The Role of Neural Style Transfer Learning in the Creation and Dissemination of NFTs

Isabelle Jaber

April 2023

Table of Contents:

Chapter 1: Introduction

Non-Fungible Tokens (NFTs) are a recent development in technology that represent digital assets like art, collectibles, and video game-related items [1]. These digital assets consist of a unit of data stored on a blockchain. A blockchain can be thought of as a "chain of blocks that contains a complete record of transactions that may be publicly or privately distributed (decentralized) to all users of the chain" [2]. The NFT data stored on this blockchain validates the digital asset as unique. This also provides a unique, identifiable certificate of ownership for the asset and allows it to be exchanged, most often with cryptocurrency, and the transaction is stored in another block in the chain. This blockchain requires all users to have access to the distributed ledger. This means that transactions are recorded only once, so the duplication of records typical of traditional business methods no longer occurs. As previously mentioned, the records contained within the blockchain are unchangeable, so if there is an error another record must be created to override the error. Each transaction is recorded as a block of information, and these blocks are linked together to form the blockchain so blocks cannot be changed or inserted between two existing blocks. This prevents any possible tampering [3]. As such, unlike most digital assets, especially photographs which can be copied, reproduced, and redistributed, NFTs provide a way to easily identify ownership of a digital art piece.

As NFTs are a fairly new technology – with the first NFT minted in 2014 [4]– there are many who seek to understand, investigate, and manage this new digital market. While there are many advantages and disadvantages to participating in, and interacting with, blockchain transactions – some of which will be discussed in Chapter 3 – one serious concern many are having with NFTs is about how to handle copyright and intellectual property disputes among creators of NFTs and those dealing with tangible goods. Neural Style Transfer learning is a type of Artificial Intelligence that, if applied to NFTs, could contribute greatly to these discussions.

Chapter 2: Neural Style Transfer Learning

Neural Style Transfer Learning is a deep-learning-driven image modification technique developed by Gatys *et al.* in 2016. This technique aims to apply the style of a reference image to that of a base image while preserving the content of the base image. In this case "style" refers to the textures, colors, and patterns in the image in relation to various spatial scales. These are the aspects of the reference image that Neural Style Transfer Learning seeks to extract and apply to the base image without disrupting the macrostructures of the base image. When Neural Style Transfer Learning was initially released, it revolutionized the world of computer vision as the results of this deep learning-based implementation were unparalleled compared to previous computer vision techniques [5].

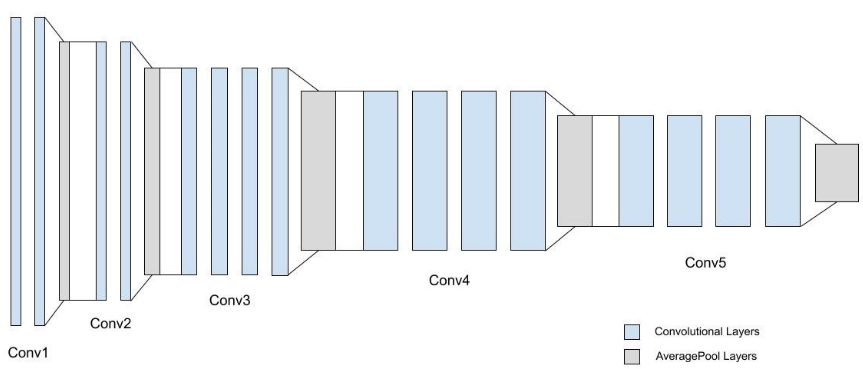

Figure 1: VGG19 used by Gatys *et al*.

Gatys *et al.* chose to use a Convolutional Neural Network for this because it is an algorithm that is considered comparable to human performance in object recognition [6]. Specifically, they used a VGG19 Neural Network, first proposed by Simonyan and Zisserman in 2015 [7], which consists of 16 convolution layers – a filter that is convolved over the image to generate a feature map that locates the higher intensity regions [8] – and 5 Average Pool layers (as seen in Figure 1).

This VGG19 network architecture takes a color image (RGB image) of a fixed size (S x S) as its input so the matrix is of the shape (S x S x 3). Now a process known as Convolution begins where a filter of size (3 x 3), a stride of 1, and given weights is shifted over the input image [9]. In this process, the network calculated the product of the filter's weights and the pixel value of the image and sums the values to be placed in the feature map (see Figure 2) [10].

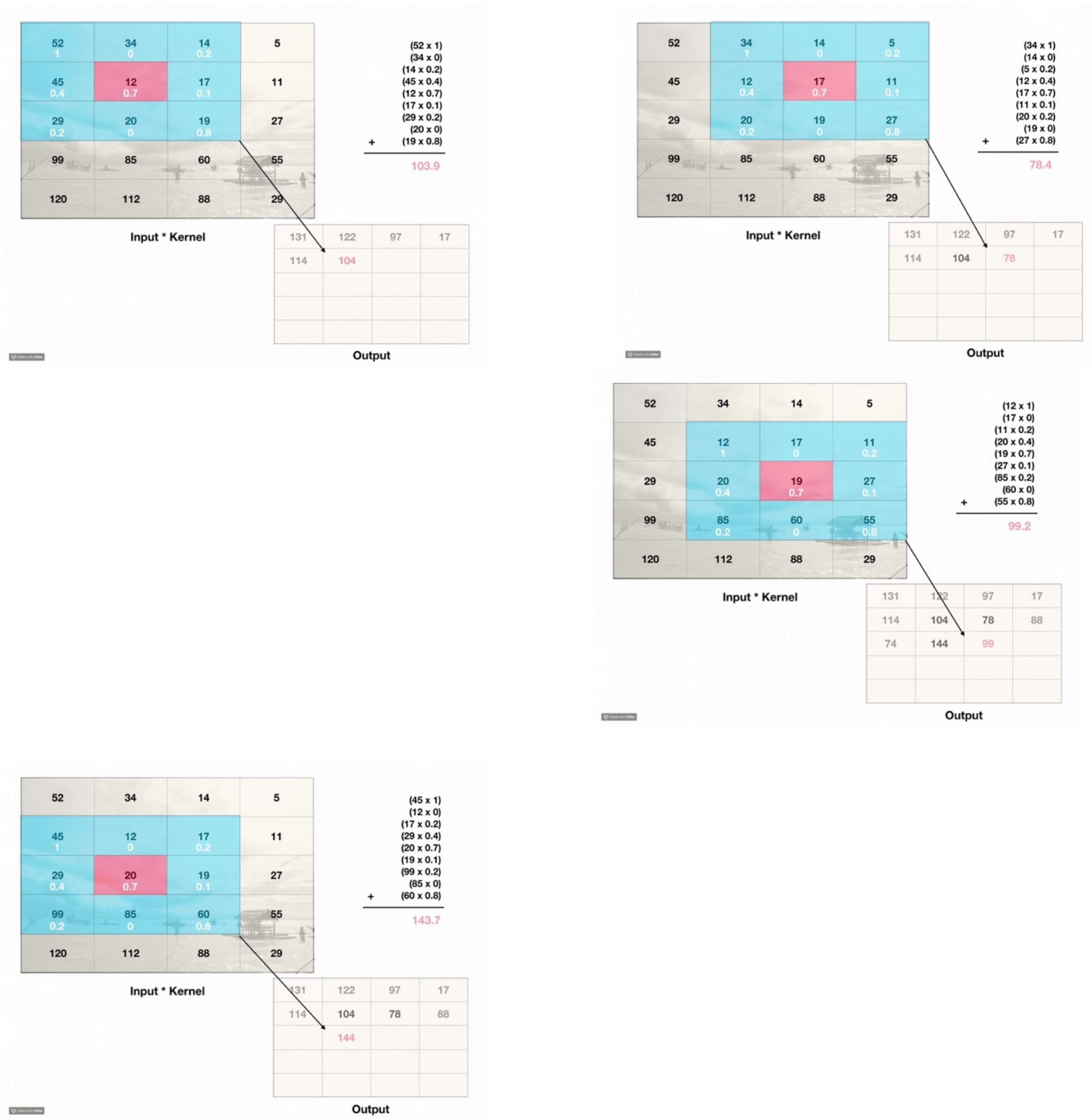

Figure 2: Process of Convolution in VGG19 Network (*Convolutional Neural Networks*)

Spatial padding is also used in this process to preserve the resolution of the image. Following these convolutional layers, pooling layers can be applied before continuing onto the next convolutional layer [9].

The pooling layers are intended to combine the outputs of the previous layer to reduce the dimensions of the next convolutional layer. There are multiple ways to do this, and typically, in VGG19 networks, the Max Pooling method is used. Max Pooling is when the brightest pixel from the cluster of outputs is selected to be in the next layer and is typically more useful for images where the background is dark, and the program is mostly focused on the information in the lighter foreground. Average Pooling averages the outputs in the cluster and smooths out the image [11]. Gatys *et al.* found that using Average Pool yielded better results than using Max Pool, so they amended their VGG19 model to reflect this [6]. In this case, Average Pool worked better than Max Pool due to the type of information we were trying to get. In cases where object recognition and edge detection are important information for the problem the program is trying to solve, Max Pool would be more beneficial since it is more sensitive to changes in pixel intensity. However, because of the kind of problem Neural Style Transfer Learning is trying to solve, Average Pool averages out the image which results in more information about the general shapes and style of the image. Each convolutional layer in the network is similar to a filter that extracts different features from the input image. Therefore, along the process of the network, the input image becomes "transformed into representations that increasingly care about the actual *content* of the image compared to its detailed pixel values" [6]. The higher layers in the network contain the high-level content (objects and their arrangement in the input image) without constraining the assigned pixel values in the reconstruction made with that layer's associated feature map. On the other hand, the reconstructions from the lower layers' feature maps contain exactly the same pixel values as were contained in the input image. It is for this reason that we will use the higher-layer feature maps to reconstruct the content of the input image. To obtain the style of an input image, a feature space, constructed of the filter responses of each layer, is used to get "a stationary, multi-scale representation of the input image, which captures its texture information but not the global arrangement" [6]. Essentially, it creates a feature space from the feature maps given by each convolutional layer, which represents the patterns and textures in the image but no information about the content. Gatys *et al.* identify the key finding of their work to be that the style and content of images are distinguishable and able to be manipulated independently of each other.

As applicable as this is to computer vision it is just as relevant as a potential method of minting NFTs. The Neural Style Transfer Learning program discussed here was designed to apply the style of a random Impressionist painter to a modern photograph for the express purpose of creating an image that can be sold as an NFT. As a result, the only change made to the initial program as outlined by Gatys *et al.* is the addition of randomly selecting a reference image from a directory with approximately 400 images of paintings from each of 10 Impressionist painters (Cezanne, Degas, Gauguin, Hassam, Matisse, Monet, Pissarro, Renoir, Sargent, and Van Gogh), and randomly selecting a base image from a directory with approximately 20 modern photographs. This would allow for multiple combinations of results and the more reference and base images placed into the directory the greater number of unique NFTs can be produced from the program.

This program, from the second edition of Francois Chollet's *Deep Learning with Python*, begins by selecting a random file path to the reference image and to the base image, defining those paths, and resizing the associated images to a shared height of 400px. This is because using images of drastically different sizes could make the transfer process more difficult.

```
import tensorflow as tf
import numpy as np
```

```
import os
import random
from tensorflow import keras

random_style_file="/students/shared/ijaber/downloads/style_data/" +
random.choice(

      [x for x in os.listdir(

            "/students/shared/ijaber/downloads/style_data"

      )
      if
      os.path.isfile(os.path.join("/students/shared/ijaber/downloads
      /style_data", x))]

)
random_base_file="/students/home/ijaber/PycharmProjects/CSThesis/bas
e_data/" + random.choice(

      [x for x in os.listdir(

            "/students/home/ijaber/PycharmProjects/CSThesis/base_data
            "

      )

      if
      os.path.isfile(os.path.join("/students/home/ijaber/PycharmProj
      ects/CSThesis/base_data", x))]

)


base_image_path=keras.utils.get_file(origin="file://" +
random_base_file)
style_image_path=keras.utils.get_file(origin="file://" +
random_style_file)

original_width,
original_height=keras.utils.load_img(base_image_path).size
img_height=400
img_width=round(original_width * original_height / original_height)
```

Next, these “auxiliary functions” will focus on processing the image. Since, whole images do not make sense to the VGG19 model, it must be converted into raw pixels in the form of a numpy array so it can be entered into the VGG19 (preprocess_image(image_path)).

Similarly, after the numpy array has been transformed by the model it must then be translated from the numpy array given back by the model, into a valid image (deprocess_image(img)).

```
def preprocess_image(image_path):
    img=keras.utils.load_img(image_path, target_size=(img_height,
      img_width))
    img=keras.utils.img_to_array(img)
    img=np.expand_dims(img, axis=0)
    img=keras.applications.vgg19.preprocess_input(img)
    return img
def deprocess_image(img):
    img=img.reshape((img_height, img_width, 3))
    img[:, :, 0]+=103.939
    img[:, :, 1]+=116.779
    img[:, :, 2]+=123.68
    img=img[:, :, ::-1]
    img=np.clip(img, 0, 225).astype("uint8")
    return img
```

Now, it's necessary to set up the VGG19 network responsible for extracting the features from the reference image. For this model, the model will be loaded with pre-trained weights from ImageNet (typically used for image classification). This is also when the feature extractor is defined as a dictionary of the activation values at each target layer.

```
model=keras.applications.vgg19.VGG19(weights="imagenet",

      include_top=False)

outputs_dict=dict([(layer.name, layer.output) for layer in
model.layers])

feature_extractor=keras.Model(inputs=model.inputs,
outputs=outputs_dict)
```

The next step is to define content loss. This will ensure that the top layer of the VGG19 model is similar enough to the base image and the combination image. In essence, it determines how different the macrostructures of the base image are from those of the combination image where the features from the reference image were applied.

```
def content_loss(base_img, combination_img):
        return tf.reduce_sum(tf.square(combination_img - base_img))
```

Now, it is important to compute the style loss. Another auxiliary function, called gram_matrix(x) is necessary for this step. A Gram Matrix is a matrix (of size SxS) which contains the dot products for all of the flattened feature vectors from a convolutional feature map (of size S). These dot products illustrate how similar two vectors are, such that the lower the dot product the higher the difference is between the vectors, and, similarly, higher the dot product, the lower the difference is between the vectors. This reveals information only about an image's style, since the feature has been flattened and performed a dot product on top of it. This auxiliary function is then used by `style_loss(style_image, combination_img)` to compare the differences in their feature vectors [12]. This determines how different the style of the reference image is from the style of the combination image.

```
def gram_matrix(x):
    x=tf.transpose(x, (2, 0, 1))
    features=tf.reshape(x, (tf.shape(x)[0], -1))
    gram=tf.matmul(features, tf.transpose(features))
    return gram

def style_loss(style_image, combination_img):
    S=gram_matrix(style_image)
    C=gram_matrix(combination_img)
    channels=3
    size=img_height * img_width
    return tf.reduce_sum(

            tf.square(S - C)) / (4.0 * (channels ** 2) * (size **
            2)

        )
```

In addition to the content and style loss calculations, the total variation loss is required to compute the "spatial continuity" [5] in the combination image to avoid having an overly pixelated result. This is to ensure that the perspectives of the objects in the combination image are not too different from those in the original. To do this it uses edge detection to find the appropriate differences between pixels [13].

```
def total_variation_loss(x):
    a=tf.square(x[:, : img_height - 1, : img_width - 1, :] - x[:,
1:, :img_width - 1, :])
    b=tf.square(x[:, : img_height - 1, : img_width - 1, :] - x[:, :
img_height - 1, 1:, :])
    return tf.reduce_sum(tf.pow(a + b, 1.25))
```

Now we average the information from all of these loss functions, to compute the general loss. First, it is necessary to identify the layers used to compute the content and style losses. The main difference in this aspect of this step is the fact that the content loss was computed using only the uppermost layer, while the style loss was computed using multiple layers that spanned both the low and high-level layers. The key feature of this to consider is that it is recommended that the content_weight be amended according to reference and base images as the higher the content_weight, the greater the likelihood that the macrostructures in the base image will be recognizable in the combination image [5]. This is because this will place higher importance on the accuracy of the content, not necessarily the style, which will make the combination image more recognizable from the base image.

```
style_layer_names=["block1_conv1",
                   "block2_conv1",
                   "block3_conv1",
                   "block4_conv1",
                   "block5_conv1"]
content_layer_name="block5_conv2"
total_variation_weight=1e-6
style_weight=1e-6
content_weight=2.5e-8

def compute_loss(combination_image, base_image,
style_reference_image):
```

```
    input_tensor=tf.concat([base_image, style_reference_image,
combination_image], axis=0)
    features=feature_extractor(input_tensor)
    loss=tf.zeros(shape=())
    layer_features=features[content_layer_name]
    base_image_features=layer_features[0, :, :, :]
    combination_features=layer_features[2, :, :, :]
    loss=loss + content_weight * content_loss(base_image_features,
combination_features)
    for layer_name in style_layer_names:
        layer_features=features[layer_name]
        style_reference_features=layer_features[1, :, :, :]
        combination_features=layer_features[2, :, :, :]
        style_loss_value=style_loss(style_reference_features,
combination_features)
        loss+=(style_weight / len(style_layer_names)) *
style_loss_value
    loss+=total_variation_weight *
total_variation_loss(combination_image)
    return loss
```

The last step is to set up the gradient-descent process. An SGD optimizer is used to decrease the learning rate, that way the program will make very fast progress initially, and then get gradually more cautious as it approaches the loss minimum.

```
def compute_loss_and_grads(combination_image, base_image,
style_reference_image):

      with tf.GradientTape() as tape:

            loss=compute_loss(combination_image, base_image,

                  style_reference_image)

            grads=tape.gradient(loss, combination_image)

            return loss, grads


optimizer=keras.optimizers.SGD(

      keras.optimizers.schedules.ExponentialDecay(initial_learning_r
      ate=100.0, decay_steps=100, decay_rate=0.96)

)
```

Now that the final step is completed, the last thing to do is to store and save the combination images. The combination image is stored in a variable as it is being updated during the training process. Next, the image is updated to reduce the style transfer loss and each combination image is saved at regular intervals.

```
base_image=preprocess_image(base_image_path)
style_reference_image=preprocess_image(style_image_path)
combination_image=tf.Variable(preprocess_image(base_image_path))
iterations=4000
for i in range(1, iterations + 1):

    loss, grads=compute_loss_and_grads(combination_image,

        base_image, style_reference_image)

    optimizer.apply_gradients([(grads, combination_image)])

    if i % 100 == 0:

        print(f"Iteration {i}: loss = {loss:.2f}")

        img=deprocess_image(combination_image.numpy())

        fname=f"combination_image_at_iteration_{i}.png"

        keras.utils.save_img(fname, img)
```

By the end of this process, the result should appear as follows:

Figure 2: Style/Reference Image

Figure 3: Content/Base Image

Figure 4: Combination Image

Chapter 3: The Impact of Neural Style Transfer Learning on the Production of NFTs

The fact is the underlying structure for Neural Style Transfer Learning is commonly available. Therefore, regardless of the mechanism, it is completely possible that this could be implemented with an accessible user interface that would allow anyone to create their own unique images to be sold as an NFT on the blockchain. This can be connected to one of the greatest controversies regarding NFTs. Recently, there has been a growing interest in the legal rights regarding new and preexisting NFTs. One of the most recent legal cases regarding NFTs was between Hermès International SA and Mason Rothschild. In the lawsuit, Hermès claimed that Mason Rothschild's MetaBirkins NFTs violated their trademark rights in "the Birkin word mark and design" [14]. Rothschild argued that, because of the artistic nature of his NFTs, the claims made against him were invalidated by the First Amendment. In the end, the jury decided in favor of Hermès and found Rothschild liable for trademark infringement, trademark dilution, and cybersquatting. Cybersquatting is the "unauthorized registration of and use of Internet domain names that are identical or similar to trademarks, company names, or personal names … with the bad faith intent to profit from the goodwill of the actual trademark owner" [15] which is federally prohibited by the Anticybersquatting Consumer Protection Act (ACPA) [15]. The outcome of this case could shape the way future NFT trademark questions are handled. However, it is important to understand that, in this case, the jury's necessary findings that Hermès had a famous mark and that Rothschild acted in bad faith might not necessarily be present in every case following this. However, this case sets an important precedent when it comes to the legal rights of NFTs [16].

Another example of this is the Yuga Labs, Inc. v. Ripps *et al.* case. While this case is still ongoing, the progress of the case thus far is very relevant to the rights of NFT creators and original artists. In this case, Yuga Labs, the $4 billion company behind the dominant NFT brand Bored Ape Yacht Club (BAYC) has sued the conceptual artist Ryder Ripps and his cohorts. In 2022, Ripps claimed that Yuga's BAYC NFTs contained racist, pro-nazi imagery. Ripps later proceeded to sell a copycat collection of 10,000 Bored Ape NFTs which shared the same names, style, and features as the originals released by Bored Ape [17]. Yuga came to a settlement agreement with Thomas Lehman who was responsible for "coding and developing the 'RRBAYC RSVP Contract,' and 'rrbayc.com' site to sell the copycat NFTs" [18], while they are still involved in the suit against Ripps for Intellectual Property Infringement. An important part of this case to note is that, in October 2022, Ripps filed "an anti-SLAPP motion arguing that his project is protected by free speech as a work of satire" [18] which was denied two months later.

Ripps' team have filed an appeal against that decision as well as their own counterclaims against Yuga.

These cases are just a few examples of how NFTs have perforated the legal sphere and made law and precedent makers question existing legislation and their applicability to this new type of technology. The fact that Neural Style Transfer learning takes the style of one image and uses it to recreate an existing image, could pose many problems for original artists and NFT creators alike. As seen, there are many legal risks presented by the use of this kind of technology.

Generally speaking, however, there are many costs and benefits to NFTs. Some benefits of NFTs are that they allow fractional ownership of a product [19]. This means that NFTs can be applied to real-world commodities in a way that allows users to invest in products they otherwise would not have financial access to. While these fractional NFTs allow more people to invest in certain assets, there are some conditions that may make it difficult for them to be readily used and available. One of the primary potential barriers to the use of fractional NFTs is regulators. Anthony Clark from *Cointelegraph* writes, "since fractional NFTs let people own a fraction of an asset, they could be classed as stocks by the United States Securities and Exchange Commission (SEC)" [20]. This means that creators of fractional NFTs need to create and present their fractional NFTs, so it isn't considered an investment product (i.e., security). This is because securities are considered fungible means of earning money, where NFTs are interchangeable and non-fungible. If they are viewed as fungible securities by the SEC, they would be legally required to be registered with the SEC which would contain all relevant information about the seller and the offering made to investors. This could pose a great issue for developers who are involved in a community that values anonymity and decentralization [21]. The current vagaries surrounding the taxation and legal status of fractional NFTs could pose a definite threat to their usability.

Another benefit to using NFTs is that creators are guaranteed ongoing royalties on their work no matter how many times it is sold. Since these digital transactions are made using a smart contract, the terms of the contract including the royalties incurred for the original creator are also automatically applied. According to Adam Chernichaw, a blockchain attorney for the law firm White and Case, the smart contract is “a method of pushing a portion of the resale proceeds back to the original creator” [22]. While the name can be a bit misleading as there are still questions about how binding this smart contract is, it is executable code that the creator could build, such that, in the case of a specific event – such as a sale – this code could transfer a portion of the resale value back to the creator. However, one of the limitations of this is that the smart contract is only applicable if the NFT is sold on the same platform it was initially sold on. As such, depending on the terms of the platform used, the NFT could be transferred to another platform where the contract may not be executed [22].

As such, some of the limitations and drawbacks of NFTs come from the vulnerabilities in the smart contracts. Some of these vulnerabilities come from the fact that these contracts are unchangeable and thus, any error in the code would be very time-consuming and expensive to correct [23]. It is also from these contracts that insecurities can be introduced into the blockchain allowing hackers to steal hundreds of millions of dollars in NFTs [24]. The questionable security of these blockchains pose a significant problem for users.

Another issue involved with NFTs is the price, and therefore the accessibility, of being able to mint an NFT. The Ethereum blockchain is a particularly popular platform for the buying and selling of NFTs, however, users who wish to mint and sell and NFT on this platform, are required to pay what is known as a gas fee [19]. These gas fees are the amount of Ether needed to conduct a transaction, and so, users must pay this fee – which includes a tip – to compensate Ethereum miners for their work in authorizing transactions and maintaining the security of the network [25]. As we can see, there are many benefits and drawbacks to using blockchains to buy and sell NFTs. Many of these issues appear to be due to the fact that the technology is so new, there is no infrastructure to ease and manage many of the issues – both legal and technological – involving NFTs.

Chapter 4: Related Works

Neural Style Transfer Learning is already being discussed in this same context. While it has yet to be peer-reviewed, "Applying Neural Style Transfer to Transform Images into a Different Style Domain by the NST Method", by Vishnukumar et al., is also discussing Neural Style Transfer Learning. In this paper, Neural Style Transfer Learning is not only discussed generally but it also briefly discusses the algorithm in the context of digital art, and the application of this program to the "application of various aesthetics" [26]. They also seek to show that Neural Style Transfer Learning can be used with various output picture sizes and that, when compared to other newer versions of the Neural Style Transfer algorithm, "their model's colorization has better visual quality and is more in line with the specified style reference" [26]. One of the features that this paper says could be explored more in depth in future research is the ability to alter the style image to content image ratio.

We can see both Neural Style Transfer Learning and the ability to adjust the style image to content image ratio in apps such as Prisma. This app allows the user to import an image and, using one of Prisma's art filters, apply it to their photos and adjust the intensity of the new style.

Deep Art Effects is another software that uses Neural Style Transfer Learning to help the user customize their photos. While this software also has the ability to adjust the intensity of the transfer, it incorporated some digital image processing to allow the user to adjust the contrast, blur, saturation, hue, greyscale, mirror, or preserve the original colors. It also allows the user to upload a new reference photo to use as their style image template, and the software will learn the image and its style feature to apply to their content image [27]. This technology is also applied – though in a different way – to the popular Dall•E 2 program by OpenAI. This program uses neural networks similar to those in Neural Style Transfer Learning to generate photos from text descriptions. It also has the capability to accept a photo as input and use its AI system to create new versions of the image from different angles or in different styles. The Dall•E 2 program was created by training neural networks on images and their associated text descriptions so that it understands how to distinguish different objects and understand the relationships between objects [28]. In order to be able to apply different styles to input images, they, by definition, must have gone through a similar process as in Neural Style Transfer Learning. The main difference is that the program is already pre-trained on examples of these other styles of art, so the largest part of the program is applying it to the content of the input image. OpenAI is also very transparent about their efforts to keep Dall•E 2 safe. They claim the program is not trained on any explicit information, so the program quite literally doesn't understand harmful concepts. It is also not trained on images of real people's faces – particularly those of public figures – so the technology can't be used to generate explicit, hateful, or mature images. They also back up this practice with both automated and human monitoring to curb any misuse of the product [29].

Considering the rapid growth of the markets for NFTs since the COVID-19 Pandemic began, all aspects of the use and development of Neural Style Transfer learning in the generation and dissemination of NFTs are extremely important. The volatility of the market, the general advantages and disadvantages of participating in blockchain transactions, and the issues of intellectual property infringements –which are exceedingly relevant to this technology – should all be taken into account. This technology can be used responsibly and with the best intentions possible, however, there are many risks that can come with transferring the style of one image to the content of another. In the example of Neural Style Transfer Learning shown and explained in Chapter 2, the program had been altered to only transfer the styles of impressionist painters. However, if this technology had no limits on what images it could accept to extract style features from, it could quickly become an intellectual property dispute between an aspiring NFT creator and a very real artist with their own unique style that they could be trying to monetize. The deciding factors are going to be how this technique is implemented and if it is done so in a way to minimize the legal and monetary risks for everyone involved. As we can see from some of the cases examined in Chapter 3, legal systems are still trying to make sense of how to deal with these issues and translate concepts of intellectual property rights in the digital sphere to that of the legal.

References:

[1] M. Nadini, L. Alessandretti, F. Di Giacinto, M. Martino, L. M. Aiello, and A. Baronchelli, "Mapping the NFT revolution: Market trends, Trade Networks, and visual features," *Scientific Reports*, vol. 11, no. 1, 2021.

[2] J. Ghosh, "The blockchain: Opportunities for research in Information Systems and Information Technology," *Journal of Global Information Technology Management*, vol. 22, no. 4, pp. 235–242, 2019.

[3] "What is blockchain technology - IBM Blockchain," *IBM*. [Online]. Available: https://www.ibm.com/topics/blockchain. [Accessed: 17-Mar-2023].

[4] J. Creighton, "NFT timeline: The Beginnings and History of NFTs," *NFT Now*, 26-Jan-2023. [Online]. Available: https://nftnow.com/guides/nft-timeline-the-beginnings-and-history-of-nfts/. [Accessed: 02-Apr-2023].

[5] F. Chollet, *Deep Learning with Python*, 2nd ed. Shelter Island, USA, NY: Manning Publications, 2022.

[6] L. Gatys, A. Ecker, and M. Bethge, "A neural algorithm of artistic style," *Journal of Vision*, vol. 16, no. 12, p. 326, Sep. 2016.

[7] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *International Conference on Learning Representations*, Apr. 2015.

[8] J. Brownlee, "How do convolutional layers work in Deep Learning Neural Networks?," *MachineLearningMastery.com*, 16-Apr-2020. [Online]. Available: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/. [Accessed: 05-Apr-2023].

[9] A. Kaushik, "Understanding the VGG19 architecture," *OpenGenus IQ: Computing Expertise & Legacy*, 26-Feb-2020. [Online]. Available: https://iq.opengenus.org/vgg19-architecture/#:~:text=VGG19%20is%20a%20variant%20of,VGG19%20has%2019.6%20billion%20FLOPs. [Accessed: 11-Apr-2023].

[10] "Convolutional Neural Networks," *Weights & Biases – Developer tools for ML*. [Online]. Available: https://wandb.ai/site/tutorial/convolutional-neural-networks. [Accessed: 11-Apr-2023].

[11] M. Basavarajaiah, "Which pooling method is better? Maxpooling vs Minpooling vs average pooling," *Medium*, 22-Aug-2019. [Online]. Available: https://medium.com/@bdhuma/which-pooling-method-is-better-maxpooling-vs-minpooling-vs-average-pooling-95fb03f45a9#:~:text=Average%20pooling%20method%20smooths%20out,lighter%20pixels%20of%20the%20image. [Accessed: 16-Mar-2023].

[12] R. Asokan, "Neural networks intuitions: 2. dot product, Gram Matrix and Neural Style Transfer," *Medium*, 21-Jun-2019. [Online]. Available:

https://towardsdatascience.com/neural-networks-intuitions-2-dot-product-gram-matrix-and-neural-style-transfer-5d39653e7916. [Accessed: 14-Mar-2023].

[13] G. Singhal, "Gaurav Singhal," *Pluralsight*, 03-Jun-2020. [Online]. Available: https://www.pluralsight.com/guides/implementing-artistic-neural-style-transfer-with-tensorflow-2.0. [Accessed: 16-Mar-2023].

[14] Y. Parsafar and C. B. Westmoreland, "Takeaways from the Hermès litigation over Metabirkins nfts," *The National Law Review*, 08-Mar-2023. [Online]. Available: https://www.natlawreview.com/article/takeaways-herm-s-litigation-over-metabirkins-nfts. [Accessed: 14-Mar-2023].

[15] "What is the definition of cybersquatting?: Winston & Strawn Legal glossary," *Winston & Strawn*. [Online]. Available: https://www.winston.com/en/legal-glossary/cybersquatting.html. [Accessed: 14-Mar-2023].

[16] M. Peltz, "Hermès v. Rothschild: A landmark decision for trademarks and nfts," *Falcon Rappaport & Berkman LLP*, 10-Feb-2023. [Online]. Available: https://frblaw.com/hermes-v-rothschild-a-landmark-decision-for-trademarks-and-nfts/#:~:text=On%20February%208%2C%202023%2C%20the,Herm%C3%A8s'%20protectable%20intellectual%20property%20rights. [Accessed: 15-Mar-2023].

[17] S. Lutz, "Yuga Labs settles with Ryder Ripps collaborator in trademark suit," *Decrypt*, 06-Feb-2023. [Online]. Available: https://decrypt.co/120685/yuga-labs-settles-with-ryder-ripps-collaborator-in-trademark-suit. [Accessed: 15-Mar-2023].

[18] T. Bochan, "Yuga Labs reaches settlement in bored ape nfts trademark lawsuit," *CoinDesk Latest Headlines RSS*, 07-Feb-2023. [Online]. Available: https://www.coindesk.com/web3/2023/02/06/yuga-labs-reaches-settlement-in-bored-ape-nft-trademark-lawsuit/. [Accessed: 15-Mar-2023].

[19] D. Team, "Everything You Should Know About NFTs — Meaning, Pros, and Cons Explained," *Daisie Blog*, 28-Jun-2022. [Online]. Available: https://blog.daisie.com/everything-you-should-know-about-nfts-meaning-pros-and-cons-explained/#:~:text=Advantages%20of%20NFTs%20include%20fractional,of%20imitation%20projects%20and%20fraud. [Accessed: 21-Mar-2023].

[20] A. Clarke, "Fractional nfts and what they mean for investing in real-world assets," *Cointelegraph*, 09-Nov-2022. [Online]. Available: https://cointelegraph.com/news/fractional-nfts-and-what-they-mean-for-investing-in-real-world-assets. [Accessed: 21-Mar-2023].

[21] L. Thomas, "Fractional nfts: The good, the bad, and the weird," *nft now*, 26-Apr-2022. [Online]. Available: https://nftnow.com/features/good-bad-weird-about-fractional-nfts/. [Accessed: 21-Mar-2023].

[22] J. Cohen, A. Chernichaw, and P. Vallabhaneni , "How do NFT royalties work?," *TalksOnLaw*, TalksOnLaw, 18-Jun-2021.

[23] C. F. I. Team, “Smart Contracts,” *Corporate Finance Institute*, 08-Dec-2022. [Online]. Available: https://corporatefinanceinstitute.com/resources/valuation/smart-contracts/. [Accessed: 02-Apr-2023].

[24] D. Parmar, “Challenges and Risks Associated with NFTs,” *Geekflare*, 08-Dec-2022. [Online]. Available: https://geekflare.com/nfts-challenges-and-risks/. [Accessed: 02-Apr-2023].

[25] W. Duggan, “Ethereum Gas Fee Definition,” *U.S. News*, 15-Sep-2022. [Online]. Available: https://money.usnews.com/investing/term/ethereum-gas-fee. [Accessed: 02-Apr-2023].

[26] A. Vishnukumar, K. E, S. Gugan, A. G. A, and M. T, “Applying neural style transfer to transform images into a different style domain by the NST method,” *Research Square*, 2023.

[27] *Create Unique & Digital Works Of Art In No Time*. Deep Art Effects.

[28] *Dall•E 2*. OpenAI.

[29] “Dall·E 2,” *DALL·E 2*. [Online]. Available: https://openai.com/product/dall-e-2. [Accessed: 05-Apr-2023].