Vanguard: A Blockchain-based Solution to Digital Piracy

Yasiru Jayasinghe¹, Kavinga Yapa Abeywardena², Tharika Munasinghe³, Sumala Mannage⁴, Thisuri Warnasooriya⁵ and Gihan Edirisinghe⁶

¹ Sri Lanka Institute of Information Technology

Received: 16 December 2019 Accepted: 1 January 2020 Published: 15 January 2020

Abstract

Online pirates and Intellectual Property (IP) holders have been in perpetual war over various products like music, movies, software, etc. since the popularity of the Internet. It is estimated that the US entertainment industry loses approximately 29 billion USD every year for pirates. Online piracy has since gone from bad to worse as growing internet users and better broadband connections enable people to share large files freely over the internet. The objective of this research is to investigate the causes and enablers for online piracy in movie industry and to come up with an anti-pirating solution. The primary outcome of the study will consist of a dedicated blockchain based anti-piracy system, Vanguard. This system will provide all-round piracy protection from a built-in streaming service to a component to actively look through the internet for pirated movies and torrents. This system will greatly deter the piracy of movies since the IP holders can ensure their IP rights through this system and quickly act against illegitimate distribution of their media.

Index terms—blockchain, streaming service, intellectual property, image matching, string matching, deep-learning.

1 Introduction

Intellectual property refers to creations of the mind. This ranges from inventions, literary and artistic works, symbols, names and images used in commerce [1]. A motion picture copyright protects the artistic expression in movies, short films, and videos, including the camera work, dialogue, and sounds [2]. It does not protect idea of the movie or characters showed in it.

The first well-known court case on copyright infringement in the movie industry was Nichols v. Universal Pictures Corporation court case. The case was won by the defendant due to the court deciding that copyright protection does not include ordinary characters in a story.

The first ever concept of streaming was brought to light when physicists were able to develop a method use of wires [3]. This resulted in the possibility of transmitting radio signals over long distances. Now, in the 21st century, streaming majorly involves live broadcasts though the internet. The technology which began in the early 1990s has now been revolutionized by companies such as Netflix, Spotify, YouTube, Hulu and Pandora [3], which are services that offer television shows, music, movies, sports and games directly to your computer.

2 a) Research Problem i. IP Protection Problem

According to United States copyright office annual report in fiscal 2018, the Office registered 560,013 claims for registration and recorded nearly 21,700 documents containing titles of more than 757,400 works [5]. At the same time according to the Motion Picture Association of America (MPAA), there are approximately 150,000 movie screens in the world, with about 40,000 of those in the United States alone. About 560 films were released at the cinema in the United States in 2010-about 11.5 every week. Of those, 419 were independent films [6].

By analyzing these figures, it is clear that ensuring IP rights of each movie is not an easy task. Among all IP rights, copyrights play a major role in movie industry and copyright law demand chain of documentation.
3 ii. The Increase of Digital Piracy

The rapid development of the internet in recent years brought about a large increase in services that provide media streaming for consumers. This also brought new ways for pirates to distribute media through illegal channels, and for users to access those channels more easily and at a much larger scale [4].

About 34% of all recorded music products sold worldwide in 2004 were pirated copies and that piracy costs the industry over $4.6 billion per annum [7]. Research by Digital TV Europe states that revenue losses for the television and film industries as a result of piracy could reach $52 billion by 2022, as shown in Figure 3 below [8].

Current Solutions Over the years there have been many attempts and researches about more robust IP protection and deterring digital piracy, through means such as DRM (Digital Rights Management) takedowns, advertising models etc. Some such solutions are Bernstein [9][10], which is solution that combines blockchain technology and IPFS, and the Traditional Copyright Process [2], which requires the submission of several documents throughout the movie creation process. After submitting all the documents, the process will go through several steps for around three to ten months, before finally resulting in the copyright certificate.

As solutions against piracy, the Advertising model [11] and other unique approaches such as iTunes’ Matching Service [11] act as alternatives to piracy. On the other hand, Custos Screener Copy [12] is a blockchain-based solution that uses forensic watermarking technology to track content. This solution uses the decentralized nature of blockchain and uses peers of the network to hunt down copies of the media.

4 ii. Research Gap a. Additional solutions for IP protection

Although there are many existing technologies and solutions that try to address the management of IP rights, there are a few key areas that can be used to specifically enhance the IP rights of movie industry using blockchain technology.

IP right enforcement agencies are overloaded with work and the process takes too long [13]. Difficulties in enforcing Licensing Agreements [14]. In the proposed solution, by using blockchain technology combined with smart contracts, it can fully automate the copyright process through obtaining and verifying documents with each milestone of copyright process during the movie creation process. At the end of the movie production, the producer can obtain their copyright ownership certificate. It can also be used to automatically enforce rights and using pre-define rules for royalties to be properly allocated and distributed.

5 ii. Additional solutions for deterring piracy

The Advertising Model and Matching Service solutions [11] mentioned above do not attempt to actively address the digital piracy issue. Instead, they aim to provide alternatives for consumers to make them less likely to consume media through illegal channels.

Custos Screener Copy [12] on the other hand, is a product that directly aims to deter piracy of media. There are two ways in which Custos Tech’s solution can be improved, they are:

Creating a more robust forensic watermarking system. Automating the watermark extraction and comparison process CustosTech’s solution to finding pirated media depends on human miners. This makes the process somewhat unreliable as they will not always be searching for a specific creator’s media. Vanguard aims to automate this process, to assure the creator that the system will be continuously searching for illegal copies of their media.

The table below summarizes the additional features that will be introduced by Vanguard. The diagram below (Figure 3) represents the system overview for Vanguard’s IP Protection component. First, the client enters registration details to create their account, which is saved in a smart contract in the blockchain. After the client has created their account, they are able to start the media licensing process. For this process, the client must upload all necessary documents during the time of producing the media, and the system will save time-stamped records of the documents in the blockchain using smart contracts. The system will compare the documents with any previously uploaded documents to assure the originality of the documentation process. After the media production process is complete, all records and documents will exist as tamper-proof records in the blockchain. The client must then upload the final media file to Vanguard, which will be used for Vanguard’s Streaming Service component (component 2), whenever the client chooses. Finally, the client can generate a fully digitalized certificate that proves their IP rights to the Media product, which they can use in cases of IP violations to prove their rights.

6 ii. Streaming Service Component Design

The diagram below (Figure ??) represents the system overview for Vanguard’s Streaming Service component. To access this service, the client must already have a registered account with Vanguard through Component 1.
7 Fig. 4: Vanguard Streaming Component Overview

When the client wishes to upload their media to the streaming service, the system will first fetch the media from the blockchain. The system will then generate the traceable forensic watermark that contains details about where it is being uploaded to, the original content owner etc. The generated forensic watermark will then be embedded into the media in a way that cannot be easily removed by pirates, while remaining imperceptible to the human eye. Finally, the watermarked media file is uploaded to the streaming service and will be available to stream for consumers, and the data contained in the watermark will also be added to the blockchain for future reference.

8 iii. Piracy Search Algorithm Component Design

As described in figure ??, the system first extracts the watermark from the downloaded video and compares it with the original watermark, if it matches, the system sends an alert to the IP holder and they can take a relevant action against the pirated video.

10 Fig. 6: Vanguard Video Copy Detection Overview

If the system cannot detect any matching watermark, then it will compare videos based on frame by frame image matching and will calculate the similarity percentage. If the similarity percentage exceeds the prescribed limit, the IP holder will be alerted. If not, as the final process the system will detect details that can't be detected from image frame matching. The most common distorts done by pirate uploaders are color unbalancing and brightness changing. The number of motions within two frames and audio files of the videos avoid those. So that the system will compare the similarities through them. This process will continue until all the downloaded videos from the search results are analyzed.

11 b) Implementation i. Media Licensing

Vanguard’s IP Protection (component 1) is responsible for developing a platform that specifically addresses movie copyrights. As the first step of implementation, the development environment is setup by combining a few different technologies. The Ganache Personal Blockchain is used as a local development of blockchain which has similar behaviors to public blockchain. This is equipped with 10 accounts that are credited with 100 ether, which allows to develop applications, deploy smart contracts and run tests. Node Package Manager is used to configure the environment for developing smart contracts. Next, the Truffle Framework is installed, which provides functionalities to develop and deploy smart contracts, which can run on Ethereum Virtual Machine and in any public Ethereum blockchain network, and is able to write tests against smart contracts before uploading to the blockchain and support client-side development. Finally, the browser extension Metamask Ethereum Wallet is installed to connect to blockchain network through the browser.

12 a. Milestone Checklist

One of the main problems identified through research gap is that not enough information has been collected during the movie development process and it is difficult to prove the originality of the work. A Milestone checklist is developed to address this problem through requesting essential documents that movie producers should upload in the traditional copyright process to obtain the certificate. This checklist is developed through smart contracts using Solidity programming language and deployed to the Ganache Ethereum blockchain to ensure the integrity of the certificate. Because storing large files on blockchain costs a large amount of gas, uploading documents to the blockchain is identified as expensive. The Interplanetary File System (IPFS) is recognized as an ideal solution. When connecting to the IPFS, Infura which is a hosted Ethereum node is used, rather than running an IPFS daemon on the host due to it being identified as faster when uploading documents. As the figure shows, the program captures the user file, converts it to a buffer, sends it to IPFS, and returns a hash.

13. Comparison and Updating the Blockchain

The hash received from the IPFS is uploaded to the blockchain using HashStorage.sol smart contract. When uploading the hash, as the figure shows it captures the timestamp of the file hash added to the blockchain. This
A. DETECTING WEB PAGES

14 ii. Forensic watermark

The generation and embedding of the forensic watermark are done automatically by Vanguard’s Streaming Service Component. The component is implemented using the Python programming language. The initial steps of creating this process required a way to hide a message inside a single frame of a video. The python library NumPy was used to represent a 3-dimensional array, each dimension representing the row, column, and color, which combine to form a single pixel of the video frame (figure ??). The watermark message is taken as an input and its binary form is computed and appended into the 3-dimensional array, and finally concatenated with the image pixels to embed the data. Fig. ??: Using NumPy to embed a Forensic Watermark a. Attention

Model One main disadvantage observed through the above approach is that any form of transformation of the watermarked frame causes the watermark to be destroyed. Prior researches such as RivaGAN [15] proved how deep learning-based approaches can be used to embed more robust watermarks in media. Following this approach, Vanguard’s watermarking system uses an Attention-based model [16] to learn the probability distribution over data dimensions for each pixel.

The Attention model is equipped with a neural network with the ability to focus on different subsets of its inputs [16]. In the case of a video frame, it has the ability to study the frame and identify different objects and textures within the frame. The data gathered through the Attention model is used to generate an Attention mask, which is used to determine which bits to embed the watermark into during the embedding process.

15 b. Training the Model

To improve the performance of the training model, Adam Optimizer is used. It is easy to implement, more computationally efficient, requires little memory space, and works well on problems with noisy or sparse gradients.

16 c. Main Functions

Some of the most important functions of Vanguard’s watermarking system is the Attention model, Encoder, and Decoder functions. For the Attention model, the Pytorch library is used to create the attention mask on a tensor level. This was chosen over Tensorflow because of how well it integrates with Python.

In the Encoder and Decoder functions, Open CV library is used to obtain and manipulate frames from video files (Figure 10). The generated Attention mask is used by the Encoder function to determine which bits to pay attention to at each pixel, and the Encoder function computes a compact form of the data tensor which includes the watermark and concatenates it with the frame. The reverse operation is done by the Decoder function to retrieve the watermark from the frame. In order to increase the resilience of the watermark against various forms of transformations, noise layers are added to the video before it undergoes the embedding process. The implemented watermarking system is currently resistant to scaling and cropping of the video.

The scaling layer re-scales the video to a random size between 80-100% of the height and width of the original. This allows the system to learn to embed the watermark in a scale-invariant manner. Similarly, the cropping layer selects a sub-window of 80-100% of the original video’s height and width randomly and uses that sub-window to embed the watermark.

17 iii. Web Crawler Algorithm

The web crawler component of the Vanguard system is focused on detecting any pirated copies of the movies that have a contract with the system. It is designed to search through known and new torrenting websites and scan through for any offending files. This component is implemented using a framework called 'Scrapy'. This is a python framework that allows the user to fine tune a web crawler to their personal needs. Scrapy allows the user to download a file that matches any of the criteria given to the web crawler. The name and other metadata of the movie will be fetched from the Blockchain database which hold all contract information.

18 a. Detecting Web Pages

The Scrapy framework provides the user with a class called a ‘Spider Class’. This class defines the rules and boundaries of the web crawler that is being made. The websites the web crawler needs to visit is defined in this class. The spider class takes the URL of the website that we are defining, and the framework will use this to make a request to the website to scan through it. The Scrapy framework scrolls through the given webpages in a depth-first order. Which means the algorithm starts at the inner-most page of the website and scans outwards to ensure that no page has been missed. The filenames that are passed onto the framework via the media server will then be compared. Scrapy uses a mechanism known as Selectors to extract data from the website. Selectors work by converting the source of the website into HTML source code and then scanning through them. The name of the content that we want to match is given through a response.css function to the framework. Once the framework finds a filename that matches any of the titles of the movies that the Vanguard system has a
contract with, the framework will automatically download it. The Scrapy framework uses a mechanism known as a Files-pipeline. Once the files-pipeline detects the division of the webpage, in which the link to the file is, it goes to the division and downloads the file. The Files-pipeline lets the user configure which file type can be downloaded. OpenCV (Open Source Computer Vision Library) has more than 2500 optimized algorithms. These algorithms can be used to detect objects, track movements, etc. Throughout the implementation of the system, OpenCV library is used with python to process videos to detect pirate movies. Year 2020

19  b. Converting Videos to Image Frames
Before being compared through watermarks and image frames, the videos should be converted into image frames. Image frames of every 0.5 seconds should be captured.

20  c. Watermark Comparison
The original watermark added by the legitimate movie company should be searched and matched to detect pirated movies among downloaded movies. OpenCV feature matching technique detects objects from images even though there are slight brightness changes, size changes, or the images are rotated into different directions.

21  d. Image Frame Matching
If the system unable to detect watermarks, converted image frames from downloaded movies and the original movie get compares with each other to detect similar images. FLANN is Fast Library for Approximate Nearest Neighbors. It contains a collection of algorithms optimized for fast nearest neighbor search in large datasets and for high dimensional features. After FLANN checks for similar key points and matches, next, the system checks and matches the image frame sizes and color channels. After checking for image frames matching, the system calculates the percentage of the image frame similarity.

22  e. Motion and Audio Detection
If the percentage of the image frame similarity is unable to confirm pirated videos, the system uses motion detection and audio detection to identify them. OpenCV motion detection technique detects the number of motions within two frames. The system uses this technique to calculate motions in videos. It detects each and every motion within two frames, which might occur inaccurate results. Because of this, the motion matching is further improved to detect motions in an area size of 1000 or more pixels. Moviepy is a python module for editing videos. Moviepy can extract audio files from a video by simply calling the audio method. Audiodiff is a simple python library that is used to compare audio files. It can detect if two audio files contain the same audio streams and normalized tags.

23  III.

24  Results and Discussion

a) Media Licensing
To ensure the smart contracts are working as intended, test cases were written to test the functionality. Tests were written using the Chai assertion library and Mocha Testing Framework which was received with truffle framework. These are client-side simulations of the smart contracts and it shows the extract results that was intended to obtain (Figure 14). To train the model, the Hollywood2 dataset is used. This dataset is a collection of over 2500 short clips taken from various movies, all depicting certain actions such as getting into a car, opening a door etc. These clips train the Attention model to identify specific objects in a frame.

After the embedding process, the resulting watermarked frame does not show any visible degradation when compared to the original image (Figure 15). The table below shows the video quality after embedding 32 bits of random data as a watermark into the video. The PSNR column is the computed peak signal-to-noise ratio. The high value of the resulted PSNR shows that the watermarked frame is of a good quality. Year 2020

b) Web Crawler Algorithm
The framework has been tested and tried on known pirating websites such as yts.ag and Piratebay. It has been able to locate the division the link to the pirated files exist, however, the downloading process is haphazard. The files-pipeline should be tuned to download magnet files, which store the hashcode for the torrented media files. This has been an issue since the framework needs to invoke a different application to download the pirated media file through the magnet file.

c) Web Crawler Algorithm
The framework has been tested and tried on known pirating websites such as yts.ag and Piratebay. It has been able to locate the division the link to the pirated files exist, however, the downloading process is haphazard. The files-pipeline should be tuned to download magnet files, which store the hashcode for the torrented media files. This has been an issue since the framework needs to invoke a different application to download the pirated media file through the magnet file.

d) Copy Detection
When using watermark comparison, the accuracy of the results decreases with the opacity (figure 16). When comparing, using low opacity image as the template and using the original image as a template, using the original image shows more accurate results than using images with low opacity. When comparing image frames, the system checks for the distance between key points, image size, and color channels. Image noise reduces the
accuracy of the results. For example, it can show high accuracy when comparing completely different images with a darker background.

27 Future Work a) IP Protection Component
The functionality of the entire component can be improved in the future by developing a user friendly interface that is compatible with the blockchain. Also: 
? The search function should be developed to search the already deployed documents and to obtain the details. 
As per now only the owner can view the details of the documents, but this should further improve to allow the owner to grant access to other users to view the documents. 
? The ownerable.sol smart contract should future develop to save the registration details of the account holder in blockchain when they first created the user account. 
The final task is to generate the digital certificate that allows IP holders to prove their ownership of the media.

28 b) Streaming Service Component
Further improvements can be made to the watermarking system. There are two key areas that can be improved further: 
? Increasing the performance of the embedding and extraction process: The system currently uses CUDA for GPU-based acceleration. The issue with this is that it is currently only supported by NVIDIA. The system must be modified so that it supports Intel and AMD based chipsets as well. 
? Further improvement of the robustness of the watermark. As of now, the watermark is resistant to cropping and scaling of the video. Further persistence can be obtained by adding Compression resistance. This can be done using another noise layer that uses discrete cosine transform (DCT) before embedding the watermark, thus forcing the system to embed the watermark in a compression-resistant manner. 
After completing the watermarking system, the Streaming service must be developed which has the ability to automatically embed the watermark into the video before hosting on the platform.

29 c) Web Crawler Algorithm
There are a few areas the framework and the controller classes for this algorithm can be improved further: 
? Increasing the accuracy of the files’ title detection is currently a main target as the title matching often turns out inaccurate. This is due to the Selector classes not being able to properly extract the desired filename. 
? The downloading process should be streamlined. 
As of now, the web crawler is only able to download the magnet file which contains the hashcode for the pirated content. The user needs to have a thirdparty application to open the hashcode and access the pirated movie file. This process will be streamlined into an automatic sequence for the ease of use. 
After the web crawler is perfected, it will be connected to the online blockchain database and the media server, where it’ll be able to access all the metadata of the movies.

30 d) Copy Detection System
Video copy detection system can be further improved to get more accurate results. 
? Image noise can be reduced from both watermark comparison and image frame comparison. Image noise affects the accuracy of the results. 
? Audio matching comparison and motion detection comparison can be calculated to get the overall comparison and check similar movies from the downloaded movie results. 
After detecting pirate movies in each step, the system should be developed to send alert messages to legitimate movie owners, allowing them to take relevant legal actions against the pirated movies.

31 Conclusion
Proper IP protection and legal action against media piracy has been a continuous struggle for many years. Over the years there has been a multitude of approaches that tackle these problems in various ways. Vanguard aims to actively search and take action against pirated media. The blockchain-based IP protection system offers a digital alternative to the traditional IP registration system, that will help ensure the owners IP rights in a case of illegitimate distribution of their media. The forensic watermarking system can help prove the origin of the media and directly prove the owner’s rights. The Web crawler and Copy detection systems cooperatively aim to actively find pirated copies of media throughout the internet and alert the owner if they are found. All these components together can provide its clients with all-round protection against digital piracy, and in turn will actively aid in deterring the continuation of piracy of digital media.
Figure 1: Fig. 1:

![Graph showing legitimate online revenues versus piracy losses over time.](image)

Source: Digital TV Research

Figure 2: Methods a)

![Diagram illustrating methods related to content creators, IP protection, streaming platforms, and piracy detection systems.](image)
Figure 3: Fig. 2:
Figure 4: Fig. 3:

Figure 5: Fig. 5:
CONCLUSION

Figure 6: Fig. 7:

Figure 7: Fig. 8:
function add(
    string memory _ipfHash,
    string memory _filename,
    uint256 _dateAdded
)
{
    count++;
    require(
        collection[_ipfHash].exist == false,
        "this hash already exists in contract"
    );
    DocInfo memory docInfo = DocInfo(_ipfHash, _dateAdded, true);
    collection[_ipfHash] = docInfo;
    emit HashAdded(_ipfHash, _filename, _dateAdded);
}

def embedding_info(plineame, saverename, text):
    try:
        im = np.array(Image.open(plineame))
    except: 
        print("Cannot obtain image, please check file name")
        time.sleep()
        sys.exit() 
    rows, columns, colors = im.shape
    embed = []
    for k in text:
        bin_sign = (bin((ord(k)))[2:]).rjust(16)
        for i in range(16):
            embed.append(int(bin_sign[i]))
    count = 0
    for row in range(rows):
        for col in range(columns):
            for color in range(colors):
                if count < len(embed):
                    im[row][col][color] = im[row][col][color] // 2 + embed[count]
                count += 1
    Image.fromarray(im).save(saverename)

def encode(self, video_in, data, video_out):
    assert len(data) == self.data_dim
    video_in = cv2.VideoCapture(video_in)
    width = int(video_in.get(cv2.CAP_PROP_FRAME_WIDTH))
    height = int(video_in.get(cv2.CAP_PROP_FRAME_HEIGHT))
    length = int(video_in.get(cv2.CAP_PROP_FRAME_COUNT))
    data = torch.FloatTensor([data]).cuda()
    video_out = cv2.VideoWriter(
        video_out, cv2.VideoWriter_fourcc('mp4v'), 10.0, (width, height))
    for i in range(length):
        ok, frame = video_in.read()
        frame = torch.FloatTensor([frame]) / 255.0
        frame = frame.permute(0, 1, 2, 3).unique().cuda() 
                              # (1, 3, L, W)
                              # (1, 3, L, W)
        img = self.encoder(frame, data)
        frame = torch.clamp(img_frame, min=1.0, max=1.0)
        frame = frame.permute(0, 2, 3, 1)
        frame = frame[:1, :, 0, :].cpu().numpy().astype("uint8")
        video_out.write(frame)
Figure 11: Fig. 12:

Figure 12: Fig. 13:

Figure 13:
Figure 14: Fig. 14:
Figure 15: Fig. 15:

Figure 16:

Figure 17: Fig. 16:
II.

Figure 20: Table 1:

| Product                      | Define milestones of copyright process | Chain of custody | Unchangeable data | Actively deters media piracy |
|------------------------------|----------------------------------------|------------------|-------------------|------------------------------|
| Traditional Copyright Process| â??”                                    | ?                | ?                 | ?                            |
| Blockchain based solution (Bernstein) iTunes Match Service Custos Screener Copy Vanguard | ?                        | ?                | ?                 | ?                            |
|                              | â??”                                    | â??”             | â??”             | â??”                         |

II.

Figure 21: Table 2:

| Model          | Bits | PSNR  |
|----------------|------|-------|
| Attention      | 32   | 42.71 |
| Attention + Noise | 32   | 42.61 |
.1 Acknowledgment

We sincerely thank our evaluation panel examiners, Mr. Amila Senarathne, Mr. Buddhika Harshanath, and Ms. Thamali Dassanayake for their valuable feedback and suggestions given during presentation, evaluation, and discussion sessions. Special thanks also go to Dr. Janaka Wijekoon for his lectures and advice given to carry out a successful research. Finally, our sincere thanks go to each and every one who aided us in every step towards the completion of this research project.

[Technical] A Brief History of Streaming Media, Pace Technical. https://www.pacetechnical.com/brief-history-streaming-media/

[Kosiorek] Attention in Neural Networks and How to Use It, Kosiorek. http://akosiorek.github.io/ml/2017/10/14/visual-attention.html

[HiLL (2007)] ‘Digital Piracy: Causes, Consequences, and Strategic Responses’. Charles W L Hill. 10.1007/s10490-006-9025-0. https://doi.org/10.1007/s10490-006-9025-0 Asia Pacific Journal of Management 2007. 24 (1) p.

[Digitaltv (2017)] DigitalTV Europe. Retrieved from Piracy to cost TV and film industry US$52bn by 2022, Digitaltv. https://www.digitaltveurope.com/2017/10/30/piracy-to-cost-tv-and-filmindustry-us52bn-by-2022/ 2017. October 30.

[Sudler (2013)] ‘Effectiveness of anti-piracy technology: Finding appropriate solutions for evolving online piracy’. H Sudler. Business Horizons 2013. 56 (2) p.

[Hackins (2018)] ‘How to Copyright A Movie or Short Film’. J Hackins. legalzoom.com Jan-2018. 16.

[Peukert and Claussen (2012)] ‘Piracy and Movie Revenues: Evidence from Megaupload’. C Peukert, J Claussen. SSRN Electronic Journal 2012.

[Zhang et al. (2019)] Robust Invisible Video Watermarking with Attention. MIT EECS, Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, Kalyan Veeramachaneni. September 2019.

[United States Copyright Office - Annual report for fiscal 2018 (2020)] United States Copyright Office - Annual report for fiscal 2018. https://www.copyright.gov/reports/annual/2018/ar2018.pdf Jan-2020. p. 23.

[Custostech] Watermarking is not enough – How Blockchain can be used to stop online piracy, Custostech. https://www.custostech.com/wp-content/uploads/2019/04/custos-technology-whitepaper.pdf

[What is Ethereum. Guide for Beginners] What is Ethereum. Guide for Beginners, https://cointelegraph.com/ethereum-for-beginners/what-is-ethereum Coin Telegraph

[What is intellectual property?] / World Intellectual Property Organization Geneva: World Intellectual Property Organization (2005)