Deepfake Detection using Capsule Networks and Long Short-Term Memory Networks

Akul Mehra, Luuk Spreeuwers and Nicola Strisciuglio
Data Management and Biometrics Group, University of Twente, Enschede, The Netherlands

Keywords: Deepfake Detection, Face Video Manipulation, Capsule Networks, Long Short-Term Memory Networks.

Abstract: With the recent advancements of technology, and in particular with graphics processing and artificial intelligence algorithms, fake media generation has become easier. Using deep learning techniques like Deepfakes and FaceSwap, anyone can generate fake videos by manipulating the face/voice of target subjects in videos. These AI synthesized videos are a big threat to the authenticity and trustworthiness of online information and can be used for malicious purposes. Detecting face tampering in videos is of utmost importance. We propose a spatio-temporal hybrid model of Capsule Networks integrated with Long Short-Term Memory (LSTM) networks. This model exploits the inconsistencies in videos to distinguish real and fake videos. We use three different frame selection techniques and show that frame selection has a significant impact on the performance of models. The combined Capsule and LSTM network have comparable performance to state-of-the-art models and about $1/5^{th}$ the number of parameters, resulting in reduced computational cost.

1 INTRODUCTION

Deepfake technology has been applied recently to tasks like de-aging people, lip-sync, and face-swapping. These applications are successful especially in the media industry, e.g. using lip-sync for dubbing a movie into another language while keeping it realistic and entertaining for the viewers. Several deepfake videos available on the web usually involve the face of a famous movie actor that has been swapped onto the face of another actor in other movies. Figure 1 shows an example frame from a deepfake video generated by Facebook on making pour-over coffee.

Figure 1: Deepfake video: how to make pour-over Coffee.

Although the advancement in deep learning and deepfake technology has many beneficial applications in daily life, business, and the film industry, it can also serve malicious purposes. In a recent example of an application of the lip-sync technology (Peele, 2018), where Barack Obama, the former president of the USA, was used to make an unpleasant statement about the current president Donald Trump. This video was an impersonation done by the famous actor/comedian Jordan Peele and deepfaked. It showed how deepfake technology can produce realistic videos and how it can influence the general public opinions when used with wrong motives. Deepfakes have already been used for fraudulent use-cases: in (Damiani, 2019), the authors described the first case where deepfake audio was used to scam the CEO of a UK-based energy firm and rob €220,000. Due to the rise of deepfakes, it is easier to cause misinterpretation of videos, spread lies, and misinformation. This kind of fake news is causing individuals to lose trust in what is real and what is not. Hence, there is a need to provide robust algorithms that can detect deepfake content and help in preventing them before they can spread misinformation.

In this paper, we propose an algorithm for the identification of deepfake videos, based on the combination of a Capsule Network (CapsuleNet) for frame-level representation with long short-term memory (LSTM) networks for the creation of a spatio-temporal hybrid model. We also analyze the impact of the selection of frame sequences on the de-

Mehra, A., Spreeuwers, L. and Strisciuglio, N.
Deepfake Detection using Capsule Networks and Long Short-Term Memory Networks.
DOI: 10.5220/0010289004070414
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 4: VISAPP*, pages 407-414
ISBN: 978-989-758-488-6

tection of fake videos. By visualizing the activation of capsules, we also explain what image inconsistencies are detected by Capsule Network to classify a sample as fake or real. We compare the results that we achieved with the proposed model with those obtained by state-of-the-art approaches. With the vast amount of video data available on the Internet, the efficiency of the fake video detection algorithm is of utmost importance. The combined Capsule and LSTM network have comparable performance to state-of-the-art models and about $1/5^{th}$ the number of parameters, resulting in reduced computational cost.

The paper is organized as follows. We present the related work about deepfake detection in Section 2, and the proposed methods in Section 3. In Section 4, we describe the data-set, the performance metrics, and the evaluation protocol that we used in the experiments. We present and discuss the results that we achieved in Section 5. Finally, in Section 6, we conclude with our findings.

2 RELATED WORK

2.1 Fake Media Detection

The earlier generation of deepfake videos was not as realistic as the actual ones and was easier to detect. They normally produced videos that showed various kinds of physical inconsistencies, such as no eye-blinks, missing reflections, or distorted parts of the faces, and earlier detection models took advantage of these inconsistencies to discriminate between fake and real videos.

XceptionNet classifier is a traditional CNN with pre-trained weights of ImageNet. (Rössler et al., 2019) provides an overview of the detection performance, where multiple models using steganalysis and CNN based networks are evaluated on the FF++ (FaceForensics++) data-set. XceptionNet performs best in all face manipulation techniques and achieves the state-of-the-art accuracy of 96.36%.

ConvolutionalLSTM (Guera and Delp, 2018) and **RecurrentConvolutional** (Sabir et al., 2019) are temporal-aware pipelines to identify deepfakes. The proposed model consists of a combination of a CNN, for frame feature extraction combined with an LSTM for temporal sequence analysis. As the deepfakes are generated frame-by-frame, each frame has a new face generated which will have inconsistencies when compared to every other frame and therefore, lacks temporal awareness between frames. These temporal inconsistencies such as flickering in frames and inconsistent choice of illuminants are used to detect deepfakes and result in an accuracy of ~97% on their data-set in ConvolutionalLSTM and 96.9% on the FF++ data-set in RecurrentConvolutional. The difference with ConvolutionalLSTM is that they use pre-trained CNNs while RecurrentConvolutional models are trained end-to-end.

Capsule (Nguyen et al., 2019) uses capsule structures for deepfake detection. The architecture is based on a previous paper that used capsule networks for forgery detection and forensics (Nguyen et al., 2018). The model uses the VGG19 network as the backbone for deepfake detection. Although CapsuleNet achieves 92.17% accuracy and XceptionNet achieves 94.81% for deepfakes in multi-class classifications, CapsuleNet has a more balanced performance for all labels in the FF++ data-set.

2.2 Capsule Network

Although CNNs perform well in the domain of computer vision, they have limitations when applied to inverse graphics. The pooling layers in CNNs cause loss of information and have local translation-invariance, which does not allow to describe the position of one object relative to another. (Hinton et al., 2011) addressed these limitations and proposed the capsule architecture to overcome these drawbacks. With the recent developments of dynamic routing and expectation-maximization routing algorithms, capsule networks have been implemented with remarkable results and outperform CNNs on several object classification tasks. In (Sabour et al., 2017), a capsule network achieved 79% accuracy on an affine test set whereas the traditional CNN model achieved 66% accuracy. These developments introduced 1) dynamic routing-by-agreement and replaced the max-pooling of CNN, and 2) squashing that replaced the scalar output feature detectors of CNN with vector output capsules. The agreement between capsules that preserves the pose information enables the capsule networks to enclose more information than a CNN with less training data required.

Capsule networks are also used for forensics and forgery detection. In (Nguyen et al., 2019), the author proposed an improved capsule-forensics network for detecting fake videos. The method achieved equivalent or better scores in comparison to state-of-the-art methods while using fewer parameters and hence, less computational cost. These advancements in several domains have motivated us to study and work with capsule networks for deepfake detection.

3 METHOD

Deepfake algorithms tamper with faces in the video by performing modifications frame-by-frame. The modified face in a frame might be different from those in other frames as the generation algorithms do not keep track of previous faces. This creates temporal inconsistencies between frames and can be exploited as a sign of tampering for deepfake detection. We thus combine the spatial description power of CapsuleNet with a recurrent neural network, to train a model to detect these temporal inconsistencies and identify deepfakes. We propose a spatio-temporal hybrid model that will exploit and detect the inconsistencies in both the spatial domain and temporal domain and identify a video as a deepfake or not.

3.1 Pre-processing

3.1.1 Frame Selection

We select 10 frames from each video to be used to train and evaluate the models. We use three different methods for frame selection and compare the performance of the proposed model. The three frame selection methods that we use are the following:

1. **First-10:** extract the First-10 frames from each video (see Figure 2a).
2. **Equal Interval:** extract 10 frames from each video with a 1-sec interval as shown (see Figure 2b).
3. **Most Changes:** select the interval of the video, of duration one second, that contains the most changes of visual appearance. The method consists of the following steps:
   (a) select 10 frames from a video at intervals of one second, compute a measure of the structural similarity (SSIM) between two consecutive frames among the 10 selected frames, and select the pair of frames that has the least SSIM;
   (b) select ten equally-spaced frames from the selected interval including the start and endpoint (see Figure 2c).

The First-10 method captures the first ten frames where the difference between the consecutive frames is least. The Most Changes method captures the transition between the most changing frames in an interval one second long and the difference between frame i and frame $i+1$ is large, which may highlight inconsistencies due to the tampering. The Equal Interval method selects 10 frames at equal intervals and the difference between frame i and frame $i+1$ is bigger than both the First-10 and Most Changes method.

Figure 2: Frame selection methods: (a) First-10 (b) Equal Interval (c) Most Changes: ten frames extracted between frame 150 and frame 180 having the least similarity score.

3.1.2 Face Detection and Cropping

As the deepfake generation algorithms focus on the face area to perform video manipulation, we only detect the face region by performing face detection and cropping. However, we include in the crop also the pixels surrounding the face, which helps in capturing the spatial inconsistencies around the face boundaries. We use pixel padding equal to 0.7 of the face crop size, such that the total width of the cropped region is 1.7 times the actual cropped face. Finally, we resize all images to 224x224 pixels.

For face detection, we use a detection algorithm based on a deep network, namely Mobilenet SSD as it has high performance with low computational cost. In case more than one person is present in the video, at each frame we select and crop the face that is detected with higher confidence in the first frame.

3.1.3 Data Augmentation

As augmentation makes the model robust, we perform additional augmentations provided by Albumentations (Buslaev et al., 2020). We perform one of the augmentations: rgbshift, random brightness/contrast, random gamma, hue saturation value shift with a probability $p = 0.1665$ and jpeg-compression with $p = 0.334$ on 33% of the training data. Additionally, we perform Horizontal Flip with probability $p = 0.5$. We normalize the images using mean=(0.485, 0.456, 0.406) and standard deviation=(0.229, 0.224, 0.225).

3.2 Proposed Model

We propose a method that combines a CapsuleNet with an LSTM network and aims at finding spatio-temporal inconsistencies in sequences of frames of deepfake forged videos. The method can be split into two parts: the CapsuleNet, which acts as a feature extractor and identifies spatial inconsistencies in a single frame, and the LSTM, which takes a sequence of feature vectors extracted by CapsuleNet from a sequence

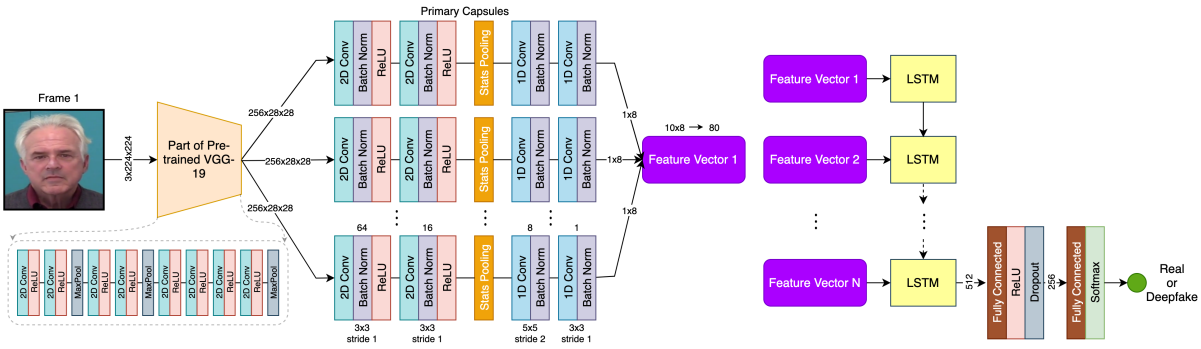


Figure 3: Detailed Architecture: CapsuleNet + LSTM Model.

of frames as input and identifies temporal inconsistencies across the given sequence of frames. The motivations behind using capsule networks are twofold. On the one hand, CapsuleNets are more robust, preserve pose information, and are equivariant in parameters like translation, rotation, scale, etc. solving some of the limitations of convolutional networks. It does not only detect features but also estimates their orientation and the spatial relation among them.

On the other hand, a CapsuleNet requires fewer parameters than a CNN while achieving similar performance. This is made possible by the squashing function and the dynamic routing-by-agreement algorithm for training. Spatial inconsistencies in deepfakes are usually blurred and flickering of faces, and no reflection in the eyes. Therefore, training a deep learning model to detect these inconsistencies will help in identifying deepfakes. In this paper, we investigate the use of CapsuleNet as an alternative to a traditional CNN for detecting spatial inconsistencies.

For the CapsuleNet part of our architecture, we use the capsule forensics model (Nguyen et al., 2019) and remove the output capsules to extract feature vectors as output. The model uses part of the pre-trained VGG-19 (until the third max-pooling layer) as a feature extractor and is equivalent to the CNN part of the original capsule network architecture. After the features are extracted from the CNN, they are passed to multiple capsules, each with different weights initialized from a normal distribution.

Using too few capsules limits the extent of detectable features (Nguyen et al., 2018). From our experiments, we observed that a large number of capsules induces the model to overfit the training samples. Therefore, we configure our model to use 10 capsules. Each capsule consists of a 2D convolution, a statistical pooling, and a 1D convolution. The statistical pooling layer was demonstrated to be effective for improving the performance of network training on forensics and forged video detection task. The statistical pooling layer includes mean and variance filters, respectively computed as follows:

$$\mu_k = \frac{1}{HxW} \sum_{i=1}^{H} \sum_{j=1}^{W} I_{kij} \tag{1}$$

$$\sigma_k^2 = \frac{1}{HxW - 1} \sum_{i=1}^{H} \sum_{j=1}^{W} (I_{kij} - \mu_k)^2 \tag{2}$$

where H and W are the height and width of the filter respectively, k is the layer index, and I is the 2-dimensional filter array. The output of the statistical pooling layer is 1-dimensional, which is then processed with a 1D convolution. The output of a capsule is an 8-dimensional feature vector. Using 10 capsules and flattening their outputs determine an output feature vectors of 80 elements that encode a spatial description of the input face image. These features help in detecting spatial inconsistencies in a given frame.

We perform the feature extraction process using a CapsuleNet on 10 frames of a video to get 10 feature vectors of size 80 each. These vectors are then given as an input to a single layer LSTM model with 512 hidden units that captures the temporal inconsistencies across multiple frames. The output of the last LSTM cell is then passed through a fully connected layer of output size 256, followed by ReLU activation and a dropout layer with a coefficient of 0.5, which helps to avoid overfitting. Subsequently, the output of the dropout layer is further processed with a second fully connected layer and a softmax function that provides a probability score between 0 (real) and 1 (fake). The second fully connected layer contributes to achieving better classification performance. A cross-entropy loss function and an AdamW optimizer are used to train the model parameters, with a learning rate of 10^{-3} and a weight decay of 10^{-4}. The proposed method thus uses both spatial and temporal features to identify a given sequence of frames from a video as real or fake. The detailed model architecture is shown in Figure 3.

4 EXPERIMENTS

4.1 Data Set

We carried out experiments on the DFDC data set (Dolhansky et al., 2020), constructed by AWS, Facebook, Microsoft, and the Partnership on AI along with other academic partners. The data set is part of the Deepfake Detection Challenge, which aims to develop machine learning models that can help to detect real and manipulated media content.

The dataset contains over 470GB of videos (19,154 real videos and 100,000 fake videos), recorded using 486 actors. Each video has a duration of about 10 seconds and is generated using 4 different deepfake generation techniques, namely Deepfake Autoencoder, MM/NN face swap, Neural Talking Heads, and Face Swapping GAN. No data augmentations are performed. Additionally, a test set is available that is used for performance comparison on Kaggle, namely for the Public Leaderboard. The Public Test Set is collected in the same way as DFDC and contains 4,000 videos (2,000 real videos and 2,000 fake videos) from 214 actors who do not perform in the DFDC data-set. The major differences with respect to the DFDC dataset are that it includes videos generated with one additional deepfake generation technique, namely StyleGAN, combined with heavy data augmentation.

We pre-process the dataset and prepare the video samples for our experiments as follows. First, we perform a subsampling. As the dataset is imbalanced with 100,000 fake videos and 19,154 real videos, we randomly subsample the fake videos such that the final data-set is balanced with 19,154 fake and 19,154 real videos. Subsequently, we split the dataset into training and test sets. As the DFDC data-set is provided in 50 parts, we perform folder-wise split to avoid mixing of actor videos across multiple folders, so that we ensure the test videos do not contain actors that are in the training videos. We use folders 0-39 for Training, 40-44 for Validation, and 45-49 for Testing.

4.2 Performance Metrics

For evaluating our model performances, we compute two metrics, namely accuracy and the area under the ROC curve (AUC). The accuracy is the ratio of correctly classified observations to all classification outputs. The AUC is a measure used for comparing the performance of classifiers. AUC is equal to the probability that the classifier will rank a randomly sampled positive example higher than a randomly sampled negative example.

4.3 Baseline Models

We compare the performance of the proposed architecture with that of the following approaches:

CapsuleNet: we use the Capsule Forensics approach (Nguyen et al., 2018), which we refer to as CapsuleNet. We configure the number of capsules as in the backbone of our model, i.e. 10 capsules. The input is a single frame and the output is the probability of it being real or fake.

XceptionNet: the XceptionNet network (Rössler et al., 2019) achieved the highest performance on the benchmark data-sets for deepfake detection. We use the pre-trained model and replace the last layer with a set of custom layers: we deploy a sequence of a connected layer (2048 to 512 units), and a final fully connected layer that transforms a 512-dimensional output into a scalar number.

4.4 Experiments

We carried out different experiments to evaluate the impact of different frame selection strategies on the performance of the proposed method in comparison with those of the methods mentioned in Section 4.3 and validate the influence that they have on the quality and generalization capabilities of the trained models.

We compare the XceptionNet model with the Capsule Network when they are trained using a single frame selected from each video: we focused on the contribution that spatial features only have for the detection of fake videos. We also compare the performance results of XceptionNet and CapsuleNet when using the frame-by-frame selection strategy (i.e. Average): the models are trained on multiple frames taken from each video to learn the spatial features. In the test phase, the predictions on multiple frames of a video are averaged to classify a test video as real or fake. To train the spatio-temporal model, i.e. CapsuleNet+LSTM, we deploy a multiple frame strategy: the models are trained on sequences of frames taken from the training videos to learn spatio-temporal features of the deepfake inconsistencies. In the test phase, test sequences are classified at once by the LSTM part of the networks.

We performed further experiments to provide insights about the regions of the frames that the networks focus on to perform the classification. We provide the CapsuleNet+LSTM model with sequences extracted from real and deepfake videos and visualized the activation maps of the capsule units using the open-source tool Grad-CAM (Ozbulak, 2019).

5 RESULTS AND DISCUSSION

5.1 Results

In Table 1, we report the results of the proposed methods and compare them with those achieved by existing methods. The combined CapsuleNet+LSTM model achieves an accuracy value, on the DFDC test set, of ~5% higher than that of the CapsuleNet model that classifies fake and real videos on a single frame. When compared to the CapsuleNet that uses an average of multiple frames, it achieved an accuracy higher by about ~2-2.5%. The LSTM contributes to extend the capability of the CapsuleNet model to detect spatial inconsistency by taking into account the temporal characteristics of such artifacts.

When compared to the baseline XceptionNet model, our model achieved lower performance on the DFDC test set. The performance gap between the two models is, however, small: the difference in accuracy is ~3.3% in the case of the CapsuleNet + LSTM model vs XceptionNet (average of multiple frames). On the public test set, the proposed spatio-temporal model achieved an accuracy value of 78.38%, which is the same as that obtained by the XceptionNet.

In general, we observed that the proposed CapsuleNet and CapsuleNet+LSTM models are relatively more robust than Xception-based models when tested on data with characteristics not included in the training data. The drop of accuracy observed when testing the considered models on the public test set is, indeed, relatively lower for the CapsuleNet-based models with respect to that of the XceptionNet model. The reason for these results may be due to the augmentations and the use of an additional deepfake technique applied for the generation of the public test set. This shows that our model is more robust towards unseen deepfake generation techniques and heavy augmentations and achieves similar performance to the state-of-the-art model.

5.2 Impact of Frame Selection

When using a single frame extracted from a video to train the models and subsequently detect deepfake alterations in videos, it can be observed that XceptionNet outperforms CapsuleNet (by ~6% in accuracy). Similar results are achieved when selecting an average of 10 frames, with a slight improvement of accuracy achieved by both models.

In general, the selection of random frames does not take into account in which parts of the video the inconsistencies occur. Hence, there is a chance that one may select a real frame from a deepfake video and the single-frame model will classify it as a real video. In Figure 4, we show the results of using the CapsuleNet on a real video and a deepfake video to predict if every single frame is fake or not. As can be seen, the model for the deepfake video predicts some frames as real. Hence, it is better to consider a sequence of frames in comparison to a single frame to detect deepfake videos.

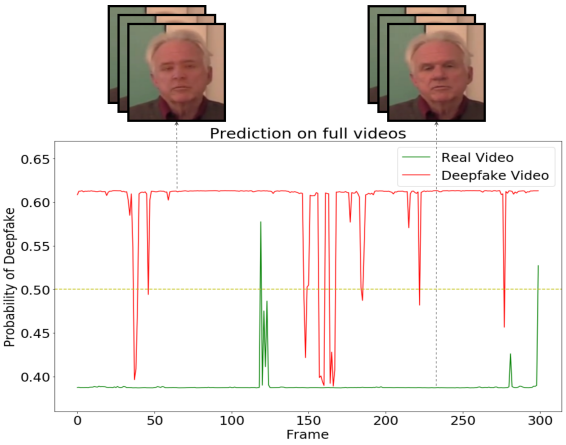


Figure 4: Output of CapsuleNet on a real and a fake video.

We report the results that we achieved using different frame selection strategies in Table 1. The *Equal Interval* frame selection method consistently contributes to achieving higher performance results than *First-10* and *Most Changes* methods for all the considered models. We observe that, in the case of selecting the First-10 frames, the impact of the frame selection method is negligible, i.e. an increase of ~0.1-1.6% in accuracy for the baseline models, while for the temporal based model, i.e. CapsuleNet+LSTM, the impact is larger. CapsuleNet+LSTM has a higher increase of ~5.6% from First-10 and ~1.9% from Most Changes in accuracy in comparison to Equal Interval. Similarly, when comparing the performance of frame selection methods on the public test set, the *Equal Interval* selection method impacts positively on the performance of all the models and contributes to achieving higher results in comparison to First-10 and Most Changes selection techniques. Hence, the models better detect spatial and temporal inconsistencies in videos when using Equal Interval frames. The frame selection method for detecting fake videos is an important aspect and the proposed *Equal Interval* approach achieves the best performance.

We show the ROC curves that we achieved using different frame selection methods on the public test set using the CapsuleNet + LSTM model in Figure 5. This analysis confirms that Equal Interval has the best performance, followed by Most Changes and then the First-10 method for each data-set.

Table 1: Comparison of the results of CapsuleNet- and XceptionNet-based models on the DFDC test set and public test set.

Model	Frame Selection	DFDC Test		Public Test	
		Accuracy	AUC	Accuracy	AUC
CapsuleNet	Single Frame	78.49%	0.8516	71.65%	0.7837
XceptionNet	Single Frame	84.48%	0.8883	75.50%	0.8153
CapsuleNet	First-10 (Average)	79.36%	0.8684	71.63%	0.7997
XceptionNet	First-10 (Average)	85.50%	0.9359	76.83%	0.8674
CapsuleNet + LSTM	First-10	77.77%	0.8599	72.73%	0.8059
CapsuleNet	Equal Interval (Average)	80.96%	0.8996	73.63%	0.8241
XceptionNet	Equal Interval (Average)	86.78%	0.9571	**78.99%**	**0.8863**
CapsuleNet + LSTM	Equal Interval	83.42%	0.9115	**78.38%**	**0.8567**
CapsuleNet	Most Changes (Average)	79.27%	0.8774	72.28%	0.8096
XceptionNet	Most Changes (Average)	86.27%	0.9460	77.34%	0.8744
CapsuleNet + LSTM	Most Changes	81.54%	0.8873	74.67%	0.8308

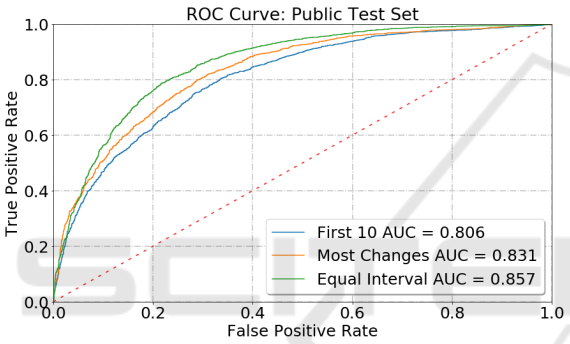


Figure 5: ROC curves achieved by the proposed CapsuleNet+LSTM model that uses different frame selection techniques on the public test set.

5.3 Computational Complexity

Although XceptionNet-based models achieve slightly better results on the DFDC test set and comparable on the public test set, the proposed CapsuleNet + LSTM network is much smaller in size than XceptionNet, requiring fewer parameters to be tuned during training. As shown in Table 2, the number of parameters of CapsuleNet + LSTM is about $1/5^{th}$ of that of XceptionNet, while in the case of size, CapsuleNet + LSTM is $1/4^{th}$ of that of XceptionNet. Hence making it lighter and with reduced computational requirements. The much smaller number of parameters and comparable performance results are indications that the proposed model is less prone to overfitting and generalizes better on unseen data. The proposed model required fewer resources and power and can be used in distributed systems or integrating into online social media platforms for real-time identification of deepfakes at a lower computational cost.

Table 2: Comparison of the size of the models.

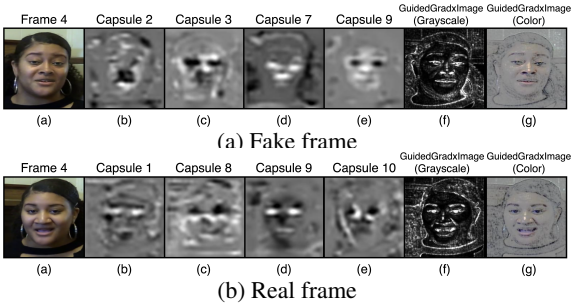
Model	Params	Size
CapsuleNet	2.79 M	6.3 MB
CapsuleNet + LSTM	4.03 M	21.1MB
XceptionNet	21.86M	87.8MB

5.4 Visualization of Spatial Features

We visualize the response maps of the feature learned in the capsules of the proposed spatio-temporal models for real and fake video, and show the results in Figure 6, where we report the input image (a) and the corresponding feature response maps (b)-(g).

In Figure 6a, (b) focuses below the mouth region, (c) focuses outside the eyes and nose region, (d) focuses on eyes and nose, (e) focuses on the whole face excluding the eyes, (f) and (g) are the output of Guided Grad X Image in grayscale and color. The capsules mostly focus on the facial regions of the whole face when classifying the given sequence of frames as fake. In Figure 6b, (b) focuses below the eyes region, (c) focuses on the lower face, (d) focuses on the eyes, (e) focuses on the region around the eyes, while (f) and (g) are the output of Guided Grad X Image in grayscale and color. The capsules mostly focus on facial regions around the eyes, nose, and mouth when identifying the given sequence of frames as real.

Most capsules focus on facial areas while some capsules fail to detect the manipulated regions. However, with multiple capsules, these features are collected and combined as spatial features. Combining these features across multiple frames to get temporal features, the LSTM-augmented models learn to overcome issues of the spatial-only features and detect inconsistencies across spatial and temporal domains.



(a) Fake frame

(b) Real frame

Figure 6: Capsule visualizations: (a) is the input image. (b)-(g) are the Grad-CAM visualizations.

6 CONCLUSIONS

This paper presents a spatio-temporal model based on a CapsuleNet integrated with an LSTM network for deepfake detection. We observed that the capsules learn different facial features in the regions of the eyes, outside eyes, nose, mouth, and whole face excluding eyes for both real and fake videos, focusing on areas where spatial inconsistencies due to the deepfake alterations occur. The LSTM network combines spatial features over time and focuses on temporal inconsistencies of local features.

On the DFDC test set, our model achieves an accuracy of 83.42%, which is ~3.3% lower than the state-of-the-art model. On the contrary, on the public test set, the proposed CapsuleNet+LSTM model and XceptionNet achieve similar accuracy (~78%). The substantial drop in the performance of XceptionNet is attributable to a lack of generalization of data that has undergone a heavy augmentation process in the public test set. The proposed CapsuleNet+LSTM model is able to generalize better on data with unseen modifications and fake videos created with new tampering techniques. The frame selection method has a significant impact on performance. Equal Interval achieves the best performance, followed by Most Changes and First-10 in each data-set.

The model we propose has around $1/5^{th}$ number of parameters (~4M) as compared to XceptionNet (~22M parameters), achieving comparable accuracy while requiring a lower computational cost. Hence, the proposed model is more suitable to be used in distributed systems and online social media platforms.

Future work could include an ensemble of models and different frame selection techniques for the improvement of deepfake detection.

REFERENCES

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., and Kalinin, A. A. (2020). Albumentations: Fast and Flexible Image Augmentations. *Information*, 11(2):125.

Damiani, J. (2019). A Voice Deepfake Was Used To Scam A CEO Out Of $243,000.

Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M., and Ferrer, C. C. (2020). The DeepFake Detection Challenge Dataset.

Guera, D. and Delp, E. J. (2018). Deepfake Video Detection Using Recurrent Neural Networks. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6.

Hinton, G. E., Krizhevsky, A., and Wang, S. D. (2011). Transforming Auto-Encoders.

Nguyen, H. H., Yamagishi, J., and Echizen, I. (2018). Capsule-Forensics: Using Capsule Networks to Detect Forged Images and Videos.

Nguyen, H. H., Yamagishi, J., and Echizen, I. (2019). Use of a Capsule Network to Detect Fake Images and Videos.

Ozbulak, U. (2019). CNN Visualizations.

Peele, J. (2018). You Won't Believe What Obama Says In This Video! - YouTube.

Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., and Nießner, M. (2019). FaceForensics++: Learning to Detect Manipulated Facial Images.

Sabir, E., Cheng, J., Jaiswal, A., AbdAlmageed, W., Masi, I., and Natarajan, P. (2019). Recurrent Convolutional Strategies for Face Manipulation Detection in Videos. pages 80–87.

Sabour, S., Frosst, N., and Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 3857–3867.