Augmenting in the Latent Space and Hybrid Convolutional Transformers for Open Radio Access Networks

Pranshav Gajjar, Anushri Bhansali, and Sounak Dutta

Abstract

Open Radio Access Networks or O-RAN allow interoperation and do not mandate vendor-specific hardware signaling substantial advancements in wireless communications. To achieve this O-RAN leverages the concept of RAN Intelligent Controllers or RICs which creates a multi-tier architecture that operates on different latency constraints and leverages different xApps (eXtended application) as a software application in the Near Real Time RIC. These xApps usually leverage sophisticated deep learning architectures which are responsible for different aspects of RAN control and are expected to be robust to channel changes. It is also extremely difficult to collect large datasets required for training such models and this is an important bottleneck that needs to be addressed. We have literature that leverages GANs, however, for xApps that work on spectrograms instead of KPMs (Key Performance Metrics), these models would succumb to the mode collapse problem due to insufficient data. Hence we aim to explore Embedding augmentation approaches like E-Mixup and E-Sticthup which would be a cost-effective way to generate new data while addressing the current limitations. It is also important to note that ML training can happen on the Non-Real Time RIC, and for continuous deployment, having a sufficiently fast training pipeline is desired. We also aim to address the Data Efficiency of different convolutional (Popular CNNs), vision transformers (ViT), and Hybrid architectures (ConvNext) to see what methodology can obtain a superlative performance with the least data requirement. We benchmark the proposed study and applicability of embedding augmentation on a publicly available dataset by the NextG Lab@NCState for the Interference Classification xApp, which primarily functions to detect a jammer through spectrogram data in a Shared Data Layer.

Index Terms

O-RAN, Data Efficiency, Embedding Augmentation, ViT, Hybrid Transformers, xApps, E-Mixup, E-Sticthup.

I. Introduction

The domain of wireless communications has seen a lot of advancements due to the field of Machine Learning, as tasks that were difficult to compute or required deterministic systems have been replaced by ML or ML-based algorithms [1]. One such avenue in wireless that has facilitated the rapid growth of applied ML is Open Radio Access Networks, which aim to incorporate interoperability and reduce or eliminate the vendor-specific requirement in radio access networks [2]. O-RAN leverages a shared database that is queried or leveraged by different ML applications of RAN [2]. These applications are usually located in the Near Real Time RAN Intelligent Controller called the **NearRT-RIC**, as an extended Application or xApps. It is an extremely difficult task to create these ML-based xApps, as to deploy one in a real-world scenario the data has to be collected through an Over-the-Air testbed, and annotating the data and creating such diverse scenarios that can be encountered daily is very difficult due to the shear complexity of telecom based applications [3]. So, the modality of ML training happens in a data-scarce scenario, and even for online learning-based systems that are nested in the **Non-Real Time RIC** or the NonRT-RIC, the resources are scarce and it is important to generate a model that is compatible with the channel changes and robust enough in the new environment or accurate enough for the task which necessitated the online training update [3].

Hence, multiple motivations in O-RAN prompt us to create an ML framework or training pipeline that can provide accurate ML models with as little data as possible. One way to look at this is through the lens of data augmentation. As the majority of xApps leverage Spectrograms or a spatio-temporal representation of the network traffic, we get to work with images or model this as a compute vision problem [4] [5]. Now, standard augmentation techniques like rotation and image flipping do not work, as we do not have access to natural images, but spectrogram data from the RIC database. So, the literature points towards leveraging a Generative Adversarial Network [6], which can generate high-fidelity images but also require a certain amount of data to converge and not succumb to the Mode Collapse problem [7].

So, by addressing the limitations and the literature, our work proposes the use of Embedding Augmentation, or augmenting in the latent space instead of directly working with spectrograms, to maintain a computationally effective experience and also have a data-efficient ML pipeline. We leverage **E-Mixup and E-Sticthup** [8] and intensively experiment with different vision models from the literature including Convolutional Models [9], Vision Transformers [10], and Hybrid architectures [11], to effectively model the data requirements for the said approaches, and to thoroughly assess the effectiveness of embedding augmentation. We prototype the approach with an Interference Classification xApp [12] which is implemented on an **Over-the-Air** testbed to obtain perceptible results for a real-world scenario. The paper is further divided into the Related Works, the proposed Methodology, the Results, and the Concluding statement.

II. Related Works

Recent advancements in machine learning have underscored the importance of data augmentation techniques, particularly in scenarios with limited training data. While Generative Adversarial Networks (GANs) have shown promise in synthetic data generation, they face significant challenges that have prompted researchers to explore alternative approaches. Ko et al. [13] introduced Embedding Expansion, a novel technique for augmenting data directly in the embedding space. This method generates synthetic points by combining feature representations, preserving the underlying data structure while enhancing efficiency. Their approach integrates seamlessly with existing metric learning losses without affecting model size, training speed, or optimization complexity. Wolfe et al. [8] proposed four innovative data augmentation techniques applicable to embedding inputs, demonstrating their effectiveness in both Natural Language Processing and Computer Vision domains. Their methods aim to improve downstream model performance by enhancing the quality and diversity of pre-trained embeddings. Liu et al. [14] explored latent space interpolation for data augmentation, demonstrating improved sample diversity by navigating the continuous space between existing data points in latent representations. This approach offers a nuanced method for expanding dataset diversity without relying on traditional GAN-based generation.Inoue et al. [15] investigated sample pairing for image classification, showcasing how combining existing samples can create new training instances. This method complements embedding-based techniques by expanding available datasets without generating entirely synthetic data. Zhang et al. [16] conducted an in-depth analysis of GAN convergence and mode collapse, revealing inherent instabilities in the training process. Their work highlighted the tendency of GANs to produce limited diversity in generated samples, a phenomenon known as mode collapse. Building on this research, Kushwaha et al. [17] explored strategies to mitigate mode collapse in GANs, proposing novel techniques to enhance the stability and diversity of generated outputs. Bau et al.[18] developed visualization techniques to identify limitations in GAN-generated outputs, providing insights into areas where alternative augmentation methods might be more suitable. Complementing this work, Mi et al. [19] conducted a comparative analysis of GANs and Variational Autoencoders (VAEs), offering a nuanced understanding of their respective strengths and weaknesses in data generation and augmentation tasks. Khanuja et al. [20] provided a comprehensive review of challenges facing GANs and proposed potential solutions, synthesizing recent advancements in the field. Their work aligns with the growing interest in embedding-based augmentation techniques as viable alternatives to traditional GAN-based methods. In conclusion, while GANs offer substantial capabilities for generating synthetic data, their susceptibility to mode collapse and other issues necessitates the exploration of alternative augmentation strategies. The advancements in embedding-based techniques, particularly those focusing on augmentation in embedding spaces and transfer learning, present promising avenues for addressing these challenges. These methods not only improve model performance in data-scarce environments but also offer more stable and diverse augmentation options compared to traditional GAN-based approaches.

III. Methodology

This section explains the proposed method in detail, the entire approach can be perceived in Figure 1. We show how the latent embeddings are obtained and a spectrogram is subjected to the feature extractor layers of a vision model and subjected to an embedding augmentation approach to obtain newer data samples. This section is divided into multiple subsections explaining O-RAN, the tested Computer Vision Models, Embedding Augmentation approaches, visualization techniques, and the dataset.

A. O-RAN (IC xApp)

The Open Radio Access Network (O-RAN) architecture [12] introduces a highly modular and interoperable approach to managing Radio Access Networks by disaggregating traditional RAN components into open and standardized functional blocks. One of the most critical elements here is the Near-Real-Time RAN Intelligent Controller (Near-RT RIC) [12], which plays a key role in hosting essential components. The Near-RT RIC [12] accommodates third-party applications developed by vendors, referred to as xApps [12]. These xApps operate as intelligent modules, leveraging machine learning algorithms to define control policies aimed at optimizing RAN performance via the E2 interface [12]. In addition to hosting xApps, the Near-RT RIC also contains a RIC database [12], which acts as a centralized repository for data used across the O-RAN system. This database ensures efficient operation and coordination among the system's components. The RIC database [12] serves as a centralized storage unit for network-related data. The database in our work is housing spectrograms, providing the basis for generating insights about the network's overall behavior. The RIC database's [12] role extends beyond data storage; it facilitates collaborative operations within the O-RAN [12] system by allowing multiple xApps to access and utilize its information. This shared access ensures that xApps can make informed decisions and collectively enhance the network's performance. The central importance of the database in maintaining a dynamic and adaptive open RAN system underscores the need to safeguard its data against potential malicious manipulations [12]. Now, the machine learning-driven Interference Classification xApp [12] is specifically engineered to detect and address interference caused by jamming devices transmitting disruptive signals. As illustrated in Fig. 2, this xApp [12] utilizes spectrograms stored in the RIC database [12] as its primary input to identify interference types and make real-time decisions about their presence within the network. Once interference is detected, the xApp sends a control message to the RAN via the E2-Lite interface [12], triggering necessary actions to optimize network performance. A data processing microservice is responsible for converting raw I/Q samples into spectrograms. The machine

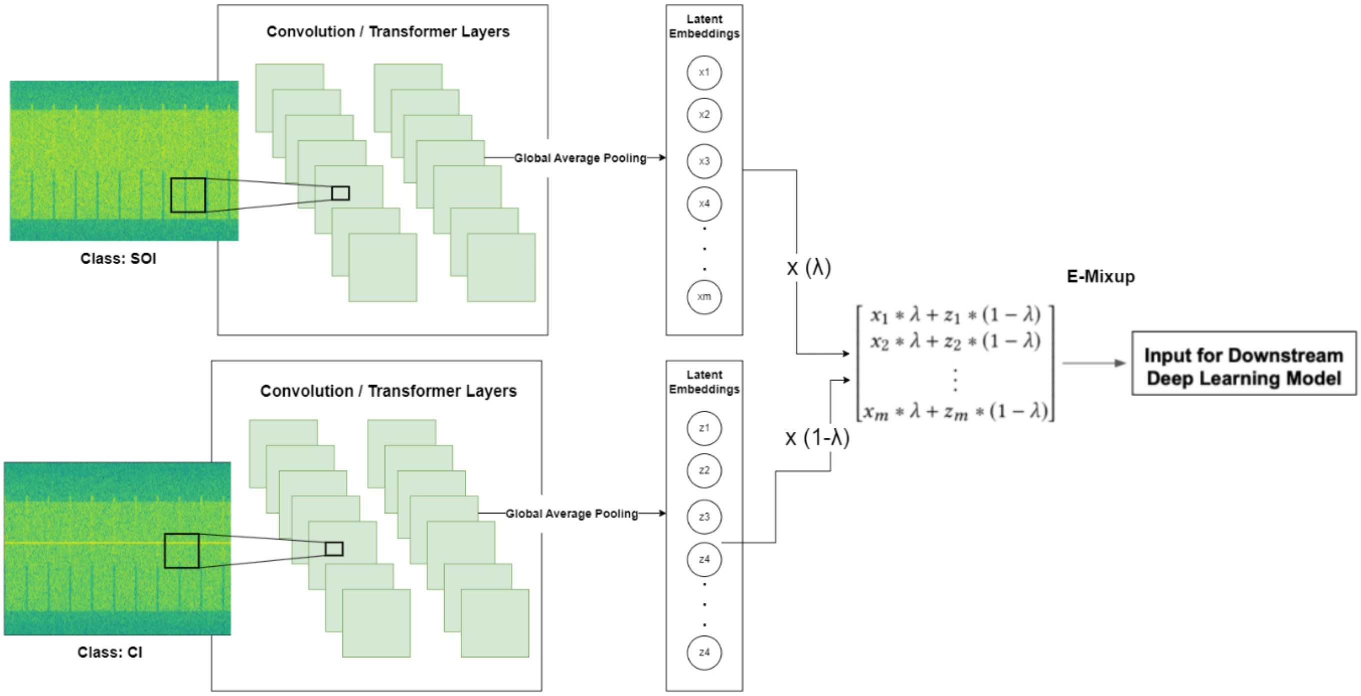

Convolution / Transformer Layers
Class: SOI
Global Average Pooling
Latent Embeddings
x1
x2
x3
x4
xm
x (λ)
E-Mixup
$x_1 * \lambda + z_1 * (1-\lambda)$
$x_2 * \lambda + z_2 * (1-\lambda)$
$x_m * \lambda + z_m * (1-\lambda)$
Input for Downstream Deep Learning Model
x (1-λ)
Convolution / Transformer Layers
Class: CI
Global Average Pooling
Latent Embeddings
z1
z2
z3
z4
z4

Fig. 1: An overview of the proposed method with E-Mixup and the obtained spectrograms from an OTA testbed.

learning models integrated within the xApp are designed to ensure high accuracy in interference classification, as better models result in improved decisions for RAN control and overall network efficiency [12].

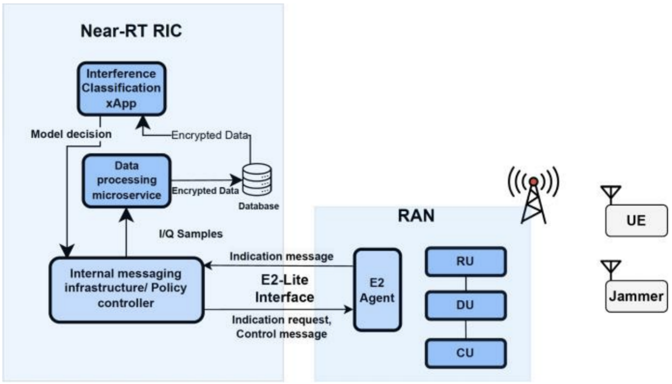

Near-RT RIC
Interference Classification xApp
Model decision
Encrypted Data
Data processing microservice
Encrypted Data
Database
I/Q Samples
Indication message
E2-Lite Interface
Internal messaging infrastructure/ Policy controller
Indication request, Control message
E2 Agent
RAN
RU
DU
CU
UE
Jammer

Fig. 2: An overview of the O-RAN architecture emphasizing the capabilities of a machine learning-driven interference classification xApp.

[12]

B. Computer Vision Models

Our work compares state-of-the-art computer vision models to work within the IC xApp in data-scarce scenarios. These models include Convolutional Neural Networks (CNNs), ResNet, DenseNet, MobileNetV2, Vision Transformer (ViT), and ConvNeXt. The following descriptions offer deeper insights as to why we have chosen the set of models for our comparative analysis.

1) CNNs: Providing a powerful architecture for processing image datasets, CNN architectures prove to be cornerstone models for our work. These comprise convolution layers that help with feature extraction. Pooling layers are also present, which help with dimensionality reduction [21]. Finally, the fully connected layers for classification. This composition helps to capture global and local data patterns and to understand complex structures. The early layers of the CNN would mainly focus on local features, such as edges or textures, which are the fine details of the image or spectrogram. Deeper layers of the CNN combine these local features to form higher-level representations, such as the overall structure of the spectrogram. CNN therefore seems to be suitable for image data like spectrograms because of their ability to learn the spatial hierarchies of features through backpropagation [21].

- **ResNet**: We used ResNet50, one of the variants of the Residual Network Architecture, introduced by [22]. One of the major features includes the use of skip connections, which allow the network to skip some layers allowing it to transfer information from earlier layers to later layers. In addition to this, the network makes use of residual functions and bottleneck design supporting parameter efficiency and making it easier to optimize and train deeper architectures. Furthermore, the use of identity mapping also facilitates the network to learn when and when not to apply transformations as required. All of these together mitigate the vanishing gradient problem and hence it appears to be a useful candidate architecture for our study.
- **DensNet**: The DenseNet architecture proposed by [23], where each layer is directly connected to every other layer comprises short paths from the earlier layers to later ones. This kind of connection facilitates the layer to access the feature maps of all preceding layers, promoting feature propagation and its reuse along with improving the gradient flow. DenseNet introduces a growth rate parameter, which controls how much new information each layer contributes to the global state. The dense connectivity pattern reduces the number of parameters while improving model performance, making it particularly suitable for complex spectrogram analysis tasks.
- **MobileNetV2**: MobileNetV2 is a highly efficient architecture designed for resource-constrained environments, making it ideal for real-time applications in O-RAN, where computational resources are limited. The architecture introduced by [24] incorporates several key innovations to optimize performance. The inverted residual structure first expands the input channels, applies a lightweight depthwise convolution, and then projects the output to a lower-dimensional representation, improving efficiency. MobileNetV2 also utilizes depthwise separable convolutions, which split standard convolutions into depthwise and pointwise operations, significantly reducing the number of parameters and computations. The inclusion of linear bottlenecks removes non-linearities in the narrow layers, helping to preserve crucial information while minimizing unnecessary complexity. Shortcut connections between bottlenecks ensure smooth information flow throughout the network, preventing performance degradation. Additionally, MobileNetV2 makes efficient use of channels by employing smaller input and output dimensions compared to its predecessors, further optimizing resource usage. These combined features allow MobileNetV2 to achieve a strong balance between accuracy and efficiency.

*2) **ViT**:* The Vision Transformer (ViT), put forward by adapts the transformer architecture, initially developed for natural language processing, to handle image recognition tasks [25]. It divides images into fixed-size patches, treating each patch as a distinct "word." These patches are flattened into vectors and passed through a linear layer to create embeddings, with positional encodings added to retain spatial context. The sequence of embeddings, including a special classification token (CLS), is processed by a transformer encoder. Within the encoder, multi-head self-attention captures global relationships between patches, while feedforward layers refine feature representation. Stability during training is enhanced through techniques like layer normalization and residual connections. The output associated with the CLS token is used to make the final classification. ViT offers significant advantages over CNNs, particularly in its ability to model long-range dependencies across an entire image using self-attention, providing a more comprehensive global perspective than the localized focus of CNNs.

*3) **ConvNext**:* ConvNeXt, introduced by [26] is a modern CNN architecture inspired by transformer design principles, combining efficiency with advanced modeling capabilities. It utilizes depthwise convolutions to optimize computations and adopts a hierarchical structure to process multi-scale representations. Features like larger kernel sizes improve spatial dependency capture, while layer normalization enhances training stability and convergence. ConvNeXt also aligns with transformers in its stage compute ratio, ensuring balanced resource allocation across layers, making it highly effective for complex visual tasks.

C. Embedding Augmentation

By generating synthetic data points in the embedding space, embedding augmentation seeks to remedy the current drawback, and our work leverages two main techniques from the paper [8] which is explained below.

*1) **E-Mixup**:* Put forward by [8] E-mixup is an advanced data augmentation technique that extends the principles of Mixup to the embedding space, in contrast to traditional Mixup which operates on raw input data. This method creates new training samples by interpolating between existing embeddings, improving data diversity, and potentially improving model performance.

The E-Mixup process involves several key steps. First, the model processes input samples to generate their corresponding embeddings. Then, two different samples are randomly selected from the training set. A mixing coefficient λ is sampled from a Beta distribution, typically Beta(α, α), where α is a hyperparameter that controls the strength of interpolation. A new embedding is created by computing a weighted average of the two selected embeddings:

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j \quad [8] \tag{1}$$

where x_i and x_j are the embeddings of the two samples. Similarly, a new soft label is created by interpolating between the original labels:

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j \quad [8] \tag{2}$$

where y_i and y_j are the corresponding labels.

*2) **E-Stitchup:*** E-Stitchup is an innovative data augmentation technique designed for enhancing pre-trained embeddings in deep learning models. This method, introduced by [8], operates directly on embedding inputs by creating new synthetic data points through the combination of existing embedding vectors, effectively expanding the dataset without altering the underlying model architecture. The technique involves selecting pairs of embeddings from the same class and generating new samples by interpolating between them thus preserving class-specific information while introducing variability. By augmenting the embedding space, E-Stitchup aims to improve the generalization capabilities of downstream models, particularly in scenarios with limited training data. The operation of E-Stitchup follows the mentioned steps.

First, the mixing coefficient λ is sampled from a Beta distribution, $\text{Beta}(\alpha, \alpha)$. For each index k in the embedding vector, a Bernoulli random variable $z[k]$ is generated, determining whether the k-th element of the augmented embedding is taken from the first or the second input embedding. The augmented embedding $\tilde{x}$ is defined as:

$$\tilde{x}[k] = \begin{cases} x_i[k], & \text{if } z[k] = 1, \\ x_j[k], & \text{if } z[k] = 0, \end{cases} \quad [8] \tag{3}$$

where $P(z[k] = 1) = \lambda$ and $P(z[k] = 0) = 1 - \lambda$. Here, $x_i[k]$ and $x_j[k]$ represent the k-th elements of the embeddings x_i and x_j, respectively.

The label augmentation process is consistent with E-Mixup and involves computing a weighted average of the corresponding labels as:

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j, \quad [8] \tag{4}$$

where y_i and y_j are the labels associated with x_i and x_j.

E-Stitchup has demonstrated promising results in enhancing model performance across various tasks, showcasing its potential as a versatile tool for improving the robustness and effectiveness of deep learning models in resource-constrained environments.

D. tSNE

t-SNE (t-distributed Stochastic Neighbor Embedding) is a nonlinear dimensionality reduction technique designed to map high-dimensional data into a lower-dimensional space, typically two or three dimensions [27]. Excels at visualizing complex datasets by preserving local structures and revealing global patterns. The process involves two main stages. First, it calculates pairwise similarities between high-dimensional points, assigning higher probabilities to similar pairs using a Gaussian distribution. Then, it maps the data into lower dimensions. This is done by minimizing the ”Kullback-Leibler divergence” between the high-dimensional and low-dimensional probability distributions, ensuring similar points remain close in the visual space.

In our study, t-SNE was applied to embeddings generated by computer vision models, and by reducing the dimensionality of feature representations, we visualized how effectively each model distinguishes between classes within our dataset. This approach provided valuable insight into the performance of the model and highlighted areas for improvement in the capture of distinct categories.

E. Dataset

From the IC xApp and O-RAN architecture that was explained before we source a total of 2100 spectrograms across three equally distributed classes. The first class represents the uplink UE signal with no interference which we call signal of interest (SOI). These SOI are transmitted at an uplink carrier frequency of 2.56 GHz [12]. The second class and third class used for training data represent scenarios with interference, specifically continuous wave interference (CWI) and chirped interference (CI) [12]. These interference signals were generated at various gain values ranging from 30 dB to 40 dB. We also generate an analogous dataset with Speckle noise [28] to simulate a noisier network that can be encountered in a real-world scenario to model a harder benchmark and see the performance of the aforementioned models on a more difficult dataset. Sample spectrograms about these classes and the analogous noisy distributions are mentioned in Figure 3.

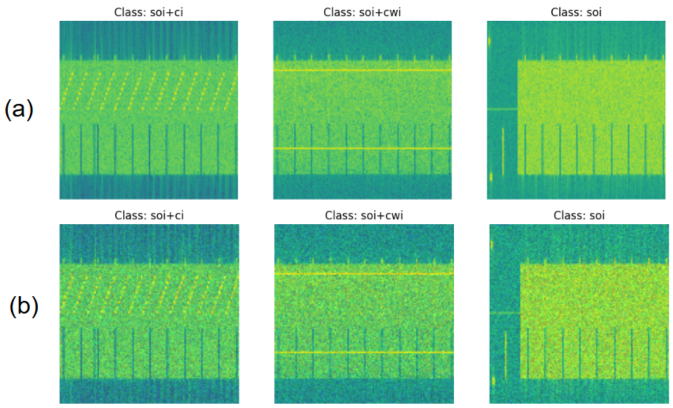


Fig. 3: The row (a) shows the natural images, and the row (b) shows examples with speckle noise.

IV. Results and Experiments

Model	Accuracy	Precision	Recall	F1-Score
ConvNeXTiny	0.32667	0.10671	0.32667	0.16087
ConvNeXTiny + EMixup	0.33238	0.11048	0.33238	0.16583
ConvNeXTiny + EStichup	0.32667	0.10671	0.32667	0.16087
ViT	0.32667	0.10671	0.32667	0.16087
ViT + EMixup	0.32667	0.10671	0.32667	0.16087
ViT + EStichup	0.32571	0.10609	0.32571	0.16005
MobileNet	0.82381	0.85282	0.82381	0.82099
MobileNet + EMixup	0.94667	0.94964	0.94667	0.94600
MobileNet + EStichup	0.94667	0.94942	0.94667	0.94669
ResNet	0.94667	0.95283	0.94667	0.94618
ResNet + EMixup	**0.95714**	**0.95737**	**0.95714**	**0.95701**
ResNet + EStichup	0.95238	0.95291	0.95238	0.95227
DenseNet	0.91238	0.92335	0.91238	0.91135
DenseNet + EMixup	0.94952	0.95552	0.94952	0.94949
DenseNet + EStichup	0.93714	0.94167	0.93714	0.93609

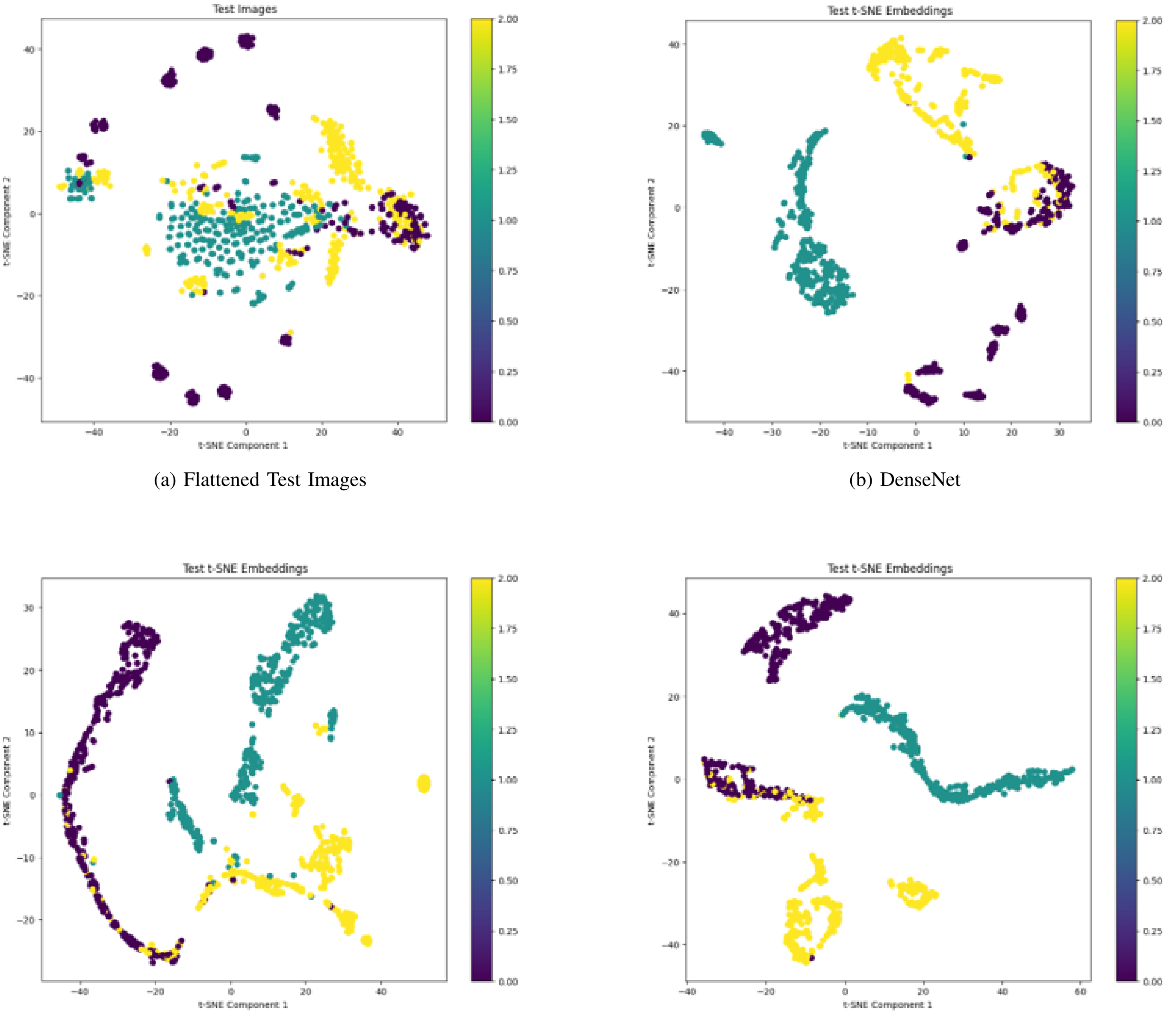
TABLE I: Results For Noise Free Data

The table 1 presents the performance evaluation of various machine learning models on noise-free data for a spectrogram-based classification task. It compares ConvNeXTiny, Vision Transformers (ViT), MobileNet, ResNet, and DenseNet, both with and without augmentation techniques like E-Mixup and E-Stitchup. The results indicate that ConvNeXTiny and ViT show poor performance, with minimal improvements even after augmentation, likely due to underfitting. In contrast, CNN-based models like MobileNet, ResNet, and DenseNet demonstrate significantly higher accuracy and F1-scores, especially when augmented. ResNet with E-Mixup achieves the highest performance (Accuracy = 95.71 percent, F1-Score = 0.957), showcasing the effectiveness of embedding-level augmentations in improving model robustness and generalization under clean data conditions. These results underscore the superiority of CNN architectures combined with augmentation techniques in data-efficient scenarios.

Model	Accuracy	Precision	Recall	F1-Score
ConvNeXTiny	0.32267	0.10411	0.32267	0.15743
ConvNeXTiny + EMixup	0.32267	0.10411	0.32267	0.15743
ConvNeXTiny + EStichup	0.32267	0.10411	0.32267	0.15743
ViT	0.34267	0.11742	0.34267	0.17491
ViT + EMixup	0.32267	0.10411	0.32267	0.15743
ViT + EStichup	0.32267	0.10411	0.32267	0.15743
MobileNet	0.60381	0.71344	0.60381	0.49743
MobileNet + EMixup	**0.92286**	**0.92501**	**0.92286**	**0.92264**
MobileNet + EStichup	0.90762	0.90763	0.90762	0.90762
ResNet	0.85619	0.89343	0.90762	0.85901
ResNet + EMixup	0.91238	0.92731	0.91238	0.91266
ResNet + EStichup	0.91333	0.92863	0.91333	0.91388
DenseNet	0.82190	0.88281	0.82190	0.82383
DenseNet + EMixup	0.87714	0.89578	0.87714	0.87771
DenseNet + EStichup	0.85238	0.89216	0.85238	0.85202

TABLE II: Speckle Noise Results

The table II evaluates the performance of models on speckle noise data for the spectrogram-based classification task. It compares the effectiveness of models like ConvNeXTiny, Vision Transformers (ViT), MobileNet, ResNet, and DenseNet, with and without latent-space augmentation techniques such as E-Mixup and E-Stitchup. The results highlight that ConvNeXTiny and ViT perform poorly under speckle noise conditions, showing no significant gains from augmentations, likely due to underfitting. In contrast, CNN-based models like MobileNet, ResNet, and DenseNet demonstrate substantial improvements when augmented, with MobileNet + E-Mixup achieving the best performance (Accuracy = 92.28 percent, F1-Score = 0.922). These findings underscore the robustness of CNN architectures with embedding-level augmentations for handling noisy data scenarios, emphasizing their utility in data-efficient, real-world applications.

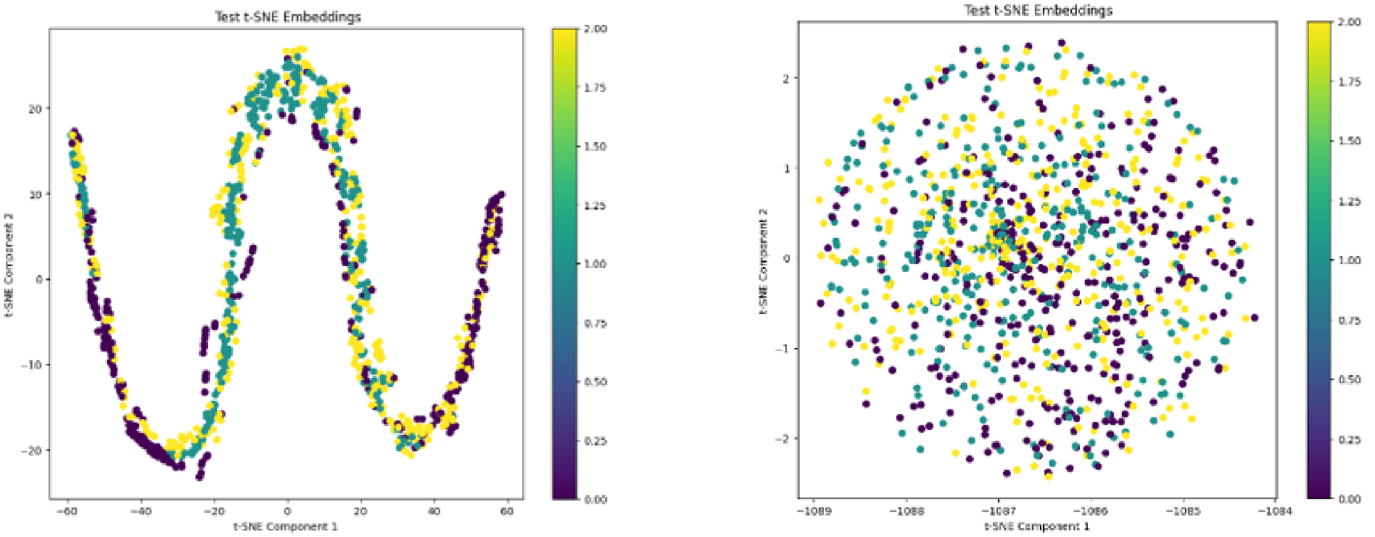


(a) Flattened Test Images

(b) DenseNet

(c) MobileNetV2

(d) ResNet

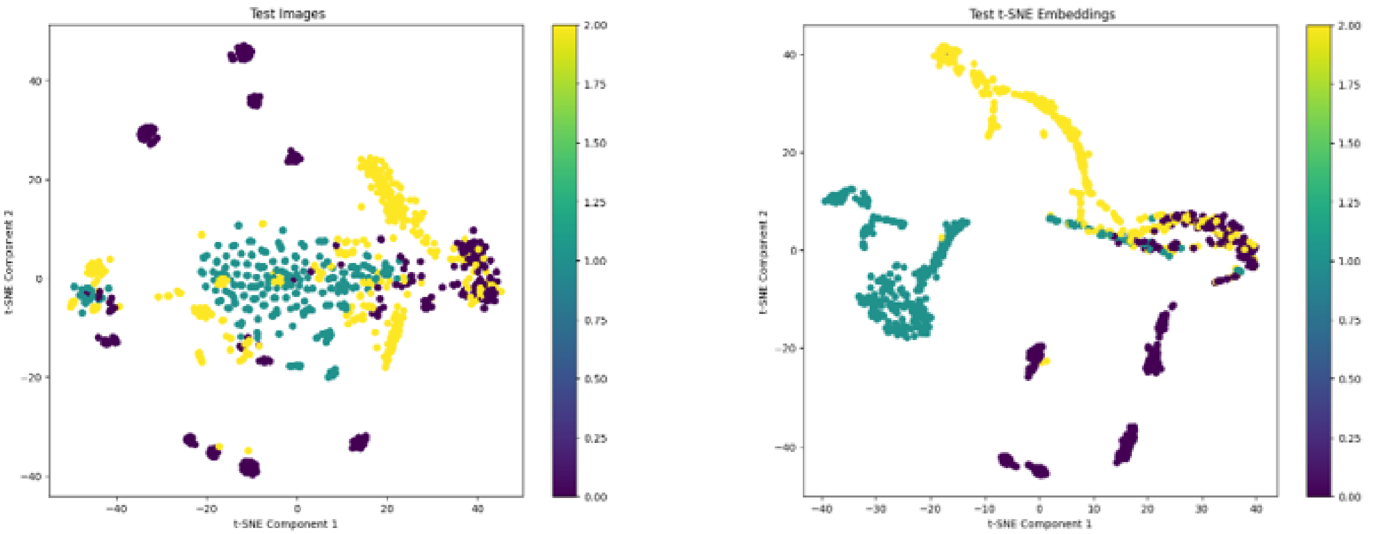


(e) ViT

(f) ConvNext

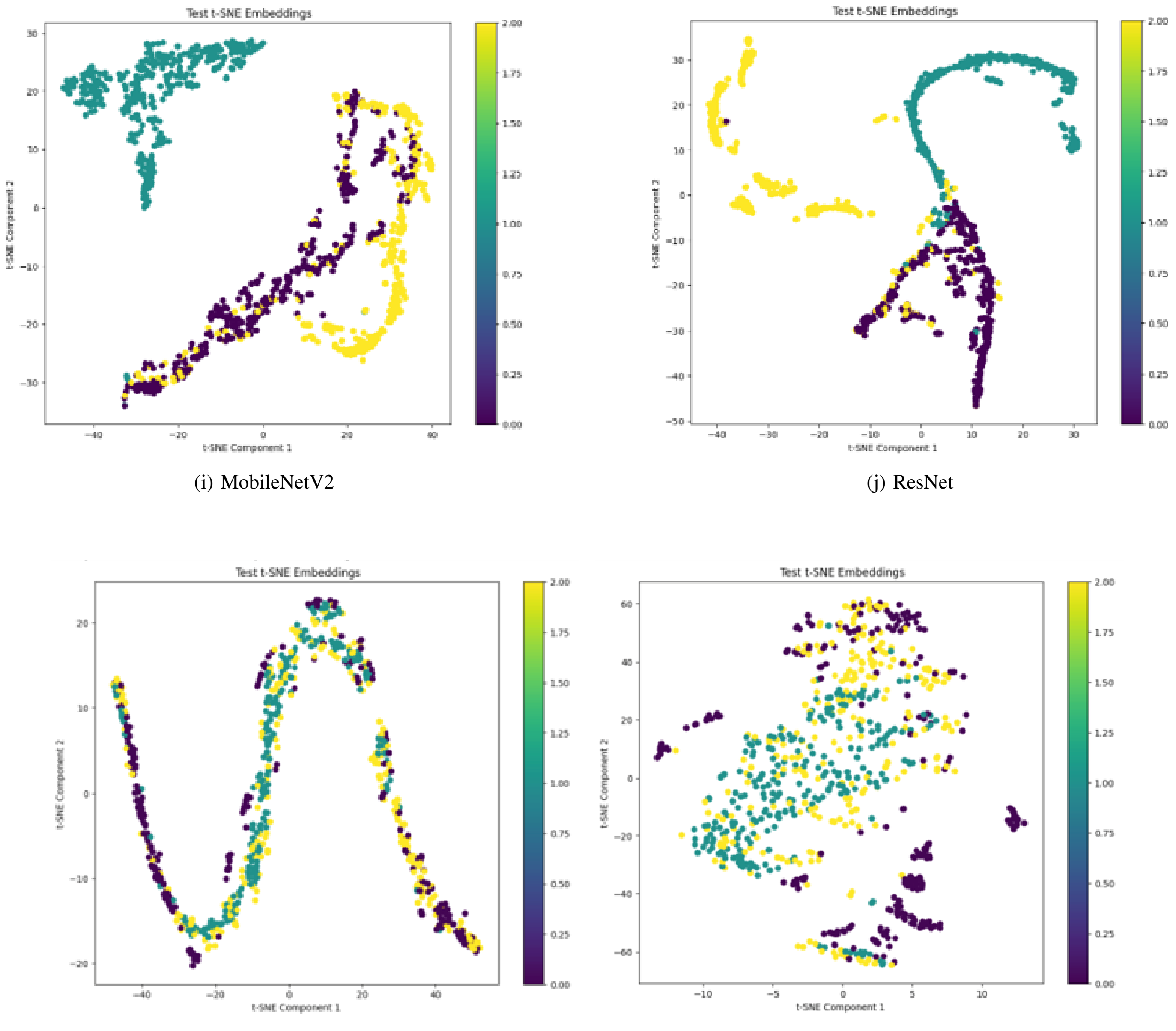
Fig. 4: t-SNE Visualizations For Noise Free

Fig. 4 depicts t-SNE visualizations of the spectrogram data under noise-free conditions, showcasing how different models separate the three data classes: Signal of Interest (SOI), Continuous Wave Interference (CWI), and Chirped Interference (CI). Models like ConvNeXT and Vision Transformers (ViT) exhibit overlapping clusters, indicating poor feature extraction and limited separability between the classes, which is likely due to underfitting or inadequate learning of class-specific features. On the other hand, models such as MobileNet, ResNet, and DenseNet demonstrate tighter and more distinct clusters, reflecting their ability to extract meaningful features and achieve better class separability. The difference in patterns arises from the architectural advantages of CNN-based models in capturing spatial hierarchies, compared to ConvNeXT and ViT, which struggle in data-efficient and noise-free scenarios. These visualizations highlight the varying effectiveness of the models in learning robust representations of clean spectrogram data.



(g) Flattened Test Images

(h) DenseNet



(i) MobileNetV2

(j) ResNet

(k) ViT

(l) ConvNext

Fig. 5: t-SNE Visualizations for Speckle Noise

Fig. 5 depict t-SNE visualizations of spectrogram data under speckle noise conditions for various models, illustrating the effects of noise on the separation of data classes: Signal of Interest (SOI), Continuous Wave Interference (CWI), and Chirped Interference (CI). The patterns reveal significant differences in the ability of models to separate these classes in the noisy feature space. Models such as ConvNeXT and Vision Transformers (ViT) show overlapping clusters, indicating poor feature extraction and underfitting under speckle noise. Conversely, CNN-based models like MobileNet, ResNet, and DenseNet demonstrate better separation of classes, though the clusters are more dispersed compared to noise-free conditions. This dispersion highlights the challenge posed by noise, which disrupts the latent space structure, reducing class separability. These visualizations emphasize the robustness of CNN architectures over hybrid or transformer models in handling noisy environments and the importance of augmentation techniques to mitigate the impact of noise on feature learning.

Model	Parameters (Million)
MobileNet	2,418,739
ResNet	23,844,467
DenseNet	7,166,259
ViT	2,925,779
ConvNeXt	27,916,883

TABLE III: Model vs. Parameters

TABLE III showcases the number of parameters across five different models, highlighting their architectural complexity. The table lists the models alongside their corresponding parameter counts. As we can see ResNet has the highest parameter count at approximately 23.8 million, indicating its significant complexity and potential for higher representational capacity. ConvNeXt follows with around 27.9 million parameters, positioning itself as a hybrid convolutional-transformer model with considerable computational requirements. DenseNet has a relatively moderate parameter count of about 7.16 million, emphasizing its efficient design for feature reuse. ViT (Vision Transformer), with approximately 2.93 million parameters, reflects its lightweight architecture compared to traditional convolution-based models. Lastly, MobileNet is the smallest model, with only 2.41 million parameters, underscoring its focus on computational efficiency and deployment in resource-constrained environments.

V. Conclusion and Future Work

Our work aimed to explore robust and data-efficient approaches for image classification tasks under challenging conditions of limited or noisy data, and our research demonstrates that CNNs, particularly when combined with latent space augmentation techniques such as E-Mixup and E-Stitchup, offer a highly effective solution. These methods substantially improved the performance of CNNs on noisy data, with MobileNet showing the most significant improvement, from 60.381 percent to 92.286 percent accuracy with E-Mixup.

The poor performance of transformer-based models on noisy data highlights potential limitations in their current architectures for handling such datasets, especially when data is limited. While transformer-based models show promise in other areas, they currently lag behind CNNs in these specific scenarios, underscoring the need for further research and development to enhance their performance in such conditions.

For future work, we aim to incorporate testbed experiments to validate the performance more accurately, and even test how these algorithms or approaches pan out when we use online learning-based methods that are crucial for the NonRT-RIC and the overall functioning of the entire O-RAN system.

References

[1] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, “Application of machine learning in wireless networks: Key techniques and open issues,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3072–3108, 2019.

[2] J. S. Vardakas, K. Ramantas, E. Vinogradov, M. A. Rahman, A. Girycki, S. Pollin, S. Pryor, P. Chanclou, and C. Verikoukis, “Machine learning-based cell-free support in the o-ran architecture: an innovative converged optical-wireless solution toward 6g networks,” *IEEE Wireless Communications*, vol. 29, no. 5, pp. 20–26, 2022.

[3] M. Polese, L. Bonati, S. D’oro, S. Basagni, and T. Melodia, “Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges,” *IEEE Communications Surveys & Tutorials*, 2023.

[4] T. Jian, B. C. Rendon, E. Ojuba, N. Soltani, Z. Wang, K. Sankhe, A. Gritsenko, J. Dy, K. Chowdhury, and S. Ioannidis, “Deep learning for rf fingerprinting: A massive experimental study,” *IEEE Internet of Things Magazine*, vol. 3, no. 1, pp. 50–57, 2020.

[5] D. Kurmantayev, D. Kwun, H. Kim, and S. W. Yoon, “Risi: Spectro-temporal ran-agnostic modulation identification for ofdma signals,” *arXiv preprint arXiv:2211.12287*, 2022.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[7] K. Zhang, “On mode collapse in generative adversarial networks,” in *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part II 30.* Springer, 2021, pp. 563–574.

[8] C. R. Wolfe and K. T. Lundgaard, “Data augmentation for deep transfer learning,” *arXiv preprint arXiv:1912.00772*, 2019.

[9] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A survey of convolutional neural networks: analysis, applications, and prospects,” *IEEE transactions on neural networks and learning systems*, 2021.

[10] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu *et al.*, “A survey on vision transformer,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 87–110, 2022.

[11] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, “Convnext v2: Co-designing and scaling convnets with masked autoencoders,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 133–16 142.

[12] P. Gajjar, A. Chiejina, and V. K. Shah, “Preserving data privacy for ml-driven applications in open radio access networks,” *arXiv preprint arXiv:2402.09710*, 2024.

[13] B. Ko and G. Gu, “Embedding expansion: Augmentation in embedding space for deep metric learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7255–7264.

[14] X. Liu, Y. Zou, L. Kong, Z. Diao, J. Yan, J. Wang, S. Li, P. Jia, and J. You, “Data augmentation via latent space interpolation for image classification,” in *2018 24th International Conference on Pattern Recognition (ICPR).* IEEE, 2018, pp. 728–733.

[15] H. Inoue, “Data augmentation by pairing samples for images classification,” 2018. [Online]. Available: https://arxiv.org/abs/1801.02929

[16] Z. Zhang, M. Li, and J. Yu, “On the convergence and mode collapse of gan,” in *SIGGRAPH Asia 2018 Technical Briefs*, 2018, pp. 1–4.

[17] V. Kushwaha, G. Nandi *et al.*, “Study of prevention of mode collapse in generative adversarial network (gan),” in *2020 IEEE 4th Conference on Information & Communication Technology (CICT).* IEEE, 2020, pp. 1–6.

[18] D. Bau, J.-Y. Zhu, J. Wulff, W. Peebles, H. Strobelt, B. Zhou, and A. Torralba, “Seeing what a gan cannot generate,” 2019. [Online]. Available: https://arxiv.org/abs/1910.11626

[19] L. Mi, M. Shen, and J. Zhang, “A probe towards understanding gan and vae models,” 2018. [Online]. Available: https://arxiv.org/abs/1812.05676

[20] H. K. Khanuja and A. A. Agarkar, “Towards gan challenges and its optimal solutions,” in *Generative Adversarial Networks and Deep Learning*. Chapman and Hall/CRC, 2023, pp. 197–207.

[21] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” *Artificial intelligence review*, vol. 53, pp. 5455–5516, 2020.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[25] A. Dosovitskiy, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[26] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A convnet for the 2020s,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 976–11 986.

[27] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne.” *Journal of machine learning research*, vol. 9, no. 11, 2008.

[28] R. Racine, G. A. Walker, D. Nadeau, R. Doyon, and C. Marois, “Speckle noise and the detection of faint companions,” *Publications of the Astronomical Society of the Pacific*, vol. 111, no. 759, p. 587, 1999.