This project has received funding from the European Union's Horizon 2020 research and innovation programme under **grant agreement No 780355**

European Commission

FANDANGO

FANDANGO DELIVERABLE

Deliverable No.:	D4.3
Deliverable Title:	Copy-move detection on audio-visual content prototypes
Project Acronym:	Fandango
Project Full Title:	FAke News discovery and propagation from big Data and artificial inteliGence Operations
Grant Agreement No.:	780355
Work Package No.:	4
Work Package Name:	Fake news identifiers, machine learning and data analytics
Responsible Author(s):	CERTH
Date:	30.07.2019
Status:	V1.0 - Final
Deliverable type:	REPORT
Distribution:	PUBLIC

Revision History

Version	Date	Modified by	Comments
V0.1	20.05.2019	Theodoros Semertzidis	First draft
V0.2	14.06.2019	Efi Kafali	State of the art
V0.3	14.06.2019	Panagiotis Stalidis	Face forensics
V0.4	20.6.2019	Efi Kafali	QuadMani documentation
V0.5	10.07.2019	Theodoros Semertzidis	Full draft
V0.6	25.7.2019		Internal Review
V1.0	30.7.2019		Quality check

TABLE OF CONTENTS

List of Figures

List of Tables

Abbreviations

Abbreviation	Description
H2020	Horizon 2020
EC	European Commisiion
WP	Work Package
EU	European Union

1. Executive Summary

The following document is the deliverable regarding copy – move detection on audio – visual content prototypes, where we describe the state – of – the – art methods for the detection of forgery on images and/or videos. We emphasize that the attack types on audio and visual content that are nowadays used are more than a lot, and thus we underline the need for the detection of not only copy – move, but also other known type of forgery attacks on audio and visual content.

To begin with, we analyze the different types of attacks that may have been applied on an image, while we also describe the methods that are used so far for their detection. We then illustrate the functionalities of the state – of – the – art methods we consider are closely connected to the detection of fake images, while we also demonstrate qualitative results on images from publicly available datasets, to reveal each method's strong points. We consider it critical that after examining this document, the reader has figured out the differences between each method and the expected result on a manipulated image, thus we provide several examples accompanied by thorough explanations.

Based on the described state – of – the – art approaches, we also analyze our proposed method, a nonlinear segmentation model for boosted localization of fake image areas. We describe our implementation and demonstrate its strengths against a state – of – the – art linear convolutional segmentation model. We finally illustrate qualitative results, to better explain its usability.

2. INTRODUCTION

Copy - move is the term used for describing a common image forgery attack, where an image area is copied and pasted in a new position of the image. The most commonly used technique of copy - move forgery is the cloning of objects, but there is also the common case where a background image patch is copied and pasted, in order to hide objects present in the image (removal).

The basic concept of copy - move attacks can be summarized as follows: There are two types of objects involved, the source, i.e. the object that has been copied, and the target, i.e. the cloned object. In the simple case, the target object is just pasted in a new position of the image without having been postprocessed (translation transformation). But there are also cases that involve the postprocessing of the cloned object, such as blurring, rotation and scaling. Furthermore, there is a more complex transformation that is performed on the objects of interest, called occlusion, where the target object is pasted in a way so that it has some overlapping pixels with other objects of the image and as a result the image looks more realistic.

The logic behind copy - move forgery attacks can be revealed by thinking of a reason why someone would need to clone areas of an image. As an example, in 2013 North Korea has been accused of manipulating their military images. Specifically, before releasing the images to the press, they added some cloned hovercrafts, in order to show the world that there were more vehicles involved in that day's training exercise. This may seem as not such a major reason for the manipulation of these photos, but it also reveals that the release of images subjected to copy - move attacks can be done for any reason and as a consequence, it is a common strategy for propaganda spreading.

The frequence and the reasons that result to the release of such photos, raise the need for the detection of copy - move forgery attacks in images. Copy - Move Forgery Detection (CMFD) is a task that aims both at the detection of manipulated images, but also at the localization of the target (cloned) objects. This task is not always easy, since the object that is being cloned comes from the same image and may have similar statistical values with the other existing objects.

Besides clean copy - move manipulation attacks, there is always the case where an alien object is inserted to an image. This manipulation technique is referred as splicing and is also a common technique used for deceiving the viewers, by adding objects that were not originally present in the image. Furthermore, the past years, image manipulation software became more popular and as a consequence the tampering of an image or video has become an easier task. This kind of software is used for plenty of image forgery attacks, such as cloning or splicing, image blurring, contrast enhancement, scaling of objects, manipulation or isolation of facial expressions, face swapping, intentional misinterpretation or isolation of speech in videos, etc.

In the context of this task we provide services for the detection of manipulation attacks on both images and videos. We argue that the key point to an effective manipulation detection tool in the aspects of audio and visual content is its capability of identifying as many different manipulation attacks as possible. That said, we consider it critical to go beyond the detection of only copy - move attacks, since there are no silver bullets on the manipulation software or the type of manipulation methods the attackers may use. In fact, these factors can vary, depending on the

kind of the information someone may need to conceal or emphasize at each time. Thus, we built the image and video services as an ensemble of five models, aiming to cover the variety of known media manipulation attacks and consider them together as a robust tool in the hands of the end user.

All the media services await for a POST request with a JSON object containing some image/video URL and return a JSON object of the form:

{"analyzer": <name>, “mask”: <base64 encoded manipulation mask>, "trustworthiness": <score>}

where the field “analyzer” is the name of the model that made the specific prediction, “mask” is a base64 encoded manipulation mask of the given image/video, illustrating the detected manipulated area on the image and the “trustworthiness” field is a score in range [0,100], indicating the probability of the given content to be valid, where 0 shows a high confidence regarding the content’ s fakeness, while 100 a high confidence regarding the content’ s trustworthiness.

Since this task is dealing with audio and visual content, we consider it critical that the outputs of the analyzers should be quite self - explanatory and easy to interpret. We value the already hard end users’ decision making process, and thus we extend our outputs to not only return a score, but also qualitative predictions on the audio and visual content, in order to simplify their further analysis.

As an example use case, the end user may enter the URL of an image he/she suspects is manipulated into the FANDANGO interface. The image will be sent to our classifiers, which will analyze it by localizing a potential manipulated area and computing the trustworthiness score. In details, each one of the analyzers will produce a mask for the image, which will localize the detected fake object/area, if any. In the case the analyzer does not find any potential manipulated area, the mask will be an one - color image, where there will be no distinguished fake areas. Each one of the analyzers will also compute a trustworthiness score for the given image. As a result the end - user will receive the predicted masks and the aggregated trustworthiness score, that is computed by taking into consideration the scores that each one of the analyzers predicted. By comparing the different manipulation masks and also examining the aggregated trustworthiness, he/she may make his/her decision on whether the image is manipulated, by prioritizing the inspection starting with the predicted manipulated areas.

In the next chapters we provide descriptions of the state - of - the - art models that we find closely connected to the detection of manipulation attacks on images and videos. We illustrate the strengths and weaknesses for each one of the described models and clarify the use cases where we expect them to behave well. We also present a qualitative analysis by comparing predictions of the image manipulation detection models on the same images, to further explain the adequacy of each technique on different attacks. Finally, we describe our novel method for boosting segmentation models, by applying nonlinear convolution filters.

3. ACADEMIC LITERATURE REVIEW AND RELATED PRODUCTS

The first attempts of copy - move detection on images, were focused on manipulation cases regarding only cloned regions of an image [5]. The main idea behind CMFD is the detection and localization of a target object, that has been cloned from another source object in the same image. At the early stages, CMFD was about the target object detection, but later got also connected with other types of manipulation attacks.

The researchers in [6] used the Fourier - Mellin Transform (FMT) to extract significant features from overlapping image blocks, and proposed it as a tool for the detection of JPEG compression and other well known attacks, such as translation and scaling. Similarly, Cozzolino et al. used the PatchMatch algorithm [7] to achieve better results against scaling and rotation attacks, by considering the similarity between neighbouring image patches.

In [8] the researchers used Fast Fourier Transform (FFT), Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) to extract features and detect copy - move forgery on images that were under various attacks, such as Gaussian noise and JPEG compression. For the detection of an object that has been both cloned and rotated, the researchers in [9] proposed the use of Zernike moments for the localization of the cloned image region, introducing additional robustness on both the aforementioned attacks, but also in rotation transformations. Similar methods were proposed by the researchers in [10] and [11], where they used blur moment invariants and PCA accordingly, to detect forged image areas that have been manipulated in some common attack technique, such as noise adding, blurring or contrast enhancing. In [12], [13] and [14] the researchers utilized SIFT keypoint - based methods for locating and recovering the transformation performed, while in [15], [16] and [17] SURF keypoint - based methods were used for the detection of geometric attacks. In [18] Zhu et al. proposed the detection of tampered images by using ORB, a more efficient algorithm than SURF and SIFT. Another keypoint-based method was used in [19], where the authors extracted triangles instead of image blocks and subjected them under similar triangle matching. Finally, the authors of [20] and [21] proposed different keypoint - based methods, where before the keypoint extraction, image segmentation in independent patches was performed, in order to detect copy - move manipulations based on the extracted patches.

Recently, some Deep Learning models have been proposed for the detection of tampered areas in images. In [22] a CNN is implemented for feature extraction from tampered images, while in [23] the researchers use a Deep LSTM for fake image classification on a patch level. In [24] the authors propose an end - to end solution, the Deep Matching and Validation Network, for both the detection and localization of tampered areas, by computing the probability of an image to be the donor of a tampered image. In [25] a two - stream model was proposed, for the detection of tampered faces in images, while in [1] the researchers built BusterNet for the detection, localization and discrimination of source and target objects. Recently, in [28] the same researchers proposed ManTra - Net, a model consisting of a submodel for synthesizing the feature maps for the manipulated areas, and a submodel for the detection of image local anomalies. Finally, the most recent research dealing with manipulated photos is in [31], where the researchers scripted Photoshop to generate fake faces photos. They approached the

detection of manipulated facial features with a CNN and also managed to reconstruct the manipulated features to their original form.

4. PUBLICLY AVAILABLE DATASETS

Copy - move detection is a relatively new field and as a result, copy - move detection models can be trained and evaluated only on a few good enough and openly available datasets. The most commonly used datasets are CASIA, CoMoFoD and the 1st Image Forensics Challenge dataset. Additionally, regarding the training, it is a common policy that researchers often create their own synthetic datasets, in order to have a larger number of training samples available and increase their models' generalization on unknown data.

However, the creation of a synthetic copy - move dataset is a difficult task. A solid and balanced dataset should consist of a lot of samples, with each one of them being as realistic as possible, so that the identification of the fake (target) object is a difficult task even for humans. Most of the times this does not happen. Synthetic datasets end up to be a collection of a lot - but not qualitative enough - samples, that result to great results on the testing samples of the created dataset, but when it comes to real world manipulated images, chances are that the model will result to several inaccurate predictions.

Due to the limited size of CoMoFoD, CASIA and the 1st Image Forensics Challenge dataset, recently researchers also started using more abstract datasets coming from image hosting services, like Reddit and Flickr, consisting of photoshopped, or in some way manipulated images. Since the available kinds of image processing attacks and image manipulation softwares are nowadays more than a lot, the utilization of more realistic datasets coming from the web, can enhance the credibility of a learning algorithm. In addition, it is surely a good idea to use data that are more similar to the creations of today's photoshop artists, since the possibility that the algorithm will perform well on actual manipulated images is increasing.

4.1. CASIA

CASIA TIDEv1.0 and CASIA TIDEv.2.0 [35] are the most standard datasets used for copy - move detection algorithms. CASIA TIDEv.2.0 [1] is the largest image manipulation benchmark that is publicly available, containing 7491 pristine and 5123 tampered RGB images. Tampered images are manipulated in different ways, it is not specified though which of them are manipulated in a copy - move manner. Additionally, CASIA does not provide binary masks for the tampered images, so the tampered region has to be located by computing the difference between the tampered image and its pristine counterpart.

1 http://forensics.idealtest.org/casiav2

4.2. CoMoFoD

CoMoFoD [36] is another standard copy - move dataset that is broadly used. The CoMoFoD small database [2] that is publicly available, contains 5000 RGB images, generated by adding manipulations on 200 base image categories. Every image set consists of the original image, a colored and a binary mask and a tampered version of the original image. Some post processing attacks are also performed on the pristine and the tampered images, such as JPEG compression, noise adding, blurring, brightness change, color reduction and contrast adjustments. CoMoFoD does not provide binary masks distinguishing the target object, but only binary masks of both source and target objects.

4.3. 1st Image Forensics Challenge

This dataset was created by performing splicing and copy - move operations on pristine images for the purposes of the 1st Image Forensics Challenge [3]. Most of the available images are high resolution and shot by cameras, without any beautification post processing techniques being present. The publicly available dataset contains 1050 pristine and 450 fake images, where the binary masks of the fake (target) objects are also available.

4.4. Reddit Photoshop Battles

Reddit Photoshop Battles is a subreddit (Reddit community) where users are free to post photoshopped images, created by manipulation software. It is a quite popular subreddit, where official weekly photoshop battles take place. For a photoshop battle to begin, an original image is posted into the subreddit and users post their manipulated versions in the comments section.

Recently, some researchers gathered and organized 11142 non manipulated images, along with all their available manipulated versions, posted by users, and created the PSBattles dataset [4] [2]. By all means, in a community like Reddit it cannot be guaranteed that the images starting a Photoshop battle are actually not already manipulated, despite the short description given for each image, which sometimes gives details about the image source. The most advantageous feature of Photoshop Battles dataset, regarding its utility in manipulation detection, is the fact that it is completely diverse, since the manipulated images come from multiple different artists.

Furthermore, the Reddit Photoshop Battles Image Provenance Dataset is also publicly available [5] [34]. Image provenance is a term used to describe the discovering of fake images that are related to an original image. This relationship between images can be represented as a graph, where its nodes are the different images and the edges are showing the connections between images, if any. The graph of this dataset is accompanied by a JSON file which has information about the relationship between an original image and its manipulated versions.

---

2 http://www.vcl.fer.hr/comofod/

3 http://ifc.recod.ic.unicamp.br/fc.submission/

4 https://github.com/dbisUnibas/PS-Battles

5 https://github.com/CVRL/Reddit_Provenance_Datasets

4.5. CERTH SYNTHETIC DATASET

We have created a synthetic dataset based on MS - COCO clean samples, in order to create forged images containing spliced areas. In details, we have defined two object classes, a source and a target class. We isolated an object from the source class by copying/extracting it from the clean MS - COCO sample using the provided segmentation mask and then pasting it to a new clean MS - COCO sample, containing an object of the target class. We did not apply any rotation, scaling or other known types of attacks to the spliced objects, but we have introduced occlusion to our dataset, to make our synthetic samples look more realistic. Thus, we pasted the object from the source class to the image of the target class so that the source and target objects had overlapping pixels with each other, with the source object being in the background, while the target object in the front. As a result, a part of the spliced object is occluded by the target object. We have totally created 42500 forged samples with spliced and occluded objects that we used in the development phase of our modules.

5. Deep Learning Models For Copy Move Detection

In this chapter we describe BusterNet, a state - of - the - art segmentation model trained on clean copy - move manipulated images and ManTra - Net, the most recent research product for the detection of manipulated images including, among many others, the detection of copy - move manipulation attacks.

5.1. BusterNet

The state of the art results for copy - move detection on images, come from an end - to - end Deep Learning model, BusterNet, that was published for The European Conference on Computer Vision (ECCV) [1]. The novelty of BusterNet in the field of copy - move detection, relies on fact that, most of the times, the predicted image masks are distinguishing the source and target objects, unlike other copy - move techniques, that resulted in mixed source-target, or only target visualizations.

BusterNet's architecture consists of two branches, with each one of them being responsible for a specific task. The Manipulation Detection Branch is used for localizing manipulated areas of an image, and returns as output a binary manipulation mask of the fake (target) object. The Similarity Detection Branch, on the other hand, is responsible for analyzing similarity between image pixels via self - correlation, in order to localize similar areas and returns as output a similarity binary mask, showing both source and target objects, but not differentiated from each other. A fusion module is then used for considering the outcomes of the two branches jointly, and predicting a three - class mask of the input image, showing differentiated source (green), target (red) and background (blue) pixels.

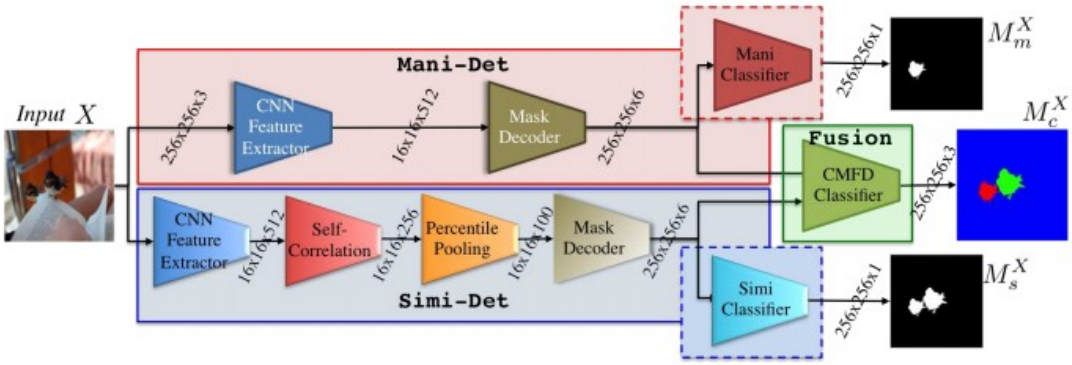


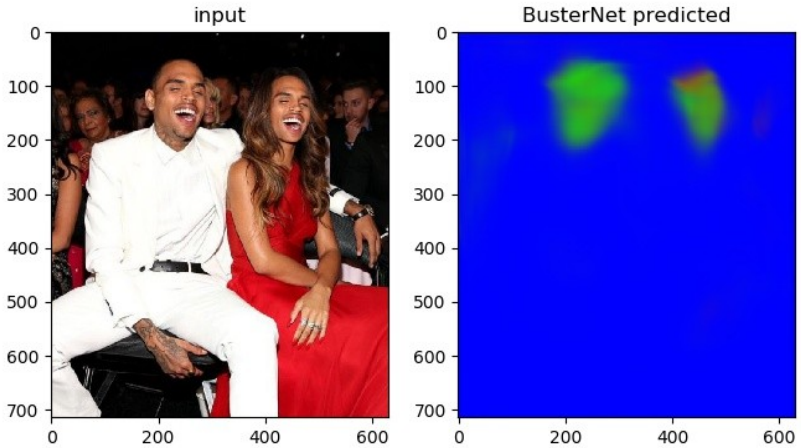
Figure 1 The Architecture of BusterNet [1]. The blue area of the resulting 3 - class mask M_c^X is the predicted image background (pristine pixels). The green pixels show the predicted source object, while the red pixels show the predicted target (cloned/manipulated) object.

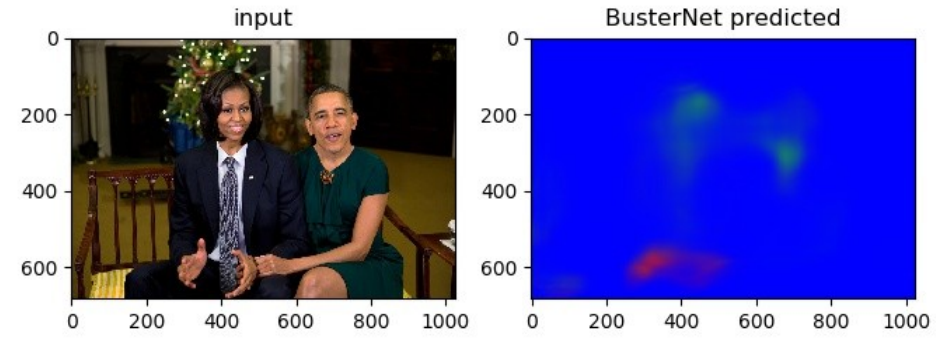
Both the Mani - Det and the Simi - Det branches use a CNN for extracting features from the input image. The researchers have used the first layers of the VGG16 architecture, but other Convolutional Neural Networks can serve as a feature extractor too, which sets BusterNet suitable for further experiments and extensions.

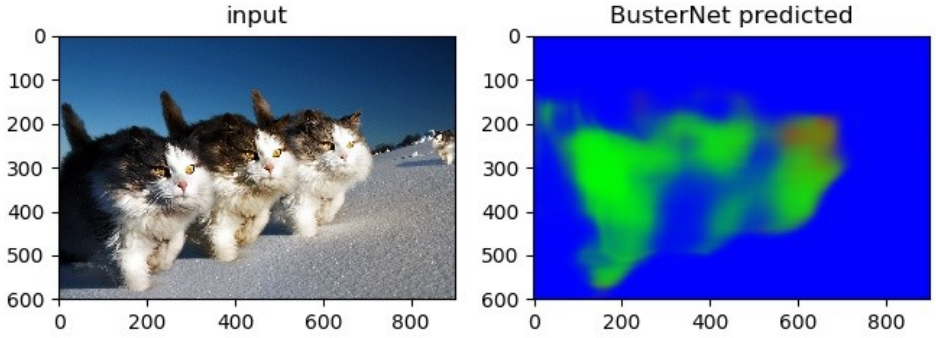
Since all the available datasets for copy - move detection are significantly small, the researchers synthesized their own dataset. Based on the MS - COCO and MIT SUN2012, they created the USCISI-CMFD dataset, consisting of 100000 samples subjected to copy - move attack, along with their three - class masks distinguishing source from target objects. The researchers followed a three - stage training: 1) Training the two branches on their tasks independently, 2) Freeze the two branches and train the fusion module, 3) Unfreeze the network and train BusterNet end - to - end. The USCISI-CMFD dataset was used for all the aforementioned training steps.

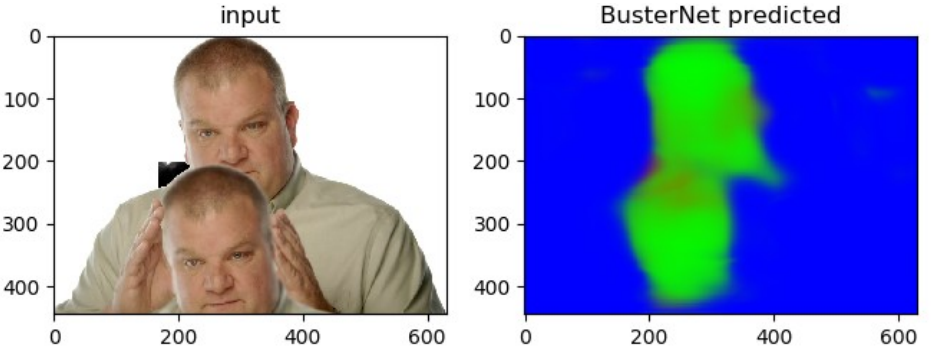
BusterNet takes as input an RGB image and returns the predicted copy - move mask. A prediction of BusterNet is a three - class mask, where blue color represents the area that the model classified as background pixels, green is for the detected source object, while red is for the object that the model classified as target, or cloned object. Obviously, the colors of the predicted masks are not always well distinguished. There are times that one of the two branches is more confident and as a result the mask may not always contain all three colors. In details, in cases that the Mani - Det's prediction is more confident, it means that the model has found manipulated areas in the image, but was not capable of predicting both source and target objects. These masks localize the possibly fake detected pixels, but their color is more red - like, since masks with target objects were used as labels to train this branch. On the other hand, when Simi - Det is more confident about its predictions, the resulting mask includes areas where the pixels are more green - like, meaning that BusterNet managed to find similar objects in the image, besides the background pixels. The masks that show bold green or red areas should be interpreted as a confident prediction of the model. A prediction with well distinguished and bold pixels should be further analyzed, even in the case it does not include both green or red areas, since the final mask has weighted impact from both branches, but also from the fusion module. Less bold predictions show cases that an image is less possible to be manipulated. Finally, there is the case that the model classifies all (or most of) the pixels as background pixels. This can be translated as a low possibility that the input image is fake.

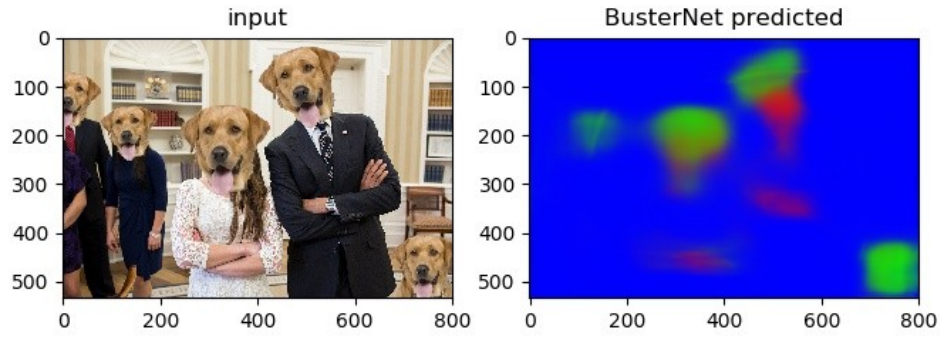
For a better illustration regarding the meaning of BusterNet's predictions, we demonstrate some results we acquired by applying the pretrained BusterNet on samples from the Photoshop Battles subreddit. No original versions of the images are included, but in most of the cases, it is quite easy for the human eye to distinguish the manipulated areas.




input
BusterNet predicted


input
BusterNet predicted


input
BusterNet predicted


input
BusterNet predicted

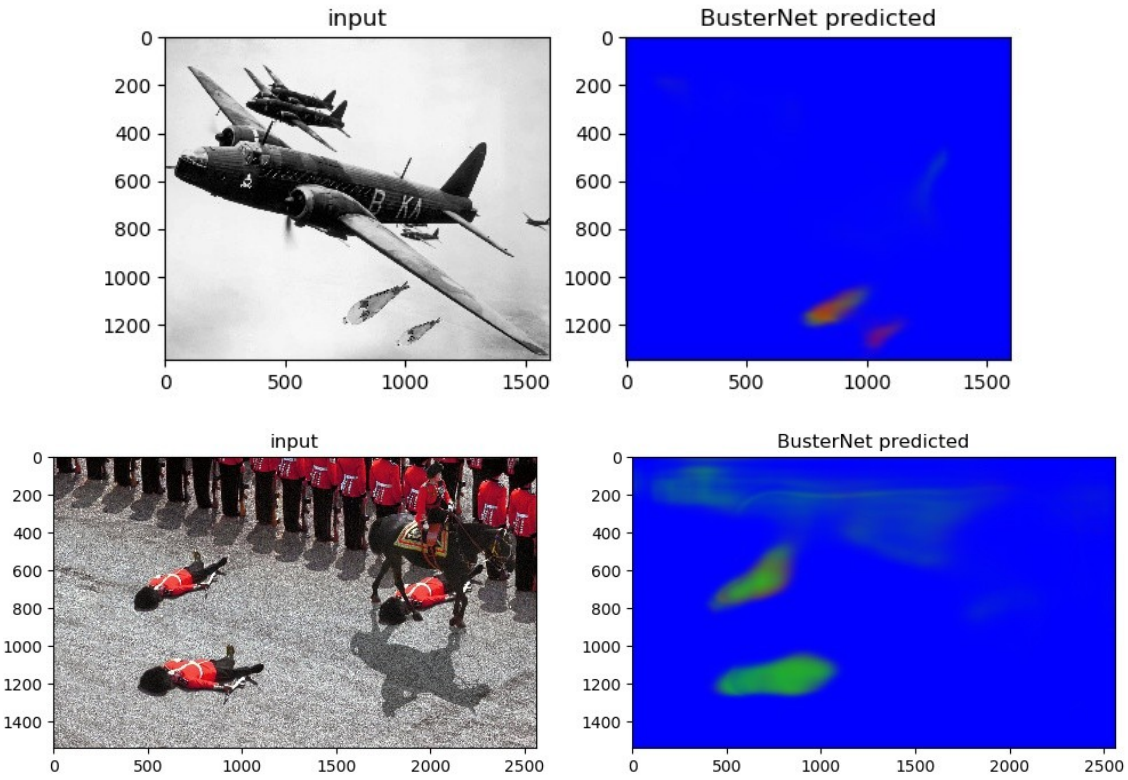


Figure 2 Qualitative results - pretrained BusterNet [1]

5.2. MANTRA - NET

ManTra - Net [28] is a very recent research product proposed for the detection and localization of a variety of image manipulation attacks. ManTra - Net's architecture consists of two different submodels: The Manipulation Tracing Network, that is responsible for the detection of image manipulation trace feature, and the Local Anomaly Detection Network, with the expertise to localize anomalies on the images.

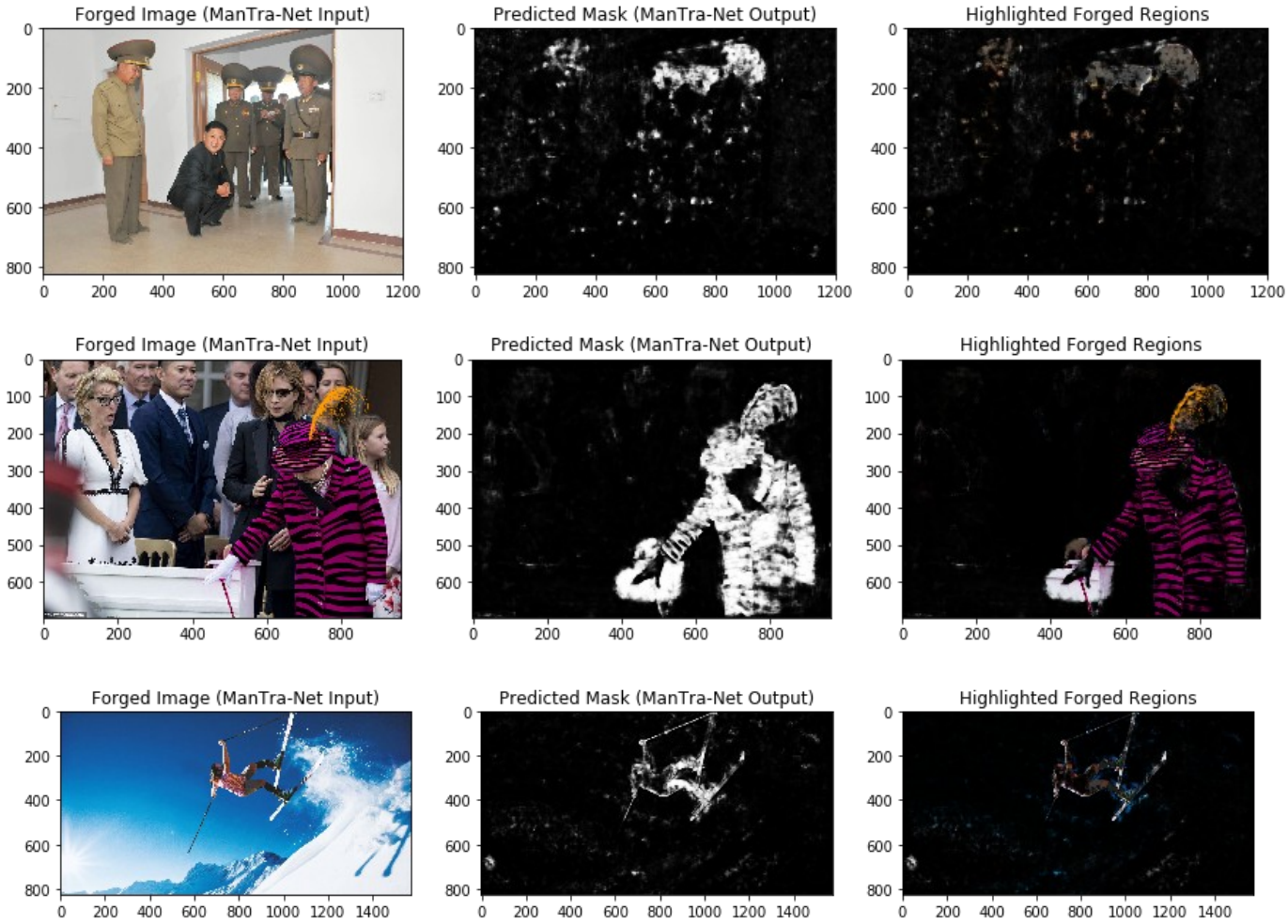
The product of this research introduces a model able to capture a wider set of known image manipulation attacks. The researchers evaluate their proposed method on 385 different known attacks and conclude that ManTra - Net can also encode more complex kinds of attacks, such as manipulations created by Deep Neural Networks. The goal of ManTra - Net is to learn mappings for discovering the difference between a forgery label and its local anomaly on a feature level.

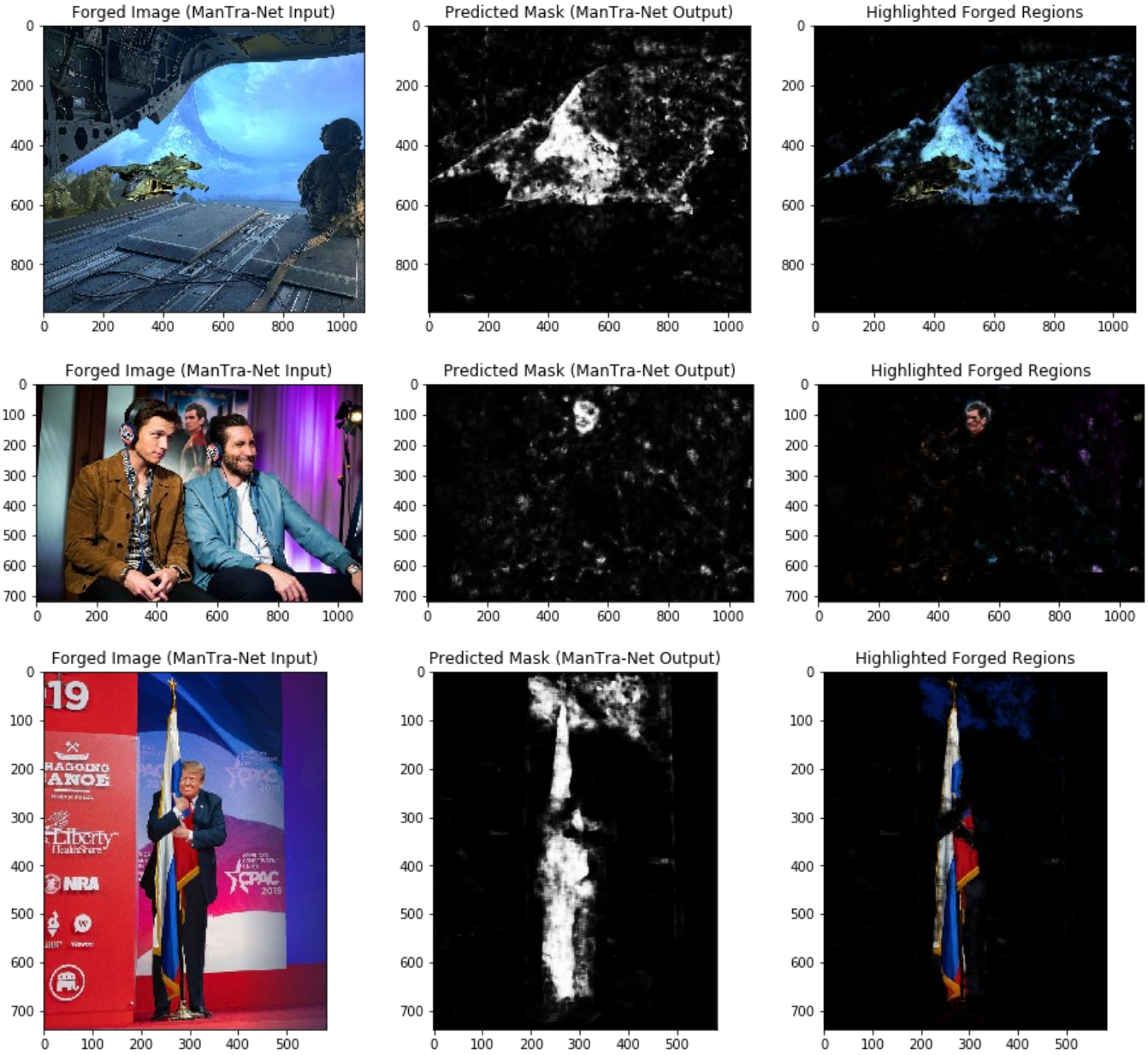
For the training process of the Manipulation Tracing Network the researchers used original patches with thresholded (low) homogeneity from the Dresden Image Database, to create training samples with random manipulations. For the evaluation of this submodel, images from the the Kaggle Camera Model Identification (KCMI) dataset were used. The Local Anomaly Detection submodel was trained on four different synthesized datasets, including the splicing dataset from [24], USCISI - CMFD from [1], a removal and an enhancement dataset.

The manipulation trace extraction is not a new technique, but until now it has been used for the detection of a small number of different manipulation attacks. The novelty of ManTra - Net comes from the fact that the researchers expanded this set of attacks, by closely examining the already known attack types and further splitting them based on small varieties that could form a

brand new attack cluster. According to the researchers, they focused on finding differences that could break down the seven more common attack clusters. As an example, they split the generic attack set Blurring to some subsets, including Gaussian and box blurring, wavelet denoising and median filtering. They kept splitting those subsets by adding more detailed parameters in each new subset, until each one of the them could be described with an individual algorithm. This consequent splitting resulted to 385 different distinguishing manipulation types.

It can be easily perceived that ManTra - Net is a multi - use tool and applies remarkably on the challenges of today's tremendous number of manipulation techniques. A prediction of ManTra - Net includes a binary mask of the image, where white pixels represent the detected manipulated areas. Furthermore, an enhanced mask is also provided, showing highlighted the detected manipulated image areas. We have evaluated the pretrained ManTra - Net on some images from the Photoshop Battles subreddit and we provide its predictions to better illustrate its behaviour.




Forged Image (ManTra-Net Input)
Predicted Mask (ManTra-Net Output)
Highlighted Forged Regions
Forged Image (ManTra-Net Input)
Predicted Mask (ManTra-Net Output)
Highlighted Forged Regions
Forged Image (ManTra-Net Input)
Predicted Mask (ManTra-Net Output)
Highlighted Forged Regions

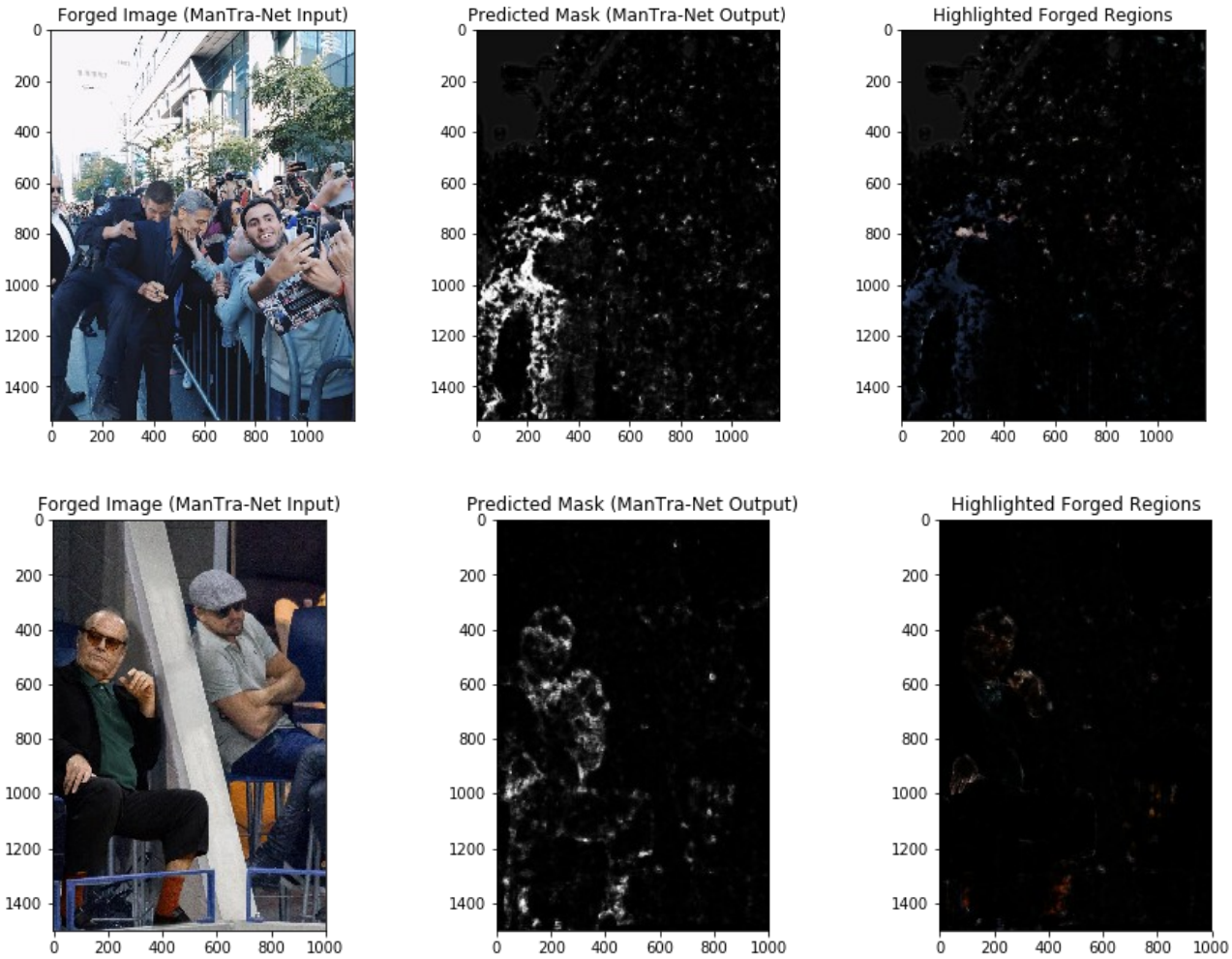


Figure 3 Qualitative results - pretrained Mantra - Net [28]

6. Detecting Fake Faces

YouTube, social media and other video/image hosting platforms are more or less an everyday routine and as a consequence, contents that go viral are likely to be seen by a vast amount of people.

Considering the rise of image and video manipulation techniques in combination with the spreading of fake news, it is possible that audio/visual content of well known people with exposure to the media to be doctored. This, among other types of attacks, includes tampering of their faces for beautification, intentional misinterpretation or falsification reasons. Especially after taking into consideration how facial expressiveness impacts the message a person is trying to convey, the tampering of a facial expression in any way may completely alter the truth of a speech, and thus is an individual type of attack that has to be faced.

In this chapter we provide a description of two models that are dealing with the detection of synthetic or photoshopped faces. We discuss our Face Forensics model and the most recent model for the detection of photoshopped facial features.

6.1. Detecting Photoshopped Facial Features

In [31], Wang et. al. based on the fact that Adobe Photoshop is one of the most commonly used image manipulation software, generated a dataset of fake faces by utilizing a widely used Photoshop tool for altering facial features, such as mouth, nose, cheekbones etc. They used a Dilated ResNet to localize the manipulated areas of fake faces and in some cases they also managed to reconstruct the original face by replacing the manipulated facial features with the most close to the original features they predicted.

The researchers scripted Photoshop to automatically apply the Face - Aware Liquify effect on a very large set of pristine face images from the Web. The Face - Aware Liquify tool is known for its high level manipulation capabilities on faces, such as the enlargement of a facial feature. This dataset was used for training a CNN on the detection of face warping, but the researchers also used a smaller dataset of fake faces, created by a professional artist, to evaluate their method.

Two different models where utilized: A global binary classifier to declare a face as fake or pristine and a local warp editor, for the localization of the tampered areas. For the binary classifier the researchers built a ResNet - 50, pretrained on ImageNet and they trained it on aggressively preprocessed fake and pristine face images (cropped, resized, flipped, JPEG compressed, brightness, saturation and contrast - altered) . The results on the binary classification task were better than the FaceForensics++ [33] and the EXIF models' [3]. For the localization task, they used a dilated ResNet originally designed for semantic segmentation. In order to localize the facial features that were tampered, they trained the model to predict warping fields measuring the distance between the pristine and the fake image on a pixel level. With the predicted optical flows obtained, they were also able to construct an approximation of the original image.

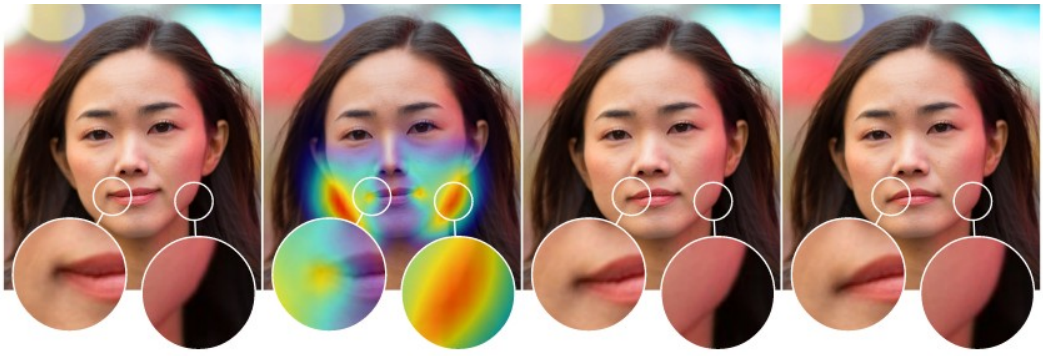


Figure 4 Qualitative results - FALDetector [31]

6.2. FACE FORENSICS

Face Forensics [32] is a large - scale video dataset consisting of around half a million images of synthetic faces, that were extracted from more than 1000 videos. For building this dataset the researchers used state - of - the - art face tampering methods on YouTube videos and they provide two smaller datasets that can be used for classification and segmentation tasks.

For the Source - To - Target Reenactment Dataset, the researchers use the Face2Face method where two images are used, one from the source actor and another one from the target actor. The synthesis of the output image is performed by applying the mouth of the source to the target actor, to produce a new image with a tampered facial expression of the target actor. The dataset is split into a training set of 704, a validation set of 150 and a test set of 150 videos and can be used for both classification and segmentation tasks.

The Self - Reenactment dataset was also generated with Face2Face approach, but the same video was used as both source and target, meaning that the output video has a fake facial expression that was generated by the combination of two different facial expressions of the same actor. As a result, forged and ground truth video pairs are acquired, which are split to training, validation and test sets the same way as the Source - To - Target Reenactment Dataset.

The researchers use the Face Forensics dataset for both fake image classification and fake image segmentation tasks. For the binary classification of whether an image is manipulated or not, they use the Source - To - Target Reenactment Dataset to train an XceptionNet. Ten frames were used from each one of the 704 fake/original training videos, 150 fake/original validation and 150 fake/ original test videos. The results of the videos with different compression types outperformed other similar approaches. For their segmentation task, the researchers also used an XceptionNet to localize the manipulated pixels. They trained the model on patches of 10 video frames from the training videos, some of them coming from the background, while others from present face and its boundaries with the background. Their results also outperformed other similar segmentation approaches they were compared with.

6.3. IMPLEMENTATION DETAILS

Following a different approach, we decided to focus the classification and segmentation tasks to the face part of the image. To this effect we utilized the dlib face detection library [37] for

locating patches that feature human faces. The extracted patches were resized to a common size of 96 by 96 pixels. Similar to [32] we used ten frames from each video to create the training, validation and test sets.

As a first stage we used a ResNet - 50 [38] model as the encoder part of a face autoencoder. The decoder consisted of the equivalent deconvolution and unpooling layers to bring the image back to the original shape. We trained the autoencoder for 100 epochs with the extracted faces from both valid and photoshopped video frames. For the training we used the mean square error loss function on the pixel values. In order to compose a manipulated face classifier we replaced the decoder part of the previous model with four fully connected layers of sizes (256, 1024, 256, 1). All of the layers were followed by a sigmoid activation. This model was first trained for 50 epochs keeping the encoder layers frozen. Finally we trained all the layers of the model for another 100 epochs to detect if a face comes from the original or the manipulated video using a binary cross entropy loss with an adam optimizer. This setup resulted in a model with comparable accuracy of 96% to the one reported in [32].

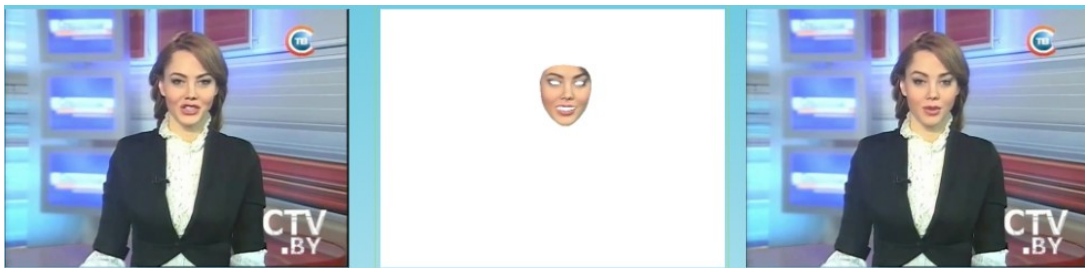


Figure 5 Face Forensics: The original image, its mask and the altered version of the image

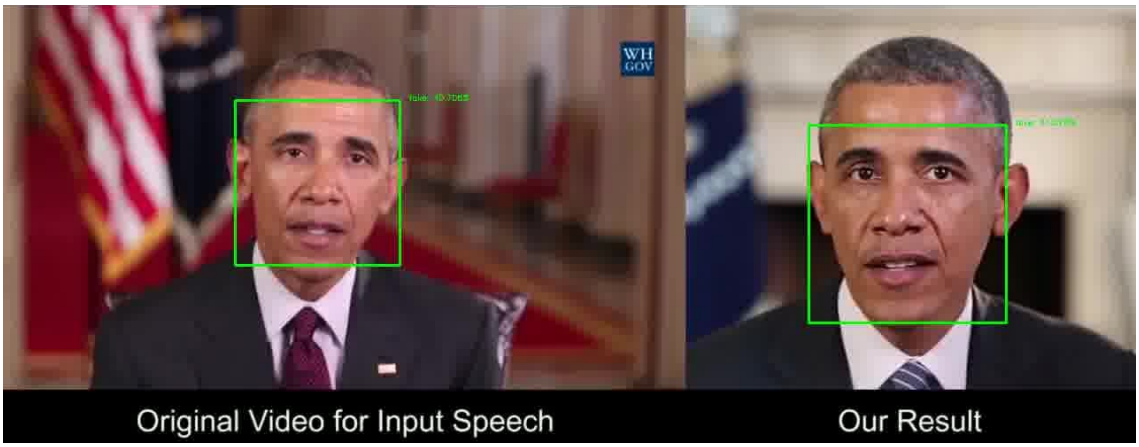


Figure 6 Qualitative results of the Face Forensics model. The left panel is showing an original video of Obama speech, while the right a fake video produced with deep learning models, using photos of Obama and audio from the same speech. The green bounding boxes show the detected faces and the probability that the face is not authentic.

7. OTHER TOOLS FOR AUDIO-VISUAL ANALYSIS

As described in chapter 2, discovering a training dataset of quality for supervised learning algorithms is a quite difficult task. Accessing pristine or tampered images may be easy, but establishing a connection between a tampered image and a ground truth label is not always a successful process, since fake images do not always come with their unmanipulated versions available. This makes supervised learning techniques a really difficult task, since manual labeling or annotation is needed.

To overcome this problem, some recent researches were based on the implementation of models that can be trained on unlabeled data. In this chapter we present the state - of - the - art models of unsupervised learning for the detection of manipulated audio or visual content.

7.1. WORKING WITH UNLABELED DATA

In [3] the researchers built a model that can detect visual image manipulations, despite the fact that it was trained only on pristine (not-manipulated) images. The learning algorithm the researchers used is detecting whether an image is self - consistent (all image regions come from the same camera), by learning features from image EXIF metadata.

The training dataset consists only of real images and the EXIF metadata are used as a supervisory signal. For each EXIF tag, a classifier is used on pairs of image patches to determine the consistency. The resulting classifiers are then jointly considered to justify the consistency between image patches of an input image.

For the purposes of the training phase, the researchers used nearly 400000 images downloaded from Flickr, while they formed their training set by random sampling on the images and extracting patches. They finally made predictions on nearly 50000 patches that had available EXIF metadata, by using a Siamese Network in order to predict whether the images share the same values for each EXIF metadata tag.

In the evaluation phase, a potential tampered image is used as an input to the model, which computes the consistency between many pairs of image patches. The resulting score is the indicator used for declaring an image as tampered or pristine. A low score indicates that it is high possible that the image was created by two (or more) different image pipelines, so possibly it is tampered.

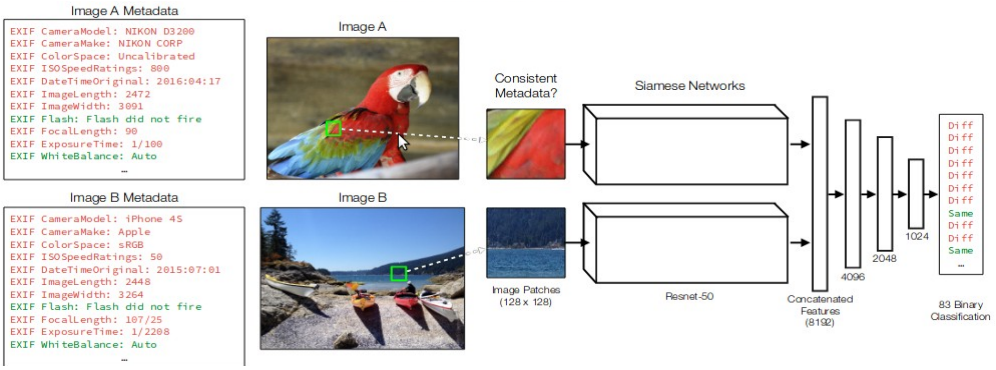


Figure 7 Self - consistency (EXIF) model: Prediction of consistency between the EXIF metadata of two pristine image patches [3]

To detect if some of the images have gone through post processing attacks, like blurring, artifact resizing etc., the researchers enhanced their model, by adding the task to predict the existence of different post processing operations between image patches. To achieve that, they added augmentation operations during training. The second task further enhances the EXIF model's performance. Even if an image patch has the exact same metadata as the rest of the image patches, the splice detection model could still locate post processing attacks on a potential fake patch.

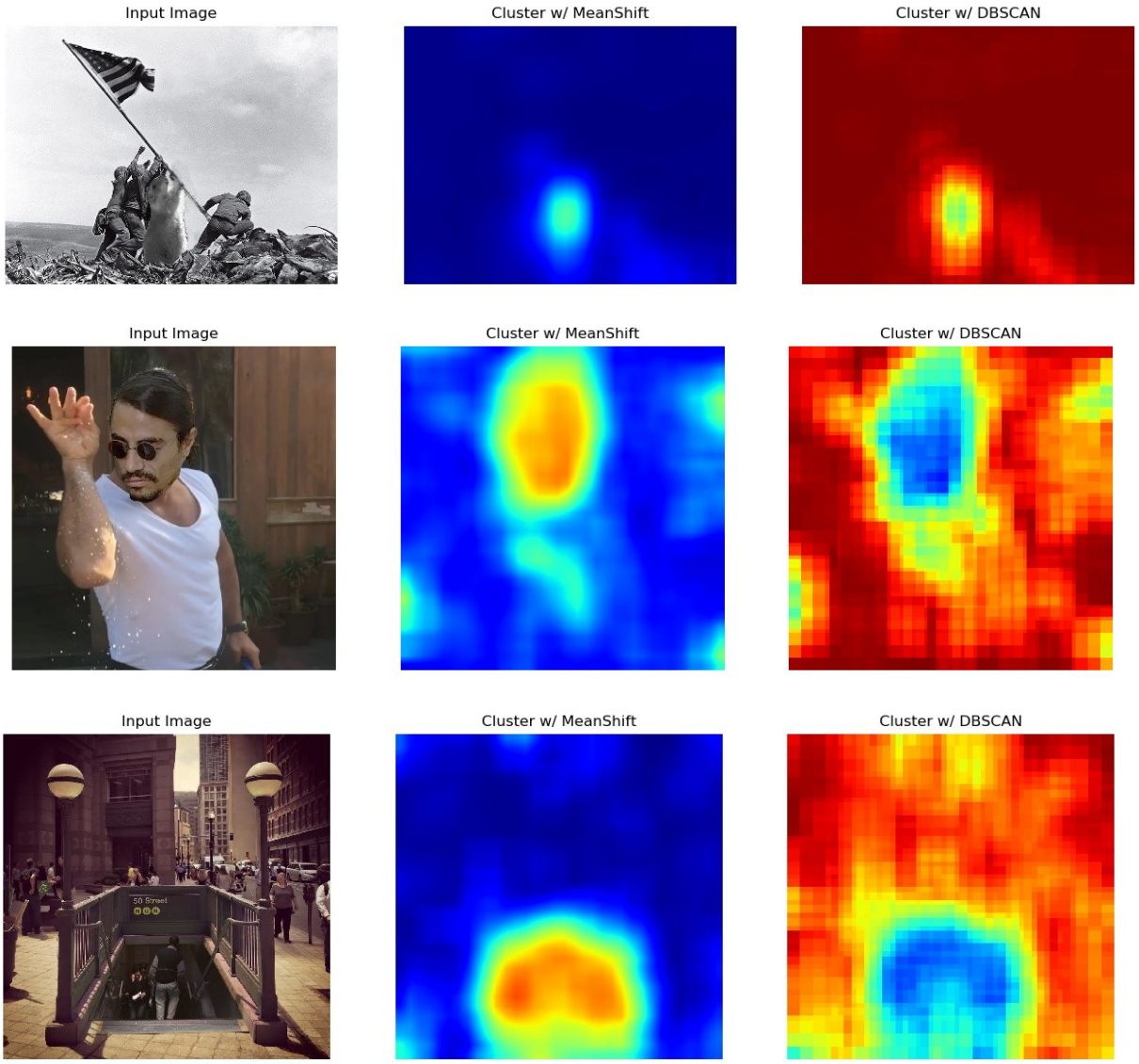
In order to evaluate the models, the researchers had to decide a metric to manage the transition from the discovery of patch consistency, to the evaluation on an image level. For each image patch they formed a consistency map by comparing it with all the rest image patches. Finally, they used Mean - Shift for the aggregation of the resulting feature maps, to one final prediction score.

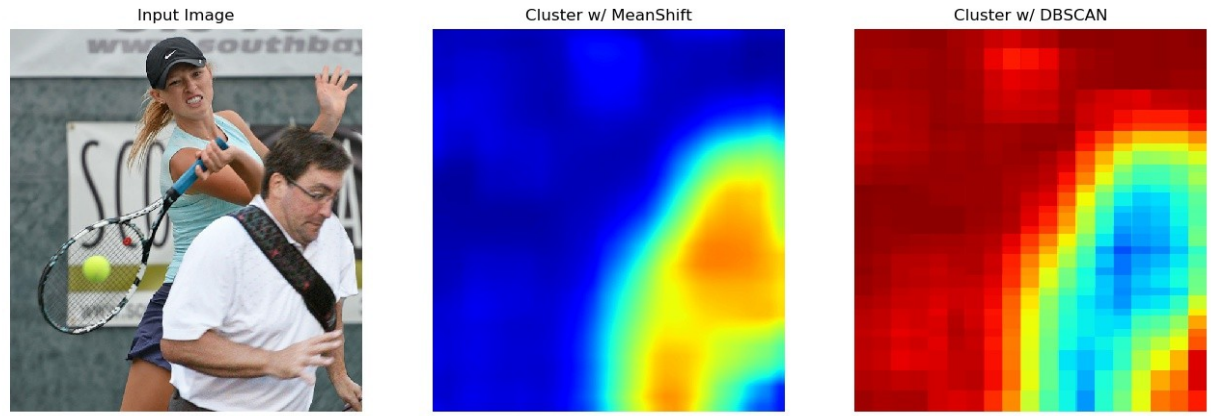
Finally, the researchers compared their splice detection and localization results on Columbia, Carvahlo, and Realistic Tampering datasets, which despite the small number of available samples, combined can provide a variety of image manipulation techniques, i.e. different post processing operations and copy - move manipulations. The researchers achieved state - of - the - art results on image splice detection and localization for two of the aforementioned datasets. The results are remarkable, especially after taking into consideration a) the fact that the model was trained only on original images, b) the absence of labeled data in the training set.

Since news articles are often accompanied by images shot by professional cameras, we consider the self - consistency model as highly suitable to the FANDANGO project. The detection of fake images in news articles can be achieved by both manipulation detection on an image level and inconsistency detection on image metadata (EXIF) level. As an example, in the case that a local news site posts a manipulated image, it is possible that the self consistency model would detect both the manipulated image area and the EXIF inconsistency and would be confident about its predictions. On the other hand, it is high possible that a more broadly known news site, removed the EXIF metadata from the manipulated images, for protecting its reputation and conserve its apparent objectivity. Even in that case that the model would not be able to consider the metadata factor, the splice detection part would still have great chances on the localization of the spliced area.

A prediction of the EXIF model is a consistency map, where self - consistent image areas get the same color. For the aggregated feature map formed with Mean - Shift, dark blue areas show a low possibility of fakeness, while the warmer the color, the more possible it is for image areas to be fake. The colors in the case DBSCAN is used are inverted.

We have applied the pretrained EXIF model on fake photos from the Photoshop Battles dataset and for a better understanding of the outputs, we demonstrate some of the results we acquired. We emphasize that the EXIF model is more suitable for real and high resolution images, because of the existence of metadata, but also for images where splicing attacks were performed.





Original Image (Removal)

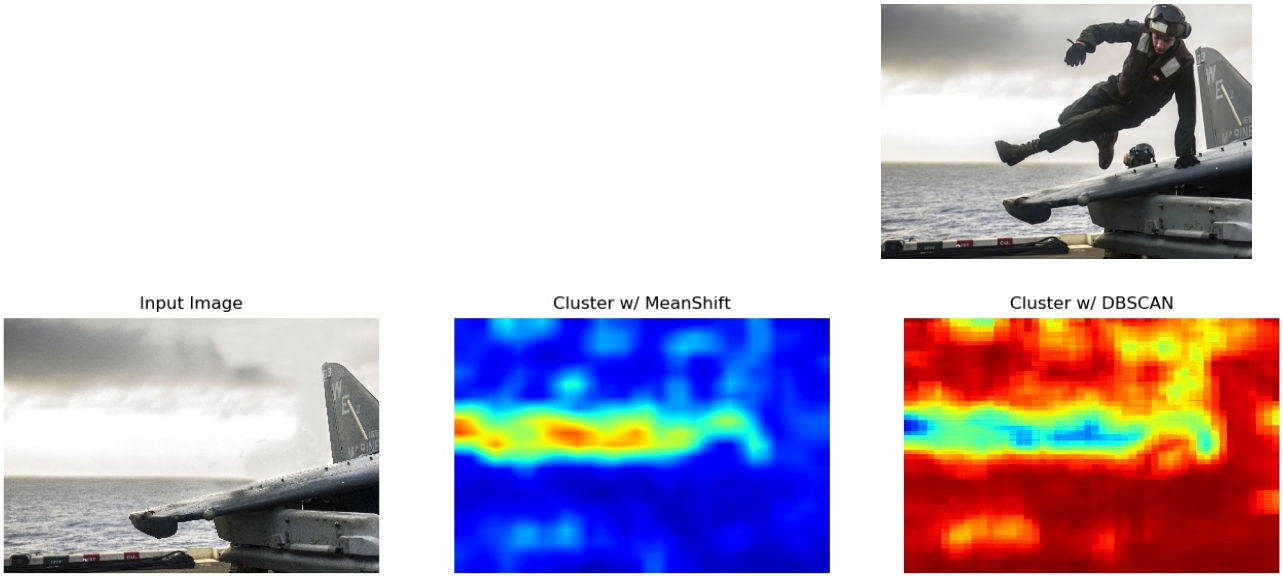


Figure 8 Qualitative results - EXIF model [3]

For the last prediction we also include the original version of the image, where a marine is jumping off the plane. In the manipulated version, the marine has been removed. This is a classic manipulation attack for hiding information by removing objects, or using background patches to cover the object of interest. The prediction of the EXIF model approximately locates the region of interest.

7.2. Identifying Sounds And Sound Sources

Inspired by human awareness about a scenery, where we use all of our senses to perceive the surrounding information, recently research topics about multisensory and localization of sound

sources in videos became popular. These tasks are really challenging, since different sounds in a video are perceived in the same single audio channel.

The localization of sound sources could be a good indicator about whether the audio of a video matches the visual content. Some interesting applications have recently been implemented with the objective to locate and visualize the on - screen source of a sound, or separate different audio streams in a video. In this chapter we further analyze three state - of - the art models dealing with sound source localization, which we consider are closely connected to the detection of fake information on audio and visual content and can result to remarkable predictions on real world videos.

7.2.1. OBJECTS THAT SOUND

Recently, some researchers implemented models that are capable of learning cross-modal embeddings from audio and visual components, while, given the audio stream, they can also locate the object that is causing the sound in the image [26]. The models were trained on unlabelled videos, accepting as inputs video frames and audio streams and their task was to determine whether the two components were in correspondence. The model conducts an automatic audio - visual alignment, to generate labels for the components (in correspondence/not in correspondence). Pairs of audio-image components that come from the same video are considered to be in correspondence, while audio-image pairs that come from different videos are not in correspondence. Thus, the method the researchers use is a self - supervised method, since the labels are not provided, but created by the model.

The models built by the researchers are combined, to provide the final correspondence/no correspondence outputs. For the extraction of visual features from images, the researchers built the Vision Conv - Net, while for the extraction of audio features they implemented the Audio - ConvNet. The two submodels are combined and used in fusion in a layer that computes the Euclidean distance between the two components, to finally form the AVE - Net, which will output the aligned vision and audio embeddings.

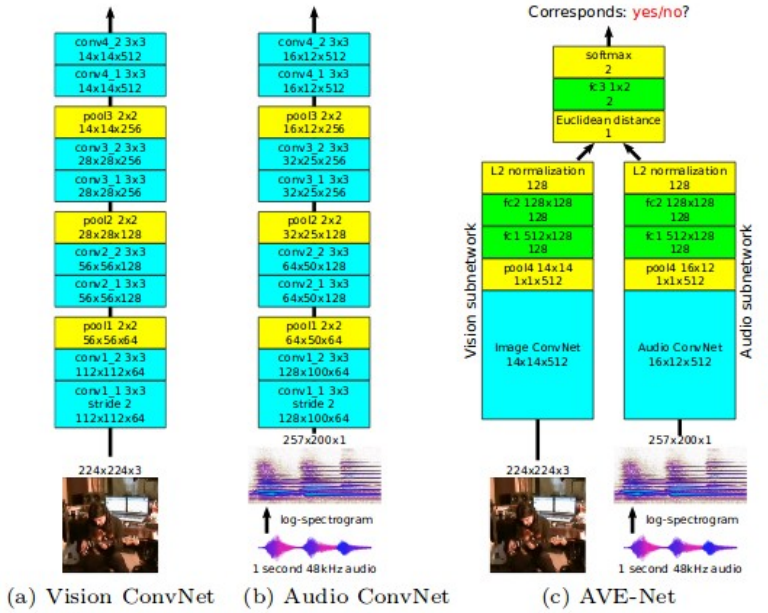


Figure 9 The Architecture of AVE - Net for the prediction of audio - image correspondence [26]

The most interesting part about AVE - Net is the fact that it is trained on different modalities (image - audio) and can distinguish semantic concepts in them. In order for the predictions to be meaningful, the model must interpret the concepts it learned and result to positive output in case they are connected.

Since AVE - Net had already the skill to establish meaningful connections between objects in images and specific sounds, the authors further improved AVE - Net to localize the objects that are causing the sounds in the images. As a result, they built AVOL - Net, where the similar parts of audio and visual modalities resulted to the localization of the object that is causing sound in the image.

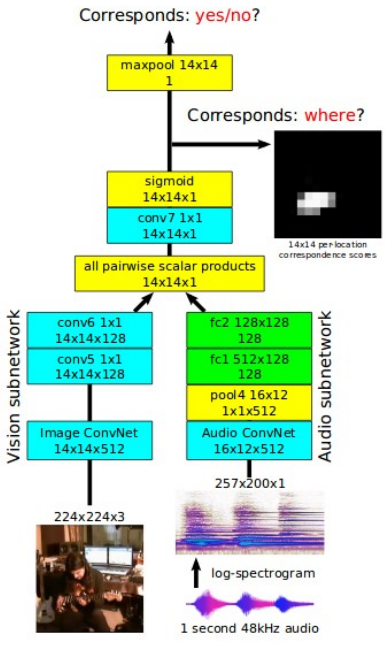


Figure 10 The Architecture of AVOL - Net for the prediction and localization of sounding objects in images [26]

Figure 11 Qualitative results of AVOL - Net [26]

7.2.2. SEPARATING OBJECT SOUNDS

Another recent research dealing with localization of sound and audio separation in videos was proposed in [27], where the authors implemented a framework to separate multiple sound sources coming from the same video. The novelty of this research is the fact that the framework the researchers propose is a multi - instance multi - class model that learns to separate different sound sources, by using state - of - the - art object detection techniques from large scale real world videos.

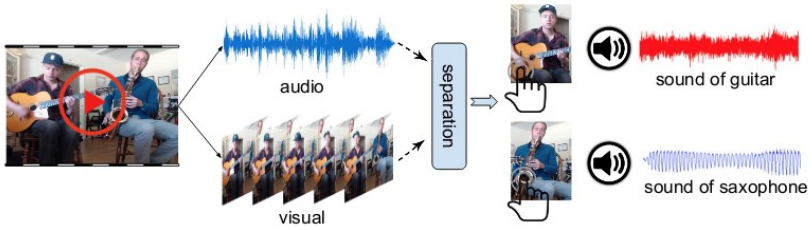


Figure 12 Learning features from unlabeled videos to separate multiple object sounds [27]

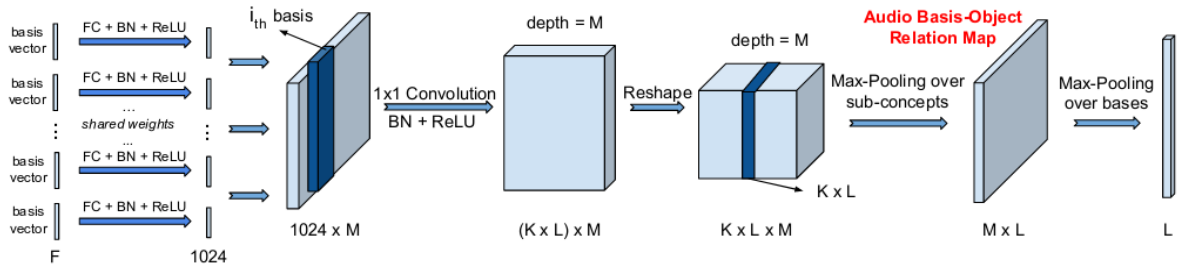


Figure 13 The Architecture of MIML model [27]

The researchers performed an unsupervised training that takes as input an unlabeled video and returns the different sound sources separated, along with the visual part of the video that caused each one of them. For the audio part, Non - Negative Matrix Factorization is performed on the audio spectrograms, while for the detection of visual objects a ResNet-152 pretrained on Image-Net is used. The final step is the Multi - Instance Multi - Label learning, which is responsible for matching each audio source with the corresponding visual object.

The MIML model's inputs are the audio vectors that were obtained by the NNMF part of the training pipeline, one for each video. The predictions of the object - detection part of the network are used as labels that characterize the detected objects and finally, the outputs of MIML are the predictions about what objects were responsible for the corresponding audio streams.

The process of matching audio streams with visually present object, is a difficult challenge. The audio streams that are given as inputs may sometimes be caused by multiple objects in the video, or even a sound that is not somehow connected to some object in the video. Additionally, the object labels used are not always correct. Since the researchers used a ResNet pretrained on Image-Net, the object detection task outcomes may sometimes be incorrect and as a result the given label may not correspond to the objects that are actually present in the video. To0 overcome these difficulties the researchers Use the Audio Basis-Object Relation map, to only select audio and object labels with matching scores and finally obtain qualitative and quantitative results that prove their implementation is robust against these factors.

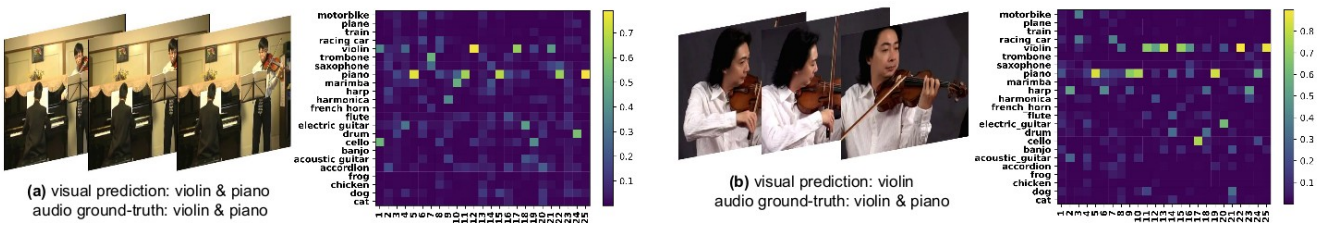


Figure 14 Qualitative results - MIML model [27]

7.2.3. SELF - SUPERVISED MULTISENSORY FEATURES

Similarly, the researchers in [4] proposed a model that is capable of predicting whether audio and visual components of a video are "temporally aligned". In their implementation they modeled both audio and visual content jointly to create a fused multisensory model, in order to use it in applications of a) localizing the source of the video sound, b) recognizing actions in videos, c) separate audio and visual content of a video.

The training phase of the multisensory model was as well a self - supervised process, i.e. the model learned to identify features from the input data by itself. As a training dataset, the researchers used 750000 4.2s videos in total. They misaligned synthetically some of the videos (the sound was shifted randomly for 2.0 to 5.8 seconds), and used them in combination with temporally aligned videos. As a result of the training phase, the 3D CNN multisensory model learned rich audio-visual feature representations.

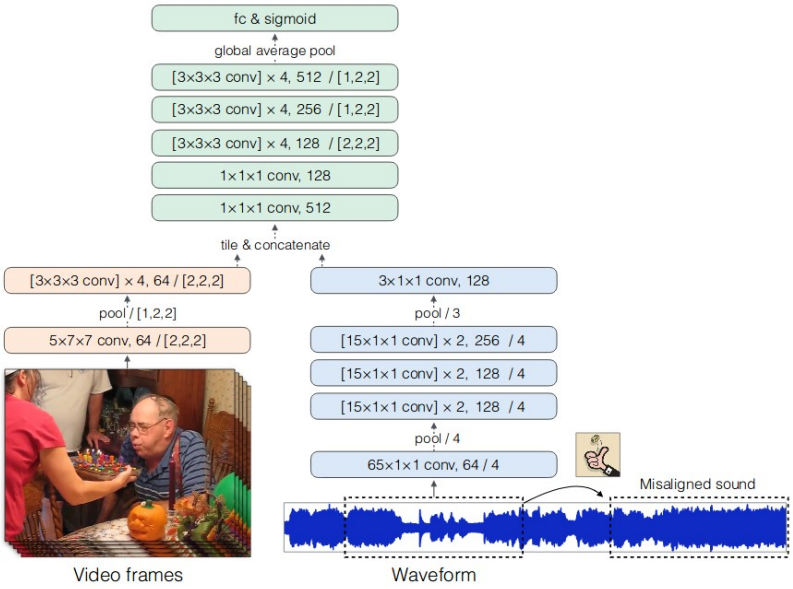


Figure 15 3D Convolutional Neural Multisensory Network for discovering audio - visual misalignments [4]

The main case of this research was the learning of audio and visual video components combined, in order to be decided whether such features can be used in action recognition and sound source localization tasks. The innovative part is the fact that the researchers used these learned features on a task of on and off screen sound separation and they managed to built the first model that is highly accurate on real world videos.

The on - off source separation model takes as input the learned features of the previous task. These features are transformed by some layers, so that the final input is a synthetic mix of sounds consisting of both on - and off - screen sounds, in the form of a spectrogram. The hypothesis was that after the training phase the model should be able to separate the two kinds of sounds, i.e., predict the spectrograms of the two separate audio inputs. The selected architecture is a U-Net.

The training of the on - off source separation model used as inputs only raw data, i.e. the videos were not labeled or preprocessed on any auxiliary way. The researchers synthesized a dataset by randomly combining video pairs and using the audio of the first as the off - screen sound for the second video.

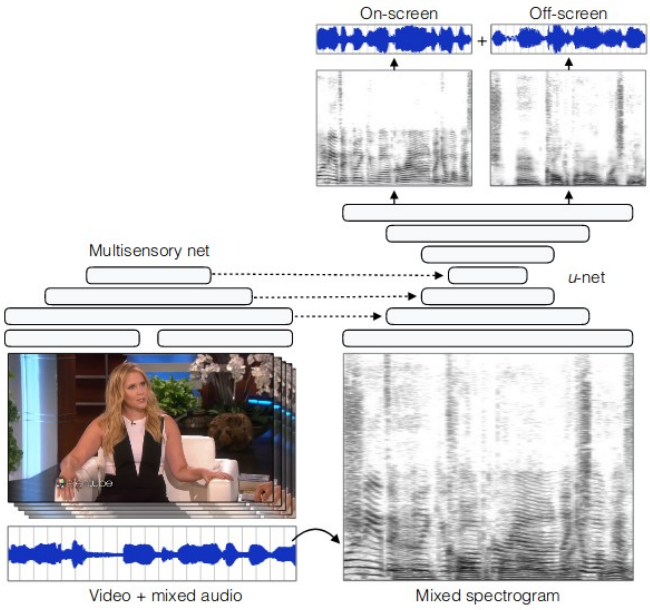
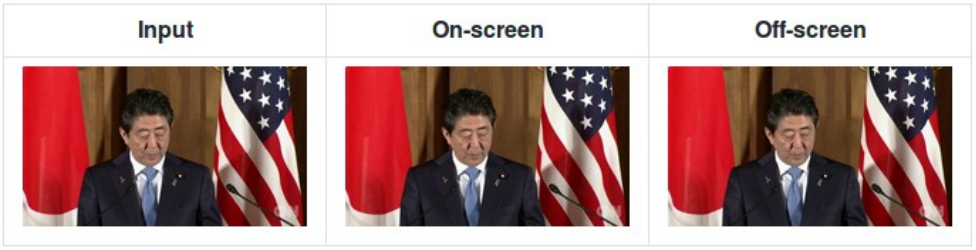


Figure 16 On - off screen source separation model [4]

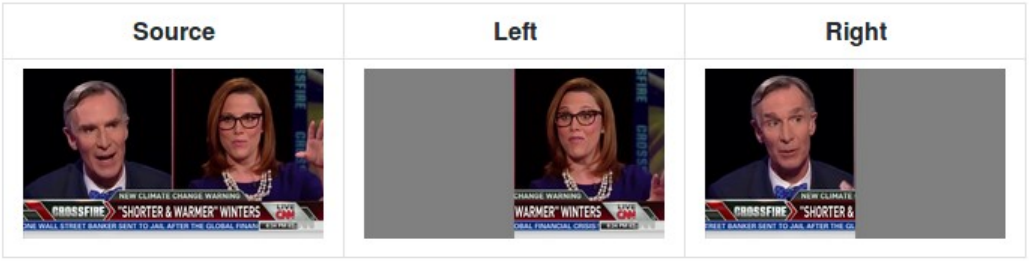
In the context of fake news, video manipulation is a common technique used by news sites to present the content of a video transformed or intentionally misinterpreted. The reasons behind this kind of manipulations could be, for example, changing the speech of a politician, either for protecting them, or spreading propaganda against them.

The most interesting part about this research is the fact that the models are very confident at making predictions on actual - real world videos. Thus, we expect the MultiSensory framework to catch high level sound misalignments that may be an indicator of fakeness in a video. Furthermore, the on - off screen audio separation methods could reveal an on screen phrase that the attackers may have intentionally concealed, by adding off screen sounds. For these reasons we consider the MultiSensory framework as a useful tool that can facilitate the needs of fake news detection on audio and visual content.



a) b) c)

Figure 17. Qualitative results of MultiSensory model on an on - off screen sounds separation task. a) An original video where we can hear the main speaker's voice, but also an off - screen voice from the translator. b) In this version of the video the off - screen sounds (translator - audience) have been removed and only the on - screen sound (speaker) can be heard. c) The off - screen sound has been isolated (only the translator's voice can be heard). The videos can be found at: https://github.com/andrewowens/multisensory



a) b) c)

Figure 18 Qualitative results of MultiSensory model on an on - off screen separation and voice removal task. a) A source video with overlapping speech by two different speakers. b) The left speaker has been visually masked out and his voice was removed, so only the right speaker can be heard. c) The right speaker has been visually masked out and her voice was removed, so only the left speaker can be heard. The videos can be found at: https://github.com/andrewowens/multisensory

Figure 19 Qualitative results of MultiSensory model on a sound source localization task. The videos can be found at: https://github.com/andrewowens/multisensory

8. Qualitative Analysis

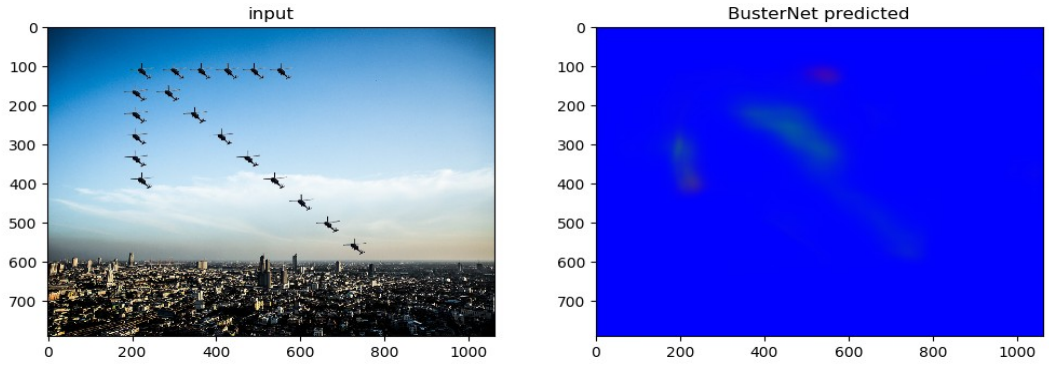
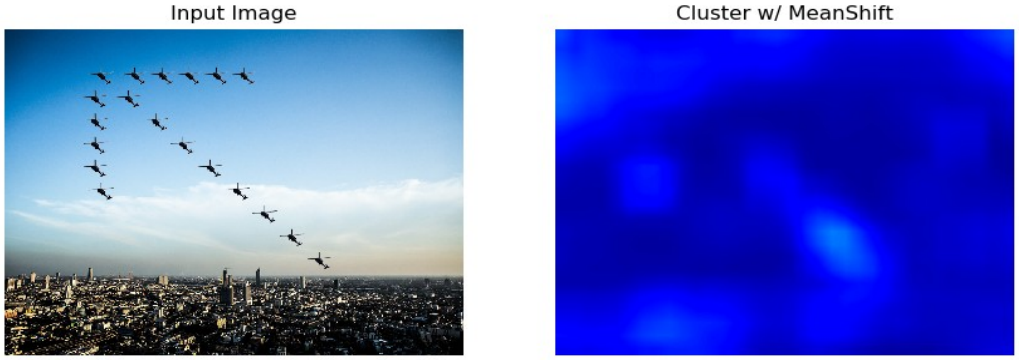
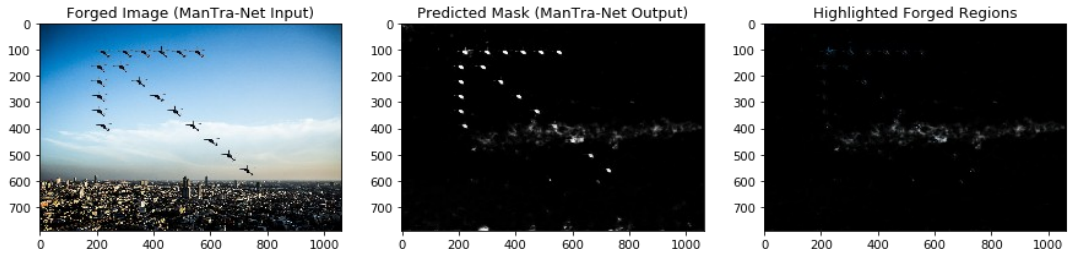
In this section, we provide the predicted masks of the image manipulation detection models we discussed earlier. We have evaluated the models on the same images, for better illustrating the use cases where BusterNet, EXIF and ManTra - Net could facilitate the inspection of a manipulated image. Unlike the images we used earlier for the descriptions of the models, here we present actual or realistically enough manipulated photos, that is not easy for the human eye to distinguish, to analyze the models' behaviour on photos that the end user may indeed need to examine for forgery attacks. The following qualitative analysis is crucial for better understanding each model's strengths and weaknesses and can reveal important hints for the end user.

8.1. Qualitative Comparison Of BusterNet, EXIF, ManTra - Net Results

We emphasize that the utility of each model is input - dependent and we expect them to be confident and concrete on their predictions in cases that the manipulated input image is closely connected to the task the model was trained on. The end user may consult the predicted mask of each image, but should also thoughtfully analyze them furtherly during their decision making process, based on the individual task each analyzer is expertised in.

By all means, we cannot be sure about the exact kind of the attack an image was subjected to when it comes to images from the Web. Furthermore, we cannot be sure about their exact manipulated areas. This challenge is realistic though, since the end - users will also be unaware of the manipulation attacks performed on the images of interest.

To sum up, we can only guess what manipulations were performed on the test images of this qualitative analysis. The marking "Attack" in the following tables should be interpreted as "We examine this image for that kind of attack". In the cases it was available, we also include the original image as a reference, but still cannot be sure that it was not already somehow manipulated.

Attack	Copy - Move
Input	Image from PSBattles subreddit
BusterNet (1)	
EXIF (2)	
ManTra - Net (3)	
Original version of the image (4)	

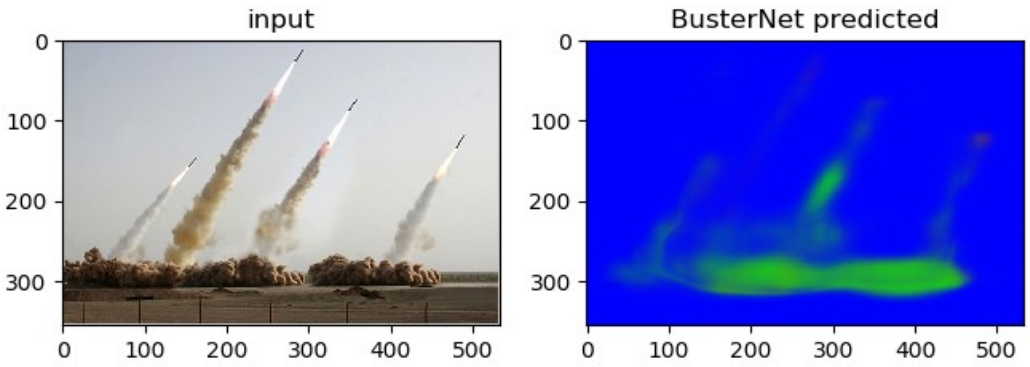
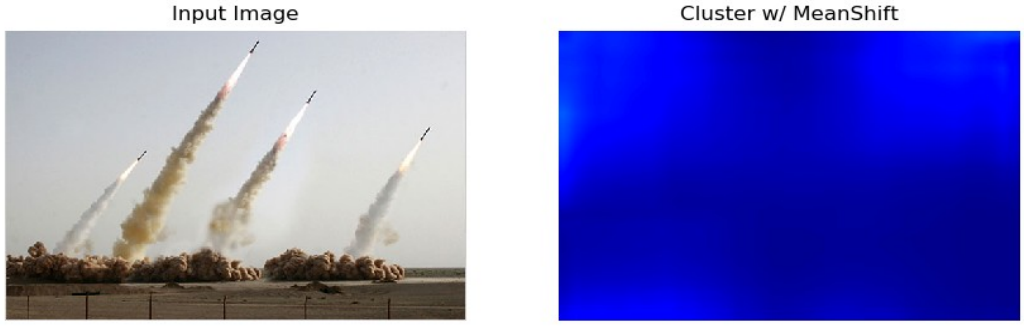
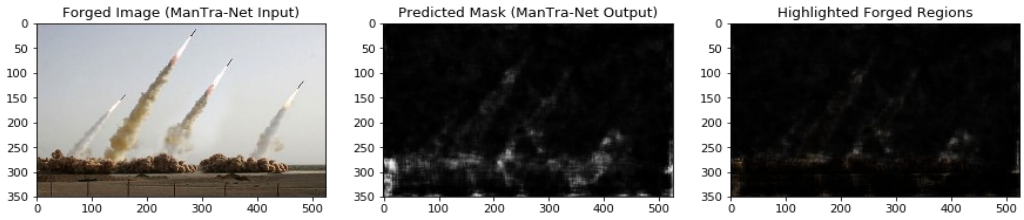
The original image (4) is showing a helicopter in the sky. In the manipulated version a brand new background is used, while the helicopters seem to be clones. Thus, we suspect that this image has been subjected to copy - move forgery attack and we expect BusterNet and ManTra - Net to result to more informative predictions.

BusterNet (1) locates most of the pasted helicopters that are shaping the arrow. Some of the predicted pixels are more green - like, while others more red - like, indicating that the photoshop artist after pasting the original helicopter on the new background, copied and pasted it many times until the arrow was formed. In the prediction we can vaguely discriminate all the three colors, meaning that both the Mani and the Simi Det have affected the final segmentation mask, but they were not very confident about their decisions on the pixels. We emphasize that, despite the fact that the background of the manipulated image can be considered as fake, since it was not present in the original version of the image, BusterNet captures only the fake objects. This is a distinguishing feature of BusterNet that discriminates it against the EXIF model we analyze next. A prediction of BusterNet will never catch fake backgrounds, since it is a segmentation model, trained to detect fake and similar objects.

The EXIF model (2) does not give a good enough mask for this image. Considering that it classifies the helicopter arrow and some other image areas (lighter blue) as self - consistent, it fails to identify and isolate the manipulated area of the image (helicopter arrow). It is an expected behaviour though, since it was not trained on the detection of cloned objects. This prediction would be considered as successful, in case that the helicopter arrow had different color from the rest image areas. The EXIF model is a classification network and will only classify image patches as self consistent, no matter what the content of each patch is (object/background).

ManTra - Net on the other hand, clearly captures the helicopter arrow on the predicted mask, since it is yet another segmentation model and copy - move was one of the many manipulation attacks it was trained on.

Considering the three predictions together, we can conclude that the image has been subjected to copy - move forgery attack and since the segmentation models result to clear enough predictions, there is no reason to take into consideration the prediction of the EXIF model.

Attack	Copy - Move or Splicing
Input	Actual manipulated image from [The Telegraph]
BusterNet (1)	
EXIF (2)	
ManTra - Net (3)	
Original version of the image (4)	

According to an article in The Telegraph, in 2008 Iran was accused of manipulating image (4) in order to hide the truth about a missile launch. More precisely, the fact they needed to conceal was that one of the missiles failed to fire. An Iranian newspaper released an image showing three missiles while firing and another one still on its launch base (the vehicle in the original photo). Later on, a new version of this image was released, showing four firing missiles, while the vehicle base was removed and a cloud of dust was added under the new (fake) missile.

As shown in the predictions, BusterNet catches the manipulated regions (1). Once again, we cannot be sure on whether the fourth missile was cloned from the same or an alien image. BusterNet localizes the spliced/cloned missile along with the manipulated environment in the front (vehicle, dust cloud). The fact that all missiles are green indicates that Simi - Det was confident that there were similarities between the four missiles and it is possible that one of them has been used as a source object. After observing BusterNet's prediction we can be more confident that the new missile was cloned from the same image.

On the other hand, the EXIF model was not able to localize the cloned (or spliced) missile, neither did with the manipulated environment. Instead, the EXIF model's prediction is a consistency map that vaguely classifies the missiles as self consistent with each other and cannot be of great help.

Finally, ManTra - Net (3) seems to capture a good proportion of the manipulated image areas, as in the predicted mask there are present both the missiles and the environment objects mentioned above. ManTra - Net was trained on the detection of different attacks, and even if we cannot be sure that this image is a clean copy - move example, its prediction shows all the manipulated areas.

By comparing the three models predictions, we could conclude that the fourth missile was cloned from the same image and so did the dust cloud. To back this argument up, we underline the main attacks on which the models were trained: Both BusterNet and ManTra - Net were trained on copy - move samples and they both predicted the manipulated areas, while the EXIF model was trained to predict self - consistent image patches.

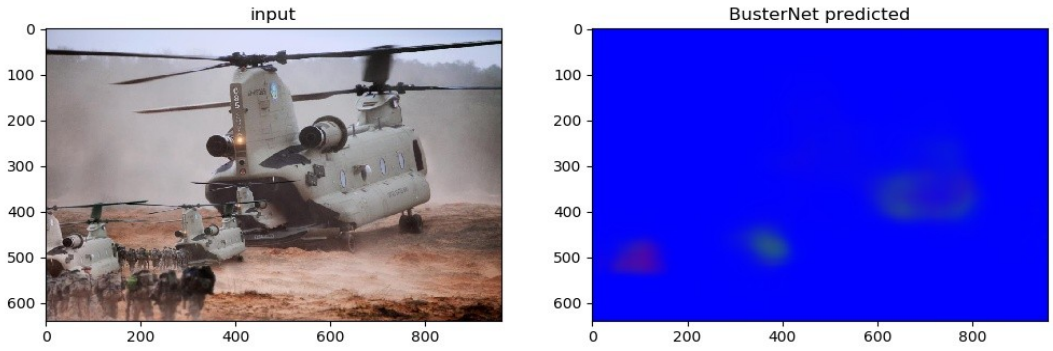
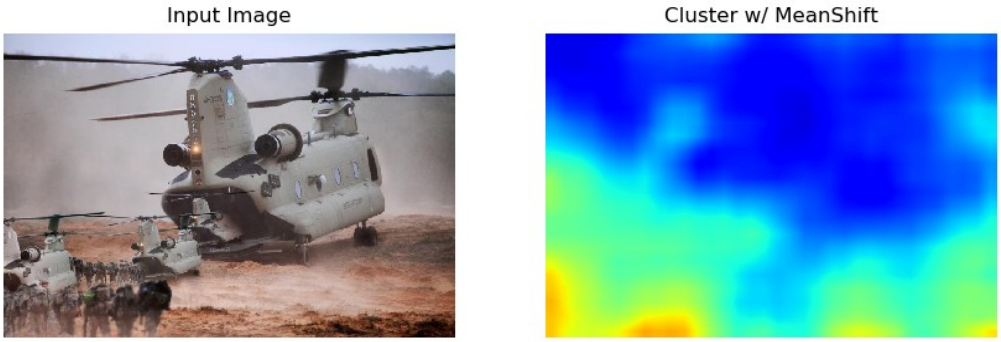
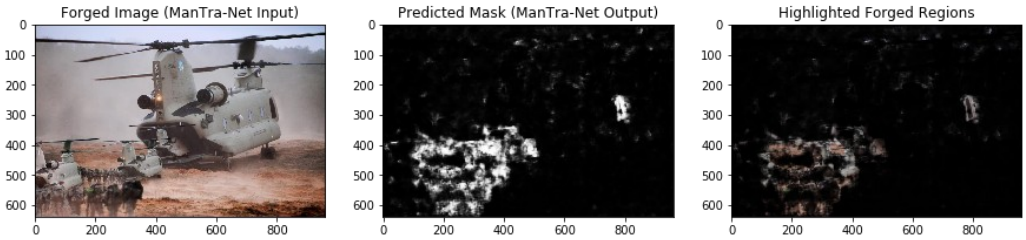
Attack	Copy - Move or Splicing
Input	Image from PSBattles
BusterNet (1)	
EXIF (2)	
ManTra - Net (3)	
Original version of image (4)	

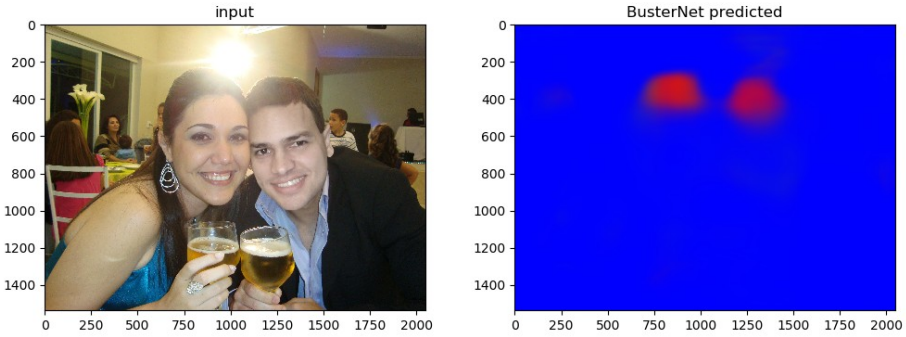
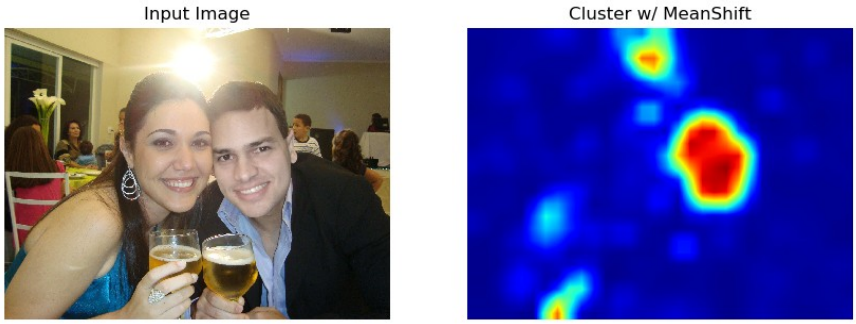
Image (4) is a sample from the Photoshop Battles subreddit. The manipulated version of this image (next to the predictions of each model) was posted in the comments section as an answer to the battle the original image triggered. By taking a closer look at the two images we can assume that possibly the original helicopter was cloned and pasted three more times in the image. Additionally, some soldiers were added in the front, who could also be clones of the soldiers in the original version. Thus, we examine this image for a copy - move and/or splicing attacks.

In prediction (1), BusterNet captures the second from the left side helicopter and paints it green, which means that possibly it was the source object for the first helicopter from the left side of the image, that is almost red. The detected area of the big helicopter is not very well distinguished, but seems like more green-like painted. This may be perceived like it has possibly, but not confidently, been copied and pasted to a new position.

In (2) the EXIF model clearly splits the image on two main parts with different self - consistency. In the front, it has captured almost all the manipulated objects, while being more confident about the spliced (or cloned) soldiers.

Finally in (3) ManTra - Net captures not only the manipulated area in the front, but also a small removed detail in the back side of the helicopter, that neither BusterNet or EXIF captured. Additionally, the highlighted forged regions are clearly showing the manipulated objects, a behaviour that was not acquired in the previous Mantra - Net's prediction we presented before.

The fact that both the segmentation (BusterNet, ManTra - Net) and the classification (EXIF) models produced decent predictions can be interpreted as that the image has been processed under both copy - move and splicing attacks.

Attack	Splicing
Input	Image from 1st Image Forensics Challenge Phase 01 dataset
BusterNet (1)	
EXIF (2)	
Original version of image: NOT AVAILABLE (3)	

In this case the fake image is from the 1st Image Forensics Challenge dataset, where all images are high resolution and shot by regular cameras. The original version of the image is not available, but we provide its binary mask, clearly showing the manipulated area (3). The manipulation attack in this case is splicing, since the head of the man in the right is fake and has been copied from an alien image.

BusterNet prediction (1) captures pixels in both heads. The good aspect of this prediction is the fact that the predicted pixels are bold red, showing confidence from the side of the Manipulation

branch. It is not perfect though, since the woman’s head is not fake. We could just conclude that this image is highly possible to be manipulated in some way.

The EXIF model’s prediction (2) is great in this case. The lights/shinings confused the classifier,but it colored them with colder colors than its main detected area. It clearly captures the fake head and paints it with red color, meaning high confidence about splicing detection and/or metadata inconsistency. This is the expected behaviour, since the researchers who proposed it claim that their model is better at catching inconsistencies in real and high resolution photos.

Since the EXIF's model is that confident and self - explanatory in this case, there is no reason to take into consideration BusterNet's prediction too.

9. Extending BusterNet With Nonlinear Convolution Kernels

The past years CNNs have been widely used due to their remarkable results on computer vision tasks. The typical CNN models are linear systems, since their output is a linear combination of the input image and the kernel. However, the progress in computer vision research has proven that nonlinear operations can enhance a CNNs robustness and result to the extraction of more complex image features. The most common way to introduce nonlinearities to a CNN is the utilization of nonlinear activation functions, such as ReLU.

Going beyond the nonlinear activation functions, a recent research product validates that the introduction of nonlinearity to the convolution operation itself can boost a CNN based learning process dealing with an image classification task [29]. As an extension, we have experimented with the addition of nonlinear convolution kernels on a CNN based image segmentation task and propose a novel method that can result to more detailed segmentation masks, in the context of manipulated image areas localization. Thus, we apply our method on the Manipulation Detection branch of BusterNet, the state - of - the - art CMFD model.

In this chapter we further analyze our novel idea for extending image segmentation models efficiency and describe our proposed method.

9.1. Nonlinear BusterNet

Based on research results from the field of neuroscience, which validate the existence of non - linear operations in the visual cortex perception, the researches in [29] proposed the use of non - linear convolution filters for boosting CNN models dealing with RGB image classification tasks.

For the introduction of nonlinearity to the convolution operation the researchers propose the adaptation of the Volterra series, as a layer plugged into a CNN, since they also use kernels to transform their inputs, like the linear convolution operation. Due to the fact that the number of model parameters is gradually over increasing, only the second order volterra kernels are used in the researchers' implementation, i.e. the matrix representing the coefficients of the quadratic connections between image pixels.

Given an input image patch x with n = k x k elements and a linear convolution kernel, the output of the standard convolution operation is computed by:

$$y(x)=w1T\cdot x+b,$$

where the kernel w_1^T is a vector with n elements, x is the respective vectorized image patch that the kernel is interacting with for each sliding step and b is the bias. For the researchers' proposed method the output of the second order volterra filters is computed by:

$$y(x)=xT\cdot w2\cdot x+w1T\cdot x+b,$$

where for the new term, w_2 is the second order volterra kernel containing n^2 x n^2 elements, while x is the same vectorized image patch that both the linear and the quadratic kernels are interacting with at each sliding step. The second term is the standard linear convolution operation described earlier and b is the bias term. In details, w_1^T contains the coefficients of the linear term:

$$\mathbf{w}_1{}^T = \begin{bmatrix} w_1^1 & w_1^2 & \cdots & w_1^n \end{bmatrix}$$

while w_2 the coefficients of the quadratic term:

$$\mathbf{w}_2 = \begin{bmatrix} w_2^{1,1} & w_2^{1,2} & \cdots & w_2^{1,n} \\ 0 & w_2^{2,2} & \cdots & w_2^{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_2^{n,n} \end{bmatrix}$$

The numeration (i, j) indicates the corresponding image pixels (x_i , x_j) that the linear and the quadratic kernel interact with.

The researchers implemented the second order volterra kernels as a layer in Torch7 [30], where all the matrix operations were implemented directly in CUDA. They plugged their layer into a Wide ResNet for the classification of benchmark datasets CIFAR10 and CIFAR100, where they outperformed the state - of - the - art results. This outcome validates the extraction of richer features from three - channel images, which can boost the results of a CNN.

Based on the research results of [29], we have examined the case that nonlinear convolution filters can also enhance the efficiency of image segmentation CNN - based models, including models dealing with the detection and localization of image forgery. Our hypothesis was that the nonlinearity would increase the spatial accuracy of the manipulated area localization on a pixel level and result to more detailed and well defined segmentation masks. To verify our hypothesis, we have experimented by plugging a nonlinear convolution layer in the feature extraction block of BusterNet's Manipulation Detection branch, which can be used as a self-sufficient model for the detection of image forgeries.

9.2. Implementing The Nonlinear Convolutional Layer

The implementation of [30] is in Torch7, where the logic behind matrix operations is CUDA - based. For implementing our novel idea, we needed to build a more flexible version of the nonlinear layer in Python, to further extend its usability. Since BusterNet is built in Keras, we have also implemented the nonlinear layer in Keras to use them in combination.

We built our Keras adaptation of the volterra layer by utilizing the proper tensor operations that were broadcastable and differentiable, to model the functionality of the volterra series as a learning process. Due to over - parameterization, we also modeled the layer to handle the second order volterra kernels, but it is easily extendable to higher orders. For the rest of this chapter, we refer to BusterNet or its submodels with the term Quad, in case that our nonlinear layer was used to affect the outputs.

As also described in section 4.1, BusterNet consists of two auxiliary segmentation submodels. Simi - Det is a submodel used for the detection of similar objects via self - correlation, while Mani - Det is a submodel for the detection of manipulated image areas. Both Simi - Det and Mani - Det branches can also be used independently for accomplishing their individual tasks. For starting our experiments, we excluded the similarity factor and plugged our layer into Mani - Det submodel, since Mani - Det' s architecture can generalize better to other types of segmentation tasks, such as object detection. Thus, we have built the QuadMani - Det to explore it as an individual model, but also expect that the existence of nonlinear kernels in this submodel can have an impact on the final output of BusterNet. Regarding the original architecture of BusterNet, it is important to emphasize its already high complexity. Simi - Det is made of 66, while Mani - Det of 50 layers. Thus, as a first experiment we have plugged our quadratic layer as the first layer of the Feature Extractor block in Mani - Det branch, to keep the complexity as inexpensive as possible.

The researchers provide BusterNet's layers, thus for starting our training process we have built four individual models, using the provided layers in combination with our Quadratic Convolutional layer. For a more detailed description of our training process, we have built and trained the QuadMani - Det branch, with its first layer being our quadratic layer and also the Simi - Det branch, as described in the publication. We have also built the QuadFusion module and plan to freeze the QuadMani and the Simi branches to train it. Finally, we plan to unfreeze the network and fine - tune QuadBusterNet end - to - end. We emphasize that with the quadratic layer plugged into the QuadMani - Det, only the Simi - Det branch will not be affected at all by our layer, since it does not share weights with the QuadManiDet - Branch.

Table 1 Number of parameters of Simi - Det, Mani - Det and QuadMani - Det submodels. Regarding the QuadMani, in this case our quadratic layer is the first layer of the model.

Branch Name	Trainable parameters	Non trainable parameters	Total number of parameters
Simi - Det	7,734,967	656	7,735,623
Mani - Det	7,789,617	132	7,789,749
QuadMani - Det	7,805,169	132	7,805,301

The Quadratic Manipulation Detection branch is a submodel that takes as input an RGB image of any size and after resizing it to (256,256,3), it extracts features by computing nonlinear interactions between the image pixels. It then upsamples the image to its original size and applies binary classification to produce a manipulation segmentation mask, distinguishing the detected manipulated object.

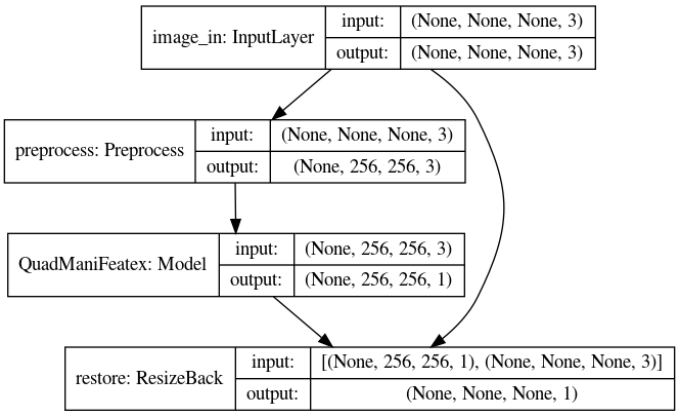


Figure 20 The Architecture of Quadratic Manipulation Detection model (QuadMani)

9.3. TRAINING DETAILS

The first step of the researchers' training process was the individual training of Mani-Det and Simi-Det branches. The cost function they use for these two tasks is the binary_crossentropy, while the optimizer they choose is Adam, with a learning rate of 0.01. For training the Fusion module they use the categorical_crossentropy loss function which they optimize with Adam, with a learning rate of 0.01. Finally, for the last step of the training where they finetune BusterNet end - to - end, they also use categorical_crossentropy with Adam, but with a lower initial learning rate of 0.00001.

For our first set of experiments, we trained all four models by sticking to the training details as described in BusterNet's publication, including the cost functions, optimizer and early stopping after 20 epochs of no validation loss improvement callback. While monitoring the training history, we noticed that our models' validation losses stopped improving at an early epoch and the trainings ended quickly, because of the early stopping callbacks. This possibly means that the models got stuck to local minimas and the 20 epochs of patience before ending the training were not enough for escaping it. After removing the early stopping callbacks, the validation losses improved after approximately 150 epochs of no improvement, for all four models. The results we obtained were not satisfying, especially after taking into consideration the training time.

In order to avoid the fact that our models got stuck to local minimas, we set different training parameters. We have modeled a new set of experiments, where we replaced Adam with the standard Stochastic Gradient Descent with Momentum optimizer. We also set a learning rate of 0.01 for both the QuadMani and Simi - Det Branches, while a lower initial learning rate of 0.00001 for both the QuadFusion and the QuadBusterNet models. We base the optimizer decision on the argument that, despite the fact that Adam's strength relies on its faster convergence, a significant number of deep learning approaches report that SGD with momentum generalizes better, since it results to better validation loss than Adam's. For these concrete reasons we expected a better overall result for our second set of experiments.

To compare our QuadMani's results with standard BusterNet's QuadMani, we have also trained Mani - Det branch with the same training parameters, to use it as a baseline.

9.4. DATASETS

We have trained QuadMani and Mani on 80000 samples from the synthetic USCISI - CMFD dataset, containing manipulated images under copy - move attacks. For the evaluation of the models we use 10000 test samples from the USCISI - CMFD dataset, but also samples from the 1st Image Forensics Challenge, CoMoFoD, CASIA and PSBattles datasets.

9.5. EVALUATION

The metric we use for evaluating QuadMani and Mani is the Dice coefficient, which is also often implemented as a loss function for training image segmentation models. Here, we use the Dice coefficient for the comparison of the predicted and the ground truth segmentation masks, which can be computed by:

$$\frac{2 * |X \cap Y|}{|X| + |Y|},$$

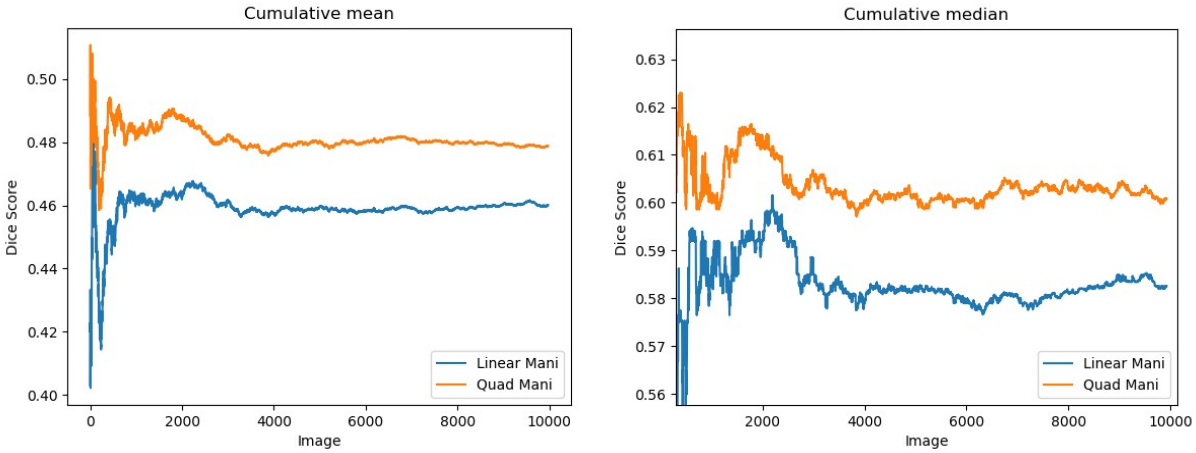
where X is the cardinality of the predicted segmentation mask and Y the cardinality of the ground truth segmentation mask. We compute the mean and median values of the Dice coefficients of each image in the USCISI - CMFD test set.

We note that the ground truth masks used as labels for training QuadMani and Mani are binary, distinguishing the target (manipulated/cloned) object. The respective submodels' predicted masks are one channel images, where each pixel is a value in range [0,1], indicating the probability of the specific pixel belonging to the foreground class (manipulated object).

9.6. RESULTS

In this section we demonstrate our results, acquired after evaluating both Mani and QuadMani - Det branches of BusterNet on manipulated test images.

After obtaining the predicted masks, we convert them to binary by applying a threshold, to only let pixels with a high fakeness probability affect the Dice scores. We have experimented with different values, but concluded to a threshold of 0.8. By adopting this evaluation protocol, some of the predicted segmentation masks that are not confident enough on their per pixel predictions, will get a zero Dice score when compared to the ground truth, for both the baseline Mani and the QuadMani submodels. This suggests that we do not make a decision on the predicted masks for which the models may have approximately or partially localized a tampered area with low probabilities (<0.8). The threshold of 0.8 resulted to a zero Dice score for 26% of the QuadMani's predicted masks and 30% of the baseline Mani's predicted masks. As a result, we have made decisions on the rest of the test images that have actually affected the mean and median values of the Dice scores.



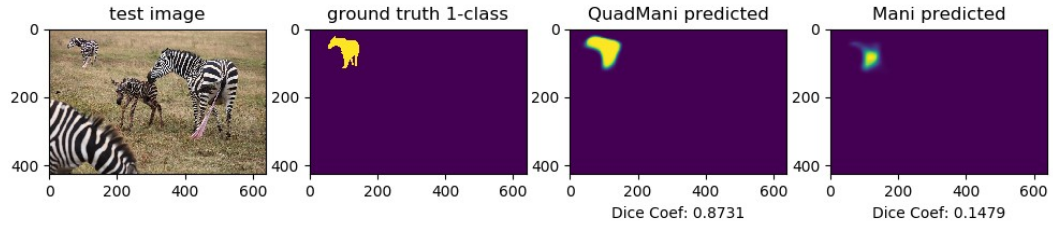
a) b)

Figure 21 Cumulative Mean (a) and Cumulative Median (b) of QuadMani and Mani Dice scores

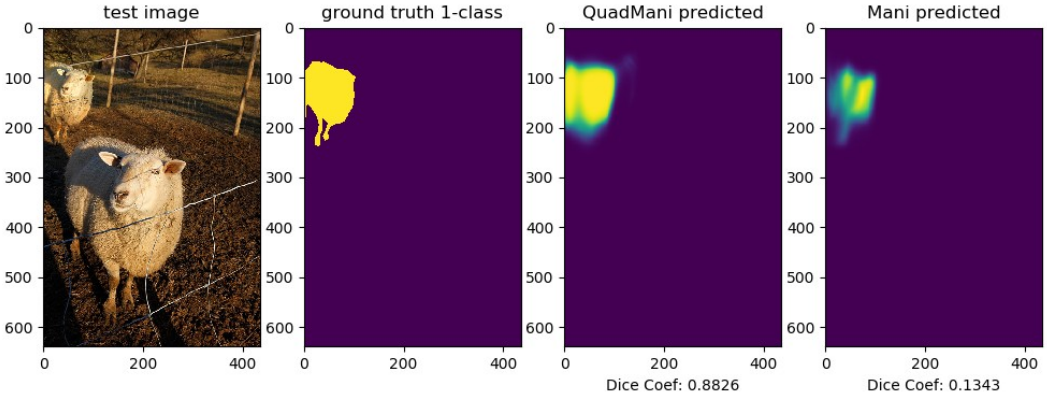
Table 2 Mean and Median values of QuadMani and Mani Dice scores

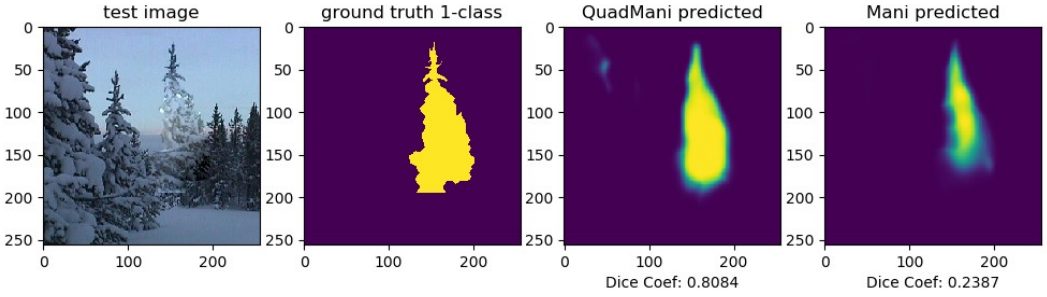
	Mean Dice score	Median Dice Score
QuadMani - Det	0.4788	0.6008
Mani - Det	0.4601	0.5825

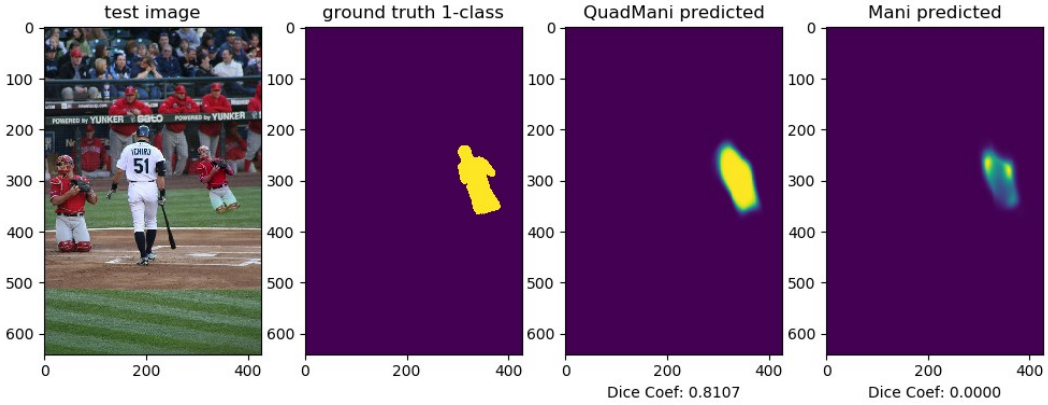
QuadMani resulted to better segmentation masks for 59.8% of the 10000 test images from the USCISI - CMFD dataset. In Figure 22 we demonstrate cases where QuadMani has predicted a better segmentation mask, in comparison with the baseline Mani's prediction and the ground truth mask of the test image, to illustrate QuadMani's strengths. The computed Dice scores for each predicted mask are also included in the figure. We note that the predicted masks that are illustrated in the next figures are not thresholded, meaning that only the bold yellow pixels were the ones that affected the Dice scores. The bolder the pixels, the highest the probability that the pixel is manipulated.


test image
ground truth 1-class
QuadMani predicted
Mani predicted
Dice Coef: 0.8731
Dice Coef: 0.1479


test image
ground truth 1-class
QuadMani predicted
Mani predicted
Dice Coef: 0.8723
Dice Coef: 0.0323


test image
ground truth 1-class
QuadMani predicted
Mani predicted
Dice Coef: 0.8826
Dice Coef: 0.1343


test image
ground truth 1-class
QuadMani predicted
Mani predicted
Dice Coef: 0.8084
Dice Coef: 0.2387


test image
ground truth 1-class
QuadMani predicted
Mani predicted
Dice Coef: 0.8107
Dice Coef: 0.0000

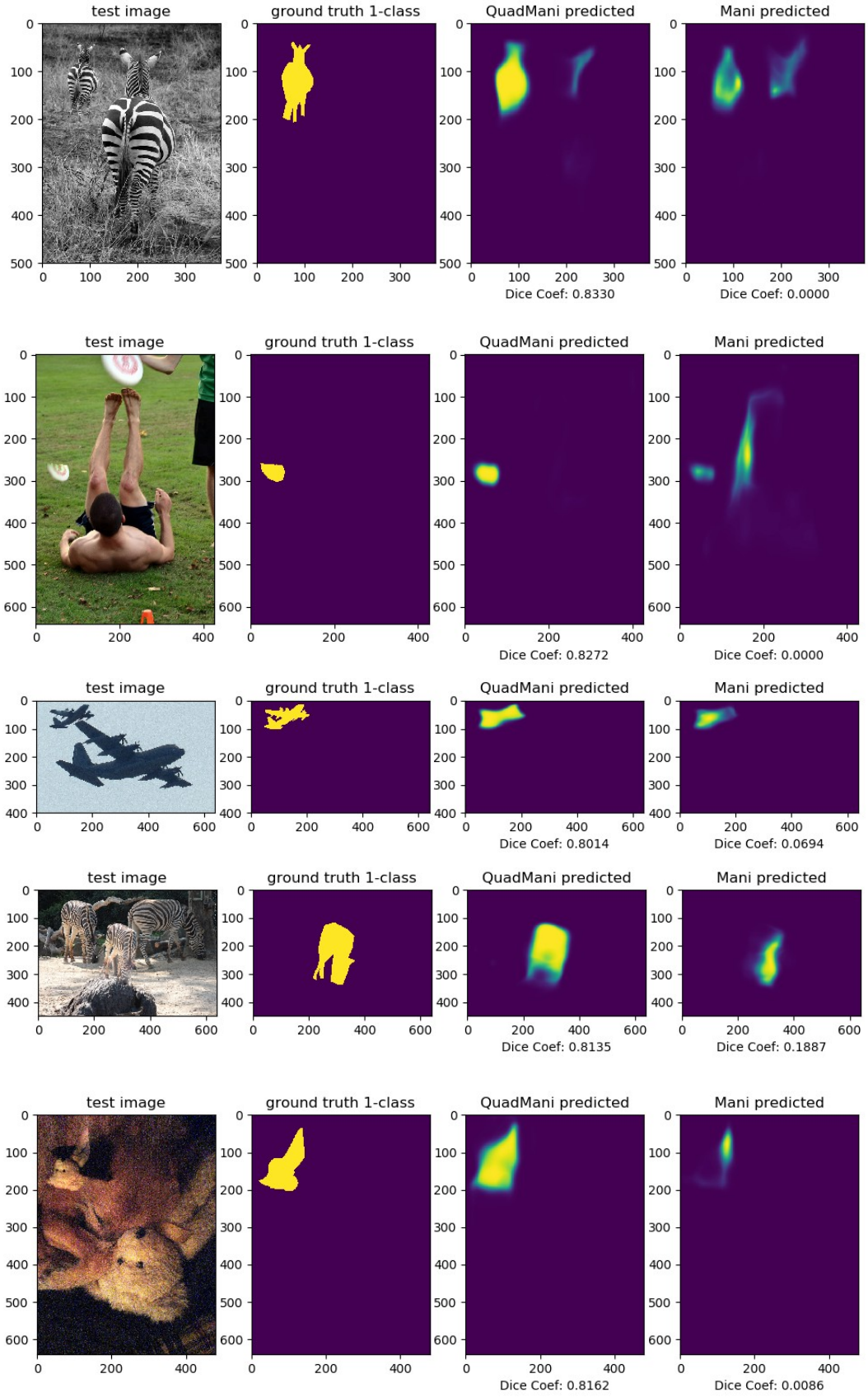
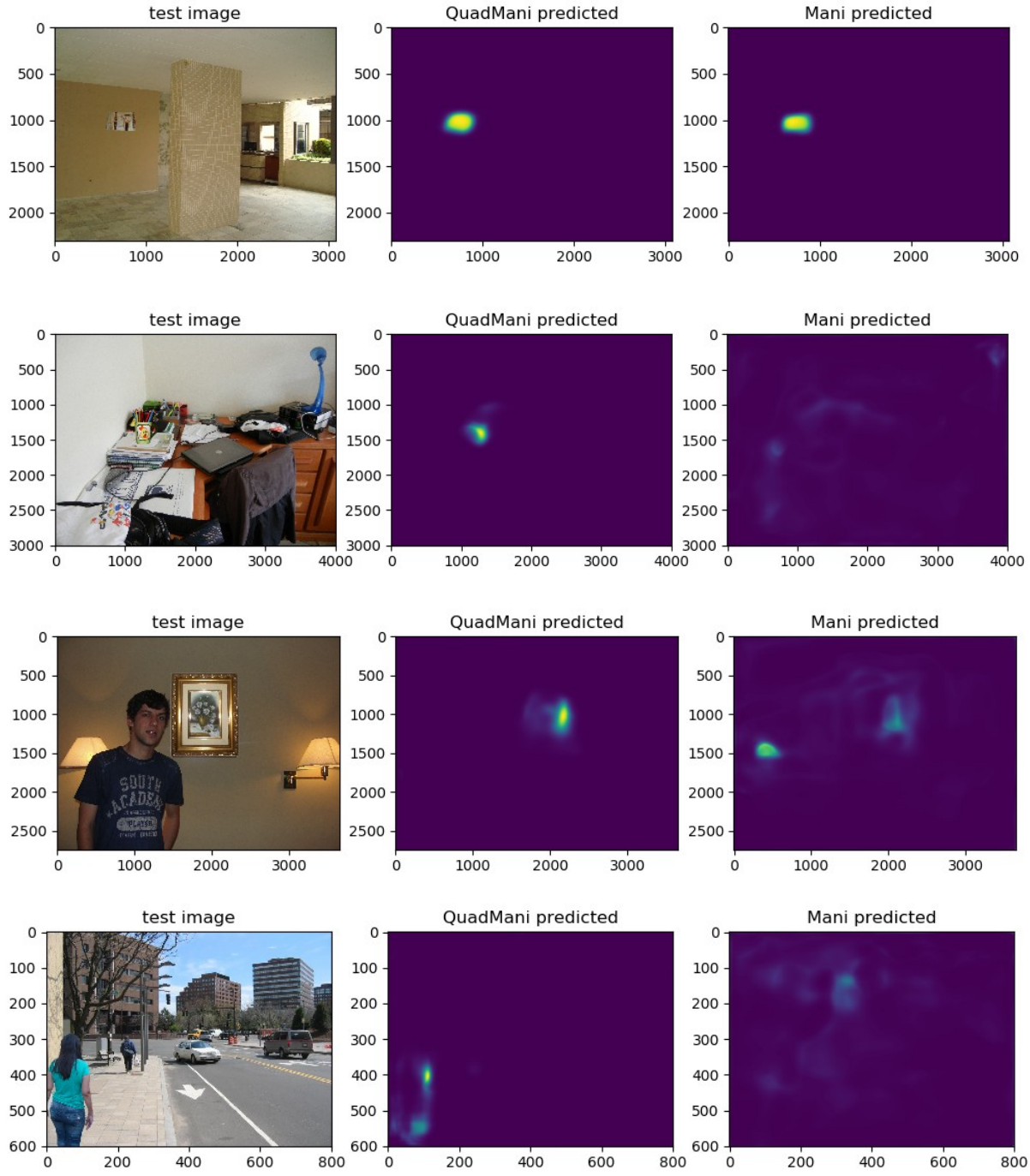
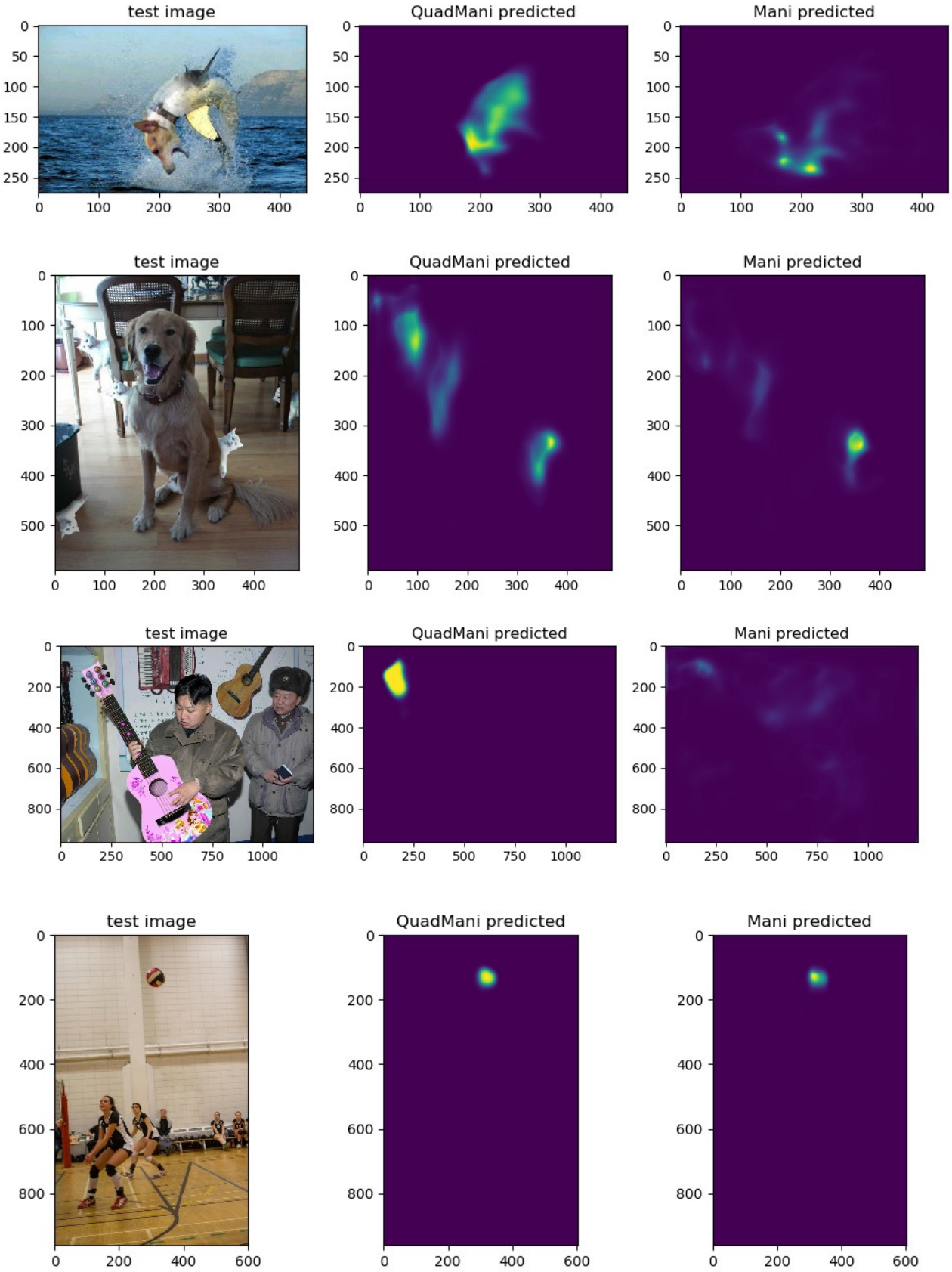


Figure 22 Qualitative results of QuadMani and baseline Mani on USCISI - CMDF test images

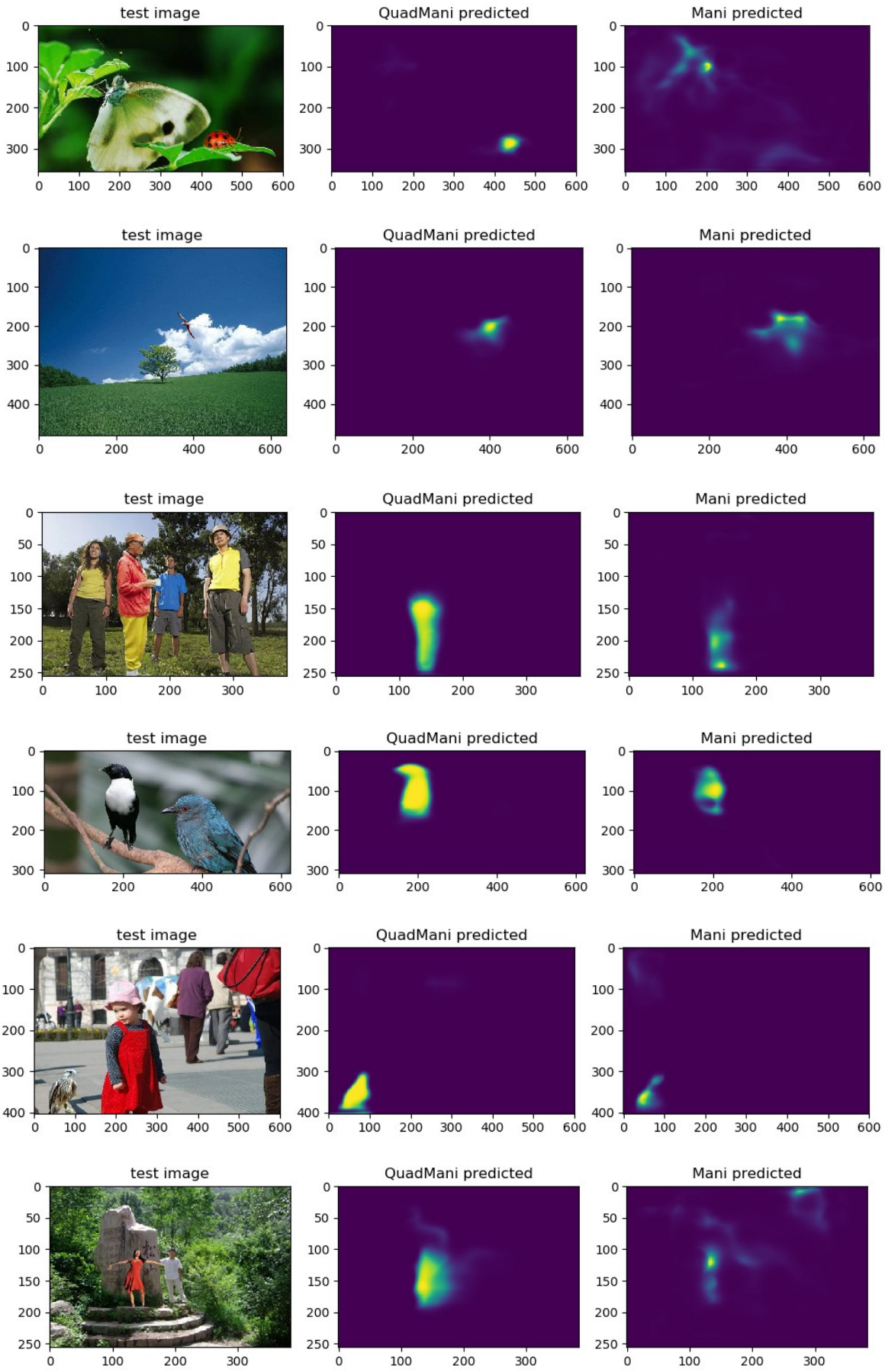
We notice quite big differences between the two models' predictions. It is clear that QuadMani's predictions are covering a bigger surface of the detected manipulated object in most of the cases, while its decisions are unquestionably confident for most of the object's pixels. As a preliminary conclusion, QuadMani results to better localization of the fake object, where its boundaries are well distinguished. To further explore QuadMani's efficiency and determine its ability to generalize on different manipulation attacks, we also evaluate the two models on manipulated images from the 1st Image Forensics Challenge, Photoshop Battles and CASIA datasets.



a)


test image
QuadMani predicted
Mani predicted
test image
QuadMani predicted
Mani predicted
test image
QuadMani predicted
Mani predicted
test image
QuadMani predicted
Mani predicted

b)



c)

Figure 23 Qualitative results of QuadMani - Mani on a) 1st Image Forensics Challenge dataset, b) Photos from Photoshop Battles dataset, c) CASIA dataset

9.7. Comments And Future Work

Based on the qualitatives results shown in Figures 20 and 21, we conclude that QuadMani has a strong ability to localize almost all the pixels of the fake object, regarding the images of the USCISI - CMFD test dataset. The boundary pixels are the only ones that get a low fakeness probability and as a result, the detected fake objects are well - defined and distinguished in the resulting segmentation masks. Regarding Figure 23, QuadMani was not equally confident, but still covered a bigger surface of the detected fake object than the baseline Mani's and in most of the cases it preserved its confidence on the prediction of fake pixels.

We should also note that in the case of QuadMani, 26% of the test images did not impact the mean and median values of the Dice scores, due to the hard threshold we applied. The same percentage is at 30% for the baseline Mani, suggesting that a larger number of images did not affect the mean/median values of the computed Dice scores. To further analyze this, a 30% of the masks that the baseline Mani predicted, consisted of uncertain probabilities for their pixels belonging to the foreground class, i.e. on an image level, the baseline Mani was uncertain about the existence of a fake object. On the other hand, QuadMani was certain for a larger number of the test images. It is common for image segmentation models to produce many false positive alerts, regarding manipulated image areas. Consequently, the high confidence that QuadMani demonstrated on its predictions is a feature that image segmentation models need, since these misclassified pixels could be constrained.

To sum up, we have experimented with the introduction of nonlinear convolution filters in the Manipulation Detection branch of BusterNet and validated our hypothesis that a nonlinear segmentation model would derive more detailed and accurate segmentation masks. We have created the novel QuadMani branch, which is an independent model that can be used for the detection of manipulated areas on images. Our experiments have verified that QuadMani has the ability to extract richer image features and exploit them to learn more complex interactions between image pixels. As a result, QuadMani has managed to produce segmentation masks that precisely and confidently localize the manipulated object area. The elimination of false positives on both pristine and manipulated images is a hard task, however, QuadMani's confident predictions take the precise detection of manipulated image areas one step further. Having confirmed the strengths of QuadMani, we will also examine whether they can affect the final output of BusterNet. Thus, as a future work we plan to train the QuadFusion module and finally fine tune QuadBusterNet end - to - end, to explore the impact of nonlinear kernels on a CMFD segmentation model.

10. CONCLUSION

In conclusion, in the context of this task's description, we demonstrate a collection of state of the art tools that can be used for the detection of manipulation attacks on images and videos. We describe the state - of - the - art deep learning products that deal with the detection of various manipulation methods, since the diversity of today's attacks cannot be faced with one single tool. We analyze each model's strengths and illustrate their behaviour on the detection and localization of manipulated image areas. We also provide a qualitative comparison of the different models' results on the same images, to better capture the particular cases where each model behaves well. Finally, we also introduce our novel approach for boosting image manipulation detection models by the introduction of nonlinear convolution kernels on the Manipulation Detection Branch of BusterNet. We conclude that QuadMani, our proposed method, is a powerful segmentation model with the strength to confidently detect fake objects on images and thus can be used as an individual tool for the detection of image forgery. Having verified QuadMani's manipulation detection we aim to test its efficiency on the final output of BusterNet.

11. REFERENCES

1. Y. Wu, W. Abd-Almageed, and P. Natarajan, "BusterNet: Detecting Copy-Move Image Forgery with Source/Target Localization," in *Computer Vision – ECCV 2018*, vol. 11210, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 170–186.
2. S. Heller, L. Rossetto, and H. Schuldt, "The PS-Battles Dataset - an Image Collection for Image Manipulation Detection," Apr. 2018.
3. M. Huh, A. Liu, A. Owens, and A. A. Efros, "Fighting Fake News: Image Splice Detection via Learned Self-Consistency," *ArXiv180504096 Cs*, May 2018.
4. A. Owens and A. A. Efros, "Audio-Visual Scene Analysis with Self-Supervised Multisensory Features," *ArXiv180403641 Cs Eess*, Apr. 2018.
5. J. Fridrich, D. Soukal, and J. Lukáš, "Detection of Copy-Move Forgery in Digital Images," p. 10
6. S. Bayram, H. T. Sencar, and N. Memon, "An efficient and robust method for detecting copy-move forgery," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1053–1056.
7. D. Cozzolino, G. Poggi, and L. Verdoliva, "Efficient Dense-Field Copy–Move Forgery Detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 11, pp. 2284–2297, Nov. 2015.
8. D.-Y. Huang, C.-N. Huang, W.-C. Hu, and C.-H. Chou, "Robustness of copy-move forgery detection under high JPEG compression artifacts," *Multimed. Tools Appl.*, vol. 76, Dec. 2015.
9. S.-J. Ryu, M.-J. Lee, and H.-K. Lee, "Detection of Copy-Rotate-Move Forgery Using Zernike Moments," presented at the LNCS, 2010, vol. 6387, pp. 51–65.
10. B. Mahdian and S. Saic, "Detection of copy-move forgery using a method based on blur moment invariants.," *Forensic Sci. Int.*, vol. 171, no. 2–3, pp. 180–189, 2007.
11. T. Mahmood, T. Nawaz, A. Irtaza, R. Ashraf, M. Shah, and M. T. Mahmood, "Copy-Move Forgery Detection Technique for Forensic Analysis in Digital Images," *Mathematical Problems in Engineering*, 2016.
12. I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, "A SIFT-Based Forensic Method for Copy–Move Attack Detection and Transformation Recovery," *Inf. Forensics Secur. IEEE Trans. On*, vol. 6, pp. 1099–1110, Oct. 2011.
13. A. Costanzo, I. Amerini, R. Caldelli, and M. Barni, "Forensic Analysis of SIFT Keypoint Removal and Injection," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 9, pp. 1450–1464, Sep. 2014.
14. B. Yang, X. Sun, H. Guo, Z. Xia, and X. Chen, "A Copy-move Forgery Detection Method Based on CMFD-SIFT," *Multimed. Tools Appl*, vol. 77, no. 1, pp. 837–855, Jan. 2018.
15. V. T. Manu and B. M. Mehtre, "Detection of Copy-Move Forgery in Images Using Segmentation and SURF," in *Advances in Signal Processing and Intelligent Recognition Systems*, 2016, pp. 645–654.
16. B. L. Shivakumar and S. S. Baboo, "Detection of Region Duplication Forgery in Digital Images Using SURF," *Int. J. Comput. Sci. Issues*, vol. 8, pp. 199–205, Jul. 2011.

17. E. Silva, T. Carvalho, A. Ferreira, and A. Rocha, “Going deeper into copy-move forgery detection: Exploring image telltales via multi-scale analysis and voting processes,” *J. Vis. Commun. Image Represent.*, vol. 29, pp. 16–32, May 2015.
18. Y. Zhu, X. Shen, and H. Chen, “Copy-move forgery detection based on scaled ORB,” *Multimed. Tools Appl.*, vol. 75, no. 6, pp. 3221–3233, Mar. 2016.
19. E. Ardizzone, A. Bruno, and G. Mazzola, “Copy–Move Forgery Detection by Matching Triangles of Keypoints,” *Inf. Forensics Secur. IEEE Trans. On*, vol. 10, pp. 2084–2094, Oct. 2015.
20. J. Li, X. Li, B. Yang, and S. Xingming, “Segmentation-Based Image Copy-Move Forgery Detection Scheme,” *IEEE Trans. Inf. Forensics Secur.*, vol. 10, pp. 507–518, Mar. 2015.
21. Chi-Man Pun, Xiao-Chen Yuan, and Xiu-Li Bi, “Image Forgery Detection Using Adaptive Oversegmentation and Feature Point Matching,” *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 8, pp. 1705–1716, Aug. 2015.
22. Y. Liu, Q. Guan, and X. Zhao, “Copy-move Forgery Detection Based on Convolutional Kernel Network,” *Multimed. Tools Appl*, vol. 77, no. 14, pp. 18269–18293, Jul. 2018.
23. J. Bunk *et al.*, “Detection and Localization of Image Forgeries Using Resampling Features and Deep Learning,” *2017 IEEE Conf. Comput. Vis. Pattern Recognit. Workshop CVPRW*, pp. 1881–1889, 2017.
24. Y. Wu, W. Abd-Almageed, and P. Natarajan, “Deep Matching and Validation Network: An End-to-End Solution to Constrained Image Splicing Localization and Detection,” in *Proceedings of the 25th ACM International Conference on Multimedia*, New York, NY, USA, 2017, pp. 1480–1502.
25. P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, “Two-Stream Neural Networks for Tampered Face Detection,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1831–1839.
26. R. Arandjelović and A. Zisserman, “Objects that Sound,” *ArXiv171206651 Cs Eess*, Dec. 2017.
27. R. Gao, R. Feris, and K. Grauman, “Learning to Separate Object Sounds by Watching Unlabeled Video,” in *Computer Vision – ECCV 2018*, vol. 11207, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 36–54.
28. Y. Wu, W. AbdAlmageed, and P. Natarajan, “ManTra-Net: Manipulation Tracing Network for Detection and Localization of Image Forgeries With Anomalous Features,” The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
29. G. Zoumpourlis, A. Doumanoglou, N. Vretos, and P. Daras, “Non-linear Convolution Filters for CNN-based Learning,” *ArXiv170807038 Cs*, Aug. 2017.
30. “Volterra-based convolution filter implementation in Torch – Visual Computing Lab.” .
31. S.-Y. Wang, O. Wang, A. Owens, R. Zhang, and A. A. Efros, “Detecting Photoshopped Faces by Scripting Photoshop,” *ArXiv190605856 Cs*, Jun. 2019.
32. A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, “FaceForensics: A Large-scale Video Dataset for Forgery Detection in Human Faces,” *ArXiv180309179 Cs*, Mar. 2018.
33. A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, *FaceForensics++: Learning to Detect Manipulated Facial Images*. 2019.

34. D. Moreira *et al.*, “Image Provenance Analysis at Scale,” *ArXiv180106510 Cs*, Jan. 2018.
35. J. Dong, W. Wang, and T. Tan, “CASIA Image Tampering Detection Evaluation Database,” in *2013 IEEE China Summit and International Conference on Signal and Information Processing*, 2013, pp. 422–426.
36. D. Tralic, I. Zupancic, S. Grgic, and M. Grgic, “CoMoFoD - New Database for Copy-Move Forgery Detection,” *Th Int. Symp. ELMAR*, p. 6, 2013.
37. Davis E. King. Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research 10, pp. 1755-1758, 2009.
38. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).