Exposing DeepFakes Using Neural Networks

Sanika Utturwar[1], Harsh Warbhe[2], Janvi Gurfude[3], Vivek Ugale[4], Santoshi Lokhande[5]
[1,2,3,4,5]*Government College of Engineering, Chandrapur*

***Abstract*—Advancements in Artificial Intelligence has been accelerated by advances in computing power and social media's ever-expanding reach and more fake face image generators have emerged worldwide owing to the growth of Face Image Modification (FIM) tools like Face2Face and Deepfake, which pose a severe threat to public trust. High levels of realism can be achieved in these synthesized videos by utilizing generative machine learning models such as Variational AutoEncoders or Generative Adversarial Networks.**

Although there have been significant advancements in the identification of certain FIM, a reliable false face detector is still lacking. Convolutional Neural Network (CNN) tends to learn picture content representations because of the structure's relative stability.

The widespread adoption of deepfake technology presents a pressing concern across diverse sectors, encompassing politics, security, and personal privacy. This paper introduces an innovative temporal-aware approach for automatically detecting deepfake videos. Our method employs a dual-stage neural network architecture, comprising a Convolutional Neural Network (CNN) for extracting features at the frame level, followed by a Recurrent Neural Network (RNN) for temporal analysis. By leveraging the inherent temporal dynamics characteristic of deepfake generation, the RNN discerns subtle manipulations to classify videos accurately. We assess the efficacy of our methodology using a comprehensive dataset comprising deepfake videos sourced from various online platforms. Our findings underscore the robustness and competitive performance of our system, underscoring its effectiveness despite its straightforward architecture.

INTRODUCTION

The first known attempt at trying to swap someone's face, circa 1865, can be found in one of the iconic portraits of U.S. President Abraham Lincoln. The lithography, as seen in Figure 1, mixes Lincoln's head with the body of Southern politician John Calhoun. After Lincoln's as assassination, demand for lithographies of him was so great that engravings of his head on other bodies appeared almost overnight [5].

Recent advances [6, 7] have radically changed the playing field of image and video manipulation. The democratization of modern tools such as TensorFlow [8] or Keras [9] coupled with the open accessibility of the recent technical literature and cheap access to compute infrastructure have propelled this paradigm shift. Convolutional autoencoders [10, 11] and generative adversarial network (GAN) [12, 13] models have made tampering images and videos, which used to be reserved to highly-trained professionals, a broadly accessible operation within reach of almost any individual with a computer. Smartphone and desktop applications like FaceApp [14] and FakeApp [15] are built upon this progress.

FaceApp automatically generates highly realistic transformations of faces in photographs. It allows one to change face hair style, gender, age and other attributes using a smartphone. FakeApp is a desktop application that allows one to create what are now known as "deepfakes" videos. Deepfake videos are manipulated videoclips which were first created by a Reddit user, deepfake, who used TensorFlow, image search engines, social media websites and public video footage to insert someone else's face onto pre-existing videos frame by frame.

Although some benign deepfake videos exist, they remain a minority. So far, the released tools [15] that generate deepfake videos have been broadly used to create fake celebrity pornographic videos or revenge porn [17]. This kind of pornography has already been banned by sites including Reddit, Twitter, and Pornhub. The realistic nature of deepfake videos also makes them a target for generation of pedopornographic material, fake news, fake surveillance videos, and malicious hoaxes. These fake videos have already been used to create political tensions and they are being taken into account by governmental entities [16].

As presented in the Malicious AI report [18], researchers in artificial intelligence should always reflect on the dual use nature of their work, allowing misuse considerations to influence research priorities and norms. Given the severity of the malicious attack vectors that deepfakes have caused, in this paper we present a novel solution for the detection of this kind of video.

The main contributions of this work are summarized as follows. First, we propose a two-stage analysis composed of a CNN to extract features at the frame level followed by a temporally-aware RNN network to capture temporal inconsistencies between frames introduced by the face-swapping process. Second, we have used a collection of 600 videos to evaluate the proposed method, with half of the videos being deepfakes collected from multiple video hosting websites. Third, we show experimentally the effectiveness of the described approach, which allows use to detect if a suspect video is a deepfake manipulation with 94% more accuracy than a random detector baseline in a balanced setting.

LITERATURE SURVEY

The explosive growth in deep fake video and its illegal use is a major threat to democracy, justice, and public trust. Due to this there is an increased the demand for fake video analysis, detection and intervention. Some of the relatedword in deep fake detection are listed below:
ExposingDF Videos by Detecting Face Warping Artifacts [1] used an approach to detects artifacts by comparing the generated face areas and their surrounding regions with a dedicated Convolutional Neural Network model. In this work there were two-fold of Face Artifacts.
Their method is based on the observations that current DF algorithm can only generate images of limited resolutions, which are then needed to be further transformed to match the faces to be replaced in the source video.

Exposing AI Created Fake Videos by Detecting Eye Blinking [2] describes a new method to expose fake face videos generated with deep neural network models. The method is based on detection of eye blinking in the videos, which is a physiological signal that is not well presented in the synthesized fake videos. The method is evaluated over benchmarks of eye-blinking detection datasets and shows promising performance on detecting videos generated with Deep Neural Network based software DF.
Their method only uses the lack of blinking as a clue for detection. However certain other parameters must be considered for detection of the deep fake like teeth enchantment, wrinkles on faces etc. Our method is proposedto consider all these parameters.

Using capsule networks to detect forged images and videos [3] uses a method that uses a capsule network to detect forged, manipulated images and videos in different scenarios, like replay attack detection and computer- generated video detection.

In their method, they have used random noise in the training phase which is not a good option. Still the model performed beneficial in their dataset but may fail on real time data due to noise in training. Our method is proposed to be trained on noiseless and real time datasets.

Detection of Synthetic Portrait Videos using Biological Signals [4] approach extract biological signals from facial regions on authentic and fake portrait video pairs. Apply transformations to compute the spatial coherence and temporal consistency, capture the signal characteristics in feature sets and PPG maps, and train a probabilistic SVM and a CNN. Then, the aggregate authenticity probabilities to decide whether the video is fakeor authentic.

Fake Catcher detects fake content with highaccuracy, independent of the generator, content, resolution, and quality of the video. Due to lack of discriminator leading to the loss in their findings to preserve biological signals, formulating a differentiable loss function that follows the proposed signal processing steps is not straight forward process.

Deepfake Videos Exposed: Due to the way that FakeApp [15] generates the manipulated deepfake video, intra-frame inconsistencies and temporal inconsistencies between frames are created. These video anomalies can be exploited to detect if a video under analysis is a deepfake manipulation or not. Let us briefly explain how a deepfake video is generated to understand why these anomalies are introduced in the videos and how we can exploit them.

CREATING DEEPFAKE VIDEOS

It is well known that deep learning techniques have been successfully used to enhance the performance of image compression. Especially, the autoencoder has been applied for dimensionality reduction, compact representations of images, and generative models learning [19]. Thus, autoencoders are able to extract more

compressed representations of images with a minimized loss function and are expected to achieve better compression performance than existing image compression standards. The compressed representations or latent vectors that current convolutional autoencoders learn are the first cornerstone behind the face swapping capabilities of [15]. The second insight is the use of two sets of encoder-decoders with shared weights for the encoder networks. Figure 2 shows how these ideas are used in the training and generation phases that happen during the creation of a deepfake video.

1. Training

Two sets of training images are required. The first set only has samples of the original face that will be replaced, which can be extracted from the target video that will be manipulated. This first set of images can be further extended with images from other sources for more realistic results. The second set of images contains the desired face that will be swapped in the target video. To ease the training process of the autoencoders, the easiest face swap would have both the original face and target face under similar viewing and illumination conditions. However, this is usually not the case. Multiple camera views, differences in lightning conditions or simply the use of different video codecs makes it difficult for autoencoders to produce realistic faces under all conditions. This usually leads to swapped faces that are visually inconsistent with the rest of the scene. This frame level scene inconsistency will be the first feature that we will exploit with our approach.

It is also important to note that if we train two autoencoders separately, they will be incompatible with each other. If two autoencoders are trained separately on different sets of faces, their latent spaces and representations will be different. This means that each decoder is only able to decode a single kind of latent representations which it has learnt during the training phase. This can be overcome by forcing the two set of autoencoders to share the weights for the encoder networks, yet using two different decoders. In this fashion, during the training phase these two networks are treated separately and each decoder is only trained with faces from one of the subjects. However, all latent faces are produced by the same encoder which forces the encoder itself to identify common features in both faces. This can be easily accomplished due to the natural set of shared traits of all human faces (e.g., number and position of eyes, nose).

2. Video Generation

When the training process is complete, we can pass a latent representation of a face generated from the original subject present in the video to the decoder network trained on faces of the subject we want to insert in the video. As shown in Figure 2, the decoder will try to reconstruct a face from the new subject, from the information relative to the original subject face present in the video. This process is repeated for every frame in the video where we want to do a face swapping operation. It is important to point out that for doing this frame-level operation, first a face detector is used to extract only the face region that will be passed to the trained autoencoder. This is usually a second source of scene inconsistency between the swapped face and the reset of the scene. Because the encoder is not aware of the skin or other scene information it is very common to have boundary effects due to a seamed fusion between the new face and the rest of the frame.

The third major weakness that we exploit is inherent to the generation process of the final video itself. Because the autoencoder is used frame-by-frame, it is completely unaware of any previous generated face that it may have created. This lack of temporal awareness is the source of multiple anomalies. The most prominent is an inconsistent choice of illuminants between scenes with frames, with leads to a flickering phenomenon in the face region common to the majority of fake videos. Although this phenomenon can be hard to appreciate to the naked eye in the best manually-tuned deepfake manipulations, it is easily captured by a pixel-level CNN feature extractor. The phenomenon of incorrect colour constancy in CNN-generated videos is a well-known and still open research problem in the computer vision field [20]. Hence, it is not surprising that an autoencoder trained with very constrained data fails to render illuminants correctly.

3. Recurrent Network for Deepfake Detection

In this section, we present our end-to-end trainable recurrent deepfake video detection system (Figure 3). The proposed system is composed by a convolutional LSTM structure for processing frame sequences. There are two essential components in a convolutional LSTM:

1. CNN for frame feature extraction.
2. LSTM for temporal sequence analysis.

Given an unseen test sequence, we obtain a set of features for each frame that are generated by the CNN. Afterwards,

we concatenate the features of multiple consecutive frames and pass them to the LSTM for analysis. We finally produce an estimate of the likelihood of the sequence being either a deepfake or a non-manipulated video.

3.1. Convolutional LSTM

Given an image sequence (see Figure 3), a convolutional LSTM is employed to produce a temporal sequence descriptor for image manipulation of the shot frame. Aiming at end-to-end learning, an integration of fully-connected layers is used to map the high-dimensional LSTM descriptor to a final detection probability. Specifically, our shallow network consists of two fully-connected layers and one dropout layer to minimize training over-fitting. The convolutional LSTM can be divided into a CNN and a LSTM, which we will describe separately in the following paragraphs.

3.2. CNN for Feature Extraction.

Inspired by its success in the IEEE Signal Processing Society Camera Model Identification Challenge, we adopt the InceptionV3 [21] with the fully-connected layer at the top of the network removed to directly output a deep representation of each frame using the ImageNet pre-trained model. Following [22], we do not fine-tune the network. The 2048-dimensional feature vectors after the last pooling layers are then used as the sequential LSTM input.

3.3. LSTM for Sequence Processing.

Let us assume a sequence of CNN feature vectors of input frames as input and a 2-node neural network with the probabilities of the sequence being part of a deepfake video or an untampered video. The key challenge that we need to address is the design of a model to recursively process a sequence in a meaningful manner. For this problem, we resort to the use of a 2048-wide LSTM unit with 0.5 chance of dropout, which is capable to do exactly what we need. More particularly, during training, our LSTM model takes a sequence of 2048- dimensional ImageNet feature vectors. The LSTM is followed by a 512 fully-connected layer with 0.5 chance of dropout. Finally, we use a SoftMax layer to compute the probabilities of the frame sequence being either pristine or deepfake. Note that the LSTM module is an intermediate unit in our pipeline, which is trained entirely end-to-end without the need of auxiliary loss functions.

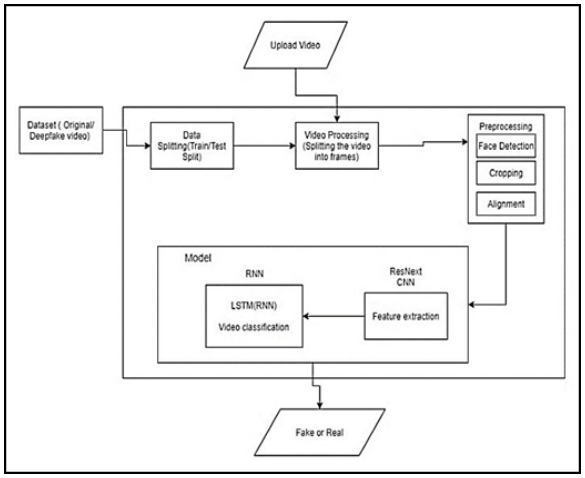


Fig. System Architecture

4. Proposed Methodology:

There are many tools available for creating the DF, but forDF detection there is hardly any tool available. Our approach for detecting the DF will be great contribution inavoiding the percolation of the DF over the world wide web.We will be providing a web-based platform for the user toupload the video and classify it as fake or real. This projectcan be scaled up from developing a web-based platform to a browser plugin for automatic DF detections. Even big application like WhatsApp, Facebook can integrate thisproject with their application for easy pre detection of DFbefore sending to another user. One of the importantobjectives is to evaluate its performance and acceptability interms of security, user-friendliness, accuracy and reliability. Our method is focusing on detecting all types of DF likereplacement DF, retrenchment DF and interpersonal DF.

A] Dataset:

In this study, we utilize the Deepfake Detection Challenge (DFDC) dataset, a widely recognized benchmark in the field of deepfake detection. The DFDC dataset comprises a diverse collection of videos sourced from various online platforms, encompassing a wide range of individuals, scenarios, and quality levels.

This dataset includes both real and manipulated videos, providing a comprehensive and representative sample for training and evaluation purposes. With its large-scale and diverse nature, the DFDC dataset enables robust model training and thorough performance

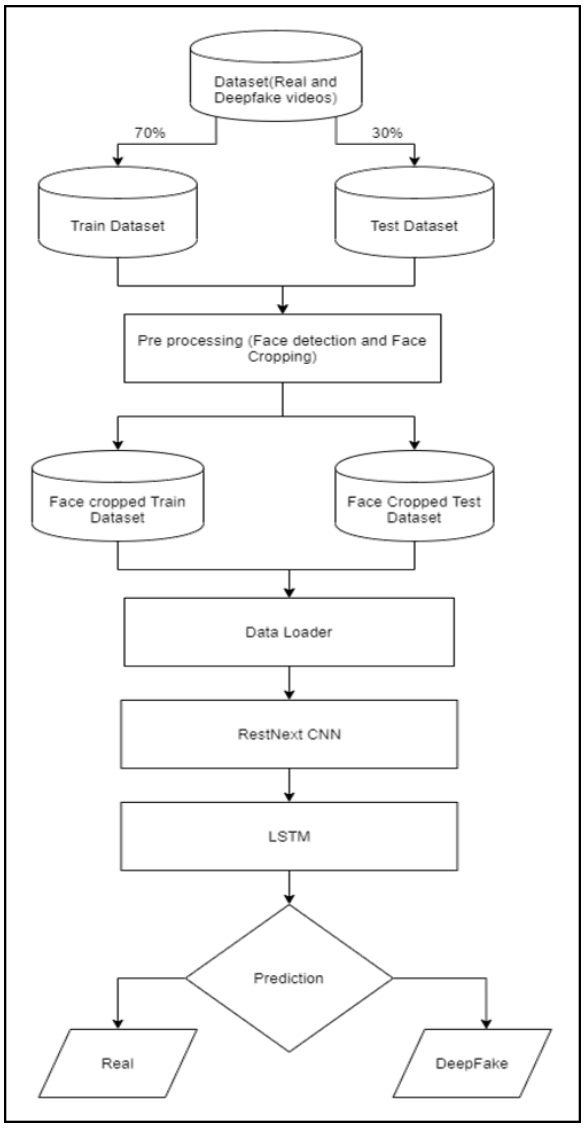


Fig. Training Flow

assessment, aligning well with the objectives of our research. Leveraging this dataset, we train our deepfake detection model using a recurrent neural network (RNN) architecture, allowing us to effectively capture temporal dynamics and achieve competitive performance in identifying manipulated videos.

B] Pre-processing:

Dataset pre-processing encompasses several essential steps to prepare the DeepFake Detection Challenge (DFDC) dataset for analysis. Initially, each video is split into individual frames, followed by the application of face detection techniques to isolate facial regions within each frame. The detected faces are then cropped to focus solely on facial features, ensuring that extraneous information is minimized. To maintain consistency in the number of frames across the dataset, we calculate the mean frame count of the videos and create a new processed dataset containing frames equal to this mean value. Frames without detectable faces are excluded during pre-processing to enhance dataset quality.

However, considering the computational demands associated with processing the entire duration of each video, we propose a pragmatic approach for experimental purposes. Given that processing a 10-second video at 30 frames per second results in a total of 300 frames, which may require substantial computational power, we suggest utilizing only the first 100 frames for training the model. This decision balances computational efficiency with the retention of crucial temporal information encapsulated within the initial segment of the videos, which typically contains significant facial movements and expressions. By adopting this approach, we aim to streamline model training while still leveraging meaningful temporal cues for effective deepfake detection.

C] Model:

The proposed model architecture comprises a ResNext50_32x4d convolutional neural network (CNN) followed by a single LSTM layer. The Data Loader module is responsible for loading the pre-processed face-cropped videos and dividing them into train and test sets. Subsequently, the frames from the processed videos are fed into the model for both training and testing, organized into mini-batches.

D] ResNext CNN for Feature Extraction:

Rather than designing a new classifier, we advocate utilizing the ResNext CNN classifier for feature extraction, thereby accurately capturing frame-level features. Post feature extraction, the network undergoes fine-tuning, incorporating additional layers as necessary, and optimizing the learning rate to ensure proper convergence of the gradient descent process. The resulting 2048-dimensional feature vectors, extracted after the final pooling layers, serve as the input to the sequential LSTM layer.

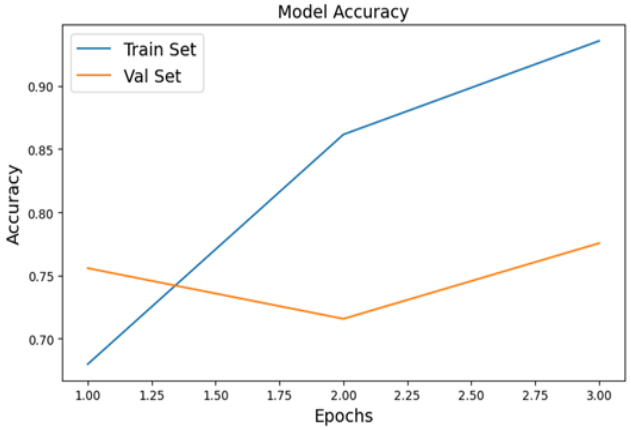
E] LSTM for Sequence Processing:

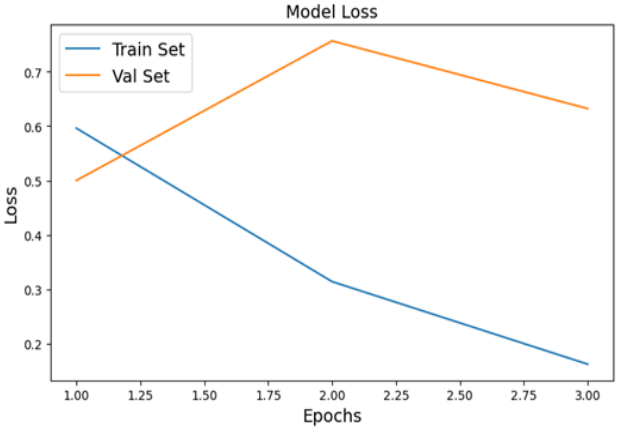
The sequential input to the LSTM layer comprises a sequence of ResNext CNN feature vectors, with the objective of distinguishing between deepfake and untampered videos. Addressing the challenge of effectively processing sequential data, we propose employing a 2048 LSTM unit with a dropout probability

of 0.4. This LSTM architecture facilitates meaningful temporal analysis by sequentially processing frames, allowing for comparisons between frames at time 't' and those at preceding time instances 't-n', where 'n' denotes the number of frames preceding time 't'.

F] Prediction:

During the prediction phase, a new video is inputted into the trained model for inference. The video undergoes pre-processing to align with the input format of the trained model, involving frame splitting, face cropping, and direct passage of cropped frames to the model for detection. This streamlined prediction process eliminates the need for storing the entire video locally, ensuring efficient real-time application of the detection model.





RESULTS

It is not unusual to find deepfake videos where the manipulation is only present in a small portion of the video (i.e., the target face only appears briefly on the video, hence the deepfake manipulation is short in time). To account for this, for every video in the training, validation and test splits, we extract continuous subsequence of fixed frame length that serve as the input of our system.

In Table 1 we present the performance of our system in terms of detection accuracy using sub-sequences of length N = 20, 40, 80 frames. These frame sequences are extracted sequentially (without frame skips) from each video. The entire pipeline is trained end-to-end until we reach a 10-epoch loss plateau in the validation set.

As we can observe in our results, with less than 2 seconds of video (40 frames for videos sampled at 24 frames per second) our system can accurately predict if the fragment being analysed comes from a deepfake video or not with an accuracy greater than 97%.

Model	Training Acc. (%)	Validation Acc. (%)	Test Acc. (%)
Conv-LSTM, 20 frames	99.5	96.9	96.7
Conv-LSTM, 40 frames	99.3	97.1	97.1
Conv-LSTM, 80 frames	99.7	97.2	97.1

Table1. Classification results of our dataset splits using video sub sequences with different lengths.

CONCLUSION

In this paper we have presented a temporal-aware system to automatically detect deepfake videos. Our experimental results using a large collection of manipulated videos have shown that using a simple convolutional LSTM structure we can accurately predict if a video has been subject to manipulation or not with as few as 2 seconds of video data. We believe that our work offers a powerful first line of defence to spot fake media created using the tools described in the paper. We show how our system can achieve competitive results in this task while using a simple pipeline architecture. In future work, we plan to explore how to increase the robustness of our system against manipulated videos using unseen techniques during training.

REFERENCE

[1] Yuezun Li, Siwei Lyu, “ExposingDF Videos By Detecting Face Warping Artifacts,” in arXiv:1811.00656v3.

[2] Yuezun Li, Ming-Ching Chang and Siwei Lyu “Exposing AI Created Fake Videos by Detecting Eye Blinking” in arxiv.

[3] Huy H. Nguyen , Junichi Yamagishi, and Isao Echizen“ Using capsule networks to detect forged images and videos ”.
[4] Umur Aybars Ciftci, ˙Ilke Demir, Lijun Yin “Detection of Synthetic Portrait Videos using Biological Signals” in arXiv:1901.02212v2.
[5] S. Lorant. Lincoln; a picture story of his life. Norton, 1969.
[6] Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. Proceedings of the IEEE International Conference on Computer Vision, pages 2242–2251, Oct. 2017. Venice, Italy.
[7] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5967–5976, July 2017. Honolulu, HI.
[8] Abadi et al. Tensorflow: A system for large-scale machine learning. Proceedings of the USENIX Conference on Operating Systems Design and Implementation, 16:265–283, Nov. 2016. Savannah, GA.
[9] F. Chollet et al. Keras. https://keras.io, 2015.
[10] A. Tewari et al. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 1274–1283, Oct. 2017. Venice, Italy.
[11] J. Thies et al. Face2Face: Real-time face capture and reenactment of rgb videos. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2387– 2395, June 2016. Las Vegas, NV
[12] I. Goodfellow et al. Generative adversarial nets. Advances in Neural Information Processing Systems, pages 2672–2680, Dec. 2014. Montreal, Canada
[13] G. Antipov, M. Baccouche, and J.-L. Dugelay. Face aging with conditional generative adversarial networks. arXiv:1702.01983, Feb. 2017.
[14] Faceapp. https://www.faceapp.com/. (Accessed on 05/29/2018).
[15] Fakeapp. https://www.fakeapp.org/. (Accessed on 05/29/2018).
[16] The Outline: Experts fear face swapping tech could start an international showdown. https://theoutline.com/post/3179/ deepfake-videos-are-freaking-experts-out?zd=1&zi=hbmf4svs. (Accessed on 05/29/2018).
[17] What are deepfakes & why the future of porn is terrifying. https://www.highsnobiety.com/p/ what-are-deepfakes-ai-porn/. (Accessed on 05/29/2018).
[18] M. Brundage et al. The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. arXiv:1802.07228, Feb. 2018
[19] Y. Liao, Y. Wang, and Y. Liu. Graph regularized autoencoders for image representation. IEEE Transactions on Image Processing, 26(6):2839–2852, June 2017
[20] Y. Qian et al. Recurrent color constancy. Proceedings of the IEEE International Conference on Computer Vision, pages 5459–5467, Oct. 2017. Venice, Italy
[21] C. Szegedy et al. Rethinking the inception architecture for computer vision. Proeedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2818– 2826, June 2016. Las Vegas, NV.
[22] EE’s Signal Processing Society - Camera Model Identification—Kaggle. https://www.kaggle.com/c/sp-society-camera-model-identification/ discussion/49299. (Accessed on 05/29/2018).