Journal of Theoretical and Applied Information Technology
15th January 2022. Vol.100. No 1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DEEPFAKE DETECTION BASED ON THE XCEPTION MODEL

[1]JAMAL EL ABDELKHALKI , [1]MOHAMED BEN AHMED, [1]ANOUAR ABDELHAKIM BOUDHIR

[1]LIST laboratory, Abdelmalek Essaadi University

[1]Computer Sciences department, Faculty of Sciences and Techniques of Tangier – Morocco

E-mail: [1]elabdelkhalki@gmail.com, [1]m.benahmed@gmail.com, [1]hakim.anouar@gmail.com

ABSTRACT

Thanks to artificial intelligence, everybody can easily create deepfakes without any particularly technic knowledge. By analyzing faces movements. But these will become more and more realistic as technological developments progress, and therefore more and more problematic ...The rapid evolution in synthetic image generation and manipulation has now come to a point where it raises significant concerns on the implication on the society. With the advent of fake news and its harmful effects on social networks, the dissemination of deepfakes on the web therefore constitutes a new technological threat. Manipulation, disinformation, humiliation, defamation ... the dangers of deepfakes will be more and more numerous.
In this post, first, we will describe in brief how deep learning with Depthwise Separable Convolutions can be the most useful and promising techniques to detect deepfakes

Keywords: *FaceForensics, CNN, Xception, MTCNN, Cyber security.*

1. INTRODUCTION

Face swapping is a type of artificial intelligence , it has become an arising subject in recent years in computer vision and graphics .In fact, currently many efforts was given to Face swapping [1], [2].Those research's overcome the unwieldy and dull manual face editing process,[3] therefore accelerate face editing development. However, the most notable use case is that this technology has created legitimate concerns, especially when others use it for nefarious and abuse purposes .Due to the popularization of deepfake, General Public and Authorities were warned from the dangerous repercussions of this dilemma. Therefore, it is compulsory to take countermeasures immediately, especially inventions of technics that can detect video hoaxes. Many groups have contributed datasets to forge detection. FaceForensics++ [4], Deep FakeDetection[5] and DFDC [6]comprising manipulated video footages.

Large efforts are being contributed, regarding the continual acceleration and growing of convoluted manipulated content, the research community is finding advanced technics and methods in order to confront face manipulation detection[7].

Traditional fake detection methods in media forensics have been commonly based on: in-camera fingerprints, the analysis of the intrinsic fingerprints introduced by the camera device, both hardware and software, such as the optical lens [8], color filter array and interpolation [9], [10] and compression [11], [12], among others, and out camera fingerprints, the analysis of the external fingerprints introduced by editing software, such as copy-paste or copy move different elements of the image [13][14]reduce the frame rate in a video [15], [16].
Deep neural networks have proven to be very effective for this task and several works can be found in the current literature [17] .The second category of facial forgeries is Identity manipulation, through this method the face of a person replace the face of another person[18]. .

This category is known as face swapping. It became popular with wide-spread consumer-level applications like Snapchat.Deepfakes performs face swapping through deep learning .While face swapping based on simple computer graphic technics running in real time but remain still insufficient, DeepFakes has more complexities and need to be trained for each pair of videos, which is a time-consuming task.

The AI technologies that power deepfakes and other fake media are evolving rapidly, making deepfakes so difficult to detect that sometimes even human evaluators cannot reliably tell the difference, as almost anyone can create deepfakes using existing tools. So far, many methods have been proposed to detect deepfakes [19],[20],[21],[22],[23]. Most of them are based on deep learning. malicious and positive uses of deep learning methods has emerged. To address the threat of face-swapping technology or deepfakes, the U.S. Defense Advanced Research Projects Agency (DARPA) of the States the U.S. Defense Advanced Research Projects Agency (DARPA) launched a media forensics research program (called Media Media Media Forensics or MediFor) to accelerate the development of methods to detect digital visual fake media [24].

In this work, we worked on automatic deepfake detection with a very reliable process using a new method, the latest in deep learning, in particular extremely powerful image features using convolutional neural networks (CNN), so what are the most reliable models compared to the most used deep learning models in deepfake detection ? Does using Xception for deepfake detection give better results than the different published approaches?

In this paper, we will propose a large scale dataset named FaceForensics++ to facilitate the research of face forgery detection towards real-world scenarios. So, our approach is first we need to extract face features to detect the face. Then we apply improved Xception model to detect whether the face is real or fake.

.

2. RELATED WORKS

Recently, the research community revealed that we can attain impressive detection results if we supervise deep learning approaches, compared to the traditional media forensics, based on the high frequency pixel-level signals. This one uses a layer of fixed high –pass filters [25],[26] , learned filters [27], or even by recasting hand-crafted features working on residuals as a convolutional neural network [28].
The paper divides some various fields in computer vision and digital multimedia forensics, in the following paragraphs, we cover the most related papers.

Face Manipulation Methods: In the last two decades interest in virtual face manipulation has rapidly increased. A comprehensive state-of-the-art report has been published by Zollhofer et al [29] . Bregler et al.[30]presented an image-based approach called Video Rewrite to automatically create a new video of a person with generated mouth movements. With Video Face Replacement [31], Dale et al introduced on the first automatic face swap methods .

For videos, the main body of work focuses on detecting manipulations that can be created with relatively low effort, such as dropped or duplicated frames [15], [32].However, most of the recent literature is concentrated on CNN-based solutions both through supervised and unsupervised learning [33], [34].Some other papers explicitly refer to detecting manipulations related to faces, like distinguishing computer generated faces from ²natural ones [35], [36], morphed faces [36], face swapping [37] and DeepFakes[37].
If we could get remarkable results from the recent publications, the problem of robustness was shown only in some works even if it has more importance in practical applications.
We can expect to have data driven methods more powerful therefore we can create better forgery detectors for facial imagery through our new data.

Recently much effort has been devoted to detection of manipulations that was made using deep learning. On the other hand, other works focused on the detection of manipulations on faces, but concerning our methods, we are not limited to a specific type of manipulations, we only need few samples to adapt to new manipulations.

3. BACKGROUND

3.1 Deep learning

Deep learning or deep neural networks (DNNs) takes inspiration from how the brain works and forms a sub module of artificial intelligence. The main strength of deep learning architectures is the capability to understand the meaning of data when it is in large amounts and to automatically tune the derived meaning with new data with brand-new data without the necessity for an area expert knowledge. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are two types of deep learning architectures predominantly applied in real-life scenarios.

Generally, CNN architectures are used for spatial data.

he concepts behind the various deep learning architectures are discussed in a mathematical way.

3.1.1 Deep Neural Network (DNN)

DEEP NEURAL NETWORK (DNN) a feed forward neural network (FFN) creates a directed graph in which a graph is composed of nodes and edges. FFN passes information along edges from one node to another without formation of a cycle. Multi-layer perceptron (MLP) is a type of FFN that contains 3 or more layers, specifically one input layer, one or more hidden layer and an output layer in which each layer has many neurons, called as units in mathematical notation. The number of hidden layers is selected by following a hyper parameter tuning approach[38].

A (FFN) feed forward neural network creates a directed graph in which a graph is composed of nodes and edges. The information passed by the FFN along edges from one node to another without formation of a cycle. (MLP) Multi-layer perceptron is a type of FFN that contains 3 or more layers, precisely one input layer, one or more hidden layer and an output layer in which each layer has many neurons, called as units in mathematical notation. The number of hidden layers is selected by following a hyper parameter tuning approach. The transformation of information from one layer to another is done in the direct without considering the past values. Moreover, neurons in each layer are fully connected. An MLP with n hidden layers can be mathematically formulated as given below:

$$(x) = (Hn{-}1(Hn{-}2(\bullet\bullet\bullet(H1(x))))) \quad (1)$$

H defines a hidden layer. This way of stacking hidden layers is typically called deep neural networks (DNNs) [Fig 1]. Shows a pictorial representation of DNN architecture with n hidden layers. It takes input:

$$X = X1, X2, \bullet\bullet\bullet, Xp{-}1, Xp \quad (2)$$

$$and\ outputs:\ O = O1, O2, \bullet\bullet\bullet, Oc{-}1, Oc \quad (3)$$

Each hidden layer uses Rectified linear units (ReLU) as the non-linear activation function. This helps to reduce the state of vanishing and error.

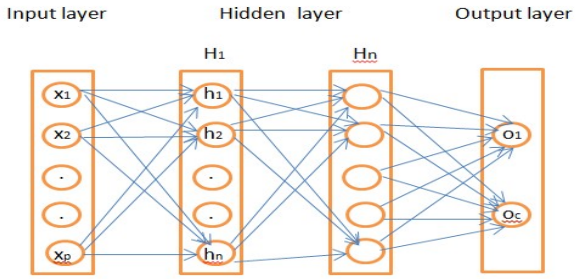

Figure 1. Architecture of DNN with n hidden layers

Gradient issue, ReLU has turned out to be more proficient and capable of accelerating the entire training process altogether[39]. ReLU is defined mathematically as follows:

$$f\ (x) = (0, x) \quad (4)$$

Where x denotes input

3.1.2 Convolutional Neural Network (CNN)

Before we review how deep learning is employed for malware classification, let us revisit how convolutional neural networks are used for image classification. An image is input to the network in its raw pixel format. The image goes through a sequence of convolutional layers which can be viewed as automatically computing image features at different levels of abstraction. The spatial dimension of feature maps decreases due to max pooling layers. Neurons in higher layers correspond to larger receptive fields of pixels in the input image over which features are being computed. These convolutional layers are followed by fully connected layers (dense layers), or in more modern architectures, by global average pooling layers. Right in the end, we have a classification output layer which outputs probabilities of the image being in different categories. For speech recognition, we can convert speech signals into a 2-D image called spectrogram in which time is one axis and other is frequency, and we can apply similar techniques[31].

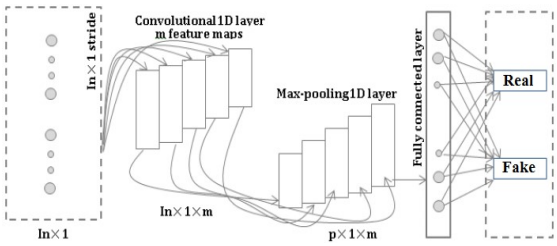

Figure 2. Architecture of CNN for Deep fakes detection[38]

It is shown in [Fig 2], where all connections and hidden layers and its units are not shown. Here, m implies the total number of filters, in denotes the number of input features & on the other hand, p implies decreased feature dimension, it depends on pooling length. In this work, CNN network consisted of convolution 1Dlayer, pooling 1D layer, and fully connected layer. A CNN network can have more than one convolution 1D layer, pooling 1D layer and fully connected layer. In a convolutional 1D layer, the filters slide over the 1D sequence data and extracts optimal features. The features that are extracted from each filter are grouped into a new feature set called a feature map. The number of filters and the length are chosen by following a hyper parameter tuning method. This in turn uses non-linear activation function, ReLU on each element. The dimensions of the optimal features are reduced using pooling 1D layer using max pooling, min pooling or average pooling. Since the maximum output within a selected region is selected in max pooling, we adopt max pooling in this work. Finally, the CNN network contains a fully connected layer for classification. In a fully connected layer, each neuron contains a connection to every other neuron. Instead of passing the pooling 1D layer features into a fully connected layer, it can also be given to recurrent layer LSTM to capture the sequence related information. Finally, the LSTM features are passed into a fully connected layer for classification [40].

Convolutional layers: These layers apply a certain number of convolution operations (linear filtering) to the image in sequence. Typically, these filters extract edge, color, and shape information from the input image. Basically, the filters operate on subregions of an image and perform computation such that it produces a single value as output for each subregion. The output (say x) of this layer is typically forwarded to a nonlinear function (called ReLU activation) which is defined as:

$$(x) = (0, x) \quad (5)$$

Pooling layers: This layer is responsible for down sampling (i.e., reducing the spatial resolution of the input layers) the data produced from convolution layers so that processing time can be reduced, and so that computational resources can handle the scale of the data. This is due to the fact that as a result of pooling, the number of learnable parameters is reduced in the subsequent layers of the network. Max pooling is a commonly used pooling technique that keeps the maximum value in a region (e.g. 2x2 non-overlapping regions of data) and discards aa values.

Fully connected layers: This layer performs classification on the output generated from convolution layers and pooling layers. Every neuron in this layer is connected to every neuron present in the previous layer. This type of layer is typically followed by a Dropout layer that improves the generalization capability of the model by preventing overfitting which is commonly occurring problem in deep learning domain [31].

3.2 Depthwise

The most important part of this architecture [Fig 3] is depthwise separable convolution operation. This operation is one step ahead of separable convolution. Spatially separable convolutions, sometimes briefly called separable convolutions are convolutions that can be separated across their spatial axes.

For a normal convolution operation the total number of multiplication is

$$Total\ number\ of\ multiplication\ per\ image = Kw * Kh * C * Nk * Nv * Nh \quad (6)$$

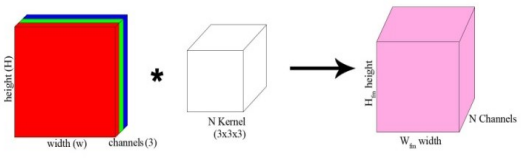

Figure 3.Basic convolution operation[41]

Where Kw is kernel width, Kh is kernel height, C is number of channels, Nk is number of kernels, Nv is

number of vertical slides, Nh is number of horizontal slides. For an image size H X W, number of vertical slide (Nv) = (H - Kh + 1) and number of horizontal slide (Nh) = (W - Kw + 1).

The spatially separated kernel, first convolves the (Kw × 1) kernel and subsequently the (1 x Kh) kernel. So with (Kw x 1) kernel total number of multiplication is

$$Kw * 1 * C * Nk * H * (W - Kw + 1) \quad (7)$$

So with (1 x Kh) kernel total number of multiplication is

$$1 * Kh * C * Nk * (H - Kh + 1) * W \quad (8)$$

So eventually we can reduce the number of multiplication operations. The point is that only a minority of nuclei are spatially separable; the majority cannot be separated this way. So if you relied on spatially separable nuclei when training a convolutional neural network, you would severely limit the network; the network will not perform well than that formed with traditional nuclei, even if it requires less resources[42].

A depthwise separable convolution holds the same characteristic as spatially separable convolutions, but it splits the kernels into two smaller ones with the same results but fewer multiplication. There are two operations that makes depthwise separable convolutions more effective:

1. Depthwise convolutions
2. Pointwise convolutions

As we've seen above, normal convolutions over volumes convolve over the entire volume, i.e., over all the channels at once, producing a Width x Height x 1 volume for every kernel. Using N kernels therefore produces a Width x Height x N volume called the feature map.

In depthwise [Fig 4] separable convolutions, particularly the first operation – the depthwise convolution – this does not happen in that way. Rather, each channel is considered separately, and one filter per channel is convolved over that channel only. See the example below:

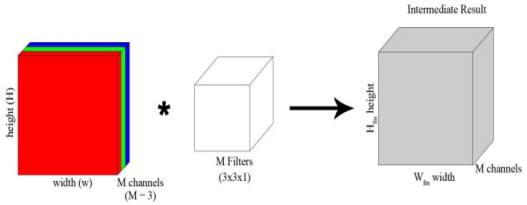

Figure 4.Depthwise convolutions[41].

Here, we would use 3 one-channel filters (M=3), since we're interpreting an RGB image. The result is not the end result but an intermediate result that is to be interpreted further in the second phase of the convolutional layer, the pointwise convolution.

Those are filters of 1×1 pixels but which cover all the M intermediate channels generated by the filters, in our case M=3.

And since we're trying to equal the original convolution, we need N of them. Remember that a convolution over a volume produces a Some Width x Some Height x 1 volume, as the element-wise multiplications performed over three dimensions result in a one-dimensional scalar value. If we would thus apply one such pointwise filter, we would end up with a Hfm x Wfm x 1 volume. As the original convolution produced a Hfm x Wfm x N volume, we need N such pointwise filters[43][Fig 5].

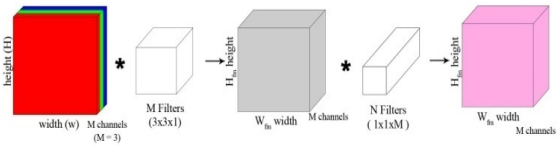

Figure 5.Pointwise convolutions[41]

First, using depthwise convolutions using M filters, an intermediate result is produced, which is then processed into the final volume by means of the pointwise convolutions. Taking those volumes together, M volume x N volume yields that the operation is equal to the original kernel volume: (3x3x1 times 1x1xM = 3x3xM = 3x3x3, the volume

of our N original kernels indeed). Since we have N such filters, we produce the same result as with our N original kernels.

If the total number of multiplication of normal full convolution is Mc, then for depth wise convolution the multiplication will be $\frac{McMc}{NkNk}$ since that's the volume of each individual filter. And for point wise convolution total number of multiplications will $\frac{Mc}{Kh * Kw}$ as the kernel volume is (1 x 1 x Nk). So, total number of multiplications in combined process will be

$$\frac{Md}{Mc} = \frac{1}{Nk} + \frac{1}{KhKd} \quad (9)$$

For Kh = 5, Kw = 5 and Nk = 10 depthwise separable convolution saved 14 more multiplication. [44]

4. METHODOLOGY OF THE PROPOSED MODELS

4.1 Deepfakes

Deepfakes means face replacement that is based on deep learning. It is made through replacing a face in a target sequence by face that has been observed in a source video or image collection. There are different public implementations of deepfakes, such as FakeApp [4] and the faceswap github [3], those are the most remarkable. The methods of those operations are made through two autoencoders with a shared encoder that are trained to reconstruct training images of the source and the target face. Face detector is used to crop and align the images, in order to create a fake image, they apply the trained encoder and the decoder of the source face of the target face. Then we find that the auto encoder output is blended with the rest of the image by applying Poisson image editing [45]. Concerning our dataset, we use the faceswap github implementation, we moderately adjust the implementation by replacing the manual training data selection with fully automated data loader, across using the default parameters to train the video-pair models. We also publish the model as part of our dataset since the training of these models is very time-consuming, as though , that will allows to easily generate additional manipulations of these persons with different post-processing [46] .

4.2 The proposed system

The proposed architecture is an Xception model which is proved as a better algorithm than vgg-16, ResNet50 and InceptionV3 in most of the classical image classification challenges. In this section, we introduce a deep learning model for fake detection using Xception model to detect forgeries. The whole pipeline is to track a face [Fig 6] based on MTCNN and then apply Xception model to classify. We used state-of-the-art face tracking method to track the face in the video and to extract the face region of the image and used a conservative crop around the center of the tracked face, enclosing the reconstructed face. This incorporation of domain knowledge improves the overall performance of a forgery detector in comparison to a native approach that uses the whole image as input.

Figure 6. Principe method tracking face[4]

There are three stages of MTCNN. The first step is to take the image and resize it to different scales in order to build an image pyramid [Fig 7], which is the input of the following three-staged cascaded network.

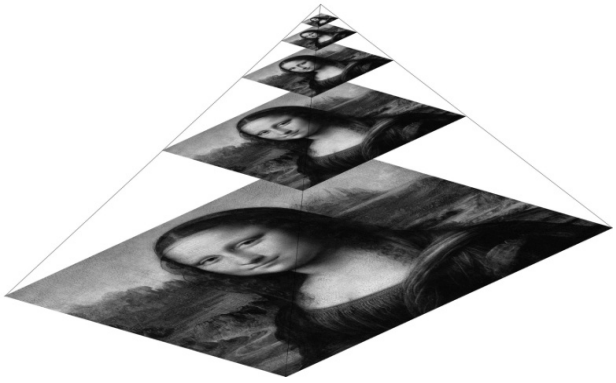

Figure 7. Example of an image pyramid[47]

Journal of Theoretical and Applied Information Technology
15th January 2022. Vol.100. No 1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

This proposal network is a fully convolutional network (FCN). The difference between a CNN and a FCN is that a fully convolutional network does not use a dense layer as part of the architecture. This Proposal Network is employed to get candidate windows and their bounding box regression vectors. Bounding box regression is a popular technique to predict the localization of boxes when the goal is detecting an object of some predefined class, in this case faces. After obtaining the bounding box vectors, some refinement is completed to mix overlapping regions. The final output of this stage is all candidate windows after refinement to downsize the quantity of candidates.[48]

All candidates from the P-Net [Fig 8] are fed into the Refine Network. Notice that this network may be a CNN, not a FCN just like the one before since there's a dense layer at the last stage of the specification. The R-Net further reduces the amount of candidates, performs calibration with bounding box regression and employs non-maximum suppression (NMS) to merge overlapping candidates. The R-Net outputs whether the input is a face or not, a 4 element vector which is the bounding box for the face, and a 10 element vector for facial landmark localization.[49]

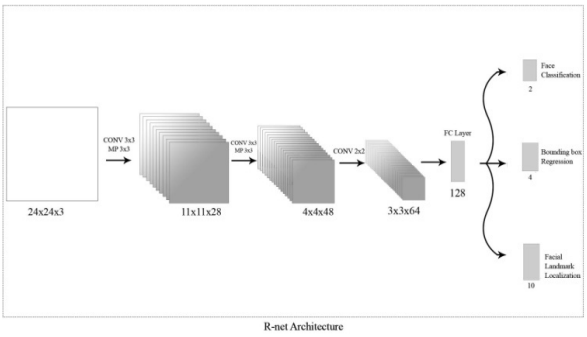

Figure 8. R-net Architecture[50]

The Xception model architecture [Fig 9] has 36 convolutional layers which forms the feature extraction base of the network. In our experimental evaluation, we will only focus on image classification and therefore the convolutional base will be followed by a logistic regression layer. Optionally, one may insert danse layers before the logistic regression layer, which is explored in the experimental evaluation section. The 36 convolutional layers are structured into14 modules, all of which have linear residual connections around them, except for the first and last modules.

In short, the Xception architecture is a stack of depthwise separable convolution layers with residual connections.[51]

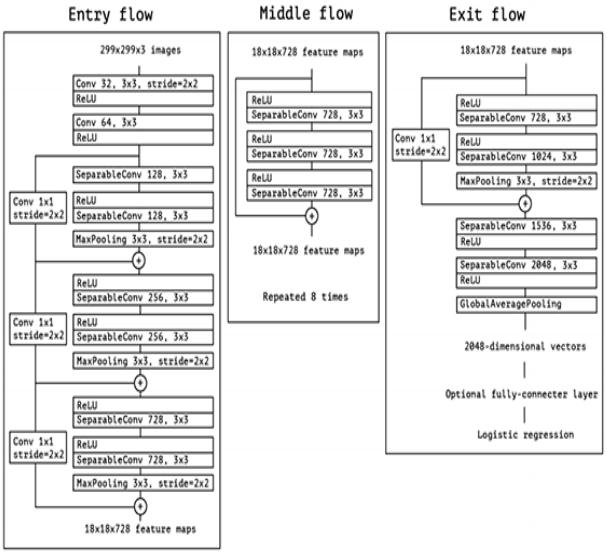

Figure 9.The full architecture of Xception model.[52]

The Instantiates the Xception architecture[53].

def Xception(include_top=True,

weights='imagenet',

input_tensor=None,

input_shape=None,

pooling=None,

classes=1000,

**kwargs):

The default input image size for this model is 299x299.

Arguments:

include_top: whether to include the fully-connectedlayer at the top of the network.

Weights: one of None (random initialization), imagenet (pre-training on ImageNet), or the path to the weights file to be loaded.

Input_tensor: optional Keras tensor (output of layers.Input ()) to use as image input for the model.

Input_shape: optional shape tuple, only to be specified if include_top is False (otherwise the input shape has to be (299, 299, 3).

It should have exactly 3 inputs channels,and width and height should be no smaller than (150, 150, 3) would be one valid value.

Pooling: Optional pooling mode for feature extraction when include_top is False.

None means that the output of the model will be the 4D tensor output of the last convolutional block.

Avg means that global average pooling will be applied to the output of the last convolutional block, and thus the output of the model will be a 2D tensor.

Max means that global max pooling will be applied.

Classes: optional number of classes to classify images into, only to be specified if include_top is True, and if no weights argument is specified

5. DATA PREPARATION AND ENVIRONMENT SETUP

The Xception model is trained on Image net dataset. The last fc layer is replaced with an output filter 2 as the number of output classes is two (real or fake). Then the model is fine-tuned on face forensic dataset with above architecture.

The proposed model Xception; We tried with many classification algorithms like Rich Models for Steg analysis of Digital Images plus SVM, Recasting residual-based local descriptors as convolutional neural networks, A deep learning approach to universal image manipulation detection using a new convolutional layer, a compact facial video forgery detection network and Distinguishing computer graphics from natural images using convolutional neural networks. And found Xception model over performed.

As the model is Xception the training data need to resize (299, 299) before going into the Xception model. We apply normalization of mean 0.5 and std 0.5 to all the channels. These are the only preprocessing that needs to apply on images before sending them to the model.

We use python and pytorch deep learning framework to execute the script. All the requirements to build the environments are

- ✓ Pillow-6.2.1
- ✓ munch-2.5.0
- ✓ numpy-1.17.4
- ✓ Keras
- ✓ pretrainedmodels-0.7.4
- ✓ six-1.13.0
- ✓ torchvision-0.4.2
- ✓ tqdm-4.40.2
- ✓ dlib-19.19.0

6. EXPERIMENTS AND EVALUATION

A core contribution of this paper is our FaceForensics++ dataset [Table 1] extending the preliminary FaceForensics dataset; we chose to collect videos from YouTube. Early experiments with all manipulation methods showed that the pristine videos have to fulfill certain criteria.To Ensure adequate video quality, we only downloaded videos that offer a resolution of 480p or higher. For every video, they save its metadata to sort them by properties later on. we first process all downloaded videos with the Dlib face detector [54], which is based on Histograms of Oriented Gradients (HOG). In this step, they track the largest detected face by ensuring that the centers of two detections of consecutive frames are pixel-wise close. The histogram-based face tracker was chosen to make sure that the resulting video sequences contain little occlusions and, thus, contain easy-to-manipulate faces. Except FaceSwap, all methods need a sufficiently large set of images during a target sequence to coach on. They select sequences with at least 280 frames. To ensure a high quality video selection and to avoid videos with face occlusions, we performed a manual screening of the clips which resulted in 1,000 video sequences containing 509,914 images[55]-[5].

Table 1.Training and Validation split[4].

Method	Train	Validation	Test
DeepFakes	366,835	68,506	73,768
Face2Face	366,843	68,511	73,770
FaceSwap	291,434	54,618	59,640
NeuralTextures	291,834	54,630	59,672

Owing to the goal of detecting fakes in real world scenarios, this work mainly explores how common distortion appearing in real scenes affects the model performance.
The baselines trained on the standard training set of FaceForensics++ achieve much better performance on the hidden test set than all other dataset. This proves the higher quality of FaceForensics++ over prior works, making it more useful for real-world face forgery detection.

Journal of Theoretical and Applied Information Technology
15th January 2022. Vol.100. No 1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

This work studies the effect of perturbations towards the forgery detection model performance. In contrast to prior work, this work tries to evaluate the baseline accuracies when applying different distortions to the training and the test sets, in order to explore the function of perturbations in face forensics dataset.

The experiments [Table 2] show that all detection approaches achieve a lower accuracy on the GAN-based Neural Textures approach.
Cozzolino et al. [28] that propose detection system based in CNN-based network ,Bayar and Stamm [27] works] that uses a constrained convolutional layer followed by two convolutional, two max-pooling and three fully-connected layer, Rahmouni et al.[35] Adopt different CNN architectures with a global pooling layer that computes four statistics. Neural Textures is training a unique model for every manipulation which results[17] in a higher variation of possible artifacts. While DeepFakes is also training one model per manipulation, it uses a fixed post-processing pipeline similar to the computer-based manipulation methods and thus has consistent artifacts.
The paper that introduces the FaceForensics dataset used in our tests presents better classification results using the state-of-the-art network for image classification.
To train the Xception classification network to distinguish between all three manipulation methods and the pristine images, we adapted the final output layer to return four class probabilities. Data the network is able to achieve 98.77% accuracy.

Table 2. Accuracy four manipulation methods (DF: DeepFakes, F2F:Face2Face, FS: FaceSwapand NT:NeuralTextures)

	DF	F2F	FS	NT
Cozzolino et al.[28]	81.78	85.32	85.69	80.60
Bayar and Stamm[27]	90.18	94.93	93.14	86.04
Rahmouni et al.[35]	82.16	93.48	92.51	75.18
MesoNet[17]	95.26	95.84	85.96	85.96
Xception	98.77	98.31	98.19	94.44

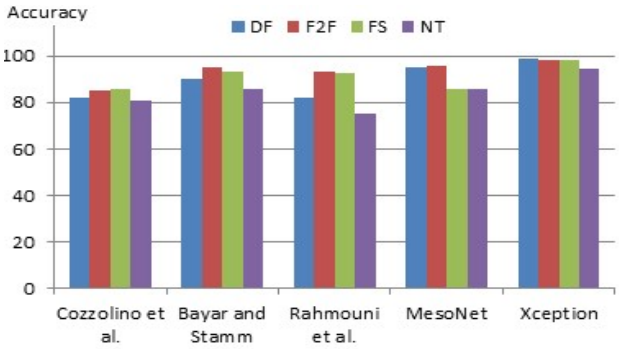

Figure 10. Accuracy comparison

Figure 11.The results of different frames

These are some sample results of full resolution images tasted on Xception with full precision. Here we are detecting the face using the MTCNN model and draw a rectangle around them to visualize and then apply Xception to classify the face. The green color is for real and the red color is for fake

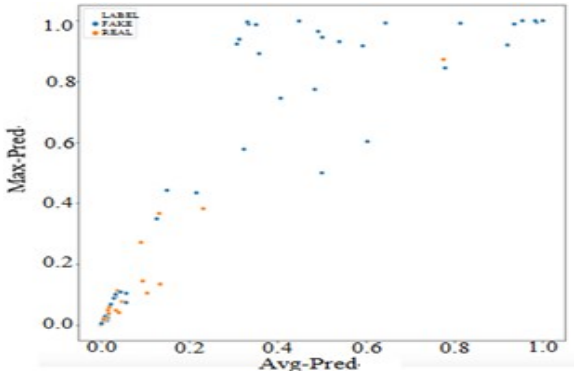

Figure 12.Average v/s max prediction

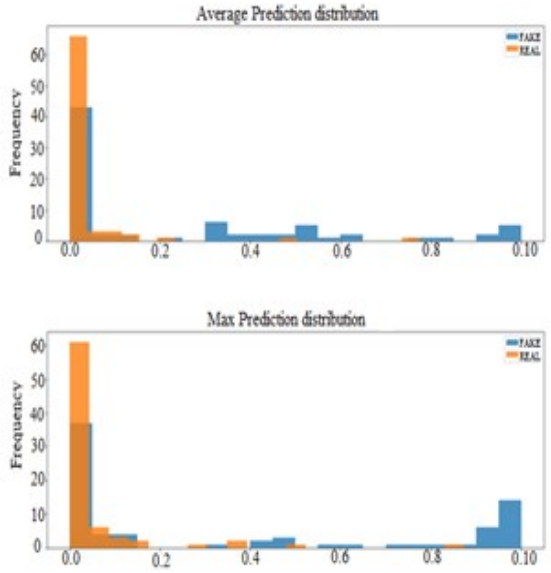

Figure 13.Average and max prediction distribution of Xception model

In the model Xception we can see that if the average prediction is more than 0.22 then there are more chances of an image to be fake. It is not 100% correct because there are some spikes at 0.44 and 0.78. Excluding these points we can set the average prediction threshold to 0.22. And in terms of max prediction the threshold is around 0.41 [Fig 12].

This work finds the accuracy is nearly 100% when the models are trained and tested on the standard set. This is reasonable because the strong baselines perform very well in a clean dataset with the same distribution. Most of the video-level methods except C3D are more robust to perturbations on test sets than Xception. This setting is very common because different distributions of the training and the test sets lead to decreasing model accuracies. Hence, the lacks of perturbations in the face forensics dataset cutbacks the model performance for real-world face forgery detection with even more complex data distribution. When the corresponding distortions are applied to the training and test sets, the accuracy increases. However, this setting is impractical because the distributions of the training and test sets are still the same.

We trained the dataset model we can see that if the average prediction is more than 0.22 then there are more chances of an image to be fake. It is not 100% correct because there are some spikes at 0.44 and 0.78. Excluding these points we can set the average prediction threshold to 0.22. And in terms of max prediction the threshold is around 0.41[Fig 13].

7. BENCHMARK

In this section we cite the best solutions presented with details of each approach used.
Selim Seferbekov [56] used MTCNN [57] for face detection and an EfficientNet B-7 [58] for feature encoding. The structured parts of the faces were removed during training as a form of augmentation.
WM [59]. The third approach NTechLab [60], used a set of EfficientNets in addition to using mixup augmentation during training. Eighteen Years Old [61] , used a set of image and video models, including EfficientNet, Xception, ResNet [62] and a SlowFast video network [63]. Finally,The Medics [64], also used MTCNN for face detection, as well as a set of 7 models, including MTCNN. as well as a set of 7 models, including 3 3D CNNs (which performed better than temporal models).
Our presented face detection and full image model of Xception, trained on the FaceForensics dataset, one frame was sampled per second of video.
When using the image-based model for detection, there are two thresholds to set the per-frame detection threshold and a threshold that specifies how many frames to exceed. the per frame threshold for a video to be identified as false (or the per-frame detection threshold). as false (or the per-frame threshold). These thresholds should be set in tandem - for good performance, a low per-frame threshold will likely result in a high per-video threshold, and vice versa. To normalize for video length, we evaluated the frames-per-video threshold only on frames that contained a detectable face. In cross-validation on the training set, we found the optimal frames-per-video thresholds that maximized.
The comparison in Table 2 is not always performed with the same datasets and protocols, so it should be interpreted with caution. Despite this, it is clear

that the Xception algorithm achieves the best results for FaceForensics++, but there are some problems and challenges discussed in the paper,
which need future research and studies to improve and increase the detection capability and solve the existing problems.

8. CONCLUSION

In this paper, we presented the different publications have been made for the detection of deepfake using deeplearning, we could demonstrate that they can be detected by trained fake detectors, even though the methods of facial image manipulation reveal outstanding results.
First, this paper presents the recognized public datasets for the four main types of face manipulation: face synthesis manipulation, face attribute manipulation, and face expression manipulation. To train the detectors, we create a novel dataset of manipulated face videos that outperforms all existing public forensic datasets.
Finally our used model based on Xception algorithm for face manipulation detection is concluded and analyzed and we were able to have better results;
We hope that this dataset will serve as a springboard for other researchers to work in the field of digital media forensics, especially focusing on facial forgeries.

REFERENCES:

[1] iperov, iperov/DeepFaceLab. 2021. Consulté le: févr. 06, 2021. [En ligne]. Disponible sur: https://github.com/iperov/DeepFaceLab
[2] deepfakes, deepfakes/faceswap. 2021. Consulté le: févr. 06, 2021. [En ligne]. Disponible sur: https://github.com/deepfakes/faceswap
[3] shaoanlu, shaoanlu/faceswap-GAN. 2021. Consulté le: févr. 06, 2021. [En ligne]. Disponible sur: https://github.com/shaoanlu/faceswap-GAN
[4] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, et M. Niessner, « FaceForensics++: Learning to Detect Manipulated Facial Images », in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), oct. 2019, p. 1-11. doi: 10.1109/ICCV.2019.00009.
[5] « Contributing Data to Deepfake Detection Research », Google AI Blog. http://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html (consulté le févr. 06, 2021).
[6] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, et C. C. Ferrer, « The Deepfake Detection Challenge (DFDC) Preview Dataset », arXiv:1910.08854 [cs], oct. 2019, Consulté le: févr. 06, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/1910.08854
[7] X. Ju, « An Overview of Face Manipulation Detection », p. 12, 2020.
[8] I. Yerushalmy et H. Hel-Or, « Digital Image Forgery Detection Based on Lens and Sensor Aberration », Int J Comput Vis, vol. 92, no 1, p. 71-91, mars 2011, doi: 10.1007/s11263-010-0403-1.
[9] A. C. Popescu et H. Farid, « Exposing digital forgeries in color filter array interpolated images », IEEE Trans. Signal Process., vol. 53, no 10, p. 3948-3959, oct. 2005, doi: 10.1109/TSP.2005.855406.
[10] H. Cao et A. Kot, « Accurate Detection of Demosaicing Regularity for Digital Image Forensics », Information Forensics and Security, IEEE Transactions on, vol. 4, p. 899-910, janv. 2010, doi: 10.1109/TIFS.2009.2033749.
[11] Z. Lin, J. He, X. Tang, et C.-K. Tang, « Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis », Pattern Recognition, vol. 42, no 11, p. 2492-2501, nov. 2009, doi: 10.1016/j.patcog.2009.03.019.
[12] Y.-L. Chen et C.-T. Hsu, « Detecting Recompression of JPEG Images via Periodicity Analysis of Compression Artifacts for Tampering Detection », IEEE Transactions on Information Forensics and Security, vol. 6, p. 396-406, juin 2011, doi: 10.1109/TIFS.2011.2106121.
[13] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo, et G. Serra, « A SIFT-Based Forensic Method for Copy–Move Attack Detection and Transformation Recovery », IEEE Transactions on Information Forensics and Security, vol. 6, no 3, p. 1099-1110, sept. 2011, doi: 10.1109/TIFS.2011.2129512.
[14] D. Cozzolino, G. Poggi, et L. Verdoliva, Splicebuster: A new blind image splicing detector. 2015. doi: 10.1109/WIFS.2015.7368565.
[15] A. Gironi, M. Fontani, T. Bianchi, A. Piva, et M. Barni, « A video forensic technique for detecting frame deletion and insertion », in 2014 IEEE International Conference on Acoustics, Speech and Signal Processing

(ICASSP), Florence, Italy, mai 2014, p. 6226-6230. doi: 10.1109/ICASSP.2014.6854801.

[16] B. C. Hosier et M. C. Stamm, « Detecting Video Speed Manipulation », in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, juin 2020, p. 2860-2869. doi: 10.1109/CVPRW50498.2020.00343.

[17] D. Afchar, V. Nozick, J. Yamagishi, et I. Echizen, « MesoNet: a Compact Facial Video Forgery Detection Network », arXiv:1809.00888 [cs, eess], sept. 2018, Consulté le: févr. 06, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/1809.00888

[18] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, et M. Nießner, « FaceForensics: A Large-scale Video Dataset for Forgery Detection in Human Faces », arXiv:1803.09179 [cs], mars 2018, Consulté le: févr. 06, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/1803.09179

[19] S. Lyu, « DeepFake Detection: Current Challenges and Next Steps », arXiv:2003.09234 [cs], mars 2020, Consulté le: déc. 21, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/2003.09234

[20] L. Guarnera, O. Giudice, C. Nastasi, et S. Battiato, « Preliminary Forensics Analysis of DeepFake Images », 2020 AEIT International Annual Conference (AEIT), p. 1-6, sept. 2020, doi: 10.23919/AEIT50178.2020.9241108.

[21] M. T. Jafar, M. Ababneh, M. Al-Zoube, et A. Elhassan, « Forensics and Analysis of Deepfake Videos », in 2020 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, avr. 2020, p. 053-058. doi: 10.1109/ICICS49469.2020.239493.

[22] L. Trinh, M. Tsang, S. Rambhatla, et Y. Liu, « Interpretable and Trustworthy Deepfake Detection via Dynamic Prototypes », in 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, janv. 2021, p. 1972-1982. doi: 10.1109/WACV48630.2021.00202.

[23] M. A. Younus et T. M. Hasan, « Effective and Fast DeepFake Detection Method Based on Haar Wavelet Transform », in 2020 International Conference on Computer Science and Software Engineering (CSASE), avr. 2020, p. 186-190. doi: 10.1109/CSASE48920.2020.9142077.

[24] « Media Forensics ». https://www.darpa.mil/program/media-forensics (consulté le déc. 21, 2021).

[25] Y. Liu, Q. Guan, X. Zhao, et Y. Cao, « Image Forgery Localization Based on Multi-Scale Convolutional Neural Networks », IEEE Trans. Geosci. Remote Sensing, vol. 56, no 12, p. 7109-7121, déc. 2018, doi: 10.1109/TGRS.2018.2848473.

[26] Y. Rao et J. Ni, « A deep learning approach to detection of splicing and copy-move forgeries in images », in 2016 IEEE International Workshop on Information Forensics and Security (WIFS), Abu Dhabi, United Arab Emirates, déc. 2016, p. 1-6. doi: 10.1109/WIFS.2016.7823911.

[27] B. Bayar et M. C. Stamm, « A Deep Learning Approach to Universal Image Manipulation Detection Using a New Convolutional Layer », in Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, Vigo Galicia Spain, juin 2016, p. 5-10. doi: 10.1145/2909827.2930786.

[28] D. Cozzolino, G. Poggi, et L. Verdoliva, « Recasting Residual-based Local Descriptors as Convolutional Neural Networks: an Application to Image Forgery Detection », arXiv:1703.04615 [cs], mars 2017, Consulté le: févr. 06, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/1703.04615

[29] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, et M. Niessner, « Face2Face: Real-Time Face Capture and Reenactment of RGB Videos », p. 9.

[30] C. Bregler, M. Covell, et M. Slaney, ACM SIGGRAPH 97 Video Rewrite: Driving Visual Speech with Audio.

[31] M. Kalash, M. Rochan, N. Mohammed, N. D. B. Bruce, Y. Wang, et F. Iqbal, « Malware Classification with Deep Convolutional Neural Networks », in 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, févr. 2018, p. 1-5. doi: 10.1109/NTMS.2018.8328749.

[32] C. Long, E. Smith, A. Basharat, et A. Hoogs, « A C3D-Based Convolutional Neural Network for Frame Dropping Detection in a Single Video Shot », in 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), juill. 2017, p. 1898-1906. doi: 10.1109/CVPRW.2017.237.

[33] T. Zhou, M. Brown, N. Snavely, et D. G. Lowe, « Unsupervised Learning of Depth and Ego-Motion from Video », in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, juill. 2017, p. 6612-6619. doi: 10.1109/CVPR.2017.700.

[34] S. Hussein, P. Kandel, C. W. Bolan, M. B. Wallace, et U. Bagci, « Lung and Pancreatic Tumor Characterization in the Deep Learning Era: Novel Supervised and Unsupervised Learning Approaches », IEEE Trans. Med. Imaging, vol. 38, no 8, p. 1777-1787, août 2019, doi: 10.1109/TMI.2019.2894349.

[35] N. Rahmouni, V. Nozick, J. Yamagishi, et I. Echizen, « Distinguishing computer graphics from natural images using convolution neural networks », in 2017 IEEE Workshop on Information Forensics and Security (WIFS), Rennes, déc. 2017, p. 1-6. doi: 10.1109/WIFS.2017.8267647.

[36] C. Galea et R. A. Farrugia, « Matching Software-Generated Sketches to Face Photographs With a Very Deep CNN, Morphed Faces, and Transfer Learning », IEEE Transactions on Information Forensics and Security, vol. 13, no 6, p. 1421-1431, juin 2018, doi: 10.1109/TIFS.2017.2788002.

[37] Y. Nirkin, Y. Keller, et T. Hassner, « FSGAN: Subject Agnostic Face Swapping and Reenactment », p. 10.

[38] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, et S. Venkatraman, « Robust Intelligent Malware Detection Using Deep Learning », IEEE Access, vol. 7, p. 46717-46738, 2019, doi: 10.1109/ACCESS.2019.2906934.

[39] X. Glorot, A. Bordes, et Y. Bengio, « Deep Sparse Rectifier Neural Networks », p. 9.

[40] T. N. Sainath, O. Vinyals, A. Senior, et H. Sak, « Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks », in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, Queensland, Australia, avr. 2015, p. 4580-4584. doi: 10.1109/ICASSP.2015.7178838.

[41] « Understanding separable convolutions – MachineCurve ». https://www.machinecurve.com/index.php/2019/09/23/understanding-separable-convolutions/ (consulté le mars 05, 2021).

[42] Z. Qin, Z. Zhang, D. Li, Y. Zhang, et Y. Peng, « Diagonalwise Refactorization: An Efficient Training Method for Depthwise Convolutions », in 2018 International Joint Conference on Neural Networks (IJCNN), juill. 2018, p. 1-8. doi: 10.1109/IJCNN.2018.8489312.

[43] L. Kaiser, A. N. Gomez, et F. Chollet, « Depthwise Separable Convolutions for Neural Machine Translation », arXiv:1706.03059 [cs], juin 2017, Consulté le: mars 05, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/1706.03059

[44] P. Zhang, E. Lo, et B. Lu, « High Performance Depthwise and Pointwise Convolutions on Mobile Devices », AAAI, vol. 34, no 04, Art. no 04, avr. 2020, doi: 10.1609/aaai.v34i04.6159.

[45] L. Zhang, T. Wen, et J. Shi, Deep Image Blending. 2019.

[46] T. Wang, J. Huan, et B. Li, « Data Dropout: Optimizing Training Data for Convolutional Neural Networks », arXiv:1809.00193 [cs], sept. 2018, Consulté le: mars 12, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/1809.00193

[47] « Image Pyramids with Python and OpenCV », PyImageSearch, mars 16, 2015. https://www.pyimagesearch.com/2015/03/16/image-pyramids-with-python-and-opencv/ (consulté le mars 05, 2021).

[48] Y. Pang, T. Wang, R. M. Anwer, F. S. Khan, et L. Shao, « Efficient Featurized Image Pyramid Network for Single Shot Detector », p. 9.

[49] Y. Wang, H. Xie, Z. Zha, Y. Tian, Z. Fu, et Y. Zhang, « R-Net: A Relationship Network for Efficient and Accurate Scene Text Detection », IEEE Transactions on Multimedia, p. 1-1, 2020, doi: 10.1109/TMM.2020.2995290.

[50] D. Zeng, F. Zhao, S. Ge, et W. Shen, « Fast cascade face detection with pyramid network », Pattern Recognition Letters, vol. 119, p. 180-186, mars 2019, doi: 10.1016/j.patrec.2018.05.024.

[51] A. Dhillon et G. K. Verma, « Convolutional neural network: a review of models, methodologies and applications to object detection », Prog Artif Intell, vol. 9, no 2, p. 85-112, juin 2020, doi: 10.1007/s13748-019-00203-0.

[52] F. Chollet, « Xception: Deep Learning With Depthwise Separable Convolutions », 2017, p. 1251-1258. Consulté le: mars 05, 2021. [En ligne]. Disponible sur: https://openaccess.thecvf.com/content_cvpr_2017/html/Chollet_Xception_Deep_Learning_CVPR_2017_paper.html

[53] Keras Applications. Keras, 2021. Consulté le: sept. 04, 2021. [En ligne]. Disponible sur: https://github.com/keras-team/keras-applications/blob/bc89834ed36935ab4a4994446e34ff81c0d8e1b7/keras_applications/xception.py

[54] D. E. King, « Dlib-ml: A Machine Learning Toolkit », p. 4.

[55] « ondyari/FaceForensics », GitHub. https://github.com/ondyari/FaceForensics (consulté le mars 12, 2021).
[56] S. Seferbekov, selimsef/dfdc_deepfake_challenge. 2021. Consulté le: déc. 21, 2021. [En ligne]. Disponible sur: https://github.com/selimsef/dfdc_deepfake_challenge
[57] K. Zhang, Z. Zhang, Z. Li, et Y. Qiao, « Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks », IEEE Signal Process. Lett., vol. 23, no 10, p. 1499-1503, oct. 2016, doi: 10.1109/LSP.2016.2603342.
[58] M. Tan et Q. Le, « EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks », in Proceedings of the 36th International Conference on Machine Learning, mai 2019, p. 6105-6114. Consulté le: déc. 21, 2021. [En ligne]. Disponible sur: https://proceedings.mlr.press/v97/tan19a.html
[59] H. Zhao, T. Wei, W. Zhou, W. Zhang, D. Chen, et N. Yu, « Multi-attentional Deepfake Detection », in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, juin 2021, p. 2185-2194. doi: 10.1109/CVPR46437.2021.00222.
[60] C. U. I. Hao, cuihaoleo/kaggle-dfdc. 2021. Consulté le: déc. 21, 2021. [En ligne]. Disponible sur: https://github.com/cuihaoleo/kaggle-dfdc
[61] Y.-J. Heo, Y.-J. Choi, Y.-W. Lee, et B.-G. Kim, « Deepfake Detection Scheme Based on Vision Transformer and Distillation », arXiv:2104.01353 [cs], avr. 2021, Consulté le: déc. 21, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/2104.01353
[62] K. He, X. Zhang, S. Ren, et J. Sun, « Deep Residual Learning for Image Recognition », in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, juin 2016, p. 770-778. doi: 10.1109/CVPR.2016.90.
[63] Y. Chen et al., « Drop an Octave: Reducing Spatial Redundancy in Convolutional Neural Networks With Octave Convolution », in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), oct. 2019, p. 3434-3443. doi: 10.1109/ICCV.2019.00353.
[64] James, DFDC - Our solution for Kaggle's Deep Fake Detection Challenge. 2021. Consulté le: déc. 21, 2021. [En ligne]. Disponible sur: https://github.com/jphdotam/DFDC