Detection and Localization of Facial Expression Manipulations

Ghazal Mazaheri
Video Computing Group
University of California, Riverside
gmaza002@ucr.edu

Amit K. Roy-Chowdhury
Video Computing Group
University of California, Riverside
amitrc@ece.ucr.edu

Abstract

Concerns regarding the wide-spread use of forged images and videos in social media necessitate precise detection of such fraud. Facial manipulations can be created by Identity swap (DeepFake) or Expression swap. Contrary to the identity swap, which can easily be detected with novel deepfake detection methods, expression swap detection has not yet been addressed extensively. The importance of facial expressions in inter-person communication is known. Consequently, it is important to develop methods that can detect and localize manipulations in facial expressions.

To this end, we present a novel framework to exploit the underlying feature representations of facial expressions learned from expression recognition models to identify the manipulated features. Using discriminative feature maps extracted from a facial expression recognition framework, our manipulation detector is able to localize the manipulated regions of input images and videos. On the Face2Face dataset, (abundant expression manipulation), and Neural-Textures dataset (facial expressions manipulation corresponding to the mouth regions), our method achieves higher accuracy for both classification and localization of manipulations compared to state-of-the-art methods. Furthermore, we demonstrate that our method performs at-par with the state-of-the-art methods in cases where the expression is not manipulated, but rather the identity is changed, leading to a generalized approach for facial manipulation detection.

1. Introduction

Facial expressions are critical in communicating our thoughts, ideas, emotions, and in responding to each other emotionally and physically. With effectual facial expressions, a person may convince others to believe in ideas without verbal communication. Due to the power of facial expressions in person-to-person communication, it is critical to determine if the facial expressions in an image or video are the individual's original expressions or manipulated by an external agent. Facial manipulations can be created by Identity swap (DeepFake) or Expression swap. In this work, *we focus on the problem of detecting facial expression manipulations while ensuring that performance does not degrade for the identity manipulation case*, thus providing a *generalized version of facial manipulation detection*.

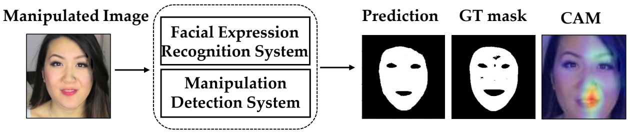

Manipulated Image
Facial Expression Recognition System
Manipulation Detection System
Prediction
GT mask
CAM

Figure 1: Our overall framework for detection of facial expression manipulation and localization. Manipulated mage is from Face2Face dataset [1]. Class activation map (CAM) is the visualization of feature map from facial expression recognition system.

Facial expression changes in the Face2Face dataset is a result of a facial reenactment system that transfers the expressions of a source video to a target video while maintaining the identity of the target person. Similarly, Neural-Textures [2] reenacts face motions of an input video to a target video mainly affecting the regions around the mouth. We hypothesize that to detect such facial expression manipulations, recognition of the expression would be helpful. Based on this key idea, we design a framework called the Expression Manipulation Detection (EMD) system. Fig. 1 presents this key idea of using facial expression recognition to guide the manipulation detection procedure. As can be seen in the figure, the main manipulations appear in the parts of the face which constitute expression change, such as, regions around eyebrows and mouth which are critical regions for facial expressions.

In order to exploit prominent features corresponding to facial expression, we use Facial Expression Recognition (FER) systems in our face manipulation detection framework (see Sec. 3.3 for details). In particular, we adopt Ensemble with Shared Representations (ESR) [3] as the backbone network of FER system. Feature maps from the penultimate layer of FER systems contain important information regarding facial expressions in faces [4], which we aim to exploit in order to improve over state-of-art manipulation detection methods.

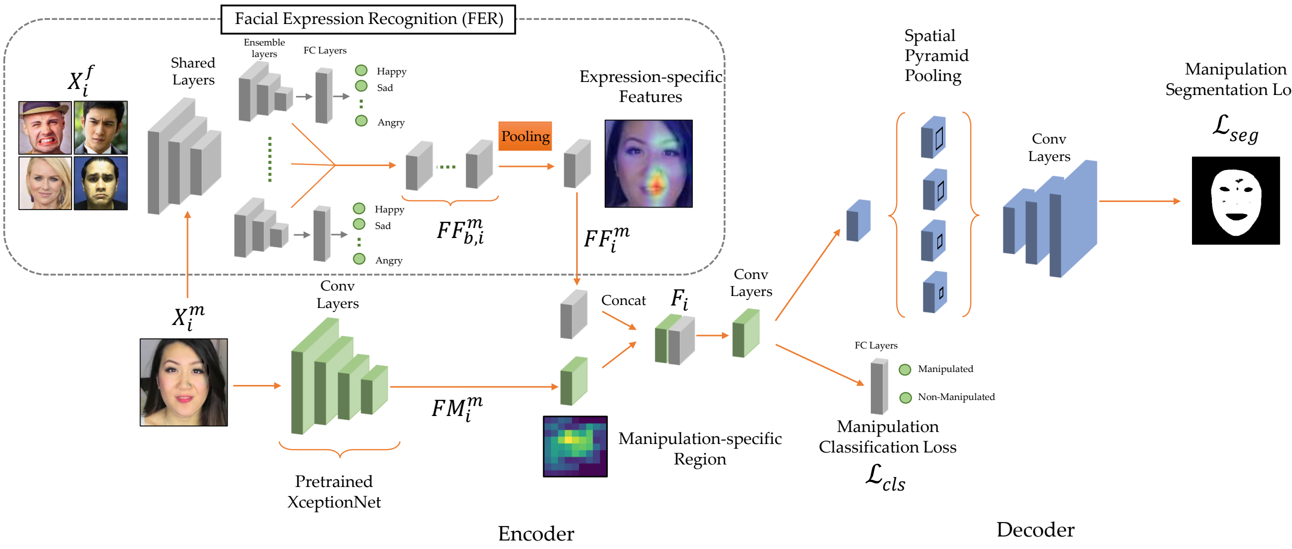


Figure 2: This figure represents our proposed approach for facial expression manipulation detection and localization. Extracted features from FER System (FF_i^m), along with the ones from manipulation detection stream (FM_i^m), are fed into the decoder for pixel-wise localization of the manipulated region. Notation is described in the text. The details are explained is Sec. 3.3 and 3.4

1.1. Framework Overview

A pictorial illustration of our facial expression manipulation detection (EMD) framework is presented in Fig. 2. EMD utilizes a two-stream network for manipulation detection. One stream (FER) is responsible for extracting important information for facial expressions. The feature maps from the last layer of FER stream provide information about the facial regions that encode the expression information. The other stream is an encoder-decoder architecture which is responsible for manipulation detection. The encoder projects the image to a lower dimensional space, where the features from the FER system are combined and then a decoder is used to predict the manipulated regions (if any) of the facial image.

Our FER system uses ESR [3] to extract expression relevant features. The penultimate layer feature map of the FER system contains features which are discriminative to detect relevant portions of the image specific for expression.

The second stream, i.e., the manipulation detection and segmentation system is an encoder-decoder architecture. The decoder takes the latent space features from the encoder and FER system combined and projects it using DeepLabv3+ [5] for manipulation localization. (see Fig. 2).

Main contributions. We propose a novel approach for facial expression manipulation detection leveraging upon a facial expression recognition system. This leads to higher performance in forgery detection where the facial expression is manipulated, as well as localizing the regions that have been manipulated. Our system achieves at par performance in the case where the identity is manipulated, ensuring generalizability of our method. Our method leads to more than 3% improvement in manipulation classification and localization over the state of the art on the Face2Face dataset [1] where the expressions are manipulated. We also show the effectiveness of our method by presenting the results on NeuralTextures dataset [2] where the facial expressions corresponding to the mouth regions have been modified. On NeuralTextures dataset, we achieve 2% higher accuracy for classification and localization. Finally, our framework achieves competitive results compared to the state-of-the-art methods on DeepFake dataset [6] and DFDC dataset [7] where identity, rather than expression, is manipulated.

2. Related Works

Multimedia forensics aims to ensure authenticity, origin, and provenance of an image or video. In recent years, there has been a variety of works in forgery classification and forgery localization. We will briefly survey existing work in both the mentioned categories, as well as facial expression recognition. There is no work that specifically focuses on the problem of detection and localization of facial expression manipulations.

2.1. Forgery Classification

In forgery classification area, there has been a variety of works in image manipulation detection [8, 9, 10, 11, 12, 13, 14] or fake faces classification in videos [15, 16, 17, 18].

Manipulation of faces in images/videos has been in the news lately. Manipulation detection in faces is challenging since exiting manipulation techniques leave almost no visual traces. In a recent work [17], authors utilize an attention mechanism to improve the feature maps for the classification task, which motivates our proposed approach for using the facial expression recognition system.

To detect face manipulation in videos, some approaches utilize video temporal features to tamper individual frames in videos causes inconsistency. The work in [19] uses CNN as feature extractor and LSTM to capture video temporal features. Some other works use physiological signals, like eye blinking in [20] and head movements in [21], that are not well presented in the synthesized fake videos. Instead of using temporal features, authors in [15, 16, 22, 23] proposed methods that utilize images from different frames of a video. Work in [24] proposed an approach based on visually exposed features in a manipulated face.

2.2. Forgery Localization

Localizing the exact position of manipulated regions in an image or video provides critical additional information. There has been a variety of works that attempted to segment out tampered regions [25, 26, 27]. Early works [28, 29, 30] reveal the tampered regions using traditional image processing-based approaches. Researchers in [26, 31, 32, 33, 34] exploit machine learning techniques in order to classify if a patch is manipulated or not.

Fake face segmentation is one of the recent challenges which has not yet been addressed extensively. Some of the proposed methods may have high performance in face manipulation detection [15] but do not address the task of segmenting the manipulated region. Multi-tasking approaches are promising in the combined classification and segmentation task. Work in [23] uses Y-shape architecture to classify manipulated videos and segment tampered faces simultaneously. In our proposed method, in addition to manipulation classification and segmentation stream, we add another stream as face expression recognition which operates jointly with the manipulation stream in order to exploit necessary information in faces. This improves the performances in both classification and segmentation.

2.3. Facial Expression Recognition (FER)

The development of machine learning and the advent of deep learning have significantly improved the research of FER. There have been variety of works in literature which obtain high performance for facial expression recognition [35, 36, 37, 3, 38, 39, 40]. The careful design of local to global feature learning with a convolution, pooling, and layered architecture produces a rich visual representation, making CNN a powerful tool for facial expression recognition. Research challenges such as Kaggle's Facial Expression Recognition Challenge suggest the growing interest in the use of deep learning for the solution of this problem.

To have more accurate FER, the networks become deeper and deeper in order to deal with more complex classification tasks. Also, attention mechanisms are introduced in many networks to improve facial expression recognition. Authors in [41] proposed a facial expression recognition network with the visual attention mechanism. Work in [3] is one of the most recent in this area showing promising results on facial expression datasets.

3. Methodology

In this section, we present our framework for facial expression manipulation detection and localization. We start with a formal description of the problem statement followed by the two streams of our framework - Facial Expression Recognition (FER) stream and the encoder-decoder based manipulation detection stream which receives information from FER for better detection.

3.1. Problem Statement

Consider we have a dataset of tuples $\mathcal{X}^M = \{(X_i^m, M_i^m, y_i^m)\}_{i=1}^N$, where $X_i^m \in \mathbb{R}^{H\times W\times 3}$ is a 2D image of faces, $M_i \in \mathbb{R}^{H\times W}$ is 2D binary mask of manipulated regions, and $y_i \in \{0, 1\}$ is an indicator whether y_i^m is manipulated or not. Given such a dataset, our main goal is to learn a model that would be able to classify a test image to be either manipulated or not, and more importantly, localize portions of the image which are manipulated.

To identify manipulations in facial expressions, we need to focus on regions specific for expressions; thus, we utilize an auxiliary task of Facial Expression Recognition (FER). We use a dataset of tuples $\mathcal{X}_F = \{(X_i^f, y_i^f)\}_{i=1}^{N'}$, where $X_i^f \in \mathbb{R}^{H\times W\times 3}$ and $y_i^f \in \{1, \dots, C\}$, and C is the number of facial expression categories. Note that we use the superscripts m and f to denote data points from the manipulation set $\mathcal{X}^M$ and facial expression set $\mathcal{X}^F$ respectively.

3.2. Algorithm Overview

Our EMD system consists of two main parts including FER and encoder-decoder. We train the FER module using the dataset $\mathcal{X}^F$. To train the encoder-decoder architecture for manipulation detection, our framework takes information from the FER module. However, we pass the images in $\mathcal{X}^M$ through both the streams - an encoder to obtain features necessary to detect manipulations, and an FER module to obtain features specific to facial expressions. We combine these features in the latent space and then pass them through a decoder to spatially localize the manipulated regions.

3.3. Facial Expression Recognition

We utilize one of the state-of-the-art methods for facial expression recognition proposed in [3] as a pre-trained model for recognizing the facial expressions.

FER system presented in [3] consists of two building blocks. The base of the network (shared layers in Fig. 2) is an array of convolutional layers for low- and middle-level feature learning. These informative features are then shared with independent convolutional branches that constitute the ensemble (ensemble layers in Fig. 2). From this point, each branch can learn distinctive features while competing for a common resource - the shared layers. This competitive training emerges from the minimization of a combined loss function defined as the summation of the loss functions (cross-entropy loss) of each branch as follows:

$$\mathcal{L}_{FER} = \frac{1}{N'} \sum_{b} \sum_{i} \sum_{c} -y^{f}_{i,c} \log(p^{f}_{b,i,c}) \tag{1}$$

where given an image X^f_i, $p^f_{b,i,c}$ is a probability mass function over the C facial expression categories for branch b. N' is total number of images in the dataset.

After training the FER system, we use the pre-trained models to predict the expression category of manipulated images and extract the feature maps needed to be combined with the manipulation detection stream. The reason is that, for manipulation detection task, feature maps which highlight the discriminative image regions important for expression recognition are useful for manipulation detection of images where facial expressions are tampered such as manipulation in Face2Face dataset. Therefore, our FER architecture provides useful expression and location-aware features needed for the manipulation detection task.

In the Facial Expression Recognition system, we pass an image X^m_i through the shared convolutional network $Conv$ and an ensemble of B convolutional networks $\{\mathcal{E}_b\}^B_{b=1}$ to generate B different feature maps of the facial expression. For the b^{th} ensemble convolutional branch, the feature map corresponding to X^m_i is,

$$FF^m_{b,i} = \mathcal{E}_b(Conv(X^m_i)). \tag{2}$$

We use a classifier $\mathcal{M}$ to infer class probabilities for the C expression classes. For b^{th} branch network and i^{th} image X^m_i, we infer the class probability vector, $\boldsymbol{\rho}_{b,i} = [\rho_{b,i,1}, \ldots, \rho_{b,i,C}]$. Here, $\rho_{b,i,j} = \mathcal{M}(FF^m_{b,i}, c_j)$ indicates the detection probability of class c_j for branch b with input image X^m_i. Therefore, the detected expression class of an image from b^{th} convolutional branch is

$$c^b_{det} = \operatorname*{arg\,max}_{c_j \in \{c_1,\ldots,c_C\}} \mathcal{M}(FF^m_{b,i}, c_j). \tag{3}$$

Considering the detection of all the branches, most frequent detected class for an image is,

$$c_{freq} = mode(\{c^1_{det}, \ldots, c^B_{det}\}) \tag{4}$$

The feature map from the branch in FER network that results in highest detection probability for the frequently detected class c_{freq} is pooled for manipulation detection task. So, the pooled feature map is

$$FF^m_i = \operatorname*{arg\,max}_{FF^m_{b,i} \in \{FF^m_{1,i},\ldots,FF^m_{B,i}\}} \mathcal{M}(FF^m_{b,i}, c_{freq}) \tag{5}$$

As will be discussed subsequently, we use the feature map FF^m_i after convolutional layers as auxilliary input to the encoder-decoder for manipulation detection. As illustrated later in Fig. 5, we also obtain the class activation maps (CAMs) for visualization purpose following [4].

3.4. Encoder-Decoder

Encoder-decoder networks using CNN architecture have been extensively used in deep learning literature, specifically for semantic object segmentation. Following the literature, we adopt Encoder-Decoder architecture [5] known as deeplabv3+ for manipulation detection and segmentation as the task of localizing manipulation regions is similar to semantic segmentation task.

Given an image X^m_i, we pass it through the encoder to obtain FM^m_i from one layer before the last convolutional layer. As we are interested in detecting manipulations in expression, we inject features from the facial expression recognition stream into the encoder-decoder manipulation detection stream. To do that, we also pass X^m_i through FER and obtain features FF^m_i. We then concatenate both the feature maps as $F_i = FM^m_i \oplus FF^m_i$ and pass the concatenated feature maps through remaining layers of encoder to obtain latent space features. As shown in Fig. 2, we have two loss functions for classification ($\mathcal{L}_{cls}$) and segmentation ($\mathcal{L}_{seg}$). We use cross-antropy loss function for classification task defined as follows:

$$\mathcal{L}_{cls} = \frac{1}{N} \sum_{i} \| y^m_i \log(a_i) + (1 - y^m_i) \log(1 - a_i) \|_1, \tag{6}$$

where a_i is the output of binary classification which determines whether or not an image is manipulated.

Next, the decoder takes the latent space features as the input. Consider that $S_i \in \mathbb{R}^{H \times W}$ is the spatial manipulation segmentation output. We compute the segmentation loss function to measure the agreement between the segmentation mask and the ground-truth mask as follows:

$$\mathcal{L}_{seg} = \frac{1}{N} \sum_{i} \| M^m_i \log(S_i) + (1 - M^m_i) \log(1 - S_i) \|_1, \tag{7}$$

Note that each pixel in S_i lies in between 0 and 1 depicting the probability of it being manipulated or not.

The total loss function we optimize to learn the encoder decoder architecture is as follows:

$$\mathcal{L}_{MANI} = \mathcal{L}_{cls} + \mathcal{L}_{seg} \tag{8}$$

3.5. Overall Algorithm

Here we discuss the overall training strategy for EMD algorithm. This is presented in Algorithm 1. Consider that the FER network is parameterized by ϕ and the the encoder-decoder for manipulation detection is parameterized by θ. We learn them separately. First we sample images from the facial expression dataset $\mathcal{X}^F$, compute the loss $\mathcal{L}_{FER}$ and update ϕ using it. We then sample images from the manipulated images dataset $\mathcal{X}^M$, pass them through both pretrained FER stream and encoder-decoder stream, compute the loss $\mathcal{L}_{MANI}$ and then update θ.

Algorithm 1 Overall EMD Algorithm

```
Inputs: 1. Expression Recognition Dataset: $\mathcal{X}^F$
        2. Expression Manipulation Dataset: $\mathcal{X}^M$
Output: Manipulation Detection Network: $\theta$
Random Init.:
        1. Facial Expression Recognition Net: $\phi$
        2. Manipulation Detection Net: $\theta$
while $notconverged$ do
    Mini-batch $B^f = \{X_i^f, y_i^f\}_{i=1}^B \sim \mathcal{X}^F$
    Compute: $\mathcal{L}_{FER}(B^f; \phi)$
    Update: $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}_{FER}$
while $notconverged$ do
    Mini-batch $B^m = \{X_i^m, M_i^m, y_i^m\}_{i=1}^B \sim \mathcal{X}^M$
    Compute: $\mathcal{L}_{MANI}(B^m; \theta, \phi)$
    Update: $\theta \leftarrow \theta - \eta' \nabla_\theta \mathcal{L}_{MANI}$
```

4. Experiments

In this section, we perform extensive experiments on three benchmark datasets from FaceForensics++ [42] to investigate the efficacy of the proposed method. We show results on two datasets (Face2Face and NeuralTextures) where the images correspond to facial expression manipulation and also two datasets (DeepFake and DFDC) where the images undergo an identity change.

4.1. Datasets

FaceForensics++ Dataset. For our experiments, we used the videos offered by FaceForensics++ [15] [1]. FaceForensics++ (FF++) contain 1,000 real videos and 1,000 Fake videos for each type of manipulation including Face2Face (F2F), DeepFake (DF) and NeuralTextures (NT). For each category of real/fake videos, the dataset was split into 720 videos for training, 140 for validation, and 140 for testing. We used videos with light compression (quantization = 23) and high compression (quantization = 40). Images were extracted from videos using the settings in [43]: 200 frames of each training video were used for training, and 10 frames of each validation and testing video were used for validation and testing respectively.

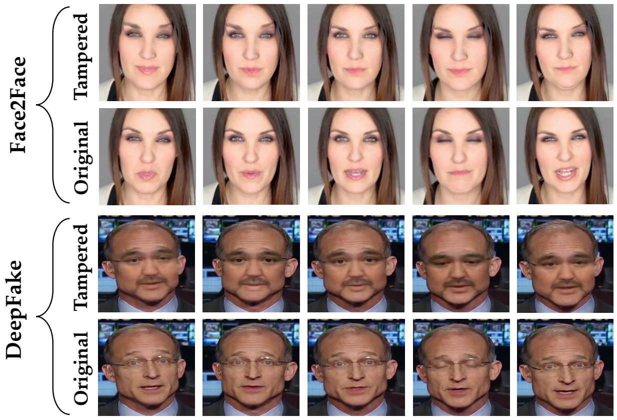


Figure 3: Two examples of pristine videos and their manipulated versions from F2F and DF datasets. As we can see, facial expression is manipulated in F2F videos while in DF datasets identities are swapped.

Table 1: Benchmark Datasets for face manipulation. Our focus in this paper is on second row, i.e., detecting expression manipulations, while not degrading performance on the first row (identity manipulations).

Manipulation	Method	Dataset
Identity Swap	FaceSwap, DeepFakes	UADFV, DF-TIMIT, DFD, CelebDF, DFDC, Deeper Forensics 1.0, FF++ (FS and DF)
Expression Swap	Face2Face, NeuralTextures	FF++ (F2F and NT)

DeepFake Detection Challenge Dataset (DFDC). Deepfakes are a recent off-the-shelf manipulation technique that allows anyone to swap two identities in a single video. In addition to deepfakes, a variety of GAN-based face swapping methods have also been published. The problem of deepfake detection has received considerable attention, and this research has been stimulated with many datasets. The DFDC dataset [7], different from the DF dataset as part of FF++ described above, is by far the largest publicly-available face swap video dataset with over 100,000 total clips sourced from 3,426 paid actors, produced with several deepfake, GAN-based, and non-learned methods.

We summarize the benchmark datasets for face manipulation in Table 1. As Table 1 shows, all the deepfake datasets are based on face identity swapping. The only dataset that contains both identity and expression manipulation is FF++ [42]. In FF++, F2F and NT are the only

[1] https://github.com/ondyari/FaceForensics

manipulation techniques that change the facial expression, while two other techniques (FaceSwap and DeepFake) are based on identity change.

To demonstrate the difference between identity swap and expression swap, we show some examples from both categories. Fig. 3 shows 5 frames from 2 different face manipulation datasets (F2F and DF). As we can see, tampered videos from F2F are undergo expression manipulation. The shape of lips and eyebrows which contribute substantially to facial expressions is changed in most of the frames from F2F vidoes. To the contrary, the tampered videos from DF do not demonstrate any major expression change in comparison to original ones. Therefore, only two datasets (F2F and NT from FF++) satisfy the criteria and we evaluate the performance on those datasets. We also present the results on DF and DFDC datasets to demonstrate that our method performs at-par with the state-of-the-art methods in cases where the expression is not manipulated, but rather the identity is changed, thus ensuring generalizability of the approach. (Further details in Sec. 4.3.1)

Facial Expression Datasets. We use AffectNet [44], a new database of facial expressions. We trained our FER system on AffectNet training set and used F2F, NT, DFDC and DF datasets as manipulation detection datasets. AffectNet contains more than 1M facial images collected from the Internet. The dataset is divided into 11 facial expression categories - neutral, happiness, sadness, surprise, fear, disgust, anger, contempt, none, uncertain, and no-Face.

4.2. Implementation

In face forensics, faces play an important role and contain key features for manipulation detection. Therefore, instead of using the whole image, we extract the faces as a pre-processing step and only use the face regions to train the models. The FER system has shared and ensemble layers consisting of CNNs using 5x5 and 3x3 convolutional windows with the stride of 1. Following each convolutional layer is a batch normalization layer [45].

The encoder-decoder architecture consists of XceptionNet with separable convolution as encoder, spatial pyramid pooling and CNNs with 3x3 convolutional windows as decoder. We transfer XceptionNet to our task by replacing the final fully connected layer with two outputs. The other layers except last convolutional layer (the layer after feature concatenation) and fully connected layer are initialized with the ImageNet weights. To set up the inserted fully connected layer, we fix all weights up to the last convolutional layer and pre-train the network for 3 epochs. After this step, we train the network for 20 more epochs and choose the best performing model based on validation accuracy.

The framework is implemented on PyTorch. We trained the network using the ADAM optimizer [46] with a learning rate of 0.001, a batch size of 16, β of 0.9 and 0.999, and ϵ equal to 10^{-8}. The implementation consumes 11GB memory GPU and it takes 126hrs (5 days) for 20 epochs. For Encoder-Decoder (deeplabv3+ with pretrained Xception, fixed parameters) memory consumption and training time is the same with doubling of the batch size.

4.3. Quantitative Comparisons

Evaluation Metrics. In terms of evaluation metrics, we use classification accuracy for the manipulation detection, which represents how many test images are correctly classified. For segmentation tasks, we use 1) pixel-wise classification accuracy which indicates whether a pixel in an image is manipulated or not, and 2) IoU (Intersection over Union). The IoU is calculated for both foreground and background, and the two IoUs are averaged to get mean IoU (mIoU).

4.3.1 Results

Expression Swap. Table 2 shows the classification accuracy for the Face2Face (F2F) and NeuralTextures (NT) datasets (with expression swap) using two types of video quality (low quality (LQ) and high quality (HQ)). As may be observed, in terms of classification accuracy, EMD (our method) reaches the best performance on both datasets. In comparison to XceptionNet, our proposed method achieves $\sim 3\%$ and $\sim 2\%$ improvements in classification accuracy on F2F and NT datasets respectively.

We also add FER to MultiTask [23] architecture which leads to improvements of accuracy by $\sim 3\%$ and $\sim 1\%$ on F2F and NT datasets with high quality videos. Furthermore, we compare our method with more of the state-of-art methods on Face2Face dataset in terms of classification accuracy. Table 3 shows this comparison. As it is clear from Table 3, our method achieves higher classification accuracy. For localization task, Table 4 shows $\sim 3\%$ improvement of segmentation accuracy on low quality videos from F2F dataset and $\sim 2\%$ imporvement on NT dataset with the same video quality. We also achieved $\sim 5\%$ and $\sim 4\%$ improvement in mIoU on F2F and NT with low quality videos.

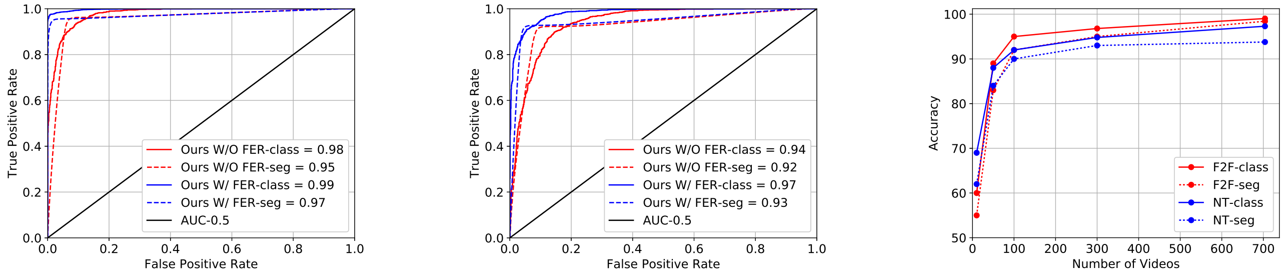
Identetity Swap. Table 2 also shows the classification accuracy for deepfake datasets where the identity is changed. We run experiments on two deepfake datasets including DFDC dataset [7] and DF [6]. Based on the results, our method achieves 89.16 % on DFDC while XceptionNet, as one the state-of-the-art methods, achieves 88.98 % in terms of classification accuracy. On DF dataset [6], we observe that with the addition of FER system, there is no fall in the performance. Thus, our method performs at-par with the state-of-the-art methods in cases where the expression is not manipulated, but rather the identity is changed. *This demonstrates the generalizability of our approach.*

4.3.2 Ablation study

To demonstrate the effectiveness of utilizing FER in manipulation detection and segmentation, we run different exper-

Table 2: Classification performance in terms of accuracy for state-of-art architectures using two types of face manipulation including Expression Swap (F2F and NT datasets) and Identity Swap (DF and DFDC datasets).

	Method	Expression Swap				Identity Swap		
		F2F(HQ)	NT(HQ)	F2F(LQ)	NT(LQ)	DF(HQ)	DF(LQ)	DFDC
W/O FER	Steg.Features+SVM [47]	74.68	76.94	60.58	60.69	77.12	65.58	-
	Cozzolino et al [43]	85.32	80.60	62.08	62.42	81.78	68.26	-
	Bayar and Stamm [8]	94.93	86.04	76.83	72.38	90.18	80.95	-
	Rahmouni et al [10]	93.48	75.18	67.08	62.59	82.16	73.25	-
	MesoNet [16]	95.84	85.95	83.56	75.74	95.26	89.52	-
	MultiTask [23]	92.77	88.05	82.31	80.67	93.92	85.77	69.76
	XceptionNet [15]	98.23	94.50	91.56	82.11	98.85	94.88	88.98
W/ FER	MultiTask+EnsFER	95.22	89.15	85.89	81.46	94.10	86.31	70.02
	EMD (ours)	99.03	96.31	94.45	83.67	99.13	95.28	89.16



(a) ROC curves for classification and segmentation with and without facial expression recognition on F2F. The solid and dotted blue lines are the proposed EMD algorithm.

(b) ROC curves for classification and segmentation with and without facial expression recognition on NT. The solid and dotted blue lines are the proposed EMD algorithm.

(c) The detection and segmentation performance of our approach by varying the training corpus size on F2F and DF datasets.

Figure 4: Analysis of performance under different conditions.

Table 3: Classification performance in terms of accuracy for state-of-art architectures on Face2Face datasets with two level of video quality.

	Method	HQ	LQ
W/O FER	LAE [48]	90.93	-
	DCNN [23]	93.50	82.13
	FT-res [49]	94.47	-
	Two-stream [50]	96.00	86.83
	Capsule-Forensics [51]	97.13	81.20
	Face X-ray [52]	97.73	-
W/ FER	MultiTask+EnsFER	95.22	85.89
	EMD (ours)	99.03	94.45

iments with variation of FER architecture. As we can see from Table 5, using FER system with multiple branches and selecting the most informative feature maps by using ESR (the one we use in our architecture) [3] achieves higher accuracy in both detection and segmentation tasks. Using simple FER (SimFER) [53] consisting of shallow convolutional layers (without Ensemble layers) leads to performance drop by $\sim 1\%$ and $\sim 2\%$ in classification and segmentation for both F2F and NT datasets with high quality videos.

4.4. Analysis of Results

Effect of FER on manipulation detection. We use ROC curves to show the benefit of FER in manipulation detection. Figs. 4a and 4b demonstrate ROCs for both detection and segmentation tasks with and without FER system. AUC score for our network with FER stream (EMD) achieves 99% and 97% for detection and segmentation tasks on F2F respectively. Thus, our method leads to $\sim 1\%$ and $\sim 2\%$ improvement in detection and segmentation AUC score in comparison to its counterpart without the FER stream. Based on Fig. 4b, our method achieves $\sim 3\%$ and $\sim 1\%$ improvement in detection and segmentation AUC score when it is trained/tested on NT dataset.

Size of training data. As shown in Fig. 4c, we evaluate our proposed network on training sets with variable sizes. For both datasets (F2F and NT), we compute classification and segmentation accuracy varying the training size from 10 to ~ 700 videos. By adding more videos to the datasets our performance increases initially. Adding more than 300, our model's performance does not change much indicating our method can perform good enough when there is not much data to train with, i.e., ~ 300 videos.

Table 4: Segmentation performance in terms of accuracy and mIoU for all evaluated architectures on Face2Face and Neural-Textures datasets with two level of video quality.

	Method	Acc(HQ)		Acc(LQ)		mIoU(HQ)		mIoU(LQ)	
		F2F	NT	F2F	NT	F2F	NT	F2F	NT
W/O FER	MultiTask [23]	90.27	88.67	87.76	84.55	81.02	73.33	74.21	70.68
	XceptionNet [15]	96.13	91.34	92.45	89.39	89.71	79.25	81.47	75.55
W/ FER	MultiTask+EnsFER	93.22	90.56	89.31	86.56	83.25	77.46	78.33	74.23
	EMD (ours)	98.43	93.78	95.22	91.54	92.13	83.21	86.71	79.44

Table 5: Classification (cls) and segmentation (seg) accuracy for different FER architectures on Face2Face and NeuralTextures datasets with high quality videos.

Method	Face2Face		NeutalTexture)	
	Cls	Seg	Cls	Seg
MultiTask+ SimFER	94.93	91.84	88.62	89.21
XceptionNet+ SimFER	98.63	96.89	95.83	92.44
MultiTask+EnsFER	95.22	93.22	89.15	90.56
EMD (ours)	99.03	98.43	96.31	93.78

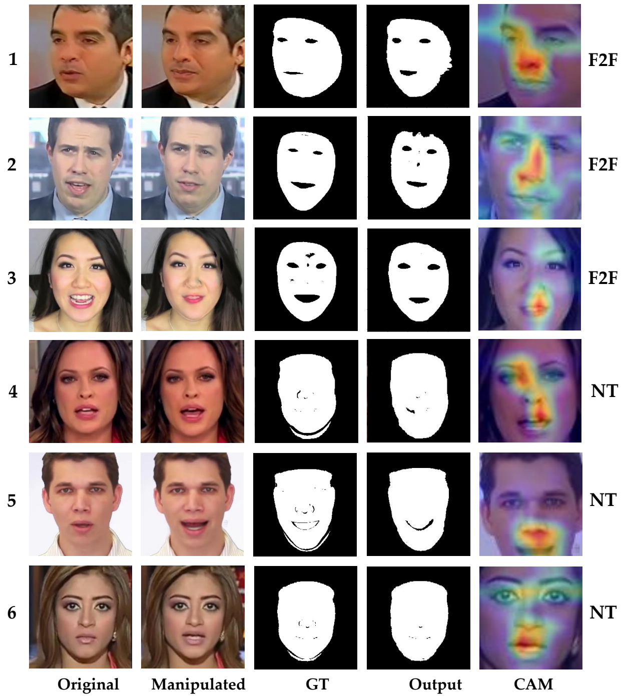


Figure 5: First and second columns show the original images and manipulated ones respectively. The black and white images in the third column are corresponding binary GT masks. Predicted masks (column 4) and generated CAMs (column 5) for manipulated images from Face2Face (row 1,2,3) and Neural-Textures (row 4,5,6) dataset

4.4.1 Why does FER help?

To show the effect of the features extracted from FER system, we visualize the last layer of CNN in our FER. For this purpose, we compute CAMs. A CAM for a particular category indicates the discriminative image regions used by the CNN to identify that category. Work by [54] has shown that the convolutional units of various layers of CNNs actually behave as object detectors despite no supervision on the location of the object provided.

In fact, the network can retain its remarkable localization ability until the final layer. This feature allows identification of the discriminative image regions which are important for manipulation detection. Specifically, for expression change detection, addition of a network which can localize regions in the face with information about the expressions helps manipulation detection methods to perform better.

As Fig. 5 shows, expression changes happen mostly around eyes, mouth and eyebrows. In the last column of figure, we generate the CAMs for manipulated and pristine images in F2F and NT datasets. As it is clear, our network can classify expressions quite well although the main FER stream has been trained on a different dataset (AffectNet).

5. Conclusions

In this paper, we propose a new approach (EMD) to exploit facial expression systems in image/video facial expression manipulation detection. Application of deep network layers rich in information about facial expressions improves the manipulation detector by making it learn the useful features for facial expression transformation. Experiments on two challenging datasets demonstrate our method has better classification and segmentation performance in facial expression manipulation detection in comparison to state-of-art results. Also, our method is close to the state-of-the-art methods for other kinds of manipulation (identity swap) detection, thus ensuring generalizability.

6. Acknowledgment

The work was partially supported by grants ORIIR180037-01-00 from the US Dept. of Health and Human Services and N00014-18-1-2252 from the US Office of Naval Research.

References

[1] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nieundefinedner. Face2face: Real-time face capture and reenactment of rgb videos. *Commun. ACM*, 62(1):96–104, December 2018.

[2] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Trans. Graph.*, 38(4), July 2019.

[3] Henrique Siqueira, Sven Magg, and Stefan Wermter. Efficient facial feature learning with wide ensemble-based convolutional neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5800–5809, Apr. 2020.

[4] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, June 2016.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[6] Deepfakes github. `https://github.com/deepfakes/faceswap`, 2019.

[7] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge dataset, 2020.

[8] Belhassen Bayar and Matthew C. Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, page 5–10, New York, NY, USA, 2016. Association for Computing Machinery.

[9] Giovanni Chierchia, Sara Parrilli, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. On the influence of denoising in prnu based forgery detection. In *Proceedings of the 2nd ACM Workshop on Multimedia in Forensics, Security and Intelligence*, MiFor '10, page 117–122, New York, NY, USA, 2010. Association for Computing Machinery.

[10] Nicolas Rahmouni, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Distinguishing computer graphics from natural images using convolution neural networks. *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2017.

[11] Jawadul H. Bappy, Cody Simons, Lakshmanan Nataraj, BS Manjunath, and Amit K. Roy-Chowdhury. Hybrid lstm and encoder–decoder architecture for detection of image forgeries. *IEEE Transactions on Image Processing*, 28(7):3286–3300, 2019.

[12] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *CVPR*, June 2019.

[13] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Learning rich features for image manipulation detection. In *CVPR*, June 2018.

[14] Ghazal Mazaheri, Kevin Urrutia Avila, and Amit K. Roy-Chowdhury. Learning to identify image manipulations in scientific publications, 2021.

[15] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. Faceforensics++: Learning to detect manipulated facial images. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[16] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7, Dec 2018.

[17] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K. Jain. On the detection of digital face manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[18] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[19] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan. Recurrent convolutional strategies for face manipulation detection in videos. *CoRR*, abs/1905.00582, 2019.

[20] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing AI generated fake face videos by detecting eye blinking. *CoRR*, abs/1806.02877, 2018.

[21] X. Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8261–8265, 2019.

[22] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[23] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos. In *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–8, 2019.

[24] Falko Matern, C. Riess, and M. Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 83–92, 2019.

[25] Yue Wu, Wael Abd-Almageed, and Prem 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*, MM '17, page 1480–1502, New York, NY, USA, 2017. Association for Computing Machinery.

[26] Jawadul H. Bappy, Amit K. Roy-Chowdhury, Jason Bunk, Lakshmanan Nataraj, and B. S. Manjunath. Exploiting spatial structure for localizing manipulated image regions. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[27] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Learning rich features for image manipulation detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[28] Seung-Jin Ryu, Matthias Kirchner, Min-Jeong Lee, and Heung-Kyu Lee. Rotation invariant localization of duplicated image regions based on zernike moments. *IEEE Transactions on Information Forensics and Security*, 8:1355–1370, 2013.

[29] Pasquale Ferrara, Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Image forgery localization via fine-grained analysis of cfa artifacts. *IEEE Transactions on Information Forensics and Security*, 7:1566–1577, 2012.

[30] Tiziano Bianchi and Alessandro Piva. Image forgery localization via block-grained analysis of jpeg artifacts. *IEEE Transactions on Information Forensics and Security*, 7:1003–1017, 2012.

[31] Bo Liu and Chi-Man Pun. Deep fusion network for splicing forgery localization. In *The European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[32] J. Bunk, J. H. Bappy, T. M. Mohammed, L. Nataraj, A. Flenner, B. S. Manjunath, S. Chandrasekaran, A. K. Roy-Chowdhury, and L. Peterson. Detection and localization of image forgeries using resampling features and deep learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1881–1889, July 2017.

[33] Zhongping Zhang, Yixuan Zhang, Zheng Zhou, and Jiebo Luo. Boundary-based image forgery detection by fast shallow cnn. *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2658–2663, 2018.

[34] Ghazal Mazaheri, Niluthpol Chowdhury Mithun, Jawadul H. Bappy, and Amit K. Roy-Chowdhury. A skip connection architecture for localization of image manipulations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[35] Behzad Hassani and Mohammad H. Mahoor. Spatio-temporal facial expression recognition using convolutional neural networks and conditional random fields. *2017 12th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2017)*, pages 790–795, 2017.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[37] Shan Li, Weihong Deng, and JunPing Du. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[38] Huiyuan Yang, Umur Ciftci, and Lijun Yin. Facial expression recognition by de-expression residue learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[39] Jiabei Zeng, Shiguang Shan, and Xilin Chen. Facial expression recognition with inconsistently annotated datasets. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[40] Amir Hossein Farzaneh and Xiaojun Qi. Facial expression recognition in the wild via deep attentive center loss. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2401–2410, 2021.

[41] Wenyun Sun, Haitao Zhao, and Zhong Jin. A visual attention based roi detection method for facial expression recognition. *Neurocomputing*, 296:12–22, 2018.

[42] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics: A large-scale video dataset for forgery detection in human faces. *CoRR*, abs/1803.09179, 2018.

[43] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection. In *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, page 159–164, New York, NY, USA, 2017. Association for Computing Machinery.

[44] Ali Mollahosseini, Behzad Hasani, and Mohammad H. Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1):18–31, Jan 2019.

[45] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015.

[46] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[47] J. Fridrich and J. Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, June 2012.

[48] Mengnan Du, Shiva Pentyala, Yuening Li, and Xia Hu. *Towards Generalizable Deepfake Detection with Locality-Aware AutoEncoder*, page 325–334. Association for Computing Machinery, New York, NY, USA, 2020.

[49] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. *CoRR*, abs/1812.02510, 2018.

[50] 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)*, pages 1831–1839, 2017.

[51] Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. Capsule-forensics: Using capsule networks to detect forged

images and videos. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2307–2311. IEEE, 2019.

[52] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo. Face x-ray for more general face forgery detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5000–5009, 2020.

[53] A. Mollahosseini, B. Hasani, and M. H. Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(01):18–31, 2019.

[54] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *CoRR*, abs/1412.6856, 2014.