By designing a high-pass pre-filtering module, the network is able to isolate the image residuals which allows for enhancing the traces of the inpainting. Specifically, a RGB image is transformed into a 9-channel image residuals by separately convolving each channel of the input image with a set of high-pass filter kernels and concatenating together the convolution results as the input of the subsequent network layers. Once isolated the traces, the end-to-end network learns discriminative features by concatenating four ResNet blocks which, finally, are enlarged by an up-sampling module, achieving the pixel-wise inpainting localization maps. Nonetheless, the network is a fully convolutional network without fully-connected layers, by its structure, it can work on images with arbitrary sizes. Obviously, the loss function used aims to address the class imbalance between inpainted and un-touched pixels.

Why did we use HPFCN? Firstly, we have to remind that this particular network has been designed for image inpainting problems. The main reason behind this critical choice is the fact that, since a video is a sequence of frames captured with a certain frequency (rate) and the state-of-the-art

of image-inpainting is rich, the application of image inpainting techniques on each frame individually of the video allows us to have access to a considerable amount of information, which is fundamental for this kind of research. Moreover, we are interested in HPFCN as it is a valid network that many networks are confronted with. Finally, we would like to make it more robust in terms of deep video inpainting detection and localization. Obviously, the application of this type of localization method to deep video-inpainting problem has particular limitations. But, with the necessary cautions in the implementation of the algorithm, our goal is to attenuate these limitations.

We approached the localization problem by deeply analyzing the behaviour of the neural network HPFCN dealing with data manipulates with the inpainting techniques GMCNN, OPN and STTN. More details of the conducted experiments are provided in **chapter 5**. Especially, we studied the OPN technique since it is paired with the Temporary Consistency Network TCN. The former is a method for inpainting, and the latter is a post-processed technique whose aim is the generation of a temporally stable output video by limiting the occurrence of flickering artifacts after the manipulation process. Since we are interested in the operating principle of TCN, we tried to apply the post-processed method to each inpainting technique in order to evaluate the High-Pass Fully Convolutional Neural Network in terms of localization of an inpainted and post-processed area. Briefly, in **Figure 5.2**, we observe the poor performance of the network when a post-processed video frame has been given as input to the network. We can infer, from several preliminary results illustrated in **chapter 5**, that the deep neural network HPFCN turned out to be a good choice for the localization tasks in deep video inpainting, except in presence of post-processed video frames where, evidently, is not able to generalize. In order to understand the reason why the network is not able to generalize in the case of post-processed videos, we have profoundly studied the TCN technique.

Possible solutions for the localization problem

We proposed some valid solutions in order to help the neural network to improve its generalization capability and recognize the deleted object also in the case of post-processed videos. Firstly, we trained and tested the network on a higher number of epochs, precisely on 20 epochs, instead of 10 like in the previous experiments. Several results discussed in **chapter 5** evidence that the number of epochs improves the confidence in localization. The second proposed solution provides the learning of the HPFCN network where the input video is re-elaborated in new residuals. Based on the high-pass filtering nature of the network HPFCN, we designed new residuals by isolating the high frequencies filtered of each video frame in Fast Fourier Domain. We noticed that by giving as input these new specific residuals to the network evident artifacts appear and preliminary classification of post-processed and not post-processed frames can be performed. More details concerning this last solution are going to be provided in the next **section 4.1.1**.

4.1.1 Designing new residuals for the localization task

Since the network HPFCN struggle with generalizing when dealing with video post-processed with TCN, we want to understand how the method affects the main features of the video. Based on the nature of our neural network HPFCN, we designed new residuals in order to isolate the high frequencies characterizing the frames of the input video and enhancing the detection and localization of inpainting. Initially, we analyzed the Fast Fourier Transform (FFT) Domain of each frame post-processed with TCN and not post-processed. The Fourier Transform is an important image processing tool that is used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the Fourier or frequency domain, while the input image is the spatial domain equivalent. In the Fourier domain image, each point represents a particular frequency contained in the spatial domain image. Since the input images appear as RGB format frames, we transformed them into the Fourier domain by computing the FFT function on each channel. Hence, we have computed their real part and concatenated them with the goal of obtaining a 3-channel FFT transform. At this point,

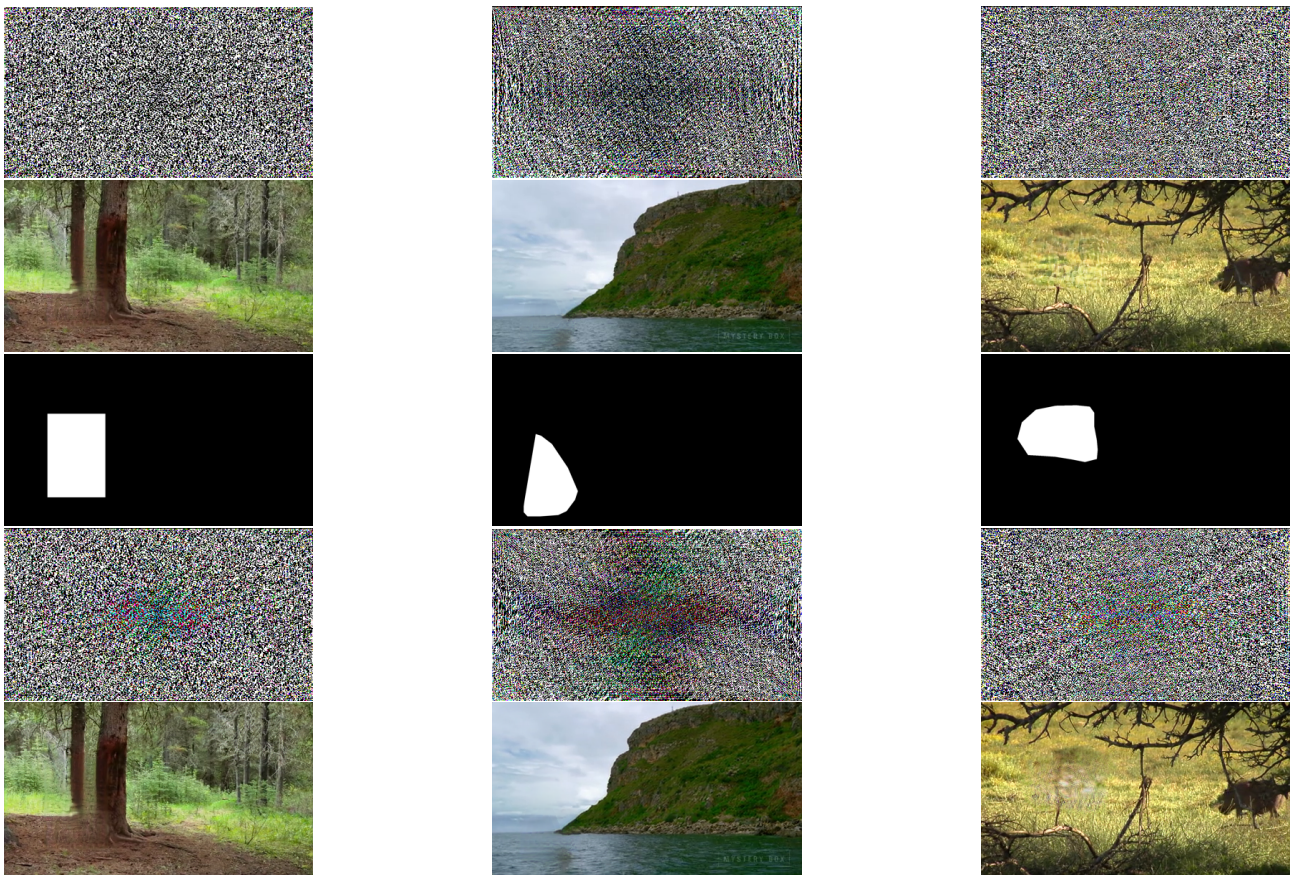
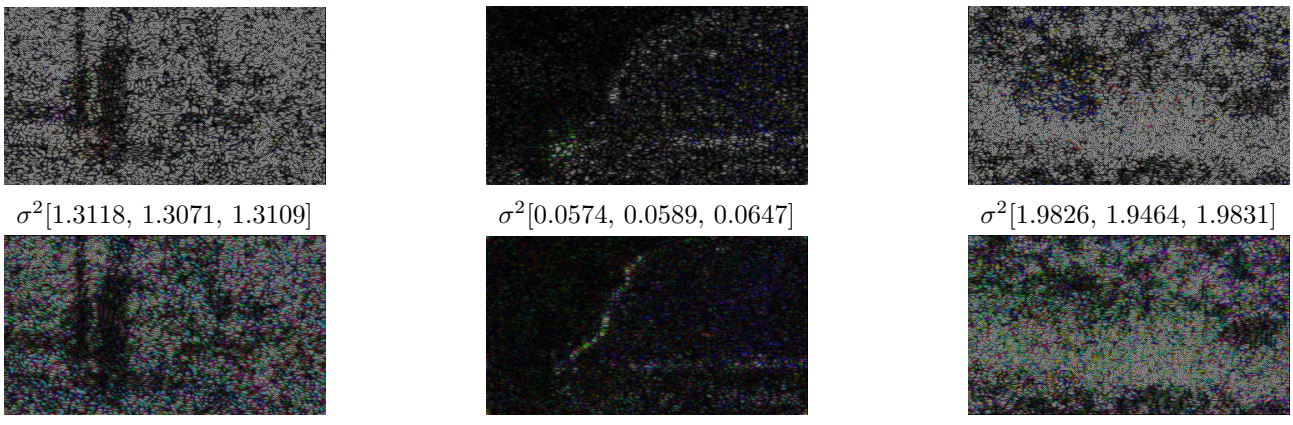


Figure 4.2: FFT transform of double compressed not TCN processed frames (first row) of three frames inpainted not post-processed (second row), ground-truth (third row), FFT transform of double compressed and TCN processed frames (firth row) of three frames inpainted and post-processed(fifth row)

we observed the persistence of green artifacts in the Fourier Domain of post-processed frames and, for this reason, we have generated new residuals starting from this domain. Once correctly isolating the high frequencies at the center of the FFT by applying a circular mask of radii equal to 50 pixels to the Fast Fourier Domain of each video frame, we computed the inverse Discrete Fourier Transform. In **Figure 4.3** we can observe some examples of new residuals obtained by the three post-processed frames in **Figure 4.2**. Additional results concerning the localization of deep inpainting are provided in **chapter 5**.



σ^2[1.3118, 1.3071, 1.3109] σ^2[0.0574, 0.0589, 0.0647] σ^2[1.9826, 1.9464, 1.9831]

σ^2[0.4357, 0.5333, 0.6241] σ^2[0.0318, 0.0330, 0.0428] σ^2[1.0185, 1.5337, 1.2551]

Figure 4.3: New residuals of the three frames inpainted no post-processed proposed in **Figure 5.3** (first row) and post-processed with TCN (second row)

4.2 Detection of deep inpainting

At this point, we have focused our research on deep video inpainting detection. Since the literature on the detection of deep-inpainting in videos is quite poor, we approached this task by implementing a statistical classifier that allows us to determine which technique has been used in the inpainting process. Actually, we are interested in finding the best model to use in the localization problem which ensures better localization maps. The classifier has been gradually implemented: firstly we performed a preliminary detection of manipulated videos by taking into account the inpainting technique used during the manipulation phase, then we improved the performance of the detector by distinguishing a post-processed video frame with respect to a not post-processed one and, in conclusion, we introduced the distinction of pristine videos from manipulated ones.

4.2.1 Detection of inpainting technique

From the localization experiments, we noticed that the HPFCN produces better location maps on the same technique it was trained with. This implies that, if the HPFCN was trained on data manipulated with GMCNN, the inpainted pixels in a location map obtained by testing HPFCN on data manipulated with GMCNN will yield a value greater than the value of the equivalent inpainted pixels obtained by testing HPFCN with STTN or OPN techniques.
According to this intuition, we have built three statistical models which allowed us to construct a distribution function for each technique and determine the used inpainting technique during the inpainting process accordingly. The distribution model with the maximum likelihood w.r.t the input frame has been updated with a score equal to the likelihood value already estimated. Hence, by associating an inpainting technique to each video frame, we can correctly classify the entire video using the majority voting criterion.

4.2.2 Detection of post-processed videos with TCN

The conducted study concerning the designed residuals has turned out to be paramount in performing a preliminary detection of manipulated videos. As we analyzed the engineered residuals, we observed some persistent artifacts in the Fourier Domain of post-processed frames and, furthermore, we noticed that in the case of post-processed frames the variance of each channel differs greatly from one another in terms of numeric value in contrast to the distribution of variances in case of a not post-processed one, where the distributions are approximately the same. If we consider the first column of the **Figure 4.3**, for instance, we can observe that the variances of the second frame are extremely different from each other while, in the case of the first one, the variances are imperceptibly different.
These intuitions enabled us to perform a preliminary classification of post-processed frames in compare with not post-processed ones by defining a score s 4.1.

$$s = \frac{\min(\sigma^2(\mathbf{I}))}{\max(\sigma^2(\mathbf{I}))} \tag{4.1}$$

Plotting a Receiver Operating characteristic Curve (ROC), **Figure 5.10**, by checking both post-processed and not post-processed with TCN, we compute a threshold which is compared with each video frame in order to define whether a frame has been post-processed. In particular, those frames which score is lower than the threshold, are evaluated as “post-processed”.

4.2.3 Detection of pristine videos

At this point, we have designed a statistical classifier capable of determining the inpainting technique used in manipulating the video and, in the case of the inpainting with OPN, whether it has been post-processed with TCN or not. In order to complete the research, we re-designed the classifier so that it could distinguish a pristine video from a manipulated one.
A pristine video is a video where any pixel has not been forged by inpainting techniques. This means that, if we perform the localization map over a pristine video, any ghost regions are recognized and most of the pixels should be untouched. This intuition enabled us to introduce a discriminator for

the video "pristine". Basically, for each element of a video, we calculated the average of pixels with a value greater than 0.5. Those pixels whose value is lower than 0.5 are labeled as "pristine". Hence, by plotting another Receiver Operating characteristic Curve, **Figure 5.11** we obtained the threshold which allows us to distinguish a pristine video from a manipulated one.

In conclusion, the final implemented framework aims to detect whether an input video has been manipulated or not and, if the inpainting is successfully recognized, will detect which inpainting technique has been used among GMCNN, STTN and OPN. Hence, if OPN has been recognized, the classifier verifies whether the video frames have been post-processed with TCN.
Given a test video $\mathbf{v_n}$ and three statistical distributions $\mathbf{\Theta}_1, \mathbf{\Theta}_2$ and $\mathbf{\Theta}_3$ each one representing the probability of classifying a video as inpainted with one of the three techniques GMCNN, OPN and STTN, such that:

$$\mathbf{v_n} = \{\mathbf{I_n}^{(1)}, \mathbf{I_n}^{(2)}, ..., \mathbf{I_n}^{(M)}\} \tag{4.2}$$

where $\mathbf{I_n}^{(1)}$ represents the first frame of the video $\mathbf{v_n}$, we can summarize the classification process as follow:

- counting the number of pixels n in the input frame greater then 0.5 and comparing it with the threshold tau_{prist} returned by 5.8; if the number n is lower then tau_{prist}, the frame is labeled as "pristine": the classifier will evaluate the next video frame. If the number n is greater then tau_{prist}, the frame is labeled as "manipulated";

- computing median and TCN scores for each video frame. The median score is necessary in the detection of the inpainting technique: the distribution model with the maximum likelihood with respect to the input frame determine which technique has been used in the inpainting process. TCN score, instead, has been calculated for the recognition of post-processing: by comparing the threshold value obtained by 5.7 and the TCN score, we can assert whether an input frame has been post-processed with TCN;

- determining the inpainting technique used for the manipulation of the input video. Once each frame of the video has been evaluated, the classifier, based on the majority voting criterion, recognizes the technique used for the manipulation of the entire frame sequence.

5 Experimental results

This chapter gives a qualitative evaluation of the several inpainting methods previously introduced. Extensive experiments allow an evaluation of the effectiveness of the networks and the classification of them based on qualitative and quantitative results. In the end, several outcomes of detection task are presented.

5.1 Localization: preliminary results

The proposed experiments involve the analysis of a manipulated video consisting of an airplane take-off. The name of the test video is "airplane4" and we are going to analyze the prediction maps obtained by the different cases of training and testing of the network HPFCN. Precisely, the network has been trained on different data in order to obtain two different models and learn in both cases the best features for the localization problem. The discriminating factor between the two models is the post-processing of the frame with TCN. The models with which we tested the HPFCN network are:

- **MODEL A**: model obtained from the training of the network HPFCN on data inpainted by using one of the three techniques previously described (e.g GMCNN, OPN, STTN).

- **MODEL B**: model obtained from the training of the network HPFCN on data inpainted by using one of the three techniques previously described (e.g GMCNN, OPN, STTN) and post-processed with TCN.

In order to clarify the data used during the current analysis, we summarized the training and the testing phases as follow:

- learning of model a 5.4a by training of HPFCN on data inpainted with a specific technique between GMCNN, OPN and STTN;
- testing of model a 5.1 on validation data inpainted with either GMCNN or OPN or STTN;
- testing of model a 5.1 on validation data inpainted with either GMCNN or OPN or STTN and post-processed with TCN;
- learning of model a 5.4a by training of HPFCN on data inpainted with a specific technique between GMCNN, OPN and STTN which has been processed with TCN too;
- testing model b 5.1 on validation data inpainted with either GMCNN or OPN or STTN;
- testing of model b 5.1 on validation data inpainted with euther GMCNN or OPN or STTN and post-processed with TCN;

The purpose of this analysis is to understand whether by training HPFCN separately on the given techniques GMCNN, OPN, or STTN, specific features of the analyzed network were learned so that the results of the corresponding prediction maps turn out to be more accurate.
As we will see later, the HPFCN network produces better localization maps when it is trained and tested with the same technique. For instance, if the HPFCN was trained on data inpainted with GMCNN, the best localization map is obtained by testing it on data manipulated with GMCNN.

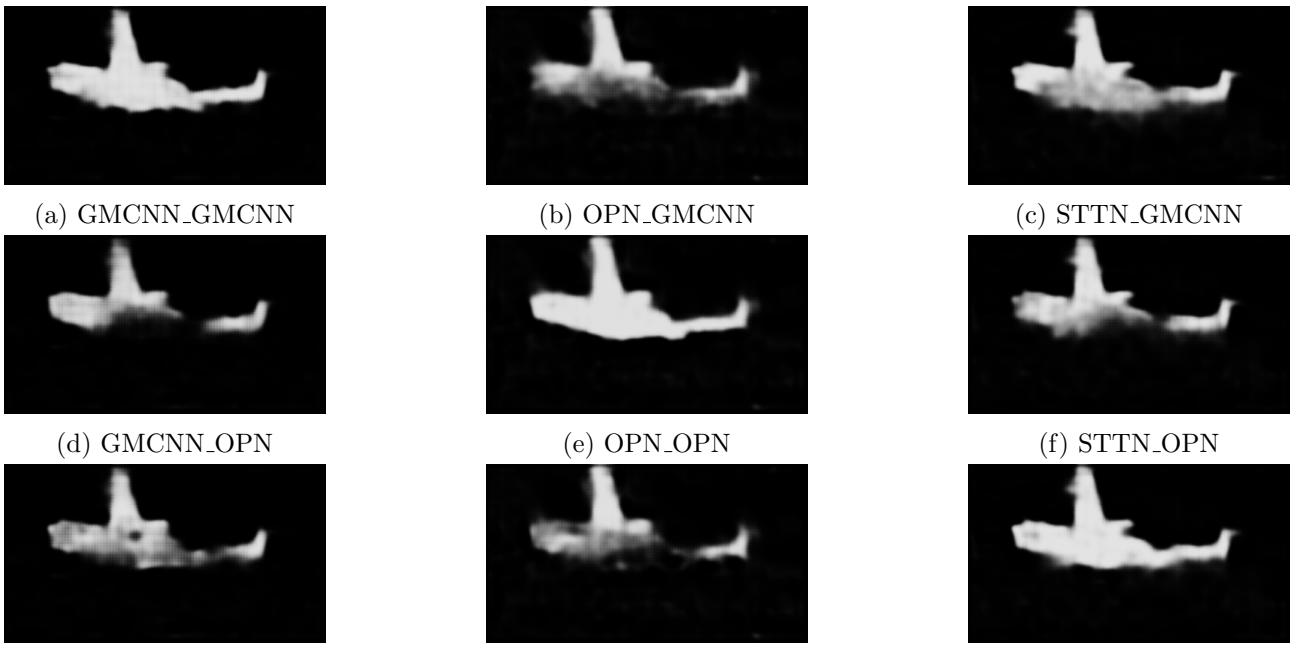
5.1.1 Robustness classification of networks GMCNN, OPN and STTN

We conducted some experiments in order to compare the performance of each technique and classify their robustness in learning and localization tasks. These experiments consist in analyzing each output frame obtained from testing model a 5.1 with each inpainting technique described. In general, considering the results shown in **Figure 5.1**, we can ascertain the validity of the three algorithms in the localization task.
From the pictures below we can recognize two different cases:

- testing of model a 5.1 on validation data inpainted with the same technique with respect to the technique utilized during model training (e.g. **Figure 5.1a**, **Figure 5.1e**, **Figure 5.1i**). It is evident that for each technique (GMCNN, OPN, and STTN) the removed object has been successfully localized. Despite some inaccuracies, especially along the edges of the object, it is evident that the techniques provide excellent results. The noise in the background is limited and no other elements present in the original video are contaminating it.
  From qualitative analysis, the network OPN comes out on top with respect to the two other techniques.

- test of model a 5.1 on validation data inpainted with a different technique with respect to the technique utilized during model training. Considering the case above, testing the neural network HPFCN with networks different from the one used in the training process, provide more imperfections in the localization task. For instance, if we consider the **Figure 5.1d**, where the network was trained on data painted with GMCNN and tested with the OPN network, the underside of the aircraft was not localized.

To summarize, it is evident that the HPFCN produces better localization maps when it is tested and trained on data manipulated with the same technique.



(a) GMCNN_GMCNN (b) OPN_GMCNN (c) STTN_GMCNN

(d) GMCNN_OPN (e) OPN_OPN (f) STTN_OPN

(g) GMCNN_STTN (h) OPN_STTN (i) STTN_STTN

Figure 5.1: Localization maps derived from training and testing HPFCN on data inpainted with each technique. In captions, the first term is the training technique and the second term is the testing technique used to, respectively, train and test HPFCN

5.1.2 Post-processing with TCN effects in localization task

In the previous experiment, we have not considered the post-processing technique TCN. Therefore, aware of the application of this technique to video inpainted with OPN, we want to analyze its behavior with respect to the other techniques.
The enforcing temporal consistency approach, on which TCN is based, can be applied both during training and testing of the HPFCN network.

In order to study the behaviour of the network, we analyzed three different cases:

- testing of model a on validation data inpainted with either GMCNN or OPN or STTN and post-processed with TCN
- testing of model b on validation data inpainted with either GMCNN or OPN or STTN
- testing of model b on validation data inpainted with either GMCNN or OPN or STTN and post-processed with TCN

From **Figure 5.2** we can claim that HPFCN, dealing with post-processed videos with TCN, fails in reproducing exactly the object and therefore data manipulated without TCN are preferable for this task. The output frames are not satisfactory: the airplane is not recognizable and the image has lots of white areas representing the noise of the background. Indeed, the distinction between the background from the airplane is not clear. Essentially, removing flickering affects some of the features used by the algorithms for inpainting localization, decreasing their accuracy.

(a) (b) (c)

Figure 5.2: (a) Localization maps obtained by testing HPFCN on double compressed post-processed frames. The frames have been inpainted with GMCNN, STTN and OPN. (a) Localization maps obtained by testing HPFCN on double compressed inpainted frames. The frames are inpainted with GMCNN, STTN and OPN and post-processed with TCN. (c) Localization maps obtained by testing HPFCN on double compressed post-processed frames. The frames are inpainted with GMCNN, STTN and OPN and post-processed with TCN.

While studying the Onion-Peel Networks for Deep Video Completion scientific paper [5] we observed an interesting detail: the nature of the first output frame. TCN network aims to enforce temporal consistency on videos by generating output frames that are temporally consistent with respect to the previous frames. In fact, the network has to set the first frame of the output video equal to the first input frame. Considering the resulting localization map an interesting detail, we conducted an experiment in order to understand whether the first output frame is enough well-defined or not. From the **Figure 5.3** we can infer that the iterative procedure applied to the target image with the aim of filling the peel region, typical of OPN, alters also the first prediction map. As a result, the first output frame turns out to be quite inaccurate.

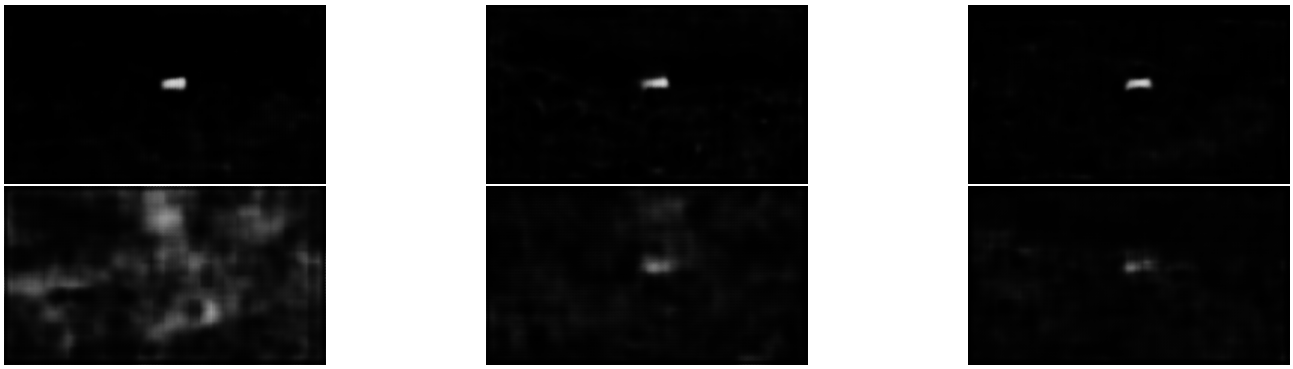


Figure 5.3: First localization maps obtained from training HPFCN on double compressed inpainted frames. The frames are inpainted with GMCNN, OPN and STTN (first row); first localization map obtained from training HPFCN on double compressed inpainted frames. The frames are inpainted with GMCNN, OPN and STTN and post-processed with TCN(second row)

5.1.3 Quantitative results

There are many metrics that can be used to measure the performance of a classifier. In order to measure quantitavely, the performance of HPFCN trained for deep video inpainting localizion, we observed them in terms of F_1-score.

In statistical analysis, the F_1-score is defined as the measure with which a test's accuracy is determined. It is computed from the precision and the recall of the test. In particular F_1-score is the harmonic mean of precision and recall.

$$F_1 = 2\frac{precision \cdot recall}{precision + recall} \tag{5.1}$$

Precision, also called positive predictive value (PPV), is calculated as the number of true positive results over the number of all positive results, including those not identified correctly.

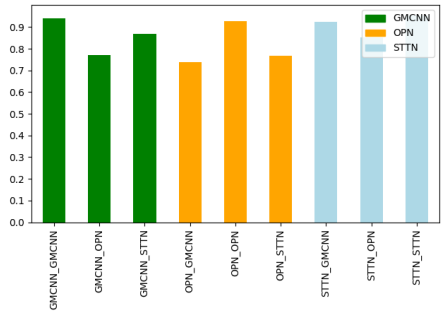
$$PPV = \frac{Number\ of\ true\ positives}{Number\ of\ true\ positives\ +\ Number\ of\ false\ positives} \tag{5.2}$$

Recall, also called sensitivity, is the number of true positive results divided by the number of all samples that should have been identified as positive.

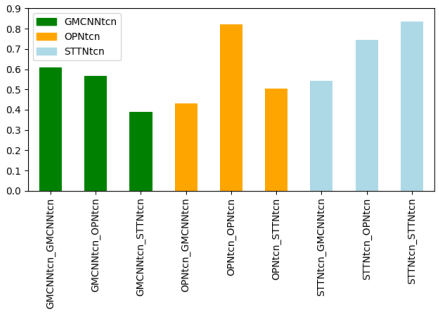
$$sensitivity = \frac{Number\ of\ true\ positives}{Number\ of\ true\ positives\ +\ Number\ of\ false\ negatives} \tag{5.3}$$

The highest possible value of an F1-score is 1, indicating that the number of true positive results is equal to the number of all positive results and to the number of all samples that should have been identified as positive. The lowest possible value is 0 which is the result of 5.1 where either the precision or the recall are equal to 0. Namely, we are interested in the model with the highest validation F1-score.

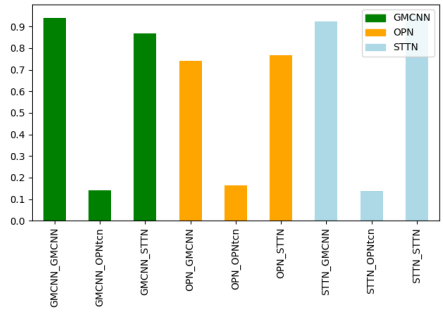
The results provided in **Figure 5.4** prove what we observed in the previous sections. Namely, **Figure 5.4a** verifies that HPFCN produces better location maps when it is tested and trained with data manipulated with the same technique and how the network works well in localization without any pos-processed data.
Figure 5.4b proves how TCN worsens the completion of a video. Specifically, the mean of F_1-score represented in **Figure 5.4a** is approximately equal to 0.8, which is surely greater than the mean F_1-score in the case of post-processing with TCN, which has been decreased to 0.5. In particular, looking at the third and fourth graphs we can compare the behavior of HPFCN in localizing tasks. Namely, in **Figure 5.4c**, we observe the effects of TCN if the network has not been trained on data post-processed with TCN and the **Figure 5.4d** illustrates how the network works after being trained on data post-processed with TCN. This last graph represents how HPFCN works on our data-set, since when the data has been manipulated with OPN, TCN has been applied. This last graph represents how HPFCN works on our data-set, since when the data has been manipulated with OPN, TCN has been applied. In particular, in order to improve the learning and generalization process of HPFCN in the case of post-processing with TCN, we trained and tested the network on more epochs, more precisely 20.
In conclusion, the deep neural network HPFCN turned out to be a good choice for the localization tasks in deep video inpainting, except in presence of post-processed video frames. Specifically, if we train the HPFN on data manipulated with OPN and post-processed with TCN, the network increases its robustness in localizing a post-processed video frame.



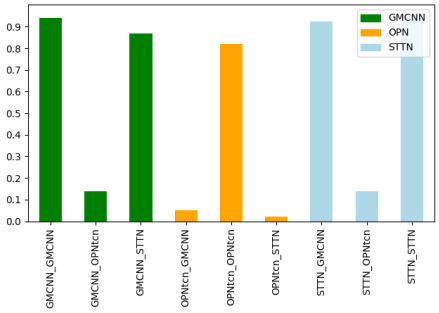
(a) Mean F_1-score of video inpainted with each technique. No post processing techniques applied. Training and testing over 10 epochs.



(b) Mean F_1-score of video inpainted with each technique. TCN Post-processing on training and testing. Training and testing over 10 epochs.



(c) Mean F_1-score of video inpainted with each technique. If OPN was applied in training process, post-processed TCN was applied too. Training and testing over 10 epochs.



(d) Mean F_1-score of video inpainted with each technique. When OPN was applied, post-processed TCN was applied too. Training and testing over 20 epochs.

Figure 5.4: Quantitative visualization of presented experiments

5.2 Detection: preliminary results

Since the literature on the detection of deep-inpainting in videos is quite poor, we focused our study on inpainting detection by implementing a statistical classifier that allows us to determine which technique has been used in the inpainting process.
We conducted two round of experiments: in the first one we used weights created with fine-tuning, but lead to poor results; in the second round of experiments another type of weights were used, they were created from scratch on data manipulated with the three inpainting techniques, especially with OPN and post-processed with TCN. In the next sections we describe the first type of weights and we discuss their non-optimal results. Finally, we propose the solution for detection based on the second type of weights.

5.2.1 Classification of inpainting technique

From the previous experiments 5.1, we noticed that the HPFCN produces better location maps when it is trained and tested with the same technique. This implies that, if the HPFCN was trained on data manipulated with GMCNN, the inpainted pixels in a location map obtained by testing HPFCN on data manipulated with GMCNN will yield a value greater than the value of the equivalent inpainted pixels obtained by testing HPFCN with STTN or OPN techniques. In particular, its value is rounded to 1 $x_{i,j}^{GMCNN} \approx 1 > x_{i,j}^{STTN} \approx x_{i,j}^{OPN}$.
According to this intuition, we have built three statistical models which allowed us to construct a distribution function for each technique and determine the used inpainting technique during the inpainting process accordingly.
Considering a set of videos $\mathcal{V}$, each one inpainted with the same technique $\mathcal{T}$ and of size $N \times I \times J \times M$ - where N represents the number of each video, I the height, J the width, and M the number of frames,

such that:

$$\mathcal{V} = \{\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_N}\}$$

we can represents each video as follow:

$$\mathbf{v_n} = \{\mathbf{I_n}^{(1)}, \mathbf{I_n}^{(2)}, ..., \mathbf{I_n}^{(M)}\}$$

where $\mathbf{I_n}^{(M)}$ represents the M frame of the n video of the set $\mathcal{V}$.
Training HPFCN on the same inpainting technique used during the inpainted process for $\mathbf{v_n}$, $\mathcal{T}$, we can evaluate a set $\mathcal{R}$ of $Y = N \times M$ localization maps $\mathbf{p_{maps}}$ of size equals of the one of the integer sampling grid $\mathcal{G} = I \times J$, for each frame of $\mathbf{V}$, such that:

$$\mathcal{P} = f(\mathbf{V}) = \{\mathbf{p_{maps}}_1, \mathbf{p_{maps}}_2, \cdots, \mathbf{p_{maps}}_Y\}$$

For each element of $\mathcal{P}$, $\mathbf{p_{maps_y}}$, we compute medium values of

- pixels of each localization maps with value greater or equal than 0.5

$$\tilde{\mathbf{H1}} = med(\{\forall(i,j) \in \mathcal{G} : \mathbf{p_{maps_y}}_{(i,j)} \geq 0.5\}) \tag{5.4}$$

- pixels of each localization maps with value less than 0.5

$$\tilde{\mathbf{H0}} = med(\{\forall(i,j) \in \mathcal{G} : \mathbf{p_{maps_y}}_{(i,j)} < 0.5\}) \tag{5.5}$$

and we modelled three distributions, each one representing the probability of classifying a video as inpainted with the respective technique. The three statistical distributions are designed such that:

$$(\tilde{\mathbf{H1}} - \tilde{\mathbf{H0}}) \subset \mathbf{\Theta_i}, \quad \forall \mathbf{p_{maps_y}} \in \mathcal{P} \tag{5.6}$$

where $\mathbf{\Theta}_i$, represents the corresponding statistical distribution.
At this point, we have estimated the likelihood of each frame $\mathbf{I_n}^{(i)}$ of each test video $\hat{\mathbf{V}}$ w.r.t. the distributions $\mathbf{\Theta}_1, \mathbf{\Theta}_2$ and $\mathbf{\Theta}_3$ by considering the hypothesis above described 5.4 and 5.5. The distribution model with the maximum likelihood w.r.t the frame $\mathbf{I_n}^{(i)}$ has been updated with a score equal to the likelihood value already estimated. Hence, by associating an inpainting technique to each frame $\hat{\mathbf{f}}_n$ of the test video $\hat{\mathbf{V}}_i$, we can correctly classify the entire video using the majority voting criteria. The accuracy of the implemented classifier is shown in **Figure 5.5** and we can see that, given a video manipulated with STTN, it imprecisely detects the technique. Currently, the classifier struggles in recognizing a video manipulated with STTN and in distinguishing a video forged with OPN and post-processed with TCN. We want to improve the accuracy of the detector by allowing it to classify a post-processed video.

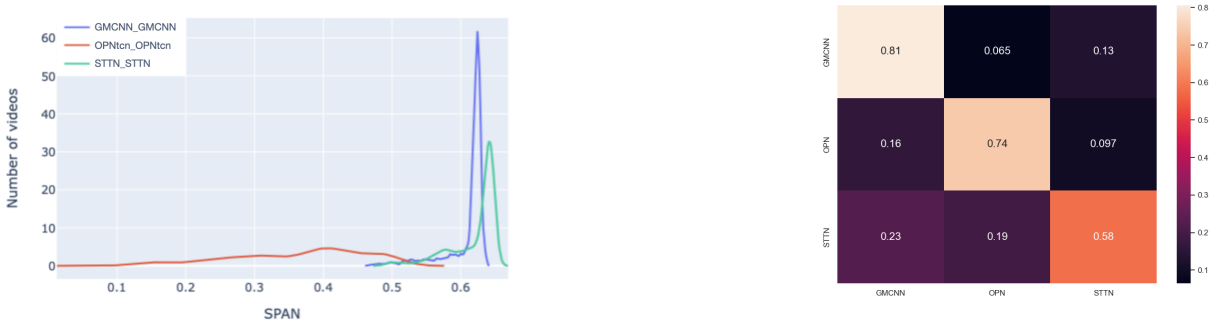


Figure 5.5: On the left the probability distribution of representing a video as the respective inpainting technique. On the right the confusion matrix obtained on double compressed inpainted frames not post processed. The frames are inpainted with STTN, OPN and GMCNN.

By calculating the median score as presented in formula 5.6, we can plot statistical distributions as the three represented in **Figure 5.5**. In particular, we can see how the pixels value of localization map are distributed in case of data manipulated with GMCNN, OPN and STTN. We have to remind that each manipulation of OPN is followed by a post-processing process with TCN. It is important to note that the three distributions are not strictly separable, in particular we can notice that the STTN and GMCNN distributions have a significant overlapping zone. This problem is less evident considering OPN distribution. Unfortunately, this condition has been reflected to the computed confusion matrix. Indeed, as we can see from the confusion matrix in **Figure 5.5**, a video manipulated with STTN is recognizes with a lower accuracy, precisely with a score of 0.58 because the classifier struggle in recognizing STTN from GMCNN. On the other hand, GMCNN is recognizible with a score equal to 0.81 and OPN produces great results as well, with a score of 0.74.

5.2.2 Classification of post-processing TCN

Once we analyzed the freshly designed residuals, we also noticed that the variance of each channel is numerically different from one another, in contrast to the distribution of variances in case of a not post-processed one, where the distributions are quite similar.
This intuition enabled us to perform a preliminary classification of post-processed frames comparing them with not post-processed ones. Firstly, we defined a score s 5.7 by looking at each video frame $\mathbf{I}$ to check whether a video was post-processed or not by looking at each frame.

$$s = \frac{\min(\sigma^2(\mathbf{I}))}{\max(\sigma^2(\mathbf{I}))} \tag{5.7}$$

In order to check the validity of the score in classifying the post-processed frames with respect to the not-post-processed ones, we tested it on video frames belonging to our validation data-set. This allows us to represent the score in two different distributions, one for video frames post-processed with TCN and one for video frames not post-processed. The plot perfectly illustrates the two distributions: we can observe that for videos not post-processed the score is close to 1, instead, for video frames post-processed we observe a score close to 0.8. Hence, by checking both post-processed and not post-

Figure 5.6: Distribution of TCN scores

processed with TCN we construct a Receiver Operating characteristic Curve (ROC) 5.7. A Receiver Operating characteristic Curve is a plot that describes the ability of a binary classifier system, in our case our score 5.7, of correctly classifying a given data. The curve is obtained by plotting the true positive rate - sensitivity, recall 5.3 - against the false positive rate - the probability of false alarm. Considering a binary classification problem, such as the recognition of post-processing of TCN on video frames, the outcomes are labeled either with positive or negative values. Based on the relationship between the outcome label and the actual value, we obtain four possible results: true positive, false positive, true negative, or false negative. Based on the number of occurrences of these results we calculate the sensitivity and, accordingly, the false positive rate and the score 5.7. Therefore, each point of the curve represents the probability of correctly classifying a post-processed frame in case of a given False Positive value, namely a possible value of the threshold. The ROC computes a threshold parameter tau_{TCN} which performs the threshold for classifying an instance as positive. Usually, the instance is classified as "positive" when its value is bigger than the parameter tau_{TCN}, "negative" otherwise. We refer to the positive and negative decision of the classifier respectively with H_1 (or

match) and H_0 (or mis-match) A good threshold has a good trade-off between True Positive Rate, TPR, and the False Positive Rate, FPR, and, generally, the FPR should never exceed a value of 0.05. Specifically, by looking at this ROC, we are able to distinguish TCN processed frames from the not processed ones with a $TPR \approx 0.75$ and a $FPR \approx 0.05$. From the confusion matrix above we can see

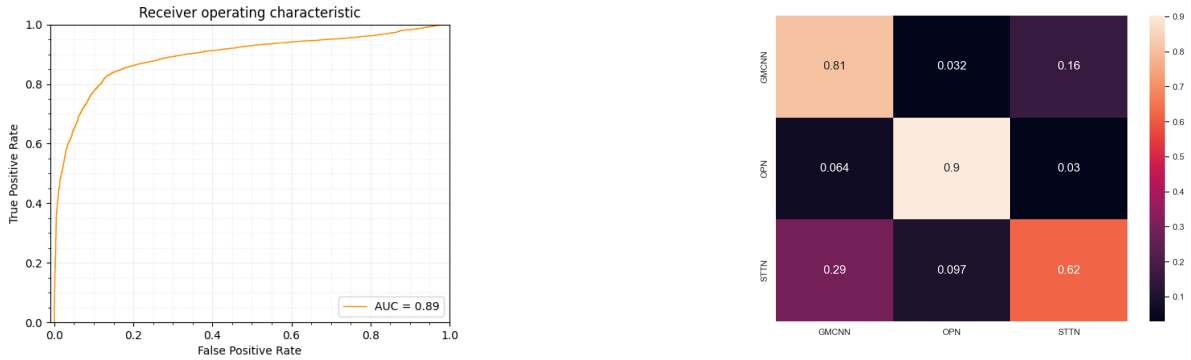


Figure 5.7: On the left the ROC obtained using 5.7 computed on double compressed frames processed with and without TCN. On the right the Confusion matrix obtained on double compressed inpainted frames. OPN inpainted frames are post-processed with TCN; GMCNN and STTN ones are not post processed.

the improvement of our classifier by adding post-processing detection. In particular, the classification of OPN has increased by about 20% from a value of 0.77 to 0.9. Additionally, we can observe that the computation of this score helps the classifier in detecting a STTN manipulated frame give a GMCNN one.

5.2.3 Classification of a pristine video

In the end, the classifier has been updated to recognize if a video is pristine or not. Considering a set of videos $\mathcal{V}$, each one inpainted with one of the presented techniques, and a set of videos $\mathcal{P}$, each one pristine of size $N \times I \times J \times M$ - where N represents the number of each video, I the height, J the width, and M the number of frames, such that:

$$\mathcal{V} = \{\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_N}\}$$

$$\mathcal{P} = \{\mathbf{p_1}, \mathbf{p_2}, ..., \mathbf{p_N}\}$$

we can represents each video as follow:

$$\mathbf{v_n} = \{\mathbf{I_n}^{(1)}, \mathbf{I_n}^{(2)}, ..., \mathbf{I_n}^{(M)}\}$$

$$\mathbf{p_n} = \{\mathbf{H_n}^{(1)}, \mathbf{H_n}^{(2)}, ..., \mathbf{H_n}^{(M)}\}$$

where $\mathbf{I_n}^{(M)}$ represents the M frame of the n video of the set $\mathcal{V}$ and $\mathbf{H_n}^{(M)}$ represents the M frame of the n video of the set $\mathcal{P}$

By evaluating a set $\mathcal{R}$ of $Y = N \times M$ localization maps $\mathbf{p_{maps}}$ of size equals of the one of the integer sampling grid $\mathcal{G} = I \times J$, for each frame of $\mathbf{V}$ and $\mathbf{P}$, such that:

By evaluating a set $\mathcal{R}$ of $Y = N \times M$ localization maps $\mathbf{p_{maps}}$ of size equals of the one of the integer sampling grid $\mathcal{G} = I \times J$ and a set $\mathcal{R}_2$ of $Y = N \times M$ localization maps $\mathbf{l_{maps}}$ of size equals of the one of the integer sampling grid $\mathcal{S} = I \times J$, for each frame of $\mathbf{V}$ and $\mathbf{P}$, such that:

$$\mathcal{P} = f(\mathbf{V}) = \{\mathbf{p_{maps}}_1, \mathbf{p_{maps}}_2, ..., \mathbf{p_{maps}}_Y\}$$

$$\mathcal{L} = f(\mathbf{P}) = \{\mathbf{l_{maps}}_1, \mathbf{l_{maps}}_2, ..., \mathbf{l_{maps}}_Y\}$$

For each element of $\mathcal{P}$, $\mathbf{p_{maps_y}}$ and $\mathcal{L}$, $\mathbf{l_{maps_y}}$, we have calculated the medium number of pixels grater

of 0.5. Hence, we have plotted the Receiver Operating characteristic Curve illustrated in **Figure 5.9** where the represented hypothesis are:

$$\tilde{\mathbf{H1}} = average(\{\forall(i,j) \in \mathcal{G} : \mathbf{p_{maps}}_{\mathbf{y}_{(i,j)}} > 0.5\}) \tag{5.8}$$

$$\tilde{\mathbf{H0}} = average(\{\forall(i,j) \in \mathcal{S} : \mathbf{p_{maps}}_{\mathbf{y}_{(i,j)}} > 0.5\}) \tag{5.9}$$

The ROC computes the measure for classifying a pristine video given a False Positive value. This measure is known as threshold and the presented enable us to distinguish a pristine video from a manipulated one with a $TPR \approx 1.0$ and a $FPR \approx 0.09$. The results of the classifier are illustrated in **Figure 5.8**.

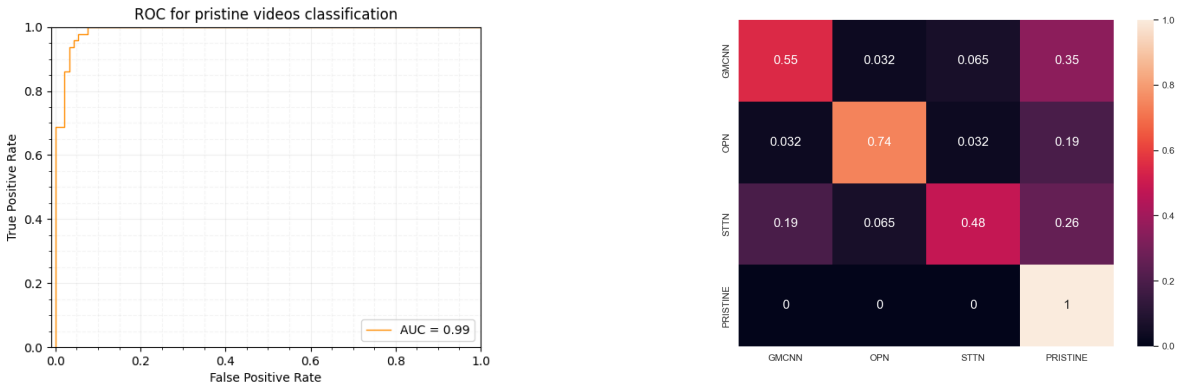


Figure 5.8: On the left the ROC obtained using 5.8 and 5.9 computed on double compressed frames inpainted and pristine video frames. On the right the confusion matrix obtained on double compressed inpainted frames. OPN manipulated frames has been post-processed with TCN. Threshold value $tau_{prist} = 124.92$

Finding a solution for detecting pristine videos

Since the results in **Figure 5.8** are quite inaccurate, we have tried to tackle this problem by decreasing the threshold with which a pristine video is recognized w.r.t. a manipulated one. Reducing the threshold allows to fewer manipulated videos to be classified as pristine. Precisely, we decreased the FPR and increased the TPR. Actually, we update FPR to $FPR \approx 0.2$ end the TPR to $TPR \approx 1.0$. Unfortunately, decreasing the threshold compensates for the accuracy of the identification of the three inpainting techniques but decreases that of the pristine. The results are illustrated in **Figure 5.9**

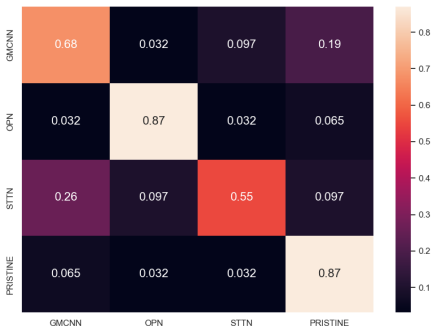


Figure 5.9: Confusion matrix obtained on double compressed inpainted frames. OPN manipulated frames has been post-processed with TCN. Threshold value $tau_{prist} = 5.20$

Looking at these results, the proposed method cannot be a definitive solution: despite the ROC

results being particularly good, the confusion matrix obtained does not produce good classifications and, for this reason, more structured and elaborate techniques must be taken into consideration. An example of a state-of-the-art technique used for video inpainting detection is the encoder-decoder architecture VIDNet. More details are provided in **chapter 2**. Nevertheless, we tried to improve the performance of our classifier by using new weights and adapting the classifier to them.

5.2.4 Solution for increasing the performance of the classifier

In order to increase the performance of classifier, we tried to repeat each detection experiment with different weights. Actually, we have trained the HPFCN neutral network on different weights that allow us to create new models. The weights designed, unlike the previous ones, have not undergone any fine-tuning process. Practically, we are searching for better models to generalize the detection. The present section aims to compare the presented weights w.r.t. the new ones by looking at the classifier results.

Classification of inpainting technique

Applying the new weights to the first version of the classifier we obtained better results. In **Figure 5.10** we observe more accuracy in distinguishing each technique. Especially, we can observe a great improvement in the recognition of manipulated videos with GMCNN. Unfortunately, uncertainty remains in determining whether a video is manipulated with STTN. From the alongside models, we can show the representation of the new three statistical distributions obtained from the training of HPFCN on the designed weights. The inaccuracy in discriminating a video manipulated with STTN from one manipulated with GMCNN is explained by the significant overlapping zone of the two distributions.

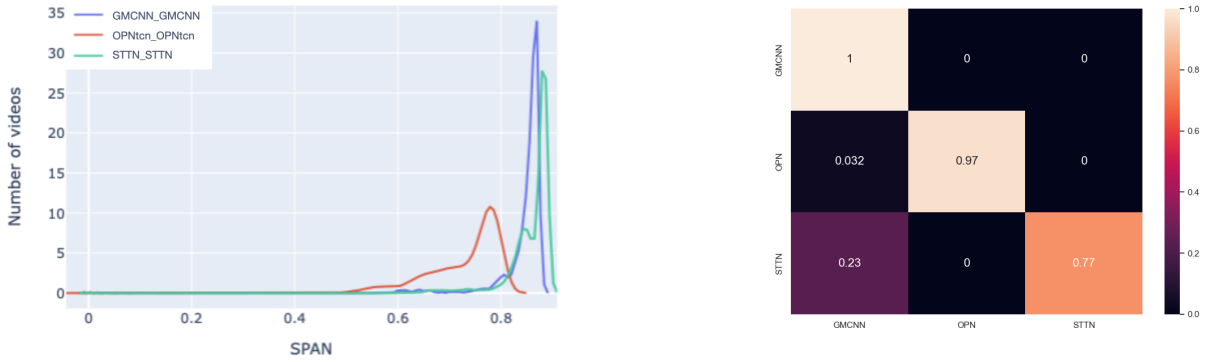


Figure 5.10: On the left the probability distribution of representing a video as the respective inpainting technique. The models are obtained by the training of HPFCN on not fine-tuned weights. On the right the confusion matrix obtained on double compressed inpainted frames not post processed. The frames are inpainted with STTN, OPN and GMCNN. The weights are not fine-tuned.

Classification of post-processing TCN

Since the classifier already performs very well in distinguishing the three techniques, classifying post-processing frames by calculating the score 4.1 by considering the variance of each manipulated video frame $\mathbf{I}$, is not advantageous for improving the performance in recognizing them. The confusion matrix obtained is the same as the one in **Figure 5.5**. This is because the nature of the new weights: they have been generated by considering data manipulated with OPN and post-processed with TCN, differently from the other weights which we have fine-tuned thus leading to more inaccuracy. Then, we preferred to train them on more epochs in order to enlarge the improvement.

Classification of a pristine video

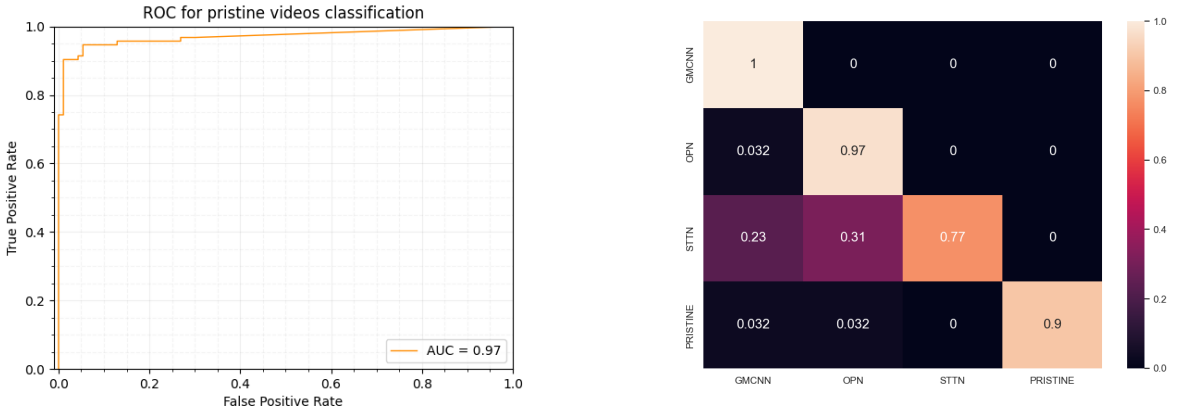


Figure 5.11: On the left the ROC obtained using 5.8 and 5.9 computed on double compressed frames inpainted and pristine video frames. On the right the confusion matrix obtained on double compressed inpainted frames. OPN inpainted frames are post processed with TCN; GMCNN and STTN ones are not post processed

The ROC in **Figure 5.11** shows the behaviour of the computed score for distinguishing video pristine from video manipulated. The ROC computes the measure for classifying a pristine video given a False Positive value. The presented enable us to distinguish a pristine video from a manipulated one with a $TPR \approx 0.96$ and a $FPR \approx 0.15$. The results of the classification are illustrated in **Figure 5.11**. The confusion matrix shows how good our chosen score is. We can see that the score valuated for distinguishing pristine videos from forged ones provides actually great results: the higher values of confidence are among the diagonal and the most of them are very close to 1. Unfortunately, we can still notice the persistent problem in recognizing a video manipulated with STTN: the classifier still falls short in recognizing STTN-manipulated videos from a given GMCNN-manipulated video or a OPN-manipulated video. However, we are satisfied with the implementation of the proposed statistical classifier because it is able to distinguish a manipulated video and recognize the inpainting technique used during the manipulation process. Nevertheless, more sophisticated methods need to be developed, for instance, deep-learning based.

6 Conclusions

This chapter sums up the content of this dissertation, focusing on the theoretical and practical results produced by our study on Deep video in-painting and my personal contribution to this ambitious project.

Results The performance of the proposed framework has been evaluated with both synthetic and realistic images in a qualitative and quantitative way. The Convolutional Neural Network HPFCN confirmed itself as our choice for the localization tasks in deep video in-painting, except in presence of video frames post-processed with TCN. In this case, the network struggles in extracting the best features from the training process and generalize in the localization of the corrupted region. By intensifying the number of epochs with the network has been training, the performance increased but not enough to consider the localization task as solved.

It must be noted that TCN strongly decrease the performance of HPFCN in localization problem. By isolating the high frequencies composing the input frames of each video, we noticed some persis-

tent artifacts by training them with HPFCN. These allows us to perform a preliminary classification of post-processed and not post-processed frames increasing the performance of the classifier, in particular in determining OPN manipulations.

As a result of this project we also proposed a score to check whether a video was post-processed or not by looking at the variance computed with the FFT in the RGB channels of the input video frames. With the computed score, we are able to distinguish TCN processed frames from the not processed ones with a $TPR \approx 0.75$ and a $FPR \approx 0.05$.
The implementation of the statistical classifier is another important milestone in our research. It has been developed and updated three times. In its first version the classifier detects which in-painting technique has been used during manipulation process. The classifier in **Figure 5.5** is able to correctly detect the used technique with a confidence equal to 0.81 in case of video manipulated with GMCNN, 0.74 in case of video manipulated with OPN and 0.58 in case of video manipulated with STTN. The inaccuracy in classification of video manipulated with STTN is due to a large overlapping zone of distributions, especially between STTN and GMCNN. The second version of the proposed classifier enabled it to accurately distinguish a post processed video from a not post-processed one. Looking at the confusion matrix in **Figure 5.7** we can observe that the post-processing on in-painted frames with OPN is detected with an accuracy equal to 90%. As we expected, recognizing the post-processing of TCN has led to an increase in the performance of the classifier. Finally, in its last version we added the ability to distinguish a pristine video from a manipulated one. More precisely, given a pristine video, the classifier is able to labeled it as pristine with a confidence of 100%. Nevertheless, given a manipulated video it is not accurate enough to distinguish it as manipulated. We designed the pristine score by counting the number of pixels in the input frame greater than 0.5. **Figure 5.8** shows that the actual issue cannot be considered solved. The miss-classification level is high and so we tried to improve the performance by decreasing the threshold of the computed ROC. Reducing the threshold allows to fewer manipulated videos to be classified as pristine. Unfortunately, the inaccuracy in distinguishing the manipulated videos is still high so the problem cannot be marked as solved. We can notice these results in **Figure 5.9**.

Since the classifier is still quite inaccurate we proposed a solution for increasing its performance. Specifically, we have trained the HPFCN neutral network on different weights that allow us to create new models. The weights designed, unlike the previous ones, have not undergone any fine-tuning process. In this way, the proposed classifier is able to determine which technique has been used in manipulating the video with much more accuracy. The first version of the classifier, **Figure 5.10**, is able to predict each technique with an accuracy greater than 0.77, in particular, can always recognize a video manipulated with GMCNN. On the other hand, since the classifier already performs very well in distinguishing the three techniques, classifying post-processing frames from the not-processed ones is not advantageous for further improving the performance in recognizing them. Lastly, we have observed that the score studied for recognizing a pristine video from a forged one is efficient: the detector performs really well also in case of distinguishing pristine from manipulated videos. The results are shown in **Figure 5.11**.

Future developments By detecting the in-painting technique used during the forging process, we can make the network more robust in terms of localization because we address the learning process on that particular technique. In this manner, the network is able to generalize in a better way and better localization maps can be achieved. For this reason, the project will continue until better performance is achieved both in terms of detection and localization of deep video in-painting. Several steps that we want to deepen are:

- **Enlarging dataset:** we want to include new in-painting techniques similar to OPN, thus with a particular interest toward techniques that use temporal stabilization and yield a more robust solution.
- **Temporal feature:** we consider to expand our research field on new temporal features. For example, features exploiting optical flow.

We are confident that our projects results' will have a positive impact, not only in the court and in forensics-strictly related fields, but also in less formal and more social ones by limiting the fake-news phenomenon that, in our information society, could cause more long-term damage than any other digital threat.

Bibliography

[1] Suraj K A, Sumukh H, Shrijan S Shetty, and Jayashree R. A technique for video inpainting using deep learning. In *2020 IEEE International Conference for Innovation in Technology (INOCON)*, pages 1–5, 2020.

[2] Davide Cozzolino and Luisa Verdoliva. Noiseprint: a cnn-based camera model fingerprint. *CoRR*, abs/1808.08396, 2018.

[3] Xiangling Ding, Yifeng Pan, Kui Luo, Yanming Huang, Junlin Ouyang, and Gaobo Yang. Localization of deep video inpainting based on spatiotemporal convolution and refinement network. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2021.

[4] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. *CoRR*, abs/1808.00449, 2018.

[5] Sungho Lee, Seoung Wug Oh, Daeyeun Won, and Seon Joo Kim. Copy-and-paste networks for deep video inpainting. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4412–4420, 2019.

[6] Haodong Li and Jiwu Huang. Localization of deep inpainting using high-pass fully convolutional network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8301–8310, 2019.

[7] Ruixin Liu, Bairong Li, and Yuesheng Zhu. Temporal group fusion network for deep video inpainting. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(6):3539–3551, 2022.

[8] Ruixin Liu, Bairong Li, and Yuesheng Zhu. Temporal group fusion network for deep video inpainting. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(6):3539–3551, 2022.

[9] Shuli Ma. Video inpainting exploiting tensor train and sparsity in frequency domain. In *2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP)*, pages 441–445, 2021.

[10] Luisa Verdoliva. Media forensics and deepfakes: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):910–932, 2020.

[11] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. *CoRR*, abs/1810.08771, 2018.

[12] Shujin Wei, Haodong Li, and Jiwu Huang. Deep video inpainting localization using spatial and temporal traces. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8957–8961, 2022.

[13] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. *CoRR*, abs/2007.10247, 2020.

[14] Peng Zhou, Ning Yu, Zuxuan Wu, Larry S Davis, Abhinav Shrivastava, and Ser-Nam Lim. Deep video inpainting detection. *arXiv preprint arXiv:2101.11080*, 2021.