Detecting Deepfakes with Metric Learning

Akash Kumar
MANAS Lab
IIT Mandi
India
akash_bt2k15@dtu.ac.in

Arnav Bhavsar
MANAS Lab
IIT Mandi
India
arnav@iitmandi.ac.in

Rajesh Verma
RFSL
Mandi
India
rkverma.fslmnd@hp.gov.in

Abstract—With the arrival of several face-swapping applications such as *FaceApp, SnapChat, MixBooth, FaceBlender* and many more, the authenticity of digital media content is hanging on a very loose thread. On social media platforms, videos are widely circulated often at a high compression factor. In this work, we analyze several deep learning approaches in the context of deepfakes classification in high compression scenarios and demonstrate that a proposed approach based on metric learning can be very effective in performing such a classification. Using less number of frames per video to assess its realism, the metric learning approach using a triplet network architecture proves to be fruitful. It learns to enhance the feature space distance between the cluster of real and fake videos embedding vectors. We validated our approaches on two datasets to analyze the behavior in different environments. We achieved a state-of-the-art AUC score of 99.2% on the Celeb-DF dataset and accuracy of 90.71% on a highly compressed Neural Texture dataset. Our approach is especially helpful on social media platforms where data compression is inevitable.

Index Terms—Video Forensics, Triplet Network, Image Classification, Deepfakes

I. Introduction

With the rapid increase of online streaming platforms, there is a dire need to check the authenticity of the videos. In Youtube alone, 300 hours of videos are uploaded every minute. On a daily basis, 5 billion videos are watched and 1 billion hours are streamed, that's Facebook and Netflix streaming combined. The rise of deepfakes in recent years seriously raises concerns about the authenticity of digital content by media and other online streaming platforms. Generative architectures are excellent for aiding in boosting the performance of deep learning architectures by satisfying the need for large datasets, and in general to explore the creative power of deep learning. However, such approaches have also resulted in Deepfakes, which are now been utilized for nefarious purposes to manipulate the images of politicians, famous actors, etc. Many politicians and actors are becoming victims of Deepfakes. For criminal purposes, forensic videos are altered using novel methods such as *faceswap* and *faceswap-GAN*.

Various applications are using human faces to transform it into sophisticated fun images like modifying the age, changing the gender, etc. In exchange, the users are giving away their face data to these companies, which can further be used for wrongful purposes.

When the manipulated videos are shared on social apps, their quality is reduced to make it convenient for uploading
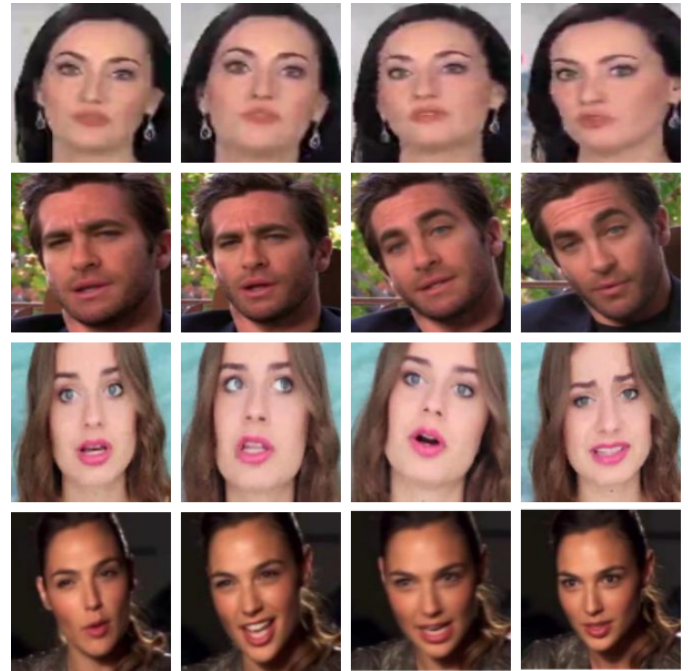

Fig. 1. The first two rows depicts the deepfake frames of FF++ and CelebDF dataset. Next two rows are examples of original sequences of the respective datasets.

and downloading through those apps. In high-quality videos, a small amount of fuzziness around the face warping can be visualized. However, in low-quality videos, users are unable to detect distinguish whether the videos are authentic or fake and the videos are forwarded to large groups of people. Such manipulations can have large scale effects, from politics to the entertainment industry. For instance, videos of politicians participating in certain events or announcing a public service, that never really existed sabotage the image of the politician. Similarly, in fake porn videos of actors commonly circulated online.

To counter (detect) such as video manipulation, several algorithms using handcrafted features, deep learning algorithms, and lately GAN-based methods are being explored. For instance, handcrafted approaches involve methods for steganalysis, detecting 3D head pose inconsistencies, etc. Several such existing approaches are summarized in [1] and [2]. However, there is still scope of improvement over the state-of-the-art

for detecting deepfakes, especially on challenging data such as the Face Forensics (FF++) dataset [3].

Contributions In this paper, our main contributions are as follows: 1) Improve the binary classification of Deepfakes on new second-generation Celeb-DF [4] dataset and the FF++ dataset [3], using a metric learning approach. 2) Experimentally demonstrating the performance of various contemporary methods and their variations to classify videos in high compression factors on the FF++ dataset. 3) Boosting the accuracy in low-resolution videos with minimal dependency on the number of frames required per video, as compared to recent existing methods.

The paper is arranged a follows: Section II discusses the existing approaches in deepfakes video generation, datasets available and classification approaches. Section III goes through the approaches we used to improvise the video classification in high compression factors. After that, we discuss our experimental results in Section IV. Further, future directions our work is discussed in Section V.

II. Related Work

A. Deepfakes Generation methods

With the rise of engagement on social media platforms, many applications are now based on face-swapping technologies such as FaceApp, Snapchat, etc. [5]–[8] Many novel approaches have been introduced in recent years. Thies *et al.* in their work Face2Face [5] produces a real-time *facial reenactment* video. Initially, the training corpus contains the identity of the target actors. Then, they track the expressions of both the source and target candidate and, smoothly apply a deformation to transfer the expression. After that, the face is reconstructed. Compared to other depth-based methods, they have used tracking across frames and appearance information in RGB videos. Deepfakes *faceswap* uses an encoder-decoder based approach to alter the face of the target candidate. Two same encoders with shared parameters for both the candidates are used to encode the facial features, after that, the decoder for the target candidate is used to transfer the facial expressions of the source candidate. *faceswap-GAN* incorporates adversarial and perceptual loss to further improve facial reconstruction. In Neural Texture synthesis [7], 3D reconstruction of images is done in imperfect geometry conditions and produced at real-time rates. High-level encoding of surface appearance and the 3D environment is captured. It helps the network to easily manipulate the re-renderings of the environment of source candidates on-target candidates. Nirkin *et al.* [6] proposed a method that does not even need the target actor to be present inside the training corpus. Moreover, they used two new loss functions stepwise consistency loss and blending loss. Stepwise consistency loss regulates the transfer of source candidate's appearance to target candidate and reconstruction loss keeps in check of the realism of facial reconstruction. Poisson blending helps to blend in the two faces seamlessly keeping in view the background environment.

B. Deepfake Video Datasets

Many new large and realistic Deepfake Video datasets start coming up from the year 2018, as the use of GANs in several research directions started blooming. [1] The Deepfake-TIMIT was the first dataset that synthesized videos using faceswap-GAN to generate 640 videos. Videos were divided into low and high quality depending upon the resolution of images that were 64 and 128 respectively. Fake Face in the Wild dataset, involves splicing and deepfake approaches to generate only 150 videos of frame sizes ranging from 480p to 1080p. A large and diverse dataset consisting of numerous manipulations for automatically generating faces, FF++ was released last year. It contains 1000 real videos carefully extracted from the Youtube-8M dataset. Then, four types of approaches were employed to re-render the expression and facial attributes from source to target candidate. Two of them were computer graphics-based and the other two utilizes a deep learning approach. It contains dataset in three types of compression factors raw, medium and high to make the models more robust towards detection. Celeb-DF dataset released later in the year 2019, contains 560 real videos and 5639 deepfake videos. Recently, Facebook has also hosted an online challenge on Kaggle, DeepFake Detection Challenge (DFDC), releasing 10,000 fake videos and 19,000 pristine videos.

C. Deepfakes Video Classification

With evolving computational capacity and threats of deepfakes, several methods evolved to classify fake videos. Learning the mismatch between visual artifacts, head poses variation, using segmentation masks and training shallow and deep networks are some of the contemporary approaches that have made use of to detect manipulations. Two-stream CNN recognizes the tampered faces by training a face classification network and a patch triplet network. The classification network uses LeNet architecture to train the model. Patch Triplet network helps the model to force the embeddings of the same type of images closer. In MesoNet [9], shallow architectures with the inclusion of the inception module learn the discriminative features from frames. On the other hand, recent works [1] have proved that deep architectures outperform the shallow networks by a large margin. HeadPose network identifies the tampering by measuring the distance between the synthesized image head pose and the original image head pose. On estimating the 3D head pose from 2D coordinates system, the landmarks of manipulated faces are shifted from the original faces. Li and Lyu, in their work, detect peculiar artifacts that are introduced by warping steps. Convolutional Neural Networks (CNNs) are trained to detect the lack of consistency between the final and initial image after applying various transformations. Matern *et al.* worked on visual artifacts to detect manipulated images. They showed that using facial attributes, there's a discrepancy between an original video and a manipulated one that is easily noticeable. However, they evaluated their approach mainly on DeepFakes and Face2Face type of manipulations. In the Multi-task learning approach, classification, segmentation, and reconstruction

are performed altogether to boost classification accuracy. An encoder-decoder approach learns to reconstruct the image and then final activation is used for classification. Capsule networks are designed to use less number of parameters to overcome the need for training millions of parameters in deep neural networks. Capsule Forensics uses a dynamic routing algorithm to generate an activation map where the face has been manipulated. As we can see in [4], these approaches don't generalize well on new and more challenging datasets. From [3], we can see many approaches [9]–[14] perform well when the resolution of the video is high. As the video is compressed by a medium or high factor, the performance of these models drops significantly. From 95+% accuracy, it drops 50-60%, and in the case of NeuralTextures, accuracy goes down to 50%, which means the network is unable to learn any features at such low resolutions and just randomly annotating the videos as fake or real.

III. METHODOLOGY

We use a Multitask Cascaded CNNs (MTCNN) [15] to extract faces out of frames. Based on the success of detection of fake and real videos of XceptionNet architecture from FF++ paper [3], we started off with it for video classification. To combat the classification in low-resolution videos, we further analyzed several methods using recurrent neural networks, convolutional 3D networks, and, then finally metric learning approach. Our architecture and methods involved are discussed in the following subsections.

A. MTCNN

In this approach, one can crop out images using Proposal, Refine and Output Net. The proposal network detects faces across multiple resolutions, following which, the refine net suppress the overlapping boxes using nonmax suppression. Finally, the output network gives the bounded face using five landmarks.

Fig. 2. Extraction of face from frame using MTCNN algorithm.

B. Transfer Learning

To make use of previous knowledge of architecture from one problem onto another problem is known as transfer learning. In our work, we used Xception [12] architecture to learn the crucial feature of real and fake faces. Xception net based on Inception V3 uses the Inception module, with modification of the spatial convolutions to depthwise separable convolutions. After separating each channel, 1x1 depthwise convolutions help network to capture the cross-channel correlations. Compared to Inception architecture convolutions, depthwise separable convolution differs in two ways: 1) Xception modules perform channel-wise convolutions first, then, 1x1 convolution, compared to Inception where the 1x1 is performed earlier, and, 2) There's no non-linearity after depthwise separable convolutions. With this, the number of layers is reduced from 159 layers in Inception V3 to 126 layers in Xception architecture. Although Xception Net was quite successful in capturing dissimilarities in high-resolution images, the performance drops significantly for low resolution compressed images.

C. Sequence Classification

Recurrent Neural networks capture the information along the temporal domain. An output vector from the previous step is fed into the next step to learn the relation of features across time domain. Their ability to connect a sequence of input frames over a period of time makes them significantly helpful for video classification purposes. LSTM supersedes the RNN as it can retain information for a long sequence of frames. In our work, we used a sequence of 16 and 32 frames per video to learn the inconsistency across the temporal domain. Our architecture contains a single layer LSTM of 32 units and then a prediction layer.

D. 3D Convolution

3D convolution model employs 3D filters that pick up the knowledge of spatiotemporal features from the videos, in contrast to 2D convolution, where the temporal domain is collapsed. In deepfakes, while transferring the appearance to target candidate, if the target candidate has a pose that's not happened in source candidate video then there's a discrepancy. To capture the spatial and temporal irregularities, we took 32 frames per video into consideration. The architecture comprises of 3x3x3 filters in convolution layers and 2x2x2 pool size in maxpooling layers. Number of filters in convolution layers are respectively 64, 128, 256, 256, 512 and 512.

E. Triplet Network

Triplet network is a type of metric learning where the similar features are grouped together and different features are placed large apart in the feature space. The network applies loss to cluster similar features together and dissimilar features farther apart in the feature space. Let's take the anchor input sample A, a sample with the same label as P, and a sample with a different label as N. The loss function of triplet is defined as follows:

$$L(A,P,N)=max(\|f(A)-f(P)\|^2 - \|f(A)-f(N)\|^2 + \alpha, 0) \quad (1)$$

where α is the margin (hyperparameter).

There are three different types of triplet generation methods based upon the distance between anchor, positive and negative embedding vectors.

- Easy Triplets: In this case, the distance between negative and anchor embedding is greater than the distance between anchor and positive embedding plus margin, i.e. $d(a, p) + margin < d(a, n)$. Hence, the loss propagated is zero and it does not help the network to learn anything.
- Semi-hard Triplets: Distance between anchor and negative is between the distance between anchor and positive, and, distance between anchor and positive plus margin, i.e. $d(a, p) < d(a, n) < d(a, p) + margin$. The loss propagated is positive and zero in this scenario.
- Hard Triplets: The distance between the anchor and negative is less than the distance between the anchor and positive plus margin, i.e. $d(a, n) < d(a, p) + margin$ Hence, the loss propagated backward is always positive in this case.

F. Architecture

Our final architecture uses a triplet network to discriminate between the fake video and real video embedding vectors in both Celeb-DF and FF++ datasets. MTCNN extract faces from the frames, then, facenet generates 512 dimension embeddings for each face in the feature space. As facenet is developed for face recognition, each unique face occupies a small cluster in the feature space. Then, we generate semi-hard triplets via online triplet mining. Using these triplets, the embeddings of fake frames and positive frames are distinctively separated through triplet loss. Embeddings is visualized in Figure 6. To the best of our knowledge, the use of metric learning for deepfakes video classification has not been explored yet.
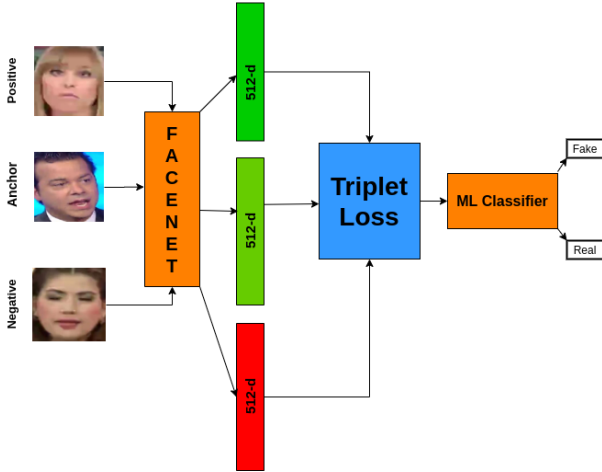

Fig. 3. Triplet architecture used for clustering and classification of fake and real videos embeddings.

IV. EXPERIMENTS ANALYSIS

A. Datasets Review

We analyzed our video classification approaches on two datasets Celeb-DF and FF++ dataset. The training and testing set is provided by the dataset moderators.

1) FF++ (c40): The dataset comprises four types of forgery videos namely: DeepFakes, Face2Face, FaceSwap, and NeuralTextures. Each category contains 1000 videos taken from YouTube randomly. There are a total 1000 pristine and 4000 forged videos. Frame size in raw, medium compressed and high compressed videos are 1080p, 720p and 480p respectively. Videos are captured in H.264 format, with compression factors of 0(raw), 23(medium) and 40(high).

2) Celeb-DF: Celeb-DF comprises of 52 celebrities whose interviews are available on YouTube. They considered various factors such as gender, age and ethnic group bias to make the dataset more challenging. They created 5639 deepfake videos by swapping the faces amongst 59 celebrities. The frame size is arbitrary in these videos. The video format is MPEG4.

B. Implementation Details

We used frame rate of 1 to save all the frames. Then, we analyzed the behavior of our approaches by increasing the number of frames from 10 frames per video until 25 frames per video.

1) Celeb-DF: Initially, with frame rate 5, the number of negative frames extracted was approximately 66.5K frames and the number of positive frames was 9.5k. Due to large data imbalance between fake and real videos, directly fine-tuning models pre-trained on imagenet data leads to poor performance. To counter the effect of data imbalance, we employed the bagging paradigm. Dividing the dataset into seven equal parts, we considered 1400 videos each time for training, totaling to approximately 19K frames. As the Imagenet dataset contains images from a varied number of classes such as plant life, sports, animals, geological formation, and person, we allowed the gradients to flow through all the layers of Xception, with the face data, to make the lower layers focused on facial attributes. As we had enough number of frames to train, 22 million parameters were optimized after 30-50 epochs. With this approach, we produced 7 prediction outputs and took the maximum voting. We got the highest accuracy score of 99.8% and AUC score of 99.2%. Using semi-hard triplets, the AUC score got boosted to 99.7%.
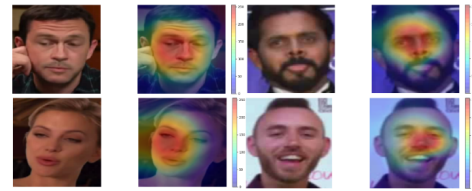

Fig. 4. GradCAM [16] activation maps. From the above figure, we can see that the network focuses on the facial features helpful for binary classification.

2) FF++: From Rossler *et al.* [3] paper, we can see that with raw and c23 compressed video classification accuracy is almost around 99% and 97%, whereas, in compression factor 40, the accuracy is below 90%. Amongst all types of forgeries in the FF++ dataset, detecting NeuralTexture forgery is the most difficult as the accuracy goes down to below 80%. To

Method	AUC Score
MesoInception4	53.6
Two-Stream	53.8
Muti-task	54.3
HeadPose	54.6
Meso-4	54.8
VA-MLP	55.0
VA-LogReg	55.1
FWA	56.9
Capsule	57.5
DSP-FWA	64.6
Xception	65.5
Ours (Frames only)	99.2
Ours (Triplets)	**99.7**

TABLE I
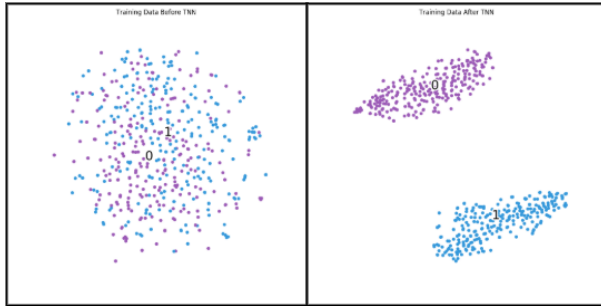AUC Scores on Celeb-DF dataset.

Fig. 5. Celeb-DF features TSNE plot before and after training. *Best viewed in color.*

start with, like Celeb-DF we extracted frames with a frame rate of five. We got around 29.5K number of frames. Amongst all imagenet models, XceptionNet outperformed other models. In [3], the authors used 243K frames for training and 34K frames for validation. With significantly reduced training data of 29K frames, we got an accuracy of around 50%.

After that, we added recurrence networks to make the network learn the sequences. It is with this idea that while transferring the source face on target face, there's information along the temporal dimension due to pose variation. From the trained XceptionNet on FF++ dataset, we generate embeddings from the first *fc* layer of feature vector dimension 512 of 25 frames. However, in this case, too, the accuracy was around 55% only. Generating features from trained end-to-end Xception was crucial, otherwise, the LSTM network overfits on the embedding data. It improved the accuracy by 2-3%, but it was still below par.

To combine the features along spatial and temporal dimensions, we used 3D convolution. Without prior knowledge as in imagenet networks, Conv 3d performs worse than even basic training using frames only.

For the triplet network, we initially took 29K frames and then generated a 512 dimension vector from a trained Xception network. As the Xception network learned nothing even after increasing the number of training frames to 43K, the embeddings generated did not provide us with any substantial gain while applying triplet loss to them. Then, we used FaceNet [17] to generate face embeddings of a dimension of 512. These

Model	Accuracy
Frames only	49.82
3d Convolution	43.47
Frames + LSTM/GRU	55.8
Triplets (Semi-hard)	**86.74**

TABLE II
FaceForensics++(c40) accuracy performance

face embeddings are clustered into n number of small groups in the feature space. Now, when we apply triplet loss to these embeddings, the network learns the discriminative features to cluster the embeddings of original and manipulated faces separately. We analyzed triplet performance by random pick and online semi-hard negative triplet mining.

C. Performance Analysis

In Celeb-DF, training the network with XceptionNet, we used Nadam [18] optimizer with a learning rate value of 0.002 and a schedule decay of 0.004. We also tried with multi-loss function as mentioned in [9] but the performance gain was insignificant. From Table I, we can see that using MTCNN for face extraction and training Xception Net on faces outperforms the AUC scores of all the previous approaches. With only 19K frames as compared to the FF paper of using 240K frames, we acquired an accuracy of 96%. By bagging and boosting algorithms, our accuracy got boosted by 2-3% to 98%. On the FF++ dataset, we look into several methods for video classification. We reported the accuracy, recall, precision and F1 score in Table II & Table III. In LSTM and C3D, we increased the number of sequence frames per video from 16 to 32, but, that does not help to distinguish between the authentic and tampered faces. Using triplet loss, however, provides us with sharp gain over traditional deep learning approaches. From Fig. 6, we can see that initially embeddings after facenet are clustered all over the feature space. Without training, just simple classification, the embeddings are mixed all together and are inseparable. After applying triplet loss, we have shown that there is a distinct boundary between the fake videos and real videos. On top of triplet loss, we applied Random Forest (RF) and Stochastic Gradient Descent (SGD) to do the binary classification of videos. We have shown an AUC and EER score of 92.9% and 10.07% that shows the robustness of our approach. When the false rejection rate and false acceptance rate are equal, then we calculate the equal error rate. The minimum the error, the better the network.

Model	P	R	F1	AUC	EER
Triplets + RF	89.84	82.73	86.14	92.85	12.14
Triplets + SGD	**90.55**	**82.74**	**86.47**	**92.9**	**10.07**

TABLE III
Models performance comparison using Semi-Hard Triplets.
Here, P=Precision, R=Recall and EER = Equal Error Rate.

V. Conclusion and Future Work

In this work, we presented a deep study for binary classification of deepfake videos. We analyzed different approaches
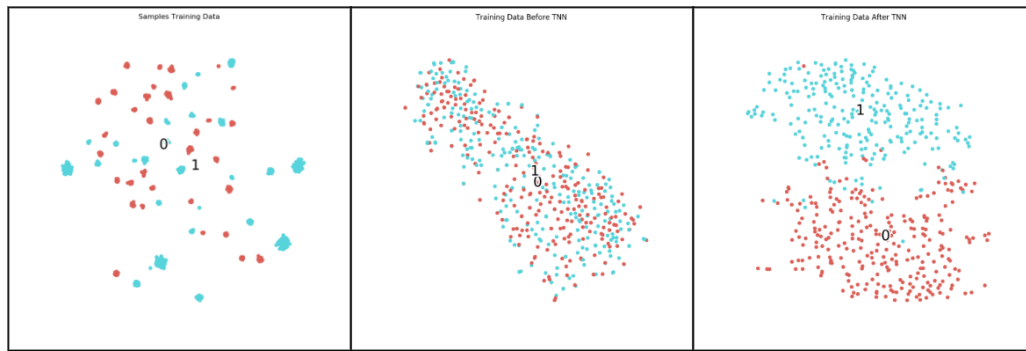
Fig. 6. TSNE plots of FF++ dataset: a) Initial face embeddings vector generated from facenet architecture; b) Distribution of embeddings without triplet network; c) Distribution of embeddings in feature space after applying triplet loss. *Best viewed in color.*

Model	NeuralTexture	Pristine
Steg. Features + SVM [14]	55.84	56.94
Cozzolino *et al.* [11]	62.15	56.27
Bayar and Stamm [10]	74.36	53.87
Rahmouni *et al.* [13]	59.99	56.79
MesoNet [9]	44.81	77.58
XceptionNet [12]	78.06	75.27
Ours	**90.71**	**82.73**

TABLE IV
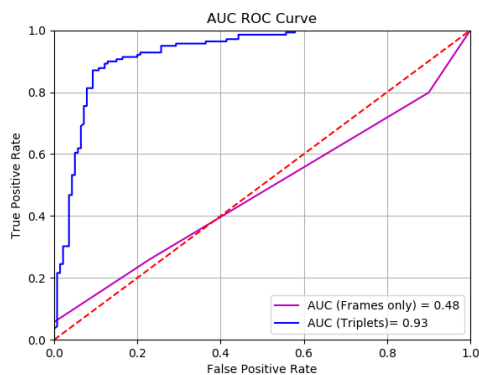COMPARISON OF OUR APPROACH TO PREVIOUS METHODS.

Fig. 7. AUC ROC Curve plots of frames only and triplets network.

to improve the video classification in high compression factors using less amount of data. Using the Triplet network, we outperform the previous results by a substantial margin utilizing only 25 frames per video. We also studied the effects of deep imagenet architectures on the second-generation deepfakes dataset. Till now, the major limitation of the approaches is their generalizability across different datasets. In the future, our aim is to use unsupervised domain adaptation to adapt the feature space from source dataset to target dataset, to make our model robust and label independent.

REFERENCES

[1] L. Verdoliva, "Media forensics and deepfakes: an overview," 2020.
[2] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "Deepfakes and beyond: A survey of face manipulation and fake detection," 2020.
[3] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," *CoRR*, vol. abs/1901.08971, 2019. [Online]. Available: http://arxiv.org/abs/1901.08971
[4] P. S. H. Q. Yuezun Li, Xin Yang and S. Lyu, "Celeb-df: A new dataset for deepfake forensics," *arXiv preprint arXiv:1909.12962*, 2019.
[5] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nieundefinedner, "Face2face: Real-time face capture and reenactment of rgb videos," *Commun. ACM*, vol. 62, no. 1, p. 96–104, Dec. 2018. [Online]. Available: https://doi.org/10.1145/3292039
[6] Y. Nirkin, Y. Keller, and T. Hassner, "FSGAN: Subject agnostic face swapping and reenactment," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7184–7193.
[7] J. Thies, M. Zollhöfer, and M. Nieundefinedner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Trans. Graph.*, vol. 38, no. 4, Jul. 2019. [Online]. Available: https://doi.org/10.1145/3306346.3323035
[8] T. T. Nguyen, C. M. Nguyen, D. T. Nguyen, D. T. Nguyen, and S. Nahavandi, "Deep learning for deepfakes creation and detection," 2019.
[9] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "Mesonet: a compact facial video forgery detection network," *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–7, 2018.
[10] B. Bayar and M. 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*, ser. IH&MMSec '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 5–10. [Online]. Available: https://doi.org/10.1145/2909827.2930786
[11] D. Cozzolino, G. Poggi, and L. 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*, ser. IH&MMSec '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 159–164. [Online]. Available: https://doi.org/10.1145/3082031.3083247
[12] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807, 2016.
[13] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizen, "Distinguishing computer graphics from natural images using convolution neural networks," in *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, Dec 2017, pp. 1–6.
[14] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, June 2012.
[15] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, Oct 2016.
[16] R. Kotikalapudi and contributors, "keras-vis," https://github.com/raghakot/keras-vis, 2017.
[17] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
[18] T. Dozat, "Incorporating nesterov momentum into adam," 2015.