University of Amsterdam

Project Paper

Deepfake detection through PRNU and logistic regression analyses

July 5, 2020

Catherine de Weever
catherine.deweever@os3.nl
UvA ID: 12782750

ing. Sebastian Wilczek
sebastian.wilczek@os3.nl
UvA ID: 12837067

Assessor:
Prof. dr. ir. Cees de Laat
delaat@uva.nl
University of Amsterdam

Supervisor:
Prof. dr. ing. Zeno Geradts
z.geradts@nfi.nl
Netherlands Forensic Institute (NFI)

Course:
Research Project 2

Abstract

Deepfakes, as in creating a video of an impersonated target by replacing the face of an actor with the face of said target, have evolved in recent years to reach convincing levels of perceived authenticity. Generated forged imagery of faces has become so advanced that it is getting more complicated to differentiate a manipulated video from an original by watching it. If Deepfakes can not be told apart from original videos, undesirable effects, such as media distrust or forensic misevaluation, may occur. Therefore, Deepfake detection methods are required to ensure that one can mitigate potentially devastating effects of fake media. In this paper, we consider multiple approaches to distinguish between authentic and Deepfake videos. Among the approaches are Photo Response Non Uniformity (PRNU), visual artifact, and frequency domain analysis, three methods that do not make use of neural nets. We discovered that none of the methods show definite proof of Deepfake presence or absence, and even discovered detection evasion methods. As an alternative, we propose a conceptual system to authenticate original media instead of discovering Deepfake forgeries.

Keywords— Deepfake, Detection, PRNU, Visual Artifacts, Media Authenticity

1 Introduction

The process known as Deepfake is an emerging technology with the goal of replacing the likeness of a person with the portrait of someone else, both visually and audibly. A perfect Deepfake would be indistinguishable from an original representation of the impersonated target.
Especially the concept of replacing a person within a video file has recently gained traction. It is becoming an increasingly common practice [6] to take a video or a set of photos of a target's face, projecting them on another, existing video.
Deepfakes are seeing an increasing level of accessibility. While not entirely arbitrary to operate, software such as *Faceswap* [11] and *DeepFaceLab* [29] enable the creation of faked video frames with relative ease. Given that one can create Deepfakes with limited knowledge of the subject, a few concerning, if not malicious, use cases are enabled through Deepfake.
Publications exist of Deepfakes and their creation, impersonating famous people such as celebrities and politicians. A notable example is a video of the former President of the United States of America, Barack Obama, which was faked both visually and audibly [12]. Another example shows the actor Keanu Reeves actively committing a crime by breaking the perpetrator's neck during a store robbery [16]. While these publications mostly gained traction due to the public interest in the technology, the shared videos are nonetheless convincing in their content. Given that the means of production are free or open-source, one can assume that they may be used to spread misinformation on purpose or to commit fraud.
Furthermore, one widespread use case for Deepfakes are adult videos. Plenty of videos shared on the Internet contain pornographic content with the faces of celebrities superimposed on the original video [7]. Such videos already have a substantial impact on both the reputation and mental health of the target person. Initially, a user of the platform *Reddit* with the name "deepfakes" published the concept. The user later turned it into a full platform for sharing creations and resources. The name of this user is also the etymological origin of the name Deepfake.
While one could argue that celebrities are already dealing with similar issues due to their inherent public life, this technology could also apply to private people, given a sufficiently large set of data. There is plenty of room for criminal and civil misdemeanor, both in intentional fraud and false information spread as well as reputational and personal attacks.
Given the potential misuse of Deepfakes, the detection of such forgeries becomes a requirement for the future, considering the legal and social aspects of life. If Deepfakes become so advanced that they can not be distinguished from other media anymore, the presentation of a video introduces reasonable doubt. For example, a video of someone committing a crime could be argued against in a court of law since the person filmed might be faked. The same applies to social life and politics. If a video of a known person is shared, how can be ensured that the person said or did what is shown in the video?
In this research paper, we investigate various detection methods of Deepfakes. To do so, we consider the evaluation of Photo Response Non Uniformity (PRNU) patterns as well as the analysis of video artifacts and frequency domain. Furthermore, we consider methods that may assist in Deepfake detection. Lastly, we examine an alternative approach of authenticating original media instead of detecting previously synthesized pendants.

2 Research Questions

To research the topic, we defined research questions that are answered by the results of this research. The main question answered is the following:

> **How can a forged Deepfake video be differentiated from an authentic one, for forensic purposes?**

We divided the aforementioned question into the following subquestions:

1. What detection methods are already available?
2. Are these detection methods still applicable to modern Deepfakes?
3. If these methods are still applicable, can they be enhanced?
4. If these methods are not applicable anymore, what other approaches could be taken?

3 Related Work

There are different methods to detect Deepfakes. According to [25], we can divide these methods into three main categories: physical/physiological, signal-level, and data-driven. The methods that fall under the physical/physiological category detect the inconsistencies found in the physical/physiological aspects in Deepfake videos. For example, in [22], a new method was introduced that makes use of a physiological signal, such as eye blinking, while in [33], they made use of inconsistent head poses.

The methods that fall under the signal-level category utilize Deepfake detection algorithms that use signal-level artifacts introduced during the synthesis process. A few examples of these methods are explained in [23] and [35].
The methods that fall under the data-driven category make use of various types of Deep Neural Networks (DNNs) trained on real and Deepfake videos to capture specific artifacts. These methods are introduced in [2], [17] and [28].
The methods mentioned above make use of DNNs, but there are also non-neural network methods. In [26], a new method is discussed to detect Deepfakes by exploiting visual artifacts. This method makes use of the visual features of the eyes, teeth, and face to detect Deepfakes. It provides two variants: a small neural network and a logistic regression model. We made use of this method in our experiments. From their experiments, they were able to achieve an Area under the ROC Curve (AUC) score of 0.866. We explain the term AUC in *4 Background*.
Another category that is not discussed by [25] is detecting Deepfakes by looking at the image and video characteristics. One method that falls under this category is introduced in [19]. This method makes use of PRNU analysis to detect Deepfake videos. We also detail PRNU patterns in the following section. This analysis makes use of the mean normalized cross-correlation to distinguish Deepfakes from real authentic videos. We also used this method in our experiments.
Another method that falls under this category is introduced in [9], which is based on a classical frequency domain analysis followed by a basic classifier. They stated and demonstrated that their approach does not need a large amount of data to achieve reasonable accuracy. They achieved an accuracy of 100% using 20 annotated samples. Because of the high accuracy scores, we decided to use this method in our experiments.

4 Background

In this section, we provide certain background information, to put our work into the proper context, and to ensure that our methodology and experiments can be understood. We detail the creation process of Deepfakes, the usage of camera-unique patterns, and logistic regression.

4.1 Generative Adversarial Networks

Deepfake operates by making use of a Generative Adversarial Network (GAN). The term GAN and its underlying concept were first introduced by Ian Goodfellow, working for the Brain research team of Google LLC and the University of Montreal [14]. In its essence, a GAN contains two parts: a generator and a discriminator. These are sometimes also referenced as an encoder and a decoder. Both are neural networks or a similar technology that can generate and distinguish information based on a set of inputs. Figure 1 shows an overview of a basic GAN.

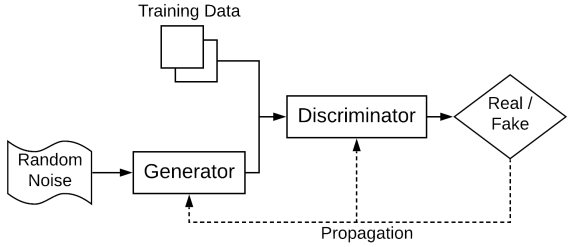

Figure 1: Overview of a Generative Adversarial Network

The job of the generator is to introduce new data to the data set, in this case, a video file frame containing the face or other to-be-faked likeness of a target. It makes use of random input vectors that are influenced by a weight value. The discriminator compares the newly generated frame to an original set of frames. In the case of Deepfake, those may be video frames of the target. GAN can also be used in other contexts, for example, to create pictures of people that do not exist [34]. In this case, the discriminator would compare the generated image to the images of living people. The discriminator calculates the likelihood of the generated picture being genuine, non-generated, and returns this loss value. This result propagates back to the generator, which can adjust its weight vectors accordingly until the discriminator is sufficiently satisfied. The exchange between the two adversarial networks creates continuous improvement in the creation of synthetic media.

4.2 PRNU Patterns

As previously mentioned, we made use of PRNU patterns to detect Deepfakes. One can consider the term PRNU as the fingerprint of a digital camera. PRNU patterns are fixed-noise patterns that the sensor of a camera creates when it converts the light captured during a photograph or a video recording to digital information. This happens because of the inhomogeneity present in the silicon used by the camera sensor. The slight variations of the material lead to responding variations in the quantum efficiency of each pixel of the sensor, meaning each pixel of the sensor converts photons to electrons slightly different. The result is a two-dimensional pattern with the same dimensions as the sensor of the capturing camera. [13]
Given a PRNU pattern of a camera, one can analyze the likelihood of an image originating from the given camera source. If one can extract the pattern from the image in question, one can compute how much the extracted pattern and the pattern from the considered camera correlate. The higher this value, the more likely the image originates from the evaluated camera. [5]

4.3 Logistic Regression

Logistic regression is a classification algorithm used to predict the probability of a target variable. It is a method used for binary classification problems. According to [8] *"The idea of logistic regression is to make linear regression produce probabilities."* In this case, these are class probabilities. This is done by taking the familiar linear model and giving it to a sigmoid function with the use of the following equation [15]:

$$y' = \frac{1}{1+e^{-(z)}}$$

where:

- y' is the output of the logistic regression model
- z is the log odds function: $b + w_1x_1 + w_2x_2 + ... + w_Nw_N$
  - w values are model's learned weights
  - b is the bias
  - x values are feature values.

The output value will be between 0 and 1. This value is the probability score. A classification threshold is defined to map this value to a binary category. One can use the metric accuracy to evaluate a classification model's predictions. Accuracy is the chunk of predictions that the model got correct. Accuracy is used when the best classification threshold is known. The metric (receiver operating characteristic curve (ROC) curve) is used when one evaluates the model across many different possible classification thresholds. Under this curve, there is the Area Under the ROC curve (AUC). The AUC gives an aggregate measure of the performance aggregated across all possible classification thresholds. AUC ranges from 0 to 1. An AUC score of 0.0 means that the models' predictions are 100% wrong, while an AUC score of 1.0 means that its predictions are 100% correct.

5 Approach and Methodology

In this section, we detail the approaches we have taken throughout the research. This includes the different considered detection methods as well as ways to evaluate their success.
Please note that most experiments detailed in this paper do not make use of neural networks. This decision is intentional and part of the scope of this project. Due to time and resource constraints, we were unable to execute proper experiments involving machine learning. Instead, we opted to consider any methodology that does not make use of neural networks for the most part, as detailed below. We only used neural networks and machine learning to create Deepfakes for our experiments and where neural network approaches were detailed as part of any related work. However, we only used these approaches if that related work proposed a similar solution that is not using neural networks.
Previous research into the topic of Deepfake detection showed that the analysis of PRNU patterns indicated the presence or absence of a Deepfake [19]. However, the creation of Deepfakes has advanced since the paper was published. Therefore, we reevaluated whether PRNU analysis is still applicable to modern Deepfakes.
To do so, we retrieved and created multiple Deepfakes to evaluate. We divided these Deepfakes into individual frames and extracted the PRNU pattern from each frame to perform cross-correlation computations with the extracted patterns of other files. If the files originated from the same source, there should be a high cross-correlation. Consequently, the computed value should be lower for files that one has tampered with using Deepfakes. We used this computed cross-correlation as an indication of detection success.
As mentioned in the related work section, another way to detect Deepfake is to make use of the visual features of the eyes, teeth, and face, as in visual artifacts[26]. This method is called the visual artifact analysis. We implemented this method and reevaluated whether we could use it with our retrieved and created data set. This method only accepts images as input, so we first needed to extract the frames from the videos. It has two variants for classification: A small multi-layer feed-forward neural network and a logistic regression model. If the AUC value is higher than 0.5, than the video is classified as a Deepfake. It is also possible to train the model with the features extracted from the input. We performed two experiments: one for the untrained model and the other one for the trained model. For the untrained model experiment, we only used the logistic regression model, and for the trained model, we used both of the models provided.
Analyzing the frequency domain of an image is another way to detect Deepfakes. This method is called the Frequency Domain Analysis [9]. The input for this method has some requirements. For our experiment, we first needed to ensure that our data set meets these requirements. We used the extracted frames, ran a face detection on them, and made them squared. This method required fake and original videos. The output of this method is an accuracy value. If the accuracy value of the video is higher than 0.5 than we classify it as a Deepfake.
We also looked into a way to evade the previously mentioned detection methods. For the detection methods that make use of a logistic regression model, we looked into an adversarial attack using Fast Gradient Sign Method (FGSM). This part of the research is abstract: our findings are theoretical, and they reference previous work. We discovered an evasion method for PRNU analysis during our experiments.
Given the results of the previous topics, we also considered the possibilities of authenticating original media. Given the time allocated for this project and the extensive scope of the topic, we decided to keep this part of the research abstract. All findings in this paper about media authentication are theoretical, referencing previous work in the same field.

6 Experiments

In this section, we introduce the experiments we conducted throughout the project. The structure of this section follows the different approaches detailed in *5 Approach and Methodology*. We define the setup of our experiments, where and how we retrieved data and information to experiment with, and how we executed each experiment. Each experiment aimed to either retrieve data for other experiments or to evaluate a method to detect Deepfakes.

6.1 Data Set Retrieval

To perform detection experiments, we required a sufficiently large set of Deepfakes. While we created some Deepfakes ourselves (see *6.2 Deepfake Creation*), the time allocated for the project did not allow us to create a large number. Therefore, we made use of two different Deepfake data sets: *FaceForensics++* [30] and *Celeb-DF* [24]. We decided to use these two sets because of their large amount of files available and their widespread usage in Deepfake detection research.
FaceForensics++ is a data set published by Google LLC and the Technical University of Munich. It consists of 1000 original videos containing front-facing faces of humans. Furthermore, it includes multiple faked videos, using different tamper methods, one of them being Deepfake. We had to request access to the data set through the Technical University of Munich, which we did before our experiments. We downloaded the data set to a server of the Security and Network Engineering course at the University of Amsterdam. We used extracts of the data set in our experiments.
The *Celeb-DF* data set was created by the University at Albany and the University of Chinese Academy of Sciences. The newest version, which we used in our experiments, contains 590 original videos, each with multiple Deepfakes, 5639 altered videos in total. Similarly to *FaceForensics++*, access needed to be requested, and we downloaded the complete set to a server, using extracts in our experiments.

6.2 Deepfake Creation

To properly check PRNU values, we required Deepfakes, of which we had the original video belonging to the faked version and an additional video shot on the same camera, ideally in a similar setting. Since the data sets available only provide the original versions of the video but not a second, unrelated video to check against, we had to create our own Deepfakes.
To create Deepfake videos, we made use of the cloud computing service Shadow by Blade [4]. The service gave us access to a virtual machine running Windows 10. The machine had access to a full `Intel Xeon E5-2678 v3` CPU, running at 2.50GHz with eight cores, and an `NVIDIA Quadro P5000` GPU with 16 GB GDDR5 VRAM. This hardware proved to be sufficient for creating Deepfakes.
To create Deepfakes, we made use of *DeepFaceLab* [29]. The software suite comes with the necessary tools to extract faces from video files, train the model required to generate Deepfakes, and render the results back to a video file.
For our experiments, we shot videos to be faked and checked. All created videos had a length of approximately 10 seconds, resulting in 220 to 260 frames, and were shot using the front-facing camera of a `Sony G8341 Xperia XZ1` smartphone. The videos to later check PRNU patterns against were shot in the same manner as the to-be-faked pendants. They had the same properties regarding length, encoding, and other comparable properties. The faces to be trained as the new faces in the Deepfake were extracted from longer videos of a different camera or taken from existing face sets.
DeepFaceLab offers two approaches to train the model, *Quick96* and *SAEHD*. The former uses fewer resources and takes fewer iterations to create a working Deepfake, at the expense of picture quality. Given the limited time of the project, we trained our models using *Quick96*. We trained the models for about 60000 to 65000 iterations, taking roughly three hours per Deepfake on average.
We merged the resulting video mask containing the face of the source video onto the original video for every Deepfake. We did this by using *DeepFaceLab*. We did not use any external video editing software. The mask was color-graded and blurred to fit the original video before merging.

6.3 PRNU Analysis

To analyze the PRNU patterns of the Deepfakes retrieved and recorded, we made use of an application called *PRNU Compare*. This software was developed by the Netherlands Forensic Institute (NFI) [27] and is used by law enforcement agencies to compare videos and photos, deriving a confidence value if two pieces of media originate from the same camera.
We used the software by importing the video files retrieved and created. *PRNU Compare* automatically extracts the frames of a video file, computing the average PRNU pattern over the set of extracted frames.
We then computed the normalized cross-correlation values between the original and the faked video and the check videos, if available. *PRNU Compare* is also able to compute a *Peak to Correlation Energy* value. This is another form of correlation indication, achieving the same goal as computing the cross-correlation. Since the computation of normalized cross-correlation takes marginally less time, we decided to use these values for comparisons.
As mentioned, we compared the PRNU pattern of the original video to the patterns of both the fake pendant and, where applicable, other videos shot using the same camera. We used the computed values to draw our conclusions about PRNU analysis accordingly.

6.4 Visual Artifacts Analysis

Another way to detect Deepfakes is to analyze visual artifacts. For our research, we used the method, abbreviated as VA, introduced in [26]. This method captures visual artifacts in the eyes, teeth, and facial contours of the generated or manipulated face. For example, this considers teeth represented as a white blob. This method has two variants: VA-MLP, which uses a small multi-layer feed-forward neural network for classification, and VA-Logreg, which uses a simple logistic regression model for classification.
It can also detect images from generated faces, Deepfakes, and Face2Face. For our research, we only made use of its detection pipeline for Deepfakes. This pipeline consists of, first, detecting and segmenting the faces from the frames, second, extracting the features of the eyes and teeth, and last, classifying these features with the previously mentioned models, VA-MLP and VA-Logreg. This method can extract the features from the input and train the models with these features, which we discuss later in this section.
As mentioned before, the input for this method needs to be images or frames. So we first extracted the frames from the Deepfakes and original videos. We did this for ten videos from the *FaceForensics++* and *Celeb-DF* data set and for one self-made video that consists of a Deepfake and an original in the same file.
Then, we processed each frame of the video with the Deepfake pipeline mentioned before. The output is a table that consists of four columns: *Filename*, *Score-MLP*, *Score-LogReg*, and *Valid*. The *Filename* column contains the filenames of the frames while the *Score-MLP* and *Score-LogReg* columns contain the classification score (AUC) of each frame. They can range from 0 to 1. If the frame is from a Deepfake, the score should be higher than 0.5, and the score should be lower when the frame is from an original video. The column *Valid* contains 0 and 1 values that indicate if the face detection and segmentation were successful or not. After each frame is processed, we calculated the average score for each video, excluding the invalid frames.
For our research, we did the above with untrained and trained models. With untrained models, we mean that we calculated the scores with the default or original classifiers. As mentioned before, the features from frames can be extracted and used to train the models to create new classifiers for each video. So with trained models, we mean that the scores were calculated with the newly created classifiers using the extracted features. Before we were able to do that, we needed to write a script that extracts the ground-truth labels (*Filename* and *Valid* column). So we created a script (see *Appendix A*) in Python that extracts these columns and saves them to a new CSV file called *labels*. We found out that the part for creating logistic regression classifiers was not working when we ran the script to create the new classifiers. So we modified the original code to create new classifiers for the logistic regression model (see *Appendix B*).

6.5 Frequency Domain Analysis

We looked into another way to detect Deepfakes, which is analyzing the domain frequency of an image. For our research, we

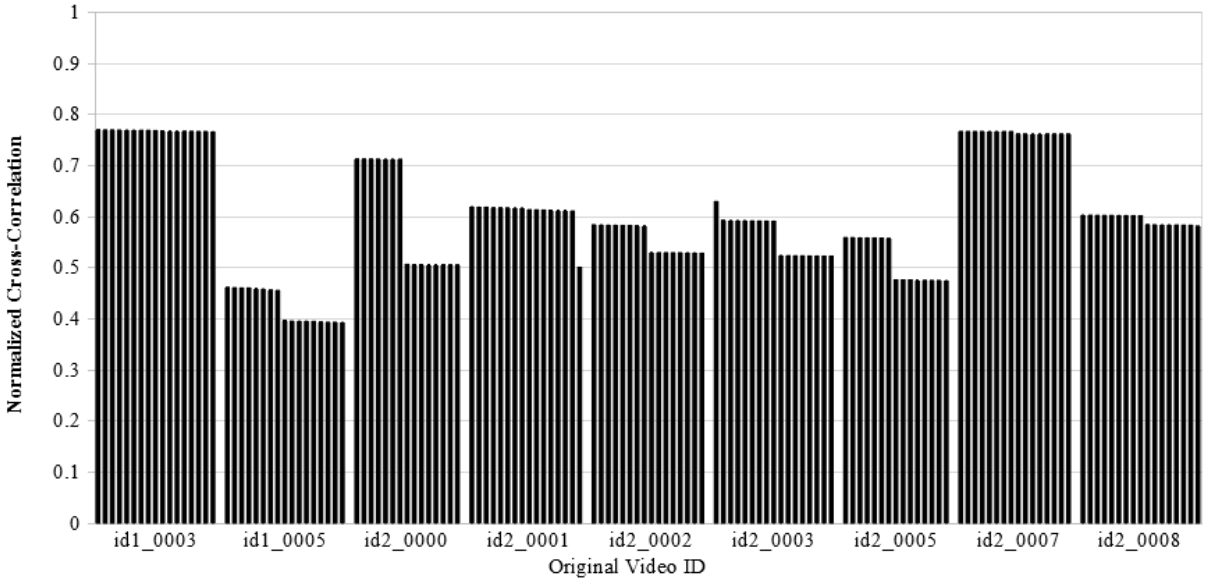

Figure 2: Normalized Cross Correlation of PRNU of *Celeb-DF* data set extract, with cross-correlation values shown for each Deepfake per original video

made use of the method introduced in [9]. The pipeline of this method contains two main blocks: the pre-processing block and the training block. In the pre-processing block, the input is transformed into a convenient domain by performing the following steps. First, the Discrete Fourier Transform (DFT) is applied to the image where the output is a 2D Amplitude Spectrum. The Azimuthal averaging is applied where the output is a 1D Amplitude Spectrum (1D representation of the FFT power spectrum). One can see the last step as compression. In the training block, these new transformed features are used to determine if a face is fake or not. This method makes use of two classification algorithms, Logistic Regression and Support Vector Machines (SVMs). We only used the Logistic Regression algorithm for better comparison to the previous method for our experiments.
Both real and fake images are needed to train this classifier. For this method, one needs the same amount of real and fake images, so there needs to be a balance. So for our experiment, we used 36 videos (eighteen real and eighteen fake) from the *FaceForensics++* data set and ten videos (five real and five fake) from the *Celeb-DF* data set. This method only accepts frames or images as input, so we used the frames that we have extracted for the previous method. Before we could use these images as input, they first needed to meet the following requirements: the (fake) face needs to be the dominant part of the input, and the images need to be squared. For our images to meet the requirements, we ran a face detection on them and made them square images using the program *autocrop* [20].
After that, we inserted them into the pipeline. The output of this method is the average classification rate per video (accuracy). The higher the average, the better, while there is a balance in the features between the fake and real video. If the average is higher than 0.5, we classified the video as fake.

7 Results

In this section, we detail the results we uncovered throughout this project. The results contain the measurable values of the experiments mentioned above and the theoretical research into further related topics, such as the authentication of original media.
Structurally, this section follows the same topics as the previously detailed methodology and experiments. We discuss the results of our PRNU analysis and the results relating to visual artifact and frequency domain analysis. We then define our results regarding the evasion of Deepfake detection techniques and our findings in the area of media authentication.

7.1 PRNU Analysis

As previously mentioned, we imported the available Deepfakes and original videos into *PRNU Compare* for comparison. Given that the computation of PRNU patterns and their cross-correlation takes time, we opted to use only extracts of the available data sets.
The data set of *Celeb-DF* contains multiple Deepfakes per original video. However, some of the Deepfakes have slightly altered aspect ratios, making a direct cross-correlation computation impossible. We went through the set and extracted the originals and Deepfakes that have the same aspect ratio until we had a reasonably-sized subset. We then computed the normalized cross-correlation for each original video against its Deepfakes. The results can be seen in Figure 2. The graph shows all computed cross-correlation values per original video in relation to multiple Deepfake videos derived from the original.
Each bar represents the cross-correlation of a given original video against one of its Deepfakes. As can be seen, the cross-correlation values range in-between 0.4 and 0.78. These are inconclusive values. Because there are no other original videos taken with the same camera as the original to compare against, it is impossible to deem any of the analyzed videos as an original or a fake.
We observed similar behavior in the *FaceForensics++* data set. This set only contains one Deepfake per original video. The re-

sults can be seen in Figure 3.

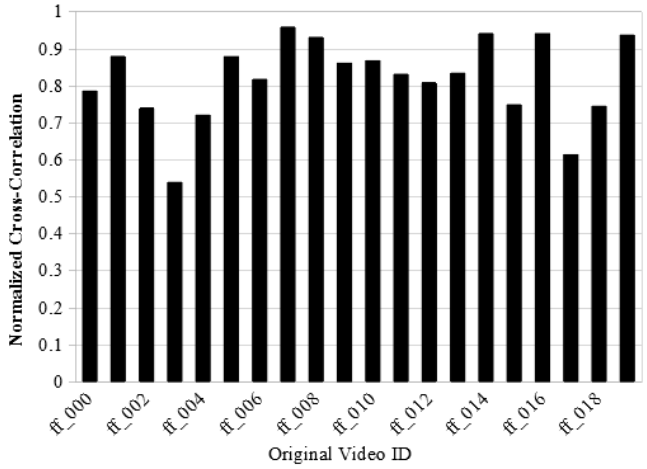

Figure 3: Normalized Cross Correlation of PRNU of *FaceForensics++* data set extract

Similar to the results of the *Celeb-DF* sets, the resulting values range in the upper half, between 0.52 and 0.95 percent. Especially the higher values make it impossible to deem a video as a fake, given that a higher cross-correlation indicates the likely equivalence of the video's origin.

However, the difference between the cross-correlation of a faked video and a video originating from the same camera against an original video might indicate Deepfakes in a file. As mentioned, neither the *Celeb-DF* nor the *FaceForensics++* data set contain such check files. Therefore, we analyzed Deepfakes we created ourselves, together with such check videos. In Figure 4, one can see the cross-correlation of the Deepfake videos and multiple check videos against two original videos.

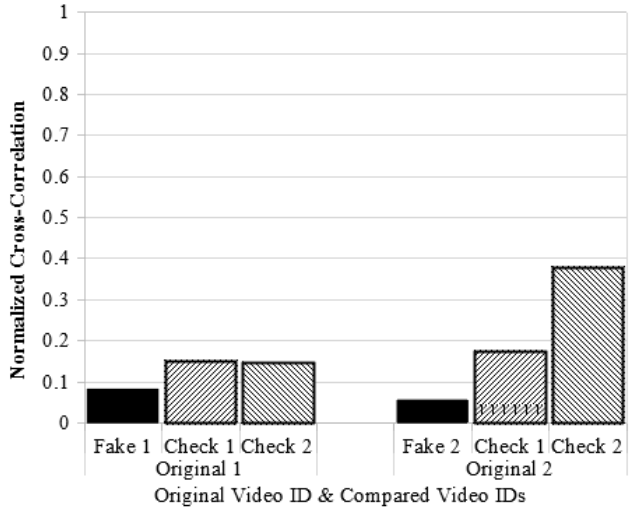

Figure 4: Normalized Cross Correlation of PRNU of created Deepfakes

While a difference between the Deepfake and check videos can be observed, all computed cross-correlation values are considerably lower than the results retrieved from the available data sets. Furthermore, the difference between the cross-correlation of the fake video and that of the check videos is only marginal. Again, we could not extract conclusive evidence about the authenticity of a video.

While analyzing the lower cross-correlation values, we realized that image stabilization influenced our results. We filmed the videos created up to this point while holding the smartphone in-hand. The slight movement of the hand triggered the built-in stabilization techniques that are increasingly common with smartphone cameras. Moving the camera blurs the resulting PRNU pattern. This blur results in cross-correlations between images to be lower, even between two images or videos recorded right after each other using the same camera.

We created another Deepfake to evaluate this theory. We mounted the smartphone to a tripod on stable ground. The recording was triggered remotely to prevent any slight collision and resulting blur when starting the recording. Other than that, we created the Deepfake in the same manner as the previous ones. The result of analyzing this new Deepfake against the new and existing check videos is seen in Figure 5.

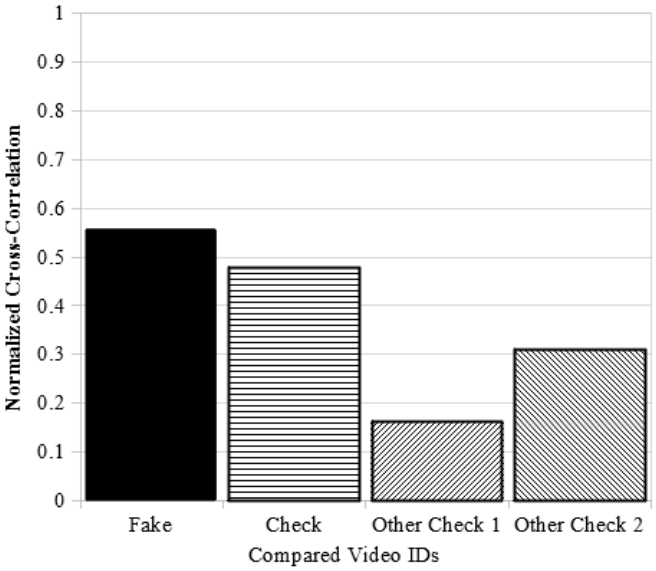

Figure 5: Normalized Cross Correlation of PRNU of created Deepfake with stabilization

The results show that the values are indeed higher if the camera is stable during recording, resulting in a more precise PRNU pattern. However, the results also show that the lower cross-correlation values of the Deepfakes previously analyzed were a coincidence. In this particular case, the cross-correlation of Deepfake and original video is even higher than the correlation between the original and a check video.

We attempted to further refine the experiments by comparing PRNU patterns extracted from videos that we previously cropped to the face of the depicted subject. Since we used *DeepFaceLab* for the creation of our own Deepfakes, the aligned and cropped images of both original and faked faces were available to us. In the same manner, we extracted the faces from the check video. We computed the PRNU patterns of all three, shown in Figure 6, 7 and 8.

Figure 6: PRNU Pattern of Original Video

Figure 7: PRNU Pattern of Deepfake Video

Figure 8: PRNU Pattern of Check Video

The cross-correlation values of the patterns depicted above are below 6%. There is only minimal similarity given PRNU values for the faked as well as another original video if we crop the videos to faces. The main issue with this approach is that the face in the check video is not necessarily at the same spot as in the original and Deepfake videos. Therefore, an extract of the PRNU pattern at a different part of the whole image is slightly distorted when compared to the extract of the original and the Deepfake, which are at the same place, given that only the face changes when comparing the videos, not the face position. Using this approach, therefore, only shows if the two faces are the same. This may be valuable if the changes are subtle, but this does not apply to full Deepfake imagery.
Overall, none of the PRNU analysis resulted in a definite proof of Deepfake or authenticity.

7.2 Visual Artifact Analysis

As mentioned in the previous section, the frames from the Deepfake and original videos were processed to calculate the classification scores. The number of frames for each video ranges from 300 to 800 frames rounded up. We calculated the average for each video to get an overall score. The videos classified as Deepfake need to have a score higher than 0.5, and the videos classified as an original need to have a lower score. We did not include all frames in the calculation, only the ones classified as valid. Occasionally, almost no valid frames were recognized, even just one in a particular experiment. That influences the visible results, and might also indicate that there is a forgery (less valid frames with fakes). However, it also happens with originals occasionally, for example, *CDF 0001 Original* only had one valid frame.

7.2.1 Untrained models

Three of the five original videos from the *FaceForensics++* data set have a score of 0.5 and lower, as can be seen in Figure 9. These three videos were classified as original. *FF 002 Original* was classified the best with a score of 0, rounded down. *FF 001 Original* was classified the worst with a score of 0.94, meaning it classified as a Deepfake, despite being original media.

For the Deepfake videos, four of the five have a score higher than 0.5, meaning that they classified as Deepfakes (see Figure 9. *FF 002 Fake* has a score of 0.95, so it was classified the best. Though, the gap between *FF 002 Fake* and the other three (*FF 000 Fake*, *FF 001 Fake* and *FF 004 Fake*) is small. *FF 003 Fake* scored the worst and is the only Deepfake video classified as original.

In Figure 9, it is depicted that four of the five original videos from the *Celeb-DF* data set are classified as original. Though they all have a score of 0.5 or lower, the range contains values higher than 0.5. It means that these videos contain frames that had a score higher than 0.5. *CDF 0000 Original* has the lowest score, lower than 0.5 with the range of standard deviation included, so this video is classified the best. *CDF 0001 Original* is classified the worst with a score of 1, meaning that the pipeline classified this original video as a Deepfake with no margin for error.

Four of the five Deepfake videos were classified as a Deepfake. *CDF 0003 Fake* has a score of 0.51 and the rest a score of 0.8 and higher (see Figure 9). *CDF 0003 Fake* contains frames with a score of up to 0.92, while the others contain frames scoring higher than 1.0, meaning that these frames classified as a Deepfake with no margin for error. *CDF 0001 Fake* classified the best by having the highest score, while *CDF 0000 Fake* classified the worst by having the lowest score, although it contains frames that classified as a Deepfake.

To depict the score difference between the frames from a Deepfake and the ones that are original, we created a video that contains frames from the Deepfake and the source video. We processed these frames and retrieved the following result depicted in Figure 10. For frame number 183, a drop to 0 can be seen where the frames start being from the original video. The other drops to 0.5 or less beforehand depict the inconsistencies

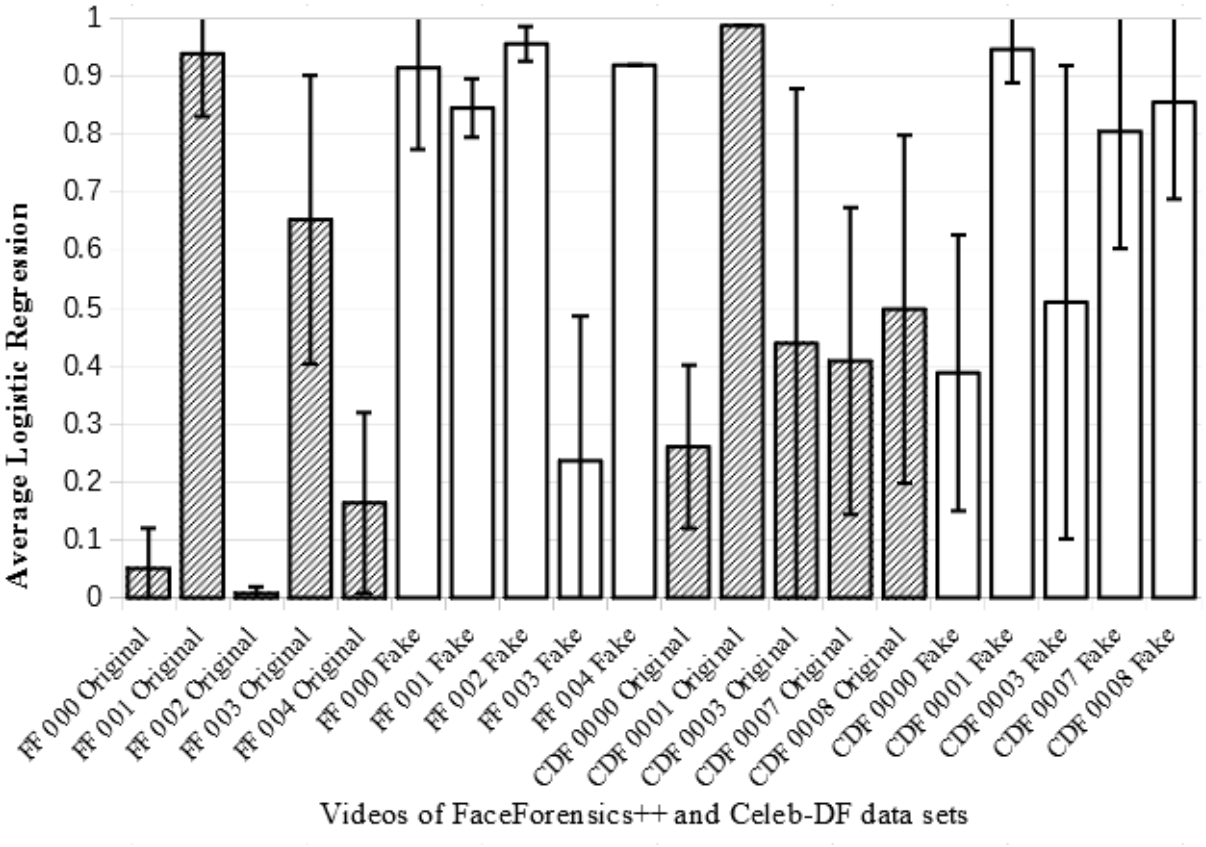

Figure 9: Average logistic regression per video of *FaceForensics++* and *Celeb-DF* data set extracts

in the results.

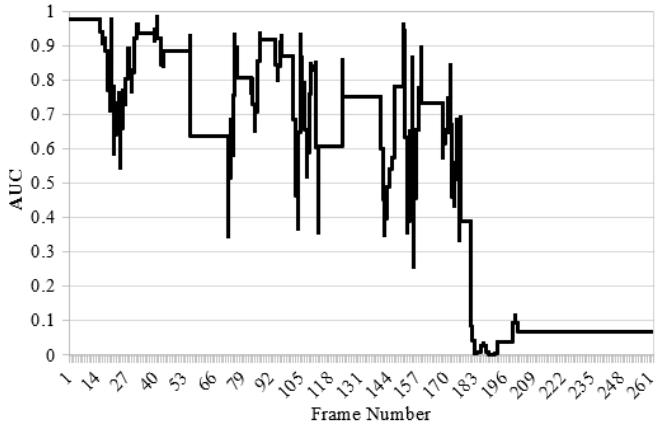

Figure 10: Logistic regression per frame of partial Deepfake video

7.2.2 Trained models

As mentioned in the previous section, we trained the VA-Logreg model using the features from the videos. We trained the model one time. For the original videos both from *FaceForensics++* and *Celeb-DF*, all have a score of 1.0 as seen in Figure 11. This also applies to the Deepfake videos. So using a trained logistic regression model improves the classification of Deepfakes but worsens the classification of the original files. After receiving these ambiguous results, we also decided to train the VA-MLP model. The original and fake videos from both *FaceForensics++* and *Celeb-DF* all have a score of 0, rounded down, meaning that they all classified as original. So using a trained MLP model improves the classification of the original but worsens the classification of Deepfakes. It is the opposite of the trained version of the logistic regression model. We also used both of the trained models with the video that contains frames from a Deepfake and an original file and got the following results. As expected, the score using MLP is 0, rounded down, whereas the score using the logistic regression model is 1. If we compare the results with the untrained model results, we observe that there is no drop, making the untrained model more reliable, even though it has some inconsistencies.

7.3 Frequency Domain Analysis

Because of the ambiguous results from the previous method, we also decided to look into the frequency domain analysis that also uses a logistic regression model, as mentioned before, the input consists of images from real and Deepfake videos. This method already calculates the average, so we did not need to do it ourselves. A note to point out is that after running a face detection on the images, only a few images were rejected (no face was detected). In this method, a lot more images were valid than for the visual artifact analysis method. This method also compares the real images with fake images, so there is no separate average for the real images.
For the *FaceForensics++* data set, all videos are classified as a Deepfake as seen in Figure 12. *FF 018* had the lowest score, while *FF 011* had a 100% accuracy. Ten of the 19 videos had an accuracy of 70% and above, which leads to a high overall accuracy for this data set.

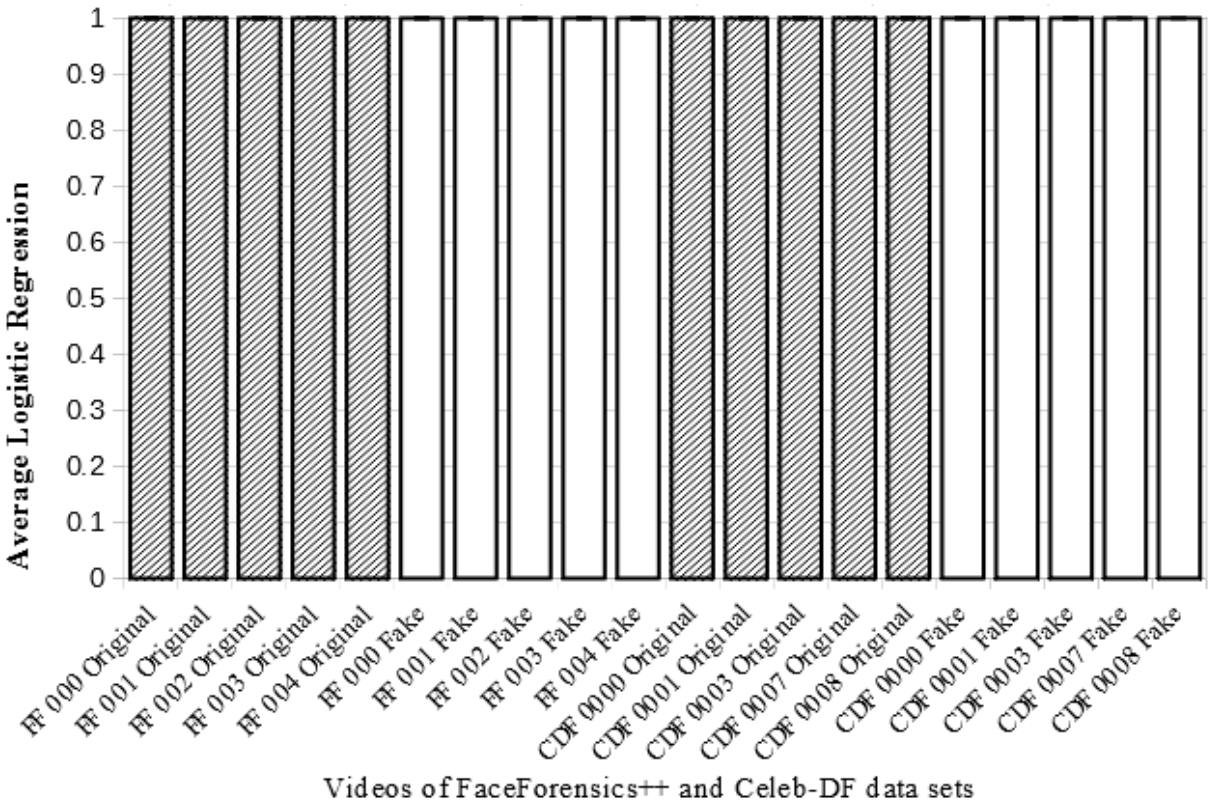

Figure 11: Average logistic regression using trained approach per video of *FaceForensics++* and *Celeb-DF* data set extracts

For the *Celeb-DF* data set, it is a different story. As depicted in Figure 12 all videos have a score of 1. It may seem to have positive results, but it is not. This phenomenon occurs when there is no balance between the features of fake and real images. In our case, there was no spectrum found on the fake images. We retrieved this information by plotting the spectra of these videos. It returns a score of 1 because there is no fake data to compare. So this method did not work on the *Celeb-DF* data set. We discuss the reason for this in *9 Discussion*.

7.4 Detection Evasion

A way to bypass the detection method is to modify Deepfakes adversarially. This way, the detection system (classifier) can be fooled and classify the Deepfake as the original by reducing the AUC value. There are many different attacks to achieve this. These attacks can perform in a white-box and a black-box approach. For a white-box approach, the attacker has full access to the detection system, including the Deepfake pipeline, the architecture, and the classification model's parameters. For a black-box approach, the opposite applies: the attacker does not have full or limited access to the classification infrastructure and its parameters.

In our case, an attack that can be performed is the FGSM. FGSM is a white-box attack which ensures misclassification. According to [18], "FGSM is to add the noise whose direction is the same as the gradient of the cost function with respect to the data." It is done in one single step, and uses the following equation:

$$X_{adversarial} = X + \epsilon.sign(\nabla_x J(X, Y)),$$

where X = *original input image*
$X_{adversarial}$ = *adversarial image*
∇ = *gradient of cost function with respect to* X
J = *Loss function*
Y = *model output for* X (*input label*)
ϵ = *parameter to scale noise* (*small number*)

First, perturbations (random noise) are created with the *sign* function. This is done by using the gradients of the loss with respect to the input image. Then the perturbations are added to the original image, X. The output is the image with the random noise added. An example is depicted in Figure 13. In [18], where they use the logistic regression model, the error rate grew from 2.9% to 7.6% when epsilon has a value of 0.2 and 22.6% when epsilon has the value 0.5. Because the visual artifact analysis and the frequency domain analysis use the logistic regression model, we can conclude that FGSM can also be applied to increase this model's error rate.

As previously mentioned, we observed that any internal camera stabilization influences the results of PRNU analysis negatively. By moving the camera, the resulting video's extracted pattern becomes blurry, even with slight camera motions. The blurred or distorted pattern would result in a lower cross-correlation score when comparing against another pattern, even if one extracted the said pattern from the same camera. In other words, if the camera moves while recording the destination video to have a face edited in, PRNU analysis is less likely to detect a Deepfake. Both original and Deepfake media will have lower cross-correlation scores, making it impossible to differentiate. Therefore, camera movements can be seen as evasions methods to prevent Deepfake detection through PRNU analysis.

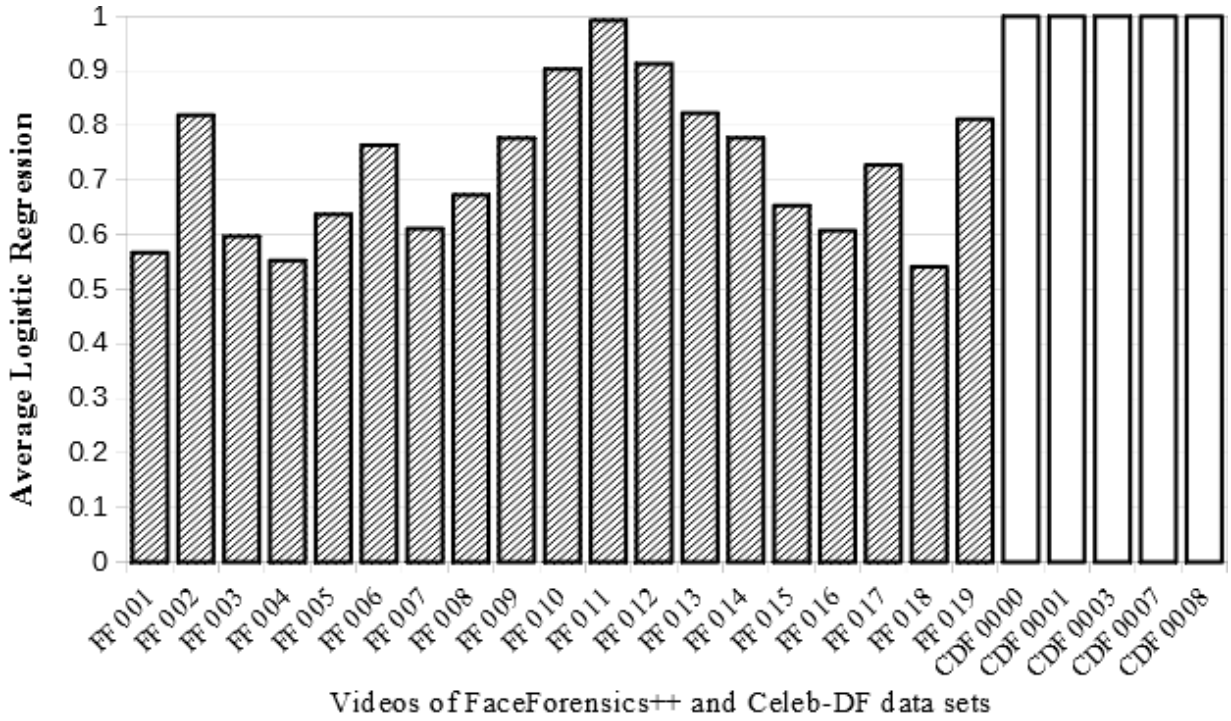

Figure 12: Average logistic regression using frequency domain approach per video of *FaceForensics++* and *Celeb-DF* data set extracts

7.5 Media Authentication

The question of how to differentiate between original and synthetic media still stands since we could not determine any definitive way to detect Deepfakes in video files. Even if one could propose a highly successful way of detecting Deepfakes, the result would be of limited value. As soon as the detection method becomes public knowledge, one can enhance the discriminator of a Deepfake-generating GAN with said detection method. Over time, Deepfakes would take these results into account, learning to avoid generating images that one can detect through these means.

In other words, Deepfake detection may prove to be dysfunctional in the future. Instead of attempting to detect Deepfakes, an alternative approach would be to recognize the original files. The primary issue caused by Deepfakes is the impact on the person falsely depicted. If the source of the video is uncertain and perhaps not trusted, one could mitigate parts of this issue. To acquire the source of a video file, said files require provenance information. [21]

Provenance is the record of origin and ownership of a product. Historically, the term stems from the art world but can apply to other items, such as video files. Provenance for digital media includes the source of production, the chain of modification done to the original file, and information about the publisher or owner.

If provenance were known for every authentic piece of digital media, Deepfakes could easily be spotted because they are missing provenance, or the provenance is pointing to an unknown or non-trusted source. However, there is only limited literature and projects available in this field. *"The problem of seeking information provenance has received little attention in comparison with its counterpart, the study of information propagation."* [3]

The problem lies in how one propagates the provenance information. Theoretically, provenance could be created directly at the source. Organizations, such as news and governmental offices, could sign files that they deem authentic using a form of public key infrastructure. However, every reading party, such as social networks or other organizations, would have to know the public key. They would have to retrieve the entire file to be able to validate the signature.

Furthermore, many organizations modify media while warranting authenticity. For instance, news organizations regularly encode media files in different formats or add other forms of visual and audible information to a video, such as text banners or backing commentaries. While not the same file, the originally depicted videography is nonetheless authentic. In this case, the file would have to be resigned, and provenance information would be lost since there is no reference to the original file. This approach is also not applicable to streaming media since the entire file needs to be available at the signature-checking party.

To mitigate these issues, Microsoft Corporation proposed a system called *AMP* [10]. The system is designed to provide and interact with provenance information on the Internet. The system makes use of authentication and watermarking. To authenticate media, one can create manifests containing hashes of the original file and the metadata of a publisher. For streamable media, hashes are created for file chunks.

Furthermore, these manifests can contain pointers to other source files. If a video such as a news report uses multiple recordings, the report's manifest can point to these sources, ensuring a chain of provenance information. These manifests are stored on a public blockchain ledger, to be immutable and identifiable.

The goal of AMP is to create a system where every piece of digital media carries a manifest as well. However, since current Internet media systems are not equipped to do so yet, they also

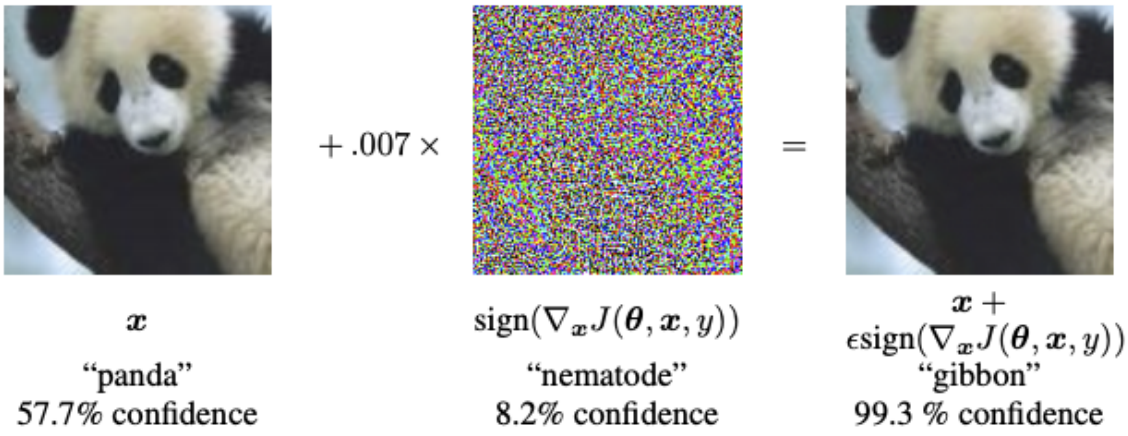

Figure 13: An example of how the Adversarial image is created using FGSM [31]

provide a queryable database to check manifests. Additionally, spread-spectrum watermarks may be inserted. These watermarks add noise to the file that is imperceptible to the human eye. This watermark carries an identifier for a manifest, for a user to later retrieve said manifest, enabling authentication.
AMP is not the only system attempting to introduce provenance to consumers. The research and development group of The New York Times Company introduced *The News Provenance Project* [32]. Like AMP, it uses blockchain to store and retrieve information about the origin of media. In extent, the project is also concerned with the consumer itself. They are researching how consumers evaluate media and their information of origin.
Furthermore, considerations such as if media without provenance information can still be trusted are taken into account. Research such as this is emerging broadly in the context of digital media. Other projects, such as the *Content Authenticity Initiative* by Adobe Inc. [1], are considering similar aspects, and will likely continue to do so in the future.
Given the proper implementation of a system such as AMP and enhancement of video players available, such as the ones on social networks and websites, it could be possible to introduce provenance information to the user. Given this information, a consumer can draw an educated conclusion about whether to trust the displayed media.

8 Conclusion

In the following paragraphs, we detail the conclusions we can draw from the experiments mentioned above and their results. Structurally, we will follow the research questions mentioned at the beginning of this paper.
Our research into related work on the topic of Deepfake detection returned many different approaches. One thing most approaches had in common was their use of neural networks. Due to the scope of this project, we did not evaluate these time-consuming methods. Instead, we focused on approaches that did not make use of machine learning primarily. Specifically, we focused on the usage of PRNU patterns and logistic regression analysis of visual artifacts and frequency domain, which have been used for Deepfake detection in other research papers. We only used neural networks to create Deepfakes ourselves and where machine learning was proposed as a subset of a broader methodology proposal, for example, while doing visual artifact analysis.
The analysis of PRNU patterns has shown drastically varying results. Cross-correlation values shifted considerably between different Deepfake and original videos, with no clear indication of an analyzed video being fake or authentic. Furthermore, we have seen cross-correlation values being higher for Deepfake videos than for other original media. We could not determine a cut-off value for a proper indication of Deepfake presence.
Additionally, PRNU analysis also has the significant drawback of requiring comparison media. To determine a cross-correlation value in the first place, the PRNU pattern of the supposed original camera needs to be present, for example, by analyzing another known piece of media. While this might be possible in some cases, for instance, in the case of statically placed sources such as surveillance cameras, this leaves out circumstances where the original camera may not be retrievable, such as private smartphones or news broadcasts.
The analysis of visual artifacts has shown varying results when using the non-trained model for both data sets. Most Deepfake videos were classified as a Deepfake; however, a considerable amount of frames from these videos were classified as invalid and not included in the calculation of the AUC score. This makes the visual artifact analysis method unreliable though an advantage of this method is that it only needs Deepfake videos for classification. The results when using the trained model were ambiguous, which led us to the conclusion that this part of this method did not work properly.
For the analysis of the frequency domain, the results differ depending on the data set used. For the *FaceForensics++* data set, it had a high accuracy score, meaning that it classified the Deepfake videos as a Deepfake. While for the *Celeb-DF* data set, it returned a biased score, meaning that something did not work properly. A drawback of this method that it needs both original and Deepfake videos, unlike the previous method.
As can be seen for the results of the PRNU, visual artifact, and frequency domain analysis, none returned results that can accurately indicate Deepfake presence or absence with a high probability of success. Throughout our experiments, we attempted multiple approaches to improve these results. Regarding PRNU analysis, we considered that the stabilization of a camera when creating a Deepfake might influence the com-

puted results. We also attempted to compute PRNU cross-correlation on extracts of the entire pattern, such as the part of the image containing a face. For visual artifact analysis, we tried to repair the code that trains the model. However, none of the improvements we attempted resulted in considerably more positive results.

In attempting to improve the detection methods, we uncovered some possibilities to evade Deepfake detection. Moving the camera when creating a Deepfake lowers cross-correlation scores during PRNU analysis, making comparisons of original and fake media less successful. Because the visual artifact analysis and the frequency domain analysis method use a logistic regression model, an adversarial attack using FGSM can be performed. Overall, all the mentioned approaches further disprove the possibilities of using the considered methodology for Deepfake detection. Aside from minor technical improvements and streamlining the analysis process through automation, we did not find a noticeable way to improve existing detection methods.

As previously mentioned, we were unable to pinpoint any reliable method to detect or even indicate Deepfake presence. Please note that this research focuses on methods that do not use neural networks. Therefore, we can not claim that a reliable method does not exist.

However, even if a reliable detection method would exist, its applicability would be of limited potential. As mentioned in *4 Background*, Deepfakes make use of GAN. Therefore, a discriminator is used to evaluate the generated results. Any system that could rate the likelihood of Deepfake presence may also be used as part of a GAN discriminator. Such a detection method would actively support the improvement of Deepfake creation since the discriminator could remove imagery that may be susceptible to detection.

Therefore, we do believe that alternative approaches are required. We detailed one such approach before. Authentication of original media may be a step to introducing provenance to digital imagery. Being able to claim authenticity is becoming ever more critical, especially for sources such as news outlets and governmental institutions.

Given the existing prototype implementation of AMP, it might be possible to deploy such a media authentication service on a large scale. However, more research is required beforehand. This research is twofold. First, the design and implementation of such a system need to be detailed. If a system is to be available to authenticate and sign original media, ideally worldwide, much careful thought has to be put into the system design.

Secondly, deploying such a system carries a heavy load of ethical concerns. One of them is the question as to who should be responsible for such an authentication system. Should it be governmental organizations or private ones? Furthermore, who should be able to sign media, and who is empowering them to do so?

Assuming such a system was in place, Deepfakes could be distinguished from original media, because they would not be signed by the person or responsible organization displayed. Anyhow, only trusting signed media carries its own set of problems and concerns. Especially in a legal environment, only trusting certain types of media may be reprehensible. For example, the recently emerging riots in the United States of America were caused by a video taken by a bystander of crime committed by a police officer. Such a video would not be signed in this case, or at least not signed by a known, trusted source. Considering the implications for the future, it may be possible that such media may not hold up in court, or the public may dismiss it because it is marked as not trustworthy.

Additionally, signing media may violate the privacy of the source. If all media were to be signed automatically for the sake of provenance, the source becomes easily identifiable. This mechanism may help in criminal cases, but it could also endanger sources that rely on anonymity, such as corporate and governmental whistleblowers.

Both the absence and presence of a method to identify original media or Deepfakes carry both technological and ethical problems that are yet to be addressed. Further research is warranted in all the mentioned topics. Please see *10 Future Work* for further information.

To summarize, plenty of approaches for Deepfake detection exist, making use of both neural nets and other means of evaluation. However, the approaches considered in this research do not exhibit results that can lead to the detection of Deepfakes and original videos with sufficient probability. Other approaches may be able to do so, but their existence may also improve the creation of Deepfakes. Enhancing them or creating even better approaches could be considered reprehensible. Therefore, alternative solutions, such as authentication of original media, are required. However, new alternatives carry their own set of issues and challenges, warranting future research.

9 Discussion

In this section, we discuss the progress and results of this research critically. We evaluate which parts of this research one could have improved upon and how one can improve them in the future.

To start, we did not have fully sufficient data sets to perform PRNU analysis. As previously mentioned, our experiments require the presence of a check video, recorded with the same camera. The *FaceForensics++* and *Celeb-DF* data sets do not include such videos. Therefore, we had to create our own, together with the accompanying Deepfakes. The process of creating Deepfakes requires time and resources, both of which were limited in this project.

One can say the same about the experiments in general. More measurements, perhaps taking more data sets into account, could have been taken in longer project time. Perhaps the experiments could have been improved with a degree of automation. In the present setups of our experiments, PRNU patterns have to be extracted and compared one-by-one, due to the software used. Similarly, one could have streamlined the evaluation of logistic regression and visual artifacts using a more automated approach. We can not claim that this would have led to more measurements since we can not estimate how long the creation of such automation would have taken.

For the visual artifact analysis, better data sets could have been used, as a considerable amount of the frames extracted from these videos were classified as invalid by this method. Additionally, the modified code for the training model can also be considered faulty.

As mentioned before, the frequency domain analysis did not work for the *Celeb-DF* data set. We found out that a class imbalance occurred by plotting the spectra of the videos. When this phenomenon occurs, the accuracy metric breaks down. Class-imbalanced problems are one of the critical flaws of accuracy. So we can conclude that the metric accuracy is an unreliable metric to use for this method. As mentioned before, a class imbalance occurred because no spectrum was found on

the fake images. A reason for this is that these frames may have been compressed to a large extend, being videos retrieved from YouTube. The creators of this method also stated that their approach might not work on videos or images that have been compressed to a large extend. [9] Two solutions for these problems could be to use the metric "precision and recall" instead, or, if accuracy is preferred, to use another data set that is not compressed.

Furthermore, some parts of this research are only taking a theoretical, abstract approach. We only checked the evasion of Deepfake detection in relation to related work. Experiments about the success of evasion using the mentioned techniques could have provided valuable information. Similarly, the concept of original media authentication and the surrounding systems is only detailed based on previous work. It would be possible to create a prototype of such a system or evaluate existing approaches, such as AMP. However, given the scope of this project, analyzing, designing, and implementing such a system should be considered future work, as shown in the following section.

10 Future Work

One can expand upon this research in a multitude of ways. First, one can extend the measurements taken during this project. Due to our limited time scope, we were only able to consider a subset of the data sets used in our experiments. One can redo the experiments with all original and Deepfake videos from the *FaceForensics++* and *Celeb-DF* data sets and any other data set containing both the original and at least one Deepfake version of any given video in the original set. This applies to both PRNU and visual artifact analysis.

Regarding PRNU analysis, one can also extend the experiments detailed here by including more custom original and Deepfake videos. Since our experiments required the existence of check videos, recorded on the same camera in the same setting, more Deepfakes and original videos could be created in the same manner to provide more comparable input to the PRNU-related experiments. Especially the check videos, which are not present in any public data set known to us, could provide valuable insight into the cross-correlation of Deepfake and original videos.

As previously mentioned, this research paper considers methodology that uses neural networks only marginally. We have found the usage of neural networks as part of Deepfake detection to be quite popular in our analysis of related work. Given the machine learning nature of Deepfakes, this connection does make sense and perhaps warrants future research. While the approaches detailed in this paper and other methodologies not using neural networks may be presently less able to detect Deepfakes, neural-network-based approaches might be more successful.

Since the discriminator of a GAN can also use any detection method that can score video input, it might be possible to look into alternatives that are unusable by a GAN. Given the continuous improvement of Deepfakes using the GAN approach, this may also be the only possible approach to be taken for Deepfake detection in the future. However, designing a detection mechanism that can not be used by a discriminator may be complicated, if not outright impossible.

Instead, as an alternative, further research into the topic of original media authentication is warranted. Systems like AMP are a start to designing and implementing a system that one could use to sign and identify original media where required. More research is needed into the technical details and limitations of deploying such a system. Such research would encompass not only the underlying signature mechanism but also infrastructural questions, such as wide-scale deployment or security of the system itself.

Given the potential of such a system, one should also consider how it should be handled in an organizational context. Questions like who is responsible for the system, both in terms of management and legal operations, have to be discussed.

Lastly, the potential of Deepfake technology also warrants further ethical discussion. If Deepfakes ever get so advanced that they are virtually indistinguishable from original media, rules and regulations, both legal and moral, need to be present. The same applies to the concept of authenticating media. If only signed media is perceived as trustworthy, a new dynamic of consuming digital information is created. Furthermore, the question who signs which media also creates more ethical concerns that one needs to address in the future. In the same way, the technology and its advancements need to be monitored, discussed, and researched.

References

[1] Adobe Inc. *Introducing the Content Authenticity Initiative.* Available at: `https://theblog.adobe.com/content-authenticity-initiative/` [Accessed 19 Jun. 2020]. 2019.

[2] Afchar, D. et al. "MesoNet: a Compact Facial Video Forgery Detection Network". In: *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. 2018, pp. 1–7.

[3] Barbier, G. et al. "Provenance Data in Social Media". In: *Synthesis Lectures on Data Mining and Knowledge Discovery*. 2013.

[4] Blade Group. *Shadow - Your gaming PC powered by Cloud Gaming.* Available at: `https://shadow.tech` [Accessed 11 Jun. 2020]. 2020.

[5] Chen, Y. and Thing, V. L. L. *A study on the photo response non-uniformity noise pattern based image forensics in real-world applications*. Singapore: Institute for Infocomm Research, 2012.

[6] Chesney, B. and Citron, D. "Deep fakes: a looming challenge for privacy, democracy, and national security". In: *Calif. L. Rev.* 107 (2019), p. 1753.

[7] Cole, S. *We Are Truly Fucked: Everyone Is Making AI-Generated Fake Porn Now.* Available at: https://www.vice.com/en_us/article/bjye8a/reddit-fake-porn-app-daisy-ridley [Accessed 02 Jun. 2020]. New York, United States of America, 2018.

[8] DataCadamia. *Machine Learning - Logistic regression (Classification Algorithm).* Available at: https://datacadamia.com/data_mining/logistic_regression [Accessed 30 Jun. 2020].

[9] Durall Lopez, R. et al. *Unmasking DeepFakes with simple Features.* Available at: https://arxiv.org/abs/1911.00686v3.pdf [Accessed 30 Jun. 2020]. 2020.

[10] England, P. et al. *AMP: Authentication of Media via Provenance.* Available at: https://www.microsoft.com/en-us/research/publication/amp-authentication-of-media-via-provenance/ [Accessed 30 Jun. 2020]. 2020.

[11] Faceswap. *Welcome - Faceswap.* Available at: https://faceswap.dev/ [Accessed 03 Jun. 2020]. 2020.

[12] Fagan, K. *A viral video that appeared to show Obama calling Trump a 'dips—' shows a disturbing new trend called 'deepfakes'.* Available at: https://www.businessinsider.com/obama-deepfake-video-insulting-trump-2018-4 [Accessed 02 Jun. 2020]. 2018.

[13] Fridrich, J. *Digital Image Forensics Using Sensor Noise.* New York, United States of America: Binghamton University, 2008.

[14] Goodfellow, I. J. et al. *Generative Adversarial Nets.* Montreal, Quebec, Canada: University of Montreal, 2014.

[15] Google. *Logistic Regression: Calculating a Probability.* Available at: https://developers.google.com/machine-learning/crash-course/logistic-regression/calculating-a-probability [Accessed 30 Jun. 2020]. 2020.

[16] Greenfield, Y. *Intentionally Misleading DeepFakes & Keanu Reeves.* Available at: https://medium.com/@ubershmekel/intentionally-misleading-deepfakes-keanu-reeves-b023a3be522a [Accessed 03 Jun. 2020]. 2019.

[17] Güera, D. and Delp, E. J. "Deepfake Video Detection Using Recurrent Neural Networks". In: *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 2018, pp. 1–6.

[18] Ken Tsui. *Perhaps the Simplest Introduction of Adversarial Examples Ever.* Available at: https://towardsdatascience.com/perhaps-the-simplest-introduction-of-adversarial-examples-ever-c0839a759b8d [Accessed 23 Jun. 2020]. 2018.

[19] Koopman, M., Rodriguez, A. M., and Geradts, Z. "Detection of deepfake video manipulation". In: *The 20th Irish Machine Vision and Image Processing Conference (IMVIP)*. 2018, pp. 133–136.

[20] Leblanc, F. *autocrop.* Available at: https://github.com/leblancfg/autocrop [Accessed 26 Jun. 2020]. 2019.

[21] Leetaru, K. *Why Social Media Provenance Is More Important Than Ever In An AI-Falsified World.* Available at: https://www.forbes.com/sites/kalevleetaru/2018/06/19/why-social-media-provenance-is-more-important-than-ever-in-an-ai-falsified-world/ [Accessed 19 Jun. 2020]. 2018.

[22] Li, Y., Chang, M., and Lyu, S. "In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking". In: *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. 2018, pp. 1–7.

[23] Li, Y. and Lyu, S. "Exposing DeepFake Videos By Detecting Face Warping Artifacts". In: *computer vision and pattern recognition* (2019).

[24] Li, Y. et al. "Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics". In: *IEEE Conference on Computer Vision and Patten Recognition (CVPR)*. 2020.

[25] Lyu, S. "Deepfake detection: Current challenges and next steps". In: *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE. 2020, pp. 1–6.

[26] Matern, F., Riess, C., and Stamminger, M. "Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations". In: *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*. 2019, pp. 83–92.

[27] Netherlands Forensic Institute. *Finding the link between camera and image - Camera individualisation with PRNU Compare Professional from the Netherlands Forensic Institute.*
Available at: `https://www.forensicinstitute.nl/binaries/forensicinstitute/documents/publications/2017/03/06/brochure-prnu-compare-professional/brochure-nfi-prnu-compare-professional_tcm36-21580.pdf` [Accessed 12 Jun. 2020]. 2017.

[28] Nguyen, H., Yamagishi, J., and Echizen, I. "Use of a Capsule Network to Detect Fake Images and Videos". In: (Oct. 2019).

[29] Perov, I. et al. *DeepFaceLab: A simple, flexible and extensible face swapping framework.* Location Unknown, 2020.

[30] Rössler, A. *FaceForensics++: Learning to Detect Manipulated Facial Images.*
Available at: `https://github.com/ondyari/FaceForensics/` [Accessed 03 Jun. 2020]. 2019.

[31] TensorFlow. *Adversarial example using FGSM.*
Available at: `https://www.tensorflow.org/tutorials/generative/adversarial_fgsm` [Accessed 26 Jun. 2020].

[32] The New York Times Company. *The News Provenance Project.*
Available at: `https://www.newsprovenanceproject.com` [Accessed 19 Jun. 2020]. 2019.

[33] Yang, X., Li, Y., and Lyu, S. "Exposing Deep Fakes Using Inconsistent Head Poses". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 2019, pp. 8261–8265.

[34] Zhang, M. *This Website Generates AI Portraits of People Who Don't Exist.*
Available at: `https://petapixel.com/2019/02/19/this-website-generates-ai-portraits-of-people-who-dont-exist/` [Accessed 02 Jun. 2020]. 2019.

[35] Zhou, P. et al. "Two-Stream Neural Networks for Tampered Face Detection". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).* 2017, pp. 1831–1839.

Appendices

A Ground-truth labels parser

```
import os
from shutil import copyfile
import pandas as pd

path='/mnt/c/Users/catde/OneDrive/Documents/SNE/RP2/results/non-trained/'

def get_results(path):
    for dir in os.listdir(path):
        for f in os.listdir(path + dir):
            if f == 'scores.csv':
                parse_label(path + dir + '/' + f, path, dir)


def parse_label(file, path, dir):
    f=pd.read_csv(file, sep=',')
    keep_col = ['Filename','Valid']
    new_f = f[keep_col]
    new_f.to_csv(path + dir + '/labels.csv', index=False)

def get_labels(path):
    for dir in os.listdir(path):
        for f in os.listdir(path + dir):
            if f == 'labels.csv':
                rename_header(path + dir + '/' + f)

def rename_header(file):
    f=pd.read_csv(file)
    f.rename(columns={"Valid": "Label"}).to_csv(file, index=False)

for dataset in os.listdir(path):
    get_results(path + dataset + '/')
    get_labels(path + dataset + '/')
```

B Modification of original code to create new classifiers

```
import os
import argparse
import numpy as np
import pandas as pd

from sklearn.linear_model import LogisticRegression
from sklearn.neural_network import MLPClassifier
from sklearn.externals import joblib

CLF_NAMES = ["mlp", "logreg"]
CLFS = [
    MLPClassifier(
        alpha=0.1,
        hidden_layer_sizes=(64, 64, 64),
        learning_rate_init=0.001,
        max_iter=300
    ),
    LogisticRegression(),
]


def parse_args():
    """Parses input arguments."""
    parser = argparse.ArgumentParser()
    parser.add_argument('-f', '--features', dest='features',
                        help='Path to features saved as .npy.')
    parser.add_argument('-s', '--scores', dest='scores',
                        help='Path to scores saved as .csv.')
    parser.add_argument('-l', '--labels', dest='labels', help='Path to labels saved as .csv.')
    parser.add_argument('-o', '--output', dest='output',
                        help='Path to save classifers.',
                        default='./output')
    args = parser.parse_args()
    return args


def main(input_features, input_score, input_labels, output_path):
    """This script fits the mlp and logreg classifers to new data.

    Processes the feature vectors and scores as saved by process_data.py
    and a .csv file containing the filenames with according labels.
    The labels .csv file is expected to have a column 'Filename' and 'Label'.
    The script provides a basic implementation to fit the
    mlp and logreg classifiers to new data.

    Args:
        input_features: Path to feature vectors as saved by process_data.py.
        input_score: Path to scores as saved by process_data.py.
        input_labels: Path to .csv with 'Filename' and 'Label' column
        output_path: Directory to save classifiers.
    """
    # read input files
    scores_df = pd.read_csv(input_score, sep=',')
    labels_df = pd.read_csv(input_labels, sep=',')
    feature_vecs = np.load(input_features)

    # filter invalid samples
    valid_idxs = scores_df.Valid.values == 1
    valid_features = feature_vecs[valid_idxs]
    valid_filenames = scores_df.Filename.values[valid_idxs]
```

```
    #log reg samples
    feature_vecs = np.nan_to_num(feature_vecs)
    filenames_log = scores_df.Filename.values
    np.nan_to_num(filenames_log)

    # get labels for valid samples
    labels = []
    for filename in valid_filenames:
        labels_row = labels_df.loc[labels_df['Filename'] == filename]
        if labels_row.size == 0:
            print "Missing label for: ", filename
            exit(-1)
        labels.append(labels_row['Label'].values[0])


    # get labels for all samples
    labels_log = []
    for filename in filenames_log:
        labels_row = labels_df.loc[labels_df['Filename'] == filename]
        if labels_row.size == 0:
            print "Missing label for: ", filename
            exit(-1)
        labels_log.append(labels_row['Label'].values[0])

    # create save folder
    if not os.path.exists(output_path):
        os.makedirs(output_path)

    for name, clf in zip(CLF_NAMES, CLFS):
        if name == "logreg":
            clf.fit(feature_vecs, labels_log)
        else:
            clf.fit(valid_features, labels)
        joblib.dump(clf, os.path.join(output_path, name + '.pkl'))


if __name__ == '__main__':
    args = parse_args()
    main(args.features, args.scores, args.labels, args.output)
```