

D-ARTNET22 V1: A Neural Network Framework Against Stolen
Digital Artworks Getting Non-Fungible Token (NFT) Labels.

by

Farhan Hasin Dipro

18101627

Sheikh Saif Simran

18201189

Mansura Akhter

18301031

Ramisa Fariha Momo

18301034

Ramisa Musharrat

18301233

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
May 2022

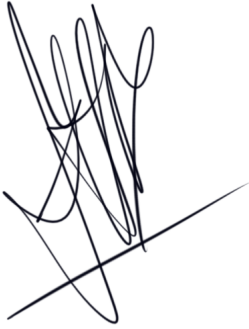
© 2022. Brac University
All rights reserved.

Declaration

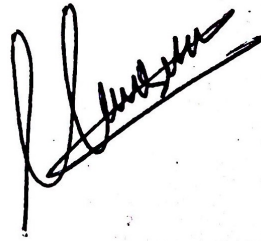
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

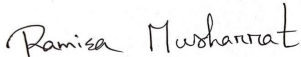
Student's Full Name & Signature:



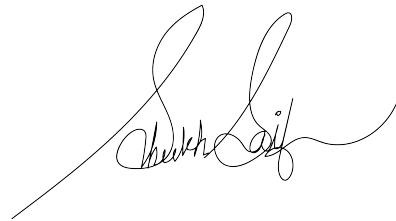
Farhan Hasin Dipro
18101627



Mansura Akhter
18301031



Ramisa Musharrat
18301233



Sheikh Saif Simran
18201189



Ramisa Fariha Momo
18301034

Approval

The thesis/project titled “D-ARTNET22 V1: A Neural Network Framework Against Stolen Digital Artworks Getting Non-Fungible Token (NFT) Labels.” submitted by

1. Farhan Hasin Dipro (18101627)
2. Sheikh Saif Simran (18201189)
3. Mansura Akhter (18301031)
4. Ramisa Fariha Momo (18301034)
5. Ramisa Musharrat (18301233)

Of Spring, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 26, 2022.

Examining Committee:

Supervisor:
(Member)



Dr. Muhammad Iqbal Hossain
Associate Professor
Department of Computer Science and Engineering
Brac University

Co Supervisor:
(Member)



Ms. Ahanaf Hassan Rodoshi
Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Assistant Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Since NFT sites gained fame for selling digital arts, NFT crimes have taken a toll on excessive amounts of digital content creators as stolen digital artworks get their ownership changed permanently in the name of the thief, further getting sold on humongous fortunes. Due to NFT sites not having any user or content verification system before registration, thieves tend to take the chance of scamming even more by adopting various forgery protocols. Artworks from social media or NFT sites are stolen, forged, and then registered under different names. On the contrary, since blockchains are immutable, the thief remains the owner of the stolen NFT forever which implies that NFT sites fail to provide a secure space for hardworking digital content creators. According to what has been researched, it is discovered that there exists no such work relating to digital media. Despite connecting some certain disjoint fields, the results were not promising and thus they were not thought to be implemented in real life. Besides, digital artwork datasets are not available online for the purpose of this field to be served. A possible methodology can be doing extensive image scraping on selective digital media platforms to extract digital artworks that may then be modified to create a fabricated artwork dataset. This dataset can subsequently be used to train deep learning or neural network models to distinguish between actual and false entities. As no verification system for NFT sites has been proposed before, it is crucial to develop a system to check the authentication of digital artworks and the user before the NFT transaction is passed into the blockchain. Therefore, for the very first time, this paper will present a framework that will check the originality of digital artworks before accepting them as an NFT permanently.

Keywords: Non-Fungible Tokens (NFT), Digital Art, Convolutional Neural Networks (CNN), You Only Look Once (YOLO), Adobe Photoshop CC, Image Forgery, Image Classification, Object Detection.

Dedication (Optional)

Creativity is a skill that everyone can practice in their own unique ways. One of the very fortunate beauties lies within the mind, the imagination that an artist can portray in forms of words, colors, tunes, and many more. It requires hours and hours of practice, failures, and experiences to form that level of strength. With that, this world is made a more beautiful place to live in. This field of work is dedicated to those people who are on their way to being virtuosos, having immense hard work, and support from family and friends. People who embrace their passion for creativity, and enhance their enthusiasm for it have had tons of inspiration behind it, only for the great masters who have contributed priceless entities of art to this world. What comes in their way is the dark world of robbery, taking things without the concern of the owner and selling in currencies of humongous fortunes in the most inappropriate ways possible. This field of work also implies the fact that anyone can contribute to making this world a beautiful place, so one must not, therefore, come as an odd experience to someone's eternity of hard work.

Table of Contents

Declaration	i
Approval	ii
Abstract	iv
Dedication	v
Table of Contents	vi
List of Figures	viii
List of Tables	x
Nomenclature	xii
1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement	2
1.3 Research Objectives	4
2 Literature Review	5
2.1 Background	5
2.1.1 World of Digital Art and the emergence of Non-Fungible Tokens (NFT)	5
2.1.2 How NFTs are registered	6
2.1.3 Creating the first-ever digital art dataset	7
2.1.4 Real/Fake Classification using Convolutional Neural Networks (CNN)	7
2.1.5 Object detection using YOLO (You Only Look Once)	8
2.2 Related Works	8
3 Methodology	18
3.1 Workplan	18
3.2 Preparation of the Dataset	20
3.2.1 Image Scraping	20
3.2.2 Creating Fake Images	21
3.2.3 Compiling the dataset	28

4	Implementing the framework	30
4.1	Data Curation	30
4.2	Classifying real and fake images using existing CNN models	31
4.2.1	Training the models	32
4.2.2	Testing/Running Prediction	32
4.3	Introducing the custom model	32
4.3.1	The architecture of D-ARTNET22 V1	33
4.4	Implementing Object Detection	36
4.4.1	Object Detection Datasets	36
4.4.2	Input Data	37
4.4.3	Training the YOLOV5 model	37
4.4.4	Testing/Running Prediction	37
4.5	Maintaining Benign Information	38
4.5.1	Input Data	38
4.5.2	Data pre-processing	38
4.5.3	Training the SVM model	39
4.6	Displaying Ground Truth	39
4.7	Granting Permission	39
5	Results	41
5.1	CNN Models	41
5.1.1	Training	41
5.1.2	Testing/Running Prediction	44
5.2	YOLO-V5 model	48
5.2.1	Training	48
5.2.2	Testing/Running Prediction	49
5.3	SVM Classifier	51
5.3.1	Training	51
5.3.2	Testing/Running Prediction	51
6	Conclusion	52
6.1	Limitations	53
6.2	Future Works	53
	Bibliography	56

List of Figures

1.1	Weekly total cryptocurrency value and average value per transaction sent to NFT platforms in 2021.	3
1.2	Illicit value received by NFT platforms between 2020 and 2021 [34].	4
3.1	Work Plan Part-I: Preparation of the dataset.	21
3.2	Manifestation of plain forgery.	23
3.3	Manifestation of StyleGAN.	23
3.4	Manifestation of Copy-Move forgery.	24
3.5	Manifestation of Splicing.	24
3.6	Manifestation of Retouching.	25
3.7	Manifestation of Caricaturization.	25
3.8	Manifestation of Blurring.	26
3.9	Manifestation of Filtering.	26
3.10	Manifestation of Brightness/Contrast.	27
3.11	Manifestation of Recoloring.	27
3.12	Labeling of REAL and FAKE images.	29
3.13	CSV file of the dataset (Benign Information).	29
4.1	Work Plan Part-II: Structure of the framework.	30
4.2	Architecture of D-ARTNET22 V1.	34
4.3	Scenario of reporting a fraud NFT post.	40
5.1	AlexNet’s training accuracy and loss against epochs	42
5.2	EfficientNet-B0’s training accuracy and loss against epochs	42
5.3	ResNet-50’s training accuracy and loss against epochs	42
5.4	VGG-16’s training accuracy and loss against epochs	43
5.5	MobileNet-V1’s training accuracy and loss against epochs	43
5.6	D-ARTNET22 V1’s training accuracy and loss against epochs	43
5.7	Prediction Accuracy of AlexNet for all forgery classes	45
5.8	Prediction Accuracy of EfficientNet-B0 for all forgery classes	45
5.9	Prediction Accuracy of ResNet-50 for all forgery classes	45
5.10	Prediction Accuracy of VGG-16 for all forgery classes	46
5.11	Prediction Accuracy of MobileNet-V1 for all forgery classes	46
5.12	Prediction Accuracy of D-ARTNET22 V1 for all forgery classes	46
5.13	Prediction Accuracy of D-ARTNET22 V1 for one test image.	47
5.14	YOLOV5’s training accuracy and loss against epochs for ARTIFICE21 V1	48
5.15	YOLOV5’s training accuracy and loss against epochs for PASCAL VOC	48

5.16	Prediction accuracy of YOLOV5 when trained with each dataset. . .	49
5.17	Output of YOLOV5 when trained with ARTIFICE21 V1	50
5.18	Output of YOLOV5 when trained with PASCAL VOC	50

List of Tables

3.1	Number of images from each target site.	20
3.2	Percentage breakdown of the number of images each forgery category holds.	22
3.3	Number of images belonging to each Photoshop category.	22
4.1	D-ARTNET22 V1 Architecture	35
4.2	Analysis of the number of objects.	38
5.1	Training accuracies and time taken for all CNN models	41
5.2	Average prediction accuracies of all chosen CNN models.	44
5.3	Prediction accuracies of all forgery categories by each CNN model.	44
5.4	Training accuracies of YOLOV5 model on each dataset	48
5.5	Prediction accuracy of YOLOV5 when trained on each dataset.	49
5.6	Performance of the SVM Classifier.	51

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

- 2D* Two-Dimensional
- 3D* Three-Dimensional
- CC* Creative Cloud
- CMFD* Copy-Move Forgery Detection
- CNN* Convolutional Neural Networks
- COCO* Common Objects in Context
- Conv* Convolutional
- CPU* Central Processing Unit
- CRF* Conditional Random Field
- CSPNET* Cross Stage Partial Network
- CSV* Comma Separated Values
- D – ARTNET* Digital-Art Neural Network
- DCN* Deformable Convolutional Layers
- DI* Difference Images
- EMA* Exponential Moving Average
- FERET* Facial Recognition Technology
- FRGC* Facial Recognition Grand Challenge
- GAN* Generative Adversarial Networks
- GB* Gigabyte
- GIF* Graphics Interchange Format
- GPU* Graphical Processing Unit
- HOG* Histogram of Oriented Gradients

IM Illuminant Map
JPEG Joint Photographic Experts Group
LFR Laplacian Filter Residual
MTCNN Multi-Task Cascaded Convolutional Neural Networks
NFT Non-Fungible Tokens
NMS Non Maximum Suppression
PNG Portable Network Graphics
PP – YOLO Paddle-Paddle You Only Look Once
R – CNN Region-based Convolutional Neural Networks
RAM Random Access Memory
ReLU Rectified Linear Activation Function
ResNet Residual Neural Network
RGB Red Green Blue
ROI Region Of Interest
SDMFR Second Difference of Median Filter
SGD Stochastic Gradient Descent
SRM Spatial Rich Model
SSD Single Shot MultiBox Detector
SSIM Structural Similarity Index
StyleGAN Style Generative Adversarial Networks
SVM Support Vector Machines
txt Text
URL Uniform Resource Locator
v1 Version 1
v2 Version 2
v3 Version 3
v4 Version 4
v5 Version 5
VGG Visual Geometry Group
VOC Virtual Object Classes
YOLO You Only Look Once

Chapter 1

Introduction

1.1 Motivation

The world-embracing technology nowadays brings a whole lot of chaos with it. As obvious as it is, online security regarding user belongings is still an unsolved issue despite having such improved methodologies. Micheal Mirafior, a media strategist, had all his NFT collections gone within a few minutes on Nifty Gateway, a famous online platform for buying and selling non-fungible tokens. The hacker used his account to buy artworks, hence transferring them into different accounts and selling them on Discord. After reporting the issue, Mirafior received his money back but all his artworks were lost forever [31]. Likewise, Derek Laufman's artwork was stolen and minted as an NFT on Rarible.com, listed for sale via his own verified ID without his concern. Before he came to know about it, the artwork was already sold. He claimed: "This is 100% NOT me. I thought the point of NFT was that the artwork and artists needed to be verified? Apparently super easy to scam people" [32]. Delvin Elle Kurtz found that her five-year-old work from her DeviantArt account was stolen and made it to the front page of Marble Cards, a famous NFT site. The person who turned her work into an NFT had it framed and watermarked all over it. Even though the image was removed, the frame remained a permanent part of the blockchain.

Non-fungible tokens are scarce virtual collectibles like Digital Art, Music, GIFs, videos, etc. and when a transaction takes place, it gets permanently written into the Ethereum blockchain. Once this happens, the information regarding an NFT cannot be modified or deleted as blockchains are immutable.

Many NFT platforms like OpenSea and Rarible do not have a strong authentication checking system before accepting a crypto art as an NFT. However, Rarible urges their buyers to do their research to verify if a digital artwork is authentic or not. As a result, the originality of an NFT is alleviated. Moreover, it imposes a great loss to the victim as the victim's work is misused by the thief to earn money. Despite blockchain technologies offering promising security, it does not, unfortunately, escalate to the various entities that exist within it, namely, applications or websites. Consequently, this lack of security and verification in NFT sites is increasing the rate of NFT thefts giving wrong people a chance to make easy money.

To prevent this problem a neural network framework is brought into thoughts that will block any stolen digital artwork before it gets added to the blockchain. This framework will use object detection and image classification features to detect if the image to be posted is real or stolen with certain forgeries performed. There are many fine art resources available on the internet that can be used for machine learning purposes, but as NFTs are mainly digital assets, the lack of digital artwork datasets imposes a huge obstacle.

NFT thefts are a rising issue that is not publicly known until someone is victimized as stealing artworks from various social platforms and selling them as NFTs without the owner's knowledge is easy, whereas, on the other hand, blockchain immutability stands in the way of retrieving originality.

1.2 Problem Statement

Despite NFTs being discovered in 2015, they pioneered fame very recently with the rise of technological abilities. Moreover, the ability of artists to afford and familiarize themselves with equipment to perform digital art is an added reason behind NFTs being famous nowadays. With these tools needed to perform digital art being so user-friendly and affordable, there is a chance for people with mid-level disposable incomes to take this opportunity and make a living.

Moreover, NFTs are any virtual asset that opens doors to any sort of digital content creator. On the contrary, the odds increase the same. The NFT platforms generally do not verify either if the owner is authentic or if the artwork is original or stolen. The system is such that it considers digital artwork coming its way as a unique piece and gives it a unique identity. Therefore, taking advantage of this, people tend to steal artworks from other artists' social media accounts such as Deviant Art, Art Station, Behance, Facebook, Instagram, etc., and then end up posting them on NFT sites to make money without the acknowledgment of the owner. Since it is not checked if the artwork is real or not, the emergence of powerful editing tools influences people to perform certain image forgery techniques to be on the safe side, with unfair means. In essence, specific objects from one digital artwork could be used in other ones or the outlook of the same artwork might be changed. Styles or colors might be manipulated to trick the human eye and lead normal users to believe that it is not a stolen digital artwork. Infact, a thief may also attempt to perform multiple forgeries in one image to make it more difficult to get detected. On the other hand, a different sort of theft case is digital artworks from one NFT site getting stolen and registered to some other NFT site with the owner being unaware of it. An artist has their emotions and deeper meanings behind their work as creativity defines it and this inflicts their thoughts on continuing their career. The rise of NFT crimes in recent times has heavily discouraged artists to invest in putting up NFTs or posting digital content to avoid them from getting stolen.

Consequently, this imposes a bigger problem. Since NFT transactions happen to take place in the Ethereum blockchain, they cannot be modified because of the im-

mutability property it holds. Thus, even if the real owner learns about the theft, nothing can be done to get the thief’s name off the NFT even if an NFT transaction does not take place, i.e: if the stolen digital artwork does not get sold. A lot of famous artists around the world have claimed NFT sites to be not safe by emphasizing the fact that no user verification is done. They also questioned how the authorities behind all of these allow it. The amount of money NFT sites hold from each transaction is a lot. The increasing number of theft cases has now become a genuine concern for digital content creators as to how much money can be earned by a thief if they remain undetected. Figure 1.1 below illustrates an idea of the amount of money NFT sites carry just within the span of a few days.

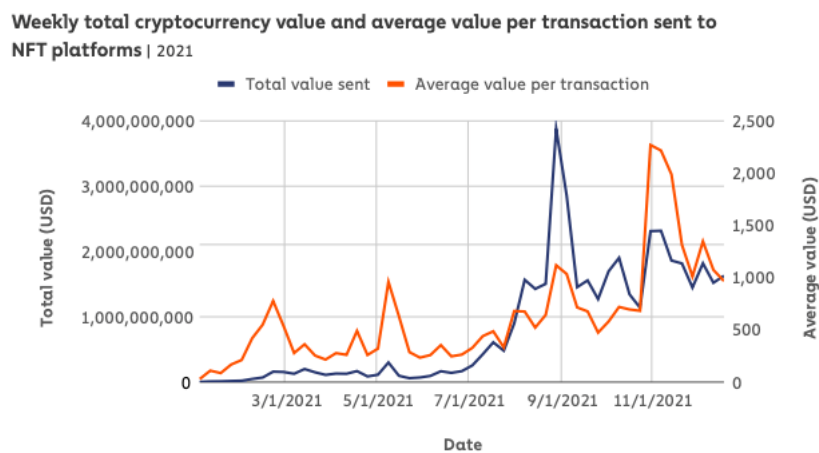


Figure 1.1: Weekly total cryptocurrency value and average value per transaction sent to NFT platforms in 2021.

The extent to which NFTs transactions take place daily consumes power equivalent to that of one whole country. This fact itself ponders the question of how many unauthorized, false NFT transactions are happening behind the scenes. Figure 1.2 shows various unlawful complaints received by NFT sites over the past 2 years.

There has been no work done to tackle this problem and this being a recent venture, a lot of people, mainly the people from the IT sector are not aware of this phenomenon yet. There are specific freelancing sites that have a verification phase to detect forgeries but NFTs being such an upbeat entity, do not, unfortunately, go through a checking phase. The existing image theft detection systems work on finding real and fake images where the dataset used was mostly photographs or fine arts in the context of the problem. However, digital art datasets are very much unprecedented. The existing works in related fields involved only classifying images or detecting objects with some available algorithms but both were never combined in such a case. The need for a deeper analysis of how artworks can be forged is needed in the works for the system to improve and tackle the effects that the future holds.

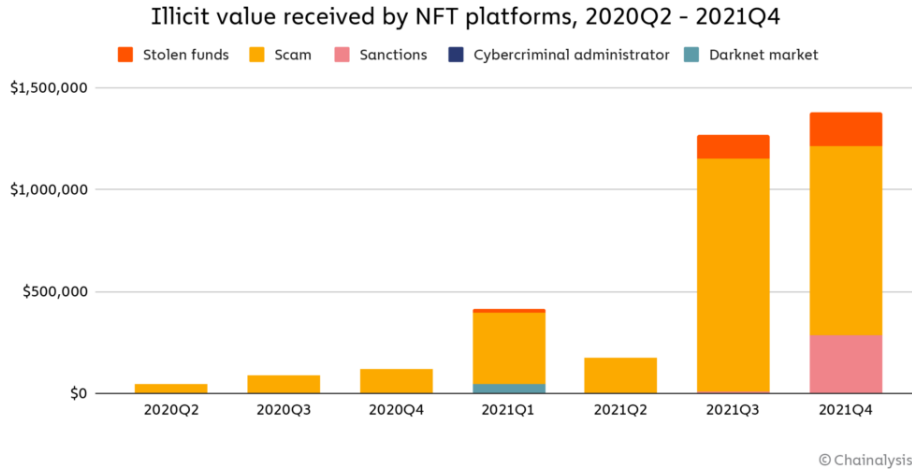


Figure 1.2: Illicit value received by NFT platforms between 2020 and 2021 [34].

1.3 Research Objectives

NFTs being a recent venture did not have anything done to get the odds tackled. This in turn opens doors to endless possibilities since NFTs hold any sort of digital assets. However, to start on a very basic level, digital artworks are the prime focus of this research. The objectives planned are listed below:

- Image scraping digital artworks throughout the internet followed by a series of curation steps.
- Performing image forgery techniques on the collected images to create fake entities.
- Use StyleGAN on a certain percentage of the collected images to create more fake entities to bring more variations.
- Create a digital art dataset combining the scraped and forged pairs hence providing the constructed dataset publicly for future research work.
- Classify as real/original and fake/fabricated by training a Neural Network model which will learn the styles of images and extract features from them.
- Using objection detection to analyze in-depth image properties. This can be used to incorporate the laws of digital property for accusing the theft.
- Merge the results of the object and style detection algorithms to run a sanity check.
- Locating the source of piracy, i.e: from which site it was taken and to which NFT site it was posted along with necessary metadata (Benign information of the real artist).
- Predict possible fabrication of the image and provide a decision of its allowance of submission in an NFT site.

Chapter 2

Literature Review

2.1 Background

2.1.1 World of Digital Art and the emergence of Non-Fungible Tokens (NFT)

People are all living in a digital world nowadays where remote work has been made possible starting from online shopping to online jobs. The digital art industry is not any different. Everything has its pros and cons and unlike fine art techniques, digital artworks create their uniqueness. Despite fine arts giving a live experience of texture and value, digital artworks do not fail to hold their aesthetics. Enthusiastic people who embrace the passion of art dwell in the kingdom of both worlds. The most obvious fine art techniques include Acrylic Media, Oil Painting, Watercolor Painting, Charcoal Sketch, and many more. While there is a chance of getting the environment dirty, digital art tools give a clean yet enriching experience with tools like Adobe Illustrator, Adobe Photoshop, Autodesk Sketchbook, etc. In recent times, if an artist wants to showcase their work, social media is the best option. Artists can create their accounts and have all their work displayed. This can at times serve as their portfolios and there are specific sites dedicated to that. The most common social media platforms used for showcasing digital art are ArtStation, DeviantArt, Instagram, Behance, etc.

Non-fungible token, as the name suggests is the token given to a digital asset or crypto art to certify it as a unique irreplaceable piece. In other words, for today's pop culture fanatics, NFTs are scarce virtual collectibles and some of the many use cases include Game skins (In-game items), Game weapons (In-game items), Digital Art, Music, Tweets, Video Clips of well known moments, Trading Cards, Virtual Land in gaming, etc.

Some of the common NFT sites are OpenSea, Rarible, NiftyGateway, etc. Trading of NFTs on the Ethereum Blockchain is done with cryptocurrency which is digital money that does not depend on banks to confirm the transactions. The most common cryptocurrency used on NFT sites is ETHER. In 2021, the NFT market has grown drastically and so have the NFT theft cases. The two possible types are stealing digital artworks from any social media site and posting them on an NFT site without the real owner's concern. Next, stealing artworks from one NFT site

and posting it to another NFT site. This way the real owners do not even know that their art is being stolen and some other person is misusing it to earn money.

The Ethereum Blockchain makes it safe to make transactions or transfer digital items as any movement of files will be recorded in the blockchain and will remain in the blockchain forever. After the transaction, the crypto art gets added to the owner's crypto wallet where all crypto art owned by that account's owner is listed [29]. However, if a thief happens to make a successful transaction, it cannot be retrieved due to the blockchain's immutability.

2.1.2 How NFTs are registered

NFT sites do not authorize the user or the artwork while someone enters through registration. The process of registering an NFT is described below followed by an explanation of the main dilemma.

1. **Creating a Crypto Wallet:** This crypto wallet is used to connect to different websites and spend tokens across the cryptocurrency network to buy entities or NFTs. A common platform for this is metamask. To commence, the app needs to be installed and an account needs to be created.
2. **Buying Cryptocurrency:** The crypto wallet consists of platforms such as Wire, Binance which can be used to buy cryptocurrencies like Ether through normal money that people contain within bank accounts. Next, they are needed to be deposited into the crypto wallet using the wallet address.
3. **Connecting the crypto wallet to an NFT site:** This step is required for a participant to be able to buy or sell NFTs from various NFT sites as the crypto wallet generates all transactions of any purchase/sell of an NFT. Thus, if someone wants to register content as an NFT, they require to connect their crypto wallet to the NFT site they choose to perform on.
4. **Minting NFTs and monetizing:** Minting any sort of digital content helps monetize them throughout the Ethereum Blockchain. Once, a content is minted, it gives permanent ownership to the registered user which then gets permanently written into the blockchain.

As it can be seen that there exists no verification process for either of the artworks or the users registering, it would be impossible to retrieve ownership once something is minted into the blockchain. The whole NFT registration process is very basic and is done as per the needs of cryptocurrency transactions but there is zero awareness about the originality of the content. Thus, something must come before this minting process to put a stop to it.

2.1.3 Creating the first-ever digital art dataset

The availability of datasets for this specific form of art is non-existent in contrast to numerous fine art datasets available online i.e. WikiArt, Paintings dataset, Picasso Art dataset, PeopleArt dataset, etc. but they are not eligible for accomplishing the required objectives.

It all needs to start with Image scraping which can be performed on the chosen NFT and social media sites respectively. According to the problem statement, it is the forgery techniques that need focus namely; Copy-move forgery, Image Splicing, Image Retouching, Image Filtering, and so on [18][14]. Thus, the implementation of certain levels of forgery techniques using Adobe Photoshop would be performed on the collected images.

However, with the increasing use of Photoshop nowadays, Artificial Intelligence is also used to create digital art. It would have been fine if someone generated a raw digital artwork from scratch by code but there are certain data augmentation techniques that can change the outlook of an image in such a way that it is unrecognizable. These technologies have gained fame with the emergence of Generative Adversarial Networks (GANs) which offer tonnes of variants to give thieves a wide range of opportunities to steal and create digital content. Thus, along with using Photoshop, the plan of this approach is to use StyleGAN. GANs are generally composed of a generator and a discriminator which constantly work in competition to fool each other to create variations in the input data. StyleGAN gives optimum performance and works by emphasizing style transfer features. In essence, if a picture is chosen to be changed via StyleGAN, a style is also chosen from another image. StyleGAN then uses style transfer features to convert the target image into the style of the chosen style. This completely changes the appearance of the newly generated fake image. Thus a certain percentage of fake images can be generated via StyleGAN to bring another set of forgery techniques that can provide diversity to the dataset that is being made.

2.1.4 Real/Fake Classification using Convolutional Neural Networks (CNN)

CNN's are a commonly chosen algorithm, for their ability to handle big datasets and their versatility to work with any kind of data. Having that said since the approach of the proposed framework requires analyzing visual representations, different incarnations of the CNN model can be used to perform the required analysis. A typical CNN model consists of several mannered neurons and its hidden layers have outdone traditional technologies which have indeed made it a specialty of CNNs. The basic features are extracted or learned about in the initial layers which are capable of measuring weights and biases. The model learns more complex features as it progresses through the other layers. The future layers help in learning the features by controlling certain parameters like dropping neurons, flattening the outputs of the convolutional layers in the fully connected layers, and using certain activation functions as per the type of result preferred. Progressively, it can detect patterns to

classify features and styles from various aspects to differentiate images. All in all, it constructs a hierarchical model to process the input images. Since the requirement of the framework is to study image forgery in detail, CNN is a perfect fit. The motif is to classify the various types of forgery that a thief can possibly perform but as the forgery categories are quite diverse and individually quite powerful, the CNN model for this approach would require a much more complex architecture with more number of layers so that the model is capable of learning the styles of all sorts of forgeries.

2.1.5 Object detection using YOLO (You Only Look Once)

Object detection is a sort of classification with the added feature of localizing the objects in the image. It generally alludes to the ability of algorithms to locate objects in images and identify them with a certain class. YOLO holds its fame for detecting and identifying certain objects in images or videos with real-time updates. It is by far the fastest object detection algorithm and has had its upgrades with its many versions recently. Training is done on the dataset with the images annotated, i.e., by assigning bounding boxes on objects in a picture. YOLO v3 makes use of features learned with the help of a deep convolutional neural network to detect an object. The first three versions were created by the same person and each version improves on assigning more bounding boxes. Next, v4 proved to outperform v3 improvising the mean average precision (mAP). Both v3 and v4 use DarkNet in their architecture which is eventually replaced by ResNet to form PP-YOLO (Paddle Paddle-Parallel Distributed Deep Learning). Its main goal was to balance effectiveness and efficiency in performance which proved to be right by surpassing v4's performance. However, v5 was later introduced with the algorithm improvising with mosaic data augmentation along with automatic learning bounding box anchors applied. It is said that this version is controversial and despite having extreme speed, competes on par with v4. The reason for choosing object detection in the proposed framework is to add some power that will make the evidence against the thieves much stronger. If the objects are detected and the coordinates of the bounding boxes are analyzed to see if there are changes, then this will act as a shred of supporting evidence with the class of forgery found by the CNN model.

2.2 Related Works

The problems regarding NFTs deeply relate to the consequences of a lot of things and no work has been done so far regarding stopping unverified post allowances, followed by checking if the artwork is stolen or fabricated. Thus, the related works found are as per the research objectives mentioned above which are thought to be the elementary steps towards an initial rendition of solving the problem. The findings in the related fields have their limitations for which the research objectives include improvising throughout the progress in terms of using a better version of an algorithm, changing the architecture of a certain model, and organizing the whole system to get a fruitful outcome.

Nemade et al. used a mixture of computer vision features to identify photos from a pool of seven painters with quite promising accuracy in performance [3]. They

discovered that HOG 2×2 had the overall best performance when evaluating the performance of different features since it can capture more balanced important spots across the pictures. Finally, the artists' tastes and styles may be reflected in the performances of the various features. To figure out a painter's style, the algorithm looks for a common thread that runs across all of the artists' works. The feature extraction stage is used to determine which components may be utilized to identify an artwork that is unique to it. The feature extraction process consists of creating feature descriptors that are used to describe an image or a patch of an image, with just the most important information retrieved and the rest discarded. This descriptor is used in the feature extraction step of image processing. HOG characteristics aren't as straightforward as they appear. They entail gradient computation, object binning, and histogram creation of features from blocks. A frequency chart is referred to as a histogram.

Smirnov et al. talked about the fact that digital art datasets do not exist on the internet and the fact that having fine art datasets like WikiArt, and Paintings datasets are not of any help while analyzing digital work [8]. Thus, they used basic data augmentation to incorporate certain art styles from the mentioned fine art datasets into normal images. Their working model architecture included two models of VGG-19 specifically used to detect certain art styles and objects respectively. The two CNN models work simultaneously and their output is combined to form a fusion of the features. This act is then followed as the input to the SVM model which declares the final result localizing the object in the artwork. However, despite achieving good results there are some limitations to this work. To start with, they used various datasets and the training and testing phase was randomly making it very unorganized and biased at times. Moreover, their compensation of introducing digitized images as datasets meant applying a certain art style into a normal image via data augmentation but this is not the same that could have been expected from a real digital artwork. As a result, their research still fails to hold the place for the absence of digital art datasets.

A combination of different neural network classifying methods to detect fake images created by GANs and humans without the use of any meta-data or image compressing data has been discussed by Tariq et al. [9]. A CNN-based model was used to separate the forged face images produced using GAN from the real images and state-of-art algorithms like MTCNN, YOLO, and SeetaFace to detect forged images generated via editing software like Photoshop. Meanwhile, due to the lack of a large range of complex human-created fake images dataset, the model did not give the best result for human-created fake images. However, the suggested framework here will not be limited to face regions only rather it would work with whole objects.

To detect GAN-generated fake images, a co-occurrence matrix on the RGB channel of images was developed and trained on deep neural network architectures by Nataraj et al.[15]. Two GAN-based approaches namely CycleGAN and StarGAN were used cross-changeably to implement generalization on their model. It has shown slightly low performance when StarGAN was trained and CycleGAN was tested due to the wide range of variance in the image category/sources of the CycleGAN dataset and the uneven distribution of class samples of StarGAN. The model

was also experimented with using compressed image information that often led to fake news generation or theft over social media. However, their model gave low accuracy for compressed data.

The image-to-image translation is the process of transforming an image of one domain into another domain. This technique to generate fake images is very powerful. Francesco et al. implemented several detection techniques on cycleGAN-generated paired images [6]. Deep neural network methods like InceptionNet, and XceptionNet showed high accuracy. They calculated the average accuracy for every category of paired images which allowed them to figure out the most challenging type of image-to-image translation. Furthermore, they did not train a particular category of an image set to ensure that classifiers can learn the common patterns of this type of generated image which will ultimately help in detection during real scenarios where the type of image translation is unknown.

A StyleGAN architecture and training methods have been reviewed by Varkarakis et al. and evaluation results are shared from retraining it on a number of different public facial datasets [24]. The quality and diversity of the generated samples are assessed via a comparative examination of a randomized set of data generated by each of the re-trained GANs. Some of the facial datasets include Labelled Faces in the Wild, CASIA-WebFace, CelebFaces, MegaFace, Ms-Celeb-1M, and VGGFace VGGFace2. An evaluation metric, Fréchet inception distance, is used that measures the similarity between the generated images and original images. Finally, the study was intended to develop a tool to show how the size and quality of the original dataset affect the quality and quantity of the final dataset that could be utilized to construct big, complex structures of synthetic facial data.

The main motivation of work for Kumar et al was to detect real or fake emotions using a CNN model for which they divided the basic emotions into individual classes and labeled them accordingly [2]. They divided the training segment into two parts, emotions and pixels where the "emotion" column consisted of the numbers the emotions have been labeled with, and the "pixels" column contained an array of pixels portraying the human face with the respective emotions. Next, for testing, the pixel column was fed to the classifier to detect the emotion columns, in essence, the labels of each emotion. Their model worked with a humongous amount of samples in their dataset and thus provided an optimum accuracy. Meanwhile, the necessary image pre-processing was not properly done as they used a very old algorithm, Viola-Jones, to detect faces as objects from the initial raw image captured via a webcam. Consequently, the CNN classifier did overshoot at times to provide a proper result.

Xiong et al. suggested that the inception module of deep network CNN architecture is presented where it detects separate structural information of an image by using specific kernels by categorizing images' similarities[10]. A new attention inception module is presented in this study, which pulls out features of an image synchronously from the convolutional method. By programmatically scattering between the attention inception models, the AI-NET is built by collecting the suggested attention inception module which can adjust and learn this construction. With fewer

trials, more representative features are learned by utilizing individual sizes for image convolutional filters and the CNN architecture. Here, within the preparing arrangement, 7×7 picture patches are extracted at each HSI pixel as the input images. Then the picture patches are given into a convolution layer with a 3×3 part measure and a max-pooling layer. Another, two attention inception modules with the left-over association are utilized here. A huge number of demonstration findings show that this suggested procedure shows better image identification.

Despite the fact that the Xception model is preferable for image identification, it hasn't been applied so often. To address this problem, Wu et al. suggested an Xception-based transmit learning model and compared its execution to that of the Inception-V3 model with transfer learning [26]. It is shown that the results on the Xception-based model perform more effectively than any other process such as Inception-V3. Additionally, the Xception has demonstrated improved resilience and generalization capabilities, with fewer overfitting issues. Here The presentation of the Xception and Inception-V3 models is thoroughly verified and contrasted. Also, By exploring the impact of the source datasets, it is shown the exchange learning strategy was capable of pretraining both important and apparently different source datasets, with the pertinent source dataset providing better classification precision than the presumably unrelated source dataset. The paper illustrates that the transfer learning procedure has enormous promise for identifying separate picture information sets with less image information. This study uses the Xception show to demonstrate the effectiveness of exchange learning in identifying certain datasets.

Tan et al. proposed a scaling method named compound coefficient that uniformly scales all dimensions of the model which are depth, width, and resolution [16]. Here, depth is the feature complexity, width is network width and resolution is the resolution of input images. Increasing each of the dimensions increases the accuracy but the accuracy gain decreases for larger CNN models. Therefore, this method is used to figure out the proper balance within the dimensions to gain maximum accuracy. They first developed an EfficientNet architecture named EfficientNet-B0 similar to MansNet to test the efficiency of the method. The model also contains squeeze and excitation layers for optimization. To check the efficiency of this method it was run on other models such as MobileNets, and Resnets, and the result was compared with EfficientNet, and the results with EfficientNet were slightly better. Another comparison was done on EfficientNet-B0 by increasing the dimensions separately and using the compound method and the compound method gave 2.5% better accuracy.

A modified version of the CNN-based pre-trained AlexNet model was used by Samir et al. to detect the forged images because of its simple structure and less memory occupation [23]. It can train data faster and detect multiple types of forgeries in an image. The AlexNet is modified to resolve the drawback of the original model by introducing batch normalization instead of local response normalization and max out instead of ReLU as an activation function. The AlexNet model is used twice in the parallel GPUs to increase the processing speed and train the model faster. The architecture extracts feature from the image input patches and classify the forgery in the output image. The evaluation gives a high performance because

of the use of a wide range of datasets to train the model and the use of the cross-validation concept that indicates the model has been trained well. Moreover, its ability to capture global pixels instead of just neighbor pixels allows it to detect the manipulations based on cues like variation in contrast.

Abdalla et al. proposed a novel technique based on neural networks and deep learning to improve copy-move forgery detection, concentrating on the convolutional neural network (CNN) architectural approach [12]. To achieve good results, the suggested method uses a CNN architecture with pre-processing stages. The results reveal that given a defined iteration limit, the total validation accuracy is 90%. Furthermore, it provides a new strategy for detecting and localizing picture counterfeiting that is based on scale variant convolutional neural networks (SVCNNs). Sliding windows with a range of scales are incorporated in customized CNNs for this technique, with the goal of constructing possibility maps that show image manipulation. The major focus of this paper is on detecting and localizing copy-move forgeries using CNNs to apply pieces that have been deleted.

Junlin et al. proposed a unique recognition strategy of image copy-move manipulation using a deep learning CNN architecture [4]. In the beginning, the framework is demonstrated by exchanging an existing database model - ImageNet, consisting of more than a thousand labeled images. After that, these are balanced marginally in the training set by utilizing little copy-move structure. In the end, these test pictures are recognized in the training model. This architecture results in an exceptional detection and also it has accomplished good execution after applying it to a small number of pictures on the training model. However, this method does not show strong results on an actual screenplay because of the CNN mapping process. So there is a scope to work in the future on this topic.

Rahul et al. suggested an efficient splicing detection and copy-move forgery pipeline architecture that focuses on recognizing the traces like noise addition, blurring, JPEG compression, contrast adjustment, and so on left by various Splicing and copy-move forgery manipulations [17]. To suppress visual content and focus solely on traces of tampering activities, the image is processed using the second difference of median filter (SDMFR) as one of the residuals, along with the Laplacian filter residual (LFR). The proposed approach achieves 95.97% accuracy on the CoMoFoD dataset, and 94.26% on the BOSSBase dataset.

Image alteration techniques such as recoloring and copy-move or splicing are recognized through neural network models by Jijina et al. [19]. This proposed strategy uses SSIM to detect the structural similarity to compare the alternation of images. The copy-move fraud discovery is based on the similitude within the pictures and detecting the fabricated portion. The Recoloring image forgery is detected by using the CNN model which includes 3 layers to show the recoloring possibilities. Initially, this procedure collected the actual image and two other images inferred from the first picture. These pictures are formed from two contemplations. The first one is - Brightening consistency and another one is the connection between the two channels. CNN model has 3 layers that extract the features of input images. Then to

merge the properties of the manipulated images a concatenation order is additionally utilized in the preceding layers to find out the features. Thus the forged images are determined.

A generative model is proposed by Yanyang et al. that can tell the difference between recolored and real photos [11]. The likelihood that the image is recolored is calculated using the real image and two extracted inputs based on illumination consistency and inter-channel correlation of the original input. The CNN-based architecture used contains three feature extraction blocks and a feature fusion module to determine forgery-relevant characteristics. Discovering from the observations that images may not preserve inter-channel correlation or illuminant consistency after the recoloring process, the difference images (DIs) and the illuminant map (IM) are generated as two sets of data of image recolored detection and sent as input along with the original image. Finally, it produces a two-dimensional vector that indicates whether the input has been recolored or not. The proposed model, RecDeNet, performs well. However, it is not very effective on datasets produced using edit propagation and palette-based re-coloring methods.

A technique for detecting differential facial retouching is developed using the FERET and FRGCv2 face databases, and an automatic database of retouched face images and unconstrained probing images by Rathgeb et al. [22]. The model extracts three types of features including texture descriptors, facial landmarks, and deep face representations, and estimates the difference vectors. Machine learning-based classifiers like SVM were used to estimate changes in feature vectors produced from texture descriptors, facial landmarks, and deep face representations, and the results detected are then merged to discriminate between retouched and unaffected facial images. In difficult cross-database evaluations, it showed a good detection performance.

Hongrui et al. used a unique 3D technique to provide the first automated method for identifying face reconstruction and automatic landmarks [30]. To do this, a dataset is created with different types of 2D caricature styles and their matching 3D forms. Then a parametric model is designed for 3D caricature faces based on vertex-based deformation space. Caricatures created by artists as well as caricatures created by machines are included in the dataset. A caricature dataset was created by finding and choosing roughly 6K distinct caricatures from various artists which was found on the internet, with each caricature having 68 identified locations. The landmark coordinates are manually revised after being established using the Dlib library. A data augmentation strategy based on CariGANs was employed to further expand the variety of this dataset. CariGANs use two generative adversarial networks (GAN), CariGeoGAN and CariStyGAN, to convert normal facial photos into caricatures. A neural network based technique was proposed where the 3D facial form was regressed and it was oriented from the entered 2D images from the built dataset and the nonlinear parametric model. The PyTorch framework was used to train the suggested model. A color caricature picture with a size of 224x224x3 is sent into CNN. All of the tests, including the technique and comparative methods, were run on a desktop PC with an Intel hexa-core i73.40 GHz processor, 16GB of

RAM, and an NVIDIA TITAN Xp GPU. This approach takes roughly 10 milliseconds to produce both a 3D model and 682D landmarks for each caricature. This rebuilt mesh has a total of 6144 vertices. The suggested algorithm design’s efficacy is demonstrated through ablation studies and comparisons to state-of-the-art approaches. Extensive test results show that the procedure works effectively for a variety of caricatures.

Rao et al. proposed a novel picture splicing detection and localization method based on a local feature descriptor acquired by a deep convolutional neural network (CNN) [21]. Using a two-branch CNN and an expressive local descriptor, hierarchical representations are automatically trained from images of different color formats. To begin, the first layer of the proposed CNN model is used to suppress picture content effects and extract the different and expressive residual characteristics, which is especially suitable for image splicing detection applications. The kernels of the first convolutional layer are implemented using an optimized combination of the 30 linear high-pass filters used in the spatial rich model (SRM) to calculate residual maps and fine-tuned using a limited learning technique in order to maintain the learned kernels’ high-pass filtering capabilities. Second, combining the contrastive and cross-entropy losses increases the proposed CNN model’s generalization capabilities. Furthermore, the block-wise dense features produced by the pre-trained CNN-based local descriptor for a test picture are combined using an effective feature fusion approach known as block pooling to deliver the final discriminative features for image splicing detection using SVM. A pre-trained CNN model is employed to build an image splicing localization technique using the fully connected conditional random field (CRF). Extensive testing on a variety of public datasets shows that the proposed CNN-based technique outperforms numerous state-of-the-art algorithms in terms of photo splicing detection and localization, as well as JPEG compression robustness.

Jaiswal et al. approached a deep learning convolutional neural network (CNN) model to anticipate faked images[13]. Pre-trained image recognition algorithms are given a huge amount of input images to train the residual neural network (RESNET-50) model and predict other images using a classifier. Three different classifiers namely Naïve Bayes, K-nearest neighbor, and Multi-Class Model using SVM Learner were used to train and test the set of original and forged images. CASIO 2.0 dataset was divided into two groups, one is original and the other tampered with. Classification algorithms use a confusion matrix for evaluation, in which the classifier divides the dataset into class labels. The experiment is run on a server with a Xeon processor and 16 GB of RAM, running Ubuntu Linux server and MATLAB R2017b tool is used.

A new method for detecting splicing in photographs that combines the great representation capability of Illuminant Maps and Convolutional Neural Networks as a way of learning the most essential signals of a counterfeit straight from accessible training data is provided by Pomari et al. [7]. This paper proposes an approach that bypasses the time-consuming feature engineering process, allows for the detection of counterfeit regions, and achieves a classification accuracy greater than 96%, beating the best methods across a variety of datasets. Analyzing several unusual everyday

pictures that went viral further highlights the suggested method’s potential applications. Furthermore, Their approach is based on a hypothesis that picture splicing introduces discrepancies that may be emphasized in illuminant maps. However, they apply deep and transfer learning approaches to perform learning and extracting appropriate features for manipulation detection, termed DSF, which are then used to train a forgery detection classifier. Finally, they present a new approach for locating the manipulated region if a picture is categorized as a composite/splicing.

A coarse-to-refined CNN (C2RNet) and diluted adaptable cluster model are suggested by Xiao et al. [27] as two aspects of a splicing forgery detection system. The stated C2RNet combines a coarse convolutional neural network (C-CNN) with an R-CNN to identify image characteristics variance between manipulated and unmanipulated areas from image patches of various sizes. Additionally, CNN operating on images is used to substitute CNN operating with patches in C2RNet to reduce computational complexity. The suggested detection approach learns the distinctions between multiple picture attributes to provide a steady observation, also the image level CNN reduces the computation errors significantly. Following the suggested C2RNet has found the suspected forging regions, the determined forged areas are constructed using the suggested adaptive clustering technique. Even under various attack conditions, the experiment findings show that the suggested detection approach delivers comparably better results when compared to the best splicing forgery detection methods.

A single shot multibox detector was used for real-time object detection in images as it provides better detection precision for real-time speed. The model proposed by Kumar et al., extracted feature information from the image using convolutional neural networks, then performed feature mapping to organize the label of classes [20]. The change in aspect ratio was handled by employing distinct filters with various default boxes, as well as multi-scale feature maps for object detection.

However, a very basic YOLO model was used by Redmon et al. and they also showed a comparative analysis with already existing object detection algorithms namely, RCNN, MultiGrasp, SSD, etc [1]. Despite its certain limitations, it performs better than the mentioned algorithms and since this was a basic YOLO model, it has its upgraded versions which offer further better performance.

Pointing to the effectiveness and precision of small-scale object classification in the current activity stream, Gongguo et al. approached a better YOLO-V3 algorithm that compares with the previous version [33]. These improved classification methods performed remarkably on mini objects. For the purpose of improvement, this method picked a one-step object classification algorithm along with the quicker detection method to increase its effectiveness. At first, this technique optimizes the YOLO-V3 structure through a fresh small object of fourfold down sampling remaining in the middle of the 2nd and the 3rd stack layers of Darknet-53. Then to progress the precision for little objects, it executes twofold up trials on the eightfold downsampling to match the result to the initial target. And then it adds the twofold up results with the 3rd residual layer’s result. After that, with all the feature combinations, this results in four times downsampling. At last, this revised algorithm

and the previous version of the algorithm are compared for the differentiation. And the output clearly shows a notable precision and recall rate of the small-scale object detection.

Wang et al. recommended the YOLO-V4 model as a better object detection method for ship missiles [25]. This YOLO-V4 method has a more accurate and rapid classification ability. This system can pull out all the attributes and detailed information and also the regression analysis in just a CNN model. The YOLO-V4 algorithm can collect images to take input and then split up the images into separated regions-v4 worked on Mosaic data upgrade strategy while taking as inputs to upgrade the information and then used the images information through scaling, editing, and also ordering the 4 pictures. It significantly enhances the image recognition information set and expanded the number of little targets and made strides in the vigor of the arrangement. Within the primary organize portion, YOLO-V4 constructs the CSP-Darknet53 process depending on the CSP-Net. The CSP-Darknet53 method improves the CNN arrange recognition capacity, but moreover guarantees the precision rate whereas lessening the sum of calculations and lessening the recall rate.

Long et al. talked about improving the performance and effectiveness of YOLO-V3 by introducing paddle-paddle (PP-YOLO) [35]. They had made a few tweaks to YOLO V3 such as using ResNet50 instead of Darknet53 and replacing a few of the convolutional layers with DCN(Deformable Convolutional layers) to increase the infer speed. They also used larger batch sizes and calculated EMA (exponential moving average) to increase the stability of the model. They ran this model on a few datasets and the results showed that this method gives slightly more accurate results than YOLO V4 and V3. However, due to a lack of accurate detection datasets, their model was not giving the result they had expected. This model gave an accuracy of 45.2% on all the datasets they had used.

Faster-YOLO derives all of YOLO's end-to-end features and directly predicts the bounding box and object class. Yin et al. stated that the four elements of Faster-YOLO are the input picture, feature extraction network, bounding box prediction, and final detection result [28]. The size of the input picture is 416×416 pixels. An $S \times S$ grid has been used to split the image. The DRKCELM and DLELMAE combined network was employed as a feature extractor for classification and detection. In each grid cell, bounding boxes and confidence ratings for those boxes are predicted. Each bounding box is responsible for forecasting four values: tx, ty, tw, th, and confidence. For the recognized item, each grid cell also forecasts C conditional class probabilities (t_{ci} , $I = 1, 2, \dots, C$). Finally, a predicted tensor is calculated and used in the regression. Finally, the non-maximum suppression (NMS) approach is used to achieve the final item detection findings. In comparison to YOLO V2, faster YOLO has substantially improved the detecting impact.

One of the components of Faster-RCNN is a region of interest polling or Rol poolings. The goal of Rol pooling is to extract fixed-size feature maps using maximum pooling on the full picture. The other component is ROI. The areas where there is a chance of finding an object are limited by a region which is referred to as the

region of interest (ROI). The extremely fast objective of ROI is to find the places in the provided input picture where there is a likelihood of object localization. It is possible to pinpoint the location of an object in a picture. The next step is to assign corresponding classes to the regions of interest identified in the previous phases. Convolution Neural Networks is the technology used by Abbas et al. [5]. It yields a Precision of 0.6 on a small Vehicle dataset. The detector was tested on a single picture of vehicle datasets in the paper. The network differentiated the region of interest objects in the picture as intended.

Single-Shot Multibox detector (SSD), Faster RCNN, and different versions of YOLO, all three of these object detection models take an image as input in the initial state. In YOLO-v4 the image is split up into separated regions to anticipate the complete picture specifically so that it can get the whole details of the target and ignore the errors. Then the information is used by scaling, editing, and ordering the images. This version of YOLO constructs the CSP-Darknet53 process depending on the CSP-Net to improve the CNN arrange recognition capacity. YOLO -V3, on the other hand, optimizes the structure through a fresh small object of fourfold down sampling remaining in the middle of the 2nd and the 3rd stack layers of Darknet-53. It executes twofold up trials on the eightfold downsampling to match the result to the initial target, then adds the results with the 3rd residual layer's result. Finally, it compares with the previous version to point out the precision of small-scale object classification. Faster YOLO takes an input image of 416 by 416 pixels in size which is then divided into an $S \times S$ grid. If the item's center exists within a grid cell, that grid cell is responsible for detecting the object. The DRKCELM and DLELMAE combined network is used as a feature extractor and a non-maximum suppression (NMS) technique is used to finally detect the object. Faster-RCNN uses a region of interest polling or ROI pooling to extract fixed-size feature maps using maximum pooling on the full picture. The region of interest (ROI) marks the region in the provided input picture where there is a likelihood of object localization. The object location is pinpointed on the picture and the corresponding classes are assigned to the regions of interest found in the previous step. SSD, however, utilizes a CNN model for feature extraction maps at various places. In every feature map, a 4×4 filter is used to determine a small low default box at each place. Later, an estimation is made on the bounding box offset for each box and the odds of each box's class. The truth boxes are matched with the expected boxes using IOU. Multi Boxes or filters of different sizes and different aspect ratios are used for increasing the object detection accuracy. It has additional convolutional layers that have multiple features with various scales and is, therefore, able to detect objects at multiple scales better.

Chapter 3

Methodology

3.1 Workplan

The proposed framework is designed to follow a layered hierarchical approach where at each level certain tasks are performed to finally get the desired result. The work plan is divided into two parts where the first part is about how the dataset is going to be prepared. The second one describes the proposed framework.

Part-I: Preparation of the dataset:

- **Image Scraping:** Collect digital artworks from various social media sites and NFT sites to prepare the dataset.
- **Creating Fake Images:** Perform various types of image forgery using Adobe Photoshop and StyleGAN to create fake images of the collected real images.
- **Compiling the dataset:** The REAL and FAKE images together will make up the dataset which will be contributed globally as the first-ever digital art dataset.

Part-II: Implementing the framework:

- **Classification of Real and Fake Images:** Train a CNN model with the REAL images and then run predictions on the FAKE images to detect styles, extract features and conclude on the type of image forgery done on the REAL image.
- **Comparing the number of objects of the Real and Fake Images:** Train an object detection model with the REAL images along with its annotations and then run predictions on the FAKE images by comparing the total number of objects in each pair and progressing with further analysis.
- **Maintaining benign information:** Train an SVM Classifier with the meta-data collected while image scraping. This phase of the framework learns about only benign information it is provided with so that the truth sets of information can be compared with the malicious ones.
- **Displaying ground truth:** Combine the results of the CNN model and the object detection model to check if any forgery is found. If yes, then SVM displays the real artist's information as evidence.

- **Granting Permission:** If any forgery is detected and contains a corresponding ground truth, the NFT post is not allowed, else, it is allowed.

3.2 Preparation of the Dataset

In light of the problem statement, the parameters required to create this very dataset are images but to add more emphasis to the dilemma, some other parameters are also required to run down a valid conclusion. It is crucial to know about the whereabouts of a certain artwork posted online starting from the real artist’s identification to time and place constraints as to when and where that certain post was made. Performing image forgery is a crucial part of this work plan as it is the strongest requirement for this dataset as per the dilemma. It is the quality of forgery work done that makes it hard to distinguish a fabricated image from its real one. Thus, the work of Photoshop would be performed by experienced personnel who have ideas about these techniques which would be an empowering asset for this dataset. This dataset will help in analyzing and solving the problems of scammers stealing artworks and then fabricating them to trick the human eye. For this reason, the dataset is being named **ARTIFICE2021V1** as the word Artifice means trickery. An overview of the plan is shown in the flowchart in Figure 3.1 followed by the detailed proceedings of how the dataset was planned to be prepared.

3.2.1 Image Scraping

To perform Image Crawling, some NFT sites and social media sites were chosen to collect digital artworks for the dataset. Meanwhile, it is also taken into consideration that the collected images had their copyrights valid throughout this entire work. A properly coded image scraper/crawler couldn’t be used as every target site had a distinct web structure for which an accurate source could not be found. The very basic ones available had the problem of compressing the resultant image into a poor resolution which could be a problem for poor classification results. Thus, 2000 images were manually downloaded from various chosen sites that are not only popular and running in recent times but also have records of NFT thefts. A balance was maintained between the number of images to be downloaded from one site. A detailed breakdown of the division is shown in Table 3.1.

Target Category	Website Name	Number of Images
NFT sites	Known Origin	180
	Bakery Swap	234
	Nifty Gateway	235
	OpenSea	215
	Rarible	233
Social Media sites	Artstation	232
	DeviantArt	234
	Instagram	233
	Behance	204
Total Number of Images		2000

Table 3.1: Number of images from each target site.

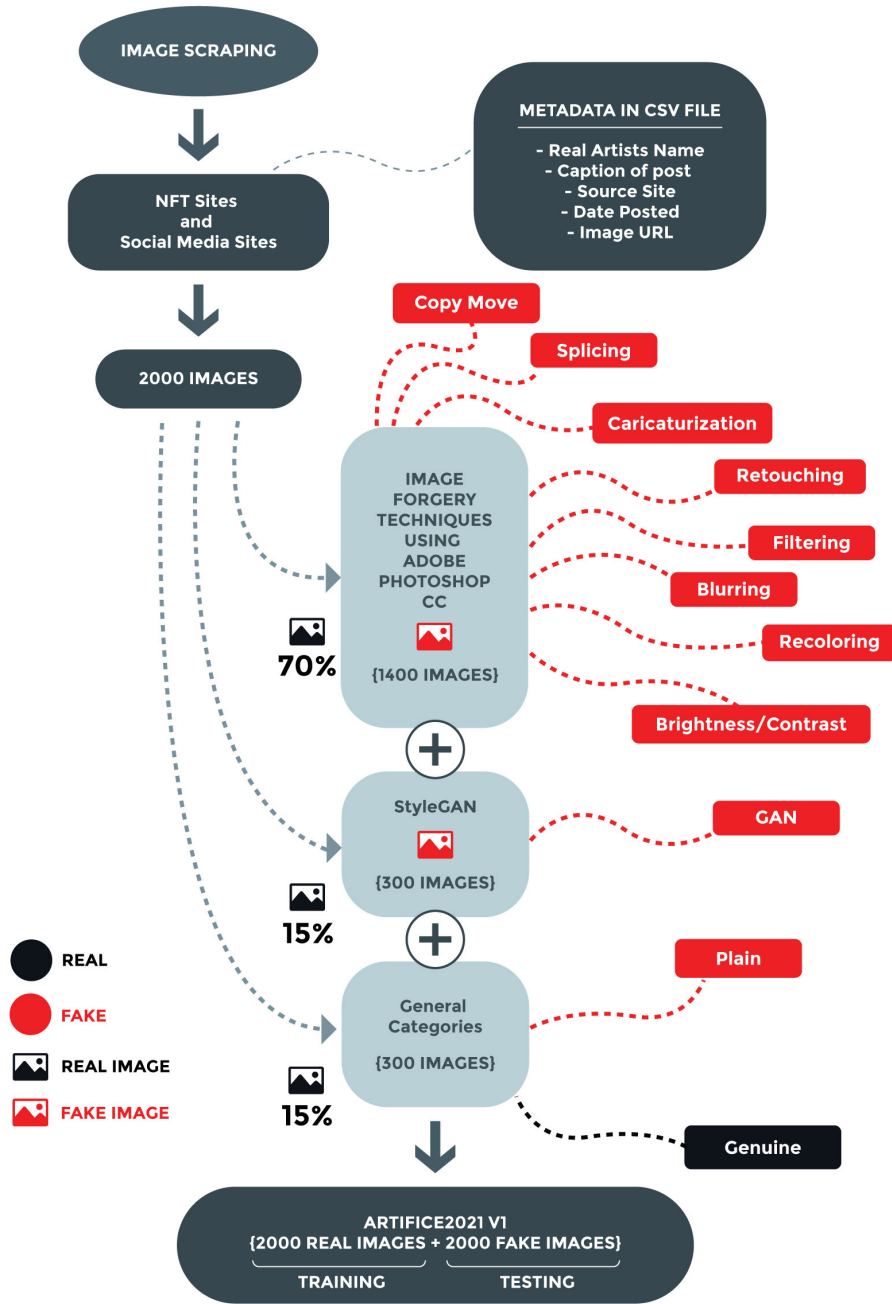


Figure 3.1: Work Plan Part-I: Preparation of the dataset.

3.2.2 Creating Fake Images

The downloaded images were manipulated with certain image forgery techniques using Adobe Photoshop and StyleGAN. After performing forgery on the 2000 downloaded images, 2000 more fake images are created. This pair of 2000 images (a total of 4000 images) make up the entire dataset. There are a total of 11 categories (classes) of image forgery in this dataset which have been maintained to reflect the diversity of approaches a thief may attempt. A detailed overview of how the images were divided among each category is shown in Table 3.2.

Category	Number of Images	Image Percentage
Photoshop	1400	70%
StyleGAN	300	15%
Genuine	200	15%
Plain	100	15%
Total	2000	100%

Table 3.2: Percentage breakdown of the number of images each forgery category holds.

A detailed breakdown of how many images reside in each Photoshop category can be found in Table 3.3.

Image Forgery Level	Photoshop Category	Number of Images
Low-level	Brightness and Contrast	140
	Sharpening and Blurring	140
	Filters	140
	Hue and Saturation	140
High-level	Copy-Move	210
	Splicing	210
	Caricaturization	210
	Retouching	210
Total	8 categories	1400

Table 3.3: Number of images belonging to each Photoshop category.

For the purpose of this work, Image Forgery for classes 2-9 were performed in Adobe Photoshop CC 2020. Images belonging to classes 0 and 10, were copied to complete their respective REAL-FAKE pairs. Fake images of Class 1 were generated using StyleGAN. The categories below are described from the thief’s point of view. A manifestation of how a particular image looks after performing the image forgeries of the 11 classes is demonstrated in Figures 3.2 to 3.11.

- **Plain (Class-0):** When the thief will not bring any changes to the real image they stole and will directly post it on an NFT site. **Figure-5** shows a demonstration of what this should look like.

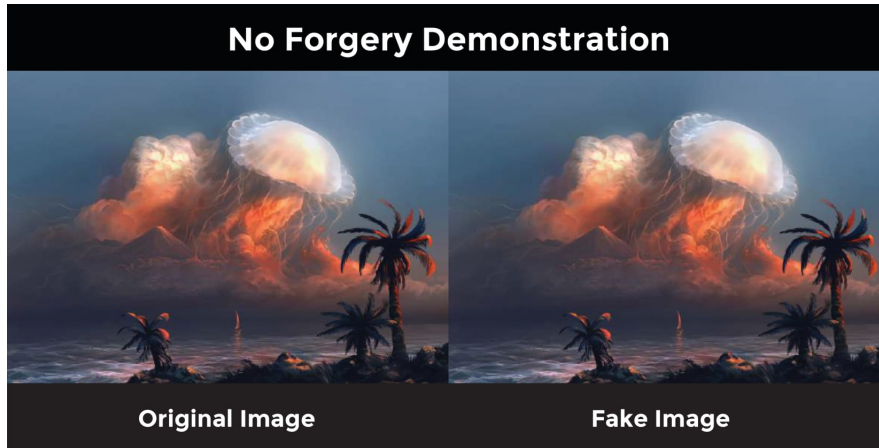


Figure 3.2: Manifestation of plain forgery.

- **StyleGAN (Class-1):** When the thief will want to transfer the style of a certain external image into the original image to create a fake image. **Figure-6** illustrates how a different style has completely changed how the original image looks.

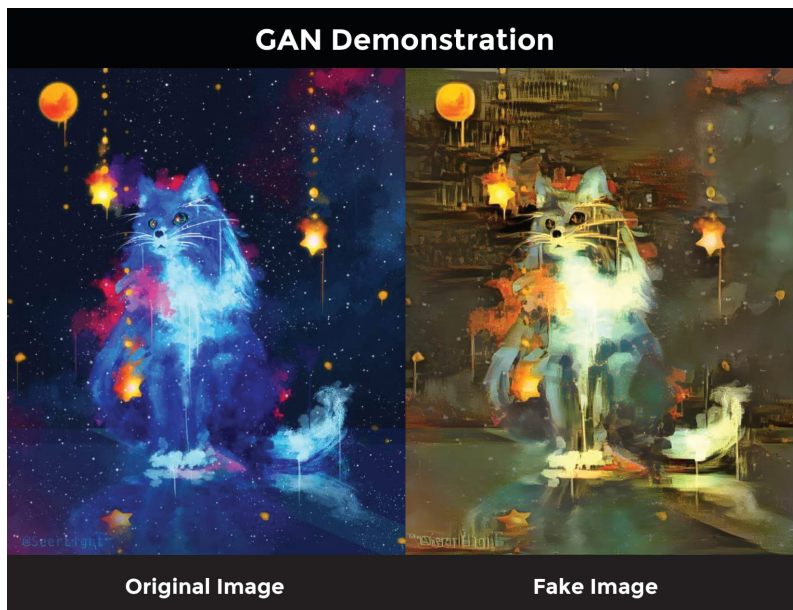


Figure 3.3: Manifestation of StyleGAN.

- **Copy-Move (Class-2):** When the thief attempts to copy a part of the real image and paste it in a different location on the same image to hide objects or generate extra meaning which is originally not present in the image in order to deceive normal users. The following image in Figure-7 is a digital artwork. To perform copy-move on this image, the object in focus is selected to be copied and pasted in a distant location as per the one-point perspective of this picture. The object which happens to be the person on the skateboard was carefully removed from the whole picture, i.e., the background was removed. As can be seen in Figure-7 that the **copied object** has a clear background. This object

is then placed in another location. This changed the outlook of the whole image as if it was real along.

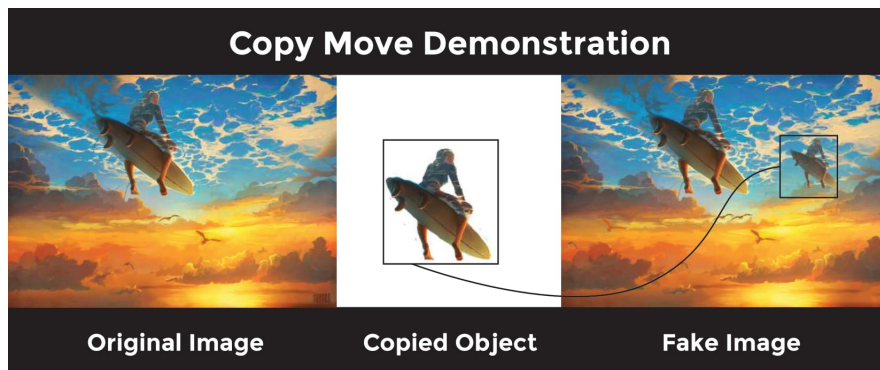


Figure 3.4: Manifestation of Copy-Move forgery.

- **Splicing (Class-3):** When the thief attempts to copy a part of some other image and paste it into the real image to hide objects of the real image or generate extra meaning which is originally not present in the image in order to deceive normal users. To perform splicing on the digital artwork in Figure-8, an external image was considered. To match the context of the original image, i.e. the rail tracks, a train is bought from another picture and pasted on the original image to make it look real. It can be seen in Figure-8 that the **External Object** has a clear background which is then pasted on the original image over the rail tracks to create the fake image. This category makes it even riskier as two artworks are to be considered here as elements of two images can be interchanged, to create two fake images.

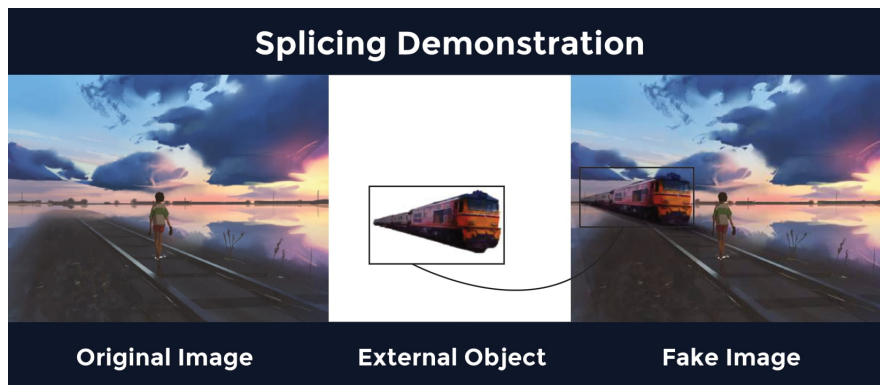


Figure 3.5: Manifestation of Splicing.

- **Retouching (Class-4):** When the thief attempts to fix the disfigurements of an image or try blending in similar environments in order to make the real image look different. In Figure-9, the texts on the signboards that are highlighted with the white boxes are removed. Here the signboards are made empty by blending the background color with the texts on them.

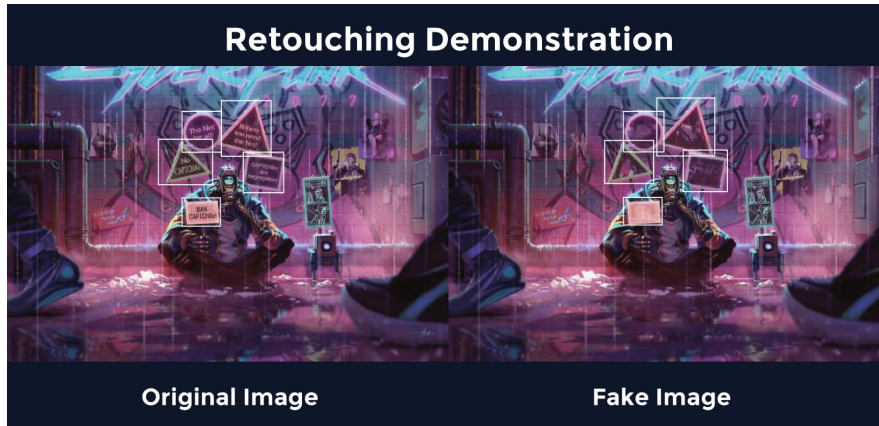


Figure 3.6: Manifestation of Retouching.

- **Caricaturization (Class-5):** When the thief intends to bloat or pucker certain parts of the human body, especially the face in human anatomy artworks. To perform caricaturization on Figure10's original image, the bloat and pucker features were used on the eyes, nose, and lips of the girl's face to distort their appearances. Lastly, the features were applied overall on the face to enlarge its shape which can be seen in Figure-10. The problem with this forgery technique is that, if someone draws a caricature of a face from scratch, then it's legal but if someone uses shortcut tools like this to caricature a portrait that is done by somebody else, then that is illegal.

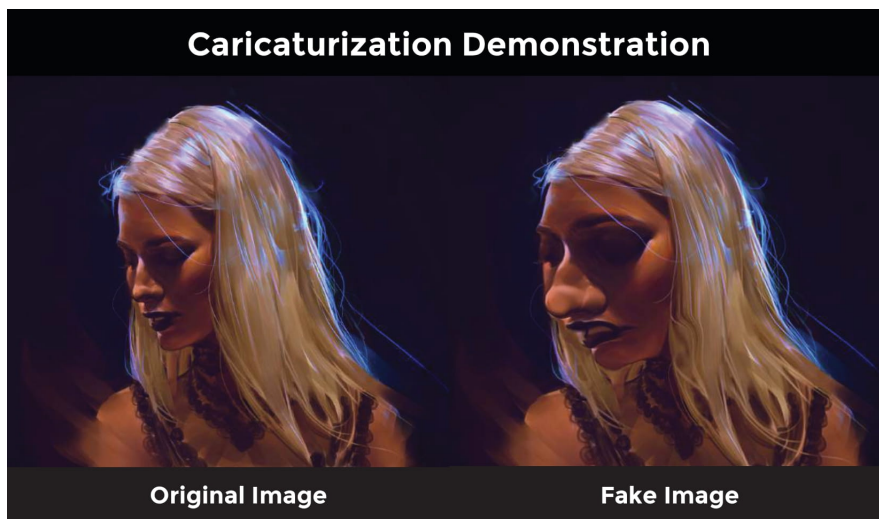


Figure 3.7: Manifestation of Caricaturization.

- **Blurring (Class-6):** When the thief blurs certain parts of the real image in order to give a deceiving impression to normal users. Figure-11 shows that the background behind the character is blurred.

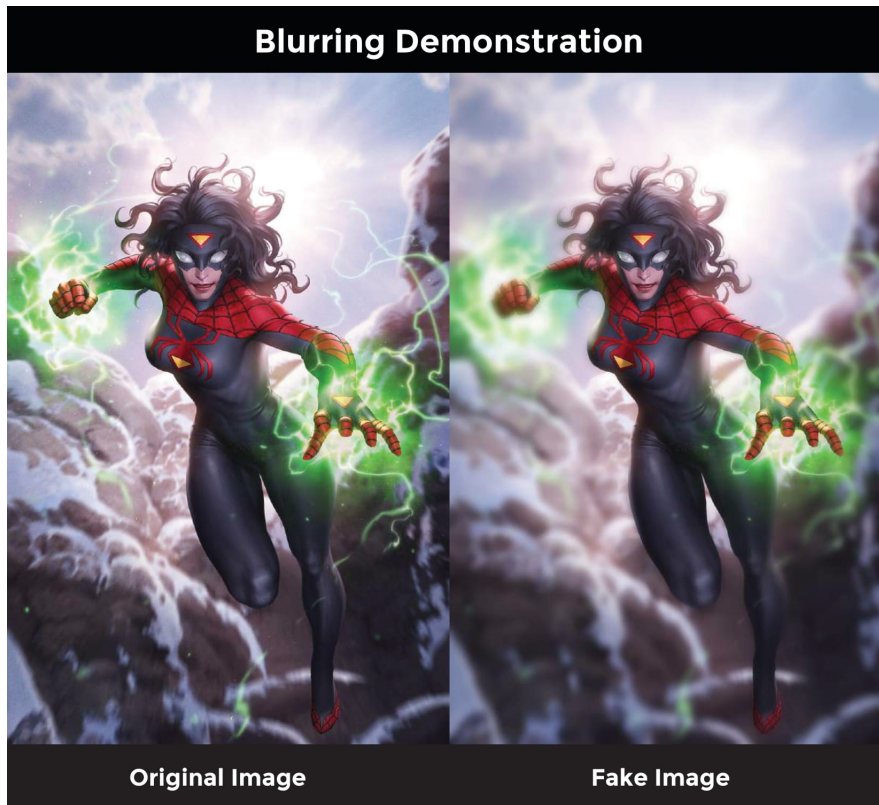


Figure 3.8: Manifestation of Blurring.

- **Filters (Class-7):** When the thief changes the outlook of the original image with the alteration of color pixels by applying filters. The original image in Figure-12 has had a filter applied to it and the results can be seen in the fake image.

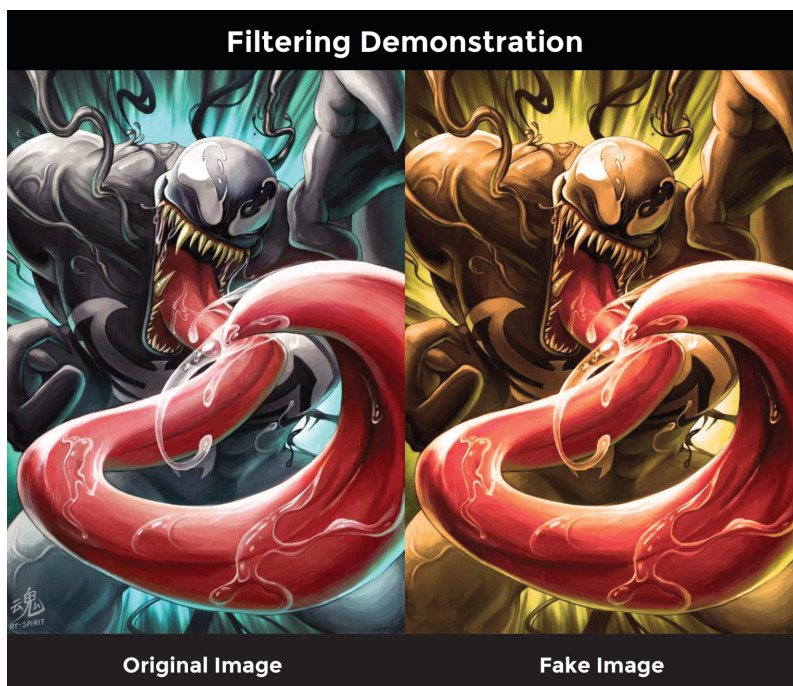


Figure 3.9: Manifestation of Filtering.

- **Brightness and Contrast (Class-8):** When the thief plays with the lightness and darkness of the real image, hence changing the brightness property followed by changing the difference in brightness between certain regions or objects of the image, the contrast property. In Figure-13, the brightness properties were increased and the contrast properties were decreased for which the final result can be presented as a fake image.



Figure 3.10: Manifestation of Brightness/Contrast.

- **Recoloring (Class-9):** When the thief brings changes to the color properties of the real image to make it look different. The original image in Figure-14 had properties like hue, saturation, tint, temperature, color balance, etc altered to get what is shown in the fake image.



Figure 3.11: Manifestation of Recoloring.

- **Genuine (Class-10):** This category is not for a fake participant. It symbolizes the authenticity of a completely new artwork that has been done for the first time by a genuine artist. So basically, all the artworks that were scraped from the target websites hold the entity of authenticity, i.e. belonging to the "Genuine" class initially. This class will help verify those artworks that were never stolen or further manipulated.

3.2.3 Compiling the dataset

At this phase, a total of 4000 images including 2000 real downloaded images and 2000 fake forged images are in this dataset and the CSV file in Figure 3.13 holds certain parameters for the downloaded real images which happen to be the benign information of the real artists. These pieces of information can be provided as evidence in order to accuse the thief. The parameters have been discussed in the following.

1. **Real and Fake Image Labels:** Two folders named, "**TRAIN**" and "**TEST**" are maintained where the real and fake images are kept respectively. Inside these folders, there exist 11 more folders representing each category of image forgery. The folder names start from 0 all the way up to 10 to represent the 11 forgery classes. This action here helps perform clustering of data which reduces the time for not having to do it by code during the data pre-processing steps. If the *n*th real image is to be labeled, "**R**" is added to the end of "**n**" and if the *n*th fake image is to be labeled, "**F**" is added to the end of "**n**". This is to distinguish between real and fake images. An example is given: **1R.jpg**, **1F.jpg**. Figure 3.12 shows how the real and fake images are labeled.
2. **Caption:** This section is to identify if the real artist posted a caption along with the artwork.
3. **Source Site:** This section identifies the site where the real artist posted the artwork.
4. **Date Posted:** This section is to record the date on which an artwork was posted.
5. **Image URL:** This section is needed so that the image URL can be shown to the fake participant in order to warn them after the framework successfully detects the theft.
6. **Class:** This is to assign each image a class number from 0 to 10 as per the 11 image forgery categories to provide them a unique identifier.

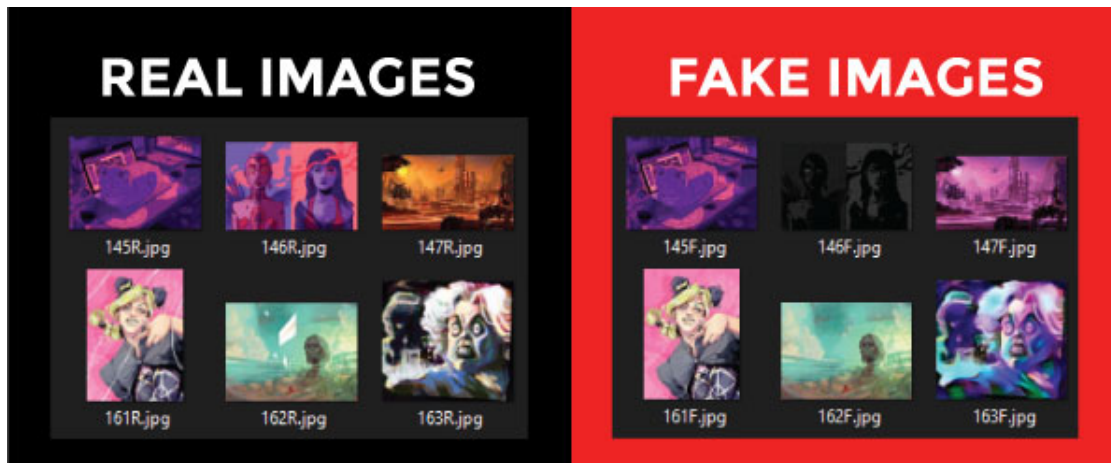


Figure 3.12: Labeling of REAL and FAKE images.

ARTIFICE2021-V1 ☆ 📄 🌐

File Edit View Insert Format Data Tools Extensions Help Last edit was seconds ago

100% | \$ % .0 .00 123 | Default (Ari... | 10 | B I U A | 🗑️ 📄 📄 | ☰ ☰ ☰ | ⌂ ⏪ ⏩ | ↻

V1865	A	B	C	D	E	F	G	S
1	Real Image Label	Real Artist	Caption	Source Site	Date Posted	Image URL	Class	Fake Image Label
2	1R.jpg	nanchaort	Ballerina	Deviant Art	September 24, 2002	https://www.deviantart.com/nanchaort/art/Ballerina-100000000	2	1F.jpg
3	2R.jpg	HJP28	A Time to Wonder	Deviant Art	August 17, 2020	https://www.deviantart.com/HJP28/art/A-Time-to-Wonder-100000000	2	2F.jpg
4	3R.jpg	MuerteMil	Wallpaper - Ori ar	Deviant Art	July 12, 2020	https://www.deviantart.com/MuerteMil/art/Wallpaper-Ori-ar-100000000	3	3F.jpg
5	4R.jpg	Sledjee	Carnage	Deviant Art	August 6, 2021	https://www.deviantart.com/Sledjee/art/Carnage-100000000	3	4F.jpg
6	5R.png	TurnipBerry	Aj x Fluttershy x I	Deviant Art	September 28, 2021	https://www.deviantart.com/TurnipBerry/art/Aj-x-Fluttershy-x-I-100000000	2	5F.jpg
7	6R.jpg	jostanfor	The-woods-beyon	Deviant Art	October 4, 2021	https://www.deviantart.com/jostanfor/art/The-woods-beyon-100000000	3	6F.jpg
8	7R.jpg	Stepyra	Darnath Lysander	Deviant Art	April 11, 2018	https://www.deviantart.com/Stepyra/art/Darnath-Lysander-100000000	3	7F.jpg
9	8R.jpg	ERA7	Waiting for you	Deviant Art	August 7, 2021	https://www.deviantart.com/ERA7/art/Waiting-for-you-100000000	3	8F.jpg
10	9R.jpg	Samuel-One	Aesthetics	Deviant Art	August 2, 2021	https://www.deviantart.com/Samuel-One/art/Aesthetics-100000000	3	9F.jpg
11	10R.jpg	E-Mann	Spidy suit	Deviant Art	January 2, 2011	https://www.deviantart.com/E-Mann/art/Spidy-suit-100000000	3	10F.jpg
12	11R.jpg	E-Mann	Cover Ben Reilly:	Deviant Art	November 10, 2011	https://www.deviantart.com/E-Mann/art/Cover-Ben-Reilly-100000000	9	11F.jpg
13	12R.jpg	E-Mann	Spider-Man Desig	Deviant Art	November 30, 2011	https://www.deviantart.com/E-Mann/art/Spider-Man-Desig-100000000	2	12F.jpg
14	13R.png	OrangeSavannah	The Emyrean Su	Deviant Art	September 13, 2021	https://www.deviantart.com/OrangeSavannah/art/The-Emyrean-Su-100000000	3	13F.jpg
15	14R.png	fox-hunt	Origin	Deviant Art	September 9, 2021	https://www.deviantart.com/fox-hunt/art/Origin-100000000	2	14F.jpg
16	15R.jpg	ChunLo	Marvel's Spider-N	Deviant Art	February 7, 2020	https://www.deviantart.com/ChunLo/art/Marvel-s-Spider-N-100000000	2	15F.jpg
17	16R.png	Spyder4lyfe	Beware The Batm	Deviant Art	August 19, 2021	https://www.deviantart.com/Spyder4lyfe/art/Beware-The-Batm-100000000	2	16F.jpg
18	17R.jpg	spidey0318	Wonder Woman -	Deviant Art	March 12, 2016	https://www.deviantart.com/spidey0318/art/Wonder-Woman-100000000	3	17F.jpg
19	18R.jpg	HalcyonMoufette	Print Pedestal	Deviant Art	August 1, 2021	https://www.deviantart.com/HalcyonMoufette/art/Print-Pedestal-100000000	2	18F.jpg
20	19R.jpg	RichardLayArt	BIG Tree	Deviant Art	August 31, 2021	https://www.deviantart.com/RichardLayArt/art/BIG-Tree-100000000	3	19F.jpg
21	20R.jpg	theCHAMBA	I Am . .	Deviant Art	November 3, 2007	https://www.deviantart.com/theCHAMBA/art/I-Am-.-.100000000	3	20F.jpg
22	21R.jpg	TamberElla	Swinging Througl	Deviant Art	January 31, 2011	https://www.deviantart.com/TamberElla/art/Swinging-Througl-100000000	2	21F.jpg
23	22R.jpg	Ry-Spirit	Venom	Deviant Art	October 5, 2018	https://www.deviantart.com/Ry-Spirit/art/Venom-100000000	7	22F.jpg
24	23R.jpg	theCHAMBA	My Spidey Sense	Deviant Art	May 18, 2010	https://www.deviantart.com/theCHAMBA/art/My-Spidey-Sense-100000000	7	23F.jpg
25	24R.jpg	theCHAMBA	Thunderbolt Hulk	Deviant Art	December 13, 2011	https://www.deviantart.com/theCHAMBA/art/Thunderbolt-Hulk-100000000	6	24F.jpg
26	25R.jpg	theCHAMBA	Digital Spider	Deviant Art	June 19, 2005	https://www.deviantart.com/theCHAMBA/art/Digital-Spider-100000000	3	25F.jpg
27	26R.jpg	Endemilk	Mesmerizing circl	Deviant Art	September 4, 2021	https://www.deviantart.com/Endemilk/art/Mesmerizing-circl-100000000	2	26F.jpg
28	27R.jpg	FacundoDiaz	Dragonsight	Deviant Art	August 14, 2021	https://www.deviantart.com/FacundoDiaz/art/Dragonsight-100000000	3	27F.jpg
29	28R.jpg	Huukomori	Catcher in the Lig	Deviant Art	August 7, 2021	https://www.deviantart.com/Huukomori/art/Catcher-in-the-Lig-100000000	2	28F.jpg
30	29R.jpg	robysdesignsofficial	Lost Sky Tree	Deviant Art	August 13, 2021	https://www.deviantart.com/robysdesignsofficial/art/Lost-Sky-Tree-100000000	3	29F.jpg
31	30R.png	HazPainting	Expedition 01	Deviant Art	September 1, 2021	https://www.deviantart.com/HazPainting/art/Expedition-01-100000000	2	30F.jpg
32	31R.jpg	FleetingEmber	The Sound of the	Deviant Art	August 18, 2021	https://www.deviantart.com/FleetingEmber/art/The-Sound-of-the-100000000	2	31F.jpg
33	32R.jpg	Drawsouls	Sould of Cunder	Deviant Art	August 17, 2021	https://www.deviantart.com/Drawsouls/art/Sould-of-Cunder-100000000	3	32F.jpg
34	33R.jpg	SetEshh	City	Deviant Art	August 8, 2021	https://www.deviantart.com/SetEshh/art/City-100000000	2	33F.jpg
35	34R.png	ExCharnv	I am the bezinimr	Deviant Art	June 13, 2020	https://www.deviantart.com/ExCharnv/art/I-am-the-bezinimr-100000000	8	34F.jpg

Figure 3.13: CSV file of the dataset (Benign Information).

Chapter 4

Implementing the framework

Now, that the images are ready, the framework comes in. It has been designed with the plan of analyzing and distinguishing fake images from real ones. Figure 4.1 demonstrates the framework starting with classifying the images with a CNN model along with analyzing the positions and quantity of objects found on the test images with an object detection model. Next, a sort of database is maintained using a Machine Learning algorithm which consists of all the ground truths. This is followed by displaying what is contained as the ground truth and finally combined with the results of the CNN and object detection models to run a conclusion about the posts allowance as an NFT.

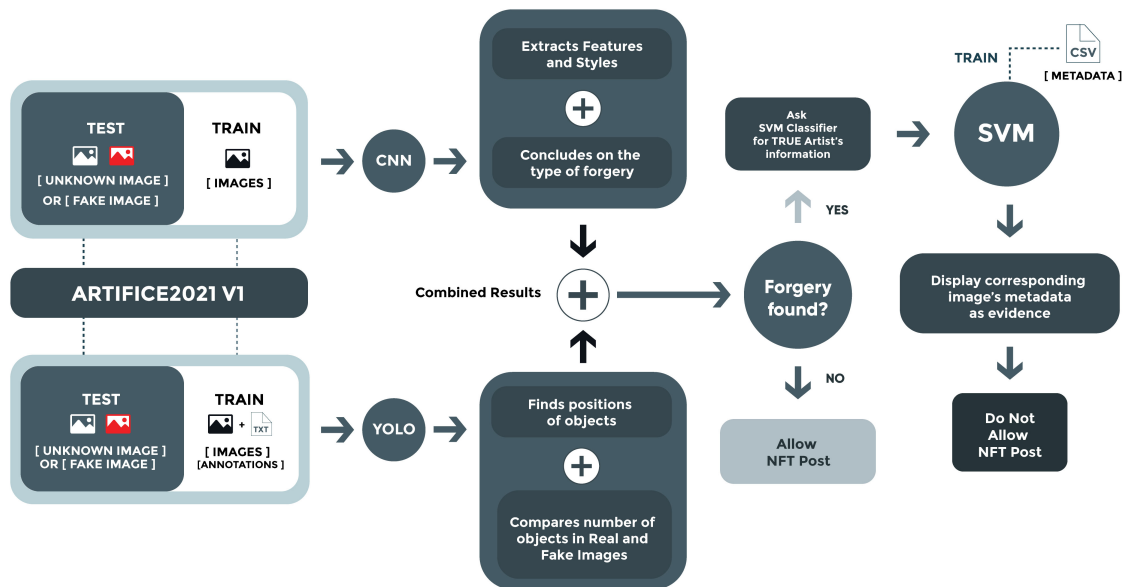


Figure 4.1: Work Plan Part-II: Structure of the framework.

4.1 Data Curation

Before starting the classification phase, necessary pre-processing steps were taken which are explained below.

1. **Input Data:** After compiling the pairs of real and fake images into the complete dataset, the input data for the intended purpose are the real images with which the CNN model is to be trained. Thus, the real images are kept in a folder named “**TRAIN**” which further contains 11 more subfolders representing the various categories of image forgery. The real images are used to train the model by keeping in mind that they are the original content created by authentic artists and the model should learn about the genuine content first in order to detect any forged version of that particular image and talking about the forged versions, the model also needs to learn about the types of forgeries that exist.
2. **Necessary Libraries:** This implementation iteration uses NumPy, OpenCV, and Matplotlib for some general tasks intended for pre-processing purposes. Furthermore, as the main framework to support Machine Learning functions, Tensorflow, PyTorch, and Keras are used from where the necessary libraries or layers required to run the chosen CNN models are imported.
3. **Data pre-processing** Since raw images were downloaded from various websites and then Photoshopped, they were in different sizes and formats. To start with pre-processing these images, firstly the images are read from the directory of the “**TRAIN**” folder and the number of categories or classes of image forgery is defined in a separate list, which here is named labels. Now, using a for loop, for each category in the list of categories certain tasks are performed. The list of categories is connected with the directory which makes the complete path to the “**TRAIN**” folder. Now each label is extracted from the index of the list of categories. Next, for each image in this connected directory, an image is chosen and then converted into arrays using OpenCV. Afterward, the image is resized into a common dimension. These are done at each iteration so that it can cover all the images of the directory. Consequently, these resized, converted image arrays and the labels are appended into a new data list. In contrast, here no validation split is required as all the real images are used to train the model. Thus, now from the list “data” the features/images (image data converted into arrays) and the labels are split into two different arrays which are then further converted into NumPy arrays. Subsequently, the NumPy array containing the features/images is scaled to bring the values between 0 and 1. The shape of this array would confirm that it has data worth 2000 images with all the images resized to the given dimension within an RGB (3) color channel.

4.2 Classifying real and fake images using existing CNN models

The main challenge for the classification process lies in the complex structure of each image the dataset contains. Since digital artworks are to be worked on, the images vary heavily in size. This is because artists have numerous options in choosing a canvas size depending on the context or theme of the artwork. For this reason, it was decided to work with already existing CNN models which have been showing

good performance recently. The models that were chosen are AlexNet, EfficientNet-B0, ResNet-50, VGG-16 and MobileNet-V1. Five models were chosen to run a comparison of their performances on the **ARTIFICE21 V1** dataset.

4.2.1 Training the models

AlexNet is a pre-trained model that has an architecture of 8 layers and is currently the leading CNN model for any classification task. On the other hand, EfficientNet-B0 has 237 layers which are specialized to work well with a lesser number of parameters. Unlike EfficientNet-B0, the other recommended model was EfficientNet-B7 consisting of 813 layers but since this is a layer heavy model, B0 was used. Furthermore, ResNet-50, a pre-trained model with 50 layers has similar contributions like the AlexNet model. It was chosen to compare the two to see which performed better. Next, VGG-16 supports up to 19 layers and it specializes especially in learning features efficiently. Lastly, MobileNet-V1, a model of 28 layers performs similar to EfficientNet with a lesser number of parameters. While choosing the models, it was also brought into a thought that all the chosen models except VGG16 specialize in working best with the ImageNet dataset while VGG 16 can work well with other datasets using its powerful learning ability. Now it is up to the performance of these models to see which performs better on the digital artwork dataset that was created. After compiling the five chosen CNN models, training is done on 50 epochs per model.

4.2.2 Testing/Running Prediction

The images from the 11 categories of the “**TEST**” folder were used to predict which class they belong to. The “**TEST**” folder contains all the fake images that were forged and thus, these images are going to be used to test their level and category of forgery in order to compare them with their respective original counterparts. The images and the labels are stored in a python dictionary during pre-processing. This dictionary was the input to the five chosen models’ prediction functions to run the testing phase. However, an individual image from the dataset can also be used to run a prediction. The result of this is supposed to show the probability of the correctly predicted class to be high while all the other classes are low.

4.3 Introducing the custom model

Despite choosing five of the recent well-performing CNN models an attempt was made to build a custom CNN model depending on the performance and architecture of the already existing CNN models. The intention was to see and analyze future possibilities for improvisation as per the comparison of the results of the five chosen models against the custom model. The model has been named **D-ARTNET22 V1** which stands for Digital-Art Neural Networks, originating in 2022 with its first version.

4.3.1 The architecture of D-ARTNET22 V1

The custom model is intentionally chosen to be sequential. This supports the architecture of the framework and allows a smooth running from the input phase. The model is built to have 24 layers allowing the model to learn with ease based on the parameters. The following describes the types of layers and parameters as to how they are appropriate for the purpose of the problem, i.e; being able to detect all sorts of forgeries within one iteration.

- **2D Convolutional Layers:** Convolutional layers are said to be the heart of CNN. They consist of a group of kernels (filters) that assist in learning while the training process takes place. These filter sizes are usually lesser than the actual image so that the filters can convolve or entwine within the image to allow the proceedings for the upcoming layers.
- **Filters:** Since the image sizes of the ARTIFICE21 V1 dataset are vast, the number of filters used started from a large amount which gradually lowered with the preceding layers. For example, the first 2D Convolutional layer starts with 256 filters or neurons whereas the last 2D Convolutional layer has 16 filters.
- **2D Max Pooling Layers:** Max pooling layers help operate pooling operations. The feature map that is produced by the filters from the 2D Convolutional Layers, has the maximum elements from its region selected by the 2D Max Pooling layer. After this operation of pooling, the generated output is a feature map consisting of the essential features of the prior feature map from the 2D Convolutional Layer.
- **Padding:** Padding ensures the number of pixels assigned to an image during its processing by Kernels. The padding is considered to be “same” in some layers so that the input and output size of that certain layer stays the same.
- **Strides:** Stride determines the number of steps taken to move while convolving through the image. Generally, in the initial layers, its value is 1 and in this model, the value of strides increases in accordance with the preceding 2D Convolutional layers and the 2D Max Pooling layers.
- **Dropout:** This layer helps to drop selective features often helping to get rid of overfitting while training.
- **Feature Maps:** Feature maps are tasked to provide results after filters are put in with the input image. Each layer outputs a specific feature map and its importance is substantiated as the features detected by the neurons are studied through feature maps. For this model, the feature map size started with $224 \times 224 \times 3$ which gradually decreased with the preceding layer’s decreasing number of filters.
- **Batch Normalization:** This layer helps with a boosted training performance by halving epochs, providing regularization, or omitting errors caused by generalization. It helps to normalize or standardize the inputs to the next layer.

- **Flatten:** The flatten layer converts all resulting 2D arrays generated by the pooled feature maps of the 2D Max Pooling layers to a single long continuous linear vector (flattened matrix). This matrix serves as input for the fully connected layer to provide the result of the classification of the image.
- **Dense:** This layer consists of neurons where each neuron takes input from each of the neurons of the prior layers. Based on the output of the previous layers, it generates the result of the classification of the image.
- **Softmax:** The softmax function serves as a squashing function which basically restricts the output between 0 and 1 hence, allowing to interpret the results as a probability distribution.
- **ReLU:** Rectified Linear Unit assists in avoiding any exponential growths within the computational power needed to operate a Neural Network model. This function does not activate all neurons simultaneously.
- **Cross-Entropy Loss:** This loss function calculates the performance of a neural network model, hence providing the output as a probability between 0 and 1.
- **Contrastive Loss:** This loss function works by creating clusters of points of the same class. They are dragged and contained in combination within the space where data is planted after dimensionality reduction. At the same time, the clusters of points from the other classes are struck away.

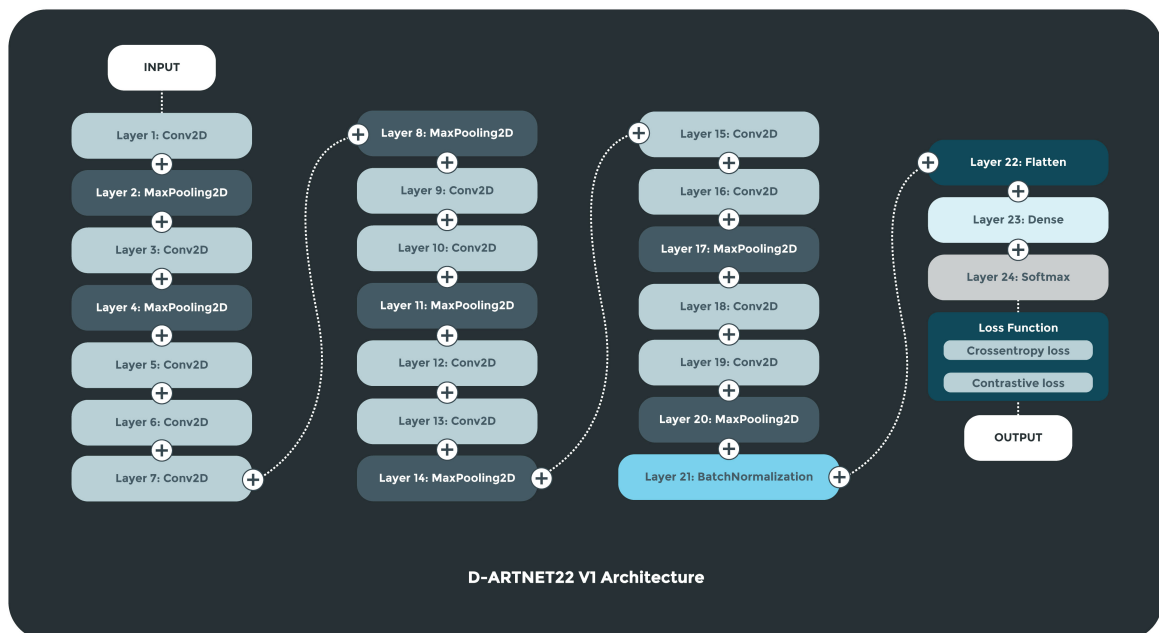


Figure 4.2: Architecture of D-ARTNET22 V1.

Figure 4.2 shows that the 2D Convolutional and the 2D Max Pooling layers are repeated except for the number of 2D Convolutional layers increasing as the layers go deeper. Only these two layers are repeated with varying parameters because the

images of the **ARTIFICE21 V1** dataset have versatile and dynamic sizes. The parameters start high and proceed with lower values so that through each layer, it can learn features from every part of the input image. Each layer is designed to learn certain parts of the image so that by the 20th layer, it learns all features of the test image. With the 20th layer wrapping up the combination of these two layers, the fully connected layers commence with the BatchNormalization layer followed by Flatten, Dense, and Softmax layers.

The BatchNormalization layer normalizes redundant features learned by all the previous 20 layers to avoid overfitting. Next, the Flatten layer transforms the linear vectored matrix from these learned features, and the Dense layer processes the final output of the classification of the image. The softmax layer is the last and the 24th layer of the model which provides the output. The model is compiled two times with the loss function once being ‘‘Cross-Entropy’’ and the next being ‘‘Contrastive’’. The average values of the two loss functions are calculated to get the overall value for training loss. The custom model was trained and tested the same way as the other chosen CNN models. A detailed description of the types of layers and parameters can be found in table 4.1.

Layers	Filters	Kernel	Pool	Pad	Strides	Feature Map Size
Input	-	-	-	-	-	$224 \times 224 \times 3$
L1:Conv2D	256	11×11	-	4×4	1×1	$55 \times 55 \times 256$
L2:MaxPool2D	-	-	3×3	-	2×2	$27 \times 27 \times 256$
L3:Conv2D	256	11×11	-	4×4	4×4	$55 \times 55 \times 256$
L4:MaxPool2D	-	-	3×3	-	2×2	$27 \times 27 \times 256$
L5:Conv2D	128	8×8	-	3×3	1×1	$27 \times 27 \times 128$
L6:Conv2D	128	8×8	-	3×3	3×3	$27 \times 27 \times 128$
L7:Conv2D	64	8×8	-	2×2	1×1	$27 \times 27 \times 64$
L8:MaxPool2D	-	-	3×3	-	2×2	$13 \times 13 \times 64$
L9:Conv2D	64	5×5	-	same	1×1	$13 \times 13 \times 64$
L10:Conv2D	64	5×5	-	same	2×2	$13 \times 13 \times 64$
L11:MaxPool2D	-	-	2×2	-	2×2	$6 \times 6 \times 64$
L12:Conv2D	32	5×5	-	same	1×1	$13 \times 13 \times 32$
L13:Conv2D	32	5×5	-	same	1×1	$13 \times 13 \times 32$
L14:MaxPool2D	-	-	2×2	-	2×2	$6 \times 6 \times 32$
L15:Conv2D	16	3×3	-	same	1×1	$6 \times 6 \times 16$
L16:Conv2D	16	3×3	-	same	1×1	$6 \times 6 \times 16$
L17:MaxPool2D	-	-	2×2	-	2×2	$6 \times 6 \times 16$
L18:Conv2D	16	3×3	-	same	1×1	$3 \times 3 \times 16$
L19:Conv2D	16	3×3	-	same	1×1	$3 \times 3 \times 16$
L20:MaxPool2D	-	-	2×2	-	2×2	$3 \times 3 \times 16$
L21	Batch Normalization					
L22	Flatten					
L23	Dense					
L24	Softmax					

Table 4.1: D-ARTNET22 V1 Architecture

4.4 Implementing Object Detection

A lot of the related works in fields like this have only used classification of a certain category using CNN models. Object Detection, on the other hand, is used more in videos or real-life scenarios, most commonly detecting and localizing pedestrians in streets. Yet, a combination of both the technologies together has never been used before in order to solve a problem like this. The initiative behind choosing to run an object detection model for this purpose is quite unique which has never been done before. Let's say the category of image forgery has been classified, but what if this piece of information was given about the number of objects in the image or if the position/coordinates of a particular object have changed or not. All of these can be used as further evidence to compare the real and fake images. Since by this time a lot of work was done starting from creating the dataset to running existing CNN models and creating a custom CNN model, the choice for the object detection model required the model to perform fast which is why YOLOV5 was chosen.

4.4.1 Object Detection Datasets

Now, to train an object detection model, annotations are required which is basically assigning bounding boxes around an object represented by a class of that type of object. Now the dilemma arises as to how many classes of objects are there. Since preparing the dataset for the classification process, running 5 algorithms, and building a custom model took a lot of time, the number of classes chosen was only one. If there were more time, then more classes representing different objects could have been worked on but now the question is why is it needed to annotate on new images when there is one of the biggest object detection datasets available, **PASCAL VOC**. The answer is that the object detection model needs to learn how objects look from an artistic perspective, i.e. how different artists tend to present objects within unique styles. Both the **PASCAL VOC** and **ARTIFICE21 V1** datasets are used to train the object detection model so that an observation can help in comparing and analyzing their effect on the model when an unknown digital artwork would be used to test the class of forgery. There are some limitations and reasons behind choosing both datasets which have been discussed below.

1. **PASCAL VOC**: This dataset contains around 1500 images for both training and testing respectively with their objects represented by 256 classes which recently took a point from 101 classes. They are still working on increasing the number of classes so that the object detection models can learn with versatility. However, the **PASCAL VOC** dataset contains objects that are photographed in real life, without any effects or filters. In that case, the model won't be able to learn about the artistic styles. Moreover, the limitation of the versatility of objects still remains as they have only 256 classes.
2. **ARTIFICE21 V1**: This digital art dataset contains 2000 images for training and 2000 images for testing where all objects are represented through one class named "**Objects**". This is for the model to be able to learn easily about any meaningful object, under only one class. **ARTIFICE21 V1** will allow the

model to learn about the objects represented in artistic styles. The place where this dataset theoretically outperforms the **PASCAL VOC** dataset is by merely decreasing the number of classes to one general class and by having more images.

4.4.2 Input Data

Here, the images from the “**TRAIN**” folder of **ARTIFICE21 V1** are to be used and the annotations are done using an online tool called MakeSense.AI where bounding boxes are assigned to each object of an image and a class is assigned to the object. The annotation can be of multiple file formats, but the one chosen here is a text file (.txt file) which represents the YOLO format. The text file contains the coordinates of the bounding boxes assigned to the objects of an image. However, for the **PASCAL VOC** dataset, the annotation files are already provided with the images and thus, images for the ratio dedicated to training are going to be the input data for running the model.

4.4.3 Training the YOLOV5 model

The YOLOV5 architecture is composed of 3 parts. The first part is the backbone of the model which works to extract essential features from the image that is fed as input. Next comes the neck of the model which in YOLOV5 uses PANet (Path Aggregation Network) as a feature pyramid method as the neck of the architecture is assigned to produce feature pyramids from the features extracted in the backbone part. Feature pyramids assist in the identification of the same object represented by various sizes. Moreover, feature pyramids help the model to achieve good performance over unknown data. Thus, PANet executes aggregation on the extracted features which are then passed to the third part of the model, which is the head. The head carries out forecasts from the anchor boxes of the objects to provide the final result. Meanwhile, Leaky RELU is used as an activation function in the hidden layers and sigmoid is used in the final layers. Additionally, SGD and Adam are used as the optimizer functions. Simultaneously, the loss function uses Binary Cross Entropy with logits loss which is used to calculate the probability of classes and the score obtained by the objects of the input image. This model is compiled and then trained on 50 epochs once with the **ARTIFICE21 V1** dataset and once with the **PASCAL VOC** dataset.

4.4.4 Testing/Running Prediction

The “**TEST**” folder contains all the fake images which are used for testing as it is required to observe a comparison between the objects of the real and fake images. At this point, the fake images will have all the objects detected which can then be compared with the objects of the real images. The variable Obj_{REAL} represents the number of objects in the real image and the variable Obj_{FAKE} represents the number of objects in the fake image. A breakdown of the analysis and final conclusion according to some specific forgery categories can be found in Table 4.2.

Comparison	Possible Forgery Category
$Obj_{FAKE} > Obj_{REAL}$ $Obj_{FAKE} < Obj_{REAL}$	Copy-Move, Splicing, Retouching.
$Obj_{FAKE} = Obj_{REAL}$	Recoloring, Blurring, Brightness/Contrast, Filters, Plain, Genuine.
$Obj_{FAKE}(w, x, y, z)$ \neq $Obj_{REAL}(w, x, y, z)$	Copy-Move, Caricaturization

Table 4.2: Analysis of the number of objects.

4.5 Maintaining Benign Information

Now that running the Deep Learning algorithms is done, one of the most essential segments of this framework requires it to be able to link benign information of all the artworks that are open to being stolen. The target plan for this segment is to serve a central database that holds information about all the artworks of all artists posted on all sorts of social media or NFT sites around the world. But now, in terms of a simulated environment, this database will contain information from the CSV file that was created while image scraping at the very beginning. In terms of theory, it is being suggested that a database would be maintained, but practically, this is done by using a Machine Learning Algorithm called Support Vector Machine (SVM). The necessary steps of this whole process are described as follows.

4.5.1 Input Data

The CSV file contains metadata, i.e. Real Artists' Names, the Caption of Artwork, Source Site, Date Posted, Image URL, and the class of forgery of the corresponding images that were scraped through various target sites. This CSV file is the input to the SVM classifier.

4.5.2 Data pre-processing

The contents of the CSV file are converted into a data frame at first. Since, all sets of information contained in this CSV file are strings, other pre-processing steps like deleting duplicate/null values, Handling/Imputing missing values, and Feature Scaling/Normalization is not needed. There are certain cells in the CSV file with information missing, like, the date of the post, or the caption. Here, the entire row can neither be deleted nor imputed as the other sets of information that are available are equally crucial. Deleting would result in losing all information for a certain artwork, and if missing values are imputed, it may work as a strength for the thief. The only necessary pre-processing step is handling categorical features. Since the dataset represents 11 classes of image forgery, the 11 classes are labeled from 0 to 10. Here, one-hot encoding is used to represent the classes so that the classifier can work with ease.

4.5.3 Training the SVM model

Support Vector Machines (SVM) is a supervised machine learning algorithm that specializes in solving linear or non-linear classification or regression problems. The scenario as per the context of the problem statement here requires it to be a classification problem. Let's say, in terms of the simulated environment, the CSV file serves as the central world database of digital artworks. The idea is to train the model with the whole CSV file so that the SVM model can know about all the true/benign information that exists.

4.6 Displaying Ground Truth

This section is basically the testing phase of the SVM classifier. By this time, the unknown image that is passed to the CNN model and the YOLO model will have its class of forgery revealed and the objects in it detected. Now, the framework will call the SVM classifier through an instance, by basically saying that "The test image has been found to be forged. Please display the true information of the real artwork". Since the SVM classifier would be trained with all the CSV file contents (benign information/ground truth), it will just cross-check the image and display the ground truth, in essence, all the true information that is available for that corresponding fake image.


4.7 Granting Permission

Following the previous segment, this part takes a decision about granting permission for a particular NFT post. Now, this segment requires some attention from the NFT authorities as NFT sites have no user or artwork verification process. While someone submits content as an NFT, they can do it pretty easily, simply by connecting a crypto wallet to the NFT site. The crypto wallet helps to maintain the transactions by maintaining credits of a cryptocurrency, most commonly ether which is used for buying or selling NFTs. This process happens within a very small amount of time for which thieves can easily register without any authorization. At this point, the team behind this research proposes the NFT authorities to run this framework and then allow the participant to mint their NFT based on the results found.

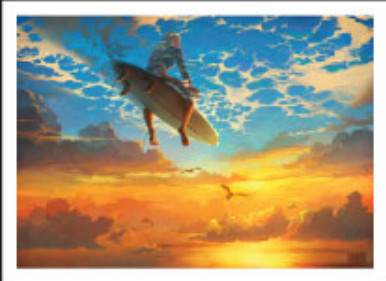
The decision breakdown is as follows. If forgery is found by the Neural Network models and ground truth is found, then the participant is not allowed to mint the artwork as an NFT. If there are no forgeries found, and no ground truth is found in the central database, it means that the test artwork is genuine and authentic, done for the first time by a real artist. If this is the case, then the participant can be allowed to mint their artwork as NFTs. An overview of how it should look is shown in Figure 4.3.

YOU HAVE BEEN REPORTED

Your Image has been found to be forged.
Details of the real owner of this image are displayed in the following.



REAL IMAGE



Posted in: DeviantArt.com

Posted on: May 17, 2014


Posted by: RHADS


Caption given: Beautiful World

Post link: <https://www.deviantart.com/rhads/art/Beautiful-World-454654081>

Post Link: <https://www.deviantart.com/rhads/art/Beautiful-World-454654081>


Social Media Site





Digital Artwork

Name of Artist



Beautiful World
by RHADS

Caption

Date Posted

Published May 17, 2014

Figure 4.3: Scenario of reporting a fraud NFT post.

Chapter 5

Results

So far, the technologies that were chosen and implemented gave the whole framework an ensemble learning experience as there is a mixture of Deep Learning, Neural Networks, and Machine Learning. Each model was trained on a high-end computer with 16GB RAM, having NVIDIA RTX 2060 6GB as its Graphical Processing Unit (GPU) and a Ryzen-5 3600 as its Central Processing Unit (CPU). The results of each of the chosen models after they were experimented on are given below.

5.1 CNN Models

5.1.1 Training

All the models were trained under 50 epochs and took a certain amount of time to finish the training process. The time taken to train the models depended on the number of layers each model had and the various sizes of the digital artworks in the dataset. The initial epochs started with low accuracy and a higher loss function which progressively improvised with the increasing number of epochs. The change of accuracy seems to have exponential growth with the increasing number of epochs while the opposite is true for the loss. The training accuracies along with the time taken are displayed in Table 5.1. A graphical representation of each model's training accuracies and training losses against the number of epochs can be found in Figures 5.1 to 5.6.

SL.	Model Name	Layers	Epochs	Time Taken	Training Accuracy
1	AlexNet	8	50	04 : 44 minutes	89.32%
2	EfficientNet-B0	237	50	25 : 15 minutes	98.17%
3	ResNet-50	50	50	16 : 45 minutes	95.21%
4	VGG-16	19	50	08 : 31 minutes	93.66%
5	MobileNet-V1	28	50	13 : 00 minutes	91.99%
6	D-ARTNET22 V1	24	50	12 : 53 minutes	98.99%

Table 5.1: Training accuracies and time taken for all CNN models

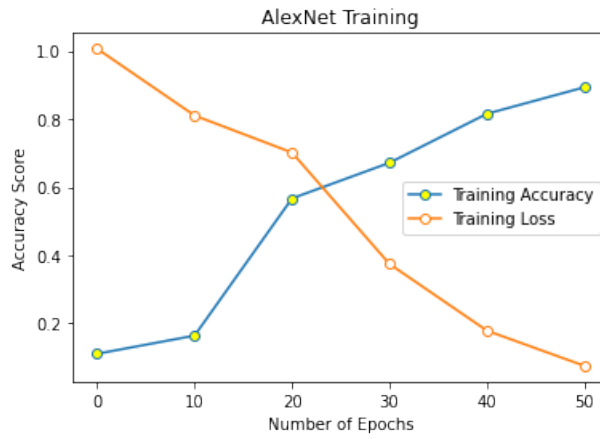


Figure 5.1: AlexNet’s training accuracy and loss against epochs

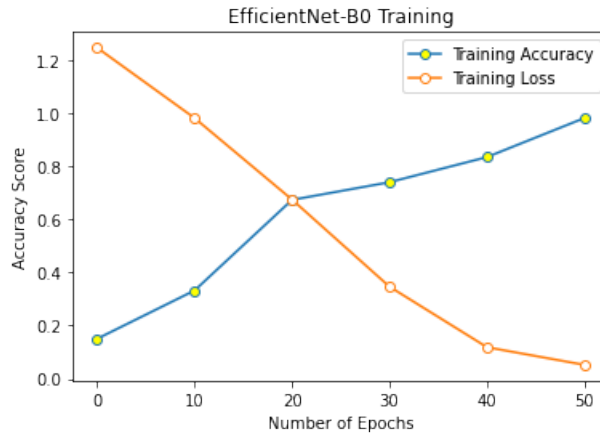


Figure 5.2: EfficientNet-B0’s training accuracy and loss against epochs

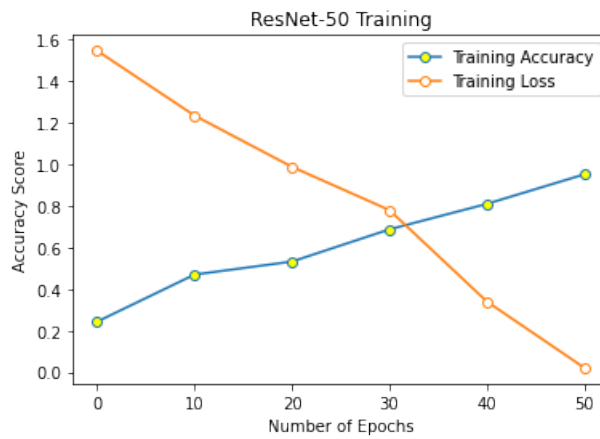


Figure 5.3: ResNet-50’s training accuracy and loss against epochs

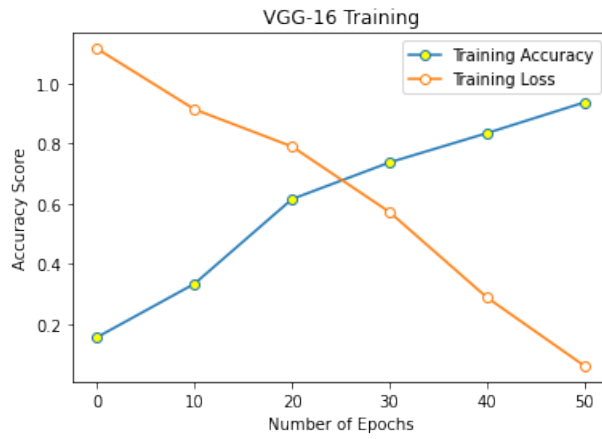


Figure 5.4: VGG-16's training accuracy and loss against epochs

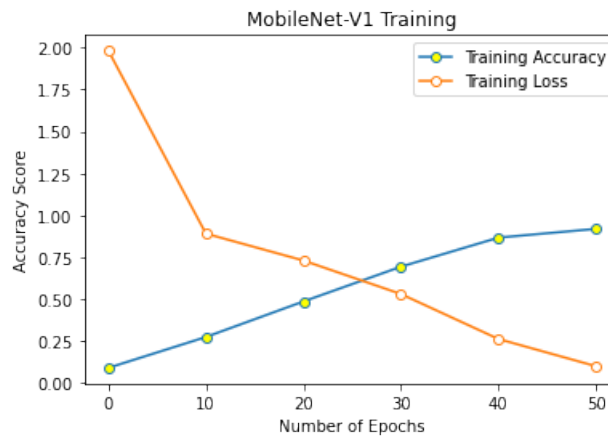


Figure 5.5: MobileNet-V1's training accuracy and loss against epochs

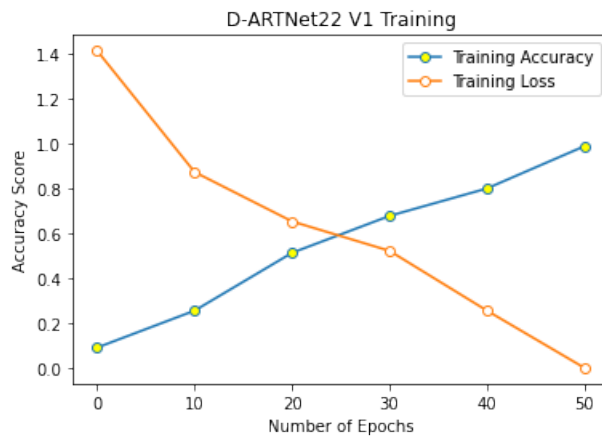


Figure 5.6: D-ARTNET22 V1's training accuracy and loss against epochs

5.1.2 Testing/Running Prediction

For the testing purpose of the classification process, as a dictionary was maintained, the key-value pairs were considered for calculating the average prediction accuracy of a certain class. The keys represent the 11 classes of forgery categories and the values of each key contain the images under those classes. The prediction/testing accuracy is calculated for each key-value pair, i.e. for all the images belonging to their corresponding forgery categories. The accuracy for each key-value pair is then averaged which is shown as the final testing/prediction accuracy. The average prediction accuracies of all the models are given in Table 5.2 and the accuracies of all the forgery categories by each of the CNN models are given in Table 5.3.

SL.	Model	Average Prediction Accuracy
Model 1	AlexNet	76.99%
Model 2	EfficientNet-B0	84.74%
Model 3	ResNet-50	80.74%
Model 4	VGG-16	80.07%
Model 5	MobileNet-V1	84.79%
Model 6	D-ARTNET22 V1	81.07%

Table 5.2: Average prediction accuracies of all chosen CNN models.

Class	Forgery	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
0	Plain	86.55%	90.76%	88.36%	90.45%	91.64%	91.74%
1	StyleGAN	67.32%	77.09%	70.18%	71.11%	79.55%	88.54%
2	Copy-Move	86.55%	90.76%	88.36%	90.45%	91.64%	91.74%
3	Splicing	72.46%	84.10%	79.15%	79.87%	84.12%	89.36%
4	Retouching	65.12%	87.53%	81.75%	68.62%	73.62%	86.79%
5	Caricatures	60.13%	80.22%	77.11%	63.55%	71.64%	87.46%
6	Blurring	74.65%	80.18%	78.24%	77.53%	86.39%	88.69%
7	Filters	87.91%	84.83%	80.26%	86.12%	89.21%	87.88%
8	B/C	81.49%	83.68%	81.63%	83.53%	89.74%	88.57%
9	Recoloring	85.37%	84.44%	81.43%	86.00%	89.44%	90.15%
10	Genuine	86.55%	90.76%	88.36%	90.45%	91.64%	91.74%

Table 5.3: Prediction accuracies of all forgery categories by each CNN model.

It can be observed that the model that performs the least is AlexNet. The reason is for it being an old model with only 8 layers. However, the models that perform the best and put up a competition are EfficientNet-B0 and MobileNet-V1. Conversely, the custom model **D-ARTNET22 V1** manages to be among the models with a mediocre performance, having a testing accuracy around the 80% mark. Figures 5.7 to 5.12 show a bar chart representation of the prediction accuracies of each model on each forgery category where the x-axis represents the forgery classes and the y-axis represents the accuracy score.

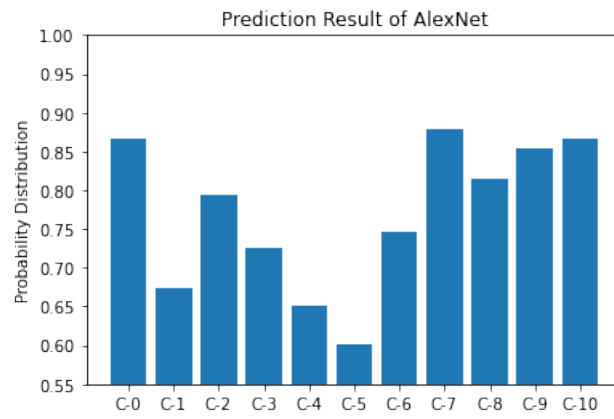


Figure 5.7: Prediction Accuracy of AlexNet for all forgery classes

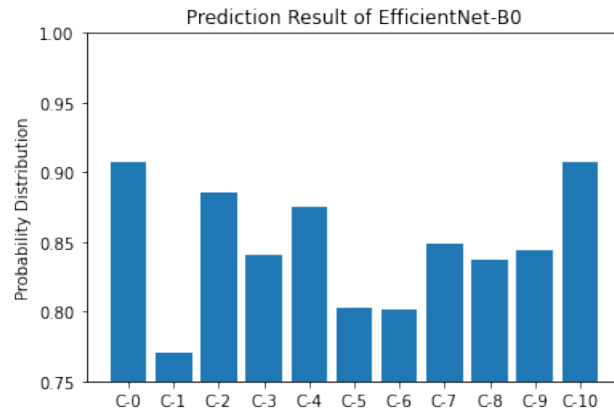


Figure 5.8: Prediction Accuracy of EfficientNet-B0 for all forgery classes

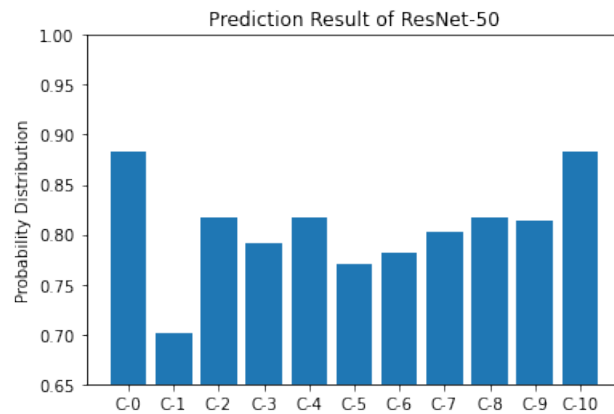


Figure 5.9: Prediction Accuracy of ResNet-50 for all forgery classes

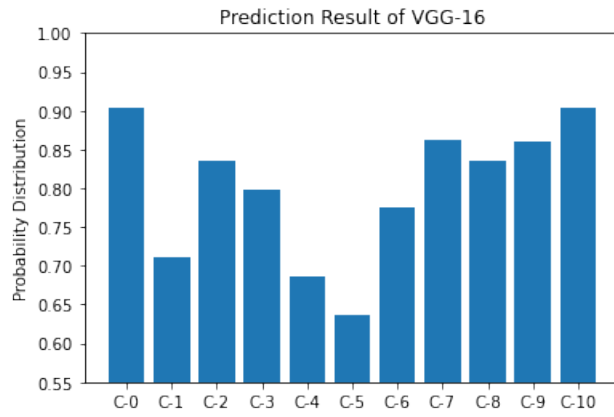


Figure 5.10: Prediction Accuracy of VGG-16 for all forgery classes

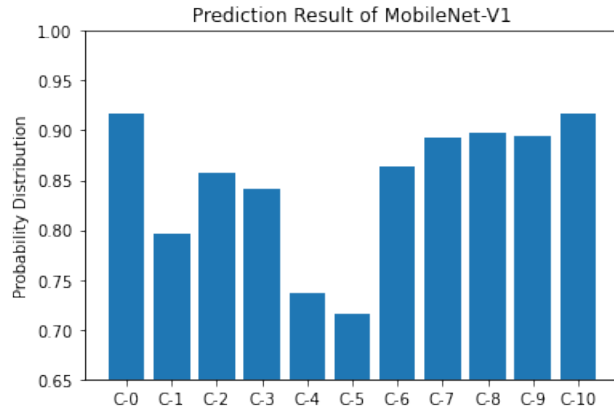


Figure 5.11: Prediction Accuracy of MobileNet-V1 for all forgery classes

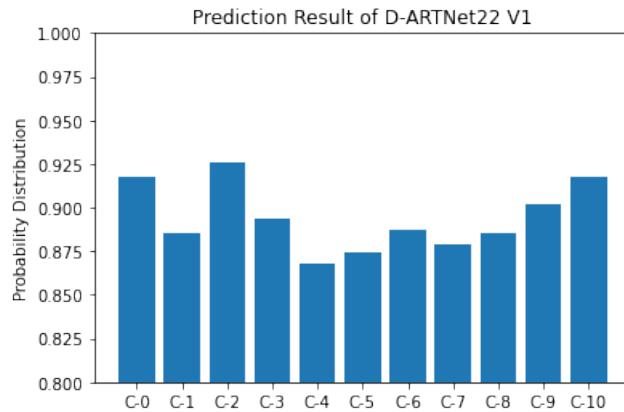


Figure 5.12: Prediction Accuracy of D-ARTNET22 V1 for all forgery classes

However, if only one image was used to predict the class of forgery, the result should look like what is shown in Figure 5.13. Since the prediction function uses the Softmax activation function, the results represent a certain probability between 0 and 1. This means that if the array containing the final result has any of the indexes valued with a higher probability, then the chosen image belongs to that index numbered class. Likewise, all the other indexes of the array must have a lower probability, essentially less than 50% to symbolize that the chosen image does not belong to those classes.

Here, the bar of class-2 shows the highest peak which means that the test image belongs to class-2 which is copy-move. Simultaneously, the model detects the test image's probability of it belonging to class-3, splicing to be almost 50%. This is because splicing is almost similar to copy-move forgery. However, the other classes show a relatively lesser probability below 30% hence confirming that the test image does not belong to those classes.

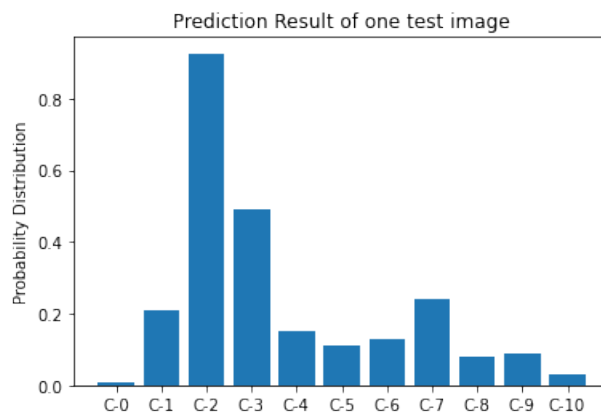


Figure 5.13: Prediction Accuracy of D-ARTNET22 V1 for one test image.

5.2 YOLO-V5 model

5.2.1 Training

The training accuracies of YOLOV5 had an inverse exponential relationship with the loss values. Figures 5.14 and 5.15 show the graphical representation of the training accuracy and loss against epochs for both the **ARTIFICE21 V1** and **PASCAL VOC** datasets. Table 5.4 shows the training accuracies of the YOLOV5 model on each dataset along with the time taken to complete the training process.

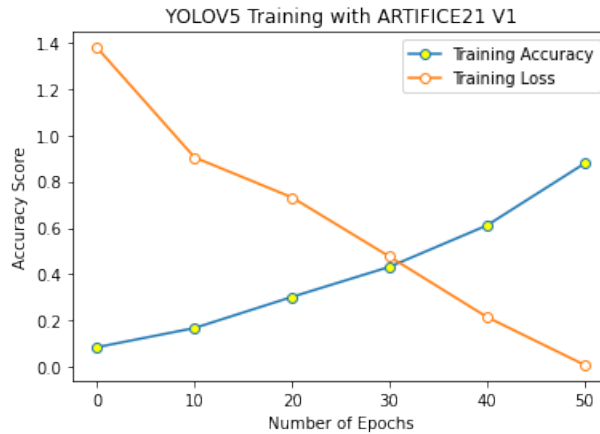


Figure 5.14: YOLOV5's training accuracy and loss against epochs for ARTIFICE21 V1

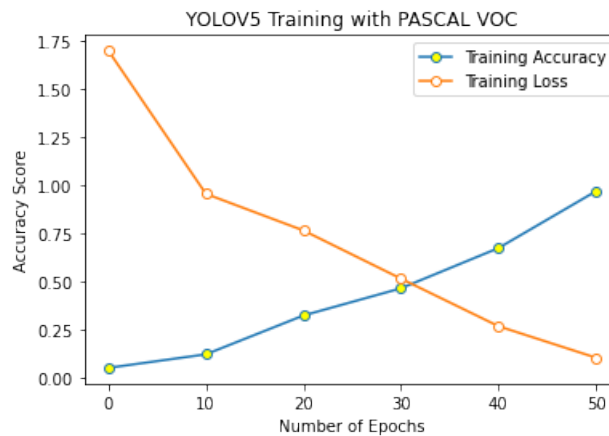


Figure 5.15: YOLOV5's training accuracy and loss against epochs for PASCAL VOC

Dataset	Epochs	Time Taken	Training Accuracy
ARTIFICE21 V1	50	23 : 37 minutes	87.78%
PASCAL VOC	50	18 : 21 minutes	96.83%

Table 5.4: Training accuracies of YOLOV5 model on each dataset

5.2.2 Testing/Running Prediction

The average accuracy of prediction of the objects in the test image from the **ARTIFICE21 V1** dataset on the YOLOV5 model was 51.88% whereas when the model is trained with the **PASCAL VOC** dataset, a test image from the ARTIFICE dataset achieved a prediction accuracy of 33.39%. Table 5.5 and Figure 5.16 manifests the testing accuracies of the YOLOV5 model when the model is trained with each dataset.

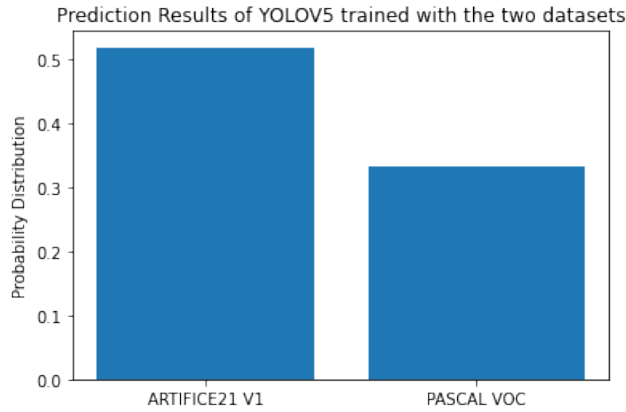


Figure 5.16: Prediction accuracy of YOLOV5 when trained with each dataset.

Dataset used to train	Prediction Accuracy
ARTIFICE21 V1	51.88%
PASCAL VOC	33.39%

Table 5.5: Prediction accuracy of YOLOV5 when trained on each dataset.

It can be seen that training with the **ARTIFICE21 V1** dataset took more time and achieved lesser training accuracy. This is because of the complex structure of digital images contained in the dataset. However, since **PASCAL VOC** is a commonly used dataset for object detection tasks it took lesser time to train with high accuracy. Conversely, in terms of predicting the objects of the test image when the model was trained with the **PASCAL VOC** dataset, the prediction accuracy is quite low. The model could detect objects but with very low probability and the anchor boxes were not exact. Unlike that, when the model was trained with the **ARTIFICE21 V1** dataset, the prediction accuracy received was a bit higher but still around the 50% mark. The anchor boxes were not exact here as well. However, the motif of comparing the number of objects in the real and fake images could be fulfilled as the model being trained under both datasets did not fail to detect a single object. Thus, the dilemma lies in a common entity for both datasets. **PASCAL VOC** does not allow the model to learn about objects represented via artistic styles, whereas **ARTIFICE21 V1** has the number of classes limited to only one. Therefore, to improve the accuracy of predicting objects in digital artworks, the **ARTIFICE21 V1** dataset has to have at least the same number of classes that the **PASCAL VOC** dataset has. This way, the model would be able to learn about artistically represented objects quite easily. Figure 5.17 demonstrates the output of the YOLOV5 model, trained with **ARTIFICE21 V1** while predicting objects in a

test image. Figure 5.18 shows the same where the model is trained with **PASCAL VOC**.

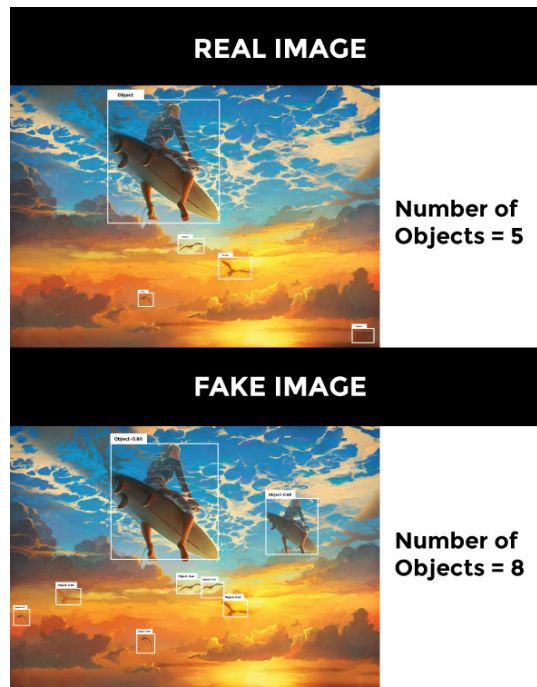


Figure 5.17: Output of YOLOV5 when trained with **ARTIFICE21 V1**.

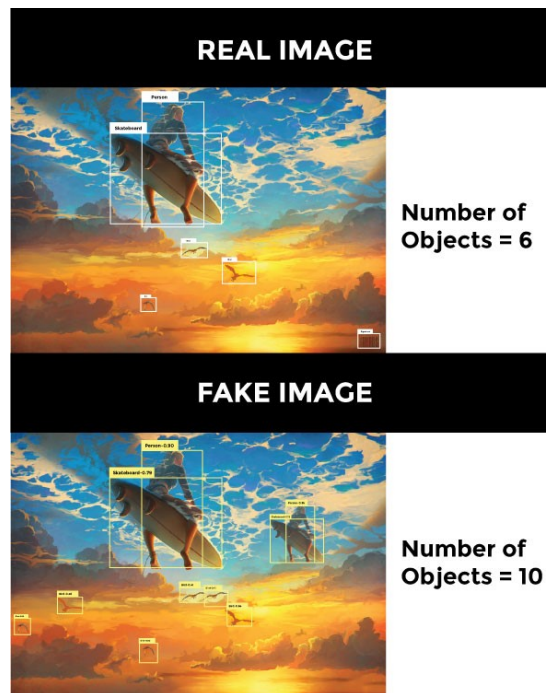


Figure 5.18: Output of YOLOV5 when trained with **PASCAL VOC**.

This test image went through copy-move forgery and it can be further confirmed by using Table 4.2 as the number of objects seems to vary for both the real and fake images. The only difference is that the output is different when the model is trained with both datasets. Since the **PASCAL VOC** dataset has 256 classes of objects,

the output shows that it detected exact objects, whereas the **ARTIFICE21 V1** dataset detected the object in focus. Both the datasets contributed differently to the YOLOV5 model in terms of detecting objects, but the intention of being able to detect and compare the number of objects in the real and fake images could be met perfectly.

5.3 SVM Classifier

5.3.1 Training

Training the SVM model with the CSV file of **ARTIFICE21 V1** took 4 minutes to train with an accuracy of 98.43%.

5.3.2 Testing/Running Prediction

When an image was found to be forged, and an instance of the SVM classifier was called, it could display the benign information of that corresponding fake image correctly, with an accuracy of 93.31%. Table 5.6 shows the summary of the performance of the SVM Classifier.

Classifier	Time Taken	Training Accuracy	Testing Accuracy
SVM	04 : 00 minutes	98.43%	93.31%

Table 5.6: Performance of the SVM Classifier.

Chapter 6

Conclusion

NFT theft cases in recent times have raised a huge concern for any artist. Digital artworks are stolen and minted as NFTs which confirms a permanent record in the Ethereum blockchain after a transaction is successful. Since no verification process exists to date, one can register NFTs effortlessly. Therefore, for the first time, an approach was thought to bring into work starting with this paper. As a result of no work existing related to this field, certain things were considered to assemble a step-by-step process, eventually making this approach novel and ensemble. One obvious barrier was that no dataset consisted of genuine digital artworks which was a challenging thing to overcome. Thus, digital artworks were scraped and the extent of fakeness was achieved through various forgery protocols. Hence, the plan stands out to present the first-ever customized digital artwork dataset which will help globally in any related research work. Moreover, the custom model also performed well with an accuracy of about 81%. The results were quite promising for which it can be convincing for the NFT site authorities to adopt this model into a live implementation where they can compare it with the entire database of digital artworks existing in this world. Conversely, as prominent it is, NFTs being any scarce virtual collectible opens opportunities to explore other NFT use cases like GIFs, video clips, 3D models, etc. Unquestionably, this can be a light of hope for artists like Michael Miraflor or Derek Laufman who belong to this career making a living out of it.

If it is thought in vintage terms, when artworks used to be physically stolen from museums, the real artist used to get famous, as people used to think that the artwork must be worth high values for which it was stolen and could be sold or auctioned in black money. In the digital world, NFTs have high demand, and low supply, hence giving it the property of non-fungibility, which is what makes NFTs so expensive. NFT sites would eventually make a lot of gas money because of the amount of NFTs getting minted daily and since there exists no verification process of either the participant performing the mint, or the content getting minted, the thief hampers the reputation of the original artist, as the original artist is not at all aware of it. Pastel Sense AI, originating in 2021, only checks the originality of an NFT if and only if it exists within the Ethereum Blockchain, but it does not consider looking throughout the whole internet to check if an artwork minted in an NFT site came from any other social platforms or not. The social media marketplace and the NFT market places are drastically different, but the way thieves connect the dots, the complaints of various famous artists proves the fact that they are getting discouraged, due to

their originality, reputation and hardwork getting questioned like this. Adding the extent of forgery, takes it to a whole new level as the forged artwork gets famous, without the acknowledgement of the real artist, causing normal people to think that the thief was the real source. Even if a common third person comes to know about the artwork getting registered under a different name, and then informs the original artist, it gets too late, as it is permanently stored in the Blockchain.

6.1 Limitations

Since this research work is thought of as a novel and ensemble approach, the biggest limitation faced was the unavailability of a digital art dataset. Despite being able to create one, the number of images was quite limited, with each forgery category having fewer variations. Most importantly, the scraped images had various image sizes due to the freedom of an artist being able to choose canvas sizes of their own. This consequently required the model to be more complex, with more layers so that the model could learn about the differently sized image features properly. The model was designed to reach that peak but eventually had the number of layers lessened, so as to run a model with more layers, more computational power would be required, that is the graphical processing power needed would be more even than of an NVIDIA RTX 2060 6GB.

6.2 Future Works

The future scope of this field is quite vast. Since the prime target was to work with only digital arts, it is intended that this field is improvised on as this is the most occurring or engaging use case out of all. In that regard, the scope of the digital art dataset is to be increased with more images and with each image forgery technique getting more attention and variation in terms of basic to advanced forms. Connecting to that, the annotations for this very dataset need more variations of classes just like the **PASCAL VOC** dataset, but the difference is that it needs to be digital artworks. This is in the plans to pull off so that **ARTIFICE21 V1** can serve as a digital artwork version of the **PASCAL VOC** dataset that can be contributed globally for various research works.

On another end, it is also substantiated that a thief may apply multiple image forgery techniques in one image for which the plan is to combine the CNN and object detection model into a hybrid entity so that if the thief forges a certain fraction of the image, the object detection model can detect that area and then the CNN model can learn features only from that area to get more prominent results.

Bibliography

- [1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [2] G. R. Kumar, R. K. Kumar, and G. Sanyal, “Discriminating real from fake smile using convolution neural network,” in *2017 International Conference on Computational Intelligence in Data Science (ICCIDS)*, IEEE, 2017, pp. 1–6.
- [3] R. Nemade, A. Nitsure, P. Hirve, and S. B. Mane, “Detection of forgery in art paintings using machine learning,” *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 6, no. 5, pp. 8681–8692, 2017.
- [4] J. Ouyang, Y. Liu, and M. Liao, “Copy-move forgery detection based on deep learning,” in *2017 10th international congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI)*, IEEE, 2017, pp. 1–5.
- [5] S. M. Abbas and S. N. Singh, “Region-based object detection and classification using faster r-cnn,” in *2018 4th International Conference on Computational Intelligence & Communication Technology (CICT)*, IEEE, 2018, pp. 1–6.
- [6] F. Marra, D. Gragnaniello, D. Cozzolino, and L. Verdoliva, “Detection of gan-generated fake images over social networks,” in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, IEEE, 2018, pp. 384–389.
- [7] T. Pomari, G. Ruppert, E. Rezende, A. Rocha, and T. Carvalho, “Image splicing detection through illumination inconsistencies and deep learning,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2018, pp. 3788–3792.
- [8] S. Smirnov and A. Eguizabal, “Deep learning for object detection in fine-art paintings,” in *2018 Metrology for Archaeology and Cultural Heritage (MetroArcheo)*, IEEE, 2018, pp. 45–49.
- [9] S. Tariq, S. Lee, H. Kim, Y. Shin, and S. S. Woo, “Detecting both machine and human created fake face images in the wild,” in *Proceedings of the 2nd international workshop on multimedia privacy and security*, 2018, pp. 81–87.
- [10] Z. Xiong, Y. Yuan, and Q. Wang, “Ai-net: Attention inception neural networks for hyperspectral image classification,” in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2018, pp. 2647–2650.
- [11] Y. Yan, W. Ren, and X. Cao, “Recolored image detection via a deep discriminative model,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 5–17, 2018.

- [12] Y. Abdalla, M. T. Iqbal, and M. Shehata, “Convolutional neural network for copy-move forgery detection,” *Symmetry*, vol. 11, no. 10, p. 1280, 2019.
- [13] A. K. Jaiswal and R. Srivastava, “Image splicing detection using deep residual network,” in *Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE)*, 2019.
- [14] K. B. Meena and V. Tyagi, “Image forgery detection: Survey and future directions,” in *Data, Engineering and applications*, Springer, 2019, pp. 163–194.
- [15] L. Nataraj, T. M. Mohammed, B. Manjunath, *et al.*, “Detecting gan generated fake images using co-occurrence matrices,” *Electronic Imaging*, vol. 2019, no. 5, pp. 532–1, 2019.
- [16] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International conference on machine learning*, PMLR, 2019, pp. 6105–6114.
- [17] R. Thakur and R. Rohilla, “Copy-move forgery detection using residuals and convolutional neural network framework: A novel approach,” in *2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC)*, IEEE, 2019, pp. 561–564.
- [18] L. Zheng, Y. Zhang, and V. L. Thing, “A survey on image tampering and its detection in real-world photos,” *Journal of Visual Communication and Image Representation*, vol. 58, pp. 380–399, 2019.
- [19] M. Jijina, L. Koshy, and G. S. Warriar, “Detection of recoloring and copy-move forgery in digital images,” in *2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, IEEE, 2020, pp. 49–53.
- [20] A. Kumar and S. Srivastava, “Object Detection System Based on Convolution Neural Networks Using Single Shot Multi-Box Detector,” *Procedia Computer Science*, vol. 171, pp. 2610–2617, 2020. DOI: 10.1016/j.procs.2020.04.283.
- [21] Y. Rao, J. Ni, and H. Zhao, “Deep learning local descriptor for image splicing detection and localization,” *IEEE Access*, vol. 8, pp. 25 611–25 625, 2020.
- [22] C. Rathgeb, C.-I. Satnoianu, N. E. Haryanto, K. Bernardo, and C. Busch, “Differential detection of facial retouching: A multi-biometric approach,” *IEEE Access*, vol. 8, pp. 106 373–106 385, 2020.
- [23] S. Samir, E. Emary, K. El-Sayed, and H. Onsi, “Optimization of a pre-trained alexnet model for detecting and localizing image forgeries,” *Information*, vol. 11, no. 5, p. 275, 2020.
- [24] V. Varkarakis, S. Bazrafkan, and P. Corcoran, “Re-training stylegan-a first step towards building large, scalable synthetic facial datasets,” in *2020 31st Irish Signals and Systems Conference (ISSC)*, IEEE, 2020, pp. 1–6.
- [25] Y. Wang, L. Wang, Y. Jiang, and T. Li, “Detection of self-build data set based on yolov4 network,” in *2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE)*, IEEE, 2020, pp. 640–642.

- [26] X. Wu, R. Liu, H. Yang, and Z. Chen, “An xception based convolutional neural network for scene image classification with transfer learning,” in *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, IEEE, 2020, pp. 262–267.
- [27] B. Xiao, Y. Wei, X. Bi, W. Li, and J. Ma, “Image splicing forgery detection combining coarse to refined convolutional neural network and adaptive clustering,” *Information Sciences*, vol. 511, pp. 172–191, 2020.
- [28] Y. Yin, H. Li, and W. Fu, “Faster-yolo: An accurate and faster object detection method,” *Digital Signal Processing*, vol. 102, p. 102756, 2020.
- [29] Barsky, “Non-fungible tokens and intellectual property law: Key considerations,” *Holland Knight*, 2021.
- [30] H. Cai, Y. Guo, Z. Peng, and J. Zhang, “Landmark detection and 3d face reconstruction for caricature using a nonlinear parametric model,” *Graphical Models*, vol. 115, p. 101103, 2021.
- [31] D. Liscia, “Reports of stolen art on nft marketplace raise issues for crypto collectors,” *Hyperallergic*, 2021.
- [32] B. Stephen, “Nft mania is here, and so are the scammers,” *The Verge*, vol. 20, 2021.
- [33] C. Xianbao, Q. Guihua, J. Yu, and Z. Zhaomin, “An improved small object detection method based on yolo v3,” *Pattern Analysis and Applications*, vol. 24, no. 3, pp. 1347–1355, 2021.
- [34] A. C. T. Tumblr and A. C. Team, *Nft money laundering and wash trading*, May 2022. [Online]. Available: <https://blog.chainalysis.com/reports/2022-crypto-crime-report-preview-nft-wash-trading-money-laundering/>.
- [35] X. Long, K. Deng, G. Wang, *et al.*, “Pp-yolo: An effective and efficient implementation of object detector. arxiv 2020,” *arXiv preprint arXiv:2007.12099*,