UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

ENTROPY INTEGRATED DYNAMIC ROUTING IN CAPSULE NETWORKS AND DEEPFAKE DETECTION APPLICATIONS

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

REZA BABAEI
Norman, Oklahoma
2024

ENTROPY INTEGRATED DYNAMIC ROUTING IN CAPSULE NETWORKS AND DEEPFAKE DETECTION APPLICATIONS

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Samuel Cheng, Chair

Dr. Shangqing Zhao

Dr. Choon Yik Tang

Acknowledgements

I would like to express my deepest gratitude to all those who have supported me throughout my journey.

Table of Contents

List of Tables

List of Figures

Abstract

Capsule networks are recognized for their ability to capture part-whole hierarchies in visual data, but they face challenges in managing uncertainty within part-object relationships. This study introduces an entropy-integrated dynamic routing algorithm to enhance both the performance and interpretability of capsule networks by embedding information-theoretic principles. Our method incorporates an entropy-based regularization term in the final iteration of the routing process, improving routing decisions and reducing reliance on uncertain capsule connections. Evaluations on CIFAR-10 demonstrate a mean accuracy of 85.58%, surpassing the baseline accuracy of 84.61% achieved with standard dynamic routing. Comparative analysis with recent entropy-based routing approaches highlights our method's balance of computational efficiency and routing flexibility, achieved without additional model complexity. These findings position entropy-integrated dynamic routing as a powerful tool for enhancing the interpretability and effectiveness of capsule networks in high-uncertainty environments.

Our study finds application in deepfake detection by embedding entropy into the capsule networks' dynamic routing process. Traditional capsule networks capture spatial hierarchies critical for identifying synthetic media, yet their scalability is limited by uncertainties in iterative routing. To address this, we put into practice our entropy-based regularization that adjusts routing coefficients based on capsule activation uncertainty, selectively prioritizing reliable connections and enhancing interpretability and model robustness without significant added complexity. Experiments on FaceForensics++ and DeepPhy datasets reveal that our entropy-integrated routing algorithm achieves higher classification accuracy in both binary and multi-class deepfake detection tasks, outperforming baseline models like CapsuleForensics. These results underscore entropy regularization's potential to improve capsule networks' capacity to address nuanced spatial inconsistencies in manipulated media, advancing scalable and interpretable solutions in deepfake detection.

Chapter 1

Introduction

Since the advent of machine learning, efforts have been directed toward replicating human visual processing through artificial neural networks. The rise of deep learning and computational power has enabled the exploration of high-dimensional data, allowing researchers to advance these networks toward human-like understanding. One of the central challenges in the field is understanding how the human brain interprets and classifies objects, inspiring models that emulate elements of human visual cognition with impressive results. A prominent model, the convolutional neural network (CNN), utilizes multiple filtering layers to discern underlying patterns in pixel arrangements, yielding strong performance in many tasks. However, CNNs fall short of fully capturing the human visual system's ability to recognize objects from varying perspectives. Humans inherently understand objects by maintaining viewpoint invariance—a cognitive process of organizing components within a spatial hierarchy that CNNs do not effectively replicate.

This motivation led to Hinton's proposal of capsule networks, a novel architecture designed to capture part-whole relationships and achieve viewpoint invariance. Capsule networks organize object parts into hierarchies, iteratively refining a "guess" about the whole object based on its components. This guess is validated against previously encountered representations of the whole, much like the brain adjusts its perception based on stored knowledge. This iterative refinement process, known as dynamic routing, distinguishes capsule networks and enhances their ability to classify challenging objects. However, like many initial models, this routing process has limitations—specifically, it lacks mechanisms to handle uncertainties arising from noise or ambiguous parts.

Entropy, a concept from information theory, addresses these uncertainties by quantifying disorder within a system. Previous studies have explored entropy in capsule networks, proposing new voting mechanisms or loss functions that promote lower entropy in decision-making. This study advances this approach by integrating entropy directly into the dynamic routing process. Unlike previous methods, our technique refines connections between parts and wholes in real-time, leveraging hidden information without altering the core architecture or increasing complexity. Additionally, this study introduces an entropy-based interpretability metric, enhancing the analysis of prediction confidence in capsule networks. By measuring entropy within the prediction distribution, this metric provides a nuanced assessment of model confidence, particularly valuable in critical applications. Specifically, it examines entropy across both predicted and alternative classes, offering insights into the model's certainty for correct and incorrect classifications alike. This novel integration not only improves accuracy but also enhances interpretability, offering insights into the network's decision-making process beyond conventional performance metrics.

An application of this entropy-enhanced capsule network architecture lies in the detection of deepfakes—a domain of growing importance due to the proliferation of manipulated media. Capsule networks, even in their early stages, have shown competitive performance in deepfake detection, surpassing sophisticated CNNs in identifying manipulated images. By applying our entropy-integrated dynamic routing to capsule forensic tasks, we achieve improved performance in deepfake detection, highlighting the method's potential in real-world applications.

The primary contributions of this thesis summarizes as follows:

- **Entropy-Enhanced Dynamic Routing in Capsule Networks:** This work introduces a novel integration of entropy within the dynamic routing mechanism of capsule networks. This enhancement improves both the performance and interpretability of capsule networks while preserving their intrinsic part-whole representational characteristics.

- **Improved Performance on CIFAR-10:** The proposed entropy-integrated approach achieves a notable improvement in accuracy on the CIFAR-10 dataset, raising it from 84.610% to 85.580%. This increase is achieved with minimal added complexity, maintaining the network's authenticity and avoiding extensive modifications typically required by other methods to achieve comparable results.

- **Application to Deepfake Detection:** The entropy-integrated capsule network demonstrates significant efficacy in deepfake detection, achieving improved classification accuracy on both binary and multi-class deepfake tasks. Specifically, the model yields a 0.72% accuracy increase for binary classification and a 1.05% increase for multi-class classification on the Face-Forensics++ and DeePhy datasets.

This thesis examines this journey, with Chapter 2 detailing capsule networks and the role of entropy, Chapter 3 exploring deepfake applications, and Chapter 4 concluding with future directions for this work.

Chapter 2

Entropy Integrated Dynamic Routing

This chapter explores the integration of entropy into the dynamic routing algorithm of capsule networks, aiming to enhance model interpretability and improve performance, particularly in complex classification tasks. Capsule networks offer a promising alternative to traditional neural networks by capturing spatial hierarchies in data, allowing models to better represent part-whole relationships through the routing-by-agreement mechanism. However, capsule networks face challenges in managing uncertainties within routing decisions, which can limit their effectiveness, especially when data is noisy or partially occluded.

In this study, we propose a novel entropy-integrated approach that refines the routing process by incorporating uncertainty quantification. This approach seeks to mitigate the ambiguity in routing decisions, stabilizing the network and improving its generalization capabilities. By adjusting routing coefficients based on entropy, we enable the network to dynamically balance between interpretability and performance, making informed routing decisions even under uncertain conditions.

The subsequent sections will detail the theoretical foundation of capsule networks and discuss related entropy-integrated methods for dynamic routing. We then introduce our proposed methodology, elaborating on the integration of entropy within the routing process, followed by an analysis of experimental results on the CIFAR-10 dataset. Finally, we compare our approach with other entropy-integrated dynamic routing strategies, discussing its advantages in handling uncertainty and supporting robust model decision-making in complex visual data scenarios.

2.1 Capsule Networks

Capsule definitions may vary depending on their roles within the network and the specific routing mechanisms used to connect them. Broadly, a capsule is defined as a group of artificial neurons that jointly represent a specific feature or entity within the neural network. Neuroscience evidence further supports this approach, suggesting that tightly connected groups of nearby neurons, known as hypercolumns, can represent vector-valued units that transmit coordinated sets of values rather than single scalar quantities [5]. Vector-valued units form the core of both capsule networks and attention mechanisms [6, 5, 7]. In capsule networks, these vector-valued units represent entities with encoded properties, including pose, color, and size. Such a biologically inspired framework offers a more interpretable and flexible representation of visual data and the corresponding part-whole hierarchies than the abstract representations used in traditional deep learning models. Formally, a capsule can be represented as a parameterized function, encapsulating its interpretative and hierarchical capabilities.

As opposed to capsules, standard neural networks (NN) face significant limitations in dynamically representing diverse part-whole hierarchies specific to each image, which hinders their ability to understand visual scenes in the human-like manner [8]. Commonly used for vision tasks, Convolutional neural networks

(CNNs) are inherently equivariant only to translations, meaning they can preserve spatial relationships under translation but not under other transformations such as rotation, scaling, or reflection. This limitation restricts CNNs' ability to handle variations in object orientation or perspective, presenting challenges for applications where viewpoint changes are common [9, 10]. Additionally, CNNs tend to focus on textures rather than shapes, contrasting with human perception, which prioritizes shape recognition [11]. A common but inefficient approach to address these limitations is data augmentation, where the model is trained on multiple views of objects. However, this solution is not universal and becomes impractical in real-world scenarios due to limited data availability.

A more efficient solution is therefore to decompose images into constituent parts and objects and leverage the linearity of part-object spatial relationships (e.g., pose matrix multiplication as used in computer graphics) to generalize across all viewpoints. Substantial evidence further supports the idea that humans interpret visual scenes by organizing them into part-whole hierarchies, modeling the spatial relationships between parts and their wholes as viewpoint-invariant coordinate transformations between their intrinsic coordinate frames [12, 13]. This is precisely the aim of capsule networks, which seek to hierarchically encode explicit pose representations of parts and objects [8, 14, 10, 15, 16]. The hierarchical structure in capsule networks enables them to encode explicit pose representations for both parts and objects, allowing changes in viewpoint to be modeled as simple linear transformations on these poses. Such hierarchical encoding enables capsules to capture and preserve the spatial relationships and transformations between parts and wholes more effectively than CNNs, which primarily focus on achieving transformation invariance, a process that often leads to the loss of critical spatial information. Unlike CNNs, capsule networks are designed not just for invariance but for approximately equivariant inference, allowing transformations in input to correspond directly to meaningful changes in capsule activations.

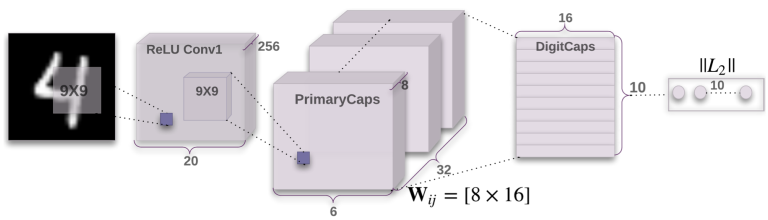

Figure 2.1: Original capsule networks architecture by Sabour et. al [1]

The core of a capsule network's ability to achieve part-whole hierarchies lies in the "routing-by-agreement" mechanism, also known as dynamic routing [1]. This mechanism enables capsules to communicate and adjust their connections based on the level of agreement between their outputs. The dynamic routing process is designed to preserve equivariance, allowing the network to recognize objects regardless of changes in orientation or position within the input space. By dynamically assigning part capsules to object capsules, the network captures relationships between parts and wholes more effectively than the fixed pooling strategies in CNNs, which select the most active features without considering their relational context. Through dynamic routing, the agreement between lower-level capsule outputs and higher-level capsule predictions filters out irrelevant features, emphasizing only those that contribute meaningfully to object understanding. This high-dimensional filtering enhances both feature detection and representation learning.

However, the routing mechanism in capsule networks encounters uncertainties from multiple sources, which can affect the accurate assembly of objects from constituent parts. For instance, occlusion of object parts, often due to certain viewpoints, can lead to incomplete information about features, creating ambiguity in part-to-object assignments. Additionally, sensor-captured data can be noisy, potentially leading to inaccuracies in input signals received by capsules. Such noise disrupts the reliability of votes cast by lower-level capsules for higher-level capsules, introducing uncertainty into routing decisions. Objects with symmetrical features, such as spherical shapes, further complicate pose determination, creating challenges for accurate capsule routing due to ambiguous part-object relationships. The dynamic routing process itself, which involves iteratively adjusting routing coefficients based on capsule agreement, can also contribute to uncertainty. As the final routing outcomes may depend heavily on initial conditions and the variability of input data, these uncertainties underscore the challenges dynamic routing face in modeling part-whole relationships accurately. They also highlight the importance of developing robust routing mechanisms that can effectively manage these uncertainties.

To quantify the uncertainty or unpredictability associated with a random variable X, we use the concept of entropy, denoted as $H(X)$. Introduced by Claude Shannon, entropy is a fundamental concept in information theory that measures the average amount of information generated by a stochastic source of data. Mathematically, the entropy of a discrete random variable X with probability mass function $p(x)$ is defined as:

$$H(X) = -\sum_{x \in X} p(x) \log p(x)$$

where $p(x)$ represents the probability of occurrence of each possible value x of X, and the logarithm is typically taken in base 2, giving the entropy in bits. In this context, the higher the entropy, the greater the uncertainty of the outcome, meaning more information is required to describe the random variable. Conversely, lower entropy indicates less uncertainty, with fewer bits needed to represent the information. Entropy plays a crucial role in various applications, including data compression, communication theory, machine learning, and deepfake detection, as it provides a measure of the efficiency of information encoding.

Building on entropy's foundational role in information theory, our study integrates entropy within the dynamic routing process of capsule networks. This approach aims to enhance both the performance and interpretability of capsule networks through an information-theoretic refinement of the routing mechanism. Specifically, our method introduces a straightforward adjustment to the routing process that improves interpretability without adding model complexity and is compatible with other routing-by-agreement methods where probability distributions are involved. This contribution advances the development of more robust routing algorithms by refining the voting process based on entropy, promoting a balance between interpretability and performance.

2.2 Related Works

Capsule networks were initially limited in their ability to manage uncertainty or ambiguity in part-object relationships, which affected their robustness in conditions involving noisy or occluded data. Entropy has since been applied in capsule networks primarily to enhance routing mechanisms, improving both network stability and performance. By incorporating entropy into the routing mechanism, capsule networks can more robustly assemble objects from their parts by accounting for uncertainty in part-object relationships. Low entropy indicates closely clustered votes, signaling a strong agreement among capsules regarding the presence of an object.

For instance, De et al. [17] utilize entropy to evaluate uncertainty and alignment among capsule votes. Their approach applies entropy in two key areas: assessing the agreement of votes for each object capsule and determining capsule activation based on this agreement.

Specifically, De et al. [17] compute the average negative entropy to quantify vote agreement, capturing the consensus of the capsule's pose parameters. This enables the model to assess the certainty in the vote patterns regarding the object capsule's pose. Furthermore, the activation of each object capsule is influenced by the level of agreement, with higher agreement leading to increased activation. The authors use a Binomial distribution to describe the expected support each capsule receives, allowing for a probabilistic evaluation of capsule activations based on vote agreement and support. This method provides a foundation for understanding capsule activation through an entropy-informed perspective.

In their Variational Bayes (VB) routing algorithm, Ribeiro et al. [18] utilized entropy to evaluate the agreement among votes from lower-level capsules. The method utilizes the differential entropy of a higher capsule's Gaussian-Wishart variational posterior distribution to capture the degree of vote alignment. In this approach, differential entropy provides a way to quantify uncertainty within the variational distribution, where higher entropy indicates greater uncertainty in vote alignment. By incorporating this measure, the VB routing algorithm can adjust capsule activations based on the estimated agreement, enhancing the model's interpretability regarding how capsules agree on vote patterns.

Venkataraman et al. [19] proposed EntrLoss, a novel loss function designed to enhance compositionality within capsule networks. This loss function is derived from the entropy of the routing weights between layers, where a lower entropy in these weights encourages the network to make more decisive, parse-tree-like routing decisions. By minimizing entropy, EntrLoss encourages the model to focus on a dominant routing weight per capsule connection, which supports compositionality in representation learning. This selective routing mechanism improves the network's ability to distinguish relevant components, resulting in superior performance in classification tasks when compared to recent capsule models.

A method called Routing Entropy Minimization (REM) was introduced by Renzulli et al. [20] to improve the explainability of capsule networks (CapsNets) by reducing entropy within the routing mechanism. By reducing entropy, REM encourages a parse-tree-like structure in the routing process, thereby strengthening the interpretability of how capsules interact. The approach minimizes the entropy of coupling coefficients between capsules, promoting sparsity and lowering variability in routing, which results in more defined relationships among capsules. This technique leads to a more explainable CapsNet by enforcing selective, stable routing paths that contribute to a clearer understanding of decision-making within the network.

A novel routing algorithm called EntMin was proposed by Edstedt et al. [21], aiming to enhance the routing process in capsule networks by minimizing the entropy of capsule output distributions. By introducing a non-linear transformation to the capsule output distribution, EntMin encourages a focused and efficient routing mechanism. The transformation decreases entropy in the capsule outputs, promoting selective connections between capsules that enhance interpretability and stability. This entropy-minimization strategy supports effective routing by maintaining diversity in the network's output estimates, contributing to improved routing performance.

An entropy-integrated loss function was introduced by Venkatraman et al. [22] to improve capsule networks' capacity for detecting changes in compositional structures. By minimizing the entropy of routing weights in the network, their approach encourages a parse-tree-like structure where one routing weight dominates, improving capsule network interpretability. Their findings reveal that capsule networks with reduced routing weight entropy are better at detecting structural changes, suggesting a promising approach for enhancing model sensitivity to compositional variations.

2.3 Methodology

Greff et al. [23] argue that current limitations of neural networks are due to the binding problem, which prevents them from incorporating sensible object-centric representations. As quoted in Reference [24] "The Binding Problem concerns how items that are encoded by distinct brain circuits can be combined for perception, decision, and action." Biological neural networks in human brains are said to overcome the binding problem by enabling flexible and dynamic binding of information belonging to separate entities [25]. Despite significant advances, most deep learning systems today still struggle with compositionality, where objects are represented in a hierarchical structure similar to a parse tree [26, 27].

2.3.1 Routing-by-Agreement Mechanism

The dynamic routing mechanism in capsule networks tackles the challenge of compositionality by iteratively adjusting routing coefficients between lower-level and higher-level capsules. These adjustments are guided by the agreement between their output vectors, enabling the network to establish structured relationships between parts and wholes. In all but the first capsule layer, the input $\mathbf{s}_j$ is a weighted sum of prediction vectors $\hat{\mathbf{u}}_{j|i}$, computed as:

$$\hat{\mathbf{u}}_{j|i} = \mathbf{W}_{ij}\mathbf{u}_i \tag{2.1}$$

where $\mathbf{u}_i$ is the output of a capsule in the layer below.

This gives:

$$\mathbf{s}_j = \sum_i c_{ij} \hat{\mathbf{u}}_{j|i} \tag{2.2}$$

where c_{ij} are coupling coefficients determined by dynamic routing, and W_{ij} is the weight matrix.

The coupling coefficients c_{ij} between capsule i and all capsules in the layer above sum to 1 and are computed using a routing softmax. The initial logits b_{ij} represent the log prior probabilities of coupling capsule i to capsule j. The log priors b_{ij}, which depend on the type and location of capsules i and j but not the input, can be learned alongside other weights. The initial coupling coefficients are refined iteratively by measuring the agreement a_{ij} between the output $\mathbf{v}_j$ of capsule j and the prediction $\hat{\mathbf{u}}_{j|i}$ from capsule i:

$$a_{ij} = \mathbf{v}_j \cdot \hat{\mathbf{u}}_{j|i}. \tag{2.3}$$

This agreement, treated as a log-likelihood, is added to b_{ij} to update the coupling coefficients. The dynamic routing process can be summarized as follows:

The loss function used in capsule networks is designed to encourage the correct classification of entities represented by the capsules. The margin loss function is commonly employed, which is defined as follows:

$$L_{\text{margin}} = \sum_k T_k \max(0, m^+ - ||y_k||)^2 + \lambda(1 - T_k) \max(0, ||y_k|| - m^-)^2 \tag{2.4}$$

where:

Algorithm 1 Dynamic Routing Algorithm

ROUTING ($\hat{u}_{j|i}$, r, l)

for all capsule i in layer l and capsule j in layer $(l{+}1)$: Initialize $b_{ij} \leftarrow 0$

for r iterations **do**

  for all capsule i in layer l: $c_i \leftarrow \text{softmax}(b_i)$

  for all capsule j in layer $(l{+}1)$: $s_j \leftarrow \sum_i c_{ij}\hat{u}_{j|i}$

  for all capsule j in layer $(l{+}1)$: $v_j = \frac{||s_j||^2}{1+||s_j||^2}\frac{s_j}{||s_j||}$

  for all capsule i in layer l and capsule j in layer $(l{+}1)$: $b_{ij} \mathrel{+}= \hat{u}_{j|i} \cdot v_j$

end for

Return: Higher-layer capsules v_j

- T_k is a binary indicator (1 if class k is present, 0 otherwise).
- m^+ and m^- are the desired margins for the presence and absence of a class, respectively.
- λ is a weighting factor that down-weights the loss for absent classes.

This loss function encourages the length of the output capsule vector to be close to 1 when the corresponding class is present and to be less than m^- when the class is absent, effectively training the network to distinguish between different classes based on the capsule outputs.

Some studies suggest that routing mechanisms may not be strictly necessary for achieving high performance in capsule networks [28, 29]. For instance, Paik et al. [28] conducted a comparative analysis of capsule networks with and without routing algorithms, ensuring the network architecture remained constant to minimize confounding factors. They tested two alternative approaches: uniform routing, where all link strengths are fixed and assigned equal values, and random routing, where link strengths are sampled from a random distribution at each iteration. Surprisingly, these simpler methods not only maintained performance but, in some cases, even improved it. This unexpected outcome may arise from the inherent limitations of traditional routing algorithms, which tend to over-polarize link strengths, leading to overconfidence.

Focusing on the foundational concern of compositionality that capsule networks aim to address, rather than merely emphasizing performance on standard datasets, Venkatraman et al. [22] investigate the challenges posed by uncertainty in dynamic routing. They highlight that traditional routing algorithms often result in high entropy in their routing weights, which can undermine the compositional learning capabilities of these networks. They argue that reducing entropy enhances the compositional structure of the learned representations, facilitating more accurate modeling of part-whole hierarchies. Without such entropy reduction, models may struggle to effectively capture compositional relationships, which are critical for tasks requiring structural understanding beyond mere performance metrics. Their experiments further demonstrated that capsule networks with lower-entropy routing weights outperformed CNNs in detecting compositional changes, outperforming prior methods like scaled-distance-agreement routing [30].

More recently, Venkataraman et al. [19] reaffirmed the foundational role of routing in capsule networks, as originally envisioned, and emphasized its effectiveness in fostering compositionality under specific conditions. They proposed a novel loss function designed to encourage routing behaviors that align with forming a tree-like structure, resulting in representations closely resembling parse trees. This structured approach underscores the importance of routing weights in capturing the hierarchical and compositional nature of visual scenes, further validating the unique strengths of capsule networks in this domain.

2.3.2 Entropy-Integrated Routing

To address the limitations of traditional dynamic routing, we propose an enhanced algorithm that incorporates entropy as a central element of the routing process. Entropy, as a measure of uncertainty, offers a systematic and efficient means to guide routing decisions, enabling the network to dynamically adjust routing coefficients based on the certainty of capsule connections. This integration is achieved by introducing an entropy regularization term during the final routing iteration, providing a principled approach to refining the routing process. The overall framework of this method is illustrated in Figure 2.2.

Importantly, this enhancement is designed to be both efficient and scalable, requiring no modifications to the network's core architecture or the addition of extensive new parameters. As a result, the proposed approach introduces minimal computational overhead while delivering noticeable improvements in performance and interpretability.

Furthermore, the entropy-based approach can be generalized to other routing-by-agreement methods, enhancing both interpretability and performance. By dynamically adjusting the regularization based on the data, the network learns

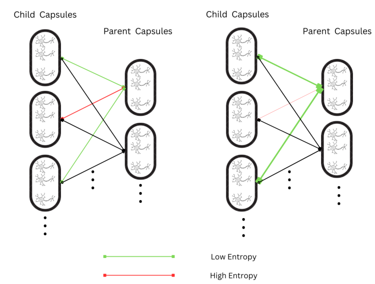

Figure 2.2: Overall entropy regularization

to balance routing assignments in a data-driven manner. In cases where routing decisions are ambiguous, entropy allows the model to retain flexibility, enabling it to delay confident decisions until sufficient information is available. This not only increases the efficiency of the routing process but also improves the network's transparency, aligning its decision-making more closely with the characteristics of the underlying data. The derivation of our proposed method is outlined below:

In the dynamic routing algorithm, we update the routing coefficients iteratively between the input capsules and output capsules. The routing coefficient, denoted as c_{ij}, determines the contribution of the i-th input capsule to the j-th output capsule. This coefficient is computed using a **Softmax** function applied to the logits b_{ij}, which represent the strength of the relationship between the input and output capsules.

We introduce an entropy term to regulate the routing process. The entropy term quantifies the uncertainty in the routing decisions. First, we calculate the entropy $H(c_j)$ of the routing coefficients c_{ij}, derived from **Softmax** function, which measures the uncertainty in the routing:

Algorithm 2 Entropy Integrated Dynamic Routing Algorithm

ROUTING ($\hat{u}_{i|j}$, r, l , λ, ϵ)

for all capsule i in layer l and capsule j in layer (l+1) : Initialize $b_{ij} \leftarrow 0$

for r iterations **do**

  for all capsule j in layer $l+1$: $c_{:,j} \leftarrow \text{softmax}(b_{:,j})$

  for all capsule j in layer (l+1): $s_j \leftarrow \sum_i c_{ij}\hat{u}_{i|j}$

  if $r =$ final iteration **then**

    $H(c_j) = -\sum_i c_{ij}\log(c_{ij} + \epsilon)$

    $s_j \leftarrow \sum_i c_{ij}\hat{u}_{i|j} - \lambda H(c_j)$

  end if

  for all capsule j in layer (l+1): $v_j = \frac{||s_j||^2}{1+||s_j||^2}\frac{s_j}{||s_j||}$

  for all capsule i in layer l and capsule j in layer (l+1): b_{ij} += $\hat{u}_{i|j} \cdot v_j$

end for

Return: Higher-layer capsules v_j

$$H(c_j) = -\sum_i c_{ij} \log(c_{ij} + \epsilon) \tag{2.5}$$

Here, ϵ is a small constant added for numerical stability to avoid taking the logarithm of zero. The entropy term $H(c_j)$ captures the degree of uncertainty in the routing: higher entropy indicates greater uncertainty, while lower entropy indicates more confident routing decisions.

We modify the routing process by incorporating the entropy into the prediction updates. The adjusted prediction vector, denoted as s_j, is computed by subtracting the entropy integration term from the routing coefficients c_{ij} and then multiplying by the input capsule predictions:

$$s_j \leftarrow \sum_i c_{ij} \hat{u}_{i|j} - \lambda H(c_j) \tag{2.6}$$

Here, entropy_coeff is a hyperparameter that controls the strength of the entropy integration. This term reduces the contribution of uncertain routes (those with higher entropy), effectively encouraging more confident routing decisions. The proposed entropy integration stabilizes the dynamic routing algorithm by adjusting the routing coefficients based on their uncertainty. The modified prediction update s_j, which incorporates the entropy term, leads to more stable and interpretable routing decisions.

Finally, the output of the dynamic routing process is the output capsule vector, which needs to be squashed to ensure that its length is within a unit ball. The squashing function is applied to the prediction vector s_j. The non-linear squashing function ensures that the output vectors have a unit norm, which is necessary to maintain the stability of the routing process.

In summary, the addition of the entropy term introduces a bias toward more confident routing decisions. Without entropy integration, the routing coefficients c_{ij} are highly sensitive to small changes in the logits, which can lead to instability when multiple routes have similar logits, resulting in high uncertainty. By incorporating the entropy term, we penalize uncertain predictions, i.e., when the **Softmax** distribution c_{ij} is more spread out (indicating high entropy). This forces the routing process to favor more certain predictions (lower entropy), reducing instability caused by uncertain routing decisions.

2.4 Experiments

We performed evaluations on our proposed modification to verify its effectiveness in practical settings as it will be explored in the following section.

2.4.1 Experimental Setup

In this section, we evaluate the performance of our proposed entropy-integrated dynamic routing algorithm using the CIFAR-10 dataset. The CIFAR-10 dataset comprises 60,000 color images of size 32x32, divided into 10 distinct classes. Among these, 50,000 images are used for training, and the remaining 10,000 images serve as test samples, providing a well-balanced foundation for evaluating classification performance.

For this experiment, we partitioned the model's parameters into two distinct groups to allow for specialized optimization strategies. The parameter groups and their corresponding optimization details are as follows:

- **General Model Parameters**: These include all primary parameters of the network, optimized with the Adam optimizer set at a learning rate of 0.001.

- **Entropy Coefficient Parameter**: The entropy coefficient, a critical component in our routing mechanism, is optimized independently using the Adam optimizer, also at a learning rate of 0.001. This separation allows for tailored adjustments to the entropy parameter without affecting the general model parameters.

The model was trained for 500 epochs to ensure sufficient learning and stability in convergence, using a batch size of 64 to balance computational efficiency with model accuracy. The experimental setup was conducted on a single NVIDIA GeForce RTX 4090 GPU. To enhance training efficiency and convergence, we employ a learning rate scheduler (`ReduceLROnPlateau`) for each optimizer. This scheduler reduces the learning rate when validation performance reaches a plateau, adapting the training process dynamically. The patience values for each parameter group were set according to their distinct convergence requirements: the general model parameters have a patience of 170 epochs, while the entropy parameter has a shorter patience of 70 epochs. This distinction reflects the importance of the entropy coefficient's rapid adaptation in guiding the dynamic routing mechanism.

The entropy coefficient was initialized at a very small negative value. Throughout the training, it was gradually optimized to stabilize the routing process, balancing between model predictions and the entropy-integrated dynamic routing. This controlled approach minimizes early-stage bias and maximizes the entropy coefficient's contribution as training progresses.

2.4.2 Results Analysis

Table 2.1 summarizes the performance of our entropy-integrated dynamic routing model in comparison to the baseline dynamic routing model, based on multiple experimental runs. The mean accuracy of our method reached 85.580%, an improvement over the baseline accuracy of 84.610%. This increase demonstrates the effectiveness of entropy integration in enhancing generalization capabilities and the robustness of the routing mechanism, particularly for complex class distinctions.

As training advanced, the entropy coefficient, initially set to -0.001, stabilized around a value of +0.007, which emerged as the optimal setting for maximizing model performance based on the CIFAR-10 dataset. This adaptive adjustment of entropy facilitated more effective routing without excessive restrictions during early training. The entropy coefficient convergence has been presented in the Figure 2.3.

To conclude, by moderating the uncertainty within the routing process, the entropy factor enhanced the network's flexibility in dynamically adjusting the connections between capsules. This adaptive routing approach contributed to improved handling of challenging or ambiguous samples, as well as superior feature representation across diverse classes. Consequently, the model achieved a higher

Table 2.1: Comparison of CIFAR-10 Test Accuracy

Method	**Backbone**	**Accuracy (%)**	**Param**
Dynamic Routing [1]	Simple CNN	84.610	8.2M
EM Routing [14]	Simple CNN	82.19	460K
Inverted Dot-Product Attention Routing [31]	Simple CNN	85.17	560K
VB-Routing [18]	Simple CNN	83.8	323K
Entropy-Integrated Dynamic Routing	Simple CNN	**85.580**	8.2M

accuracy of 85.580%, underscoring the benefits of entropy in refining the routing mechanism and improving classification performance.

2.4.3 Entropy and Distribution Spread

We also found that applying entropy is particularly effective when the number of child and parent capsules is smaller. Fewer capsules lead to more pronounced diversity in routing assignments, allowing entropy integration to enhance decision-making. In this context, entropy promotes uncertainty where needed, refining connection strengths and enabling exploration of multiple possibilities. However,

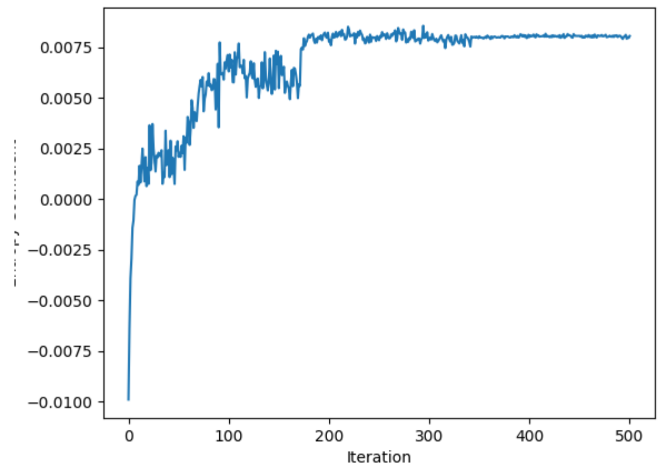

Figure 2.3: Entropy coefficient during 500 epochs of training

as the number of child-to-parent capsule connections increases, the impact of entropy on routing diversity diminishes. With more capsules, the network has greater pathways to explore, making iterative refinement of coupling coefficients more critical than entropy adjustments. While entropy's contribution to diversity may fade in larger networks, it remains vital for maintaining interpretability, preventing overly confident routing decisions, and ensuring robust outcomes.

Consider the entropy function $H(p) = -\sum_{i=1}^{n} p_i \log p_i$. To show that entropy increases with a more uniform distribution, we can use the concept of *concavity* of the logarithm function.

For any distribution $p = (p_1, p_2, \ldots, p_n)$, entropy is maximized when $p_i = \frac{1}{n}$ for all i (uniform distribution). This is because $H(p)$ is a concave function, meaning that the more spread out the probabilities, the higher the entropy. If the distribution becomes more concentrated (i.e., one p_i is much larger than others), the entropy decreases. This is because uncertainty is reduced, and the logarithm term diminishes for low-probability votes.

When the number of contributing votes n increases and the distribution becomes more uniform, the entropy increases, leading to greater uncertainty. When n is smaller and the distribution p is more peaked, the entropy is lower, implying greater confidence in the few key votes. This behavior justifies why the proposed method works better with fewer votes contributing to the entropy: the lower entropy allows the system to focus on the most relevant votes, resulting in a more decisive and confident outcome.

2.4.4 Staged Entropy Integration

Our experiments indicate that allowing dynamic routing to function freely during initial iterations, followed by incorporating entropy adjustments in the final layer, yields superior performance compared to applying entropy from the outset. Initially, dynamic routing benefits from exploring potential connections between child and parent capsules, refining coupling coefficients based on raw votes. This approach enables the network to establish foundational relationships without the constraints of entropy integration. By delaying entropy application until the final iteration, the network can emphasize certainty and robustness in decision-making. In this crucial stage, entropy reinforces connections where certainty is high while fostering cautious connections where certainty is low. This final adjustment consolidates learned knowledge and enhances the model's interpretability regarding which parent capsules to activate. In contrast, applying entropy from the beginning can overly restrict the model's exploration of diverse connections, leading to suboptimal routing patterns. By introducing entropy only in the final iteration, we preserve flexibility during earlier stages, enabling the model to integrate broader knowledge and evaluate multiple hypotheses. This phased approach strikes a balance between exploration and certainty, resulting in more efficient and interpretable decision-making.

2.5 Discussion

The original dynamic routing algorithm relied on dot products to compute agreement between votes, making it highly sensitive to even small differences between capsule vectors. This could result in inefficiencies in identifying the correct capsule activations. Incorporating entropy into the dynamic routing process plays a crucial role in improving the efficiency and interpretability of the capsule network. In this context, entropy serves as a regularization mechanism that encourages stronger child-parent capsule connections when there is high certainty and reduces overly confident connections when certainty is low. By integrating this concept, the routing process becomes more informed, balancing the certainty of routing decisions with the need for flexibility in ambiguous cases.

When the coupling coefficients between child and parent capsules exhibit higher certainty (i.e., the softmax probabilities are skewed toward specific parent capsules), the entropy of the distribution is low, and the algorithm solidifies these connections. This ensures that when the model is confident, the dynamic routing process strengthens those child-parent capsule links, leading to clearer and more reliable decision-making. Conversely, when the certainty is lower, the entropy is higher, encouraging a more exploratory approach, allowing input capsules to consider multiple parent capsules as potential candidates. This flexibility allows the model to integrate more information from different perspectives, which is vital for complex tasks where multiple interpretations of the data might be valid.

Moreover, dynamic routing is known to be computationally expensive due to the iterative nature of the process. Each iteration involves updating logits, applying softmax, and calculating weighted sums. By leveraging entropy, the algorithm optimizes the limited iterations, guiding the routing process more efficiently. Entropy provides an interpretable measure that adjusts the strength of routing based on the certainty of the connections. This reduces the likelihood of unnecessary computation and focuses computational resources on refining connections where certainty is already high.

Overall, the use of entropy, which is an interpretable concept from information theory, not only enhances the efficiency of the routing process but also provides insight into how capsules interact and make decisions. By maximizing the effectiveness of the iterations through entropy, the model incorporates more meaningful knowledge into the routing process while maintaining computational efficiency. Thus, entropy acts as a bridge between maximizing the information flow in the network and ensuring the interpretability of the decision-making process within the dynamic routing framework.

2.5.1 Comparing Entropy-Integrated Routing with Existing Methods

Unlike our dynamic routing-based approach that directly integrates entropy into the routing process, Ribeiro et al. [18] introduced the Variational Bayes (VB) routing algorithm, which incorporates entropy through the differential entropy of the variational posterior distribution of the higher capsule's Gaussian-Wishart parameters. The VB routing approach explicitly models uncertainty over capsule parameters and routing coefficients, thereby addressing issues like variance-collapse that can occur in maximum likelihood estimation (MLE)-based models. In VB routing, the variational posterior for capsule j is defined as:

$$q^*(\mu_j, \Lambda_j) = N(\mu_j | m_j, (\kappa_j \Lambda_j)^{-1}) \mathrm{Wi}(\Lambda_j | \Psi_j, \nu_j),$$

with an entropy approximation as follows:

$$h[q^*(\mu_j, \Lambda_j)] \approx E[\ln \det(\Lambda_j)] = D \ln 2 + \ln \det(\Psi_j) + \sum_{i=0}^{D-1} \psi \left(\frac{\nu_j - i}{2} \right).$$

This framework allows VB routing to estimate capsule activation probabilities, a_j, as:

$$a_j = \sigma \left(\beta_a - (\beta_u + E[\ln \pi_j] + E[\ln \det(\Lambda_j)]) \odot r_j \right),$$

achieving competitive performance with a slightly higher error rate than the traditional dynamic routing algorithm on CIFAR-10 but using fewer parameters.

In contrast, our approach does not require modeling uncertainty over capsule parameters or complex variational inference steps. Instead, we directly enhance the routing process by modifying the routing coefficients c_{ij} based on an entropy term. where $H(c_j)$ denotes the entropy, calculated to promote robust routing decisions. This entropy-integrated refinement directly influences the routing logits, allowing our approach to dynamically adjust routing behavior based on uncertainty without requiring a separate variational distribution.

By embedding entropy directly into the routing process, our method achieves a simplified yet effective mechanism for handling uncertainty, enhancing stability and interpretability without the computational overhead of VB routing. This approach enables capsules to activate based on entropy-modified agreement during the routing process itself, striking a balance between performance and interpretability.

Furthermore, in our comparison with approaches that incorporate entropy into the loss function, such as the EntrLoss method introduced by Venkataraman et al. [19], we observe that EntrLoss aims to enhance compositionality by minimizing the mean entropy of the routing weights for each shallower capsule type, defined as:

$$\text{EntrLoss} = \frac{1}{N_l|G|}\sum_{g\in G}\sum_{i=0}^{N_{l-1}}\left(-c_{ij}^{l+1}(g)\log(c_{ij}^{l+1}(g))\right),$$

thereby enforcing a selective routing mechanism. While EntrLoss emphasizes clear part-whole relationships by prioritizing a singular routing weight per capsule type, the accuracy metric of the non-regularized model was slightly higher than that of the regularized model on the CIFAR-10 dataset, despite having almost twice as many parameters as dynamic routing. Our method, however, allows for the dynamic adjustment of complex interrelations among capsules, without introducing complexity to the base network.

Our approach also stands apart from the EntMin non-iterative algorithm by Edstedt et al. [21], which achieved 1.6% increase in validation accuracy compared to the dynamic routing algorithm on the CIFAR-10 dataset, enhancing capsule routing through entropy considerations. Our method modifies routing coefficients c_{ij} by integrating entropy integration. Conversely, the EntMin algorithm minimizes the entropy of capsule output distributions using a non-linear transformation:

$$p_j = \text{EntMin}(\hat{P}_j) = \frac{||\hat{P}_j||^{\alpha}\hat{P}_j E[||\hat{P}_j||]}{E[||\hat{P}_j||^{\alpha+1}]},$$

Here, $\alpha > 0$ acts as a hyper parameter, representing how much the entropy of the distribution should be reduced.

While both approaches promote diversity in output distributions, our proposed method emphasizes a direct link between output and entropy.

In the work by De et al. [17], the authors introduce a method that utilizes negative entropy to quantify vote agreement in capsule networks, emphasizing the certainty of capsule poses. They compute the vote certainty as the negative entropy $-\mathcal{H}(M_j)$ of a Gaussian distribution $N(M_j|\mu_j, \sigma_j)$, where $\mathcal{H}$ denotes the entropy function, and μ_j and σ_j represent the mean and standard deviation of the capsule votes. This entropy-integrated mechanism adjusts activations, favoring capsules with low entropy, which indicates high vote agreement and facilitates consistent pose assignment, although, the performance of the proposed model was not evaluated on the CIFAR-10 dataset.

In contrast, our proposed method innovatively integrates entropy integration directly into the routing coefficients c_{ij}, thereby refining the vote weights throughout the routing process. By penalizing uncertainty in the distribution of votes, our approach enhances routing consensus, enabling a more effective alignment of capsules. While De et al. primarily focus on pose certainty via negative entropy, our method introduces a novel mechanism that applies entropy integration to dynamically adjust routing weights, offering a more nuanced approach to handling vote distributions.

And finally, The Routing Entropy Minimization (REM) technique [20] reduces the entropy

$$H_j = -\sum P(\tilde{c}_{\xi-j}|y_\xi = j) \log_2 P(\tilde{c}_{\xi-j}|y_\xi = j)$$

targeted at the j-th output capsule, with the total entropy defined as $H = \frac{1}{J}\sum_j H_j$. The REM updates routing coefficients using a softmax function influenced by capsule output alignment, resulting in static, low-entropy paths

$$c_{ij} = \frac{1}{k} \frac{e^{b_{ij}+t}}{\sum_{r=1}^{t} v_r u_j W_{ij}}$$

However, they encountered unexpected classification accuracies on the CIFAR-10 and Tiny ImageNet datasets, attributing this to noise introduced in their coupling coefficients, which led to increased uncertainties in the classification task.

In contrast, our method adjusts routing coefficients based on calculated entropy during the final iteration, resulting in more informative capsule outputs, while achieving competitive performance on the CIFAR-10 dataset.

Chapter 3

Application to deepfake detection

The rise of deepfake technology has transformed both digital media and the challenges associated with authenticating visual content. As advances in generative AI have enabled increasingly realistic manipulations, from facial swaps to synthetic voice generation, the potential for malicious misuse has grown significantly. This chapter investigates the applications of capsule networks in deepfake detection, presenting an entropy-integrated dynamic routing mechanism that enhances the model's ability to discern manipulated media from authentic content. Capsule networks, known for their hierarchical feature representation capabilities, are particularly well-suited for detecting subtle inconsistencies in deepfake images and videos. By integrating entropy into the routing algorithm, this approach aims to improve interpretability and robustness in high-stakes environments where accuracy and reliability are essential.

The following sections provide a comprehensive overview of deepfake technology, its applications, and societal implications. A discussion on existing deepfake detection methods highlights the challenges and gaps that capsule networks aim to address. We then introduce the proposed entropy-integrated capsule network architecture, detailing the methodology, training process, and evaluation setup. Experimental results demonstrate the effectiveness of entropy-integrated regularization in enhancing classification performance, followed by an analysis of the model's interpretability improvements. Finally, the paper discusses the broader implications of this approach in the context of evolving detection needs and the ethical considerations surrounding deepfake technology.

3.1 DeepFake Technology

Deepfake technology leverages generative artificial intelligence (generative AI) to create realistic-looking media, either by manipulating existing content or generating fake content entirely from scratch. The term "Deepfake" gained traction in late 2017 when a user on *Reddit* began sharing manipulated videos that swapped the faces of celebrities with those of adult film actors, bringing widespread public attention to the capability of AI to produce misleading content [32, 33]. This technology enables alterations to facial attributes, face swapping, and even the creation of media where individuals appear to say or do things they have not actually done [34]. Since 2021, advancements in deepfake technology, especially with the advent of large generative models like vision transformers and diffusion models, have enhanced the realism of generated content. These sophisticated models allow for the creation of deepfakes that may not have any real-world source, introducing new dimensions in content generation [35, 36, 37].

Deepfakes can be categorized into two primary types based on their generation method: those that manipulate existing media and those that utilize AI models to create content without a real-world origin. Conventional deepfakes typically alter existing media through methods like face swapping, where the face of one person is overlaid onto another in a video or image (illustrated in Fig. 3.2), or puppet-master techniques, which animate a target image based on the movements of a source image (shown in Fig. 3.3). The release of user-friendly applications like *FakeApp* and *DeepFaceLab* has enabled individuals without technical expertise to create deepfakes, raising ethical and security concerns regarding their potential for misuse [38, 39, 40]. While conventional deepfakes are often highly convincing, they are typically grounded in real content, making it possible in some cases to trace them back to their original source.

Deepfake technology has found diverse applications across several fields, notably in entertainment, education, healthcare, and marketing. In entertainment, deepfakes enable realistic video dubbing for actors to appear multilingual [41], and create special effects, such as reanimating historical figures [40]. Applications like *Reface* let users swap faces with celebrities, enhancing social media engagement [42]. In education, lifelike representations of historical figures foster an interactive learning experience, while in healthcare, deepfakes support medical training simulations, provide personalized patient avatars for treatment visualization, and enhance telemedicine interactions with realistic digital consultations [43, 40]. Furthermore, Alzheimer's patients benefit from animated representations of loved ones for therapeutic purposes [44]. In marketing, brands utilize deepfakes for personalized advertisements, superimposing customers' faces on models to boost engagement and sales [45]. However, the technology's positive potential is overshadowed by its association with negative uses, particularly in spreading misinformation and fake news. Realistic deepfakes are exploited to mislead the public, with videos manipulating political perceptions and potentially impacting elections [46]. They are also used for harassment, blackmail, and fraud, facilitating identity theft through fake identities in financial scams [47, 40, 48, 39]. These concerns over privacy and misinformation have spurred regulatory and detection efforts, such as Facebook's Deepfake Detection Challenge (DFDC) [49]. Fig. 3.1 summarizes some of the positive and negative applications of deepfake technology.

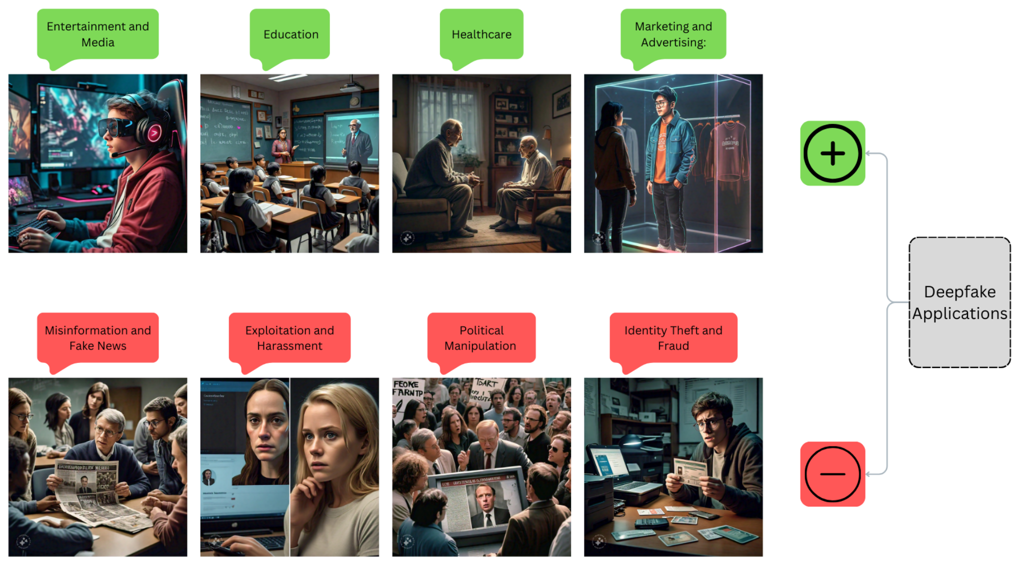

Figure 3.1: Beneficial and Harmful Applications of Deepfakes

3.1.1 Deepfake Detection

Deepfake detection methods can be categorized into three main approaches: forensic-based, machine learning, and hybrid methods. Forensic techniques in digital media analysis typically use signal processing to detect anomalies or manipulations, such as examining frequency components and biological signals. Traditional forensic methods often analyze statistical characteristics, like frequency domain analysis, to uncover image tampering [50, 51, 52]. These methods are adept at identifying artifacts left by deepfake generation processes, including resolution inconsistencies, warping, and visual anomalies like lighting mismatches or unnatural facial movements [53, 54, 55]. Machine learning (ML) techniques, on the other hand, employ advanced computational methods capable of autonomously extracting and analyzing complex features from data. Hybrid approaches combine various techniques to enhance detection accuracy and reliability, often integrating deep learning models such as CNNs for capturing spatial features and RNNs for temporal aspects in videos [56, 57, 58]. A multi-modal approach is also gaining traction, using both visual and audio components to detect inconsistencies, like mismatches in lip movement and speech, for higher accuracy in deepfake identification [59, 60, 61]. As deepfake technology advances, detection algorithms face increasing challenges, leading to a continuous arms race between attackers and defenders, making ongoing development crucial for effective detection [41].

Amidst the rising sophistication of deepfake content, capsule networks have emerged as a promising architecture for detecting manipulated media. Capsule networks, due to their ability to capture hierarchical relationships and pose information, are inherently suited for identifying subtle inconsistencies in the spatial and structural composition of images—features that are frequently manipulated in deepfake content. Unlike traditional neural networks, which often struggle to represent part-whole hierarchies in visual scenes, capsule networks can capture these relationships as viewpoint-invariant transformations, making them ideal for tasks requiring spatial hierarchy and transformation recognition [8]. This hierarchical structure enables capsule networks to detect unnatural alterations in pose, alignment, and other attributes, providing an interpretable, biologically inspired framework highly suitable for discerning between authentic and synthetic media. Thus, capsule networks offer a robust and interpretable approach, effectively positioning them as valuable tools in the ongoing development of deepfake detection systems.

Figure 3.2: Deepfake generation utilizing the Fast Face-Swap technique. Left: original image; middle and right: trained CageNet [2].

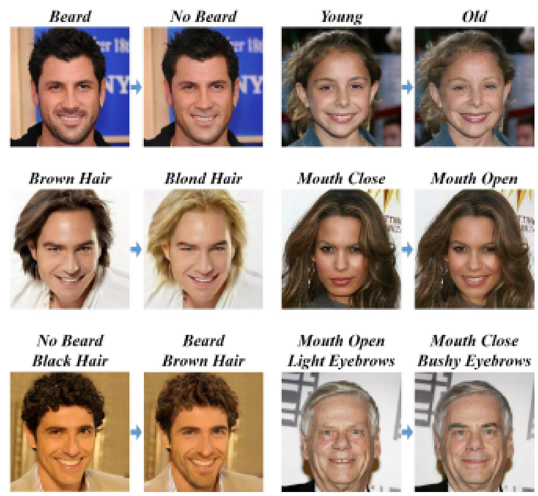

Figure 3.3: Facial attribute editing results from the AttGAN technique [3].

3.2 Capsule-based Deepfake Detection

Capsule networks have demonstrated significant potential in detecting various types of digital manipulations, such as spoofed audio, deepfake videos, and image forgeries, through their advanced feature representation capabilities. Capsule architectures have been adapted across multiple studies to enhance detection accuracy, leveraging their strengths in hierarchical feature capture and spatial consistency.

In [62], capsule networks were tailored for audio spoofing detection, specifically targeting synthetic and replay attacks. The authors introduced a modified dynamic routing algorithm that enhances the model's focus on subtle artifacts characteristic of spoofed audio. This approach achieved improved performance on the ASVspoof2019 dataset, with the network showing superior Equal Error Rates (EER) and tandem detection cost function (t-DCF) metrics compared to the standard capsule network. The modified dynamic routing mechanism in their system helped capture local audio variations essential for distinguishing between genuine and spoofed signals.

Capsule networks have also been applied to speaker verification systems for detecting replay attacks. [63] integrated capsule layers to process input spectrograms, utilizing dynamic routing to classify speech as either genuine or spoofed. Testing various input features, the model achieved competitive EER and t-DCF scores, notably improving detection accuracy with both Short-Time Fourier Transform (STFT) and Constant-Q Transform (CQT) spectrograms.

ArCapsNet, introduced by [64], focuses on audio splicing forgery detection and incorporates attention routing mechanisms to refine classification accuracy. Using cochleagram images of audio and an EfficientNet B4 backbone for feature extraction, ArCapsNet demonstrated high precision and recall metrics on datasets derived from TIMIT. Capsule layers dynamically routed features extracted from pre-trained models, yielding high F1 scores and reliable performance in audio manipulation detection.

In a different application, [65] proposed FACILE, a lightweight capsule network designed to classify malware images efficiently. By reducing the number of capsules while enriching hierarchical feature extraction, FACILE outperformed traditional CNNs across multiple datasets, achieving a favorable trade-off between accuracy and model complexity.

Further, [66] proposed the IR-Capsule model, a two-stream network combining a capsule-based feature extractor with an Inception ResNet, for face forgery detection. The capsule stream effectively captured hierarchical relationships and facial pose features, leading to notable improvements in classification accuracy over benchmarks like XceptionNet.

In the context of detecting deepfake videos, [67] developed a CapsuleNet-based framework that targets spatial inconsistencies within individual video frames. By combining CapsuleNet with an LSTM, the model captures temporal dynamics across frames, enhancing its accuracy in distinguishing real from manipulated media, even though it slightly trails XceptionNet in overall accuracy.

Further, [68] employed capsule networks to address generalization issues, particularly in face-swapping detection. Their model combines ResNet-18 for initial feature extraction with a multi-layer capsule network to capture spatial dependencies between facial components. This design, evaluated on the DFDC-P dataset, showed marked improvements in recall and F1 score, underscoring its robustness against imbalanced datasets. Similarly, [4] extended capsule networks for replay attacks and computer-generated content detection, using a VGG-19 feature extractor combined with primary and output capsules. Their approach showed strong performance in digital forensics, achieving near-perfect accuracy on specific datasets.

The Capsule-Forensics method in [69] also leveraged capsule networks for manipulated media detection, achieving efficient performance with fewer parameters by using primary and output capsules for binary and multi-class classification. Similarly, [70] combined capsule networks with the VIOLA-Jones algorithm, employing a capsule-based graph network for deepfake detection, which captured spatial relationships and achieved significant accuracy gains on various datasets.

Collectively, these studies underscore the adaptability of capsule networks in handling diverse manipulation detection tasks, from audio spoofing to malware classification. By tailoring capsule architectures to specific data types and introducing custom routing mechanisms, these works demonstrate the effectiveness of capsule networks in achieving high detection accuracy across various multimedia forgeries.

3.3 Entropy Integrated Capsule Forensics

This study is based on the Capsule Forensics model [69], serving as a baseline framework that leverages capsule networks with dynamic routing for face forgery classification. The capsule Forensics used a combination of primary capsules and output capsules. For feature extraction, the VGG-19 network pre-trained on the ImageNet dataset, truncated at the third max-pooling layer. The pre-trained network enabled transfer learning, allowing the capsule network to learn features specific to forgery detection. The output from VGG19 is passed to 10 primary capsules, consisting of a 2D convolutional layer, statistical pooling, and a 1D convolutional layer, which allows for effective feature extraction. These primary capsules are followed by class capsules in the final layer, which aggregate information from primary capsules to produce the classification results. The Capsule Forensics model included two output capsules corresponding to binary labels (real and fake) and extended the output layer to four capsules for multi-class classification (real, deepfake, Face2Face, FaceSwap). The detailed architecture of the Capsule Forensics model is illustrated in Figure 3.4.

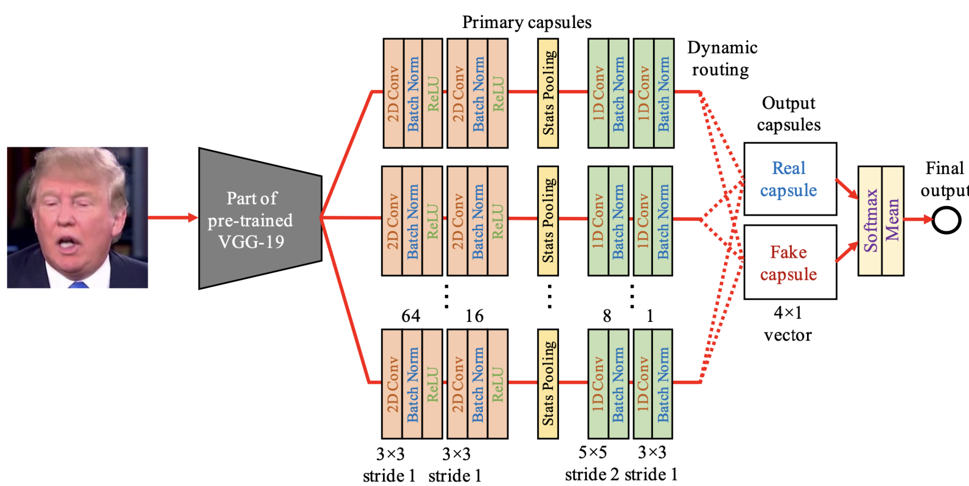

Figure 3.4: Capsule forensics capsule network architecture [4]

To enhance the effectiveness of our entropy-integrated dynamic routing, we applied the approach from Chapter 2 to the challenging problem of deepfake detection. By incorporating entropy integration, we discourage overconfidence in routing coefficients, encouraging a more distributed and interpretable voting process. This is achieved by penalizing highly concentrated coefficient distributions, preventing single capsules from dominating output votes. We hypothesize that this method could improve capsule-based deepfake detection.

In this approach, the addition of an entropy term introduces a bias toward more confident routing decisions. When entropy integration is absent, the routing coefficients c_{ij} are highly sensitive to small changes in the logits, which can cause instability in cases where multiple routes have similar logits, leading to high uncertainty. By incorporating the entropy term, uncertain predictions are penalized; specifically, when the **Softmax** distribution c_{ij} is more spread out (indicating high entropy), the routing process is forced to favor more certain predictions (lower entropy). This approach reduces instability by decreasing the likelihood of uncertain routing decisions.

3.3.1 Entropy Metric

The entropy H, calculated over the class probability distribution $\mathbf{p}$, quantifies prediction uncertainty, where higher entropy indicates increased uncertainty. In a binary classification setting, entropy is calculated for both correctly and incorrectly classified samples, allowing entropy comparisons between the predicted class and its alternative. This design provides insight into whether the model's lower entropy (greater confidence) correlates with prediction accuracy, thus serving as a robustness measure for the model's interpretability.

The study further proposes a fourfold categorization to interpret entropy values systematically. These categories include cases where:

1. **True Positive (TP)**: Entropy is typically lower for the correctly predicted

class, indicating higher confidence in the model's decision.

2. **False Negative (FN)**: In certain cases, higher entropy may appear for a correctly classified instance, suggesting residual uncertainty despite the correct prediction.

3. **False Positive (FP)**: For misclassified instances, lower entropy is often observed for the predicted (but incorrect) class, implying the model's over-confidence in an incorrect decision.

4. **True Negative (TN)**: Higher entropy for the predicted class, in the case of incorrect predictions, may reflect hesitation or ambivalence, showing the model's uncertainty in classifying the instance.

These cases enable an assessment of whether the model's confidence aligns with its predictive accuracy, while also aiding in the identification of uncertainty patterns that can guide model refinement. The proposed entropy-integrated metric advances interpretability by systematically linking confidence with accuracy in capsule network predictions, supporting the deployment of these models in high-stakes, reliability-sensitive environments.

3.4 Experimental Setup

To examine our hypothesis, we evaluate the performance of both the baseline Capsule Forensics model and our enhanced approach to assess its effectiveness in detecting fake images and videos generated by various computer manipulation techniques. The experimental setup consists of three main components: data preprocessing, network architecture, and evaluation metrics.

3.4.1 Data and Preprocessing

We utilized FaceForensics++ and DeepPhy datasets to assess the model's generalizability across different types of image and video manipulations.

The FaceForensics++ dataset is a comprehensive resource for deepfake detection research, comprising over 1,000 real videos and 4,000 manipulated videos generated using four popular manipulation techniques: DeepFakes, Face2Face, FaceSwap, and NeuralTextures. This dataset provides videos at multiple compression levels (raw, lightly compressed, and heavily compressed) to simulate real-world conditions, enhancing the dataset's relevance for robust model development. FaceForensics++ serves as a benchmark for evaluating the efficacy of deepfake detection methods, with a focus on detecting facial manipulation in uncontrolled, practical environments.

The DeePhy dataset is a novel deepfake phylogeny collection consisting of 5,040 manipulated videos designed to simulate evolutionary deepfake generation patterns. Each video is synthesized using three generation techniques: FSGAN, FaceSwap, and FaceShifter, applied sequentially up to three times, resulting in unique multi-stage manipulations. Real video sources are drawn from publicly accessible YouTube content featuring diverse demographic attributes. DeePhy aims to support research in deepfake detection, focusing on the challenges posed by phylogenetic deepfake evolutions, model attribution, and the sequential order of applied generative techniques.

Preprocessing involved resizing each frame to 300x300 pixels and dividing images into patches as needed for some experiments. Additionally, facial regions were cropped using a detection algorithm for more accurate classification in face-related tasks.

3.4.2 Training and Evaluation

The model was trained using the Adam optimizer with a learning rate of 5×10^{-4}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. A dropout rate of 5% and a batch size of 32 were used, and the training was run for 25 epochs. Random noise was introduced to the routing matrices during training to reduce overfitting as recommended, particularly useful for smaller datasets.

Performance was measured across these below metrics:

- **Binary Classification Accuracy**: Percentage of correctly classified real and fake images.

- **Classification Accuracy per Class**: For multi-class classifications, we calculated accuracy separately for each class to evaluate the model's robustness across different manipulation techniques.

3.4.3 Results Analysis

Tables 3.1 and 3.3 shows the results of our entropy-integrated routing algorithm compared to the baseline capsule forensic model. For the binary classification task, our method improved the accuracy from 91.22% to 92.27%. In the multi-class classification task, the accuracy improved from 92.02% to 92.74%. These improvements demonstrate the effectiveness of entropy integration in improving the discriminative power of the capsule network, particularly for challenging classification tasks like deepfake detection.

Method	**Accuracy (%)**
XceptionNet (299*299)	91.46
Capsule-Forensics (Old) + Noise (128*128)	88.11
IR-Capsule	91.87
Capsule-Forensics (New) + Dropout + Noise (300*300)	92.02
Our model	92.74

Table 3.1: Binary Classification Accuracy on FaceForensics++

Method	**Accuracy (%)**
XceptionNet (299*299)	91.33
Capsule-Forensics (Old) + Noise (128*128)	87.12
Capsule-Forensics (New) + Dropout + Noise (300*300)	91.22
Our model	92.27

Table 3.2: Multi Classification Accuracy on FaceForensics++

In comparison to baseline models like Capsule Forensics, our entropy integrated capsule network achieved competitive results without introducing complexity, demonstrating effective detection across varied datasets and attack types. The entropy coefficient, once optimized to 0.001, effectively enhanced the routing mechanism by managing uncertainty in capsule activations. The improvement in accuracy from 91.22% to 92.27% for binary classification and from 92.02% to 92.74% for multi-class classification indicates that the entropy-integrated routing contributes to better generalization, even in complex classification scenarios involving multiple manipulation types. The entropy integration led to more refined and reliable predictions, particularly in detecting subtle manipulations in deepfakes.

In our evaluations, we trained the proposed model on the FaceForensics++ dataset and performed cross-testing on the binary classification task using the DeePhy dataset. The results of these tests are presented in the table below, including class-specific accuracies and entropy metrics. The findings demonstrate that our model significantly outperforms the baseline model (Capsule-Forensics (New) + Dropout + Noise) in this cross-evaluation task.

Method	**Acc (%)**	**Real-class acc**	**Fake-class acc**	**FP (%)**	**TP (%)**	**TN (%)**	**FN (%)**
Capsule-Forensics	80.36	0.83	0.76	10.01	41.58	13.58	34.81
Our model	81.54	0.84	0.78	8.05	44.74	10.40	36.81

Table 3.3: Cross evaluation, initially trained on the FaceForensic++, tested on the DeePhy dataset

3.5 Discussion

In our proposed approach, we incorporate entropy into the routing coefficients to enhance both interpretability and efficiency within capsule networks. By introducing a regularization parameter λ, we control the influence of entropy, encouraging a balanced allocation of routing weights. This results in a smoother routing process by discouraging overly confident assignments and enhancing interpretability, particularly in higher capsule layers.

Our experimental results validate the hypothesis that entropy integration within the dynamic routing mechanism can enhance deepfake detection, especially in capsule-based models. This approach improves upon the baseline Capsule Forensics model through a simple yet novel modification in dynamic routing, preserving the part-whole relationship characteristic while boosting interpretability and performance. Our work builds on existing research showing that capsule networks are highly promising for the challenging task of deepfake detection. Moreover, as demonstrated by Narayan et al. [71], capsule networks exhibit robustness in detecting complex, unseen deepfakes, as evidenced in testing with the DeePhy dataset.

Our findings also indicate that deferring entropy integration until the final routing iteration maximizes its benefits. Initially, allowing the dynamic routing algorithm to freely explore potential capsule connections enables it to optimize raw coupling coefficients based on preliminary votes. This exploratory phase helps the network establish foundational relationships without being constrained by entropy integration. This approach enhances interpretability by reinforcing confident connections and promoting caution in areas of uncertainty, improving the model's ability to activate appropriate parent capsules.

Chapter 4

Conclusion

In this thesis, we applied the entropy-integrated routing mechanism to enhance the performance and interpretability of capsule networks in deepfake detection. By applying entropy integration selectively in the final routing iteration, our method effectively balances the inherent uncertainty in capsule activations, resulting in more confident and accurate predictions. This approach improves upon the standard capsule network framework, achieving better classification accuracy across datasets with diverse manipulation techniques. The results illustrate that incorporating entropy into routing offers a promising pathway to mitigate capsule networks' iterative voting limitations, setting the stage for future explorations into scalable capsule architectures that retain their distinctive ability to capture spatial hierarchies. This work not only advances the application of capsule networks in digital forensics but also provides a foundation for further research in optimizing routing mechanisms for complex multimedia detection tasks.

This entropy-integrated method offers distinct advantages over prior routing techniques, particularly in tasks requiring high interpretability and efficient handling of complex visual hierarchies. Furthermore, our results show that incorporating entropy enhances the robustness of capsule networks, contributing to improved feature representation and classification accuracy.

4.1 Limitations and Future Work

One key limitation of this study is the inherent challenge of scalability and computational efficiency in capsule networks. Unlike transformer models, which benefit from highly parallelized attention mechanisms, capsule networks rely on iterative routing processes to capture part-whole relationships—a sequential characteristic that complicates parallelization and impacts processing speed. This limitation has led to capsule networks receiving comparatively less attention than transformers in the broader research landscape. In future work, we aim to address these scalability constraints by exploring methods rooted in information theory to enable greater parallelization while preserving the routing mechanism's integrity. We hypothesize that adapting the routing process for more efficient computation could unlock further potential for capsule networks in high-performance applications like deepfake detection.

Our findings, along with previous studies, highlight both the computational limitations and future possibilities for capsule networks. For instance, Barham et al. [72] demonstrated that convolutional capsule models required four times fewer floating point operations (FLOPS) and sixteen times fewer parameters than traditional convolutional neural networks (CNNs). Despite these theoretical efficiencies, both TensorFlow [73] and PyTorch [74] implementations have encountered significant slowdowns and memory limitations, even in smaller capsule architectures. This disparity underscores the need for optimized computational frameworks that align with the unique demands of capsule-based models.

While several efficient capsule routing methods have been proposed [75, 76, 18, 31], challenges persist beyond the routing stage and extend to the capsule voting process. Barham et al. identified that current computational frameworks are optimized for standard operations commonly used in traditional model architectures, but these optimizations do not effectively support the distinct computations involved in capsule routing and voting. This leads to slower performance when handling these non-standard workloads, emphasizing the need for more flexible, efficient computational frameworks tailored to capsule networks.

Hooker et al. [77] also pointed out that capsule network operations can be executed relatively efficiently on CPUs, but performance suffers on hardware accelerators like GPUs and TPUs. These accelerators are optimized for typical deep learning workloads, limiting their efficiency for capsule network operations that require diverse computations. Addressing this hardware compatibility issue is another important direction for future research, as it would enable capsule networks to leverage the full potential of modern acceleration hardware.

Further research will also explore the application of entropy-integrated routing in other domains, such as natural language processing and multimodal data tasks, where interpretability and uncertainty handling are critical. Investigating novel approaches to optimize the routing and voting procedures could yield significant improvements in computational efficiency and scalability. Additionally, dynamically tuning entropy integration across multiple capsule layers may enhance the adaptability and robustness of capsule networks. These directions indicate that entropy integration could play a pivotal role in expanding the practical utility of capsule networks across a range of high-complexity applications.

References

[1] S. Sabour, N. Frosst, and G. E. Hinton, Advances in neural information processing systems **30**, (2017).

[2] I. Korshunova, W. Shi, J. Dambre, and L. Theis, in *Proceedings of the IEEE international conference on computer vision* (PUBLISHER, ADDRESS, 2017), pp. 3677–3685.

[3] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, IEEE transactions on image processing **28**, 5464 (2019).

[4] H. H. Nguyen, J. Yamagishi, and I. Echizen, in *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE (PUBLISHER, ADDRESS, 2019), pp. 2307–2311.

[5] Y. Bengio, Y. Lecun, and G. Hinton, Communications of the ACM **64**, 58 (2021).

[6] D. Bahdanau, arXiv preprint arXiv:1409.0473 (2014).

[7] S. Sukhbaatar, J. Weston, R. Fergus, *et al.*, Advances in neural information processing systems **28**, (2015).

[8] G. Hinton, Neural Computation **35**, 413 (2023).

[9] T. Cohen and M. Welling, in *International conference on machine learning*, PMLR (PUBLISHER, ADDRESS, 2016), pp. 2990–2999.

[10] G. E. Hinton, A. Krizhevsky, and S. D. Wang, in *Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21*, Springer (PUBLISHER, ADDRESS, 2011), pp. 44–51.

[11] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, arXiv preprint arXiv:1811.12231 (2018).

[12] G. Hinton, Cognitive Science **3**, 231 (1979).

[13] D. Kahneman, A. Treisman, and B. J. Gibbs, Cognitive psychology **24**, 175 (1992).

[14] G. E. Hinton, S. Sabour, and N. Frosst, in *International conference on learning representations* (PUBLISHER, ADDRESS, 2018).

[15] A. Kosiorek, S. Sabour, Y. W. Teh, and G. E. Hinton, Advances in neural information processing systems **32**, (2019).

[16] A. Roy, M. Saffar, A. Vaswani, and D. Grangier, Transactions of the Association for Computational Linguistics **9**, 53 (2021).

[17] F. De Sousa Ribeiro, G. Leontidis, and S. Kollias, Advances in neural information processing systems **33**, 6490 (2020).

[18] F. D. S. Ribeiro, G. Leontidis, and S. Kollias, in *Proceedings of the AAAI Conference on Artificial Intelligence* (PUBLISHER, ADDRESS, 2020), No. 04, pp. 3749–3756.

[19] S. R. Venkataraman, S. Balasubramanian, A. Anand, and R. R. Sarma, Sādhanā **49**, 215 (2024).

[20] R. Renzulli, E. Tartaglione, and M. Grangetto, arXiv preprint arXiv:2204.01298 (2022).

[21] J. Edstedt, Towards Understanding Capsule Networks, 2020.

[22] S. R. Venkatraman, A. Anand, S. Balasubramanian, and R. R. Sarma, arXiv preprint arXiv:2010.01488 (2020).

[23] K. Greff, S. Van Steenkiste, and J. Schmidhuber, arXiv preprint arXiv:2012.05208 (2020).

[24] J. Feldman, Cognitive neurodynamics **7**, 1 (2013).

[25] S. Löwe, P. Lippe, M. Rudolph, and M. Welling, arXiv preprint arXiv:2204.02075 (2022).

[26] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, arXiv preprint arXiv:2112.10741 (2021).

[27] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, in *International conference on machine learning*, Pmlr (PUBLISHER, ADDRESS, 2021), pp. 8821–8831.

[28] I. Paik, T. Kwak, and I. Kim, in *Asian Conference on Machine Learning*, PMLR (PUBLISHER, ADDRESS, 2019), pp. 489–502.

[29] J. Gu and V. Tresp, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (PUBLISHER, ADDRESS, 2020), pp. 7285–7293.

[30] D. Peer, S. Stabinger, and A. Rodriguez-Sanchez, arXiv preprint arXiv:1812.09707 (2018).

[31] Y.-H. H. Tsai, N. Srivastava, H. Goh, and R. Salakhutdinov, arXiv preprint arXiv:2002.04764 (2020).

[32] A. Malik, M. Kuribayashi, S. M. Abdullahi, and A. N. Khan, Ieee Access **10**, 18757 (2022).

[33] M. Masood, M. Nawaz, K. M. Malik, A. Javed, A. Irtaza, and H. Malik, Applied intelligence **53**, 3974 (2023).

[34] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, Information Fusion **64**, 131 (2020).

[35] Y. Zheng, J. Bao, D. Chen, M. Zeng, and F. Wen, in *Proceedings of the IEEE/CVF international conference on computer vision* (PUBLISHER, ADDRESS, 2021), pp. 15044–15054.

[36] A. Khormali and J.-S. Yuan, Applied Sciences **12**, 2953 (2022).

[37] X. Cao and N. Z. Gong, in *International Conference on Digital Forensics and Cyber Crime*, Springer (PUBLISHER, ADDRESS, 2021), pp. 360–378.

[38] F. Juefei-Xu, R. Wang, Y. Huang, Q. Guo, L. Ma, and Y. Liu, International journal of computer vision **130**, 1678 (2022).

[39] T. T. Nguyen, Q. V. H. Nguyen, D. T. Nguyen, D. T. Nguyen, T. Huynh-The, S. Nahavandi, T. T. Nguyen, Q.-V. Pham, and C. M. Nguyen, Computer Vision and Image Understanding **223**, 103525 (2022).

[40] L. A. Passos, D. Jodas, K. A. Costa, L. A. Souza Júnior, D. Rodrigues, J. Del Ser, D. Camacho, and J. P. Papa, Expert Systems **41**, e13570 (2024).

[41] Y. Mirsky and W. Lee, ACM computing surveys (CSUR) **54**, 1 (2021).

[42] L. Bode, D. Lees, and D. Golding, The digital face and deepfakes on screen, 2021.

[43] M. S. H. Mukta, J. Ahmad, M. A. K. Raiaan, S. Islam, S. Azam, M. E. Ali, and M. Jonkman, Journal of Sensor and Actuator Networks **12**, 61 (2023).

[44] A. Kaur, A. Noori Hoshyar, V. Saikrishna, S. Firmin, and F. Xia, Artificial Intelligence Review **57**, 1 (2024).

[45] F. Sun, N. Zhang, P. Xu, and Z. Song, Security and Communication Networks **2021**, 2482942 (2021).

[46] S. Kingra, N. Aggarwal, and N. Kaur, Information Sciences **645**, 119341 (2023).

[47] A. Firc, K. Malinka, and P. Hanáček, Heliyon **9**, (2023).

[48] I. U. Haq, K. M. Malik, and K. Muhammad, ACM Transactions on Multimedia Computing, Communications and Applications (2023).

[49] B. N. Jacobsen and J. Simpson, Information, communication & society **27**, 1095 (2024).

[50] Y. Li and S. Lyu, arXiv preprint arXiv:1811.00656 (2018).

[51] A. Haliassos, K. Vougioukas, S. Petridis, and M. Pantic, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (PUBLISHER, ADDRESS, 2021), pp. 5039–5049.

[52] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (PUBLISHER, ADDRESS, 2021), pp. 2185–2194.

[53] F. Xia, L. Akoglu, C. Aggarwal, and H. Liu, IEEE Intelligent Systems **38**, 32 (2023).

[54] L. Verdoliva, IEEE journal of selected topics in signal processing **14**, 910 (2020).

[55] L. Guarnera, O. Giudice, C. Nastasi, and S. Battiato, in *2020 AEIT international annual conference (AEIT)*, IEEE (PUBLISHER, ADDRESS, 2020), pp. 1–6.

[56] O. De Lima, S. Franklin, S. Basu, B. Karwoski, and A. George, arXiv preprint arXiv:2006.14749 (2020).

[57] H. Qi, Q. Guo, F. Juefei-Xu, X. Xie, L. Ma, W. Feng, Y. Liu, and J. Zhao, in *Proceedings of the 28th ACM international conference on multimedia* (PUBLISHER, ADDRESS, 2020), pp. 4318–4327.

[58] D. Güera and E. J. Delp, in *2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS)*, IEEE (PUBLISHER, ADDRESS, 2018), pp. 1–6.

[59] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha, in *Proceedings of the 28th ACM international conference on multimedia* (PUBLISHER, ADDRESS, 2020), pp. 2823–2832.

[60] T. Oorloff, S. Koppisetti, N. Bonettini, D. Solanki, B. Colman, Y. Yacoob, A. Shahriyari, and G. Bharaj, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (PUBLISHER, ADDRESS, 2024), pp. 27102–27112.

[61] A. Naitali, M. Ridouani, F. Salahdine, and N. Kaabouch, Computers **12**, 216 (2023).

[62] A. Luo, E. Li, Y. Liu, X. Kang, and Z. J. Wang, in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE (PUBLISHER, ADDRESS, 2021), pp. 6359–6363.

[63] M. Ouyang, R. K. Das, J. Yang, and H. Li, in *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, IEEE (PUBLISHER, ADDRESS, 2021), pp. 1–5.

[64] B. Ustubioglu, S. Dincer, A. Ustubioglu, and G. Ulutas, in *2024 47th International Conference on Telecommunications and Signal Processing (TSP)*, IEEE (PUBLISHER, ADDRESS, 2024), pp. 298–301.

[65] B. Zou, C. Cao, L. Wang, S. Fu, T. Qiao, and J. Sun, Computers & Security **137**, 103606 (2024).

[66] K. Lin, W. Han, S. Li, Z. Gu, H. Zhao, J. Ren, L. Zhu, and J. Lv, Cognitive Computation **15**, 13 (2023).

[67] A. Mehra, Master's thesis, University of Twente, 2020.

[68] S. S. Khalil, S. M. Youssef, and S. N. Saleh, in *2020 International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, IEEE (PUBLISHER, ADDRESS, 2021), pp. 1–6.

[69] H. H. Nguyen, J. Yamagishi, and I. Echizen, arXiv preprint arXiv:1910.12467 (2019).

[70] K. Venkatachalam, P. Trojovskỳ, and Š. Hubálovskỳ, PeerJ Computer Science **9**, e1313 (2023).

[71] K. Narayan, H. Agarwal, K. Thakral, S. Mittal, M. Vatsa, and R. Singh, in *2022 IEEE International Joint Conference on Biometrics (IJCB)*, IEEE (PUBLISHER, ADDRESS, 2022), pp. 1–10.

[72] P. Barham and M. Isard, in *Proceedings of the Workshop on Hot Topics in Operating Systems* (PUBLISHER, ADDRESS, 2019), pp. 177–183.

[73] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, in *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (PUBLISHER, ADDRESS, 2016), pp. 265–283.

[74] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, Advances in neural information processing systems **32**, (2019).

[75] K. Ahmed and L. Torresani, Advances in neural information processing systems **32**, (2019).

[76] T. Hahn, M. Pyeon, and G. Kim, Advances in neural information processing systems **32**, (2019).

[77] S. Hooker, Communications of the ACM **64**, 58 (2021).