Effectiveness of Capsule Networks in Detecting Deepfakes Instead of Traditional CNNs

M. C. Weerawardana, T. G. I. Fernando
Department of Computer Science, University of Sri Jayewardenepura, Nugegoda, Sri Lanka

***Abstract*—As artificial intelligence has advanced, computer-generated fake content has become increasingly prevalent. Deepfake is an advanced fake creation generated using deep learning-based technologies, and deepfake images, videos, and voices have spread rapidly. The distinction between real and fake content is very hard for the naked eye, making reliable detection essential. Existing deepfake detection methods have achieved success, but still face limitations in keeping pace with the rapid evolution of deepfake generation techniques. In particular, CNN-based approaches may require a large number of parameters and may not fully capture spatial hierarchies. In this research, we investigate whether Capsule Networks can provide an effective and parameter-efficient alternative for deepfake detection. We proposed four different Capsule Network architectures by altering the size, complexity, and configuration. A comparative analysis is conducted against state-of-the-art Capsule models and CNN models across various datasets, utilizing AUC% and the number of parameters as evaluation criteria. For transparency, we note that some baseline CNN and Capsule models follow the training protocols and datasets reported in their original studies, which may differ across implementations. Our experimental results show that the proposed Capsule Network models achieved over 98% AUC% on the evaluated datasets while using fewer parameters than several CNN-based models. These findings suggest that Capsule Networks exhibit greater efficacy in detecting deepfakes compared to traditional CNN-based methods and represent a promising direction for future research.**

***Keywords*—Deepfake; deep learning; capsule network; convolutional neural network**

I. Introduction

Currently, we are exposed to a new generation fueled by artificial intelligence (AI). Day-by-day, AI generates novel creations that become integrated into the world. AI-related applications and tools play different roles in social media, websites and the Internet. Among many applications in AI, deepfake [2] is a significant and critical technology. Deepfakes involve digitally reenacting and manipulating facial appearances, utilizing deep neural networks to swap faces [1] with someone else's likeness. Deepfakes are generated in a complex process using deep neural networks. At present, lots of deepfake-generated tools like GAN [3], FaceSwap-GAN [4], FaceApp [5], Face2Face [6], and DeepFake [7] are available freely. Deepfake generates fake images, voices, and videos. The speciality of these deepfake-generated forgeries is as real as seen by the naked eye. Deepfake creation tools have been developed with new technologies and release highly accurate and quality outputs. The produced fake information can mislead society by spreading fake news, defaming celebrities and public figures, creating pornographic videos for blackmail, or perpetrating financial fraud [2].

The challenge lies in the fact that the human eye may struggle to discern between real and fake content within deepfakes. So, deepfake detection is a big challenge because synthetic forgeries are deceiving the naked eye. Many researchers have presented different approaches to detecting deepfakes during the last few years. These methods can be categorized into several technologies like feature-based [8], artifacts [9], inconsistencies of physiological signals [10], [21] and deep convolutional approaches [11], [12], [13]. These existing deepfake detection methods have achieved success but come with significant limitations. The key limitation in existing deepfake detection research is the insufficient examination of cross-dataset generalization. Many prior studies report performance primarily under intra-dataset train-test splits, which may lead to overly optimistic estimates of real-world effectiveness. In practical scenarios, however, detection must handle previously unseen manipulation techniques, varying compression levels, and diverse data distributions. The existing detection methods have failed to respond to the new technologies with the rapid evolution of deepfake creation, leaving a gap where deepfakes outpace detection techniques. Additionally, there are no flawless detection methods, particularly in Convolutional Neural Network (CNN)-based approaches. To avoid this situation, we moved to a new way rather than traditional methods.

The increasing realism of deepfakes poses significant challenges to digital security and misinformation control. Traditional CNN-based detection methods often struggle to capture subtle structural inconsistencies in manipulated content. Motivated by the need for more robust detection methods, this study explores the potential of Capsule Networks, which are better suited to model spatial and hierarchical relationships, to improve deepfake detection performance. This experimental protocol enables a rigorous analysis of robustness to domain shift.

The main contribution of this study is the development and comparative evaluation of multiple Capsule Networks for deepfake video detection, demonstrating that Capsule Networks are more effective than traditional CNN-based models. We conduct a comparative analysis of the performance of the proposed Capsule Network models against state-of-the-art Capsule models, and traditional CNN-based models across various datasets utilizing AUC% and the number of parameters as the criteria for comparison. So, to achieve this target, we utilized multiple Capsule Network models for the experiment,

altering several parameters, changing size, complexity, and configuration, etc. For the comparison, we selected five existing CNN methods. They were MesoNet [15], Head Pose [17], Visual Artifact [8], XceptionNet [16], and Multitask-learning [19] architectures, which were fine-tuned on the 140K Real and Fake Faces Kaggle dataset [20]. These models have been implemented on several technologies. We used five different datasets: UADFV [17], DF-TIMIT [21], FaceForensics++ [22], DFDC [23][24], and Celeb-DF [25]. To address the generalization problem, we conduct systematic cross-dataset evaluations by training the proposed models on a single dataset and assessing the performance across multiple unseen datasets.

Our study is structured as follows: in the first part, we present a brief introduction about deepfakes, along with their background, problem, motivation, and contribution, and outline the structure of our study. The second part represents a literature summary on deepfake detection methods and Capsule Networks. Next, we detail our methodology, including our proposed Capsule models and datasets used. In section four, we explore our implementations and detail the results obtained throughout the experiment. Lastly, we summarize the conclusions drawn.

II. Related Work

Under the related works, we cover the existing best deepfake detection methods and explore the improvements and unique advantages of the Capsule Network models on deepfake detection.

A. Deepfake Detection Methods

Recently, deepfake is a more concerned phenomena due to the rapid spread of various deepfake creations [2]. Deepfake generation and detection are developing in parallel. From 2018 onwards, there has been increased socialization of forgery arts, applications, fake video creations, and deepfake generation tools. At the same time, financial frauds, revenge on politicians and popular people, pornographic videos of celebrities and blackmails, etc. started to increase gradually [2]. With the growth of misleading information, attention has been drawn to finding deepfake detection methods. Recently, a considerable number of research publications regarding deepfake detection have been available.

Numerous researchers have proposed various approaches for deepfake detection, including feature-based [8], artifact analysis [9], inconsistencies in physiological signals [10], [21], and deep convolutional methods [11], [12], [13]. Deep convolutional approaches may be quite reliable because they do not depend on feature limits. XceptionNet architecture [16] has been successfully used in many approaches [22], [26], [27]. In 2020, Kumar et al. [28] presented a successful model with an accuracy of 99.2% on the CelebDF dataset using XceptionNet architecture as a feature extractor. Several papers have presented deepfake video detection methods using Recurrent neural networks [30], [32] or LSTMs [31], [33]. Lima et al. [29] combined several 2D and 3D convolutional layers in their model to leverage the temporal dimension. Very recently, Google has presented a watermark-based detection tool called SynthID [34] to identify AI-created images. But our opinion is that the Watermark technique is not very successful because the sources of image creation are varied. This method is not valid for widespread usage because it is restricted only to limited resources like Google Cloud and some specific customers, such as Vertex AI [34].

Deep learning methods, machine learning methods, computer vision methods, multimodal and scalable techniques can be used for deepfake detection. The problem lies in deepfake creation, which continuously evolves with new technologies and advances day-by-day. Therefore, the state-of-the-art deepfake detection methods may fail in the face of new technical advances. Recently, a few researchers in the domain of deepfakes have explored the use of Capsule Networks.

B. Capsule Network Models and Their Advantages

According to the literature, Capsule Networks are not a novel concept in the field of deepfake detection. Hinton et al. [35] first introduced the concept of the Capsule Networks in 2011. Then again, Hinton et al. [36] introduced the training and propagation of capsules using the routing by agreement algorithm in 2017. Capsule Networks are composed of vectorial units that encode information in multiple dimensions, including pose information, unlike scalar units in Convolutional Neural Networks. Fig. 1 presents the general architecture of the Capsule Network employed in this work. The model processes the input through a primary capsule layer that encodes local features as vector representations. These capsules are then aggregated in a higher-level capsule layer using a routing mechanism to model hierarchical relationships. The resulting capsule outputs are used in the decision stage, and the final predictions are obtained at the output layer.

Through experiments, it has been shown that Capsule Networks could improve upon the limitations of traditional CNN methods in detecting deepfakes. Hinton et al. [35] have found some drawbacks in the use of a pooling layer in a CNN architecture. Mainly, the authors [35] mentioned that information like the position of an object is lost in the pooling layer. In the CNN model, neural networks strongly consider key elements like the eyes of a face rather than its position because the pooling layer deletes other information like position. But position is very important to increase the accuracy of the method. On the other hand, vectors help to collect more information about the object, like position, color, texture, and skewness, etc. and help to get more accurate predictions. In 2018, Shahroudnejad et al. [37] identified an additional drawback: the need for more training data when employing CNN architectures with a pooling layer. Also, the authors in [37] described that CNNs require longer duration training data and Capsule Networks need a wider range of training data. Recent research has also explored transformer-based and hybrid architectures for deepfake detection. Transformer backbones such as Swin Transformer have been proposed to capture global context and spatial consistency, improving robustness across datasets in deepfake video classification [44]. Furthermore, hybrid models integrating CNNs with vision transformers have been developed to leverage local feature extraction and global attention mechanisms for enhanced detection efficacy [45]. However, these approaches often require large-scale training data and substantial computational resources, which can limit their practical development.

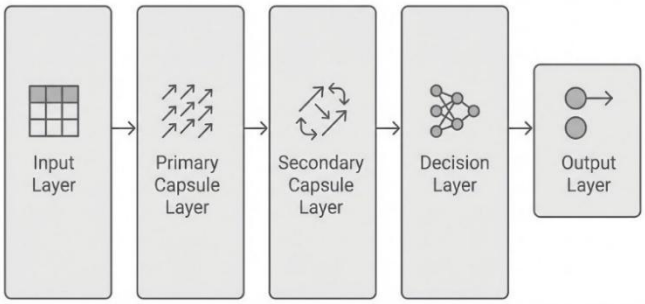

Fig. 1. Common architecture of the capsule network.

Jindong et al. [38] have presented the advantages and disadvantages of the Capsule Network architecture. Advantages are the ability to generalize, a small number of parameters needed, various transforms applied to the image and resistance to attacks. According to the authors [38], the disadvantages are that the capsules have limited learning capabilities and sometimes Capsule Networks show poorer assessment than traditional CNNs on large datasets like ImageNet [39].

Recently, Nguyen et al. [40] have improved a Capsule-Forensics Network [40], [41] for deepfake video detection. This method has shown higher performance compared to the state-of-the-art methods while utilizing less training data, a smaller number of parameters and lower computational cost, etc. [40], [41]. These superior advancements across multiple domains have motivated them to turn towards the Capsule Networks.

C. Improvements of the Capsule-Forensics Architecture

In 2019, Nguyen et al. [40] improved the Capsule-Forensics architecture for deepfake video detection. In the following paragraph, we briefly described the configuration and shortcomings of the Capsule-Forensics architecture.

Capsule-Forensics architecture has been configured along with a part of the pre-trained VGG-19 model [14], which has been used to extract the features and has been transformed into primary capsules. The authors [40] have included a 'stat pooling layer' to calculate the mean and standard deviation for filters. The Capsule-Forensics architecture has utilized from 3 to 10 primary capsules, each with 8 features, to extract relevant features. Next, the architecture was configured along with a dynamic routing algorithm. Here, the dynamic routing algorithm performs two iterations, calculating the sum of output vectors from the primary capsules. The authors [40] have used two output capsules (for deepfake and real with 4 x 1 vectors). The resulting vector has passed through the SoftMax layer. The authors [40] used an average function to predict the probability values of deepfakes. Capsule-Forensics architecture has used 3.9 million parameters for the whole network. Nguyen et al. [40] showed 93.11% accuracy on binary classification with FaceForencis++ dataset. Tolosana et al. [42] utilized the same architecture and demonstrated a 99.52% AUC on the FaceForensics++ dataset and 82.46% AUC on the Celeb-DF dataset. This architecture has exhibited superior performance by utilizing fewer parameters, a smaller amount of training data and lower computational costs compared to state-of-the-art methods. Hence, many research domains have started incorporating Capsule Networks, particularly within the field of deepfakes.

But we identified that the Capsule-Forensics architecture [40], [41] has several shortcomings. The authors [40], [41] and [42] utilized a small number of primary capsules, typically ranging from 3 to 10. It appears that their configuration might be considered weak since primary capsules hold significant importance in computing the values of secondary capsules. Additionally, the authors of [40] and [42] employed only 2 iterations for the dynamic routing algorithm. A higher number of iterations can significantly impact the accuracy of the final prediction. The Capsule-Forensics architecture also incorporated the 'statistical pooling layer.' According to the principles of Capsule Networks, the pooling layer's function involves discarding a considerable amount of valuable information, potentially leading to a decrease in the performance of the capsules. We believe that this architecture has the potential to achieve higher performance by addressing these existing shortcomings and implementing necessary improvements.

III. Methodology

In the research carried out in this study, our primary objective was to implement multiple Capsule Network architectures for detecting deepfake videos, addressing the existing shortcomings of the Capsule-Forensics architecture. We aimed to compare their performances and evaluate the effectiveness of modifications introduced to address the earlier discussed shortcomings. To achieve this, we centered our attention on the Capsule-Forensics architecture introduced in 2019. While retaining the fundamental elements of the Capsule-Forensics network, we made essential modifications necessary to implement diverse Capsule Network architectures for deepfake detection. We altered the size, complexity, and configuration of each Capsule architecture from its original specifications. Furthermore, we conducted a comparative analysis of the performance of Capsule Network models and CNN models for deepfake detection. Our objective was to demonstrate that Capsule Networks possess greater efficacy in detecting deepfakes compared to traditional CNN methods.

We configured four different Capsule Network models for the analysis, elaborated upon in detail in the following section. During experimentation, the DFDC dataset was utilized for training the Capsule models. To compare the performance, we employed five state-of-the-art CNN models: specifically, the MesoNet [15], Head Pose [18], Visual Artifact [17], XceptionNet [16], and Multitask-learning [19] architectures. All these models are feature-based deepfake video detection methods. They were initially pre-trained on the ImageNet dataset [39], and their output layers were subsequently replaced with fully connected decision layers for the purposes of this study. Initially, we fine-tuned and retrained these models using the 140K Real and Fake Faces Kaggle dataset during the training phase. To conduct a comparative analysis, we assessed the performance of the mentioned CNN models across five datasets: UADFV, DF-TIMIT, FaceForensics++, DFDC, and Celeb-DF dataset. Subsequently, we analyzed the results and compared the performance of each Capsule model with that of the state-of-the-art CNN models in detecting deepfake videos. The Area Under Curve (AUC) values are presented in the results and discussion section for comparison analysis.

A. Proposed Capsule Network Architecture

Our primary objective was to demonstrate that Capsule Networks possess greater efficacy in detecting deepfakes compared to traditional CNN methods. For this task, we need to implement a better Capsule Network architecture and to show better performance instead of the state-of-the-art CNN methods. We tried to rectify the few shortcomings present in the state-of-the-art Capsule models for detecting deepfake videos. We started with the state-of-the-art Capsule-Forensic architecture [40], enhancing identified shortcomings, and subsequently assessed various Capsule Network architectures listed below to enhance their performance. While keeping the fundamental elements of the Capsule-Forensics architecture utilized in [40] as the basis, we made some necessary modifications to our Capsule Network architectures described as follows.

We modified their size, complexity, and configuration as necessary. We require an existing pretrained model to extract the necessary features for the primary capsules in our Capsule models. We chose the VGG-19 CNN architecture [14] as a pretrained model. In the model configuration, the larger models were set up with the initial 8 convolutional layers and totaled around 1.6 million parameters. The smaller models were composed of the initial 3 convolutional layers, resulting in a total of around 0.4 million parameters. The authors of [40] and [42] utilized a limited number of primary capsules, typically ranging from 3 to 10. Their configuration appears weak because primary capsules play a crucial role in computing output values for the secondary capsules. Secondary capsules are of great importance to predict the correct values. We assumed that a large number of primary capsules may eliminate noise and prevent poor predictions. Thus, we conducted experiments with varying numbers of primary capsules, starting from 1,800 primary capsules.

Initially, we used two secondary capsules for each class. Later, we conducted tests with different quantities of secondary capsules for each class. In most instances, there were 16 capsules of 16x1 dimensions used. The fully connected decision layer was used to distinguish real and deepfakes. The number of iterations in the dynamic routing algorithm plays a role in determining the weights assigned to the secondary capsules. The better weights of the secondary capsules ensure better accuracy of the predictions. The authors of [40] and [42] have used only 2 iterations in their research. Instead of using only 2 iterations, we implemented a broader range of iterations in the dynamic routing algorithm, employing a minimum of 3 and a maximum of 5 iterations. Finally, we obtained a comparative analysis of each architecture's performance in comparison to the state-of-the-art methods. Below is the list of all the implemented Capsule Network models used in our experiment. Fig. 2 shows a graphical representation of the architectural configuration of each model.

- Capsule Network model 1: This model was configured with 1,800 primary capsules, each consisting of 8 dimensions, and 2 secondary capsules, each comprising 16 dimensions.
- Capsule Network model 2: This model was designed with 7,200 primary capsules, each having a length of 8, and 10 secondary capsules, each with a length of 16. Additionally, a fully connected layer was incorporated as the decision maker.
- Capsule Network model 3: This specific Capsule Network model comprised 1,568 primary capsules, each having a length of 8, and 16 secondary capsules, each with a length of 16. During the training process, the dynamic routing algorithm utilized 3 and 5 iterations. Additionally, a fully connected layer was included for decision-making purposes.
- Capsule Network model 4: This model shares similarities with Capsule Network model 3, except for model 4, which includes 8 secondary capsules instead of 16.

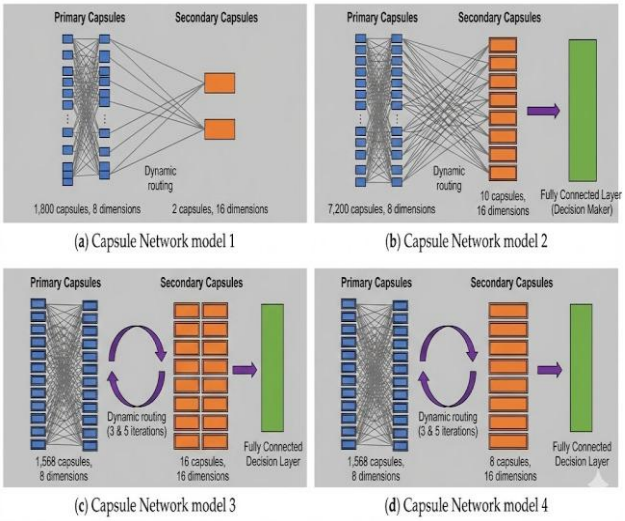

Fig. 2. Proposed capsule network architecture.

B. Dataset Used

When selecting a dataset for deepfake detection, several crucial characteristics should be considered, including short video length (10-15 seconds), focus on close-up human faces, confirmation of consent from actors involved, the total number of videos available, and the overall quality of the video content. These factors are most affected in determining the accuracy and reliability of a deepfake detection method. Currently, many open-source free deepfake datasets are available. We chose five distinct datasets that were very suitable for our experiment: UADFV, DF-TIMIT, FaceForensics++, DFDC, and Celeb-DF. The dataset contained both real and deepfake video clips, each lasting around 5 to 10 seconds in duration. We need to break them down into individual frames for further analysis and processing. Each frame is fed into the detection architecture. To create frames, we selected multiple video clips from both real and deepfake sources found within the DFDC, UADFV, DF-TIMIT, FaceForensics++, and Celeb-DF datasets.

When selecting video clips, we specifically targeted those that provided better focus on the face region. This consideration was crucial since deepfakes primarily manipulate and alter facial features. We employed "Windows Movie Maker" to split the video clip into individual frames. The first 256 frames of each video clip were extracted and resized to 299x299. Thus, we aimed to exclusively extract the facial regions from each frame. For our experiment, after splitting the video, we adopted an

appropriate preprocessing pipeline as follows to extract the necessary features from each frame. Initially, we performed face detection, followed by detecting facial landmarks, eliminating the background, and ultimately resizing the image as part of our preprocessing pipeline. These steps were repeated for all 256 frames within each of the selected video clips. Here, we utilized DLib-ml [43] for detecting the face region and identifying facial landmarks. DLib-ml is a machine learning based toolkit designed for face detection, developed by King in 2009 [43]. We applied DLib-ml for every frame, extracting 68 landmarks from each frame. The resulting landmarks were then mapped onto a standard 3D facial landmark model consisting of 68 points.

We used two datasets for the training process. The 140K Real and Fake Faces Kaggle dataset was used for the CNN models, while the Deepfake Detection Challenge (DFDC) dataset served as the dataset for the Capsule Network models. The DFDC dataset was chosen for training the Capsule Network models because its video-based deepfake samples capture complex spatial relationships, pose variations, and manipulation artifacts that align with the Capsule Network's ability to model hierarchical part–whole representations; therefore, DFDC enables the Capsule Network to exploit its architectural advantages more fully, while the 140K Real and Fake Faces Kaggle dataset remains more suitable for CNNs that perform effectively on static image classification tasks. Below are the six distinct datasets we selected for our research:

- UADFV dataset: This dataset includes 49 original videos collected from YouTube and 49 fake videos generated using deep neural networks.
- DF-TIMIT dataset: This dataset includes 640 real and fake videos. Deepfake videos were generated using FaceSwap-GAN. This dataset is divided into the DF-TIMIT-LQ and DF-TIMIT-HQ datasets.
- FaceForensics++ (FF++) dataset: This dataset consists of 1,000 videos from both real and fake sources. Real videos were collected from YouTube, and Deepfake videos were generated using FaceSwap.
- Deepfake Detection Challenge (DFDC) dataset: This dataset encompasses 4,113 fake videos, which were generated using 1,131 real videos. DFDC used 66 individuals of diverse age, gender, and culture.
- Celeb-DF dataset: This dataset contains 590 real videos and 5,639 fake videos, which were generated using FaceSwap technology.
- 140K Real and Fake Faces Kaggle dataset: This dataset consists of 140,000 human face images, including 70,000 real faces and 70,000 fabricated faces, which were generated using StyleGAN. This dataset is segmented into three subsets: 100,000 images for training, 20,000 for testing, and 20,000 for validation purposes.

C. Model Training

Our primary purpose was to evaluate the effectiveness of both Capsule Network models and modified versions thereof in comparison to state-of-the-art CNN methods regarding their performance. We conducted repeated training and testing of the models with the intention of achieving higher accuracy.

In the training process, we utilized the DFDC dataset for the Capsule models. We selected 10^{-3} as the learning rate value, maintaining a learning rate decay factor within the range 0.5 and 0.8. For the model training, we opted for 15 epochs, utilized the Adam optimizer, set a batch size of 16, and employed 3 and 5 iterations for the dynamic routing algorithm as the required parameters.

We selected five established state-of-the-art CNN models, which were previously pretrained on the ImageNet [39] dataset. During our experiment, we fine-tuned these models in accordance with our specific requirement of classifying real and fake videos. Here, we trained the CNN models on the 140K Real and Fake Faces Kaggle dataset, with a fixed learning rate of $5x10^{-4}$, a learning rate decay factor set to 0.92, running for 10 epochs and utilizing the Adam optimizer with a batch size of 64. Also, we selected Binary Cross-Entropy as the loss function.

In scripting the programs, we utilized Jupyter notebooks with Python. Google Collaboratory was instrumental in executing all our notebooks. In managing the extensive dataset, we opted to partition it into three distinct categories to facilitate training, testing and validation. Specifically, we allocated 100,000 images for the training set and reserved 20,000 images each for the testing and validation sets. These images were drawn from the 140K Real and Fake Faces Kaggle dataset and the DFDC dataset. Saved models were evaluated with five distinct datasets and compared their performance using the Area Under the Curve (AUC%) values.

IV. Results and Discussion

We used Windows Movie Maker to split the video clip into the first 256 frames. Each frame was processed using the "DLib-ml" tool to detect the face region and facial landmarks, marking 68 landmarks of each frame. The resultant landmarks were subsequently mapped onto a standard 3D facial landmark model consisting of 68 points. Fig. 3 shows the detected facial landmarks identified by the "DLib-ml" tool. In Fig. 3, the original real image is shown in Image (A). Image (B) displays the facial landmarks (indicated by red dots) of the central face region of the real face, while Image (C) displays the facial landmarks (indicated by blue dots) of the entire face region of the real face. All frames have been resized to 299x299. The first 256 frames of every video clip underwent this process.

Image (A) – Original Image (B) – Red dots Image (C) – Blue Dots

Fig. 3. Facial landmarks detection.

A. Performance of the Different Deepfake Video Detection Methods

In this research, our objective is to conduct a comparative analysis of the performance of the proposed Capsule Network models against state-of-the-art Capsule models and CNN models across various datasets. We aim to utilize AUC% and the number of parameters as the criteria for comparison.

Firstly, we determined the superior performance of the Capsule Networks over traditional CNN models. Secondly, we evaluated whether our Capsule Network models could achieve comparable or superior performance compared to state-of-the-art Capsule Network (CapsNet) models [41], [42] and CNN models. In this experiment, AUC% values were employed for evaluating the performance of each model. Table I showcases the results obtained from the four proposed Capsule Network models alongside the selected state-of-the-art CNN methods, evaluated across five distinct datasets. The proposed Capsule Network model 1, as depicted in Table I, attained the highest AUC% score of 99.55% on the FaceForensics++ dataset, outperforming existing CNN-based approaches and other models. Table I reveals that all detection methods, apart from Capsule Network models, encounter the most significant challenge in detecting deepfakes within the Celeb-DF dataset. Their performance rates with this dataset have remained below 70% to date. However, the proposed Capsule models outperform others, specifically on Celeb-DF. The following Fig. 4 enhances the clarity of the results in Table I.

Fig. 4 shows the grouped bar chart illustrating the performance of proposed Capsule Network models and state-of-the-art CNN models across different datasets. The graph shows the AUC% values of each model on the y-axis and the datasets on the x-axis. Fig. 4 clearly visualizes the higher AUC% values of the proposed Capsule models 1 and 2 across the five different datasets.

The Capsule Network models 3 and 4 were not tested on the UADFV, DF-TIMIT and DFDC datasets. The Capsule Network models 3 and 4 were designed specifically for comparison to the state-of-the-art CapsNet models and other proposed models. Therefore, models 3 and 4 were solely tested on the FaceForensics++ and Celeb-DF datasets, as shown in Table II.

Moreover, the difficulty levels in detecting fakes using the aforementioned methods vary across different datasets, such as DFDC and Celeb-DF. This suggests that the contents of recently developed datasets are still difficult to detect due to the varying collection of deepfake videos. Among the CNN-based models listed in Table I, XceptionNet exhibited the best performance across all datasets, attaining a notable accuracy of 99.0% with the FaceForensic++ dataset. Nonetheless, the proposed Capsule models we tested exhibited impressive AUC% scores, surpassing 99% on the FaceForrensics++ dataset, with some even exceeding 99.5%.

Based on Table I, it's evident that the proposed Capsule Network models achieve competitive performance over the existing CNN models across the datasets evaluated in this study.

Table II shows the AUC% values of the proposed Capsule models and state-of-the-art Capsule models on both the FaceForensics++ and Celeb-DF datasets. We evaluated the performance using AUC% and the number of parameters. According to Table II, it is noticeable that the majority of Capsule Network models, including both our experimental models and state-of-the-art models, have reached an AUC% score exceeding 95%. Capsule Network model 1 has achieved an AUC% score of 99.55%, surpassing the AUC% scores of existing Capsule Network approaches on the FaceForensics++ dataset. Further, except for Capsule Network model 1, the proposed Capsule Network models showed better performance compared to state-of-the-art Capsule Network architectures on the Celeb-DF dataset.

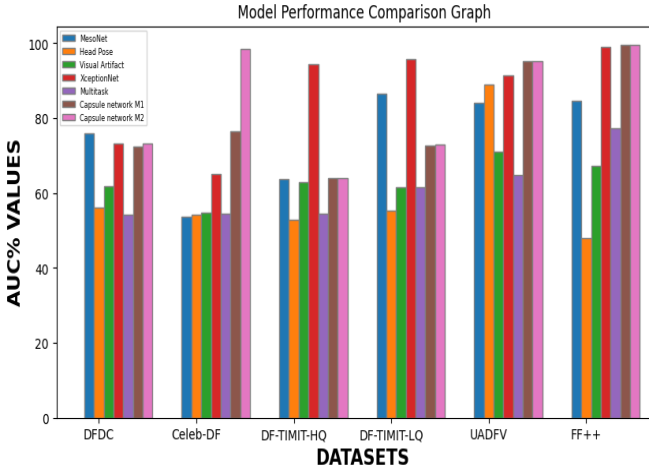

Fig. 4. Comparative performance of proposed capsule network models and CNN models across different datasets.

TABLE I. COMPARATIVE ANALYSIS BETWEEN THE PROPOSED CAPSULE NETWORK MODELS AND STATE-OF-THE-ART CNN-BASED MODELS ACROSS MULTIPLE DATASETS FOR PERFORMANCE EVALUATION

Model	Dataset					
	DFDC	*Celeb-DF*	*DF-TIMIT*		*UADFV*	*FF++*
			HQ	*LQ*		
MesoNet [15]	76.0	53.7	63.8	86.5	84.2	84.5
Head Pose [18]	56.0	54.3	52.8	55.3	89.0	47.9
Visual Artifact [17]	61.8	54.9	63.0	61.5	71.0	67.2
XceptionNet [16]	73.2	65.1	94.3	95.8	91.3	99.0
Multitask [19]	54.1	54.5	54.6	61.6	64.9	77.2
Capsule Network model 1 (Our model)	72.5	76.43	63.9	72.8	95.1	99.55
Capsule Network model 2 (Our model)	73.1	98.4	64.0	72.9	95.2	99.51

TABLE II. COMPARATIVE ANALYSIS BETWEEN THE PROPOSED CAPSULE NETWORK MODELS AND THE STATE-OF-THE-ART CAPSULE NETWORK MODELS ON THE FACEFORENSICS++ AND CELEB-DF DATASETS

Model	AUC %		No. of parameters
	FaceForensics++	Celeb-DF	
Nguyen et al- CapsNet [41]	93.11	57.50	3.9 M
Tolosana et al- CapsuleNet [42]	99.52	82.46	3.9 M
Capsule network model 1	99.55	76.43	5 M
Capsule network model 2	99.51	98.40	16 M
Capsule network model 3: 3 iterations	98.40	97.76	6 M
Capsule network model 3: 5 iterations	99.00	99.00	6 M
Capsule network model 4	98.31	97.46	4 M

Tolosna et al. [42] introduced an architecture that attained 82.46% AUC% on the Celeb-DF dataset and 99.52% on the FaceForensics++ dataset. In contrast, our proposed Capsule model 3 demonstrated the best performance of 99% on the Celeb-DF dataset and 99% on the FaceForensics++ dataset.

Tolosna et al. [42] introduced an architecture that attained 82.46% AUC% on the Celeb-DF dataset and 99.52% on the FaceForensics++ dataset. In contrast, our proposed Capsule model 3 demonstrated the best performance of 99% on the Celeb-DF dataset and 99.00% on the FaceForensics++ dataset.

Fig. 5 illustrates the grouped bar chart showing the AUC% values of proposed Capsule Network models and state-of-the-art Capsule Network models on the FaceForensics++ and Celeb-DF datasets. The graph shows the AUC% values of each model on the y-axis and the datasets on the x-axis.

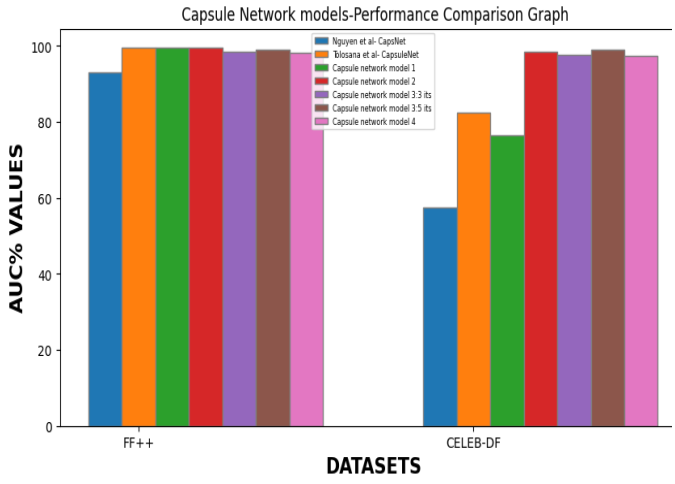

Fig. 5. Comparative performance of proposed capsule network models and state-of-the-art capsule network models on the faceforensics++ and celeb-DF datasets.

Fig. 5 shows a clear comparison of the performance of each model on each dataset. According to Fig. 5, the AUC% values of our proposed Capsule models display higher values than state-of-the-art Capsule models on both the FaceForensics++ and Celeb-DF datasets.

Additionally, we experimented with several iterations of the dynamic routing algorithm. Increasing the number of iterations in the Capsule Network can lead to higher performance. For instance, in Capsule Network model 3, transitioning from 3 iterations to 5 iterations within the same architecture enhanced the performance from 98.40 to 99.00 on the faceForensics++ dataset and from 97.76% to 99.00% on the Celeb-DF dataset, respectively. Nevertheless, a higher number of iterations also leads to increased prediction time, rendering it unsuitable for real-time tasks.

In addition, we conducted an experiment that demonstrates the superiority of Capsule Networks over Convolutional Networks. For instance, Capsule Network models 3 and 4 utilized 6 million and 4 million parameters, respectively, whereas the CNN models employed over 10 million parameters. Despite processing fewer parameters, the Capsule Network models achieved results with over 97% AUC%, surpassing the performance of the CNN models. Through experimentation, we found that maintaining over 1.9 million parameters in each Capsule Network model allowed them to achieve superior performance compared to CNN models.

The following Fig. 6 shows the line chart comparing the number of parameters used for the Capsule models and CNN models. The graph will illustrate the model complexity and resource requirements between the two types of models. Capsule Networks achieve consistently high AUC scores across all parameter sizes, demonstrating strong performance even with fewer parameters. In contrast, CNN models show a gradual improvement in AUC as the number of parameters increases, indicating a higher dependence on model size. Notably, CNNs require substantially more parameters to approach the performance level of Capsule Networks. These results highlight the superior parameter efficiency and robustness of Capsule Networks compared to conventional CNN architectures.

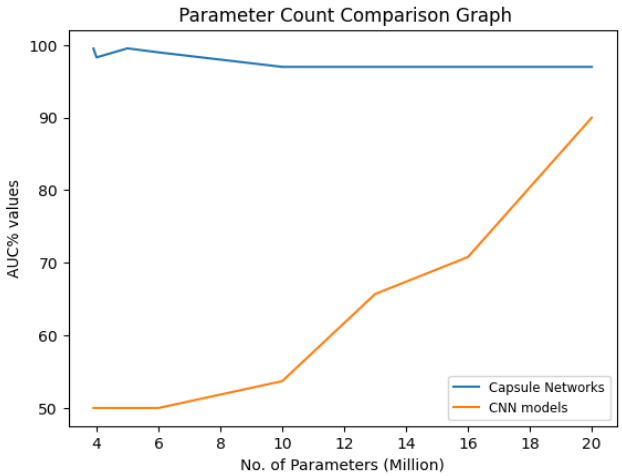

Fig. 6. Comparison of the number of parameters used for both capsule network models and CNN models.

The reported results are presented as single-run AUC values obtained under a fixed and consistent experimental protocol. Although additional statistical validation, such as repeated trials or confidence interval estimation, would further strengthen the robustness assessment, the observed performance trends remain

stable across five cross-dataset evaluations. All models were tested on the same five unseen datasets to assess generalization; Capsule Networks were trained on the DFDC dataset, while CNN models were trained on the 140K Real and Fake Faces Kaggle dataset due to experimental design considerations. Consequently, the observed performance improvements reflect empirical results under the current experimental setup rather than a strict isolation of architectural superiority or generalization capability. Future work will include controlled experiments and extended statistical analysis to more rigorously evaluate architectural generalization and performance stability.

We encountered several challenges during the model training process, particularly in terms of the amount of time it takes to train the models. Achieving higher accuracy requires more than 100,000 data points for training, which may be time-consuming. The time required for successful training can vary depending on factors such as hardware resources, computational cost, size of the dataset, number of training epochs, and type of neural network architecture of the detection model. In our experiment, we used Google Collab online for training and for the evaluation process. We used Windows Movie Maker for the frame extraction. While this tool is not conventional for research, the frame extraction procedure was consistent across all videos, ensuring that the experimental results are reproducible. For future studies, the same extraction procedure can be implemented using standard tools such as OpenCV, which would provide fully reproducible frame sequences.

V. Conclusion

Deepfake-generated fake images, voices, and videos are rapidly spreading across various social media platforms and websites. The deepfake creation tools are highly recommended for their output accuracy, as the synthetic forgeries they produce can appear indistinguishable to the naked eye. Detecting deepfakes poses a significant challenge due to the high accuracy of these fabricated creations.

Through this study, we aimed to present a novel and robust approach to deepfake detection distinct from existing state-of-the-art methods. We identified that Capsule Networks have more effective power in detecting deepfakes instead of traditional CNN-based methods. According to our results, despite having fewer parameters, the Capsule Network models achieved results with over 98% AUC%, showing stronger performance. Our proposed Capsule Networks exhibited reduced performance degradation compared to traditional CNNs and existing Capsule (CapsNet) models, indicating improved generalization capability. We emphasize the following points and present the summarization of the research as follows.

- For the model training, we employed 15 epochs for Capsule models and 10 epochs for CNN models. However, the proposed Capsule Network models outperformed the CNN models, completing the training process in a shorter period.
- Capsule Networks can retain more information compared to traditional CNNs, even with less training data.
- Capsule Networks demonstrated stronger performance under the given experimental configuration.
- As the number of iterations of the dynamic routing algorithm increases, the performance of the Capsule models also improves, even with a smaller number of parameters.
- Existing CNN-based deepfake detection methods have shown lower accuracy, often falling below approximately 65%, primarily due to the generalizability problem. Capsule Networks, however, offer a solution that avoids the generalizability problem.

Our results show that Capsule Networks exhibit greater efficacy in detecting deepfakes compared to traditional CNN-based methods. Therefore, Capsule Networks are better suited for deepfake detection due to their capacity for generalization, utilization of fewer parameters, and the ability to handle image rotation, transformation, and dimensionality. These qualities make Capsule Networks an ideal choice for deepfakes detection. Novel methods within Capsule Networks for deepfake detection should be further improved in the near future. These findings will be a significant step forward for the new researchers in the deepfake domain.

References

[1] S. Haysom, "People are using face-swapping tech to add Nicolas Cage to random movies and what is 2018.", January 2018. [Online]. Available: https://mashable.com/2023/09/15/nicolas-cage-face-swapping deepfakes/?europe=true.

[2] M. C. Weerawardana and T. G. I. Fernando, "Deepfakes detection methods: a literature survey," *2021 10th International Conference on Information and Automation for Sustainability (ICIAfS)*, 2021, pp. 76-81, doi: 10.1109/ICIAfS52090.2021.9606067.

[3] B. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, Courville, A. Bengio, "Generative Adversarial Networks", 2014.

[4] Shaoanlu, "Faceswap-GAN," 2018. [Online]. Available: https://github.com/shaoanlu/faceswap-GAN.

[5] L. Guilloux, "Fakeapp," March 2019. [Online]. Available: https://www.fakeapp.org/

[6] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Niessner, "Face2face: Real-Time face capture and reenactment of RGB videos," *In CVPR*, pp. 2387–2395, June 2016.

[7] ModelIncubator, "Deepfake project - nonofficial project based on original deepfakes thread," 2018. [Online]. Available: http://www. github.com/deepfakes/faceswap

[8] F. Matern, C. Riess, and M. Stamminger, "Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations," *in 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW),* pp. 83-92, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/ 8638330/

[9] X. Zhang, S. Karaman, and S. Chang, "Detecting and simulating artifacts in GAN fake images," *preprint arXiv: 1907.06515*, 2019.

[10] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, and S. K. Nayar, "Face swapping: automatically replacing faces in photographs," *ACM Trans. Graph,* vol. 27, pp. 1–8, August 2008.

[11] D. Cozzolino and L. Verdoliva, "Noiseprint: a cnn-based camera model fingerprint," *arXiv preprint arXiv: 1808.08396*, 2018.

[12] K. L. Du and M. N. S. Swamy, "Neural networks and statistical learning," *Springer London*, London, 2019.

[13] T. D. Nhu, I. Na, and S. H. Kim, "Forensics face detection from gans using convolutional neural network," 2018.

[14] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "Deepfakes and beyond: a survey of face manipulation and fake detection," *Information Fusion*, vol. 64, pp. 131–148, 2020.

[15] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: a compact facial video forgery detection network," *in 2018 IEEE International Workshop on Information Forensics and Security (WIFS). IEEE,* 2018, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/8630761/

[16] F. Chollet, "Xception: deep learning with depthwise separable convolutions," *in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* Honolulu, HI, USA, 2017 pp. 1800-1807.

[17] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent Head Poses," *in ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* vol. 2019-May, IEEE, 2019, pp. 8261–8265. [Online]. Available: https://ieeexplore.ieee.org/document/8683164/

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, July 2017, pp. 4700-4708.

[19] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task learning for detecting and segmenting manipulated facial images and videos," 2019. [Online]. Available: http://arxiv.org/abs/1906.06876

[20] "Real and fake detection Kaggle dataset." [Online]. Available: https://www.kaggle.com/xhlulu/140k-real-and-fake-face

[21] P. Korshunov and S. Marcel, "DeepFakes: a new threat to face recognition? assessment and detection," pp. 1–5, 2018. [Online]. Available: http://arxiv.org/abs/1812.08685

[22] B. A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and ¨ M. Niebner, "FaceForensics++: learning to detect manipulated facial images," *in 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019, pp. 1-11. [Online]. Available: http://arxiv.org/abs/1901.08971

[23] Kaggle, "Join the deepfake detection challenge (DFDC)," December 2019. [Online]. Available: https://deepfakedetectionchallenge.ai/

[24] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The DeepFake detection challenge dataset," 2020. [Online]. Available: http://arxiv.org/abs/2006.07397

[25] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: a large-scale challenging dataset for DeepFake forensics," *in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020 pp. 3204-3213. [Online]. Available: http://arxiv.org/abs/1909.12962

[26] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. Jain, "On the detection of digital face manipulation," *in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[27] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The DeepFake detection challenge (DFDC) dataset", 2020.

[28] BA. Kumar and A. Bhavsar, "Detecting deepfakes with metric learning," arXiv preprint arXiv:2003.08645, 2020.

[29] O. de Lima, S. Franklin, S. Basu, B. Karwoski, and A. George, "Deepfake detection using spatiotemporal convolutional networks".

[30] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent convolutional strategies for face manipulation detection in videos," *in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops,* 2019.

[31] S. Tariq, S. Lee, Simon S. Woo, "A convolutional LSTM based residual network for deepfake video detection", 2020.

[32] D. Guera and E. Delp, "Deepfake video detection using recurrent neural networks," *in 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Auckland, New Zealand, 2018, pp. 1-6.

[33] S. Dan-Cristian, B. Ionescu, "Deepfake video detection with facial features and long-short term memory deep networks"

[34] "Watermark tool to identify AI created images." [Online]. Available: https://www.geeksforgeeks.org/google-launches-watermark-tool-to-identify-ai-created-images/

[35] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," *In International Conference on Artificial Neural Networks*, Springer, 2011, pp. 44-51.

[36] E. Hinton, S. S. Nicholas, and F. Geoffrey, "Dynamic routing between capsules," *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 2017.

[37] B. A. Shahroudnejad, P. Afshar, K. N. Plataniotis and A. Mohammadi, "Improved explainability of capsule networks: relevance path by agreement," *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Anaheim, CA, USA, 2018, pp. 549-553, doi: 10.1109/GlobalSIP.2018.8646474.

[38] V. Tresp, J. Gu1, B. Wu, "Effective and efficient vote attack on capsule networks," *International Conference on Learning Representations (ICLR),* 2021.

[39] J. Deng, W. Dong, R. Socher, L. Li, K. Li and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009, pp.248-255.

[40] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Use of a capsule network to detect fake images and videos," October, 2019

[41] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-Forensics networks for deepfake detection," *Advances in Computer Vision and Pattern Recognition,* Springer, Cham, 2022.

[42] R. Tolosana, S. Romero-Tapiador, J. Fierrez, and R. Vera-Rodriguez, "DeepFakes evolution: analysis of facial regions and fake detection performance," 2020.

[43] D. E. King, "Dlib-ml: a machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755-1758, December 2009, doi/10.5555/1577069.1755843.

[44] L. Y. Gong, X. J. Li, and P. H. J. Chong, "Swin-Fake: A consistency learning transformer-based deepfake video detector," *Electronics*, 13, vol. 15, pp. 3045, 2024.

[45] A. H. Soudy, O. Sayed, H. Tag-Elser, "Deepfake detection using convolutional transformer and convolutional neural networks," *Journal of Neural Computing and Applications*, vol. 36, pp. 19759-19775, 2024.