Local and Holistic Methods for Image Forensics

OpenMFC 2021
Dr. Lakshmanan Nataraj
Principal Research Scientist. Trimble Inc.
(formerly Senior Research Scientist, Mayachitra,
Co-PI, Media Forensics (MediFor) project)

December 7th, 2021

Agenda

- Introduction
- Detecting tampered images using Resampling Features
- Identifying Seam Carved Images
- Detection, localization and attribution of GAN generated images
- A Holistic approach to image manipulation detection
- Conclusion

Introduction

- Modern software tools and advances in image processing and machine learning have made it very easy to manipulate or tamper digital images

- Examples of image manipulations include resampling, splicing, copy pasting, object removal, seam carving, to name a few

- Image Forensics deals with identifying images that have been manipulated

- Here, we will explore various local and holistic methods to detect tampered images

Resampling in Digital Forgeries

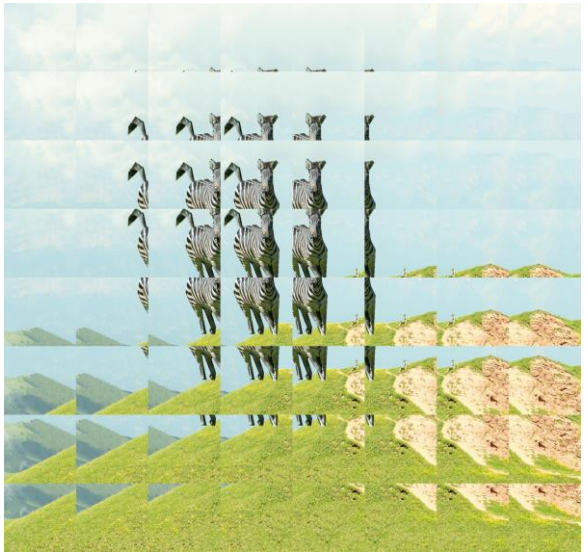
Cut out object

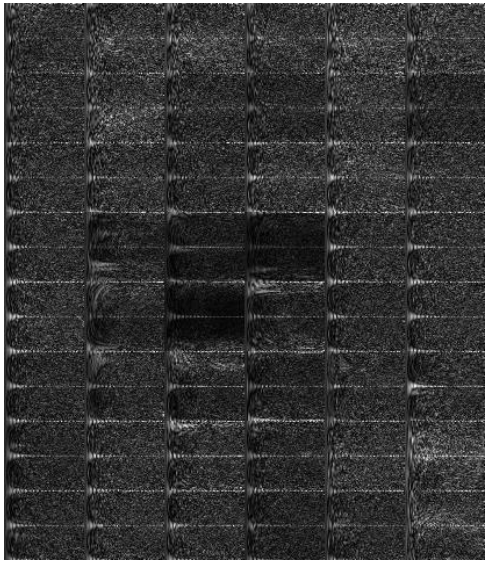
Upsample 30%

Splice into another image

The result will probably be saved as a JPEG (post-compressed)

Detection and Localization of Resampling Forgeries



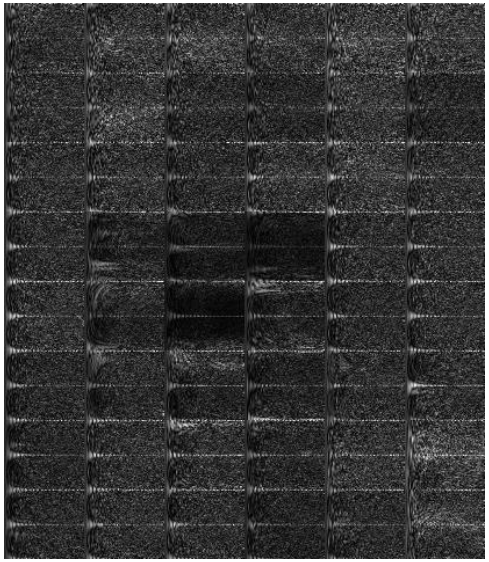


Divide image into overlapping patches

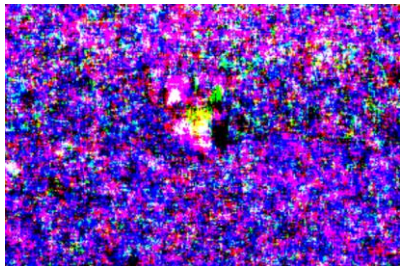
Extract a feature vector from each patch:

1. Linear predictor residual
2. Radon transform projections
3. 1D FFT to detect periodic signal

Detection and Localization of Resampling Forgeries

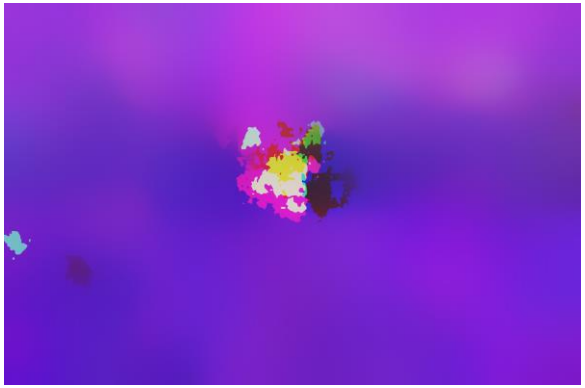


Extract a feature vector from each patch



To each feature vector, apply machine learning classifiers to characterize any resampling:

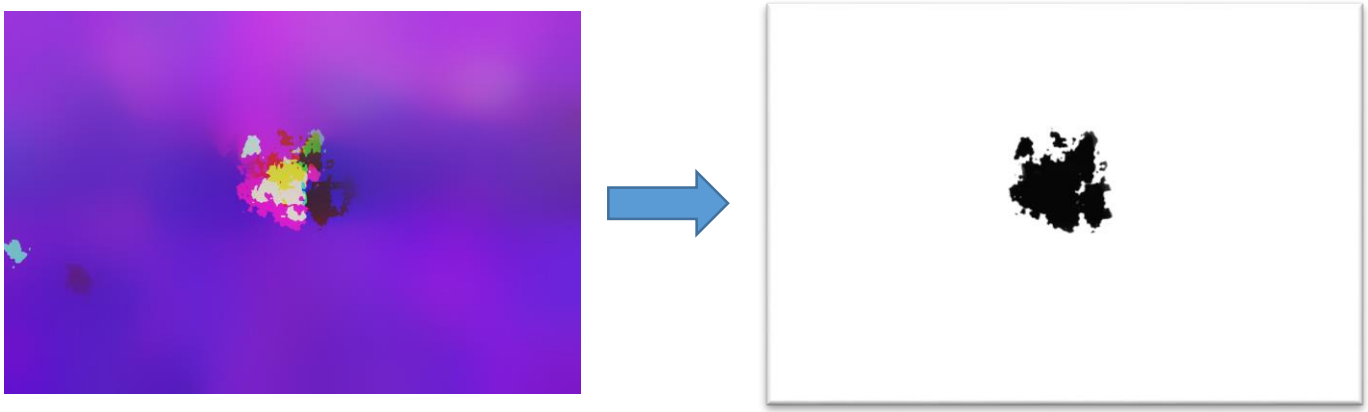
- rotation clockwise?
- counterclockwise?
- upsampling?
- downsampling?



Use fully-connected CRF to enhance unary potentials from each classifier.

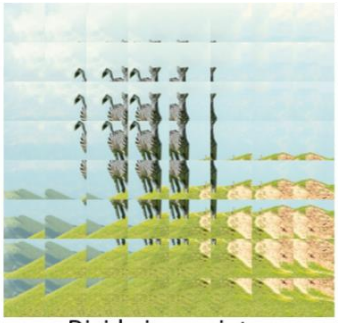
Each color represents a different resampling classifier's output.

Detection and Localization of Resampling Forgeries

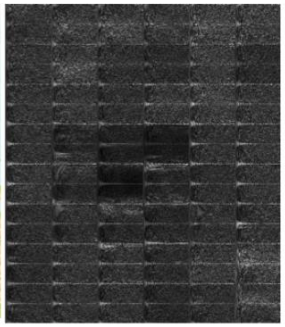


Extract manipulated region from resampling classifier map.

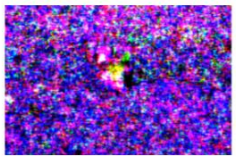
Summary of Resampling Detection Pipeline



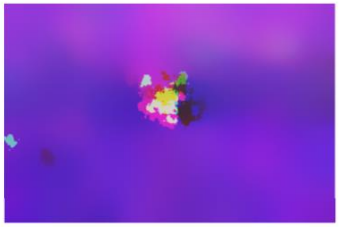
Divide image into overlapping patches (current experiments are 96x96 patches)



Extract a feature vector from each patch



To each feature vector, apply machine learning classifiers to characterize any resampling.

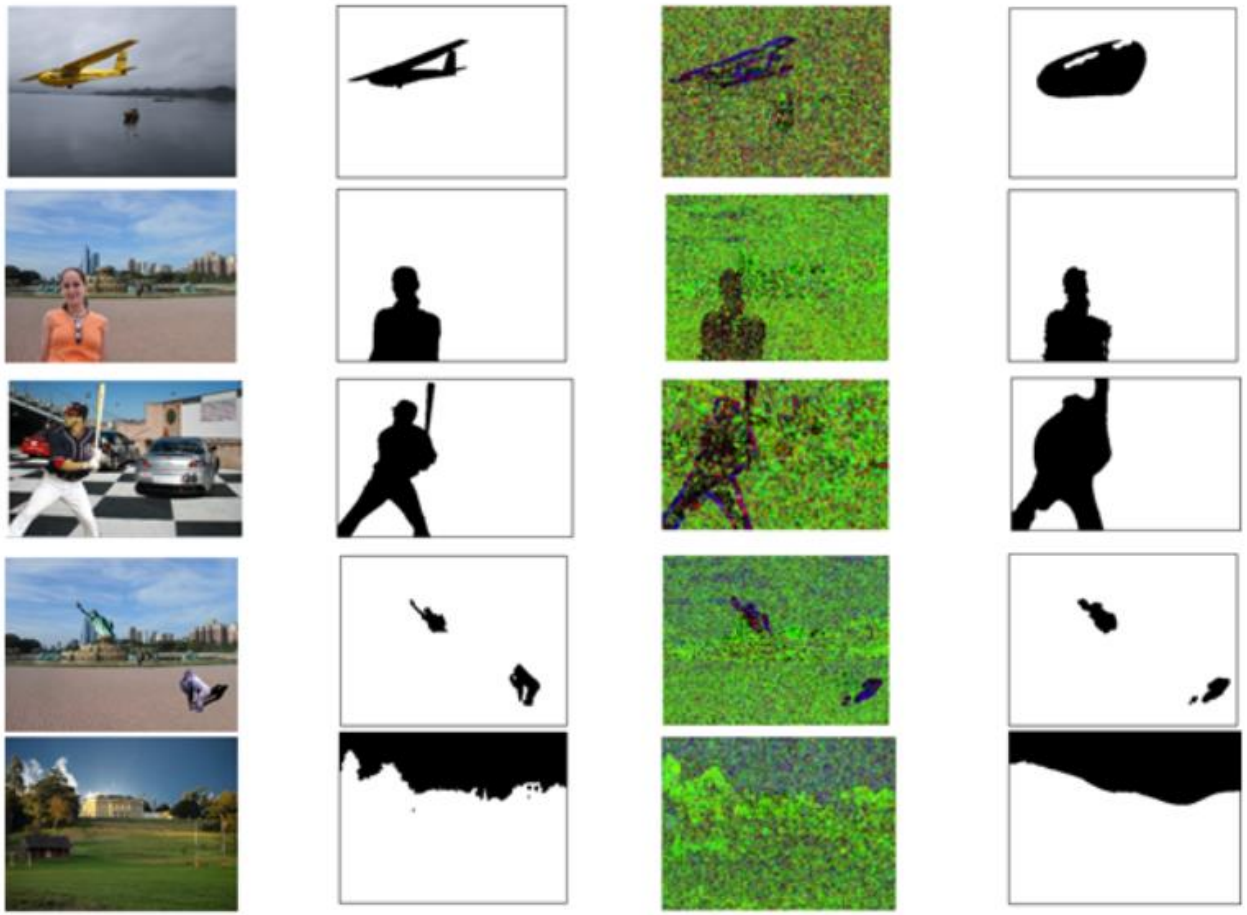


Use fully-connected CRF to enhance unary potentials from each classifier.

Extract manipulated region from resampling classifier map.

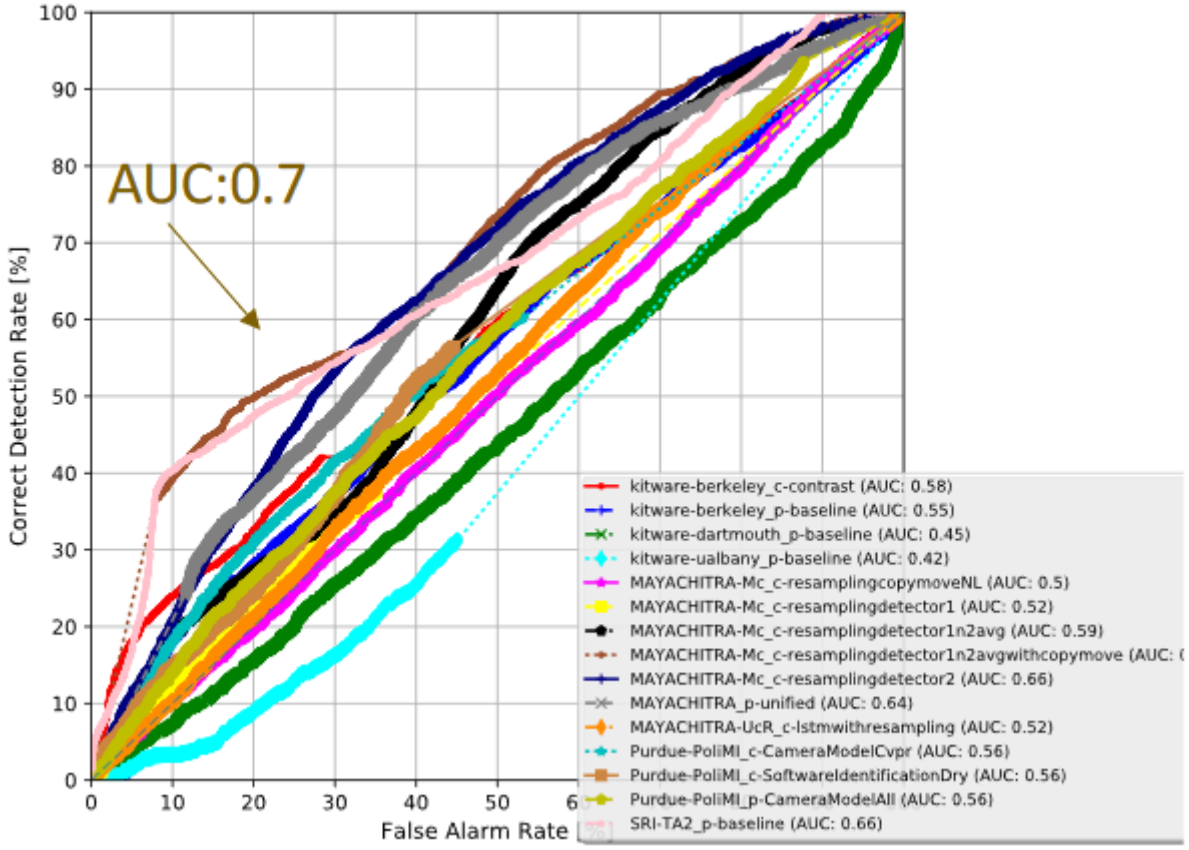
- Paper:
  - Bunk, Jason, Jawadul H. Bappy, Tajuddin Manhar Mohammed, Lakshmanan Nataraj, Arjuna Flenner, B. S. Manjunath, Shivkumar Chandrasekaran, Amit K. Roy-Chowdhury, and Lawrence Peterson. "Detection and Localization of Image Forgeries Using Resampling Features and Deep Learning." In *CVPR Workshops*. 2017.

Visual Examples



Tampered images | Ground Truth | Predictions | Estimated Masks

Top Score in NC 2017 Evaluations



Source: https://www.nist.gov/system/files/documents/2017/07/31/nist2017mediaforensicsworkshop_20170726.pdf

Agenda

- Introduction
- Detecting tampered images using Resampling Features
- **Identifying Seam Carved Images**
- Detection, localization and attribution of GAN generated images
- A Holistic approach to image manipulation detection
- Conclusion

Seam Carving and Seam Insertion

- Seam Carving and Seam Insertion are *Content-aware image resizing* methods which resize an image in a non-uniform way by preserving “important” content in an image

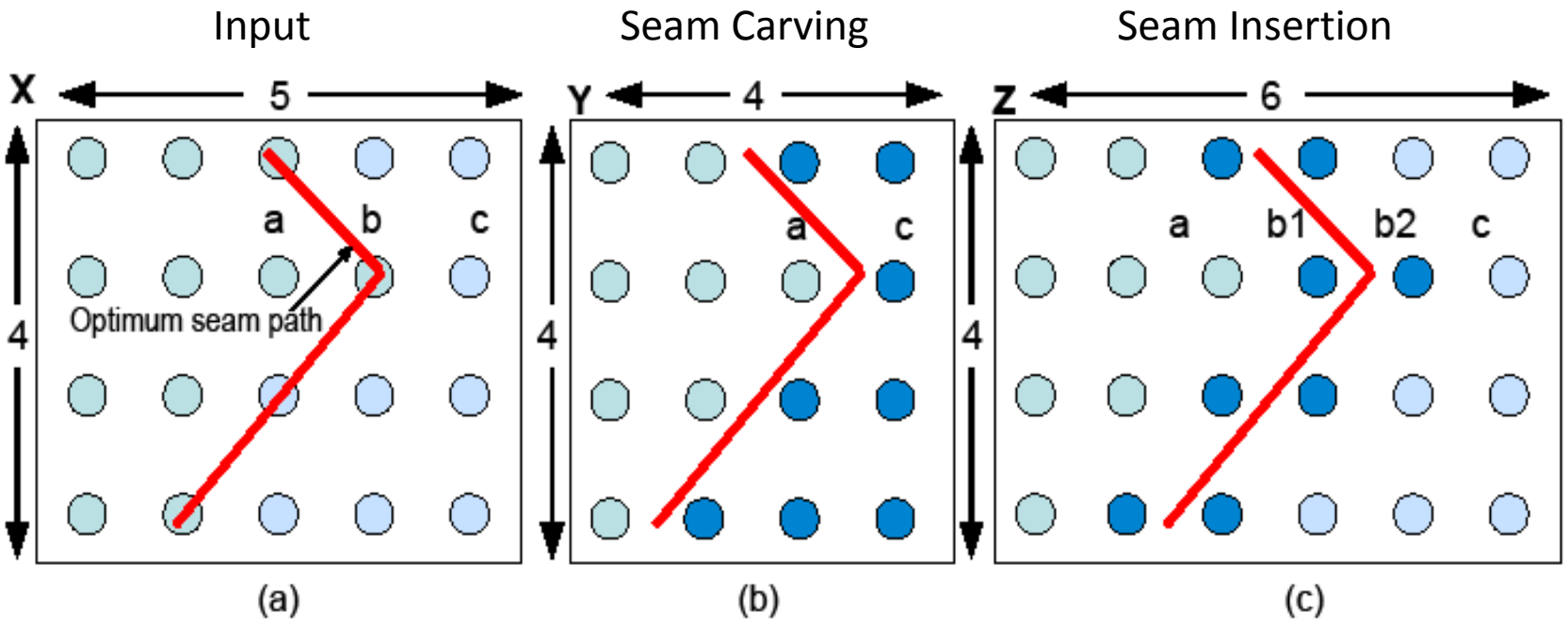
- A vertical/horizontal seam is a *path of 8-connected pixels* which traverses the image vertically/horizontally

- For seam carving, an image is reduced in size by deletion of seams whereas for seam insertion, two pixels are introduced for every deleted seam

- The path is obtained as the solution to an energy function related optimization problem - the optimal choice of seams maintains the image quality

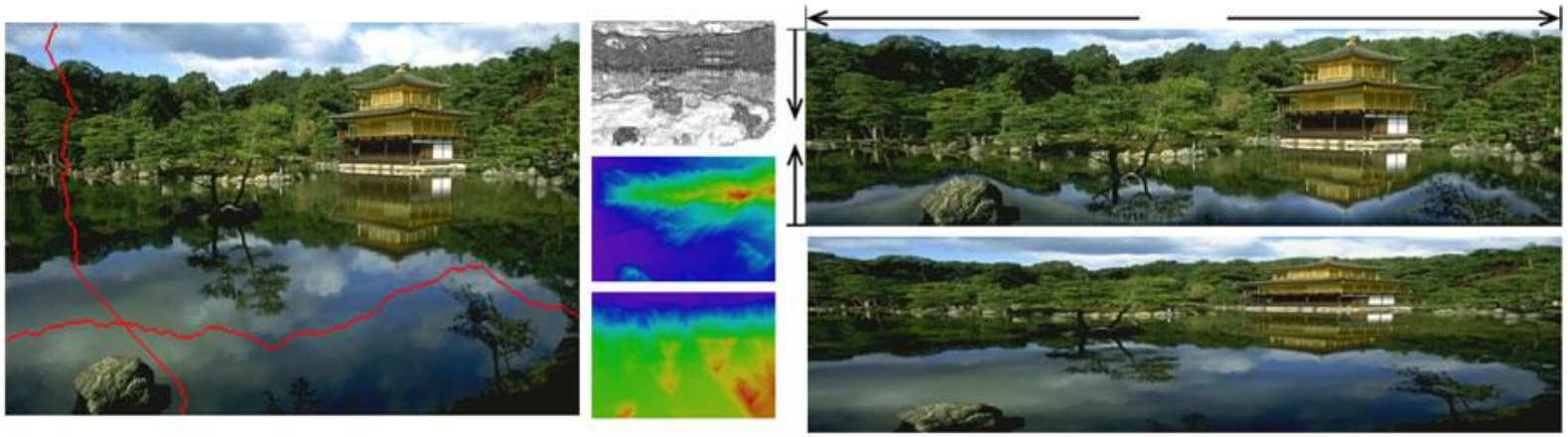
- Since the image content and/or its dimensions are changed, we treat the seam carved/inserted image as a tampered image

Seam Carving and Insertion



b1 = (a+b)/2
b2 = (b+c)/2

Seam Carving for Content-Aware Image Resizing



Source: https://www.faculty.idc.ac.il/arik/SCWeb/imret/

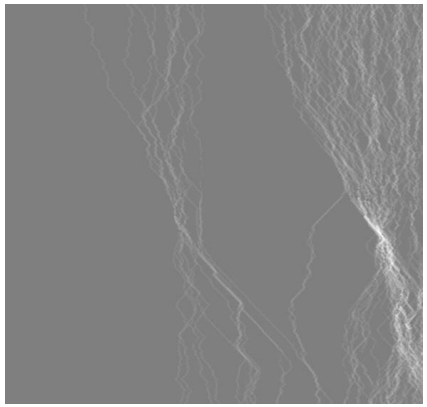
Avidan, Shai, and Ariel Shamir. "Seam carving for content-aware image resizing." *ACM SIGGRAPH 2007 papers*. 2007. 10-es.

Steps involved in Seam Carving

Original Image

Seam Carved Image

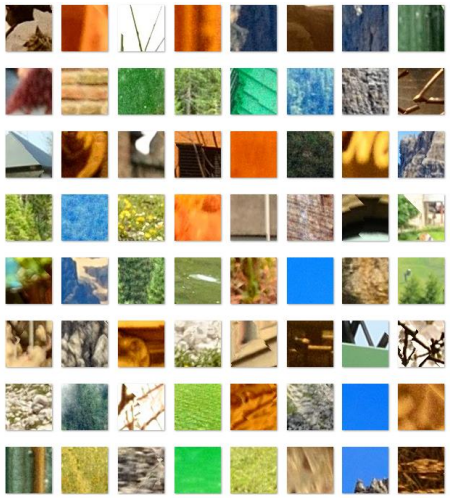
Original Image with Seams overlaid



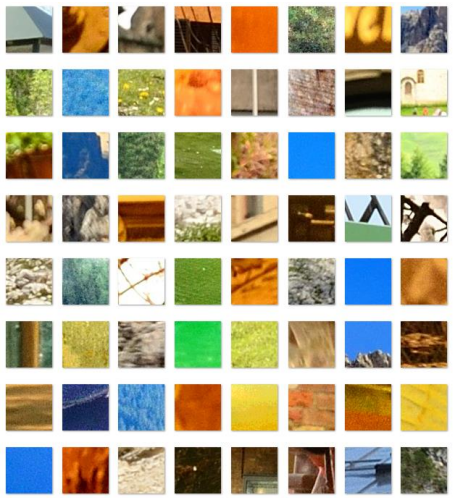
Mask Image after 100 seams removed

Seam Carving Detection and Localization

- Step 1: Detect Seam Carved Image patches
  - Create a dataset of seam carved patches and non-seam carved patches (64x64)
  - Train a CNN to identify seam carved patches (CNN1)
  - Output: a score in the range 0-1 whether the patch has been seam carved (1) or not (0)



Seam Carved Patches

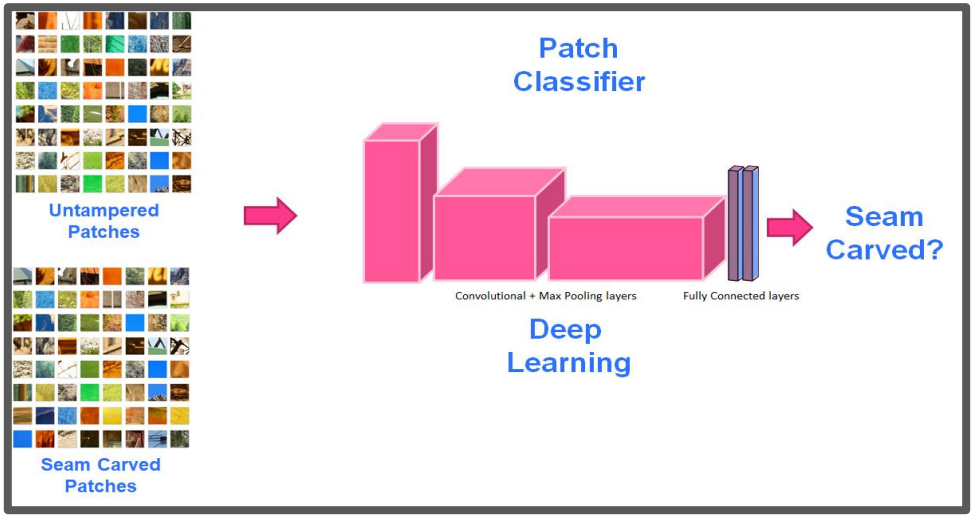


Non-Seam Carved Patches

Seam Carving Detection and Localization

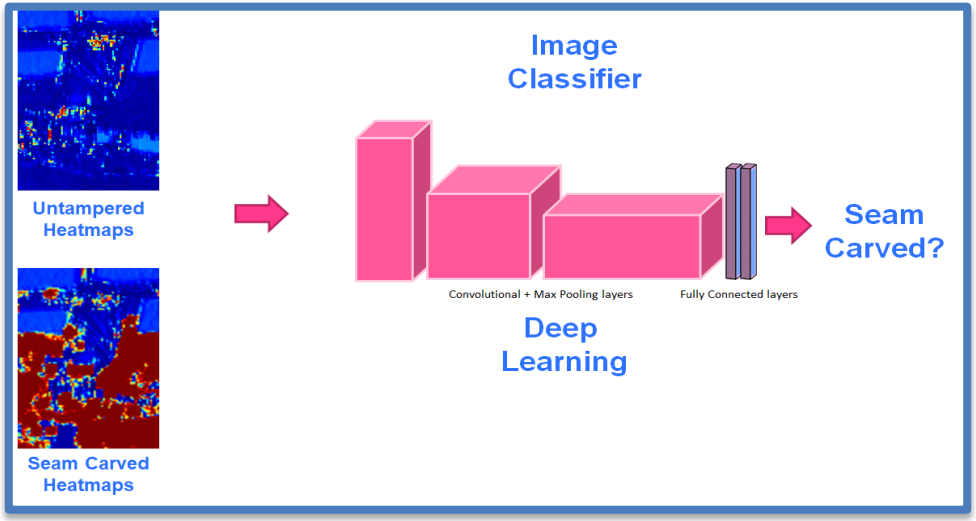
- Step 2: Detect Seam Carved Images
  - Divide an image into overlapping patches (64x64)
  - For every patch, compute the prediction score if it has been seam carved or not (1.0 – seam carved, 0.0 – not seam carved)
  - Obtain a heatmap which lights up in seam carved areas
  - Compute heatmaps for seam carved and non-seam carved images
  - With heatmaps as input, train a CNN to detect seam carved images (CNN2)

Two-Stage Approach



Stage 1 – Patch Level Detection

Train a patch detector to detect if a patch is seam carved or not and generate a heatmap

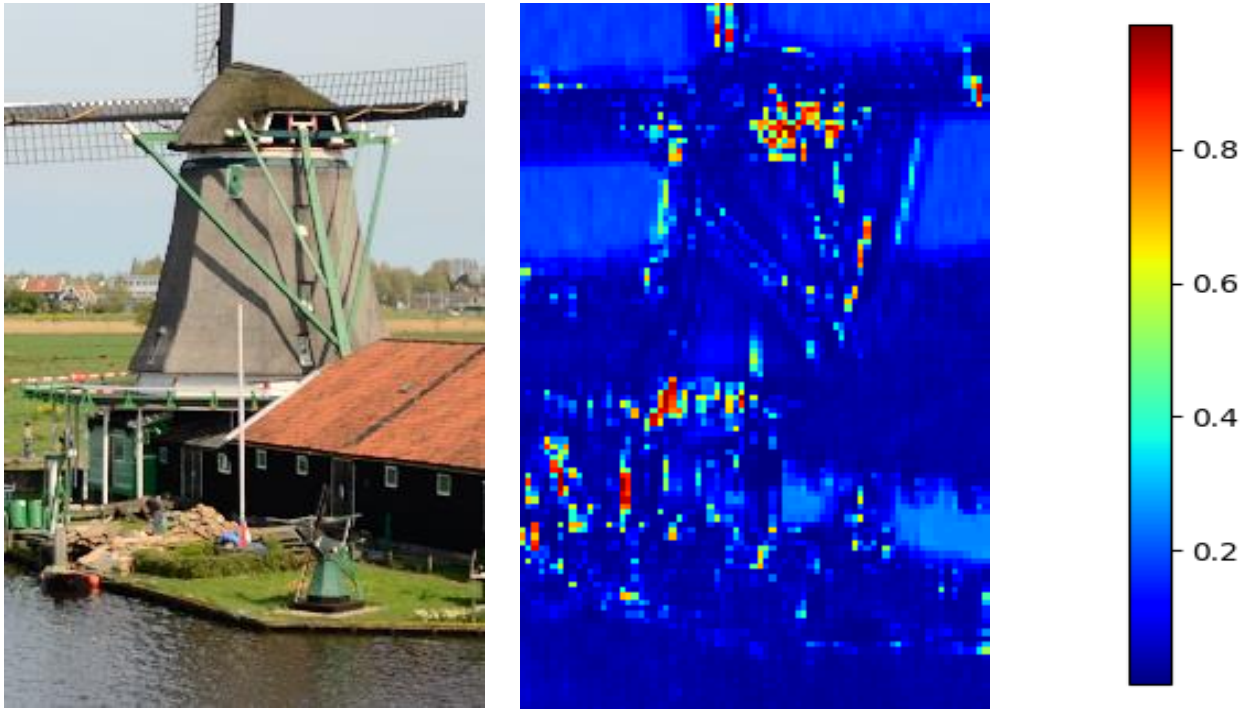


Stage 2 – Image Level Detection

Train an image level detector to detect if the image has been seam carved or not

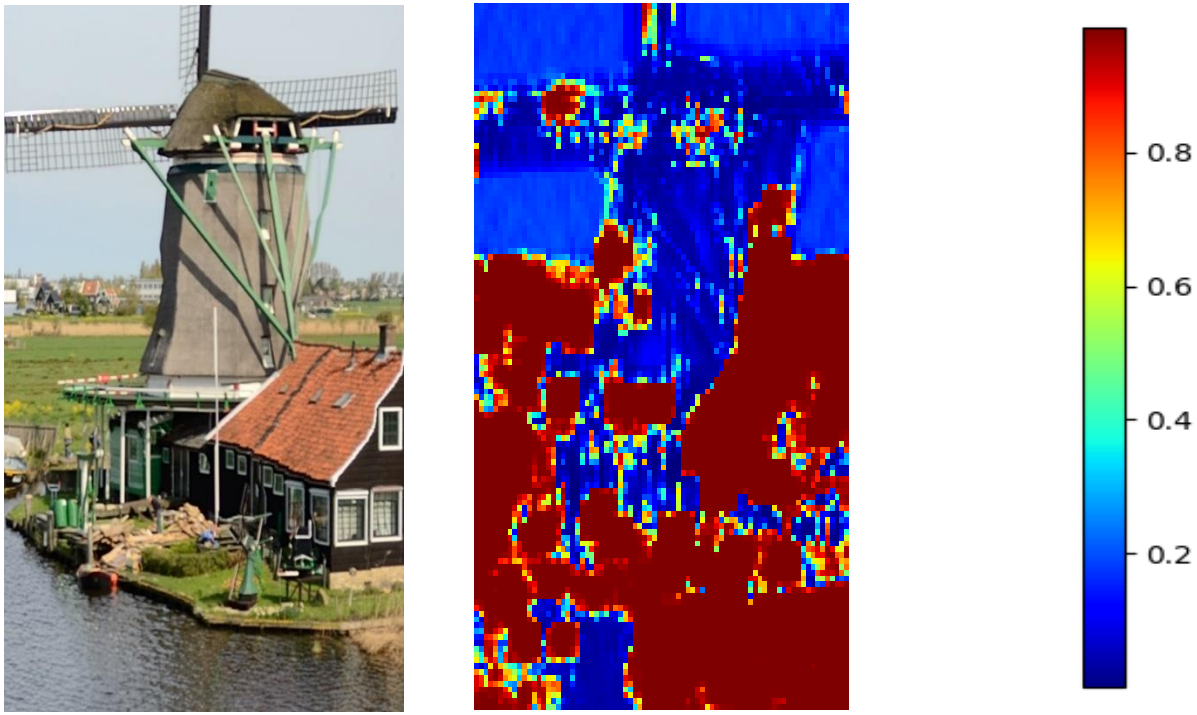
Results

No seam carving



Results

Seam carving

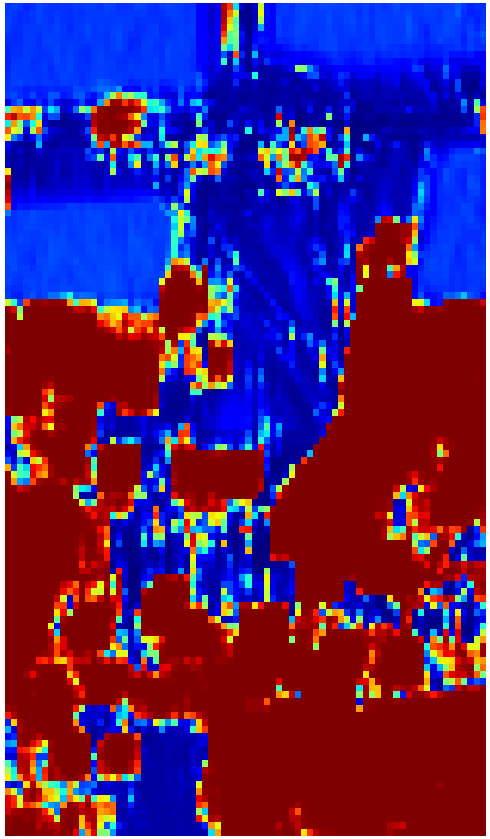


Analyzing the heatmaps

Seam carved image

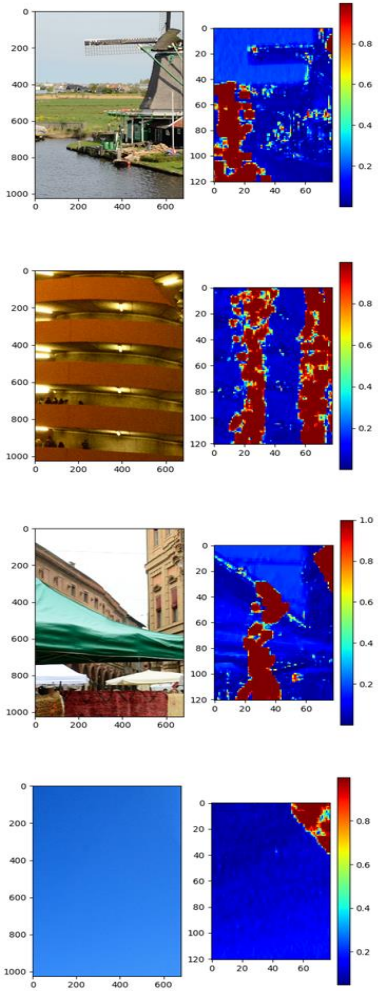
Iterative mask

Detected Heatmap

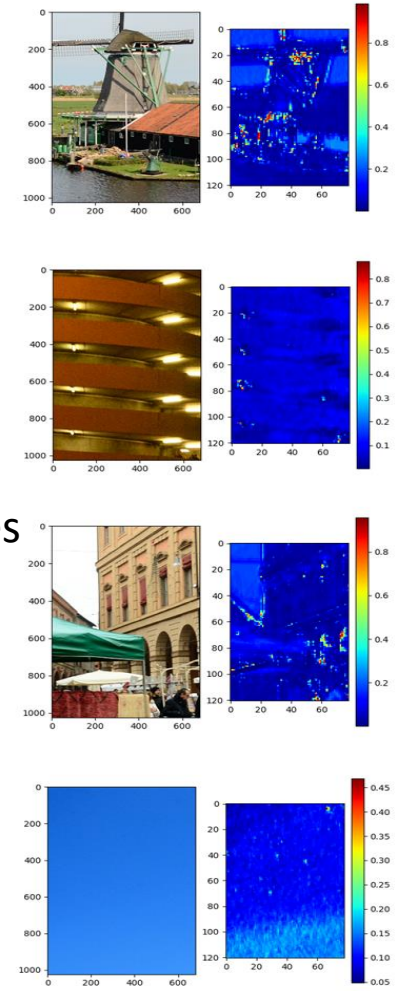


Results

Results on
Seam Carved Images

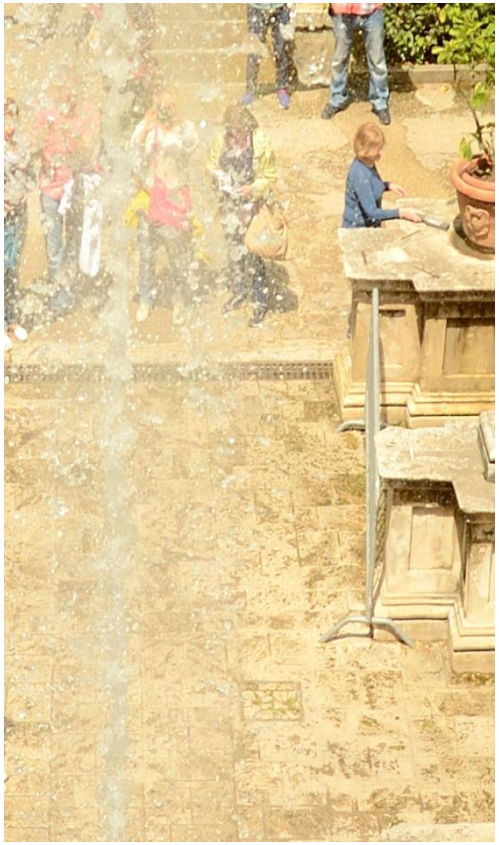


Results on
Non-Seam Carved Images

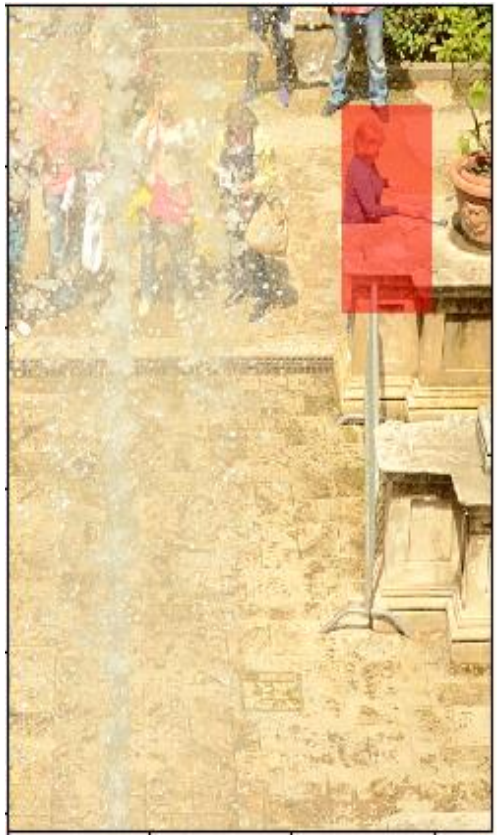


Object removal - Explainability

Original image

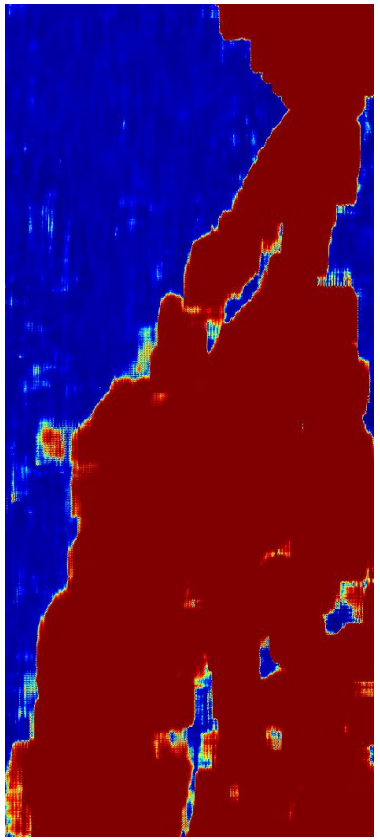


Object marked for removal



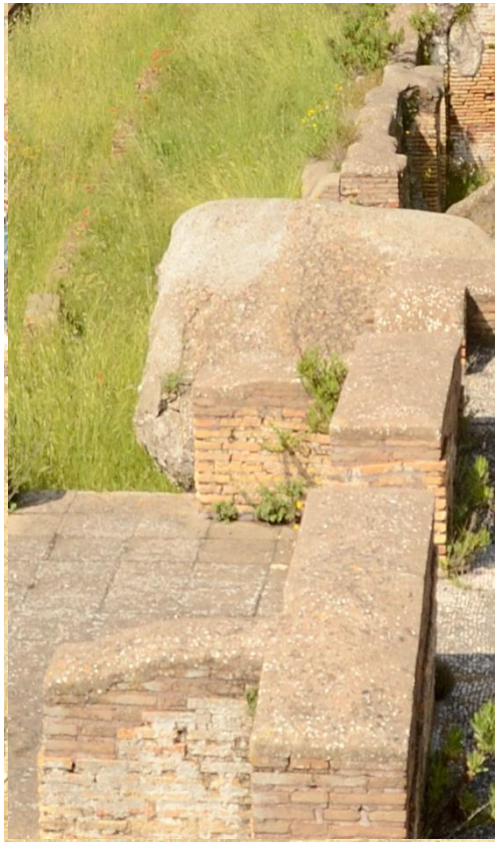
Object removed

Detected Heatmap



Object removal - Explainability

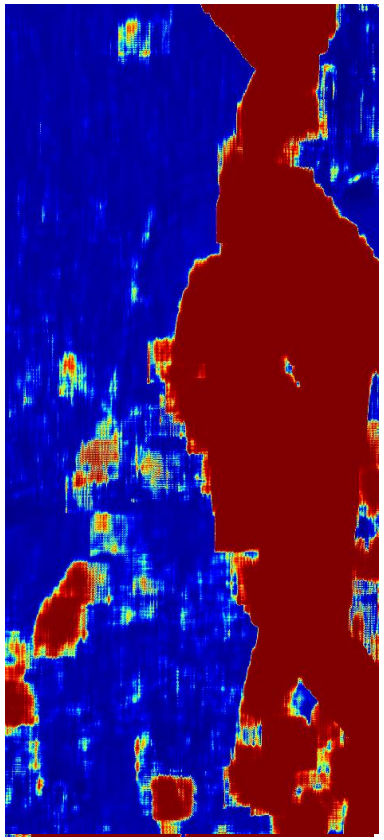
Original image



Object marked for removal

Object removed

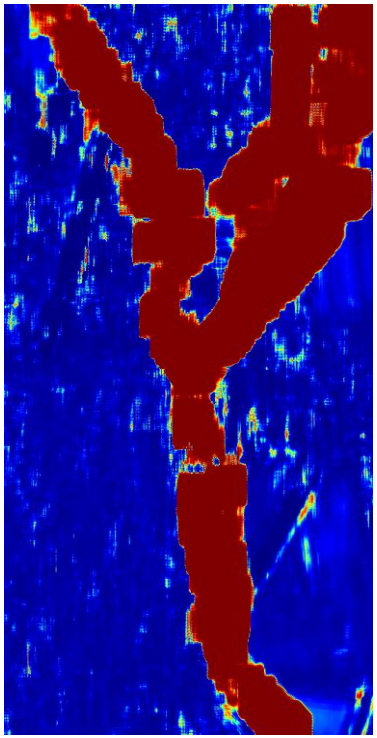
Detected Heatmap



Object removal (distortions) - Explainability

Original image | Object marked for removal | Object removed | Detected Heatmap

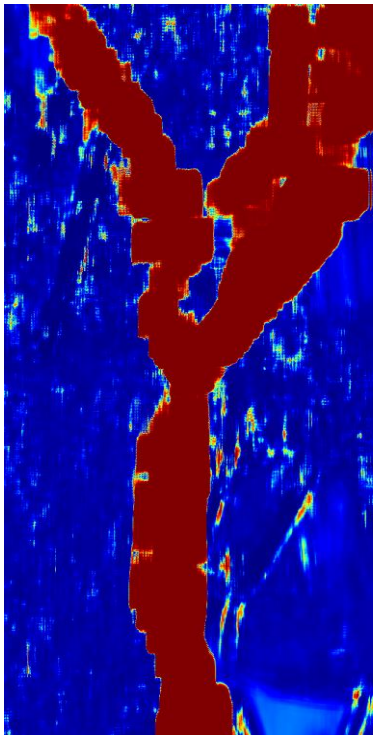


Object removal and preservation - Explainability

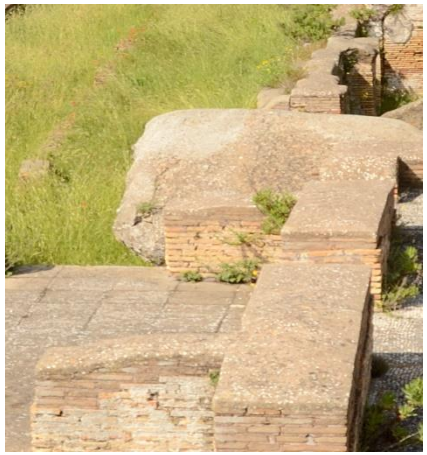
Original image | Object marked for removal | Object removed | Detected Heatmap



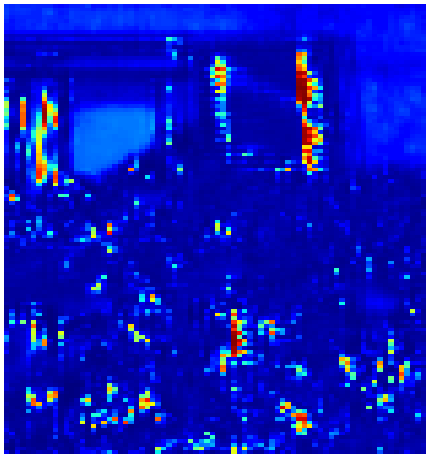
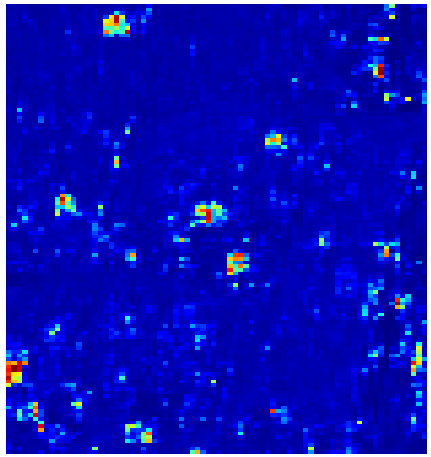
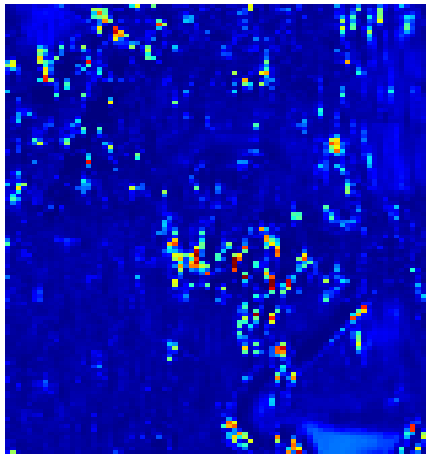



Heatmaps for original images

Non seam carved images

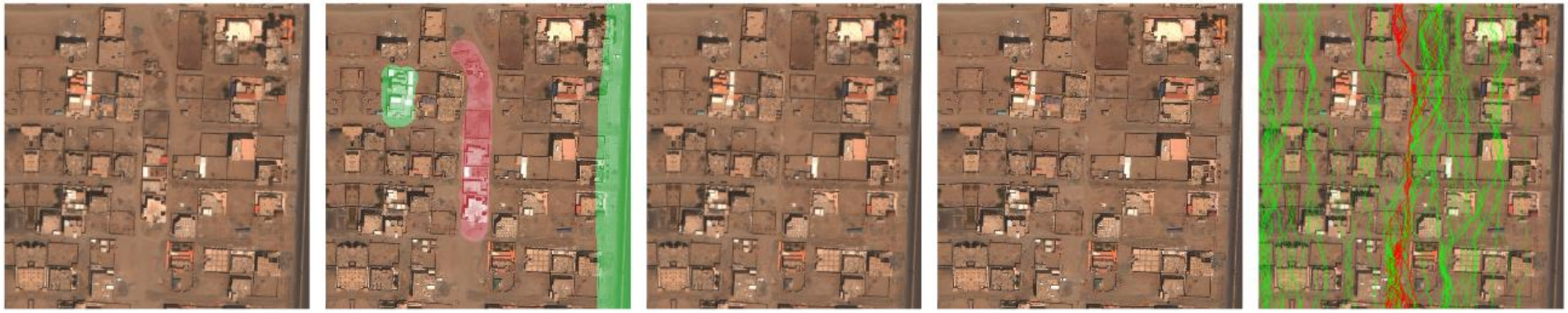




Heatmaps

References

- **Paper**
  - Nataraj, Lakshmanan, Chandrakanth Gudavalli, Tajuddin Manhar Mohammed, Shivkumar Chandrasekaran, and B. S. Manjunath. "Seam Carving Detection and Localization Using Two-Stage Deep Neural Networks." In *Machine Learning, Deep Learning and Computational Intelligence for Wireless Communication*, pp. 381-394. Springer, Singapore, 2021.
- **Extensions to Satellite Seam Carving Detection**
  - Gudavalli, Chandrakanth, Erik Rosten, Lakshmanan Nataraj, Shivkumar Chandrasekaran, and B. S. Manjunath. "SeeTheSeams: Localized Detection of Seam Carving based Image Forgery in Satellite Imagery." *arXiv preprint arXiv:2108.12534* (2021).



Agenda

- Introduction
- Detecting tampered images using Resampling Features
- Identifying Seam Carved Images
- **Detection, localization and attribution of GAN generated images**
- **A Holistic approach to image manipulation detection**
- Conclusion

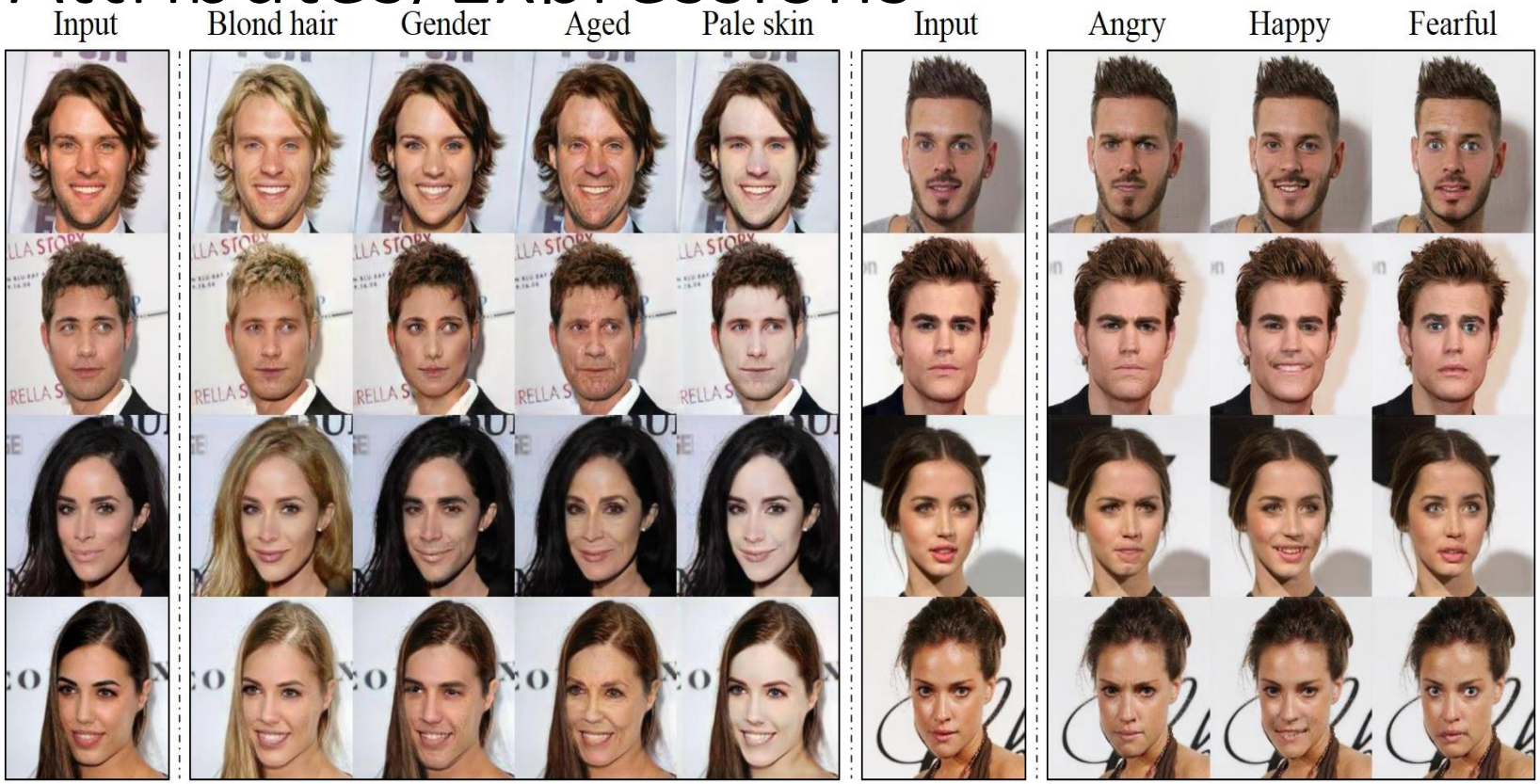
Introduction to GANs and DeepFake

- Fake news and fake media are making headlines everyday
- Recent advances in Machine Learning (ML) and Artificial Intelligence (AI) have made it very easy to synthesize digital manipulations in images and videos
- Developments such as Generative Adversarial Networks (GANs) and DeepFakes have brought in newer attack avenues
  - computer generated (CG) faces, augmenting faces with CG attributes/expressions, seamless transfer of texture between images

Introduction to GANs and DeepFake

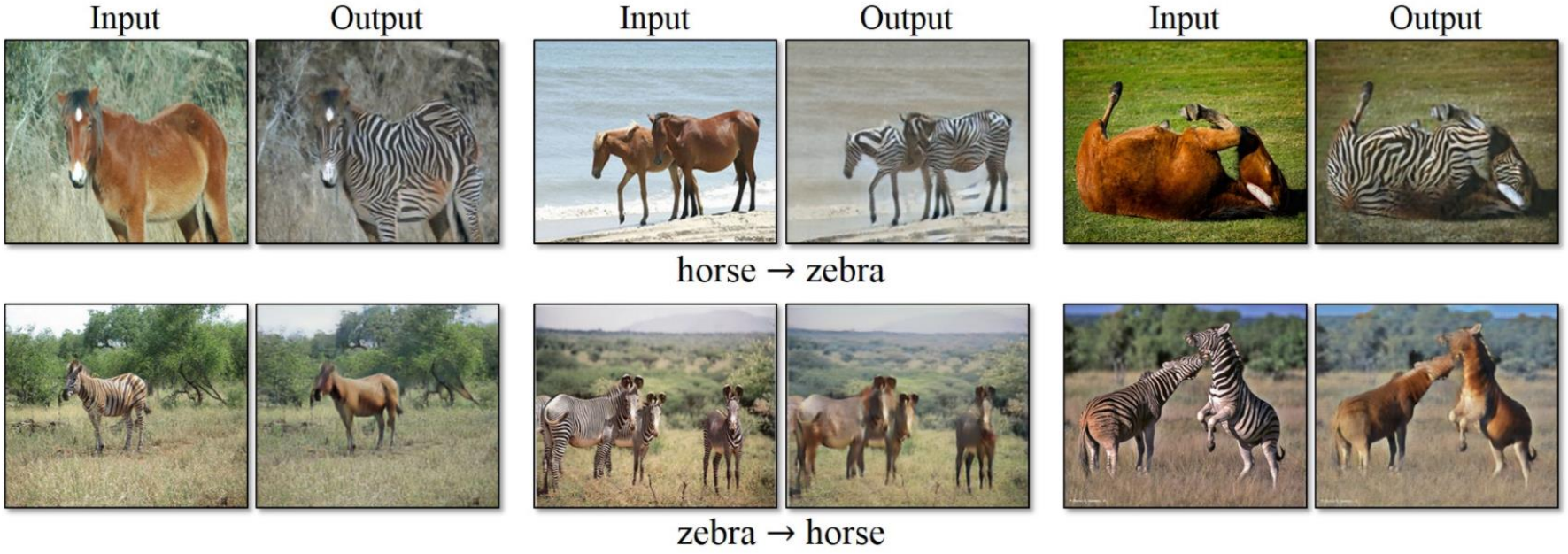
Source: https://indiaai.gov.in/article/news-information-and-politics-in-the-age-of-deepfakes

GAN based Manipulation of Facial Attributes/Expressions



StarGAN - Choi, Yunjey, et al. "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation." IEEE Conference on Computer Vision and Pattern Recognition (2017)

GAN based Image-Image Translation using CycleGAN

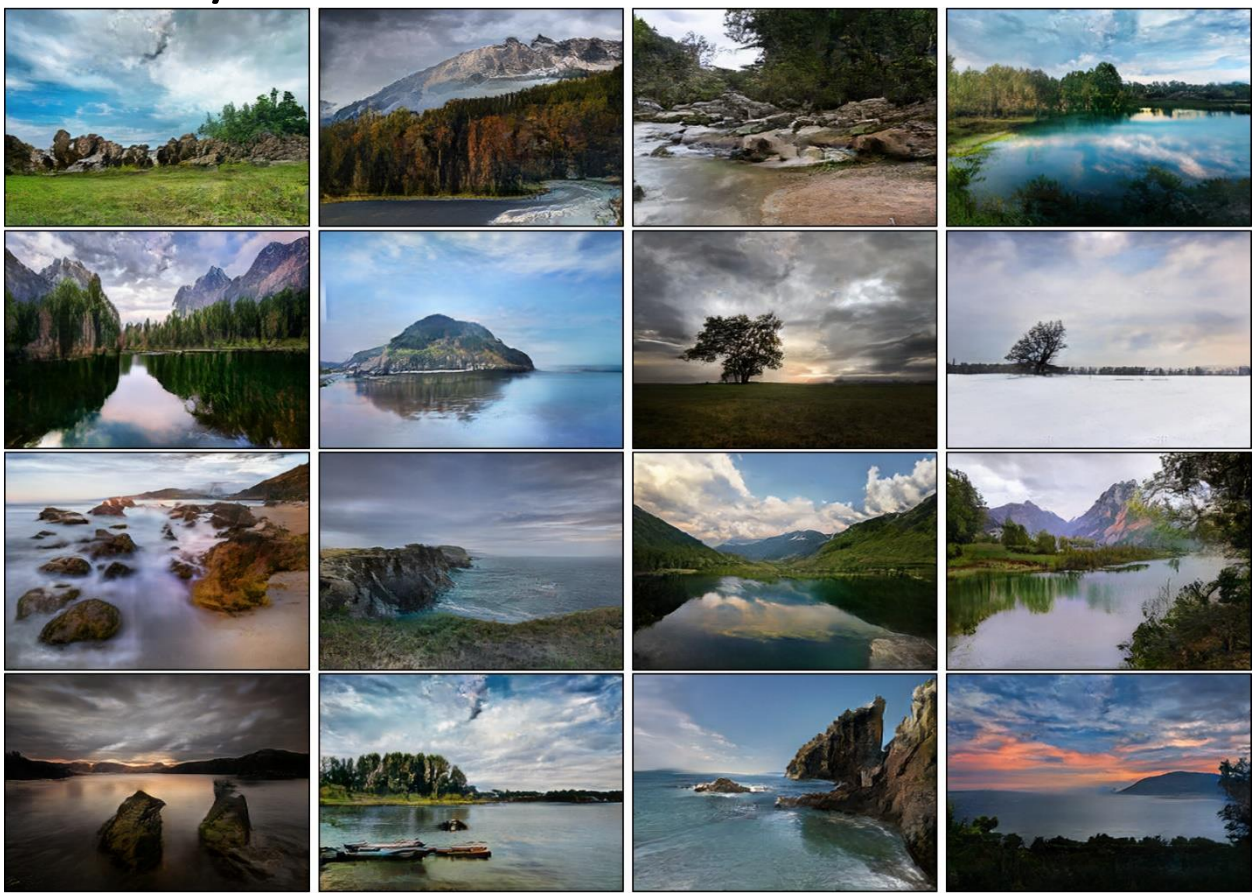


CycleGAN – Zhu, Jun-Yan, et al. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks." IEEE International Conference on Computer Vision (2017)

AI generated Faces using ProGAN

Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." *arXiv preprint arXiv:1710.10196* (2017).

AI Generated Natural Scenes using GauGAN/SPADE



Park, Taesung, et al. "Semantic image synthesis with spatially-adaptive normalization." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.
https://nvlabs.github.io/SPADE/

Generating High Quality Faces using StyleGAN/StyleGAN2

Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2019. https://github.com/NVlabs/stylegan

Karras, Tero, et al. "Analyzing and improving the image quality of stylegan." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020. https://github.com/NVlabs/stylegan2

Deepfakes and FaceSwaps

https://spectrum.ieee.org/tech-talk/artificial-intelligence/machine-learning/facebook-ai-launches-its-deepfake-detection-challenge

https://www.businessinsider.com/deepfake-tech-create-fictitious-faces-cats-airbnb-listings-2019-2

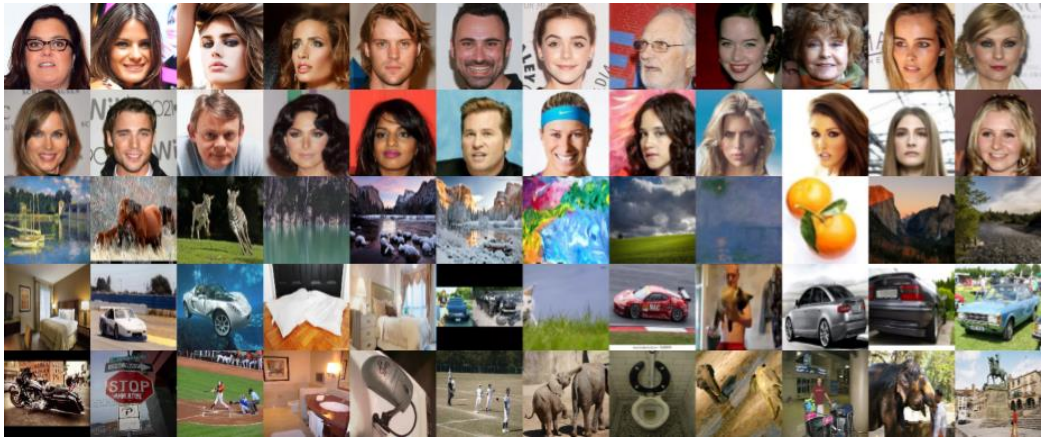
Deepfakes and FaceSwaps

https://medium.com/@jsoverson/from-zero-to-deepfake-310551e59aa3

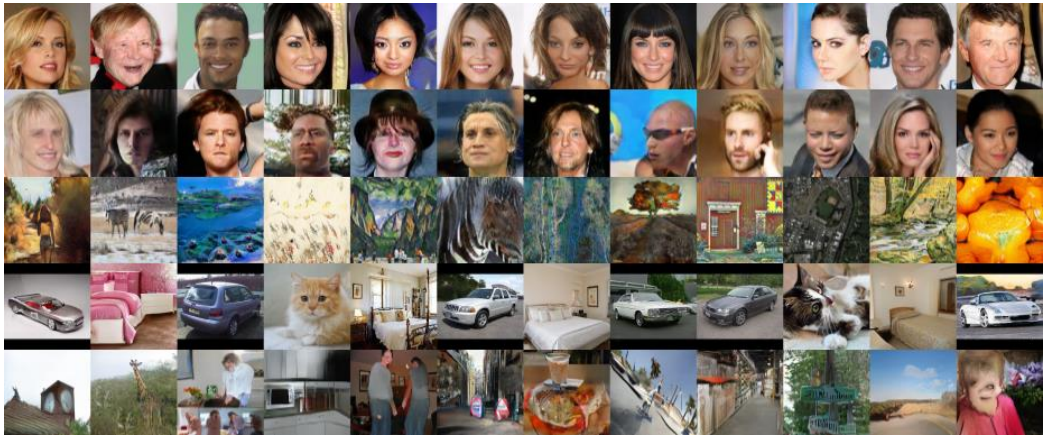
https://mashable.com/article/elon-musk-the-rock-photoshop-memes

Real Images and AI Generated Images

Real Images



AI Images



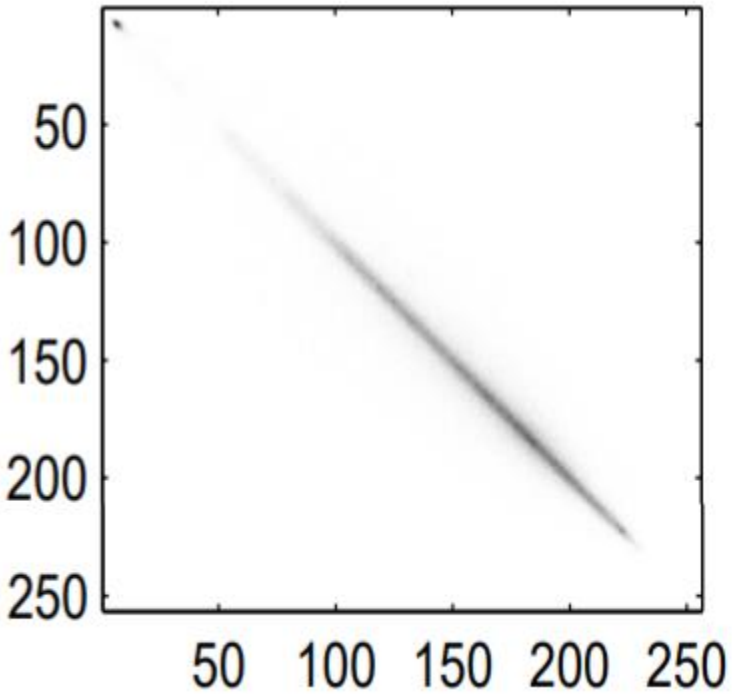
Detecting GAN Generated Images

- Though AI generated images are difficult for humans to detect, the *pixel level statistics* are altered

- Hence, features based on natural image statistics or steganalysis can be effective in detection

- One such feature is Pixel Co-occurrence Matrix

Example of a Pixel Co-Occurrence Matrix



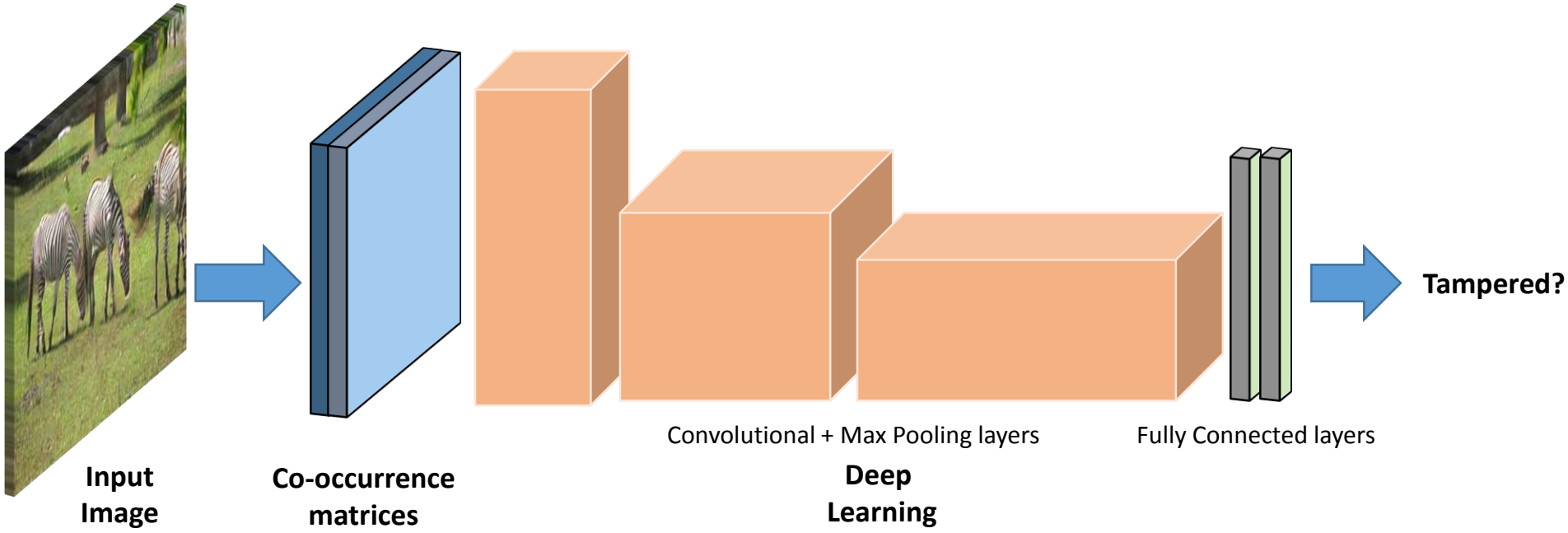
Source: https://vision.ece.ucsb.edu/sites/default/files/publications/05SPIEKen.pdf

Co-occurrence Matrix and Deep Learning

- Past methods computed hand-crafted features on Co-occurrence Matrices and then passed them through a machine learning classifier

- Here we pass the Co-occurrence matrices (computed on color channels) through a Convolutional Neural Network (CNN) framework and let the network extract the features

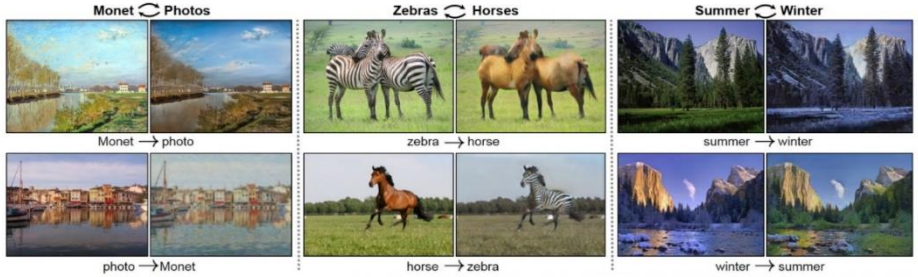
Co-occurrence Matrix and Deep Learning



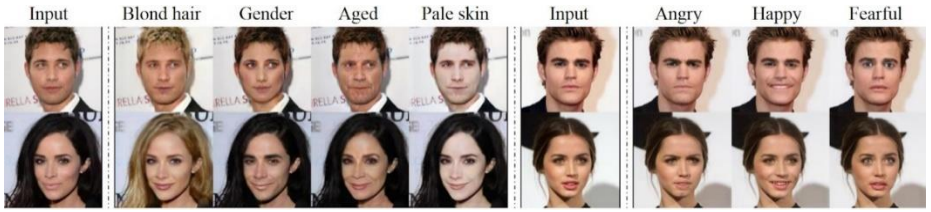
Preliminary Results

- Dataset
  - CycleGAN
  - StarGAN
- Experiment 1
  - 50% training
  - 25% validation/testing
  - CycleGAN Accuracy = 99.71%
  - StarGAN Accuracy = 99.37
- Experiment 2
  - Generalizability
  - Train on one and test on other

CycleGAN
(8,188 images)



StarGAN
(19,990 images)

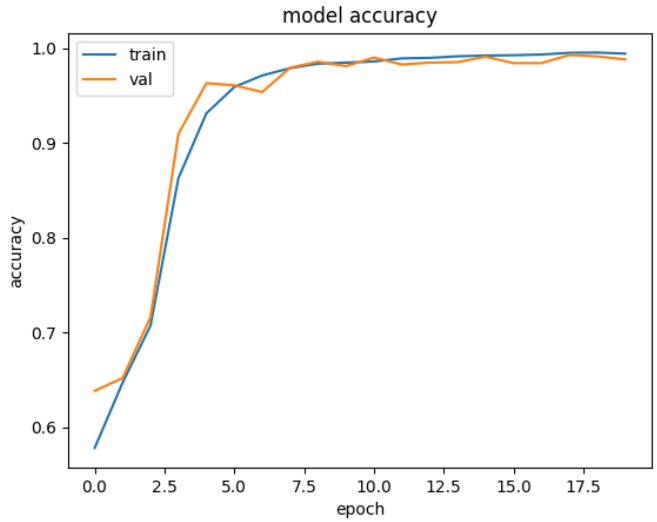


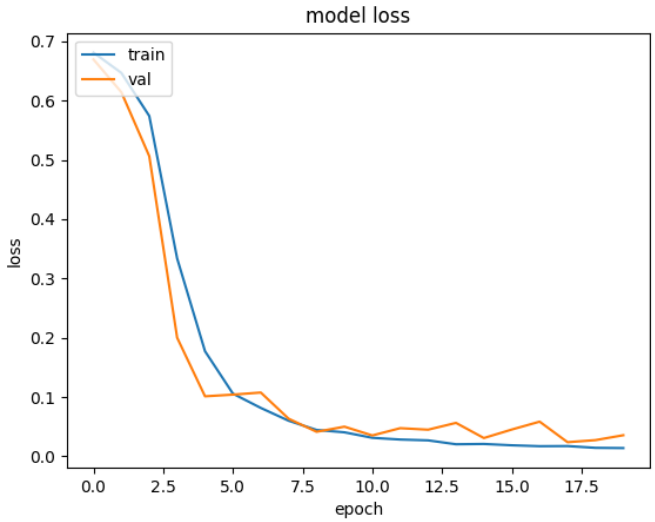
Training Dataset	Testing Dataset	Accuracy
CycleGAN	StarGAN	99.49
StarGAN	CycleGAN	93.42

Choi, Yunjey, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8789-8797. 2018.
CycleGAN – Zhu, Jun-Yan, et al. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks."
IEEE International Conference on Computer Vision (2017)

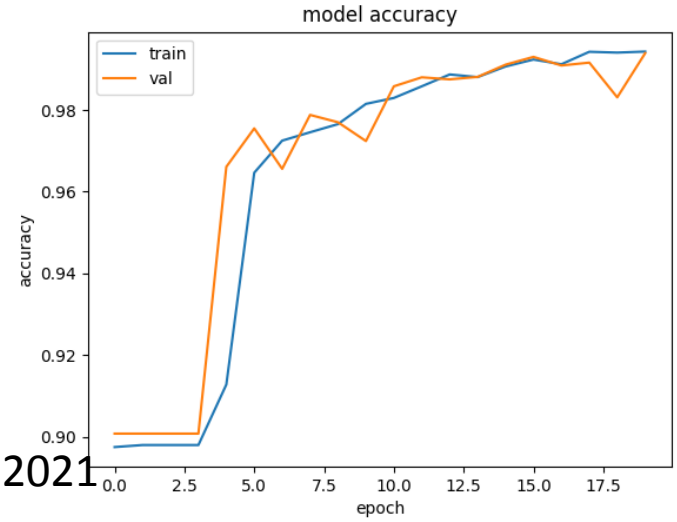
Training Accuracy and Loss

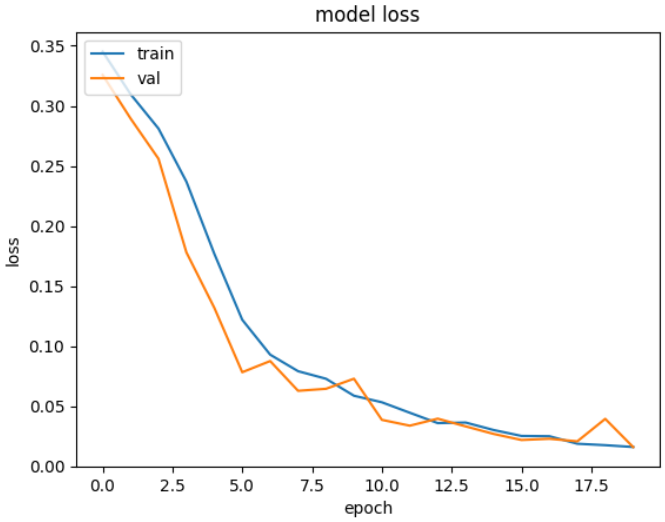
CycleGAN





StarGAN





December 7th, 2021

Benchmark Test on CycleGAN

- Benchmark Experiment
  - Different categories of CycleGAN dataset
  - Leave-one-category-out test

Method	Average Accuracy
Steganalysis features[1]	94.40
Cozzalino2017[1]	95.07
XceptionNet[1]	94.49
Nataraj2019[2]	**97.84**

[1]Marra, F., Gragnaniello, D., Cozzolino, D., & Verdoliva, L. (2018, April). Detection of gan-generated fake images over social networks. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)* (pp. 384-389). IEEE.

[2]Nataraj, L., Mohammed, T. M., Manjunath, B. S., Chandrasekaran, S., Flenner, A., Bappy, J. H., & Roy-Chowdhury, A. K. (2019). Detecting GAN generated fake images using co-occurrence matrices. *Electronic Imaging*, *2019*(5), 532-1.

Unified Framework for Detection, Attribution and Localization

- Detection
  - Is an image GAN generated or not?
- Attribution
  - Which GAN is it coming from?
- Localization
  - Which part of the image is GAN generated?

M. Goebel, L. Nataraj, T. Nanjundaswamy, T.M. Mohammed, S. Chandrasekaran ,B.S. Manjunath, “Detection, Attribution and Localization of GAN generated images”, Electronic Imaging 2021

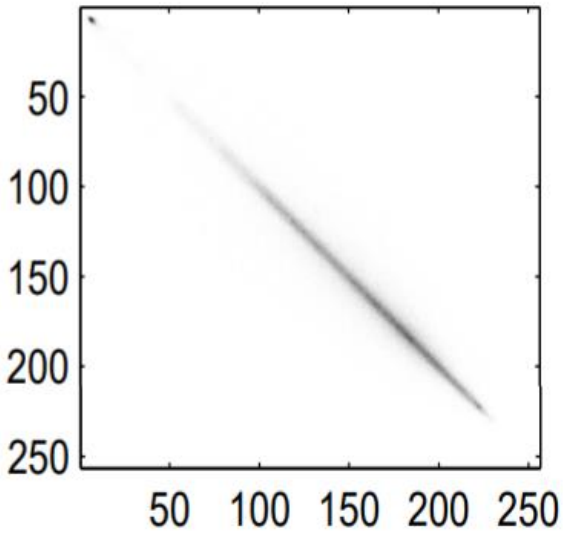
Co-Occurrence Matrix for GAN Detection

- Though GANs produce images that are difficult for humans to detect, the *pixel level statistics* are altered

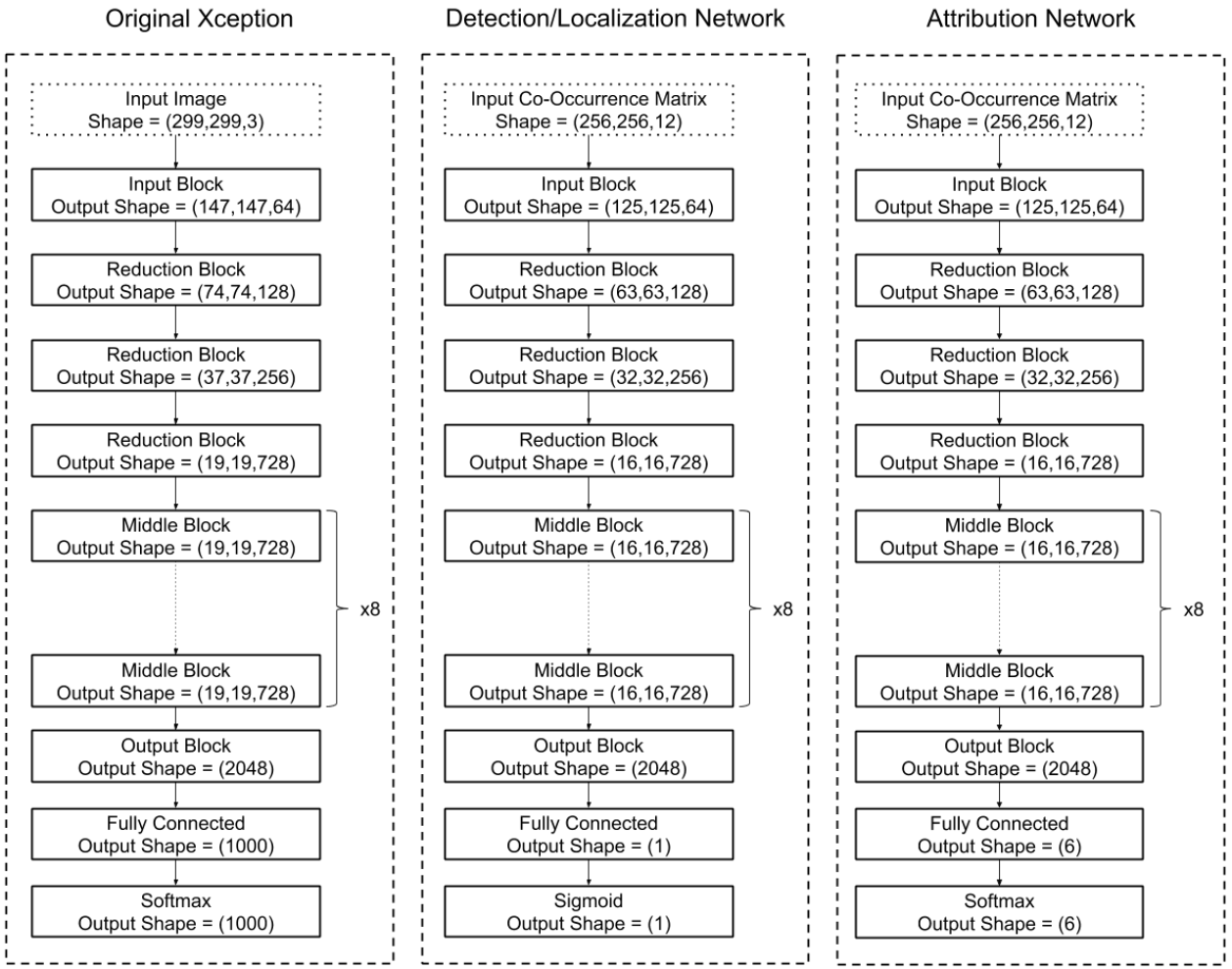
- Features based on natural image statistics or steganalysis can be effective in detection

- One such feature is Pixel Co-occurrence Matrix

- Use combination co-occurrence matrix and deep neural networks for Detection, Attribution and Localization



Framework for Detection, Attribution and Localization



Datasets – Large Scale Evaluation

- One of the largest evaluations on 2.6+ Million Images
  - 1.6M+ non-GAN images
  - 1M+ GAN images

- Datasets
  - ProGAN
  - StarGAN
  - CycleGAN
  - StyleGAN
  - SPADE/GauGAN
  - StyleGAN2 (testing)

	real	fake
base	**1696998**	**1073582**
├── stargan	3279	29511
│ └── celeba	3279	29511
├── cyclegan	18151	18151
│ ├── map2sat	1096	1096
│ ├── ukiyoe	1500	1500
│ ├── vangogh	1500	1500
│ ├── horse2zebra	2401	2401
│ ├── cezanne	1500	1500
│ ├── cityscapes	2975	2975
│ ├── apple2orange	2014	2014
│ ├── summer2winter	2193	2193
│ ├── monet	2572	2572
│ └── facades	400	400
├── progan	30000	74000
│ └── celeba_hq	30000	74000
├── spade	145497	145497
│ ├── ade20k	22210	22210
│ └── coco_stuff	123287	123287
└── stylegan	1500071	806423
├── bedroom_lsun	500000	278790
├── cat_lsun	500071	279867
└── car_lsun	500000	247766

Detection Experiments

- Experimental Setup
  - 90% training, 5% validation and 5% testing
  - XceptionNet
  - Adam optimizer and cross-entropy loss.
  - Batch size of 64

- Results
  - Accuracy = **99.16%**

Comparison with State of the Art

- Benchmark Experiment
  - Different categories of CycleGAN dataset
  - Leave-one-category-out test

Method	Average Accuracy
Steganalysis features[1]	94.40
Cozzalino2017[1]	95.07
XceptionNet[1]	94.49
Nataraj2019[2]	97.84
Zhang2019[3]	97.20
Proposed Method	**98.17**

[1]Marra, F., Gragnaniello, D., Cozzolino, D., & Verdoliva, L. (2018, April). Detection of gan-generated fake images over social networks. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)* (pp. 384-389). IEEE.

[2]Nataraj, L., Mohammed, T. M., Manjunath, B. S., Chandrasekaran, S., Flenner, A., Bappy, J. H., & Roy-Chowdhury, A. K. (2019). Detecting GAN generated fake images using co-occurrence matrices. *Electronic Imaging*, *2019*(5), 532-1.

[3]Zhang, X., Karaman, S., & Chang, S. F. (2019). Detecting and simulating artifacts in gan fake images. *arXiv preprint arXiv:1907.06515*.

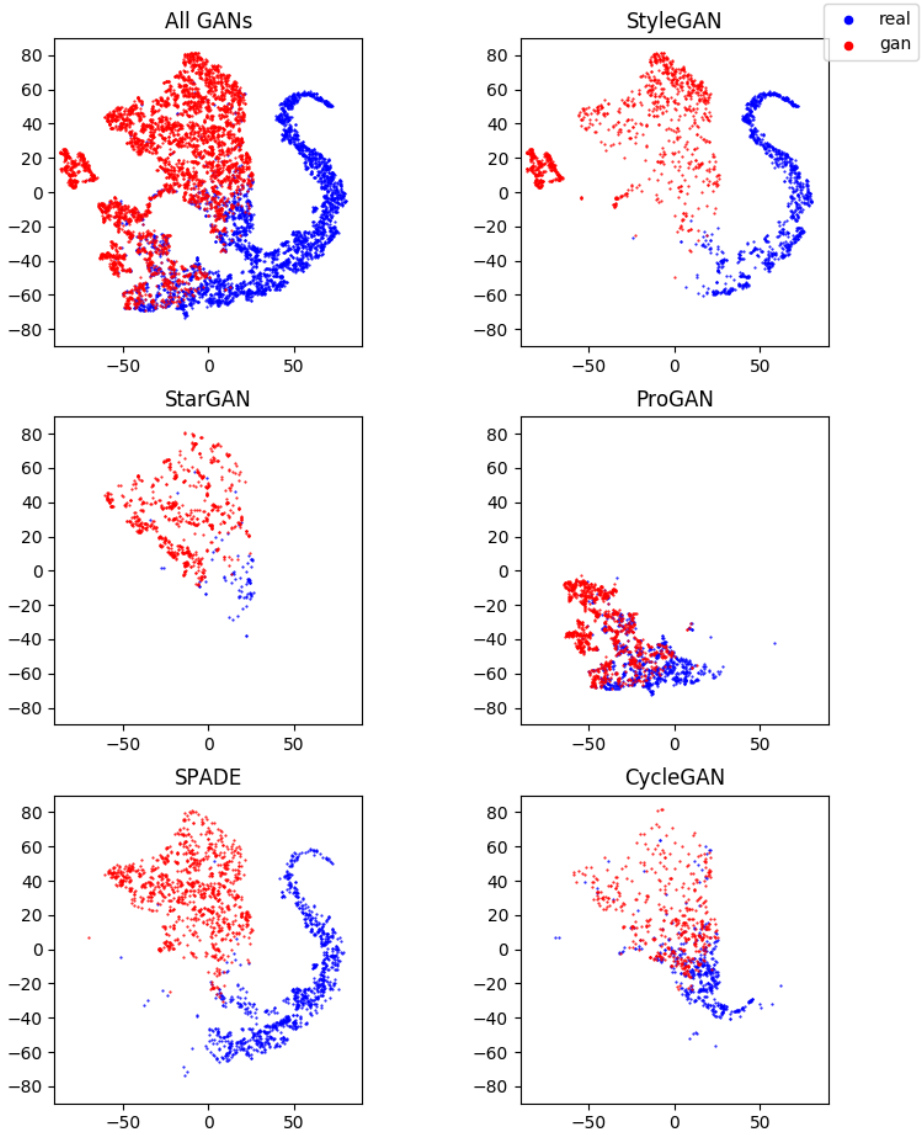
Which GANs are easily detectable?

- Leave-one-GAN-out setting
- Images patches from 4 GANs for Training, 1 GAN for Testing
- Images from which GANs need to be used for training

GAN	Testing Accuracy
StarGAN	84.90
CycleGAN	74.11
ProGAN	67.68
SPADE/GauGAN	98.74
StyleGAN	82.65

Visualization using t-SNE

- Under the Leave-one-GAN-out setting, 1,000 random images are considered
- t-SNE algorithm on the outputs of last layer of CNN
- StarGAN, Spade/GauGAN and StyleGAN – more separable
- CycleGAN, ProGAN – less separable



Results on StyleGAN2

- StyleGAN2 – more challenging and realistic than StyleGAN
- Without Fine-tuning
  - Randomly chose 100,000 StyleGAN2 images
  - Accuracy = **94.64%**
- With Fine-tuning
  - 100,000 non-GAN images from different datasets
  - 100,000 StyleGAN2 images
  - 40% training, 10% validation, 50% testing
  - Accuracy = **99.72%**

Source: https://github.com/NVlabs/stylegan2

GAN Attribution

- Given a test image, which GAN does it belong to?
- 6 class classification problem:
  - Non-GAN, StarGAN, CycleGAN, ProGAN, Spade/GauGAN and StyleGAN
- Dataset Distribution

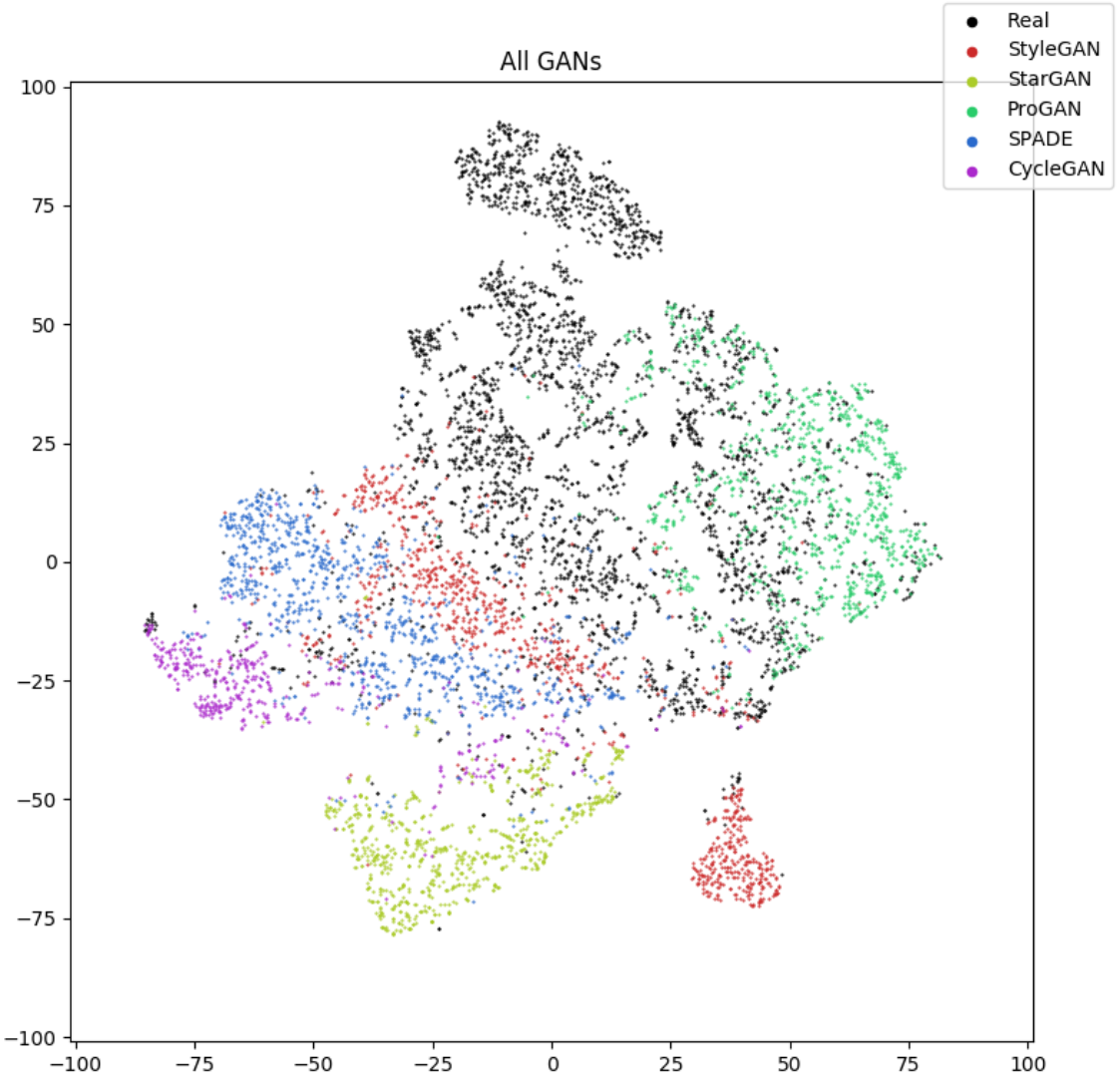
Dataset	Training	Validation	Testing
Non-GAN	1,612,202	42,382	42,397
StarGAN	28,062	738	711
CycleGAN	17,265	439	439
ProGAN	70,286	1,833	1,881
SPADE/GauGAN	138,075	3,717	3,704
StyleGAN	766,045	20,220	20,158

GAN Attribution – Confusion Matrix

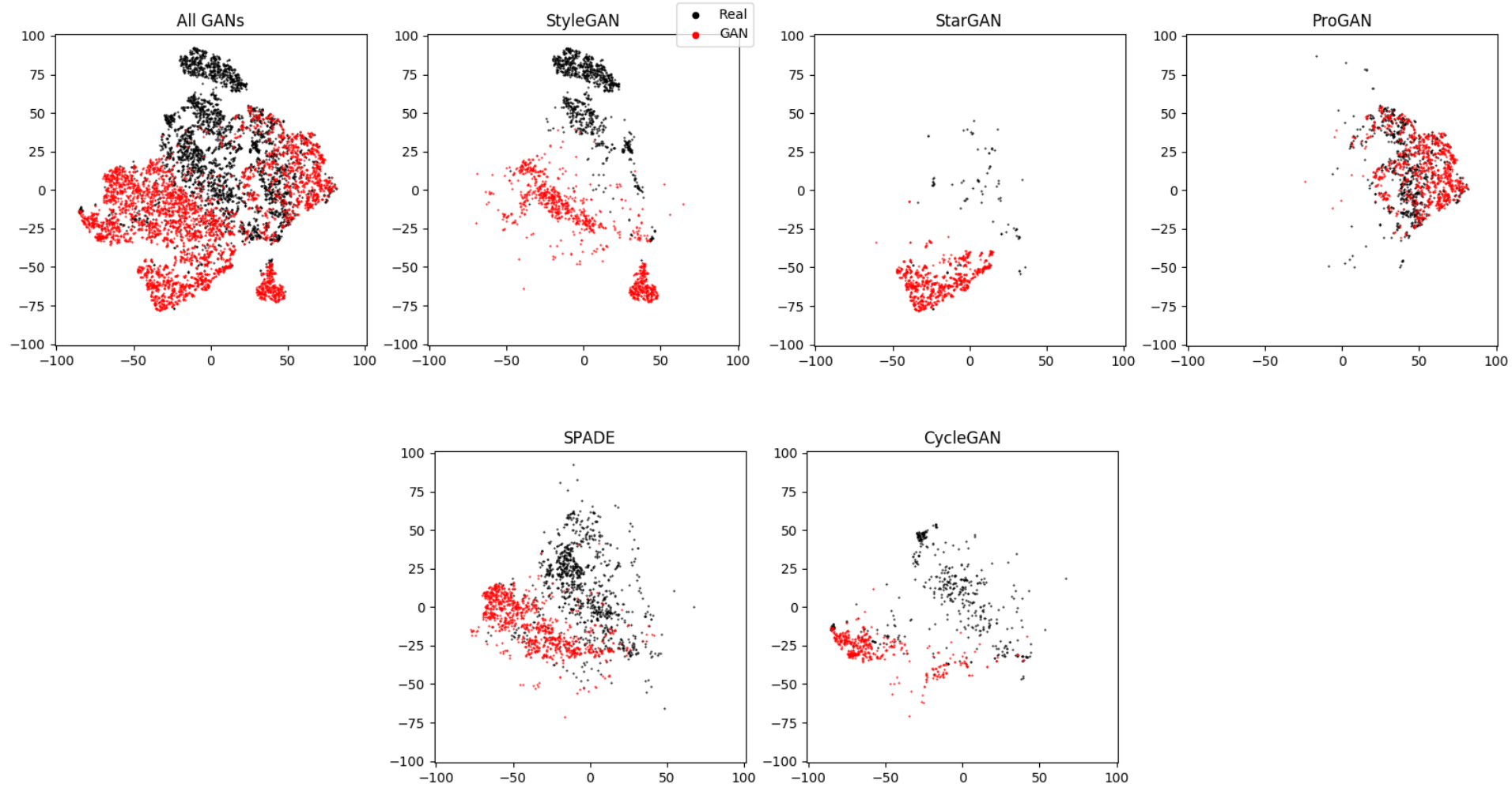
Overall Classification Accuracy = 96.54%

GT/Predicted						
	Non-GAN	StarGAN	CycleGAN	ProGAN	SPADE/ GauGAN	StyleGAN
Non-GAN	**97.5**	0.0	0.0	1.6	0.2	0.6
StarGAN	0.0	**97.6**	1.4	0.0	1.0	0.0
CycleGAN	0.0	0.0	**96.4**	0.0	3.6	0.0
ProGAN	0.0	0.0	0.0	**100.0**	0.0	0.0
SPADE/GauGAN	0.1	0.0	1.9	0.0	**97.5**	0.5
StyleGAN	0.7	0.0	2.2	0.0	6.8	**90.2**

Visualization using t-SNE algorithm



Visualization using t-SNE algorithm per GAN class

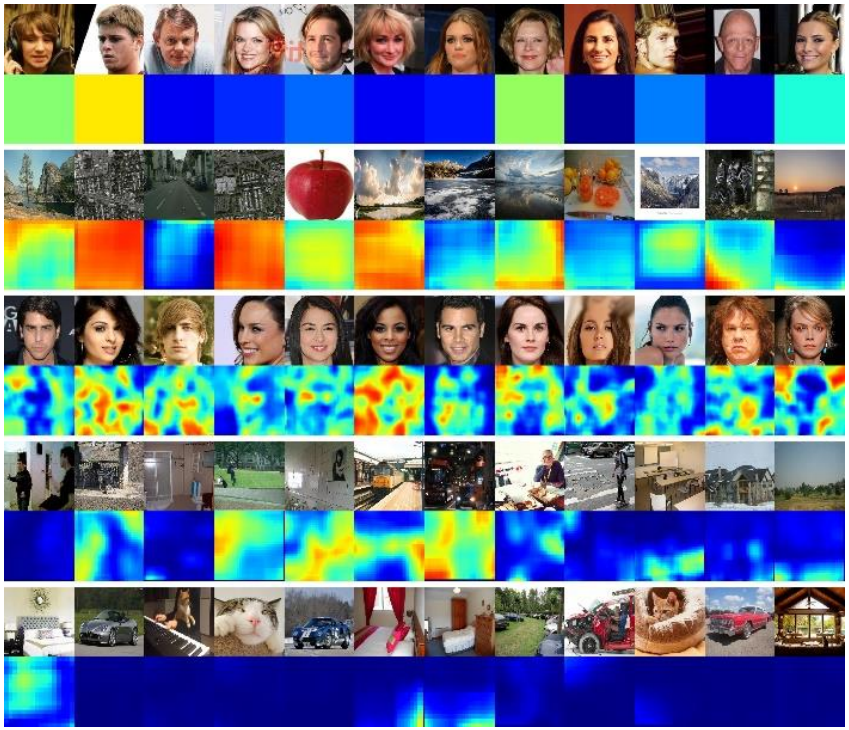


GAN Localization

- Localize which part of an image has been generated by GAN
- Training on image patches
- Divide an image into overlapping patches
  - Patch size: 128x128, Stride: 8
  - Xception network used for training

Localization Results

Results on Real Images



Results on GAN Images

Summary

- Presented a unified framework for Detection, Attribution and Localization of GAN generated images
- Based on pixel co-occurrence matrix that captures pixel level statistics
- One of the largest known evaluations on 2.6M+ images
- Achieves high accuracy and generalizable

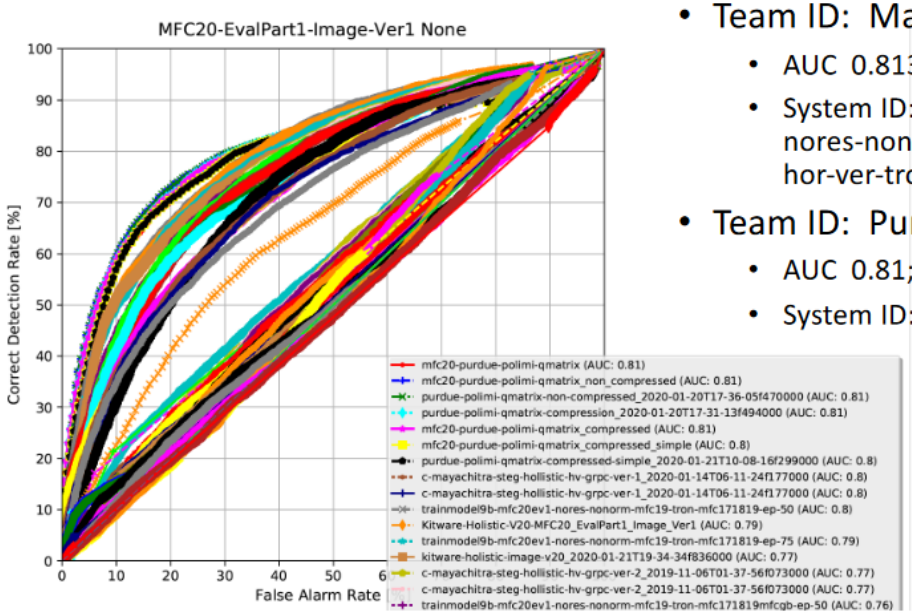
Extension to Image Manipulation Detection

- Compute Co-occurrence Matrices on Authentic images and Tampered Images and pass them to a Deep Learning classifier
- Use Media Forensics Challenge (MFC) Development (Dev) Datasets for Training and Evaluation (Eval) datasets for Testing
- Prior MFC Evaluation datasets are later added to the Dev datasets and models are re-trained
- Consistent high scores in MFC evaluations

Top score in MFC Eval 2020

Image Manipulation Detection Results: Full Data

- 20K probe images
- 12 teams:
  - Honeywell FIBBER
  - Kitware_Berkeley
  - Kitware_UAlbany
  - Kitware
  - Mayachitra
  - Purdue_Polimi
  - Purdue_TA11a
  - SRI-PRNU-TA1
  - UMD
  - USCISI-TA1.1
  - USCISI-TA1.2
  - USCISI
- 82 image detection systems as 04/09/2020.



- Highest AUC on full MFC20 EP1:
  - Team ID: Mayachitra
    - AUC 0.81384;
    - System ID: trainmodel9b-mfc20ev1-nores-nonorm-adam-def-apr2020-mfc19-hor-ver-tron-mfc171819-ep-40
  - Team ID: Purdue_Polimi
    - AUC 0.81; (CD@0.05FA = 0.436)
    - System ID: mfc20-purdue-polimi-qmatrix

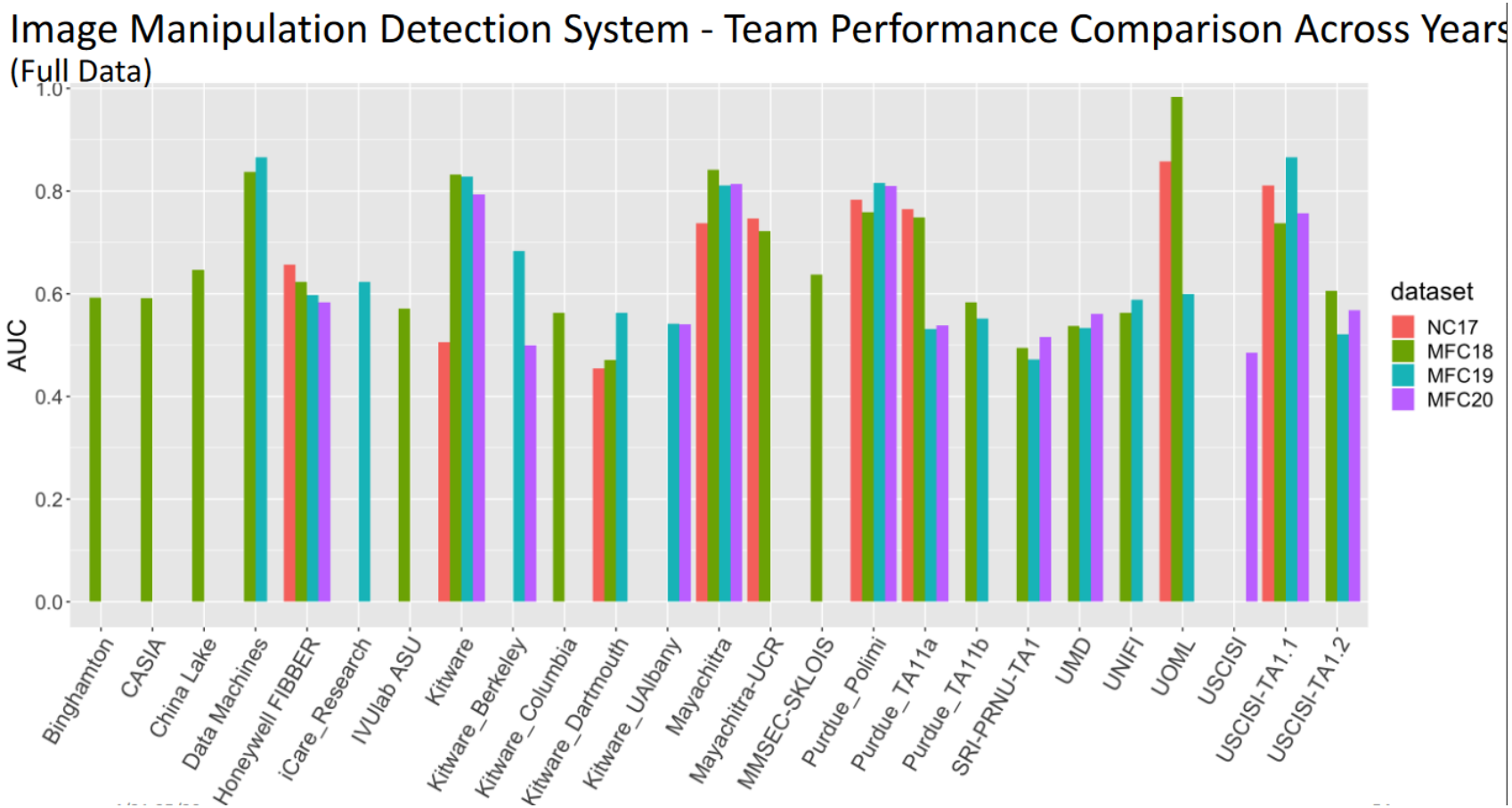
Figure: TA1 system MFC20 EP1, All probes (regardless of OptIn)

Source: https://mig.nist.gov/MFC/Web/PIMeeting2020/NIST_MFC20_PIMeeting_All_Final_formated.pdf

Results on Selective Manipulation Types

Manipulation Type	AUC-ROC
Splice	0.73
Clone	**0.82**
SpliceClone	0.73
Crop	0.78
Resize	**0.82**
Global Intensity Normalization	**0.94**
Intensity Change	**0.81**
Antiforensic-PRNU	**0.83**
Antiforensic-CFA	**0.81**
Social Media Laundering	**0.93**
Global Blurring	**0.80**
Local Blurring	**0.83**
GAN	**0.83**
Non-GAN CGI	**0.85**
Distortion	**0.89**

Consistency in Different Evaluations



Source: https://mig.nist.gov/MFC/Web/PIMeeting2020/NIST_MFC20_PIMeeting_All_Final_formated.pdf

Publications and Acknowledgements

- Bunk, Jason, Jawadul H. Bappy, Tajuddin Manhar Mohammed, Lakshmanan Nataraj, Arjuna Flenner, B. S. Manjunath, Shivkumar Chandrasekaran, Amit K. Roy-Chowdhury, and Lawrence Peterson. "Detection and Localization of Image Forgeries Using Resampling Features and Deep Learning." In *CVPR Workshops*. 2017.
- Nataraj, Lakshmanan, Chandrakanth Gudavalli, Tajuddin Manhar Mohammed, Shivkumar Chandrasekaran, and B. S. Manjunath. "Seam Carving Detection and Localization Using Two-Stage Deep Neural Networks." In *Machine Learning, Deep Learning and Computational Intelligence for Wireless Communication*, pp. 381-394. Springer, Singapore, 2021.
- Nataraj, Lakshmanan, Tajuddin Manhar Mohammed, B. S. Manjunath, Shivkumar Chandrasekaran, Arjuna Flenner, Jawadul H. Bappy, and Amit K. Roy-Chowdhury. "Detecting GAN generated fake images using co-occurrence matrices." *Electronic Imaging* 2019, no. 5 (2019): 532-1.
- Goebel, Michael, Lakshmanan Nataraj, Tejaswi Nanjundaswamy, Tajuddin Manhar Mohammed, Shivkumar Chandrasekaran, and B. S. Manjunath. "Detection, attribution and localization of gan generated images." *Electronic Imaging* 2021, no. 4 (2021): 276-1.

Acknowledgement: This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

Thank You

- Questions?