AFRL-RI-RS-TR-2021-180

PHYSICAL AND SEMANTIC INTEGRITY MEASURES FOR MEDIA FORENSICS

UNIVERSITY OF MARYLAND

OCTOBER 2021

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2021-180 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /

JEFFREY T. CARLO
Work Unit Manager

/ S /

JAMES S. PERRETTA
Deputy Chief, Information
Exploitation & Operations Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
OCTOBER 2021	FINAL TECHNICAL REPORT	MAY 2016 - MAY 2021

4. TITLE AND SUBTITLE

PHYSICAL AND SEMANTIC INTEGRITY MEASURES FOR MEDIA FORENSICS

5a. CONTRACT NUMBER
FA8750-16-2-0191

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62303E

6. AUTHOR(S)

Abhinav Shivastava
Yaser Yacoob
Larry Davis

5d. PROJECT NUMBER
MEDI

5e. TASK NUMBER
40

5f. WORK UNIT NUMBER
13

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Maryland
Office of Research Administration
3112 Lee Bldg 7809 Regents Dr.
College Park MD 20742-0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIGC
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2021-180

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This is a final report on research and development program carried out in accordance with its proposal entitled "Physical and Semantic Integrity Measures for Media Forensics". The research spans multiple media types, images, video and metadata, and of which may have been manipulated from original form. Methods of manipulation include parts or whole images and video, specific objects such as faces or general scenes of all varieties.

15. SUBJECT TERMS

Media forensics, integrity analytic, media manipulation, multi-media analytics

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			**JEFFREY T. CARLO**
U	U	U	UU	48	19b. TELEPHONE NUMBER (*Include area code*) **NA**

TABLE OF CONTENTS

Section **Page**

LIST OF FIGURES

List of Tables

1.0 Summary

In this effort a range of algorithms for automatic detection of image and video tampering were developed and most were transitioned to DARPA and other DOD partners. The research effort set out to model and uncover physical clues in an image and use these to confirm or detect inconsistent cues in images and video. This was accomplished for the specific domains of faces where there is sufficient constraints to carry the task. The problem proved harder for unconstrained open world scenery where structural variability complicates the analysis.

A second area of research focused on metadata tampering of images, two problems were considered, in the first a single image metadata is changed and the objective is to confirm or detect inconsistency between metadata and image content. The second is insertion of an image to a collection of images that represent an event, posing the problem as an imposter image detection.

Much of the landscape of forensic analysis changed during the performance period as new threats emerged (namely Deepfake images and videos, adversarial attacks). Significant effort was directed to these new threat and significant progress was made on analysis and detection of these threats.

2.0 Introduction

Image manipulation has existed since photography was invented. It allows changing the message conveyed for benign or maligned objectives. Massive expansion and sharing of imagery and video make the problem much wide spread and intractable. Particularly, in public discourse, the generation ad dissemination of media makes it impossible for humans to inspect and analyze media. Therefore, automation is critical to cope with the quantitative and qualitative scale of manipulations.

MEDIFOR sought to create algorithms that operate at scale to detect tampering in media. It also defined a very broad scope of manipulations that encompasses all aspects of media handling. Our research was focused on the physical and semantic cues that are available in a media item or a collection of items. In both cases the focus was on inconsistencies in information present in the media item. For example, in the case of face tampering in a single image, we developed algorithms for estimating prevalent illumination at each face and comparing these models among faces in an image.

The research described encompassed a wide range of topics, as it reflects the wide scope of media tampering that may be present. The scope includes images, videos, image sets and metadata. The media was significant in size, encompassing millions of images, over 150,000 video clips and tens of thousands of metadata items.

The research resulted in tens of publications as well as software packages that were transitioned to DARPA, DOD and commercial entities.

3.0 Methods, Assumptions, and Procedures

The research employed and advanced the state of the art in computer vision algorithms. It employed the frontier of understanding of image formation where it is available and effective to tackle the open-world scope of the problem. It also employed data-centric approaches where patterns are learned from data collections and these patterns serve as a metric for comparing new media. The latter approach overcomes, to some extent, the intractable nature of the data that is available and the large space of manipulations that may be used.

In certain algorithms (e.g., face illumination, GANs, and Deepfakes) it is necessary to employ full understanding of the attributes of the face as possible. This improves our ability to model and detect manipulations. In the case of illumination analysis, we employed and advanced the state of the art on this topic. In the case of Deepfake videos, we developed temporal analysis algorithms to uncover subtle motion differences between natural and manipulated videos.
In other algorithms, such as metadata tampering and event association, there is very little a priori information to employ, an image or a collection of images can be of unpredictable domains. We devised procedures for data collection and analysis to enable comparing data in multi-dimensional spaces. Thus conforming to MEDIFOR program's objectives of providing a single authenticity score to a given media item regardless of the content and source of manipulation.
As synthesized media came to exist in the form of GAN faces, other assumptions and processes needed to be introduced, as the media item no more represented directly the real world. Synthesized faces have attributes that can be studied only in the context of the generation capability. In this case we rely on access to data generated by the adversaries' algorithms to find fingerprints that enable detection.

3.1 Face Tampering Detection

People post photos every day on popular social websites such as Facebook, Instagram and Twitter. A considerable number of these photos are authentic as they are generated from people's real life and shared as a part of their social experience. However, maliciously or not, more and more tampered images, especially ones involving face regions, are emerging on the Internet. Image splicing, which is the most common tampering manipulation, is the process of cutting one part of a source image, such as the face regions, and inserting it in the target image. To make the tampered result more realistic, adjustments on the shape, boundary, illumination and scaling are necessary, which make tamper detection challenging. Given advances in face detection and recognition techniques, anyone is able to swap faces with low cost using mobile applications [1] or open-source software [2]. Some tampered image examples generated from commercial software are shown in ***Figure 1***. Even after close inspection, people mistake the tampered faces as original, and the current face verification techniques will also determine the source face and the tampered face are from the same identity. The consequence would be even more serious if manipulated images are used for political or commercial purposes. Detecting spliced or tampered faces in images is an important and challenging problem. We developed two approaches to detect such manipulations.

Figure 1. *Examples of tampered faces. (a) Original image. (b) Tampered image. The face in the middle has been tampered. (c) Original image. (d) Tampered image. The face on the right has been tampered.*

3.1.1 Using Illumination

We developed techniques that rely on a global analysis of a face to compute illumination features that are then used to identify if two faces in an image have inconsistent illumination properties and thus are likely to have originated from different images. It operates on whole faces, analyzes pairs of faces at a time, and does not require tampered faces as training examples. We base our irradiance estimation on the SIRFS framework of Barron and Malik. Their framework simultaneously attempts to disentangle 3D shape, reflectance and illumination at once, given visual priors that are common for objects in general scenes.

We constrain this SIRFS estimation process by using face detection, face fiducials detection, head pose, and morphable models to derive better constrained estimates of 3D shape, reflectance and illumination. The illumination model is expressed in a second order spherical harmonic representation for the spatial distribution of irradiance across the spatial area of the face. This is a 27 dimensional vector (9 parameters for each of RGB). Generally, the distance between illumination models reflects similarity (or constancy of illumination). At the same time, the illumination distance can be affected by highly diverse skin color (e.g., white versus brown skin in the same image), which requires further differentiation.

As our research evolved, this system has been replaced by a Deep network that estimates illumination directly. These systems were trained on data derived from SIRFS models computed of MEDIFOR datasets. The software containers were incorporated in the MEDIFOR system. Since there was only small differences in performance between these approaches, the fastest approach was submitted for MEDIFOR use, this is names the **umd_illumination_DPR** system and uses components from earlier containers.

3.1.2 Two-Stream Network for Digital Signatures

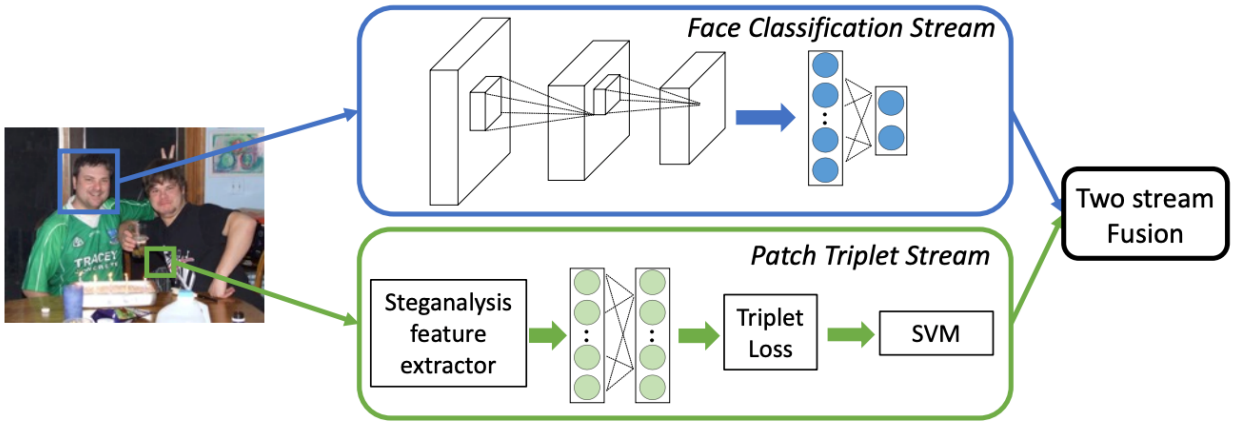


***Figure 2**. Illustration of our two-stream network where the scores of two streams are fused to recognize a tampered face.*

This technique relies on a patch-based analysis to determine if the statistics of the patches within a face bounding box are consistent with the statistics of non-face patches within the test image. It operates on face patches, analyzes a single face at a time, and does not require tampered faces as training examples. While the method relies on training, this is done on-the-fly and does not require tampered examples, since the classifier is trained to identify patch statistics of the test image versus patch statistics of other images. An overview of the network is shown in Figure 2, where the face classification stream is a CNN trained to classify whether a face image is tampered or authentic and learns the artifacts created by the tampering process. The Patch-triplet stream is trained on steganalysis features of image patches with a triplet loss, models the traces left by in camera processing and local noise characteristics. For this, we first extract 512 dimensional steganalysis rich model features for image patches in a sliding window fashion (window size = 128, stride = 64). Then a triplet network is trained to enforce that a pair of patches from the same image are closer in the learned embedding space, while the distance between a pair of patches from two different images is large. Finally, we apply the trained SVM on face patches, and the average score of face patches is used as the face-level score.

The algorithms were tested on virtual splicing of faces, and a dataset was shared with the MEDIFOR community. Please refer to [3] for details on this work.

3.2 Metadata Tampering Detection from Weather Records

Tampered image metadata is frequently encountered in image forensics. Ease of metadata access and modification using simple EXIF tools has resulted in tampered images that are difficult to detect. Our goal is to detect information such as sun altitude angle, temperature, humidity and weather conditions using multi-task deep learning. We then compare the inferred properties to the same information collected from the Internet based on image metadata to detect if there is any tampering.

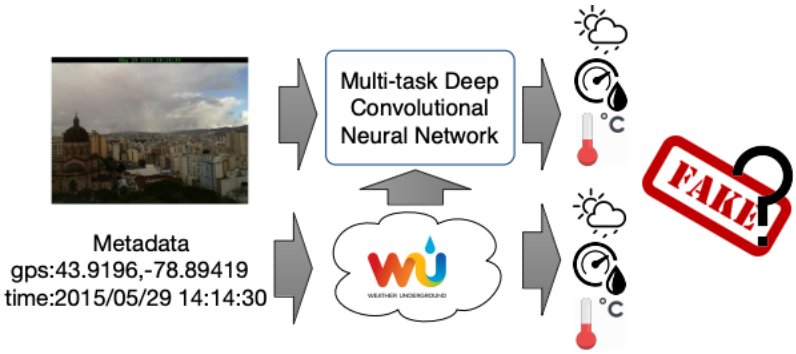


Figure 3. *Metadata Tampering Detection from Weather Records*

We used convolutional neural network (CNN) models to predict sun angle and meteorological information. We experimented with two variants of convolutional models for our prediction tasks: AlexNet and ResNet-50. AlexNet contains five convolutional layers followed by three fully connected layers, while ResNet-50 contains 49 convolutional layers with residual connections followed by one average pooling layer. We used AlexNet to experiment with different loss functions (mean squared and mean absolute losses) due to the advantage of its training speed and use ResNet-50 to train our final model to obtain better prediction results.

We trained a CNN for regression tasks of temperature, humidity and sun angle estimations. We first replace the last layer of the CNN with a single output using a distance-based loss function. Since the outputs of our regression models should always lie in certain ranges (e.g. zero to ninety degrees for sun angle), we use a sigmoid or an extra ReLU-like nonlinear layer to clip the output from both sides before the final loss layer; but they improve performance only in some cases whereas decrease performance in others. We also weight the training loss based on the probability distribution of the ground truth labels and call these the weighted regression models. This helps to give more importance to the examples that are less common in the training set and tries to solve the problem that the dataset is not uniformly distributed. Finally, we train the network with our AMOS+M2 dataset.

We also trained a CNN for general weather classification using our AMOS+M2 dataset. We first separate our training data into four different classes: sunny, cloudy, rainy, and snowy. Since our training set is highly unbalanced, as sunny and cloudy images together take around 85% of the training set, directly training the network would cause the model to be biased toward sunny and cloudy. To address this issue, we apply data oversampling with augmentation: for each image class, we first oversample the images to make each class have roughly the same size, and then we apply data augmentation to each oversampled image by first randomly resizing and keeping the smallest side of the image between 256 to 512 pixels. We then randomly crop the image down to 227 227 and randomly apply a left-right flip to the image. Finally, we adopt the softmax cross entropy loss function to optimize the network parameters. In order to reduce the training time, we initialize the weights of our network to a model pretrained on ImageNet dataset.

Since all of the meteorological information we use is correlated, it is natural to wonder if one model can benefit from the others. Therefore, we use multi-task learning to learn a joint model that can predict all the meteorological information at the same time. This is achieved by weight sharing on all the regression and classification networks with a joint loss function. We adopt the same network

architecture, ResNet-50, for all tasks so that we can share the weights crossing all four tasks.

Please refer to [4] for details of this work.

3.3 Image Verification with respect to Events

Verifying the authenticity of a given image is an important topic in media forensics research. Many current works focus on content manipulation detection, which aims to detect possible alteration in the image content. However, tampering might not only occur in the image content itself, but also in the metadata associated with the image, such as timestamp, geo-tag, and captions. We address metadata verification, aiming to verify the authenticity of the meta- data associated with the image, using a deep representation learning approach. See example in Figure 4, where images with the blue border are positive samples and images with the red border are negative samples.

We proposed a deep neural network called Attentive Bilinear Convolutional Neural Networks (AB-CNN) that learns appropriate representation for metadata verification. AB-CNN address several common challenges in verifying a specific type of metadata – event (i.e. time and places), including lack of training data, fine- grained differences between distinct events, and diverse visual content within the same event. Experimental results on three different datasets show that the proposed model can provide a substantial improvement over the baseline method.

Figure 4. *Image Verification with respect to Events: (a) Given set of images with the same metadata (i.e. taken from a known event) and (b) some probe images.*

Following is the system overview of the proposed network configuration. The system contains three main modules. (1) An augmented training set is collected from the Internet and used to pre-train a multi-class for transfer learning. (2) Bilinear pooling is adapted to model the feature correlation. (3) An attention module is utilized to automatically learn to handle the diverse visual content within an event image set.

3.4 Image Tampering Detection

With the advances of image editing techniques and user-friendly editing software, low-cost tampered or manipulated image generation processes have become widely available. Among tampering techniques, splicing, copy-move, and removal are the most common manipulations. Image splicing copies regions from an authentic image and pastes them to other images, copy-move copies and pastes regions within the same image, and removal eliminates regions from an authentic image followed by inpainting. Often, complicated post-processing techniques are used

to further improve the quality of images. Our goal is to figure out which region/pixels has been manipulated. We propose two approaches towards it.

3.4.1 RGB-Noise Network

Our network adopts Faster R-CNN within a two-stream network and perform end-to-end training. A summary of our method is shown in Figure 5. The RGB stream models visual tampering artifacts, such as unusually high contrast along object edges, and regresses bounding boxes to the ground-truth. The noise stream first obtains the noise feature map by passing input RGB image through an SRM filter layer, and leverages the noise features to provide additional evidence for manipulation classification. The RGB and noise streams share the same region proposals from RPN network which only uses RGB features as input. The RoI pooling layer selects spatial features from both RGB and noise streams. The predicted bounding boxes (denoted as 'bbx pred') are generated from RGB RoI features. A bilinear pooling layer after RoI pooling enables the network to combine the spatial co-occurrence features from the two streams. Finally, passing the results through a fully connected layer and a softmax layer, the network produces the predicted label (denoted as 'cls pred') and determines whether predicted regions have been manipulated or not.

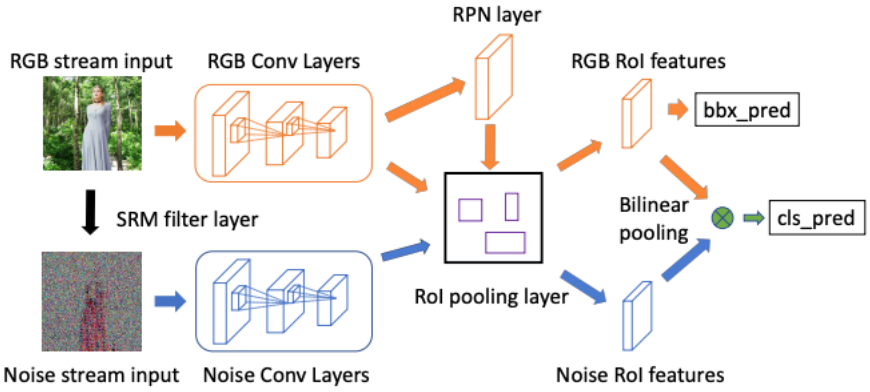


Figure 5. *Illustration of RGB-N Network, our two-stream Faster R-CNN network.*

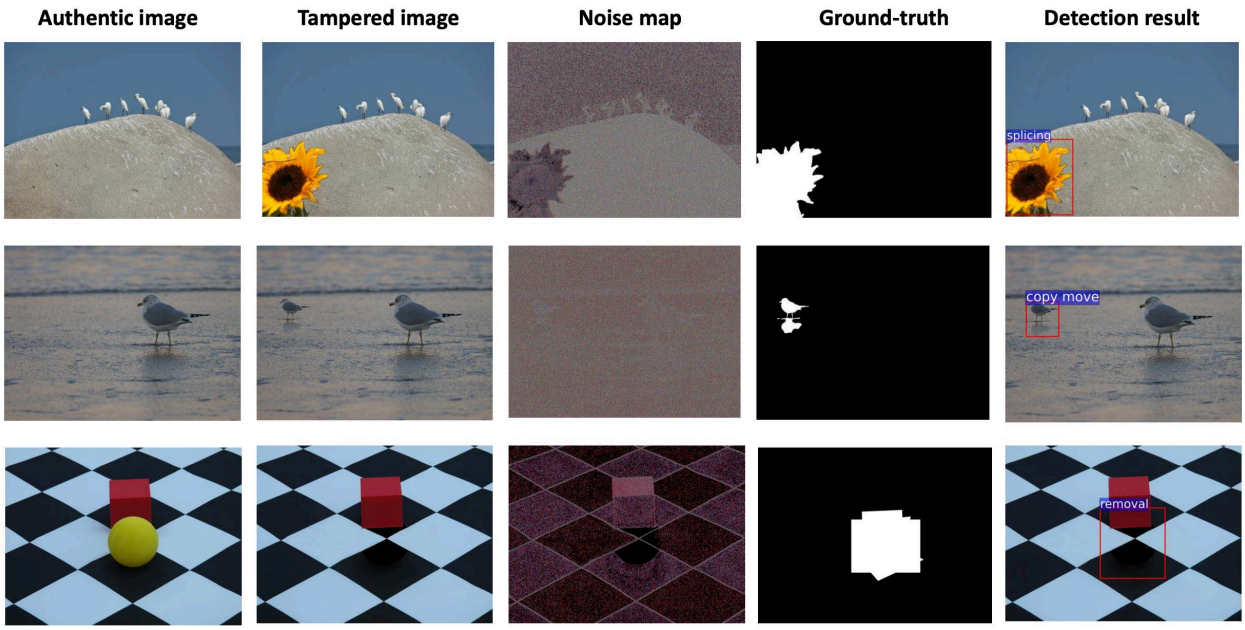


***Figure 6**. Qualitative results for multi-class image manipulation detection on NIST16 dataset.*

The rich feature representation of our network enables it to distinguish between different manipulation techniques as well. We explore manipulation technique detection and analyze the detection performance for all three tampering techniques. NIST16 contains the labels for all three tampering techniques, which enables multi-class image manipulation detection. We change the classes for manipulation classification to be splicing, removal and copy-move so as to learn distinct visual tampering artifacts and noise features for each class. Qualitative examples are shown in ***Figure 6*** where RGB and noise map provide different information for splicing, copy-move and removal. By combining the features from the RGB image with the noise features, RGB-N produces the correct classification for different tampering techniques.

Please refer to [5] for details.

3.4.2 GSR Network

We introduce a two-pronged approach to (1) address the lack of comprehensive training data, as well as, (2) focus the training process on learning to recognize boundary artifacts better. We adopt GANs for addressing (1), but instead of relying on prior GAN methods that mainly explore image level manipulation, we introduce a novel objective function that optimizes for the realism of the manipulated regions by blending tampered regions in existing datasets to assist segmentation. That is, given an annotated image from an existing dataset, our GAN takes the given annotated regions and optimizes via a blending based objective function to enhance the realism of the regions. Blending has been shown to be effective in creating training images effective for the task of object detection, and this forms our main motivation in formulating our GAN. To address (2), we propose a segmentation and refinement procedure. The segmentation stage localizes manipulated regions by learning to spot boundary artifacts. To further prevent the network from just focusing on semantic content, the refinement stage refines the predicted manipulation boundaries with authentic background and feed the new manipulated images back to the segmentation network.

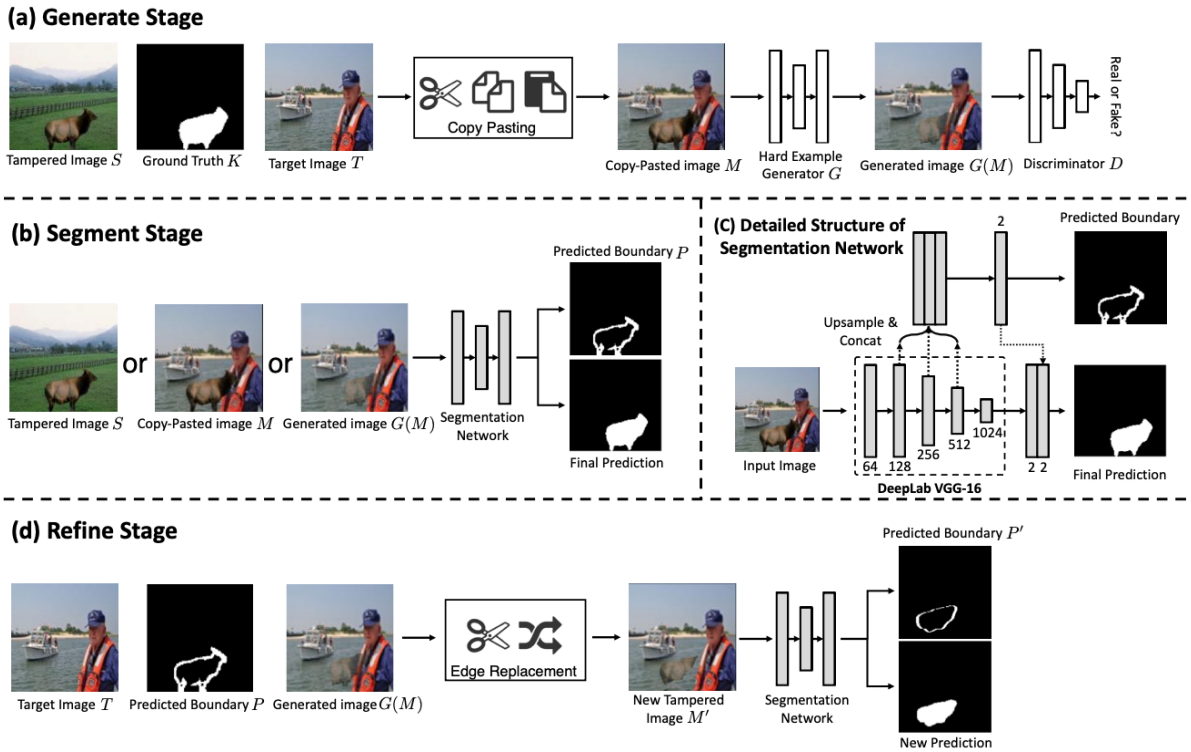


***Figure** 7. GSR-Net framework overview. (a) Generate Stage, (b) Segment Stage, (c) Detailed Structure of Segmentation Network, (d) Refine Stage.*

We design an architecture called GSR Net which includes these three components – a generation stage, a segmentation stage, and a refinement stage. The architecture of GSR Net is shown in ***Figure** 7*. Given a tampered image S, an authentic target image T, and the ground truth mask K in (a), the generation stage generates hard example G(M) starting from a simple copy-pasting image M. Feeding the training images, copy-pasted images or generated images as input, the segmentation stage (b) learns to segment the boundary artifacts and fill the interior to produce the final prediction. The segmentation network (c) concatenates lower level features to predict boundary artifacts and then concatenate back the boundary feature to the segmentation branch for final prediction. The refinement stage (d) creates a novel tampered image with new boundary artifacts by replacing the predicted manipulated boundaries of segmentation stage with original authentic regions and learns to make a new prediction. During training, we alternatively train the generation GAN, followed by the segmentation and refinement stage, which take as input the output of the generation stage as well as images from the training datasets. The additional varieties of manipulation artifacts provided by both the generation and refinement stages produce models that exhibit very good generalization ability. The three stages of training are: **Generation stage**: a GAN-based network learning to make hard example from direct copy-pasting images; **Segmentation stage**: a two-branch DeepLab segmentation network to predict both the manipulated regions and boundaries; and **Replacement stage**: an additional stage based on the prediction of segmentation stage to make the network focus more on manipulation artifacts.

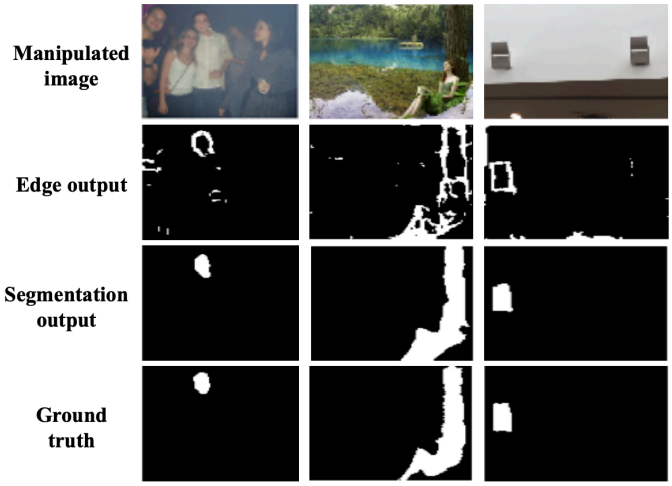
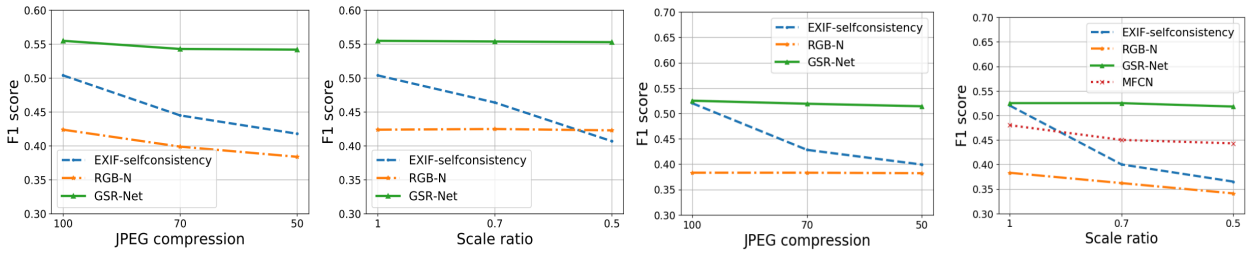


Figure 8. *Qualitative visualization.*

Figure 8 shows some qualitative results for our GSR Network where the first row shows manipulated images on different datasets. The second row indicates the final manipulation segmentation prediction. The third row illustrates the output of boundary artifacts branch, and the last row is the ground truth. We further analyze the robustness of GSR Network against JPEG compression and image scaling attacks on test images of In-The-Wild and Carvalho datasets. Figure 9 shows the results, which indicates our approach yields more stable performance than prior methods.



(a) In-The-Wild JPEG attack (b) In-The-Wild scale attack (c) Carvalho JPEG attack (d) Carvalho scale attack

Figure 9. *Analysis of robustness under different attacks. Attacks with JPEG compression consists of quality factors of 70 and 50; scale attacks use scaling ratios of 0.7 and 0.5. (a) JPEG compression attacks on In-The-Wild. (b) Scale attacks on In-The-Wild. (c) JPEG compression attacks on Carvalho. (d) Scale attacks on Carvalho.*

Please refer to [6] for details. Software provided to DARPA, container name is **umd_GSRNET** with unlimited distribution. GSRNet superceded **RGBNNET**.

3.5 Detection and Attribution of GAN-generated Media

Photorealistic image generation and manipulation techniques have rapidly evolved. Visual contents can now be easily created and edited without leaving obvious perceptual traces. Recent breakthroughs in generative adversarial networks (GANs) have further improved the quality and photorealism of generated images. The adversarial framework of GANs can also be used in conditional scenarios for image translation or manipulation in a given context, which diversifies media synthesis.

There is a widespread concern about the impact of this technology when used maliciously. This issue has also received increasing public attention, in terms of disruptive consequences to visual security, laws, politics, and society in general. Therefore, it is critical to look into effective visual forensics against threats from GANs. While recent state-of-the-art visual forensics techniques demonstrate impressive results for detecting fake visual media, they have only focused on semantic, physical, or statistical inconsistency of specific forgery scenarios, e.g., copy-move manipulations or face swapping. Forensics on GAN-generated images shows good accuracy, but each method operates on only one GAN architecture by identifying its unique artifacts and results deteriorate when the GAN architecture is changed. It is still an open question of whether GANs leave stable marks that are commonly shared by their generated images. That motivates us to investigate an effective feature representation that differentiates GAN-generated images from real ones. See Figure 10 where Inception features are highly entangled, indicating the challenge to differentiate high-quality GAN-generated images from real ones. However, our result shows any single difference in GAN architectures, training sets, or even initialization seeds can result in distinct fingerprint features for effective attribution. In this work, we demonstrate the existence, uniqueness, persistence, immunizability, and visualization of GAN fingerprints by addressing the following questions:

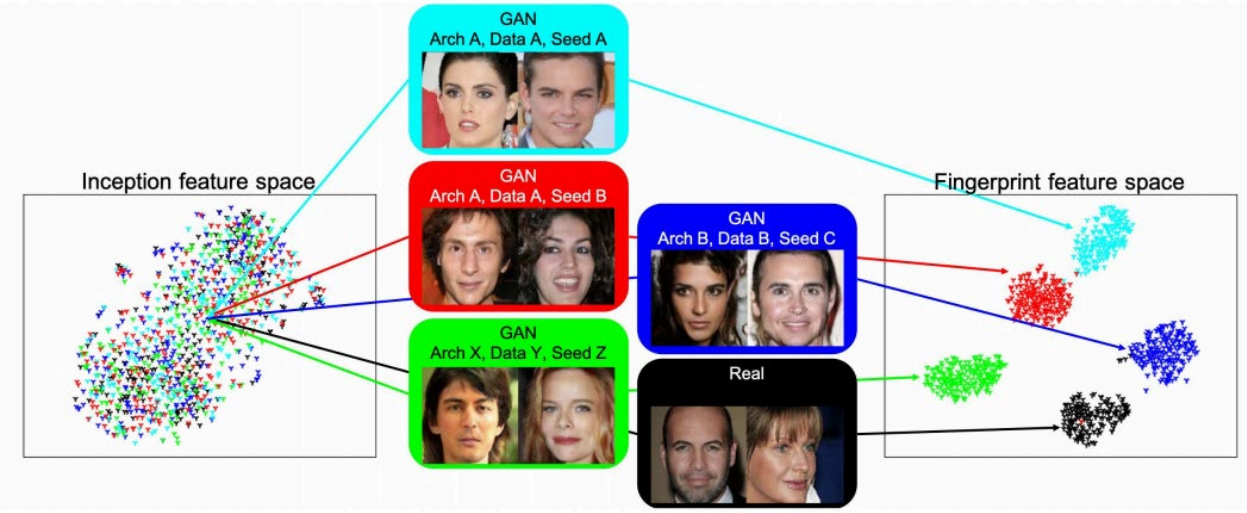


Figure 10. *A t-SNE visual comparison between our fingerprint features (right) and the baseline inception features [15] (left) for image attribution.*

Existence and uniqueness: Which GAN parameters differentiate image attribution?
We present experiments on GAN parameters including architecture, training data, as well as random initialization seed. We find that a difference in any one of these parameters results in a unique GAN fingerprint for image attribution.

Persistence: Which image components contain fingerprints for attribution?
We investigate image components in different frequency bands and in different patch sizes. In order to eliminate possible bias from GAN artifact components, we apply a perceptual similarity metric to distill an artifact-free subset for attribution evaluation. We find that GAN fingerprints are persistent across different frequencies and patch sizes, and are not dominated by artifacts.

Immunizability: How robust is attribution to image perturbation attacks and how effective are the defenses?
We investigate common attacks that aim at destroying image fingerprints. They include noise, blur, cropping, JPEG compression, relighting, and random combinations of them. We also defend against such attacks by finetuning our attribution classifier.

Visualization: How to expose GAN fingerprints?
We propose an alternative classifier variant to explicitly visualize GAN fingerprints in the image domain, so as to better interpret the effectiveness of attribution.

3.5.1 Fingerprinting Learning for Attribution

Inspired by the prior works on digital fingerprints, we introduce the concepts of GAN model fingerprint and image fingerprint. Both are simultaneously learned from an image attribution task.

Model fingerprint. Each GAN model is characterized by many parameters: training dataset distribution, network architecture, loss design, optimization strategy, and hyperparameter settings. Because of the non-convexity of the objective function and the instability of adversarial equilibrium between the generator and discriminator in GANs, the values of model weights are sensitive to their random initializations and do not converge to the same values during each training. This indicates that even though two well-trained GAN models may perform equivalently, they generate high-quality images differently. This suggests the existence and uniqueness of GAN fingerprints. We define the model fingerprint per GAN instance as a reference vector, such that it consistently interacts with all its generated images.

Image fingerprint. GAN-generated images are the outcomes of a large number of fixed filtering and non-linear processes, which generate common and stable patterns within the same GAN instances but are distinct across different GAN instances. That suggests the existence of image fingerprints and attributability towards their GAN sources. We introduce the fingerprint per image as a feature vector encoded from that image.

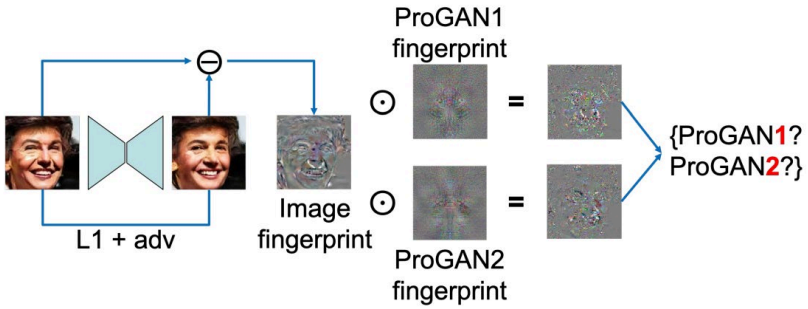


Figure 11. *Fingerprint visualization diagram.*

Fingerprint visualization. We describe a model to explicitly represent them in the image domain. But in contrast to their hand-crafted PRNU-based representation, we modify our attribution network architecture and learn fingerprint images from image-source pairs. We also decouple the

representation of model fingerprints from image fingerprints. ***Figure 11*** depicts the fingerprint visualization model. We train an AutoEncoder and GAN fingerprints end-to-end.

3.5.2 Attribution Network

We formulate image attribution and GANs identification as a classification problem, where a classifier in a neural network architecture is trained, the source of an image is predicted. The source can be 'real image' or a GAN model belonging to a finite set which is composed of pre-trained GAN instances. ***Figure 12***(a) depicts an overview of our attribution network. Tensor representation is specified by two spatial dimensions followed by the number of channels. The network is trained to minimize cross-entropy classification loss. The pre-downsampling network example in (b) down samples input image to 8 × 8 before convolution. The pre-downsampling residual network example in (c) extracts the residual component between 16×16 and 8×8 resolutions. The post-pooling network example in (d) starts average pooling at 64 × 64 resolution.

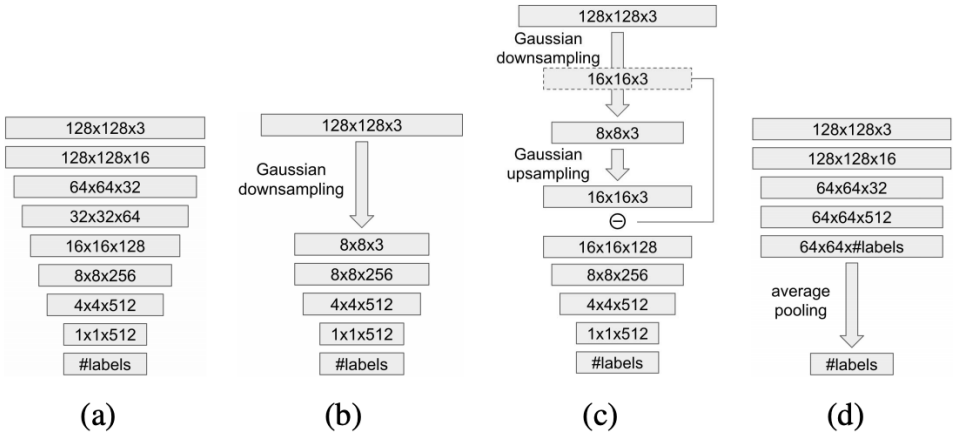


Figure 12. *Different attribution network architectures. (a) Attribution network, (b) Pre-downsampling network example, (c) Pre-downsampling residual network example, (d) Post-pooling network example.*

Component analysis networks.
In order to analyze which image components contain fingerprints, we propose three variants of the attribution network:
Pre-downsampling network. We propose to test whether fingerprints and attribution can be derived from different frequency bands. We investigate attribution performance w.r.t. downsampling factor. ***Figure 12***(b) shows an architecture example that extracts low-frequency bands.
Pre-downsampling residual network. Complementary to extracting low-frequency bands, ***Figure 12***(c) shows an architecture example that extracts a residual high-frequency band between one resolution and its factor-2 downsampled resolution.
Post-pooling network. We propose to test whether fingerprints and attribution can be derived locally based on patch statistics. We investigate attribution performance w.r.t. patch size. ***Figure 12***(d) shows an architecture example.

Results. Our experiments show that even a small difference in GAN training (e.g., the difference in initialization) can leave a distinct fingerprint that commonly exists over all its generated images. That enables fine-grained image attribution and model attribution. Further encouragingly, fingerprints are persistent across different frequencies and different patch sizes, and are not biased

by GAN artifacts. Even though fingerprints can be deteriorated by several image perturbation attacks, they are effectively immunizable by simple finetuning. An example of our model and image fingerprint visualization in shown in ***Figure 13***, where the pairwise interactions are shown as the confusion matrix.

Software provided to DARPA, container name is umd_ganfingerprint with unlimited distribution. Please refer to [7] for more details.

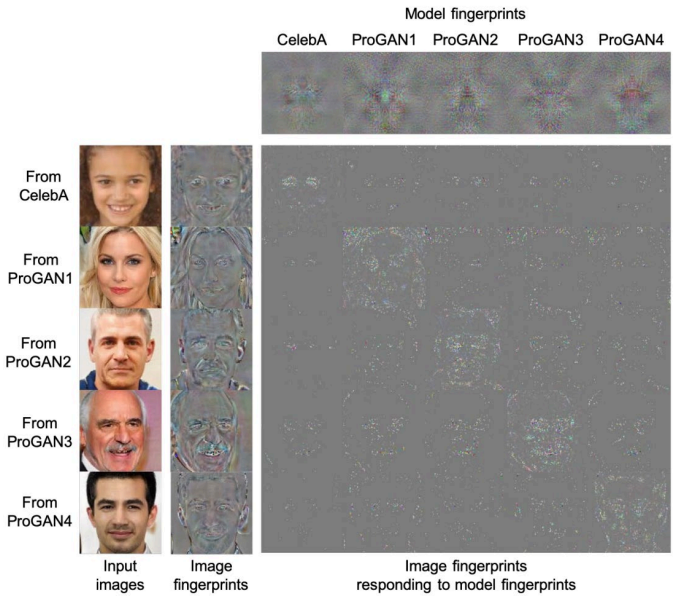


Figure 13. *Visualization of model and image fingerprint samples.*

Extensions. We further extend this work to include a more diverse setup of samples while also employing simpler network architectures. Our dataset consists of 12 classes with a specific train test split. Four of the classes correspond to images from real datasets, namely, CelebA, CelebA-HQ, ImageNet and LSUN-Bedroom. The rest of the eight classes consist of images generated from different GANs trained on one of the real datasets. Unlike our earlier work, where we train a classifier on only four GANs and one real world dataset, namely CelebA, we chose this setup so that we include a diverse set of images generated from different GANs or sampled from multiple real-world datasets. On testing our algorithm on the test set, we get attribution accuracies of nearly ~99%. Extending our dataset from 12 classes to 25 with an increased number of real datasets and GAN sources maintains our attribution accuracy at ~99%. This shows that attribution of real and GAN generated images to their actual classes is a relatively straightforward task in the supervised scenario.

3.6 Improving GAN-generated Media

Image/video synthesis and fake content detection are two related problems. With the rapid advances in image and video synthesis and manipulation techniques, it has become crucial to train detectors that can distinguish between real and fake content. However, training robust detectors can make use of photo-realistic synthetic data as negative examples during training. Therefore, we need controllable ways of image and video synthesis and manipulation. Two such approaches we study are discussed below.

3.6.1 Style-based encoder pre-training for multi-modal image synthesis

In this work we propose a novel style-based pre-training strategy for multi-modal I2I translation. We first pre-train an encoder, using a proxy task, to encode the style of an image, such as color and texture, into a low-dimensional latent style vector. We then train a generator to transform an input image along with a style-code to the output domain. Our generator achieves state-of-the-art results on several benchmarks with a training objective that includes just a GAN loss and a reconstruction loss, which simplifies and speeds up the training significantly compared to competing approaches. We also show that the learned style embedding is not dependent on the target domain and generalizes well to other domains. Since the proposed style-based pre-training strategy results in a latent space that clusters similar styles. This can be used to cluster together images generated by the same GAN or DeepFake technique. This can be achieved by replacing the style-based triplet sampling by method-based triplet sampling of images generated by different generative techniques.

Advantages of the proposed style-based encoder pre-training:

- It learns a more powerful and expressive latent space. Specifically, we show that:
    1. Our pre-trained latent space captures uncommon styles that are not well represented in the training set, while BicycleGAN-based baselines fail to do so and instead tend to simplify such styles/appearances to the nearest common style in the train set.
    2. Pre-training yields more faithful style capture and transfer.
    3. Finally, the better expressiveness of the pre-trained latent space leads to more complex style interpolations compared to BicycleGAN-based baselines.
- The learned style embedding is not dependent on the target dataset, and generalizes well across multiple domains, which is especially useful for the case of having limited training data.
- Style pre-training simplifies the training objective by requiring fewer losses, which also speeds up the training.
- Our approach improves the training stability and the overall output quality and diversity.

Key results of our approach are:

- Style transfer and sampling: ***Figure 14*** shows style transfer to images from the validation set of different datasets. For each dataset, we show output for applying different styles to the same input image. Note how the style transfer copies the weather conditions in the Space Needle and Night2day datasets. ***Figure 15*** shows the results of sampling random styles using two methods: (a) by enforcing a weak zero-mean prior on the latent style codes to enable sampling from a normal distribution with zero mean and an empirically computed standard deviation. (b) sampling using a proposed mapper network that maps the unit gaussian to the latent style distribution.
- Style interpolation: ***Figure 16*** shows style interpolation by linearly interpolating between two latent vectors. Our method shows complex non-linear interpolation of appearance; for example, note the smooth change in lighting and cloud patterns when going from cloudy to sunny in the Space Needle dataset (second row).
- Visualizing the pre-trained latent space: ***Figure 17*** visualizes the pre-trained latent space learned by the style encoder E. The visualization shows meaningful clusters of similar weather conditions for the Space Needle timelapse dataset.

Figure 14. *Style transfer for different datasets.*

Figure 15. *Style sampling results: for each input image in the left column, we generate the output under different randomly sampled styles.*

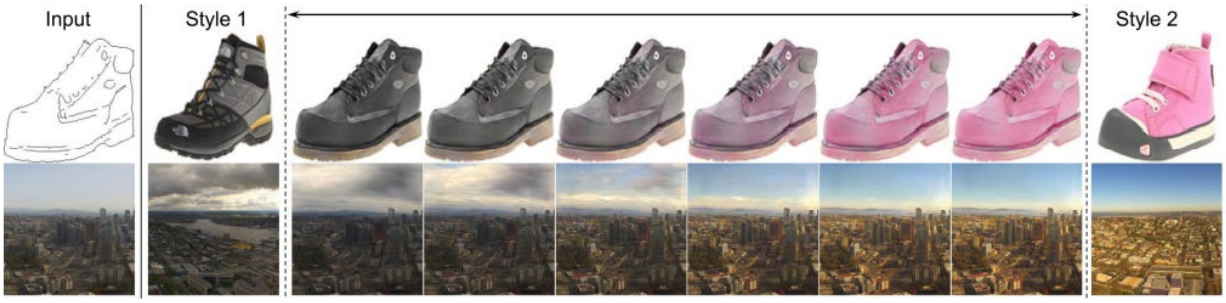


Figure 16. *Style interpolation between two images using linear interpolation in the latent space.*

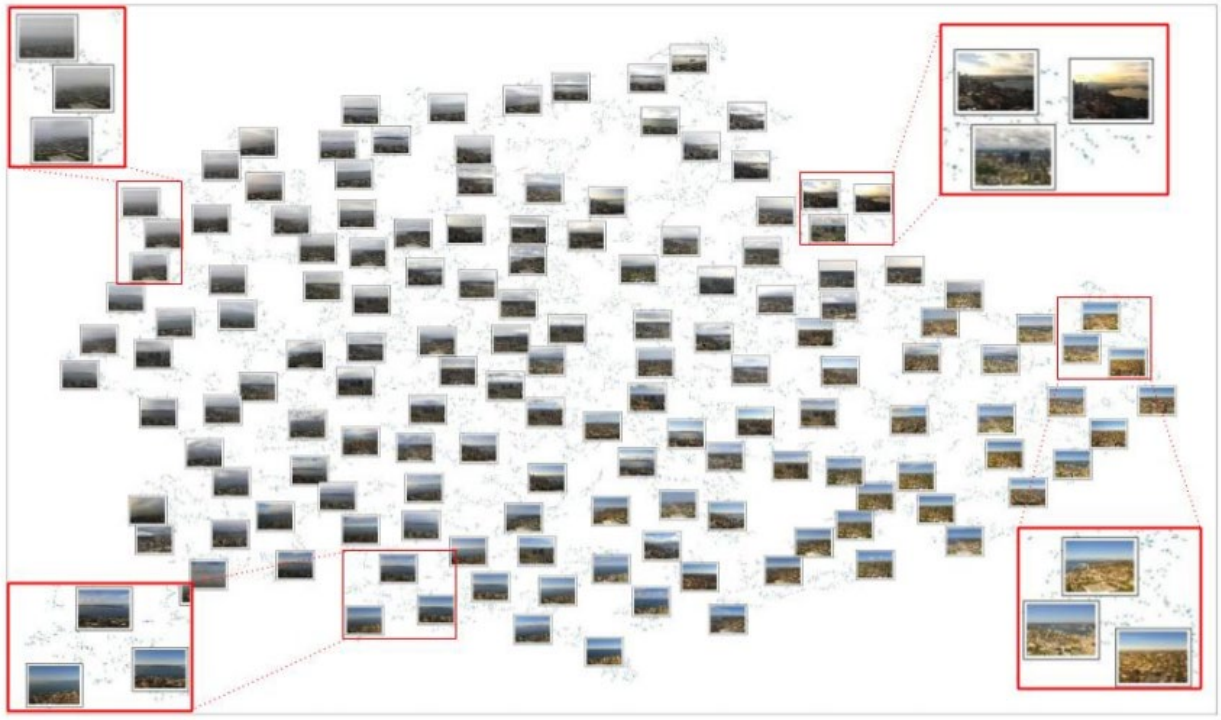


Figure 17*. t-SNE plot of a pre-trained style latent space.*

3.6.2 Two-step face synthesis using learned spatial maps

We explore realistic few-shot synthesis of faces, which can be utilized to training DeepFake detection systems. Given target facial landmarks, we train a network to predict a latent spatial map for the target image. We then synthesize the final image conditioned on the predicted spatial input. One form of the intermediate spatial representation is to predict semantic labels, which allows further control over the synthesis by editing the semantic map. On the other hand, we show that the network can learn a latent spatial representation that is better suited for the synthesis process, which leads to better quality and realism of the synthesized image. We explored training a discriminator network to observe the domain gap between predicted spatial maps for real and fake data. One limitation to training a discriminator directly on image space is that the discriminator focuses on network fingerprints that provide an easy signal to discover fake data generated by one specific technique, but causes the discriminator not to generalize to other deep fake techniques. On the other hand, training the discriminator on a shared latent representation between real and fake data can lead to better generalization across different image synthesis and deep fake techniques. ***Figure 18*** shows preliminary results for the proposed approach.

Input frames Target landmarks Learned spatial map GT semantic map Output synthesis Ground truth

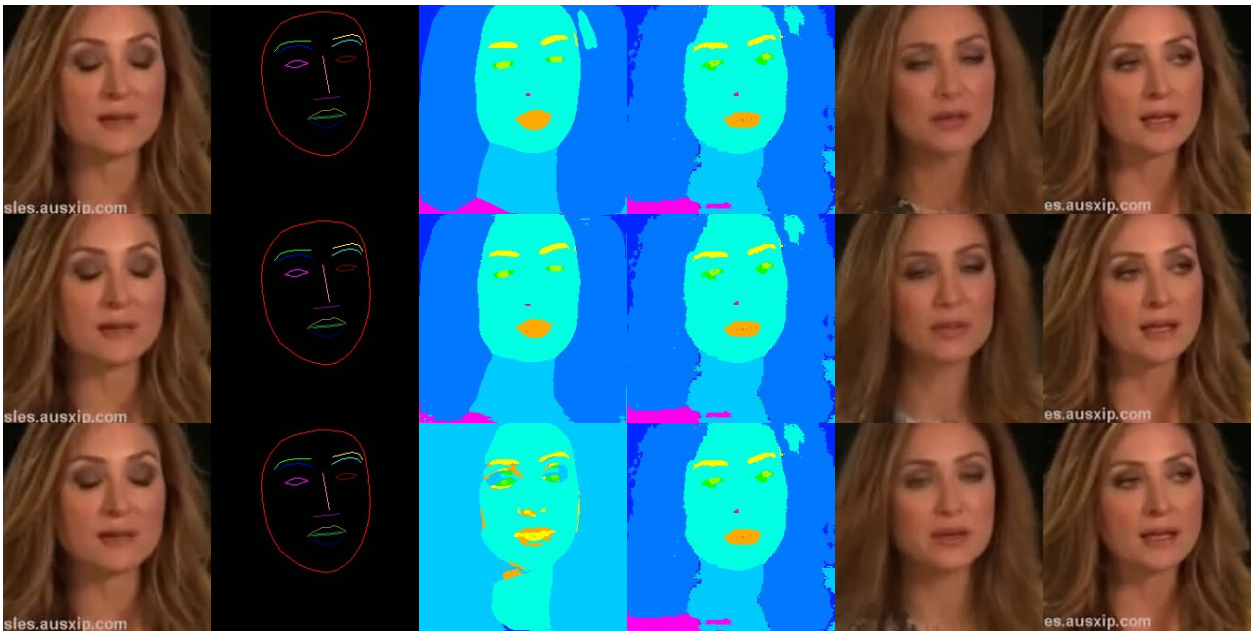


***Figure 18**. Example synthesis for 3 variations of our method. Top: using a fixed pre-trained semantic map prediction network. Middle: fine-tuning the semantic map prediction network jointly with the generator. Bottom: predicting a latent spatial map that is not supervised to correspond to semantic labels.*

3.7 DeepFake Video Detection

DeepFake videos emerged as a target for forensic media analysis during the performance period and was not part of the initial goals. Initial scarcity of data until the summer of 2019 has been replaced by medium and large sized datasets that support research and evaluation of detection algorithms.

The research focused on temporal cues that may be present in DeepFake videos. The primary conjecture is that maintaining temporal coherence and physical realism is relatively harder for automated as well as human tampering agents. Physical realism is the domain of Hollywood, at least for now. At the same time, we acknowledge that there are types of falsifications that do not alter the video temporally (e.g., changing the iris color of the face, whole face splicing of face videos, etc.). Therefore, in these cases temporal information will not have any discrimination power.

The features we compute are spatiotemporal measures from the eyes, mouth, and facial action units AUs. Specifically, the average speed and acceleration as well as variance in feature opening in a sequence. For the 9 AUs, we computed average speed and accelerations, resulting in 18 features. Three features were computed for the mouth, average speed and accelerations of lip movements, and variance of mouth spatial opening over the length of the video (normalized by the size of the mouth in the image). Similarly, 6 features were computed for the right and left eyes (similar features for the mouth). Finally, 4 covariance measures were computed between the eyes (all averaged across the video), namely covariance of frame to frame motions, Euclidean distances of the motion between the two eyes, and the speed and acceleration (i.e., first and second order differences). In addition, blinking patterns was analyzed and performance evaluated with respect

to the problem. Each video-clip is represented by a 35-dimensional feature vector that is classified with respect to authenticity. The blinking features add 36 additional features, for a total of 71 dimensions.

The diversity of sources creates a realistic challenge for developing algorithms. Specifically, we used the following datasets FaceForensics++ (with video data generated by DeepFake, Face2Face, FaceSwap and Neurotextures), Google-DF, Celeb-DF-v2 and Facebook's challenge dataset. The generation/tampering algorithms are diverse in objectives, quality, levels of automation, viewpoints, number of people, etc. This diversity creates significant challenge for developing general algorithms. It is also challenging to evaluate detection performance due to the multitude of variables involved in generation. For most of these datasets it is unknown how generation was done, and given the size of the data (near 150K videos) it is impossible to view and inspect more than a few video clips. It is important to note that some of the algorithms for tampering change only a very small part of the face, others splice and deform whole faces, and yet others seem to involve blurring and illumination tampering.

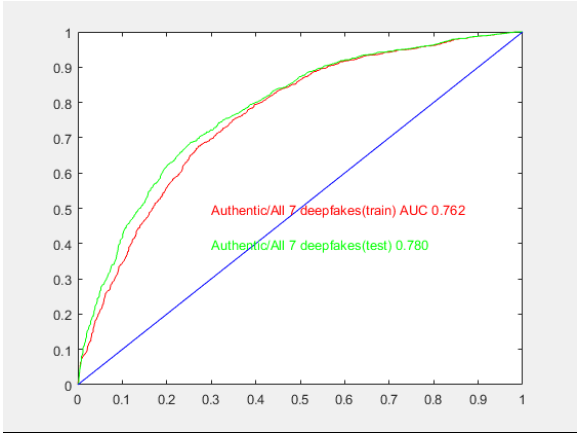


Figure 19. *ROC for the 7 model classification of DeepFakes.*

Overall, we processed nearly 150,000 video clips that cover a large universe of DeepFake generators. This provided with statistical insights and a level of confidence of the performance. ***Figure 19*** shows blended modeling from seven large datasets. The ROC reflects balanced modeling from 7 DeepFake datasets (4 from FaceForensics++, Celeb-DF-v2, Facebook and Google-DF). We show the ROC for the trained data and the test data. The results show consistency between training and testing.

Software provided to DARPA, container name is **umd_openface-base** with unlimited distribution

3.8 Video Inpainting Detection

Video inpainting, which completes corrupted or missing regions in a video sequence, has achieved impressive progress over the years. The ability to produce realistic videos that can be used in applications like video restoration, virtual reality, etc., while appealing, brings significant security concerns at the same time since these techniques can also be used maliciously. By removing objects that could serve as evidence, malicious inpainting can result in serious legal and social implications including swaying a jury, accelerating the spread of misinformation on social

platforms, etc. Our goal in this work is to develop a framework for detecting inpainted videos constructed with state-of-the-art methods (see ***Figure 20*** for a conceptual overview).

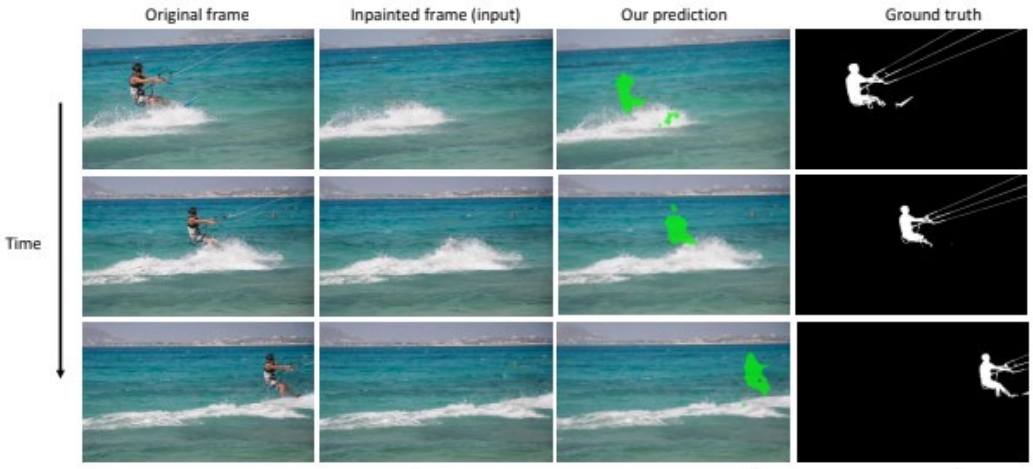


Figure 20. *Given an inpainted video (second column), we localize the inpainted region, both spatially and temporally.*

To address this, we introduce VIDNet, a video inpainting detection network, which is an encoder-decoder architecture with a quad-directional local attention module to predict inpainted regions in videos. In particular, at each time step, VIDNet takes as inputs the current RGB frame together with its corresponding Error Level Analysis (ELA) frame to the encoder, truncated from a pretrained VGG network. Since video are compressed based on discrete cosine transforms (DCT) and frames extracted are usually stored in JPEG formats, we leverage ELA images as an additional signal to reveal artifacts like compression inconsistency. Instead of using ELA images directly, which tends to produce false alarms, we extract features from both ELA and RGB images with the encoder, producing five different multimodal features at different scales, that are further jointly trained for inpainting detection. In addition, given a missing region to fill in, inpainting methods leverage information from surrounding pixels of the region to make the region coherent spatially. Motivated by this, for RGB features from the last layer of the encoder, we introduce a quad-directional local attention module to attend to the neighbors of a pixel to detect whether that pixel is inpainted or not. This allows us to explicitly model spatial dependencies among different pixels to identify inpainted pixels.

Finally, with multimodal features encoded at different scales, we leverage a four-layer Convolutional LSTM, serving as a decoder for inpainting detection. More specifically, the ConvLSTM at a certain layer not only takes in features from a previous time step but also features upsampled from a coarse level (i.e., a lower decoding layer). In this way both spatial relationships across different scales and temporal dynamics over time are leveraged to produce inpainted masks over time. The framework is trained end-to-end with backpropagation. ***Figure 21*** shows an overview of our approach.

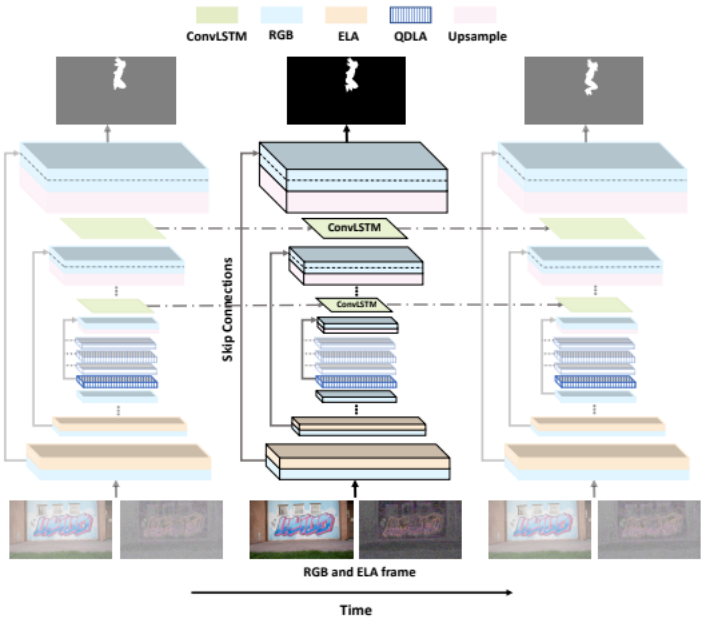


***Figure 21**. VIDNet Framework Overview.*

Since we target at a relatively new task, to the best of our knowledge, we introduce the first learning based approach for video inpainting detection, and a new benchmark for evaluating the generalization of such approaches. We evaluated our approach on DAVIS 2016 for inpainting detection, which consists of 30 videos for training and 20 videos for testing. We generated inpainted videos using SOTA video inpainting approaches: VI [8], OP [9] and CP [10], with the ground truth object mask as reference. To show both the performance and generalization, we choose two out of the three inpainted DAVIS for training and testing, leaving one for additional testing. The training/testing split follows DAVIS default setting. VIDNet successfully detects inpainted regions under all different settings and outperforms by clear margins competing methods. We also show that VIDNet can be generalized to detect out-of-domain inpainted videos that are unseen during training. We compare our approach with other methods for detecting video inpainting using Mean IoU and F1 score as the metrics. Example quantitative results are shown in ***Table 1*** and qualitative results are shown in ***Figure 22***.

***Table 1.** mean IoU and F1 score comparison on inpainted DAVIS. `*' denotes that the model is trained on these inpainting algorithms.*

	VI		**OP***		**CP***	
Methods	IoU	F1	IoU	F1	IoU	F1
NOI	0.082	0.137	0.090	0.137	0.072	0.132
CFA	0.103	0.142	0.083	0.137	0.076	0.121
HPF	0.342	0.444	0.409	0.510	0.676	0.773
GSR-Net	0.302	0.426	0.736	0.818	0.801	0.849
Ours RGB (baseline)	0.308	0.417	0.705	0.773	0.777	0.859
VIDNet-BN (ours)	0.301	0.415	**0.801**	**0.860**	**0.837**	**0.915**
VIDNet-IN (ours)	**0.386**	**0.493**	0.740	0.820	0.810	0.869

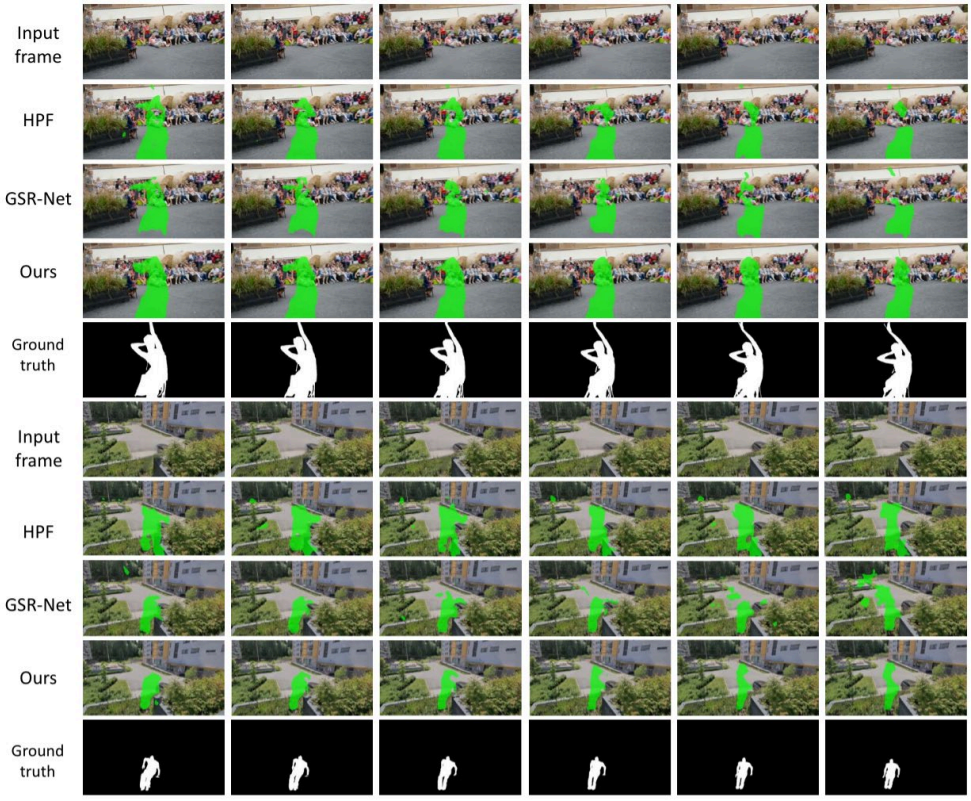


Figure 22*. Qualitative visualization on DAVIS. The first row shows the inpainted video frame. The second to fourth row indicates the final predictions from different methods. The fifth row is the ground truth.*

3.9 Adversarial and Compression Robustness

Many forensics software to detect multimedia tampering and computer-generated media rely heavily on low-level high-frequency details, which are susceptible to several attacks. Common attacks include cheap attacks (e.g., compression, blur) and expensive adversarial attacks. In this series of works, we focus on developing robustness to such attacks.

3.9.1 JPEG Artifact Correction & Robustness

Standard compression approaches, e.g., JPEG, introduce artifacts and alter such information during the compression process. Our goal is to develop approaches that can correct artifacts caused by JPEG compression and employ with task-specific networks (e.g., GAN attribution) with such artifact correction approaches. We developed a state-of-the-art method for correcting artifacts caused by JPEG compression. Our method solves several existing problems in prior works:

1. Prior works train a single model for each JPEG quality. This requires training and deploying many models, which is expensive. Moreover, the JPEG quality is not stored with the image, so a real system has no way to pick the correct model
2. Prior works generally deal with grayscale images, our method includes novel handling of the color channels which are more heavily compressed.

3. Prior works focus on regression which provides good metric results, but are often qualitatively blurry, we add a texture restoring GAN loss which help sharpen the image and restore textures to flat regions

Our single network is able to operate on different quality levels by using the JPEG quantization matrix as a side-channel input. The quantization matrix is computed from the quality level and stored in the JPEG file. This matrix determines the amount of rounding applied to the DCT coefficients, and therefore the amount of quality loss. Our method restored color images by using the grayscale restoration to guide the color restoration. Since the grayscale channel is stored at a much higher quality than the color channels, the color channels are often missing fine details and structures that can be restored well in the grayscale channel. By giving the color network access to the grayscale restoration, we show a significant improvement in color channel correction. Finally, we fine tune our model with a GAN loss that is designed to restore texture by replacing the traditional perceptual loss, which is often a VGG network trained on imagenet, with a VGG network trained on the MINC material classification dataset. By including this loss, the goal is to make the restoration as close as possible to the original images in material appearance, e.g. the material classification network should classify them as the same material. We find that this manifests as texture restoration.

Qualitative results are presented in ***Table 2*** and qualitative results for restoration are presented in ***Figure 23***. Please refer to [11] for details.

Table 2. *Our work achieves state-of-the-art performance on color image restoration. The format of the results is PSNR/PSNR-B/SSIM.*

Dataset	Quality	JPEG	ARCNN [8]	MWCNN [27]	IDCN [51]	DMCNN [49]	Ours
Live-1	10	25.60 / 23.53 / 0.755	26.66 / 26.54 / 0.792	27.21 / 27.02 / 0.805	27.62 / 27.32 / 0.816	27.18 / 27.03 / 0.810	**27.65 / 27.40 / 0.819**
	20	27.96 / 25.77 / 0.837	28.97 / 28.65 / 0.860	29.54 / 29.23 / 0.873	**30.01** / 29.49 / 0.881	29.45 / 29.08 / 0.874	29.92 / **29.51 / 0.882**
	30	29.25 / 27.10 / 0.872	30.29 / 29.97 / 0.891	30.82 / 30.45 / 0.901	-	-	**31.21 / 30.71 / 0.908**
BSDS500	10	25.72 / 23.44 / 0.748	26.83 / 26.65 / 0.783	27.18 / 26.93 / 0.794	27.61 / 27.22 / 0.805	27.16 / 26.95 / 0.799	**27.69 / 27.36 / 0.810**
	20	28.01 / 25.57 / 0.833	29.00 / 28.53 / 0.853	29.45 / 28.96 / 0.866	**29.90** / 29.20 / 0.873	29.35 / 28.84 / 0.866	29.89 / **29.29 / 0.876**
	30	29.31 / 26.85 / 0.869	30.31 / 29.85 / 0.887	30.71 / 30.09 / 0.895	-	-	**31.15 / 30.37 / 0.903**
ICB	10	29.31 / 28.07 / 0.749	30.06 / 30.38 / 0.744	30.76 / 31.21 / 0.779	31.71 / 32.02 / 0.809	30.85 / 31.31 / 0.796	**32.11 / 32.47 / 0.815**
	20	31.84 / 30.63 / 0.804	32.24 / 32.53 / 0.778	32.79 / 33.32 / 0.812	33.99 / 34.37 / 0.838	32.77 / 33.26 / 0.830	**34.23 / 34.67 / 0.845**
	30	33.02 / 31.87 / 0.830	33.31 / 33.72 / 0.807	34.11 / 34.69 / 0.845	-	-	**35.20 / 35.67 / 0.860**

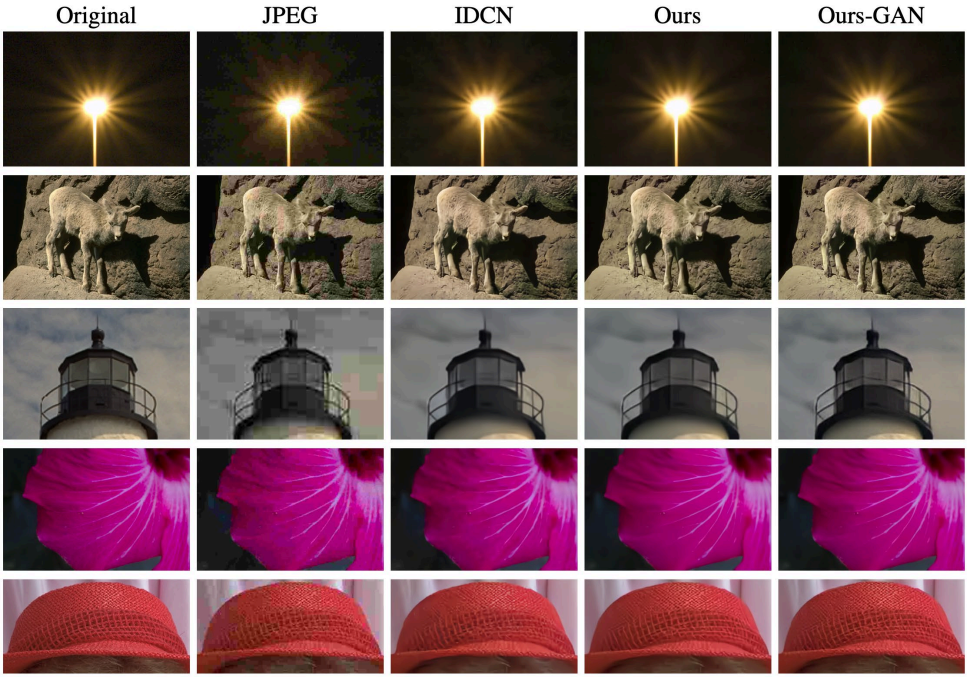


Figure 23. *Qualitative Results. All images were compressed at Quality 10.*

Task Guided Correction for Media Forensics

We extend the artifact correction method discussed in the previous section to aid media forensics. Heavy or repeated compression can cause significant accuracy penalties for certain forensic tasks. For example, GAN detection/attribution is known to suffer with even moderate compression, and we find that most tasks struggle for heavy compression. In these cases, fine tuning the network with JPEG-as-data-augmentation is hit or miss. Sometimes, it helps even on uncompressed images and sometimes it does not help at all.

Our proposed method uses gradient from the mistakes that the task network makes on compressed images to fine-tune the artifact correction network. This way, the correction network is guided towards a reconstruction that prioritizes performance on the downstream task. Note that this is an entirely self-supervised process: the fine-tuning only needs access to the difference in the predicted labels from JPEG images and uncompressed images.

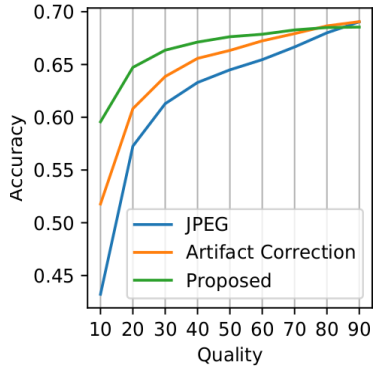


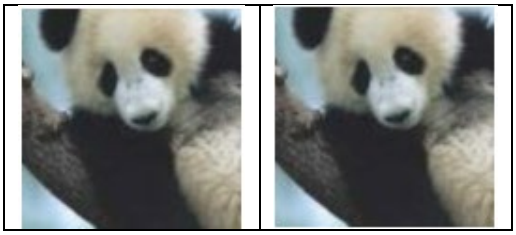
Figure 24. *Task-guided artifact correction.*

We find that this is more stable and provides benefits across-the-board due to the artifact correction networks targeted design for JPEG. In ***Figure 24***, we show the performance of our proposed method in green compared to standard artifact correction in orange (the method from the previous section without modification) and using the JPEG images directly in blue on a pretrained ResNet 50 for ImageNet classification.

3.9.2 Adversarial Robustness

Adversarial examples are an easy way to bypass forensics software. Making a model robust to adversarial examples is computationally expensive. Our goal is to develop a scalable approach for robustness to adversarial examples.

The performance of the image classifiers has been greatly increased during the recent years, thanks to the power of the deep neural networks in describing visual data. The top1 accuracy on the ImageNet dataset has been noticeably improved from 50% in 2011 to more than 88% in 2020. However, these deep learning models are not free of problems especially when it comes to their deployment in high-stake applications. As shown in recent years, it is possible to fool image classifiers by adding a small inconceivable "adversarial perturbation". Figure 25***Figure 25*** shows such an example.



Class: Panda Class: Gibbon

Figure 25*. An adversarial example [12].*

In this work, we consider two types of attacks (i.e., per-image adversarial attacks and universal adversarial attacks) and propose efficient defense mechanisms to overcome them.

Efficient Adversarial Training against Per-Image Attacks

As mentioned previously, it is possible to generate inconceivable per-image noises to fool even the strongest classification networks. Considering the existence of such adversarial attacks, making the classification models robust is critical. Adversarial training is among the most effective approaches towards defending against adversaries. However, conventional adversarial training makes training around 7 times slower and intractable on large datasets. In our recent work [13], we study this problem and introduce an efficient adversarial training approach with almost no additional cost over conventional training of neural networks, while achieving the same robustness as regular adversarial training. Developing efficient methods to make image classifiers robust is an important research direction that is necessary for the deployment of classifiers trained to distinguish real from fake images.

Universal Adversarial Training

In classic per-image adversarial attacks, given an image, the goal is to find an imperceptible perturbation that causes the classifier to misclassify that *specific image*. In contrast, universal

attacks aim to find a *unique* imperceptible perturbation which when added to all samples in the test set, causes a noticeable number of them to be misclassified. Previous works on universal adversarial generation rely on an iterative approach which is computationally expensive. This makes *adversarial training* on such generated noises intractable. The goal of our study is to develop efficient methods towards universal adversarial generation which make universal adversarial training possible.

Universal Adversarial Attack: Previous works for generating universal adversarial perturbation was based on an iterative method called DeepFool. This iterative approach noticeably slows down the process of generating universal adversarial noise. In contrast, we propose to generate universal perturbations directly based on the classification loss. That is, we find a perturbation common to all training samples, which when added to the images maximizes the classification loss. The main problem with such an approach is that the classification loss is unbounded from above. As a result, it is possible to maximize the classification loss, by pushing the loss for a single sample to infinity. In [14], we discuss such problems and propose workarounds for dealing with them. As shown in the results section, our approach is both more accurate and noticeably faster.

Universal Adversarial Training: Having a more efficient algorithm for generating universal adversarial perturbations, we study the possibility of universal adversarial training for the first time. Adversarial training works by generating perturbations at each training step and training the network on them. Although our proposed algorithm is considerably faster, generating a universal adversarial noise per each training iteration is not tractable. However, we show it is not necessary to create a perturbation for each step from scratch. Instead, we keep updating the existing adversarial noise during the training. To do so, alternating gradient descent is used to update the model weights and the universal adversarial noise alternatively.

Results: ***Figure 26*** shows the universal perturbations generated for different network architectures. Interestingly, the discovered universal perturbations have high-frequency patterns depending on the network architectures.

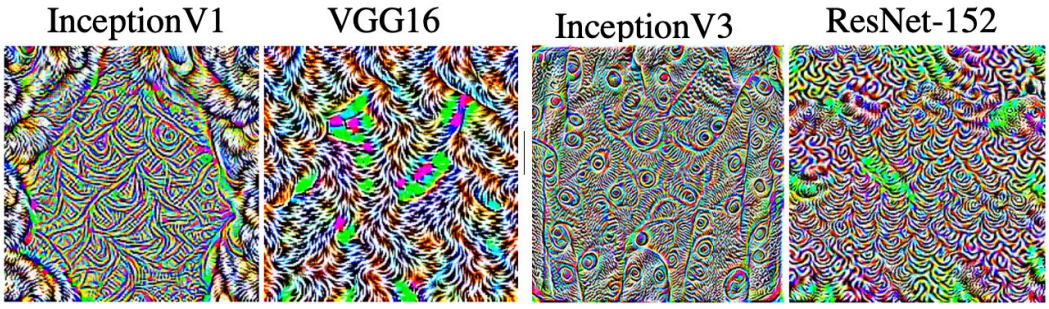


Figure 26*. Universal Perturbations created for different network architectures.*

The accuracy on the CIFAR-10 dataset for the white-box setting is shown in ***Table 3***. In this setting, it is assumed that the attacker knows the model and its weights. Each row shows a model trained with a different training strategy. Columns represent different types of attacks. Our universally trained model is the most robust model against universal attacks. Although unlike other defenses in this table, we did not train the network for defending against per-example attacks, it is considerably robust against such attacks as well.

Table 3. *Accuracy on CIFAR-10 (white-box setting)*

		Attack method			
		UnivPert	FGSM	R-FGSM	PGD
(Robust) models trained with	Natural	9.2%	13.3%	7.3%	0.0%
	FGSM	51.0%	95.2%	90.2%	0.0%
	R-FGSM	57.0%	97.5%	96.1%	0.0%
	PGD	86.1%	56.2%	67.2%	45.8%
	Ours	**91.8%**	37.3%	48.6%	17.2%

Table 3 shows the results for the black-box setting. In this case, a PGD attack is generated based on the models in each column and then applied to the models listed in the rows.

Table 4. *Accuracy on CIFAR-10 (black-box setting)*

	Attack source				
	Natural	FGSM	RFGSM	PGD	Ours
Natural	-	34.1%	64.9%	77.4%	22.0%
FGSM	53.9%	-	14.1%	69.6%	22.7%
RFGSM	71.5%	16.0%	-	71.7%	20.3%
PGD	84.1%	86.3%	86.3%	-	76.3%
Ours	90.0%	90.8%	91.0%	70.4%	-
Average	74.9%	56.8%	64.1%	72.3%	**35.4%**

As can be seen, our universally trained model is robust against attacks generated based on other models. It should be also noted that the attack generated based on our model is transferred very well to other models, showing the importance of further studying universal perturbations.

Please refer to [14] [13] for details.

4.0 Extension ResearchReport

This section describes two focused areas of the extended research that was related to this contract**.**

4.1 Satellite Image Forensics

The research effort covered a wide range of topics related to analysis of satellite images to enable forensic capabilities for analysis of images. Generation and detection objectives were studied for relevant problems. Multiple sources of data were used, IARPA-CORE3D, GRSS, high resolution aerial images and other public data. As a result, both RGB and multi-spectral data were used and variety of algorithms were developed and used to explore the technical problems.
The research effort consisted of the following components

1. Image tampering detection using signal level analysis
2. Temporal analysis of image series
3. Image tampering detection using semantic data
4. Inpainting generation and detection
5. Software transitioned to NGA

4.1.1 Satellite Image Tampering Detection using Deep Learning

UMD has developed an image tampering detector for the MediFor effort (GSRNET [6]). This approach was geared towards detecting splicing in natural images. Nevertheless, the approach could be used for simple satellite data splicing (e.g., Sentinel-2-forged), where it performed at AUC of 0.93. However, we recognized that the sophistication of this forged data is below the standard of open-world problems.

We extended this approach to Satellite images by creating new training and testing datasets of spliced buildings. The objective was to cope with semantically meaningful tampering. We relied on high resolution aerial image datasets (3+ inch ground resolution) to enable realistic high quality image tampering. The dataset included 90 spliced instances for Washington DC buildings, 43 from Gaithersburg and 93 from Baltimore. The ground resolution was 3-6 inches. Figure 27 shows an example from the GSRNET detection when it was trained and tested on this dataset. The pixel level detection AUC varied among Areas of Interest, between 58% for Baltimore and 85% for Gaithersburg. This experiment suggested that a semantic analysis may be needed to cope with the complexity of the task. This semantic analysis will employ background information as well as multiple raw data sources.

In a different experiment we repeated the experiments on AOIs from Vienna Austria (61 tampered images), Belize (46 tampered images) and San Fernando, Argentina (53 tampered images). The performance went down, ranging from random to 0.71 on per-pixel detection. As a result, this approach was then supplemented by a GIS based approach using OpenStreetMap to reduce false alarm and constrain the search.

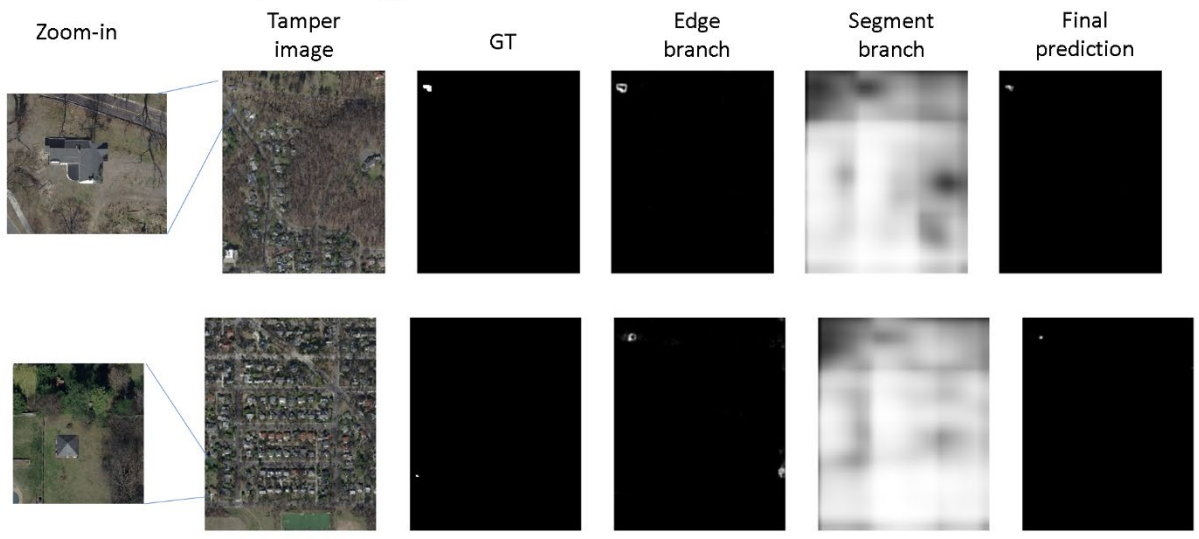


Figure 27. *Example of spliced buildings in large tiles, and the respective detection stages.*

4.1.2 Satellite Temporal Tampering Detection

Image dating and forgery of dated satellite imagery are challenging in the absence of metadata. At the same time it is well known that dis-information can employ miss-dating of imagery as a powerful tool. We developed algorithms for dating a satellite image given a sequences of ground trothed images. In one experiment we used 4306 different images of 50 areas in Omaha MSI dataset to train and evaluate a Deep Neural Network for determining the month a given image is taken. Obviously this problem is applicable to geographic locations with mild seasonal variations. Figure 28 shows the confusion matrix of month predications. Overall the trained network performed at 92% accuracy in this experiment.

In a different experiment, we assessed if a network can score a sequence of images as a correct sequence or not. It implicitly answers the question of whether a probe image in the middle of the sequence conveys the same features observed in the rest of the sequence. This prediction task performance was 83% for this experiment. It is important to note that the networks relied on models of general images (e.g. ResNet pretrained on ImageNet), instead of satellite-focused representations.

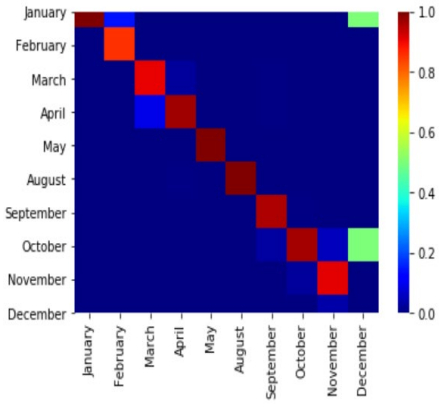


Figure 28. *Confusion Matrix for months predicted for Omaha*

4.1.3 Detecting change in Satellite Images using Semantics

Detecting change in satellite images is a critical capability due to the massive data that is available or can be acquired, and the complexity of visual analysis of complex scenes. The problem becomes more challenging due to natural seasonal changes that affect the appearance of structures on the ground (vegetation, snow-cover, etc.). As a result, significant research effort that employs semantic analysis was expended to address this challenge. We developed two approaches to incorporate semantic information in analysis. The first using publically available GIS data (e.g., OpenStreet Maps, or AI based Microsoft building detector operating at scale). The second employs multi-pass satellite data over the same Area of Interest, in conjunction with low-level building structure detectors. Figure 29 shows an example of the OpenStreet Map data that can be used for detection of areas of change. When the data is available and is accurate it forms a strong basis for relatively accurate detection of areas of change. This analysis can confirm the presence and properties of detected structures with respect to the GIS data.

Figure 29. *Left, OpenStreet Map data overlaid on a satellite image (MSI pass over UCSD). Right, example of detected spliced building into a satellite image, using our building detector and OpenStreet Map.*

We also employed a combination of algorithms to semantically analyze image sets to identify changes. Specifically, a UMD's own building class detector, ICTNet and XDXD detector, were used in a combination to detect building bottom-up and flag inconsistent buildings across detectors or images. Figure 30 shows an example of UMD and ICTNet detection of an industrial area in Baltimore. It was extended to handle multi-pass semantic analysis to determine areas of change in an image and report these areas as heatmaps. In one case, we used a stack of 43 passes over UCSD from May 2014 to August 2017 to detect construction activity on the ground. The combined power of the detectors above enabled fusing information to flag areas of interest. Each detected area is rated based on the frequency of it being flagged as inconsistent among the 43 passes. In the figure below (Figure 31), this is illustrated on a satellite image (among the 43) for detecting a change on the ground, in this case a construction site that is marked in green. Similar experiments and results were performed on other AOIs such as Omaha and San Fernando where IARPA-CORE3D provided multiple passes over each area.

Figure 30. *(Left) UMD detection of buildings,(Right) ICTNet detection*

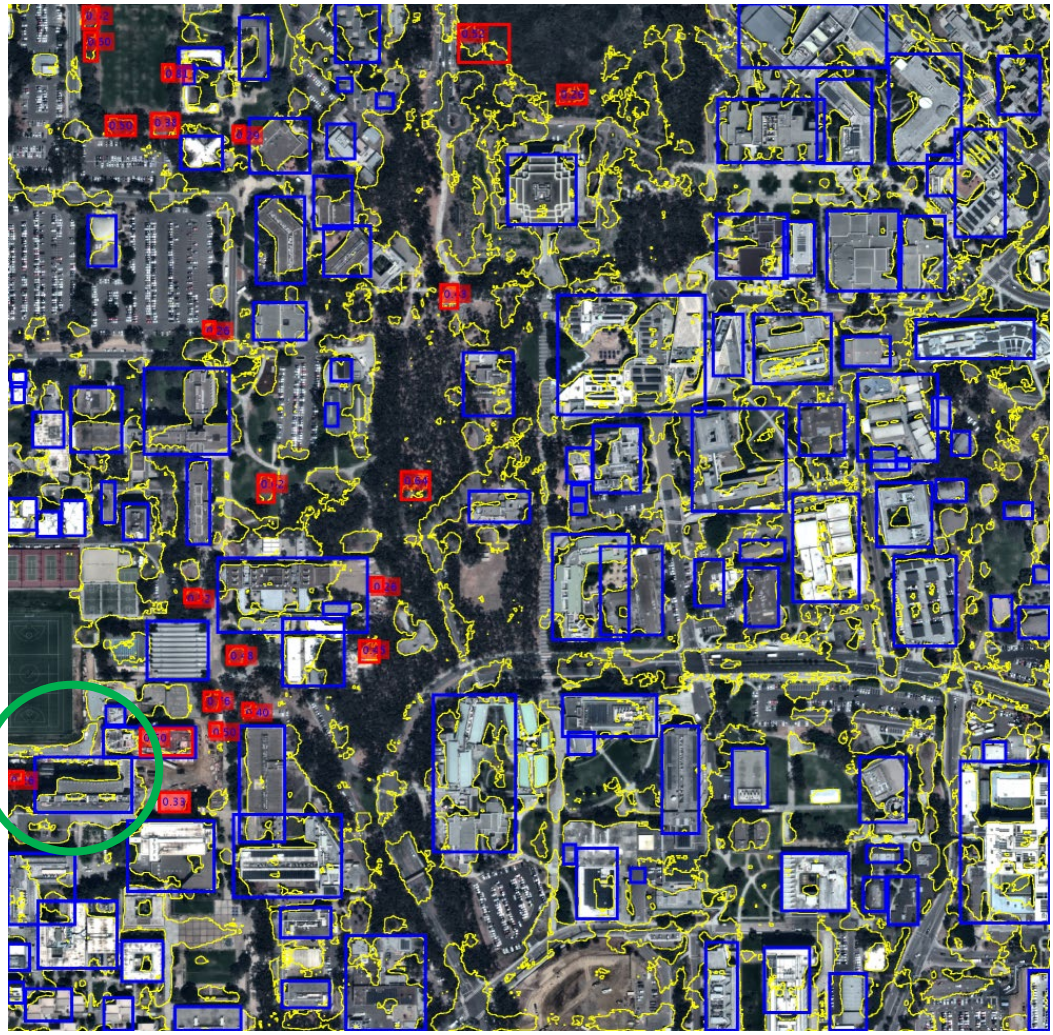


Figure 31. *Detection of changes on the ground in one image with respect to a stack of 43 images from UCSD to detect areas of interest, including construction activity.*

4.1.4 Inpainting partial and full structures in Satellite images

Inpainting is the process of removing objects/regions from imagery and replacing it by background data. The objective is altering the contents of an image and therefore is of high significance. Automatic inpainting remains undeveloped. So some of our effort was to develop algorithms to anticipate inpainting performance down the road. We trained a network that is able to inpaint partial or full building structures by the use of trees. The contribution here was to train a tree dis-occlusion network, specifically a network that uncover a hallucinated structure under a tree via the context of the scene. This same network can be used to change structures to background structures. Figure 32 provides an example of the process. A tree is placed over a building structure we want to inpaint, and the network creates an inpainting given the scene context (to right). The algorithm as described was visually and at the signal level effective, so that our tamper detection algorithm (GSRNET) could not detect the inpainted pixels and was performing at random. We did not develop a new inpainting detector in this effort, so it remains open for research. Inpainting detection of regular images and videos is a very challenging problem because signal level cues tend to be diminished since the generation algorithms use the inpainted image as a source of signal reconstruction. Consequently, we think that semantic-based image analysis is the most appropriate approach for detecting this type of tampered images.

***Figure 32**. Example inpainting of part of a building. Top left is the original image, (right) inpainted part of a building, bottom left is the tree placement over the structure and the mask of the tree (in red).*

We also studied and developed guided inpainting algorithms. In this case, a structure in an image is automatically replaced by data from another class, present in the same image or in another image. For example replacing a building with a class of shrub, forest, or road. We employed OpenStreet Maps to automate the detection of source and detect patches, then a patch Match algorithm was used to create generic textures that can be further adjusted to the specific textures in the image. Figure 33 shows an example of this inpainting in an image where specific structures were inpainted into vegetation.

Figure 33. *Building structures were inpainted into forested areas.*

We conducted research on seasonal inpainting (meaning changing the appearance of an image from one season to another. The algorithms employed OpenStreet Map data and histogram mapping to adjust the appearance of images. For example, a source image of Omaha in Sept was transformed to Nov using this approach (see Figure 34). The image shows zoomed in regions of interest to convey the real and seasoned appearance of vegetation on the ground.

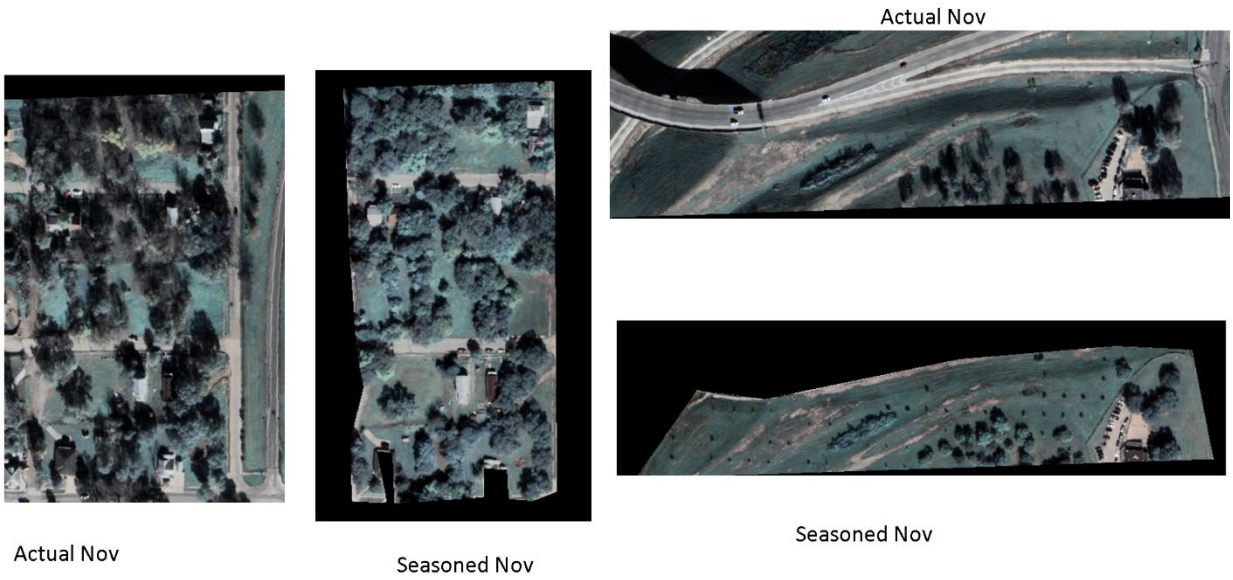


Figure 34. *Season inpainting in Omaha, changing Sept image to Nov image.*

We also conducted research on inpainting of text regions and overlays on images. The objective was to cover-up the presence of overlaid text and graphics. The algorithm use Deep Inpainting Priors to generate such images. In the Figure below (Figure 35), from left to right are examples of input image, text overlay and inpainted output. The performance was highly realistic. Our research on detecting the inpainted pixels was not fruitful, since the residual signals after inpainting can easily blend within the normal signal in the images.
In summary, detection of inpainting is a very challenging problem, it remains open for research and require considerable effort.

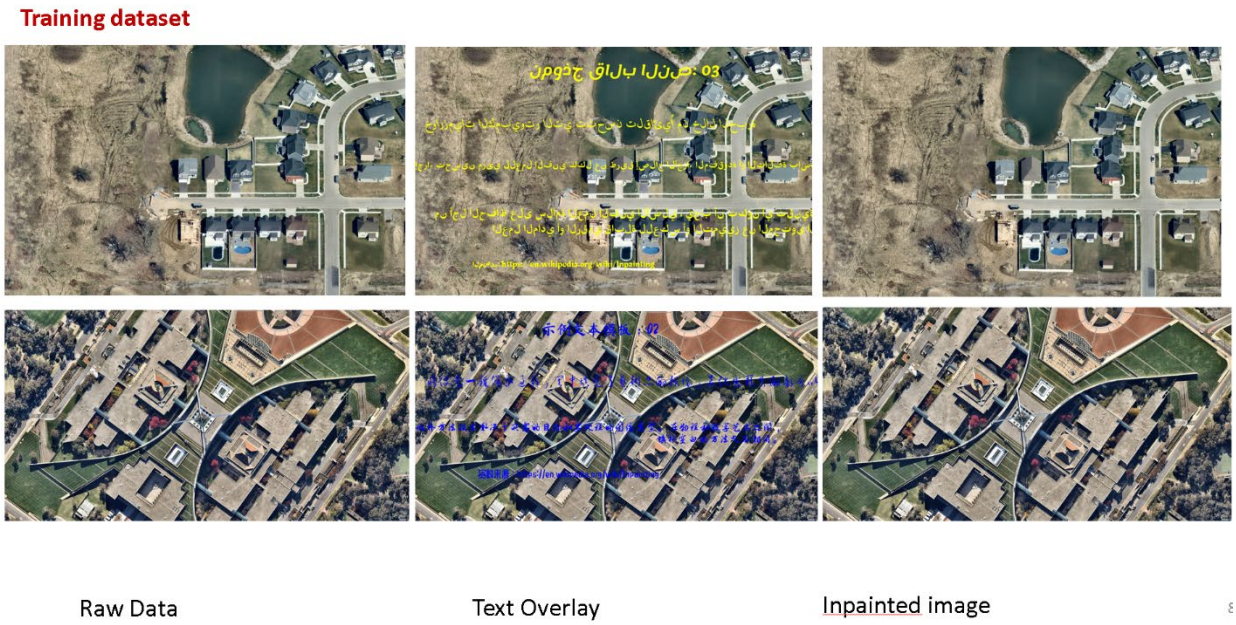


Figure 35. *Text and graphics inpainting in satellite images*

4.1.5 Software Transition

Software developed under this effort was transitioned in three stages. In the first general tools for analyzing MSI data were dockerized and made available to NGA. Specifically, the software included

1. MSI to RGB image transformations
2. Shadow detection and image correction
3. Semantic segmentation into GRSS classed (buildings, vegetation etc.)
4. Image to Digital Height Map regression

In total 8 capabilities with respective models were packaged and transitioned to NGA. The models involved combinations of RGB and or MSI images to widen the scope of their utilization. Figure 36 summarizes this package.

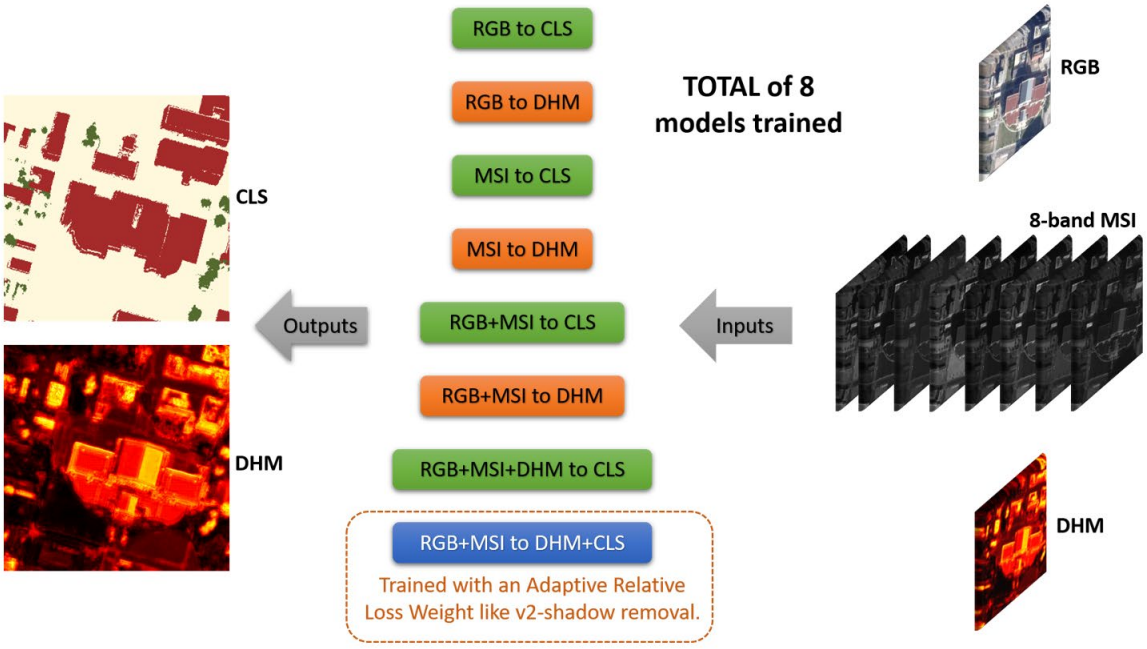


Figure 36. *The first batch of software transition to NGA included 8 distinct capabilities.*

The second stage transition focused on tampering detection in a satellite image. The Deep Learning network GSRNET that is trained on satellite data was packaged and transitioned to NGA. This software is trained and tested on RGB data, and to a lesser extent on RGBs generated data.
In the third stage the semantic-based multi-pass change detection was packaged and uploaded to gitlab. The package included two additional building detectors (ICTNet and XDXD). And included automated computation of heat-mapped areas based on the detections in multiple-passes.

4.2 Frequency Perspective of Adversarial Robustness

Even Though the field of adversarial robustness is studied extensively, not many focus on studying its relation between frequency space of an image. In this section we shall explore some preliminary results towards understanding the same.

The DCT space
We analyze the frequency space of an image using the Discrete cosine transform framework. DCT gives us a representation of any signal as a linear combination of cosine transforms. The coefficients of the constituent cosine waves inform us about the apparent "importance" of each frequency.

Part I: Learning the inverse
The DCT is a lossless transform. Even though the coefficients give us a fair idea about which components are weighted more, the "importance" of each component to a neural network can be very different. To understand this, we plan to "learn" the inverse DCT transform and have an attention head highlight the most important frequencies.

Input representation: The DCT coefficients are arranged in a zig zag pattern for each 8x8 block. We rearrange it in such a way that all coefficients are in ascending order and are accessible by channel index. Essentially a DCT batch of N,C,H,W is rearranged to N,64*C,H/8,W/8.

Model: To learn the inverse, we utilize a U-net style network with an attention module at the end of the decoder. The network takes in a modified DCT representation of the image and outputs a reconstructed RGB image. To ensure that the attention map indeed corresponds to the respective frequencies, we apply group convolutions throughout the network, leading up to the attention layer. In the end we add a subpixel CNN layer to rearrange the channels, in order to reconstruct the RGB image. We ran the model on Tiny Imagenet dataset and the corresponding reconstructions are shown below:"

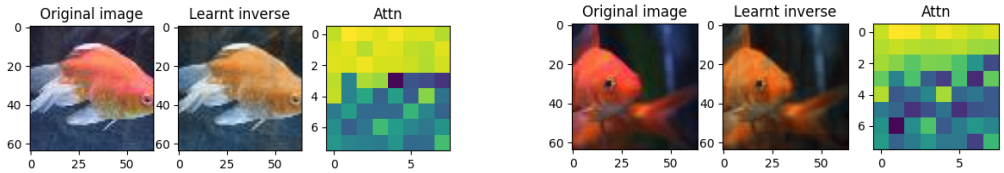


Figure 37: *Images from TinyImagnet Validation along with their learnt inverse and corresponding attention map. Note that the attention map is a 64x1 vector, resized to 8x8 for* visualization.

Application to adversarial attacks

In this section, we shall explore the initial experiments on how we can apply the above results to bolster the efficacy of adversarial attacks. The architecture diagram of the proposed system is shown below (Figure 38). We aim to utilize the attention maps to accurately identify the specific frequencies that matter to the model and then proceed to adversarially attack them.

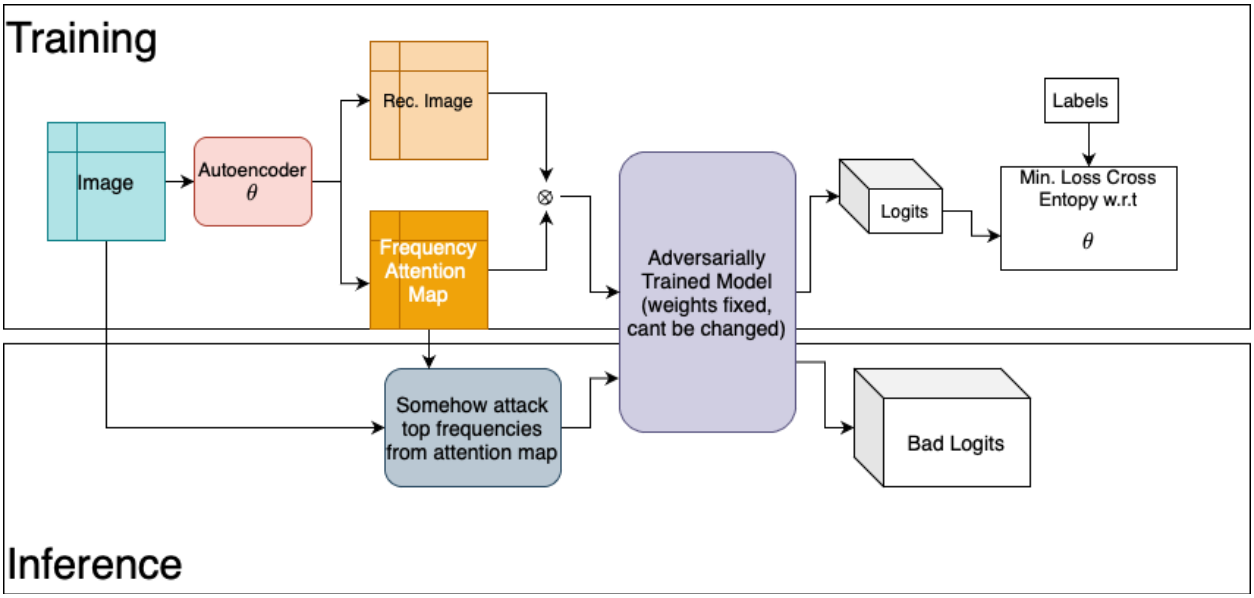


Figure 38: Proposed Adversarial Attack architecture

Once we obtain the attention maps of which frequencies are important, we plan to utilize this knowledge to target a trained classifier on these specific frequencies only. To achieve this we

modify the existing PGD algorithm to focus only on the current set of frequencies. The results of the transfer experiments on an adversarial model (trained with eps = 4/255) are shown below (Figure 39). In the figure the black line is the transfer results with standard PGD, while the blue line represents transfer using our method. The X-axis represents the "top-k" frequencies chosen from frequency attention map and the Y-axis represents the accuracy after attack. We can see that for eps = 4/255 (left), our method under performs PGD based transfer.

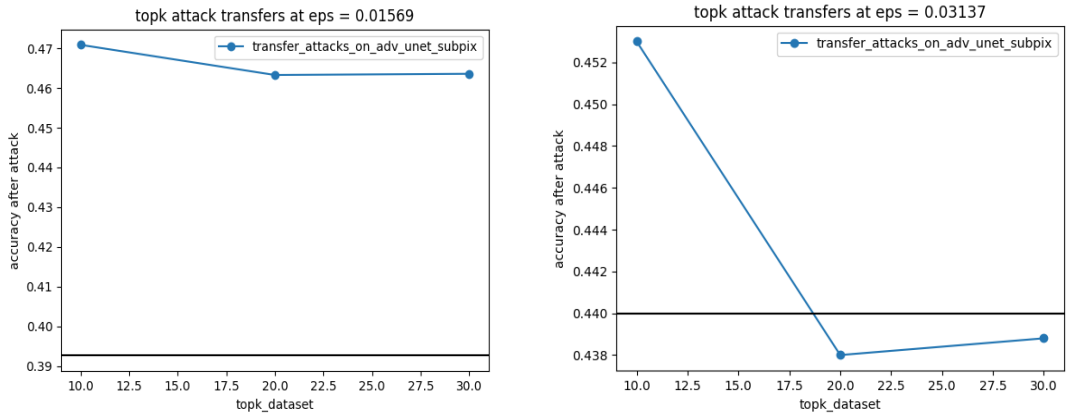


***Figure** 39. Top-k attack transfers, (left) eps = 0.01569 (= 4/255), (right) eps = 0.03137.*

Part II: Learning from logits

The primary idea behind this is simple and is illustrated in the diagram below.

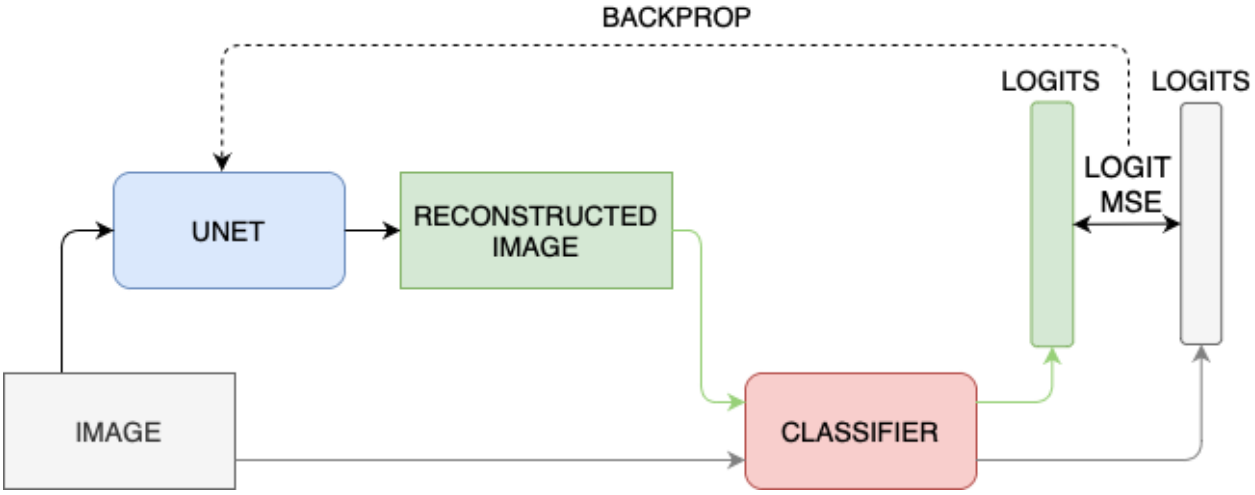


***Figure 40:** Perceptual Loss based image reconstruction.*

We take an RGB image, pass it through an image as well as a pre-trained, frozen classifier. Now, instead of the traditional UNet based learning where we have a MSE loss between reconstructed images and original, here we pass the reconstructed image through the classifier as well and have an MSE loss between the two logits.

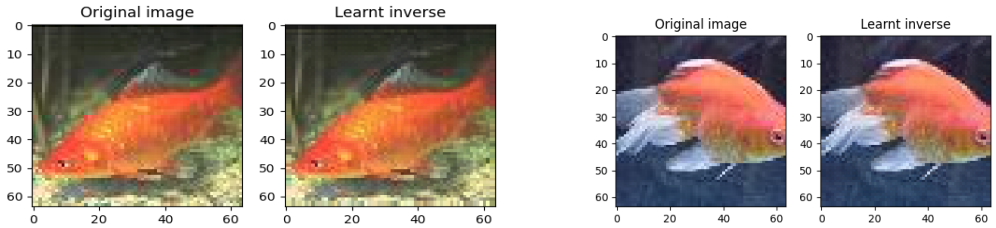


***Figure 41**: Reconstructions on TinyImagenet validation set.*

Now we modify the original architecture slightly to include a pre-trained adversarial classifier.

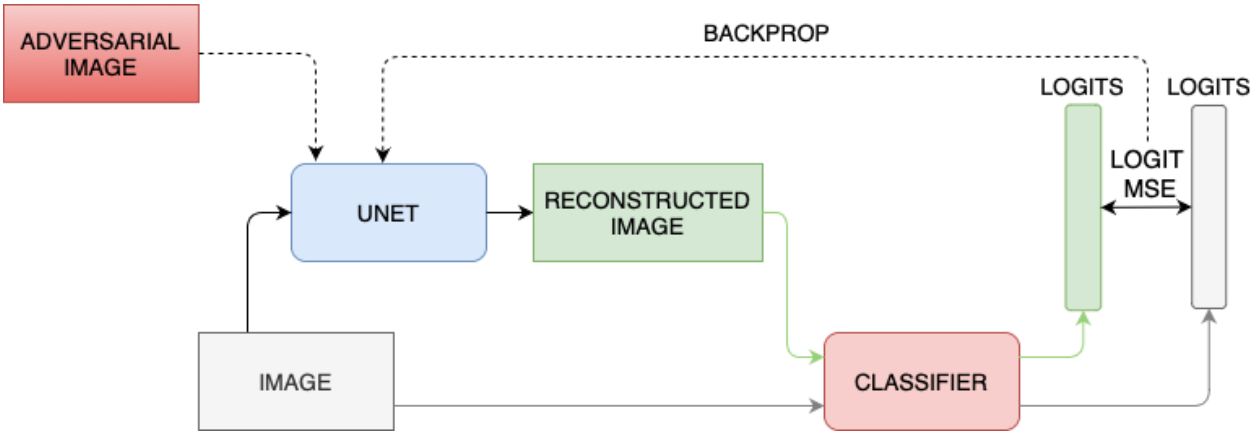


***Figure 42:** Proposed adversarial defense utilizing perceptual loss with respect to trained classifier.*

We study whether this modification can be used as a "preprocessor" to weed out adversarial samples? If the Unet can learn to map the adversarial image to a reconstruction that produces the same robust logits, then this should lead to an increased model robustness. The results are shown in Table 5 below:

***Table 5** - Performance of 2 models one with normally trained classifier and another with adversarial classifier (trained with epsilon = 4/255, shown in bold).*

Unet-Logit adv with Res18 normal			
Attack	2/255	**4/255**	8/255
FGSM	15.29	**4.97**	1.87
Baseline res18 normal FGSM	10.06	**2.8**	1.24
PGD 10	10.85	**1.28**	0
Baseline res18 normal PGD 10	4.68	0	0
Unet-Logit adv with Res 18 adv			
Attack	2/255	**4/255**	8/255

FGSM	37.45	**28.18**	15.29
Baseline res18 adv FGSM	37.36	27.96	14.91
PGD 10	37.09	**26.6**	11
Baseline res18 adv PGD 10	36.97	26.04	10.42

We see that this pre-processing provides a good boost in normally trained networks and a modest boost in accuracy for adversarially trained models. We wish to further investigate the behavior of these networks in this space.

5.0 Results and Discussion

This research effort has encompassed several types of media, namely images, video, metadata and image sets. A wide range of media tampering was tackled, from local changes to a media item, for example splicing a face of changing a data of a metadata item, to fully synthesized media items (GAN generated faces and scenery).

The research objectives expanded in quality and quantity as the program progressed and the landscape evolved. Many of the topics we described did not exist and therefore require us to refocus our research to cope with the increased sophistication that came on line. It is also the case that the field will continue to evolve to challenge our assumptions and knowledge as the nature of the eco system is that there is systemic competition between generation and detection.

We developed a large set of algorithms, described many of them in this report, and transitioned operational software to DOD. These algorithms were tested on at least tens of thousands of media items, and statistical performance analysis was conducted in house and by NIST to characterize the performance of these algorithms. However, the changing landscape will likely change the performance characteristics, the adversarial cycle is very alive.

As datasets increased in size it became very important to focus on reducing false alarms in algorithms. This is an ongoing effort which is very much complicated by the nature of the search being a needle in a hay stack. Only a limited number of media items are of consequence as far as manipulation is concerned, detecting those without an overwhelming response remains a challenging and open problem. Semantic and contextual analysis may be of some assistance, as well more adding media signatures at creation and multiple types of watermarks.

6.0 Conclusions

The analysis of media for fingerprints of tampering is an enduring problem and it is likely to increase with the growth in media generation, real and synthesized. Our research tackled a snapshot of the open challenges in this domain. The proposed algorithms used and expanded the state of the art in detection of traces of manipulation in different types of media and taking into account different objectives of manipulation.

7.0 References

[1] "SwapMe," [Online]. Available: https://itunes.apple.com/us/app/ swapme-by-faciometrics/ *This company has been acquired by Facebook and no longer available in AppStore.

[2] "FaceSwap," [Online]. Available: https://github.com/ MarekKowalski/FaceSwap/.

[3] P. Zhou, X. Han, V. I. Morariu and L. S. Davis, "Two-Stream Neural Networks for Tampered Face Detection," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017 .

[4] B.-C. Chen, P. Ghosh, V. I. Morariu and L. S. Davis, "Detection of Metadata Tampering through Discrepancy between Image Content and Metadata using Multi-task Deep Learning," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017.

[5] P. Zhou, X. Han, V. I. Morariu and L. S. Davis, "Learning Rich Features for Image Manipulation Detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[6] P. Zhou, B.-C. Chen, X. Han, M. Najibi, A. Shrivastava, S. N. Lim and L. S. Davis, "Generate, Segment and Refine: Towards Generic Manipulation Segmentation," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2020.

[7] N. Yu, L. Davis and M. Fritz, "Attributing Fake Images to GANs: Learning and Analyzing GAN Fingerprints," in *IEEE International Conference on Computer Vision*, 2019.

[8] D. Kim, S. Woo, J. Y. Lee and I. S. Kweon, "Deep video inpainting," in *IEEE/CVF Conference on computer vision and pattern recognition*, 2019.

[9] S. W. Oh, S. Lee, J.-Y. Lee and S. J. Kim, "Onion-peel networks for deep video completion," in *IEEE International Conference on Computer Vision*, 2019.

[10] S. Lee, S. W. Oh, D. Won and S. J. Kim, "Copy-and-paste networks for deep video inpainting," in *IEEE International Conference on Computer Vision*, 2019.

[11] M. Ehrlich, L. Davis, S.-N. Lim and A. Shrivastava, "Quantization Guided JPEG Artifact Correction," in *European Conference on Computer Vision*, 2020.

[12] I. J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[13] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor and T. Goldstein, "Adversarial training for free!," in *NeurIPS*, 2019.

[14] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor and T. Goldstein, "Universal Adversarial Training," in *AAAI*, 2020.

[15] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In Advances in Neural Information Processing Systems (NIPS), 2226-2234, 2016.

8.0 List of Acronyms

AB-CNN - Attentive Bilinear Convolutional Neural Networks

AU – Action Unit

CIFAR – Canadian Institute For Advanced Research

CNN – Convolutional Neural Network

DCT - Discrete Cosine Transform

ELA – Error Level Analysis

EXIF – Exchangeable Image File Format

GAN – Generative Adversarial Network

GSR - Generate, Segment and Refine

JPEG – Joint photographic expert group image format

MSI – Multi Spectral Imaging

NGA – National Geospatial-Intelligence Agency

NIST – National Institute of Science & Technology

LSTM – Long Short Term Memory

PGD – Projected Gradient Descent

ReLU – Rectified Linear Unti

RoI – Region of Interest

RPN – Region Proposal Network

PRNU – Photo-response non-uniformity

R-CNN – Region-based CNN

ROC – Receiver operating characteristic

SOTA – State Of The Art

SVM – Support Vector Machine

t-SNE – t-Distributed Stochastic Neighbor Embedded

UCSD – University of California San Diego