FernUniversität in Hagen
Faculty of Mathematics and Computer Science

AIG
Artificial Intelligence Group

Self-Supervised Learning for Wideband Signal Recognition

Master's Thesis

in partial fulfillment of the requirements for
the degree of Master of Science (M.Sc.)
in Data Science

submitted by
Roman Ziske

First examiner:	Prof. Dr. Matthias Thimm Artificial Intelligence Group
Advisor:	Prof. Dr. Matthias Thimm Artificial Intelligence Group

Statement

I declare that I have written the master´s thesis independently and without unauthorized use of third parties. I have only used the indicated resources and I have clearly marked the passages taken verbatim or in the sense of these resources as such. The assurance of independent work also applies to any drawings, sketches or graphical representations. The work has not previously been submitted in the same or similar form to the same or another examination authority and has not been published. By submitting the electronic version of the final version of the master´s thesis, I acknowledge that it will be checked by a plagiarism detection service to check for plagiarism and that it will be stored exclusively for examination purposes.

I explicitly agree to have this thesis published on the webpage of the artificial intelligence group and endorse its public availability.

Software created for this work has been made available as open source; a corresponding link to the sources is included in this work. The same applies to any research data.

Friedrichshafen, 25.08.2025 Roman Zeske

(Place, Date) (Signature)

Zusammenfassung

Diese Arbeit untersucht systematisch domänenspezifisches Pretraining mit Self-Supervised Learning (SSL) zur Erkennung von Radiofrequenzsignalen (RF) im breitbandigen Spektrum. Angesichts des rasanten Wachstums drahtloser Technologien ist das RF-Spektrum zunehmend ausgelastet. Zuverlässige Spektrumüberwachung und Signalerkennung sind daher entscheidend, um dynamische Frequenzzuweisungen zu ermöglichen und Interferenzen zu vermeiden.

Radio Frequency Machine Learning (RFML) gilt dabei als vielversprechender Ansatz für die automatisierte Detektion und Klassifikation von Signalen. Eine Hürde bleibt jedoch der Mangel an gelabelten Daten, da deren Erhebung teuer ist und die Annotation spezialisiertes Fachwissen erfordert.

Dieses Problem wird addressiert, indem drei in der Computer Vision etablierte SSL-Pretraining-Verfahren, DenseCL, VICRegL und Masked Autoencoders (MAE) auf RF-Spektrogramme übertragen, um aus unlabelten Daten aussagekräftige Repräsentationen zu lernen. Evaluieren werden die Methoden in zwei Szenarien: Dateneffizienz auf TorchSig Wideband (1k–100k Samples) und domänenübergreifender Transfer von Kommunikations- zu Radarsignalen mit dem RadDet-Datensatz.

Unsere Ergebnisse zeigen deutliche Vorteile domänenspezifischen SSL-Pretrainings, insbesondere von VICRegL. Für die Signalerkennung erreicht VICRegL 43.50 mAP (+10.67 mAP, +32.50%) und für die Detektion 71.39 mAP (+6.70 mAP, +10.36%) gegenüber Training from scratch. Zudem erzielen SSL-vortrainierte Modelle mit nur 20% der gelabelten Daten eine vergleichbare Performance.

Zugleich treten Grenzen des domänenübergreifenden Transfers zutage: Im Transfer auf Radarsignale erzielt ImageNet-Pretraining den größten Zugewinn (+15,51 mAP) und übertrifft domänenspezifisches SSL-Pretraining mit Kommunikationssignalen (VICRegL: +9,62 mAP). Das legt nahe, dass allgemein gelernte visuelle Merkmale bei stark abweichenden Signalcharakteristika robuster übertragen werden als spezialisierte RF-Features.

Die systematische Evaluierung für SSL-Pretraining zur Breitband-Signalerkennung zeigt erhebliche Verbesserungen der Dateneffizienz innerhalb spezifischer Domänen und legt zugleich wichtige domänenübergreifende Einschränkungen offen, die auf die Notwendigkeit RF-spezifischer Foundation-Modelle hinweisen, die auf vielfältigen Signaltypen vortrainiert sind.

Abstract

This thesis presents a systematic evaluation of domain-specific self-supervised pretraining (SSL) for wideband radio frequency (RF) signal recognition. The rapid growth of wireless technologies has led to increasingly congested RF spectrum, making effective spectrum sensing and signal recognition critical for radio systems to dynamically manage frequency allocation and avoid interference.

Enabling intelligent spectrum management, RF machine learning (RFML) has emerged as a promising approach for automated signal detection and classification. However, RFML faces a fundamental challenge: labeled data are scarce because collection is costly and annotation requires expert knowledge.

We address this limitation by adapting three computer vision SSL pretraining methods, DenseCL, VICRegL, and Masked Autoencoders (MAE), to learn meaningful representations from unlabeled RF spectrograms. We evaluate these methods systematically in two scenarios: data-efficiency experiments on TorchSig Wideband (1k—100k samples) and cross-domain transfer from communication to radar signals using the RadDet dataset.

Our results demonstrate that domain-specific SSL pretraining, particularly VICRegL, delivers substantial improvements. VICRegL achieves 43.50 mAP (+10.67 mAP, 32.50%) for signal recognition and 71.39 mAP (+6.70 mAP, 10.36%) for signal detection compared to training from scratch. SSL-pretrained models achieve comparable performance using only 20% of the labeled data.

However, cross-domain transfer experiments reveal critical limitations. Domain-specific pretraining on communication signals (VICRegL: +9.62 mAP) is outperformed by ImageNet pretraining (+15.51 mAP) when applied to radar signals, suggesting that general visual features may be more universally applicable than specialized RF features, when signal characteristics differ significantly.

This systematic SSL pretraining evaluation for wideband signal recognition demonstrates significant improvements in data efficiency within specific domains. It also reveals cross-domain transfer limitations, highlighting the need for RF-specific foundation models pretrained on diverse signal types.

Contents

List of Figures

List of Abbreviations

AI	Artificial Intelligence
AMC	Automatic Modulation Classification
AP	Average Precision
AWGN	Additive White Gaussian Noise
BN	Batch Normalization
CFO	Carrier Frequency Offset
CNN	Convolutional Neural Network
DETR	Detection Transformer
DenseCL	Dense Contrastive Learning
DL	Deep Learning
DNN	Deep Neural Network
FFT	Fast Fourier Transform
FPN	Feature Pyramid Network
IoT	Internet of Things
IoU	Intersection over Union
IQ	In-phase and Quadrature
JEA	Joint Embedding Architectures
JEPA	Joint Embedding Predictive Architectures
LAD	Localization Algorithm with Double Thresholding
LN	LayerNorm
MAE	Masked Autoencoder
mAP	Mean Average Precision
ML	Machine Learning
MLP	Multi-Layer Perceptron
MSA	Multi-Head Self-Attention

MSM Masked Spectrogram Modeling

NLP Natural Language Processing

NMS Non-Maximum Suppression

PAM Pulse-Amplitude Modulation

RF Radio Frequency

RFML Radio Frequency Machine Learning

ResNet Residual Network

ReLU Rectified Linear Unit

RPN Region Proposal Network

RoI Region of Interest

RoIAlign Region of Interest Align

R-CNN Region-based Convolutional Neural Network

RT-DETR Real-Time Detection Transformer

SFP Simple Feature Pyramid

SGD Stochastic Gradient Descent

SNR Signal-to-Noise Ratio

SSL Self-Supervised Learning

TL Transfer Learning

VICReg Variance-Invariance-Covariance Regularization

VICRegL VICReg for Local Visual Features

ViT Vision Transformer

YOLO You Only Look Once

1. Introduction

The Radio Frequency (RF) spectrum is the backbone of modern wireless communication, enabling everything from radio broadcasting to satellite navigation. However, this finite resource faces growing congestion as wireless technologies expand and demand for spectrum access increases. The rapid growth of 5G networks, Internet of Things (IoT) devices, and new cognitive radio systems have created complex electromagnetic environments where multiple signals coexist, overlap, and interfere across wide frequency [69].

Traditional RF signal processing approaches rely heavily on manual feature engineering and expert knowledge, requiring carefully designed preprocessing chains to extract signal characteristics for pattern recognition algorithms. While effective in controlled scenarios, these methods often struggle in dynamic real-world environments where signal conditions vary unpredictably [69]. The emergence of Radio Frequency Machine Learning (RFML) has offered a data-driven alternative, using Deep Neural Networks (DNNs) to automatically learn relevant features directly from raw RF data. This shift has enabled progress in spectrum sensing applications, especially in wideband signal recognition—the joint task of detecting, localizing, and classifying multiple signals across broad frequency ranges [66].

Wideband signal recognition presents unique challenges that distinguish it from traditional signal classification. Rather than processing pre-isolated signals with known center frequencies and bandwidths, wideband recognition must simultaneously address signal detection, time-frequency localization, and classification within complex spectral environments containing overlapping transmissions and unknown signal placements [7]. Drawing inspiration from computer vision, researchers have adapted object detection techniques to process spectrogram representations of RF signals, treating each signal as an object to be detected and classified within the time-frequency domain [7, 44, 58].

Despite these advances, RFML faces a major challenge: data scarcity. Unlike computer vision and natural language processing, which benefit from large labeled datasets like ImageNet [54] and big text collections, the RF domain has limited access to labeled real-world data. Collecting and labeling RF signals needs special equipment, expert knowledge, and is often restricted by privacy and security rules [27]. As a result, most RFML research uses synthetic datasets, creating a gap between lab conditions and real-world use. Recent work has shown that achieving robust real-world performance can require order of magnitudes more labeled data than han any existing open RF dataset provides [27].

Self-Supervised Learning (SSL) has become a useful way to address data scarcity in machine learning. By learning meaningful representations from unlabeled data using simple pretext tasks, SSL methods can reduce the amount of labeled data needed for downstream tasks [3]. In computer vision, methods like contrastive learning [10], non-contrastive approaches [4], and masked image modeling [21] have shown strong results in learning features that help with sample efficiency and

generalization. Recent work has also focused on dense prediction tasks like object detection, with methods such as Dense Contrastive Learning (DenseCL) [65] and VICRegL [5] learning both global and local features needed for localization and classification.

While SSL has shown promising results in RFML applications, especially for modulation classification [12, 73], its potential for wideband signal recognition is mostly unexplored. Existing approaches focus on single-signal classification instead of the more complex multi-signal detection and recognition tasks found in wideband environments. Furthermore, current SSL evaluations in RFML often lack proper baselines, usually comparing domain-specific pretraining only against random initialization instead of established computer vision models pretrained on large datasets like ImageNet.

This thesis addresses these limitations by developing and evaluating domain-specific SSL pretraining strategies tailored for wideband signal recognition. We systematically investigate how modern SSL methods can be adapted to learn meaningful representations from unlabeled spectrogram data and assess their effectiveness in improving data efficiency and cross-domain transfer capabilities. Our approach leverages state-of-the-art SSL techniques, including contrastive learning through DenseCL and non-contrastive methods via VICRegL, adapting them with RF-specific augmentations to capture the unique characteristics of wideband signal environments.

The main contributions of this work include: (1) the first evaluation of domain-specific SSL pretraining for wideband signal recognition, showing clear improvements in data efficiency compared to training from scratch; (2) systematic comparison against ImageNet-pretrained baselines, giving practical insights about the trade-offs between domain-specific and general-purpose pretraining; (3) cross-domain transfer learning evaluation from communication signals to radar signals, showing the limitations of small scale domain specif pretraining; and (4) suggestions for future research, including the development of RF-specific foundation models that could help bridge the gap between domain transferability.

This thesis is organized as follows. Section 2 provides essential background on the RF spectrum, RFML, object detection in computer vision, and SSL fundamentals. Section 3.1 reviews related work in wideband signal recognition and SSL applications in the RF domain. Section 4 details our problem formulation, datasets, preprocessing pipeline, and adapted SSL methods. Section 5 presents our experimental setup and results for both data efficiency and transfer learning scenarios. Finally, we conclude with key findings and directions for future research in RFML. All code and experiment configurations are available at `https://github.com/romanziske/selfrf-repogithub.com/romanziske/selfrf-repo`.

2. Background

This chapter summarizes the technical background required for this thesis. It covers the Radio Frequency (RF) spectrum and its operational challenges, the emerging field of RFML, object detection methods from computer vision applied to spectrograms, and Self-Supervised Learning (SSL) fundamentals. These topics provide the interdisciplinary foundation for adapting SSL techniques to wideband signal recognition and motivate the design choices used throughout this work.

2.1. Introduction to the Radio Frequency Spectrum

The Radio Frequency spectrum refers to a specific segment of the electromagnetic spectrum, ranging from $3\,\mathrm{kHz}$ to $300\,\mathrm{GHz}$ [55]. Within this range, electromagnetic waves can propagate through both the atmosphere and space, making them essential for wireless communication as well as various applications like radar systems and navigation services.

To prevent interference and ensure efficient management, the RF spectrum is divided into distinct frequency bands, each possessing unique characteristics and serving specific functions [55]. For instance, the High Frequency band is known for its long-range communication capabilities, while the Ultra High Frequency band is commonly used for television broadcasting, Wi-Fi, and GPS. Figure 1 illustrates the allocation of these frequency bands across the RF spectrum.

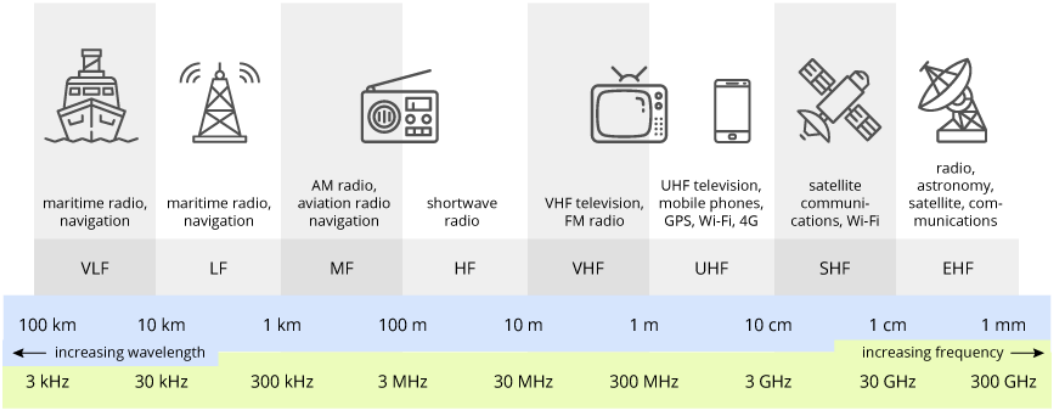


Figure 1: Overview of standardized RF bands, with typical frequency ranges and representative applications (navigation, broadcast, mobile, satellite, radar) [55]. The figure highlights how different services occupy distinct bands.

Spectrum allocation is strictly regulated by national and international authorities [26]. These organizations ensure that the spectrum is utilized efficiently and free from harmful interference [26]. In many instances, portions of the spectrum are sold or licensed to private operators, such as cellular telephone companies or broadcast

television stations, who are then tasked with managing these frequencies for their specific services [57].

However, the RF spectrum is a finite resource, and with the rapid rise in wireless technologies and increasing demand, it has become increasingly congested in recent years [20]. This congestion causes significant issues, including signal interference, which degrades communication quality and reliability, and a shortage of available frequencies for emerging services [20].

The growing demand for spectrum access increases the need for more efficient spectrum utilization and management strategies. To tackle these challenges, advanced technologies like cognitive radio and dynamic spectrum access are being developed [20]. Cognitive radio leverages intelligent algorithms and real-time spectrum sensing to detect unused frequency bands ("spectrum holes") and dynamically adjusts its transmission parameters to avoid interfering with licensed users [20]. Similarly, dynamic spectrum access allows radio systems to adapt their operating frequencies based on real-time availability, offering more flexible and efficient spectrum use [20]. Additionally, spectrum sharing—enabling multiple users or services to operate within the same frequency band under controlled conditions, further enhances spectrum efficiency and helps alleviate congestion.

Artificial Intelligence (AI) and Machine Learning (ML) are set to significantly enhance these technologies in next-generation networks. By improving spectrum sensing accuracy, predicting usage patterns, and optimizing frequency allocation in real time, AI and ML enable more effective spectrum management [69]. These advancements are made possible by signal recognition, a key focus of this work: ML models can be trained to identify and classify RF signals, allowing systems to adapt to changing spectrum conditions, mitigate interference, and improve overall transmission performance.

2.2. Radio Frequency Machine Learning

Radio Frequency Machine Learning (RFML) is defined as the application of ML, particularly Deep Learning (DL), techniques to a variety of RF signal processing tasks, such as signal detection, specific emitter identification, and anomaly detection, primarily within the wireless communications domain [69]. These tasks are inherently complex due to the unpredictable nature of RF environments, which are often affected by noise, interference, and signal overlap.

Historically, RF signal processing relied heavily on manual feature engineering, a process where domain experts carefully designed specific features to characterize signals. Such expert designed pre-processing chains were widely used to extract signal attributes, which were then processed by pattern recognition algorithms such as decision trees or support vector machines [69]. Although these methods worked well in controlled scenarios, they often faltered in real-world settings, where dynamic conditions and different signal variations demanded greater generalization than manual approaches could provide [69].

In contrast, RFML adopts a data-driven paradigm, utilizing raw RF data as direct input to DNNs. This approach significantly reduces the reliance on expert-defined features and extensive prior knowledge of the electromagnetic spectrum. By automatically extracting relevant features from raw data, RFML bypasses the need for computationally costly preprocessing steps [69]. This capability shines in applications like spectrum sensing and cognitive radio, where systems must rapidly adapt to unfamiliar or fluctuating conditions. The strength of RFML lies in its ability to learn from diverse datasets, enabling it to generalize across different RF environments and effectively handle novel or impaired signal types [69].

RFML has already demonstrated successes in many practical applications, such as Automatic Modulation Classification (AMC) [43] and signal detection [44]. These accomplishments show the potential of RFML as a disrupting technology for next-generation wireless standards, including 5G and 6G [32]. These communication standards present challenges like complex signal structures, higher frequencies, and dense network deployments. RFML can address these challenges by providing robust, adaptive solutions that enhance network performance, security, and reliability.

In the following sections, we introduce two key task in RF signal processing: narrowband signal classification and wideband signal recognition. Narrowband signal classification focuses on identifying signals within limited frequency ranges, typically targeting specific communication channels. In contrast, wideband signal recognition, the primary focus of this thesis, addresses the more complex challenge of detecting and classifying multiple signals across broader spectrum. Both task play critical roles in spectrum sensing.

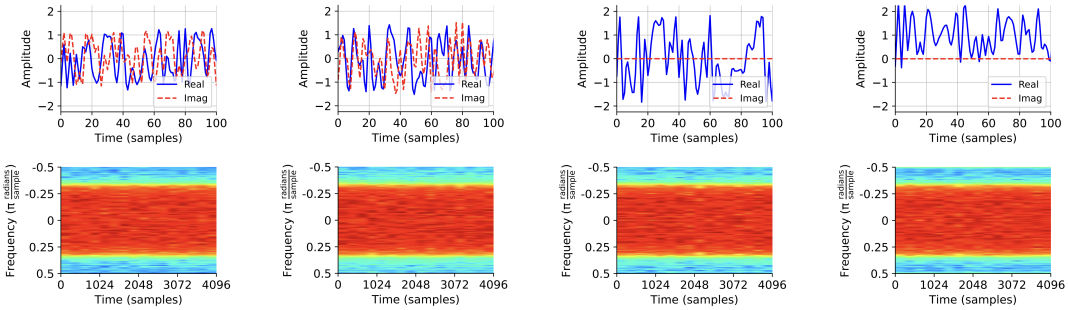
2.2.1. Narrowband Signal Classification

Narrowband signal classification is analogous to the image classification task in computer vision. Image classification involves assigning a label or class to an entire image, typically using single-object images. For example, we can classify images of handwritten digits by assigning them a label ranging from 0–9 [30].

Similarly, in narrowband signal classification, the goal is to assign a label to an entire signal [6]. The signal is typically represented either in the time domain [6, 43, 45] or as a spectrogram [39], which is a time-frequency representation, and the classification task involves determining the signal type. For example, in Automatic Modulation Classification, the goal is to determine the modulation scheme of the signal. Figure 2 illustrates four different digital modulation schemes in time- and spectrogram-domain. While a detailed explanation of these modulation techniques falls outside the scope of this machine learning-focused thesis, readers interested in the signal processing fundamentals can refer to the comprehensive work by [35].

2.2.2. Wideband Signal Recognition

Wideband signal recognition is a more challenging task than narrowband signal classification. In narrowband signal classification, it is assumed that the center fre-



(a) 16PSK (b) 32QAM (c) 4ASK (d) 8PAM

Figure 2: Representative examples from the TorchSig Narrowband dataset showing four digital modulation schemes (16PSK, 32QAM, 4ASK, 8PAM) visualized in the time and time–frequency (spectrogram) domains [6]. These examples highlight characteristic amplitude/phase trajectories and spectral patterns that enable machine learning models to discriminate between modulation types.

quency and bandwidth of the signal are known prior to classification, and the signal has already been filtered and channelized into a narrower frequency channel [6]. In contrast, wideband signal recognition addresses signals in a wideband environment, where a signal must first be detected and localized before it can be classified [7]. In the RFML domain, signal recognition presents a challenge analogous to object detection in computer vision, where the objective is to identify and localize objects within an image.

A formal definition of signal recognition is provided by West et al. [66]. They broke wideband signal recognition into the following hierarchical tasks:

1. Signal Detection: A binary classification task to determine if a signal is present in the spectrum.
2. Signal Localization: Localizing the detected signal within the spectrum.
3. Signal Classification: Classifying the type of the detected signal.
4. Signal Recognition: Combining the previous steps to detect, localize, and classify the signal.

Therefore, they defined signal recognition as a spectrum sensing problem that jointly requires detection, localization in time and frequency, and classification [66]. The distinction between these tasks is important because while well-established techniques exist for individual tasks like signal detection or localization, the field of joint signal recognition combining all tasks simultaneously is relatively new and strongly influenced by machine learning research from computer vision [33, 66].

Traditional approaches to signal detection and localization in wideband environments typically rely on signal processing techniques for specific subtasks. For

example, energy detection [8] determines the presence of a signal by measuring its energy and comparing it to a predefined threshold, though it struggles in low Signal-to-Noise Ratio (SNR) conditions. Another established approach is the Localization Algorithm with Double Thresholding (LAD), which estimates frequency boundaries in power spectra using dual thresholds. Its two-dimensional extension, LAD-2D [62], enhances this capability by localizing signals in both time and frequency domains through threshold-based analysis of spectrograms.

Spectrograms provide a three-dimensional visualization of signal characteristics, where the frequency spectrum evolves over time. Created by computing successive Fast Fourier Transforms (FFTs) of windowed signal segments, spectrograms display frequency on the vertical axis, time on the horizontal axis, and signal power through color intensity [35]. This representation captures both temporal and spectral characteristics of the signal, enabling analysis of frequency components and their dynamic behavior over time.

Modern ML approaches treat spectrograms as images, enabling the application of established object detection techniques from computer vision [7, 39, 41, 42, 44, 58, 59, 66]. To train object detection models, spectrograms must be annotated with bounding boxes that identify the time-frequency boundaries of each signal. This annotation allows models to jointly detect, localize, and classify RF signals within the wideband spectrum. Figure 3 illustrates this approach with a spectrogram containing five distinct RF signals, each enclosed by its corresponding bounding box.

An overview of related work on ML approaches for signal recognition is provided in section 3.1, where existing methods and open challenges are discussed in detail.

2.2.3. Data Needs and Scarcity

RFML faces unique challenges that hinder research and development, primarily data scarcity, labeling complexity, and privacy concerns [69]. Unlike the datasets in vision or language domains, RF datasets are scarce and difficult to collect and label. Therefore most research is limited to synthetic datasets. While synthetic datasets provide a good foundation for initial research, they often lack realism, failing to accurately reflect real-world channel conditions and hardware imperfections [40]. Moreover, this reliance on synthetic data underscores another critical limitation in the field—the absence of standardized benchmarking datasets.

There is no standard benchmarking dataset in RFML comparable to ImageNet [54] for image classification or COCO [38] for object detection. Early synthetic datasets such as RadioML 2016 [45], designed primarily for modulation classification, were created using GNU Radio and featured ten modulation types at various SNR levels. However, these datasets lacked realism and were too simplistic for practical, real-world applications. This gap was addressed in the RadioML 2018 dataset [43], which incorporated 24 modulation schemes and featured more advanced simulations. Additionally, the authors later introduced a wideband dataset aimed at signal detection and recognition tasks, serving as both research and benchmark resource [66].

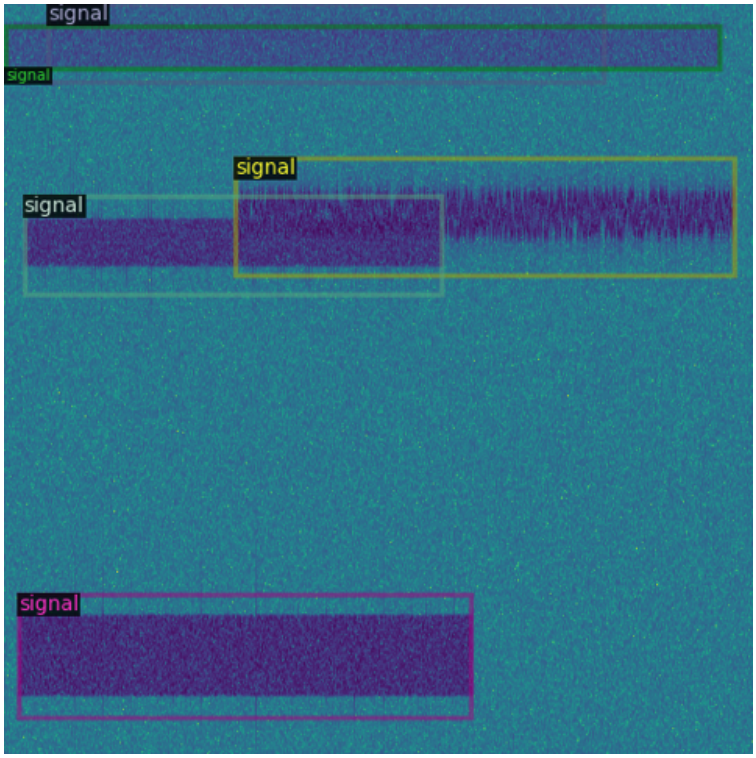


Figure 3: Spectrogram from a TorchSig Wideband sample [7] showing five distinct RF emissions with their time–frequency bounding boxes. All instances are annotated with a generic *signal* label; the image illustrates typical wideband challenges—overlapping transmissions, varying durations, bandwidths and SNR.

Despite improvements, the existing RadioML datasets still suffer from limited signal diversity, complexity, and realism, creating notable gaps between synthetic data and operational environments. Experimental findings [6] demonstrate quantitative limitations, as contemporary neural network architectures like EfficientNet quickly overfit to RadioML 2018, showing negligible performance scaling with increased model complexity. This indicates that modern neural networks capabilities have surpassed the dataset complexity provided by RadioML. Therefore, the authors of [6, 7] introduced the TorchSig family of datasets including the Narrowband Dataset [6] for single-signal classification and the Wideband Dataset [7] specifically designed for multi-signal recognition tasks. Both datasets were developed to address the limitations of earlier collections by providing more realistic channel impairments, greater signal diversity, and higher complexity suitable for modern neural networks. Initially featuring 53 different modulation schemes, the TorchSig datasets were later expanded to include 57 distinct modulation types [60], representing the most comprehensive RF signal collections available to the research community. Unlike previous datasets,

TorchSig also incorporates standardized training-validation splits and unified evaluation metrics, facilitating meaningful comparison between different approaches.

Despite these advancements in synthetic dataset realism and complexity, transitioning from synthetic to real-world data remains an open and critical challenge. Acquiring labeled real-world RF data is labor-intensive, costly, and often constrained by privacy and security regulations, restricting data sharing for academic work [27]. Furthermore, labeling RF signals demands expert knowledge and specialized equipment, resulting in high potential for mislabeling and negatively impacting model accuracy [27, 40].

Limited and biased data leads to poor model generalization and degraded performance in operational conditions different from training scenarios. RFML models trained on one hardware setup or channel environment suffer significant performance drops when tested under new conditions, illustrating the critical importance of addressing domain mismatch [67, 68].

Deep learning models, increasingly central to RFML applications, require substantial training data to achieve good generalization. Clark et al. [27] investigated this challenge specifically for modulation classification, developing methods to forecast the required dataset size for target performance levels. Their research confirmed that many existing RFML datasets are insufficient in volume to support robust performance in real-world operational environments, where signal conditions vary significantly from controlled laboratory settings.

To mitigate these issues, recent research emphasizes three main strategies:

- **Data Quantity Forecasting:** Techniques such as "Training from Zero" quantitatively forecast the amount of training data required to achieve desired model performance, enabling strategic and efficient data collection planning and better allocation of limited resources [27].

- **Transfer Learning (TL):** TL adapts pretrained models from related domains or synthetic datasets to real-world scenarios, dramatically reducing required labeled data. TL shows substantial performance improvements in domain adaptation tasks like modulation classification across different SNR or hardware variations [67, 68].

- **Self-Supervised Learning (SSL):** SSL leverages unlabeled data to create robust representations without manual annotation [3]. By training on carefully designed pretext tasks, this approach reduces labeled data requirements for downstream applications like modulation classification [12, 73]. While SSL has shown promise in narrowband contexts, its potential for wideband signal recognition remains largely unexplored, a gap this thesis aims to address.

Integrating these strategies could lead to more robust and generalizable RFML models capable of maintaining high performance across different operational scenarios. This thesis focuses specifically on the third strategy, exploring the potential of SSL for wideband signal recognition. The following section provides a brief introduction

to self-supervised learning fundamentals, while a comprehensive overview of SSL in the RF domain is presented in Section 3.2.

2.3. Object Detection in Computer Vision

Object detection is a fundamental computer vision task that seeks to *simultaneously* classify and localize every instance of a set of pre-defined object categories in an input image [15]. Unlike image classification, which assigns a single label to an entire image, object detection must identify multiple objects, determine their precise locations using bounding boxes, and classify each detected object. This dual requirement of localization and classification makes object detection significantly more challenging than simple classification tasks.

For an RGB image $I \in \mathbb{R}^{H \times W \times 3}$, a detector outputs $\mathcal{D} = \{(b_i, c_i, s_i)\}_{i=1}^{N}$, where:

- $b_i = (x_i, y_i, w_i, h_i)$ denotes the predicted bounding box,
- $c_i \in \{1, \ldots, K\}$ is the category label, and
- $s_i \in [0, 1]$ is a confidence score.

The evolution of object detection has been closely tied to advances in deep learning architectures. Early approaches like Region-based Convolutional Neural Network (R-CNN) [17] introduced the concept of region-based detection, where the detection problem is decomposed into two stages: first generating candidate object regions (region proposals), that are likely to contain objects, and then classifying these regions and refining their bounding boxes. This two-stage approach allowed leveraging powerful Convolutional Neural Network (CNN) features for object recognition while systematically exploring potential object locations. Subsequent developments like Faster R-CNN [53] integrated region proposal generation directly into the detection pipeline through the Region Proposal Network (RPN), which learns to generate object proposals end-to-end. More recent advances have explored single-stage detectors like You Only Look Once (YOLO) [52] and transformer-based approaches like Detection Transformer (DETR) [9].

Single-stage detectors such as YOLO achieve significantly faster inference speeds by performing detection in a single forward pass, making them particularly suitable for real-time applications where low latency is critical. Unlike two-stage methods that first generate region proposals and then classify them, single-stage detectors directly predict bounding boxes and class probabilities from feature maps, trading some accuracy for substantial speed improvements.

Transformer-based approaches like DETR have introduced a paradigm shift by treating object detection as a direct set prediction problem. DETR eliminates the need for post-processing steps such as Non-Maximum Suppression (NMS) that are typically required in conventional detectors to remove duplicate detections. Instead, DETR uses learnable object queries and bipartite matching during training to directly output a fixed set of predictions, simplifying the detection pipeline and

enabling end-to-end optimization. While the original DETR suffered from slow convergence and inference speeds, recent real-time adaptations such as Real-Time Detection Transformer (RT-DETR) [77] have addressed these limitations, achieving competitive inference speeds while maintaining the advantages of the transformer-based approach.

Modern developments have also seen the emergence of hybrid approaches that combine the strengths of different paradigms. ViTDet [34] represents a notable example of modern two-stage detection, adapting Vision Transformers as backbone networks within the Faster R-CNN framework. By leveraging the global receptive field and attention mechanisms of transformers while maintaining the proven two-stage detection pipeline, ViTDet achieves strong performance on complex detection tasks.

This section examines the key components of modern object detection systems. We begin by discussing the generic detection pipeline that forms the foundation of most approaches. We then explore backbone architectures, focusing on convolutional networks like Residual Network (ResNet) and Vision Transformer (ViT), which serve as feature extractors. Finally, we examine detection head networks that transform backbone features into object detections. Understanding these components is crucial for applying object detection techniques to RF signal recognition, where spectrograms serve as image-like inputs containing multiple signals that must be detected, localized, and classified.

This thesis specifically focuses on two-stage detection frameworks due to their modular design that facilitates systematic investigation of different backbone architectures and pretraining strategies. Two-stage detectors have become the standard choice in SSL research for object detection due to extensive tooling support in frameworks like Detectron2 [71], enabling straightforward integration of SSL methods [5, 65] without requiring significant architectural modifications.

2.3.1. Generic Detection Pipeline

Modern object detection systems usually follow a standardized pipeline that transforms raw input images into structured object detections through a sequence of specialized neural network components. This pipeline consists of four main stages: input preprocessing, backbone feature extraction, neck feature aggregation, and detection head prediction, as illustrated in Figure 4.

The pipeline begins with input preprocessing, where raw images are standardized through operations such as resizing, normalization, and data augmentation.

The backbone network serves as the primary feature extractor, processing the preprocessed input through multiple layers to generate feature representations at different spatial resolutions. These features capture both low-level details like edges and textures but also high-level semantic information necessary for object recognition. Common backbone architectures include convolutional networks like ResNet and transformer-based models like ViT, each offering distinct advantages in terms of

feature extraction capabilities and computational efficiency.

The neck component, typically implemented as a Feature Pyramid Network (FPN), addresses the challenge of detecting objects at multiple scales by combining features from different backbone layers. FPN creates a feature pyramid that merges high-resolution features from early layers with semantically rich features from deeper layers, enabling effective detection of both small and large objects within the same input.

Finally, the detection head processes the neck's multi-scale features to generate the final object predictions. Often followed by additional post-processing steps like NMS to filter out duplicate detections.

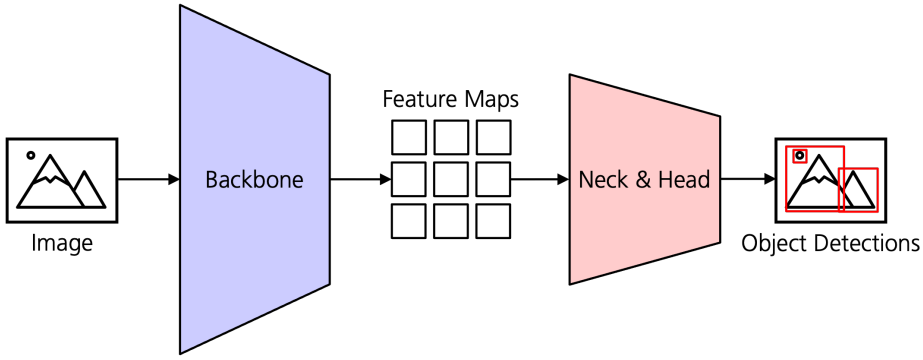


Figure 4: Generic object detection pipeline showing the flow from input image through backbone feature extraction, neck and detection head prediction to final object detections. Each component serves a specific role in transforming raw inputs into structured predictions.

2.3.2. Backbone Networks

The backbone network serves as the feature extraction component of the object detection pipeline, transforming raw input images into feature representations that capture both low-level details and high-level semantic information.

Modern object detection systems predominantly employ two types of backbone architectures: CNN and ViT [14]. CNNs, like ResNet [23], leverage hierarchical feature extraction through local convolutions and have strong inductive biases for spatial data. Vision transformers, on the other hand, use global self-attention mechanisms to capture long-range dependencies and provide more flexible feature representations. The choice between these architectures involves fundamental trade-offs in computational efficiency and data requirements.

This section provides an overview of the two most widely used backbone architectures in object detection: ResNet and ViT. These architectures have been extensively studied and adapted for various detection tasks.

ResNet. The Residual Network [23] is an influential CNN architecture that marked a breakthrough in deep learning by enabling the effective training of significantly deeper networks than previously feasible. This success established ResNet as a foundational architecture in computer vision. As a CNN, ResNet is well-suited for processing images by applying convolutional filters hierarchically to progressively extract features from local patterns and combining them into more complex structures across its deep layers.

ResNet has built-in assumptions that are helpful for image data: it assumes that nearby points in the image are related, an assumption known as *locality bias*, so it uses small filters to look at local areas first. It also assumes that if a pattern appears in one part of the image, it's the same pattern if it appears elsewhere; this property is called *translation equivariance*. ResNet's breakthrough was the introduction of *skip connections* also referred to as *residual connections* or *identity shortcuts*. These skip connections allow the input to bypass intermediate convolutional layers, ensuring that the gradient can flow more directly during backpropagation. This makes it much easier for very deep networks to learn effectively, preventing information from getting lost and enabling the creation of networks with hundreds or even thousands of layers. An overview of a typical ResNet architecture is shown in Figure 5.

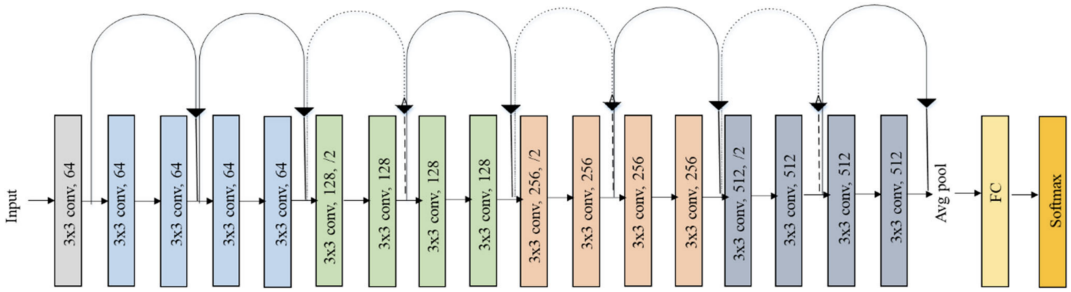


Figure 5: Overview of the ResNet-18 architecture, showing how residual blocks are stacked into stages with downsampling. The inset illustrates the structure of a basic residual block with its skip connection. From [51]; based on the architecture proposed in [23].

The fundamental building unit of ResNet is the *residual block*. Instead of forcing a stack of layers to directly learn a desired underlying mapping $H(x)$ from an input x, the core idea is to reframe the problem. The layers within the block are tasked with learning a *residual function* $F(x) = H(x) - x$. The block's output is then formed by adding the input x back to the output of these layers via an *identity shortcut*:

$$H(x) = F(x) + x \tag{1}$$

This formulation, introduced by He et al. [23], rests on the hypothesis that optimizing the residual mapping $F(x)$ is often simpler than learning the target mapping $H(x)$ directly, particularly for very deep networks. Consider the scenario where the

optimal transformation is close to the identity function, $H(x) \approx x$. It is presumed easier for the stacked layers to learn a near-zero residual function $F(x)$, thereby approximating the identity, than for the same layers to replicate the identity mapping through complex non-linear transformations. This architectural design significantly eases the training of deep networks by mitigating the vanishing gradient problem. The identity shortcut provides a direct pathway for gradient flow during backpropagation, preventing signal degradation across many layers. The block implementing the residual function $F(x)$ typically employs standard CNN components, including convolutional layers, Batch Normalization (BN) to stabilize learning, and Rectified Linear Unit (ReLU) activation functions to introduce non-linearity.

A full ResNet is constructed by stacking these residual blocks sequentially. The network is typically organized into several stages. Within each stage, the blocks operate on feature maps of the same spatial resolution and channel depth. Between stages, a downsampling operation is applied, commonly implemented using a convolutional layer with a stride > 1. This reduces the spatial height and width of the feature maps while increasing their channel depth. As a result, the network forms a hierarchical feature pyramid, capturing increasingly abstract features with larger receptive fields in deeper layers.

For downstream tasks, the feature map extracted from one of the later stages of the ResNet backbone is typically utilized. Commonly, this is the output of the final stage, often referred to as *res5* in standard implementations. This output feature map can be described as a 2D grid of feature vectors, where each vector corresponds to a specific spatial location in the original input and encodes high-level, abstract features learned by the network. For image-level classification tasks, this feature map is often globally pooled to produce a single feature vector representing the entire input. However, for object detection, the spatial grid structure is crucial, as this spatially rich feature map retains the localization information necessary for the detection head.

Vision Transformer. The Vision Transformer (ViT) [14] represents a significant shift in computer vision architectures, adapting the highly successful Transformer model [63] to handle image data directly. Rather than using convolutional layers to build hierarchical features, ViT divides the input into fixed-size patches and treats them as a sequence. This sequence-based approach allows it to leverage the powerful self-attention mechanism of Transformers.

A key distinction from traditional CNNs lies in the receptive field: while CNNs build up representations through layers of local convolutions, the self-attention mechanism of ViT enables each patch representation to potentially interact with and incorporate information from all other patches across the entire input from the very first layer, providing a global receptive field. This global attention mechanism can be advantageous for capturing long-range dependencies and complex spatial patterns. An overview of the ViT architecture is represented in Figure 6.

To adapt the standard Transformer architecture, which expects a 1D sequence of token embeddings, ViT first processes the input image $x \in \mathbb{R}^{H \times W \times C}$, where H is

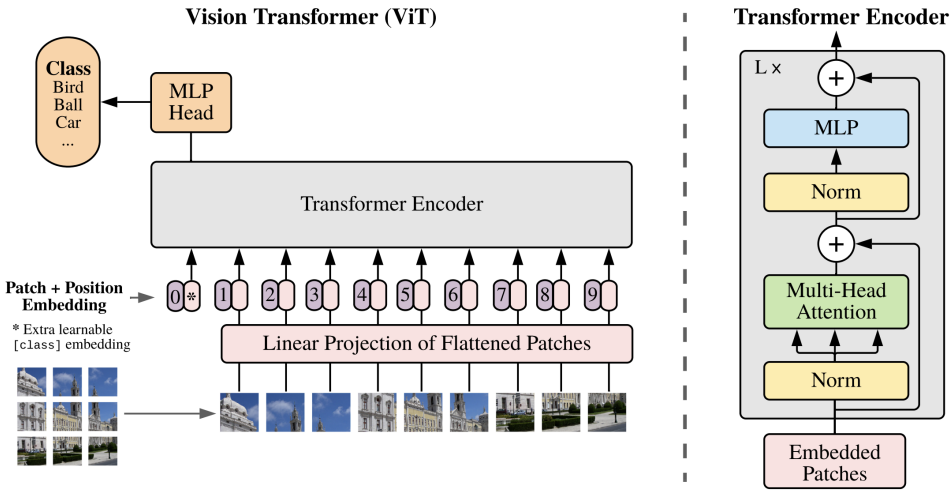


Figure 6: Vision Transformer architecture overview. The input image is divided into patches, embedded and augmented with positional embeddings, and optionally prepended with a `[class]` token. This sequence is then processed by a standard Transformer encoder. For classification tasks, the `[class]` token's output state is used and passed through a simple classification layer [14].

the height, W is the width, and C represents the number of channels. The image is reshaped into a sequence of non-overlapping 2D patches, each of size (P, P). Common patch sizes P are 16 or 32. This process, known as *patchify*, results in a sequence of $N = HW/P^2$ patches. Each patch is then flattened into a vector of size $P^2 \cdot C$. Subsequently, each flattened patch vector x_p^i is linearly projected into a D-dimensional embedding space using a learnable weight matrix $E \in \mathbb{R}^{(P^2 \cdot C) \times D}$. This projection creates a sequence of patch embeddings $[x_p^1 E, x_p^2 E, \ldots, x_p^N E]$.

The Transformer's self-attention mechanism is inherently permutation-invariant, meaning it processes the input sequence as an unordered set. Consequently, the model initially lacks built-in awareness of the original spatial arrangement of the patches. To incorporate this crucial spatial information, ViT adds positional embeddings to the patch embeddings. These are typically learnable 1D positional embeddings $E_{pos} \in \mathbb{R}^{(N+1) \times D}$, where each vector corresponds to a specific position in the sequence. Additionally, similar to BERT [13], a special learnable `[class]` token embedding x_{class} is often prepended to the sequence. The resulting input sequence z_0 is:

$$z_0 = [x_{class}; x_p^1 E; x_p^2 E; \ldots; x_p^N E] + E_{pos}$$

The sequence of patch embeddings, augmented with positional information z_0, is then processed by the Transformer encoder, which consists of a stack of L identical blocks. Each block refines the patch representations by applying two main sub-

layers: Multi-Head Self-Attention (MSA) and a Multi-Layer Perceptron (MLP) block. LayerNorm (LN) is applied before each sub-layer, and residual connections are used around each sub-layer [63].

The MSA mechanism enables the model to capture global context by allowing each patch embedding in the sequence to interact with and aggregate information from all other patch embeddings. This is achieved by projecting the normalized input into three distinct representations using learned linear transformations: Queries (Q), Keys (K), and Values (V). Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions by projecting the Q, K, and V matrices h times with different learned linear projections for each head i.

The computations of the Transformer encoder can be summarized as follows:

$$z_0 = [x_{class}; x_p^1 E; x_p^2 E; \ldots; x_p^N E] + E_{pos} \tag{2}$$

$$z'_l = \text{MSA}(\text{LN}(z_{l-1})) + z_{l-1}, \quad l = 1 \ldots L \tag{3}$$

$$z_l = \text{MLP}(\text{LN}(z'_l)) + z'_l, \qquad l = 1 \ldots L \tag{4}$$

$$y = \text{LN}(z_L^0) \tag{5}$$

Equation 2 shows the input sequence creation. Equations 3 and 4 detail the computations within a single Transformer block, which are repeated L times. The final output state corresponding to the `[class]` token, z_L^0, is often normalized to produce y (Eq. 5), which serves as an aggregated representation typically used for classification tasks. However, for dense prediction tasks like object detection, the sequence of output patch embeddings $z_L = [z_L^1, \ldots, z_L^N]$ is generally used as the feature map for the detection head, as these retain spatial information necessary for localization tasks, following approaches like ViTDet [34].

Comparison. The choice between CNN and ViT backbones involves fundamental trade-offs, particularly in the context of SSL pretraining. CNNs rely on hierarchical feature extraction using local convolutional filters, which are well-suited for capturing local patterns and textures. Their strong inductive biases, such as locality and translation equivariance, make them effective for learning meaningful representations even with smaller datasets. This aligns well with many SSL methods that leverage local relationships in the data, such as contrastive learning or reconstruction-based approaches.

In contrast, ViTs use global self-attention, enabling them to model long-range dependencies and global context from the very first layer. This flexibility allows ViTs to capture more complex patterns in the data, which can be advantageous for certain SSL tasks, such as masked token prediction or global contrastive objectives. However, the weaker inductive biases of ViTs mean they often require larger datasets, strong data augmentations or more pretraining to achieve comparable performance to CNNs.

2.3.3. Necks & Detection Head Networks

The final components of the object detection pipeline are the neck and detection head networks, which work together to transform backbone features into object predictions. The neck component addresses multi-scale detection challenges by aggregating features from different backbone layers, while the detection head specializes in object localization and classification.

This section examines two prominent detection frameworks that exemplify modern neck and head design: Faster R-CNN with FPN for ResNet backbones, and ViTDet for ViT backbones. These frameworks represent the current state-of-the-art in two-stage object detection and provide the foundation for the detection architectures employed in this thesis.

Faster R-CNN. Faster R-CNN [53] with FPN [37] represents a two-stage object detection framework for CNN backbones like ResNet. The architecture consists of three main components working in sequence: the backbone feature extractor, the RPN, and the Region of Interest (RoI)-based detection head.

The backbone network, typically ResNet, extracts multi-scale feature representations from the input image. FPN serves as the neck component that addresses the fundamental challenge of detecting objects at vastly different scales within the same image. Traditional CNN backbones produce feature maps with a trade-off between spatial resolution and semantic richness—early layers contain high-resolution features with fine spatial details but limited semantic information, while deeper layers provide semantically rich features at reduced spatial resolution. FPN resolves this by constructing a feature pyramid through a top-down pathway, upsampling features from deeper layers and combining them with corresponding shallower layers through element-wise addition. This creates a multi-scale feature representation where each pyramid level maintains both semantic richness and spatial precision, enabling effective detection of both small and large objects.

The detection head component of Faster R-CNN employs a two-stage design that processes features through sequential refinement stages. The RPN operates as the first stage of detection, sliding a small network over the FPN feature maps to generate class-agnostic region proposals. At each spatial location across all pyramid levels, the RPN predicts objectness scores and bounding box coordinates using multiple anchor boxes of different scales and aspect ratios. These anchors serve as reference points that are refined to produce precise object proposals. By operating on multiple pyramid levels, the RPN can effectively propose regions for objects of varying sizes.

In the second stage, features for the proposed regions are extracted from the backbone's feature maps using Region of Interest Align (RoIAlign), which accurately pools features corresponding to each proposal and warps them to a fixed spatial size. This ensures consistent input dimensions for the subsequent classification and regression heads. The final detection head consists of fully connected layers that perform two tasks: classifying each proposal into one of the predefined object classes

and regressing the bounding box coordinates to achieve more precise localization.

ViTDet. ViTDet [34] represents an innovative approach to adapting Vision Transformers for object detection tasks while maintaining the simplicity and effectiveness of plain ViT architectures. Unlike hierarchical transformer variants that attempt to mimic the multi-scale structure of CNNs, ViTDet demonstrates that a standard, non-hierarchical ViT can serve as an effective detection backbone without requiring architectural modifications.

The backbone component of ViTDet employs a plain ViT encoder that processes the input image as a sequence of patches. A standard ViT backbone produces a single, semantically rich feature map representing the spatial downsampling from the original input resolution. Unlike CNNs that naturally generate multi-scale features through their hierarchical structure, ViT outputs a flat sequence of patch embeddings that must be adapted for the spatial requirements of object detection.

The plain backbone produces a single feature tensor at a fixed stride of typically 16, which is insufficient for detecting objects at multiple scales. To address this, ViTDet introduces the Simple Feature Pyramid (SFP) starting from the single, semantically rich ViT feature map $F \in \mathbb{R}^{C \times (H/16) \times (W/16)}$, it applies lightweight 1×1 convolutions in parallel, followed by up- and down-sampling to produce pyramid levels. Unlike a traditional FPN that requires complex lateral connections and top-down fusion to combine features from different CNN backbone layers, SFP operates solely on the single ViT output and uses no lateral merges. This simple design is both fast and memory-efficient, yet provides the multi-scale representations required for reliable detection of objects of varying sizes. The key advantage lies in eliminating the need for hierarchical feature fusion while maintaining the standard pyramid levels that detection heads expect.

The detection head component of ViTDet maintains full compatibility with established two-stage detection frameworks. The multi-scale features generated by the Simple Feature Pyramid can be directly fed into unmodified detection heads such as Faster R-CNN. This compatibility enables ViTDet to leverage proven detection architectures without requiring specialized adaptations for transformer-based features. The RPN and RoI-based classification components operate on the pyramid features in the same manner as with CNN backbones, ensuring consistent training procedures and performance expectations.

A key practical consideration in ViTDet is computational efficiency during fine-tuning. When processing high-resolution images, the global self-attention mechanism of standard ViT becomes computationally expensive due to its quadratic scaling with input size. ViTDet addresses this by replacing global self-attention with non-overlapping window attention, which limits attention computation to local windows rather than across the entire image. Importantly, these architectural modifications are applied only after pretraining, allowing the model to benefit from global self-attention during representation learning while maintaining computational efficiency during downstream fine-tuning.

The strength of ViTDet lies in its minimalist design philosophy, which demonstrates that competitive detection performance can be achieved without introducing complex hierarchical token structures or extensive architectural modifications. This approach allows advances in plain ViT pretraining, including SSL methods like MAE, to translate directly into object detection improvements without requiring specialized adaptation techniques.

2.4. Self-Supervised Learning Fundamentals

Deep learning models have traditionally relied on large labeled datasets to reach state-of-the-art performance. However, collecting and annotating such data is expensive, time-consuming, and often requires domain experts in specialized settings [2]. This dependency on labels limits scalability and motivates methods that can exploit large amounts of unlabeled data.

One such approach is Self-Supervised Learning (SSL), which constructs supervisory signals directly from the data via *pretext tasks*. In effect, the model "supervises itself" by predicting masked parts of the input or relationships across augmented views, without human annotations [2]. This differs from classic unsupervised learning, which models the data distribution without explicit targets [3]. When scaled in data and compute, SSL now rivals, and in some cases surpasses, fully supervised learning across multiple domains [3]. Given its potential to enable AI systems to acquire broad foundational knowledge, SSL has been described as the *dark matter of intelligence* [31].

SSL typically follows a two-stage pipeline of *pretraining* and *downstream fine-tuning*, as seen in Figure 7. In the first stage, an encoder is trained on large unlabeled corpus to solve a pretext task that encourages compact, semantically meaningful representations [3]. In the second stage, the pretrained encoder is transferred to a supervised downstream task with limited labels, a lightweight task head (e.g., for classification or detection) is trained on top. The encoder's pretrained weights may be frozen to reuse the learned representation or jointly fine-tuned with the head to maximize downstream performance.

In Natural Language Processing (NLP), masked language modeling predicts hidden tokens in a sequence [13]. Early vision work explored rotation prediction [16] and colorization from grayscale [75], among others.

Recent SSL advances can be grouped into three families [3]: *Generative architectures*, *JEA*, and *JEPA*. Generative methods reconstruct the input; JEA learn invariances by pulling augmented views of the same image together in embedding space while pushing others apart, JEPA predict the embedding of one part/view from another without reconstructing pixels, but predicting the abstract latent space representation.

2.4.1. Generative Architectures

One of the earliest and most straightforward SSL approaches is *generative reconstruction*. An autoencoder has two parts: an encoder that compresses the input into a

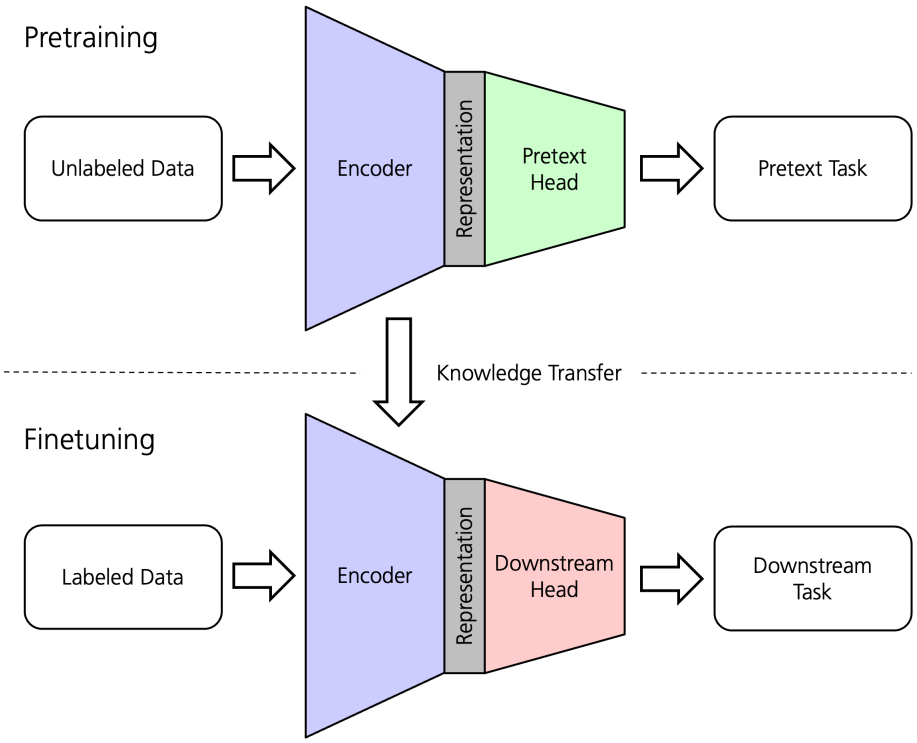


Figure 7: Self-supervised workflow. During pretraining, an encoder learns representations from unlabeled data via a pretext task. The encoder is then transferred to a supervised downstream task, where a task-specific head is trained (with the encoder frozen or fine-tuned).

smaller representation, and a decoder that tries to reconstruct the original input. By doing this, the model is forced to capture the most important patterns in the data [74]. However, simple autoencoders can just memorize pixel-level details without learning useful high-level features. To fix this problem, researchers add noise or corruption to the input, so the model must learn to reconstruct clean data from corrupted versions, leading to better representations [64]. Context Encoders [47] took this further by masking parts of images and training models to fill in the missing pieces, which requires understanding what should be there.

In natural language processing, *masking* works very well as a training signal. Models learn to predict hidden words based on the surrounding context [13]. Language models also learn by predicting the next word in a sequence [50]. These approaches show that predicting missing content helps models learn useful representations without human labels.

These ideas work for images too when using transformers. Vision Transformers treat images as sequences of patches [14], which makes *masked image modeling* possible. The idea is simple: hide some image patches and train the model to reconstruct the missing patches. Masked Autoencoder (MAE) use this approach with a ViT-

based encoder-decoder that rebuilds images from heavily masked inputs, creating representations that help with classification and detection tasks [21].

Figure 8 shows a generic encoder-decoder formulation. An input y is corrupted by some process $c(\cdot)$ to create $x = c(y)$. The encoder f maps x to a representation $s_x = f(x)$, and the decoder g produces the reconstruction $\hat{y} = g(s_x)$. Training tries to minimize the difference $D(y, \hat{y})$, using measures like ℓ_1/ℓ_2 loss or cross-entropy, encouraging f to learn useful structure beyond just copying the input.

Researchers have also tried generative approaches in RFML. Early work used autoencoders to learn representations of RF signals for tasks like clustering [46]. More recently, MAE has been applied to spectrograms by masking time-frequency patches and reconstructing the missing parts, which improves performance on tasks like spectrum segmentation [1]. We discuss these applications more in Section 3.2.

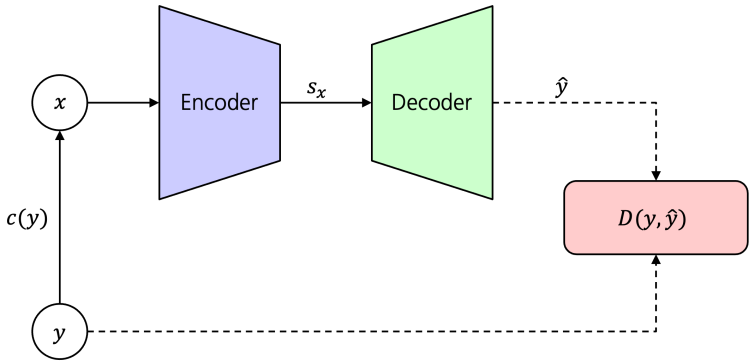


Figure 8: Encoder–decoder view of generative pretraining. A corruption process $c(\cdot)$ masks or changes the target y to produce x. The encoder f maps x to a representation s_x, and the decoder g reconstructs $\hat{y}$. Training tries to minimize the difference $D(y, \hat{y})$. Inspired by [29]

2.4.2. Joint Embedding Architectures

Joint embedding architectures (JEAs) are discriminative SSL methods that learn invariances by aligning representations of related inputs. Instead of reconstructing pixels, JEAs place embeddings of similar inputs close together in latent space and push apart embeddings of dissimilar inputs, typically with far lower computational cost than generative approaches [3].

Figure 9 illustrates the setup: two related views x and y (e.g., augmentations of the same image) are passed through an encoder to produce s_x and s_y. Training aims $D(s_x, s_y)$ to be small for positive pairs and large for negatives, or to satisfy collapse-avoidance constraints in non-contrastive variants.

SimCLR [10] creates positive pairs by applying strong data augmentations like cropping, color changes, and blurring to the same image. All other samples in the training batch serve as negatives. The contrastive loss function pulls positive

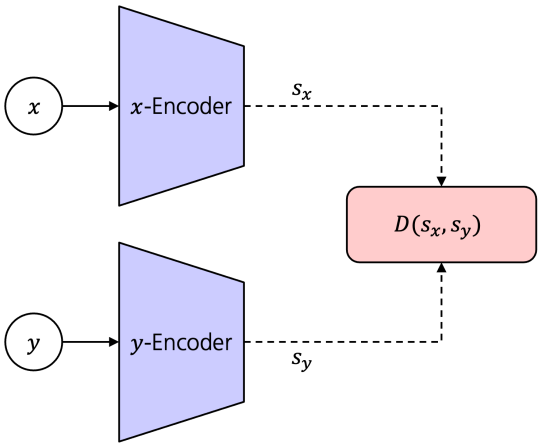


Figure 9: Joint Embedding Architecture (JEA). Two related views are embedded and trained to align in representation space. Inspired by [29]

pairs together in embedding space while pushing negatives apart. However, this approach needs large batch sizes to provide enough negative examples, which increases memory requirements and computational cost [3, 28]. DenseCL [65] extends contrastive learning to dense prediction tasks by matching local features between different views, making it more suitable for object detection and segmentation.

To avoid the need for explicit negative samples, several non-contrastive methods have been developed. BYOL [19] uses two networks: an *online* network that learns to predict the representation from a slowly updating *target* network (updated via exponential moving average). A stop-gradient operation and small predictor head prevent the model from learning trivial solutions. VICReg [4] takes a different approach, avoiding negatives entirely by using three loss terms: (i) *invariance* ensures that different views of the same image have similar embeddings, (ii) *variance* prevents all embeddings in a batch from becoming identical, and (iii) *covariance* encourages different embedding dimensions to capture different types of information. VICRegL extends this to local patch-level features. These methods achieve results comparable to contrastive learning but are less sensitive to batch size and augmentation strategies [3].

Contrastive JEA methods have shown promise for RF representation learning, e.g., modulation classification [12, 70, 73]. We discuss RF-specific adaptations in Section 3.2.

2.4.3. Joint Embedding Predictive Architectures

Joint embedding predictive architectures (JEPAs) learn to predict one embedding from another rather than matching two embeddings directly [29]. Typically, a *context* encoder produces s_x from input x (e.g., a visible region), while a *target* encoder produces s_y from a related but disjoint input y (e.g., a masked/held-out region).

A predictor maps s_x to $\hat{s}_y$, and training minimizes a discrepancy $D(\hat{s}_y, s_y)$. Unlike generative methods, JEPAs operate in representation space without expensive pixel reconstruction and, unlike JEAs, use an asymmetric context–target setup that explicitly predicts missing information in embedding space.

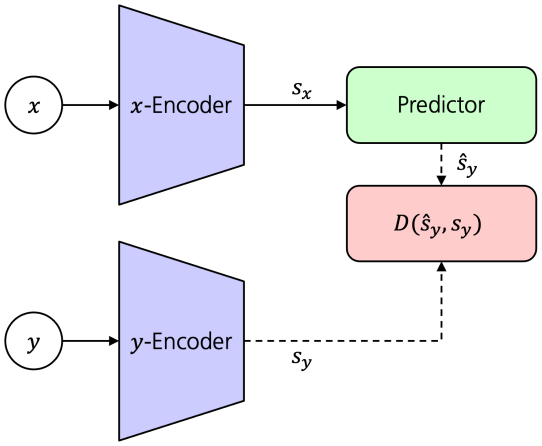


Figure 10: Joint Embedding Predictive Architecture (JEPA). A predictor uses the context embedding s_x to predict the target embedding $\hat{s}_y$, trained to match s_y. Inspired by [29]

JEPAs are comparatively new and less standardized than generative or JEA methods. Best practices and benchmarks are still evolving, but results so far are promising [3, 29].

2.4.4. Pretraining for Object Detection

Many popular SSL methods like SimCLR, were first evaluated on image classification, emphasizing *global* representations of an entire image. Such features are not tailored to *local* details, which is crucial for dense prediction tasks like object detection [3]. Because wideband signal recognition in this thesis requires localizing and classifying multiple signal instances in time–frequency space, we focus on SSL approaches that explicitly strengthen both global and local features.

DenseCL [65] adapts contrastive learning to dense prediction by forming positive pairs at the pixel level. Given two augmented views, local features are matched (e.g., via nearest neighbors in feature maps) and optimized with a contrastive loss, encouraging spatially consistent representations across views. This leads features better suited to object detection than purely global contrastive objectives.

VICRegL [5] extends VICReg by adding a local term before global pooling. Besides matching global embeddings, it enforces invariance, variance, and covariance locally on patch features aligned by the known geometric transforms between views. The result is stronger spatial sensitivity that improves dense prediction while retaining the simplicity of non-contrastive objectives.

Methods that incorporate local features better align SSL pretraining with the needs of detection-style tasks. In the context of wideband signal recognition,where multiple signals must be localized and classified, such pretraining can produce more robust, task-relevant features than purely global objectives.

3. Related Work

This chapter reviews the current state of research in wideband RF signal recognition and SSL applications in the RF domain. We begin by examining machine learning approaches to wideband signal recognition, focusing on object detection and semantic segmentation methods adapted from computer vision to process spectrogram representations of RF signals. We then explore the emerging field of SSL in RFML, highlighting both reconstruction-based and contrastive learning approaches that aim to address data scarcity challenges. The chapter concludes by identifying critical gaps in the current literature, particularly the absence of domain-specific SSL pretraining strategies for wideband signal recognition tasks and the limited availability of comprehensive datasets for evaluating cross-domain transfer capabilities. These identified gaps provide the foundation and motivation for the contributions presented in this thesis.

3.1. Machine Learning Approaches to Wideband Signal Recognition

Machine learning has emerged as a powerful tool for addressing the challenges of wideband signal recognition, where the goal is to jointly detect, localize, and classify signals across a wideband frequency spectrum. Drawing inspiration from advancements in computer vision, researchers have adapted image analysis techniques to process RF signals by converting them into spectrograms, these are image-like representations of a signal's time-frequency characteristics that enable the application of established computer vision methods.

Two primary approaches have gained prominence in this domain. Object detection methods [7, 25, 44, 58, 59] identify and localize signals using bounding boxes within spectrograms, treating each signal as a distinct object to be detected and classified. Semantic segmentation approaches [7, 66] perform pixel-level classification of time-frequency bins, enabling fine-grained signal boundary estimation and dense prediction across the entire spectrogram.

The evolution of these approaches has progressed from early proof-of-concept implementations [44] to comprehensive frameworks that handle complex scenarios with multiple overlapping signals [7, 58]. Recent work has expanded beyond communication signals to include radar detection [25] and specialized applications such as drone detection [41, 76] and IoT environment monitoring [42].

The following sections provide a detailed overview of the key contributions in each approach and their applications to wideband signal recognition.

Learning Robust General Radio Signal Detection Using Computer Vision Methods. The first adaptation of a computer vision deep learning model for wideband signal detection was published by O'Shea et al. [44] in 2017. This work marked a significant shift from traditional signal processing techniques to machine learning-based approaches in RF signal detection. The authors generated a synthetic wideband

dataset comprising 20,000 spectrogram images, each representing various modulated signals. To enable real-time processing, they employed a tiny-YOLO model, a lightweight variant of the YOLO object detection framework that requires fewer computational resources while maintaining acceptable detection performance.

The scope of this work was limited to signal detection within a wideband spectrum, intentionally simplifying the problem by assigning a single signal class to each bounding box rather than classifying signal types. The model was trained using the Adam optimizer with a batch size of 64 over 40,000 iterations. Although the authors assert that the model successfully detects signals in a wideband spectrum, they did not provide a formal performance evaluation, which limits the ability to quantitatively assess its accuracy and robustness.

Despite this, the model achieved an inference time of 3 milliseconds per spectrogram on a single desktop machine, highlighting its potential for real-time operation. This effort laid the groundwork for following research into applying computer vision techniques to RF signal analysis.

Joint Detection and Classification of RF Signals Using Deep Learning. Another YOLO-based approach was presented by Vagollari et al. in 2021 [59]. The authors generated a synthetic dataset of wideband signals, consisting of 5,000 spectrogram images with 10 different modulation types. They trained the YOLOv3 model to perform both signal detection and signal recognition, jointly detecting and classifying signals within the spectrograms. For the detection task, the model achieved an Average Precision (AP) of 87.16% and an Intersection over Union (IoU) of 90.53%. For the signal recognition task, which involved multiple signal classes, they reported an AP of 86.54% and an IoU of 90.12%, maintaining a strong performance regarding of the number of classes to recognize.

Furthermore, the authors evaluated model performance across different SNR levels and compared it with a Faster R-CNN model trained on the same dataset. At lower SNR levels, the Faster R-CNN model significantly outperformed YOLOv3. However, at higher SNR levels, YOLOv3 matched the performance of Faster R-CNN. A notable advantage of YOLOv3 was its faster inference time: 0.0029 seconds per spectrogram compared to 0.4 seconds for Faster R-CNN on the authors' hardware. Despite these strengths, the study identified limitations in classifying certain modulation classes that were not easily distinguishable in the spectrogram images.

This classification limitation was addressed in a follow-up work by Vagollari et al. in 2023 [58]. The authors proposed an end-to-end framework for wideband signal recognition, utilizing the YOLOv5 model and incorporating data augmentations such as horizontal and vertical flipping, random translation, and scaling during training. These augmentations enhanced the model's performance in wideband signal recognition. The training was conducted on a benchmarking dataset from West et al. [66].

The processing chain of the proposed end-to-end framework is illustrated in Figure 11. The framework first generates spectrograms from the wideband input signal,

which are then processed by the YOLOv5 model to simultaneously detect, localize, and classify RF signals across multiple modulation schemes. For signals using Pulse-Amplitude Modulation (PAM), which are difficult to classify directly from spectrograms, the signals are extracted using the predicted bounding boxes. These extracted signals were preprocessed and fed into an additional model specifically trained for modulation classification on narrowband signals. By routing challenging signals through this extra classifier, the authors achieved consistently strong classification results across all modulation classes.

With the adoption of YOLOv5 and the use of data augmentations during training, the model achieved an IoU of 91.44% for single-class signal detection and an IoU of 88.49% for multi-class signal recognition.

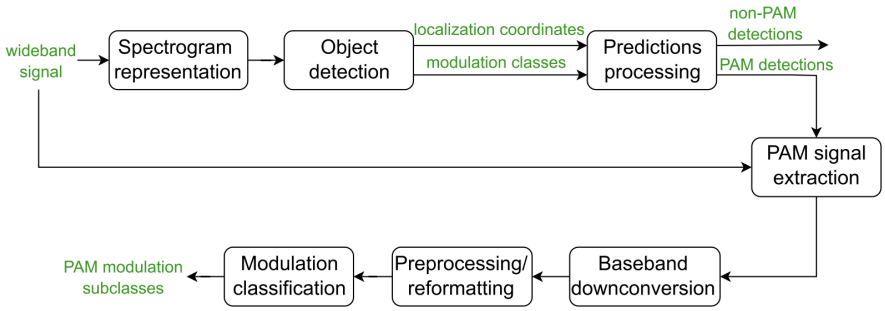


Figure 11: End-to-end framework for wideband signal recognition proposed by Vagollari et al. [58]. The pipeline converts wideband I/Q into spectrograms, applies a one-stage detector (YOLOv5) for joint detection, localization and classification, and routes ambiguous signals to a dedicated modulation classifier.

A Wideband Signal Recognition Dataset. West et al. [66] present one of the first comprehensive datasets and benchmarks explicitly designed for the wideband signal recognition task. While most spectrum sensing research focuses either on signal detection or classification, their work introduces the joint task of *signal recognition*, including detection, localization in time and frequency, and classification simultaneously. This formulation closely mirrors object recognition in computer vision.

The authors propose a new dataset released in SigMF [24] format, consisting of 130 annotated recordings representing wireless environments with realistic signal densities and distributions. The dataset includes a broad range of modulation types, such as PSK, QAM, OFDM, FSK, AM, and FM, each generated with randomized parameters, such as bandwidth, start time and SNR.

To evaluate wideband signal recognition model, West et al. advocate for the use of mAP and IoU, metrics adapted from computer vision. Their approach enables consistent and interpretable evaluation across detection, localization, and classification

subtasks.

For model design, the authors propose a neural network-based spectral segmentation architecture using U-Net, transforming the spectrogram into a pixel-wise signal mask. Post-processing is performed via connected components analysis, where each contiguous detection region is mapped to a bounding box in the time-frequency domain. Compared to traditional approaches, the U-Net model achieves significantly improved recall at low SNRs while maintaining high precision.

Despite these contributions, a significant limitation of the work is the absence of detailed quantitative evaluation. While the authors introduce metrics adapted from computer vision, they do not provide detailed performance statistics such as mAP or IoU values across different signal classes or SNR levels. The evaluation primarily focuses on qualitative comparisons to traditional approaches rather than establishing clear benchmarks with their proposed metrics. This lack of detailed performance reporting makes it difficult to compare their U-Net architecture against different approaches in the literature.

Large Scale Radio Frequency Wideband Signal Detection & Recognition. A survey on wideband signal recognition was published by Boegner et al. [7]. The researchers released one of the largest publicly available synthetic wideband datasets, containing up to 53 different modulation classes. Alongside this dataset, they introduced TorchSig [6], an RFML library designed for generating synthetic RF data and training deep learning models, complemented by various data augmentation techniques and impairment transforms.

Their large-scale analysis evaluated four different models across two paradigms: object detection (YOLOv5 and DETR) and semantic segmentation (Mask2Former and PSPNet). The semantic segmentation models performed pixel-wise classification with additional post-processing to convert the resulting segmentation masks into bounding boxes, enabling direct comparison with object detection approaches. All models were trained for 1M steps using a batch size of 32 with the Adam optimizer.

The researchers conducted two experiments: single-class signal detection,where signals were assigned the same class regardless of type, and multi-class signal recognition. For the latter, rather than using all 53 modulation types individually, they grouped them into six modulation families to focus on broader signal categorization rather than fine-grained classification as explored in their previous work [6].

In the signal detection evaluation, DETR achieved the highest performance with an mAP of 86.98%, followed by Mask2Former at 81.05%. YOLOv5 and PSPNet showed comparable results with mAP values of 73.64% and 73.59%. The authors also assessed computational efficiency: YOLOv5 processed up to 13,352.32M samples per second, while DETR processed only 74.70M samples per second. This speed gap limits DETR's practical deployment potential despite its superior accuracy. When tested across different SNR conditions, DETR demonstrated exceptional robustness at low SNR levels, where YOLOv5 and PSPNet showed significant performance degradation.

For multi-class signal family recognition, DETR again substantially outperformed other approaches with an mAP of 80.65%, compared to YOLOv5's 58.50% and PSP-Net's 51.84%. Mask2Former performed poorly with an mAP of only 27.03%, likely due to its architectural complexity.

This study provides valuable insights through its comprehensive analysis of different deep learning architectures on a large-scale RF dataset. It demonstrates the effectiveness of transformer-based object detection models like DETR for wideband signal recognition while highlighting the trade-offs between accuracy and computational efficiency. Additionally, the study contributes to the research community by providing both a challenging benchmark dataset and TorchSig, a convenient open-source toolkit for future research in RFML.

A Wideband Dataset for Real-Time Radar Spectrum Detection. Huang et al. introduced *RadDet*, a wideband radar spectrum detection dataset and benchmark tailored for real-time operation. The dataset comprises 40,000 spectrogram frames generated from 1,M complex I/Q samples over a 500,MHz band, spanning 11 radar classes, six SNR settings, two radar-density regimes, and three time–frequency resolutions. To establish baselines, the authors evaluated several compute-efficient one-stage detectors and transformer-based models on RadDet as well as a modified NIST-CBRS radar classification set, reporting mean Average Precision and inference throughput to highlight the trade-off between accuracy and real-time feasibility. Although RadDet targets radar emissions rather than general communication signals, its problem formulation (joint detection and localization in wideband spectra) and evaluation protocol closely mirror those used in wideband RF signal recognition, making it a valuable complementary benchmark for future work on spectrum-wide detection and recognition.

Further Studies. Beyond the works discussed in detail, several other studies have explored wideband signal recognition using object detection-based methods, contributing to the growing body of research in this field.

In one such study, Nelega et al. [41] utilized YOLOv5 to detect and classify RF signals emitted by drones. Their dataset included signals from seven distinct drone models across various scenarios, encompassing clean signals as well as those with Bluetooth and Wi-Fi interference. The authors reported an mAP of 95.4% across all drone types, demonstrating the robustness of YOLOv5 for RF-based drone detection. Similarly, Zhao et al. [76] from the University of California employed YOLOv5 for drone RF signal detection and fingerprinting, further highlighting the applicability of this model in such contexts.

Wideband signal recognition also extends to IoT environments. Nguyen et al. [42] investigated signal recognition in dense $2.4\,\mathrm{GHz}$ networks, employing YOLOv4 to detect and classify common wireless protocols including Wi-Fi, Bluetooth, and ZigBee. Their work showcased the practical utility of object detection methods for

monitoring the increasingly crowded spectrum occupied by IoT devices, offering insights into the scalability of these approaches in real-world settings.

3.2. Self-Supervised Learning in the Radio Frequency Domain

Self-supervised learning in the RFML domain has recently gained significant attention as a promising approach to address data scarcity challenges in RF signal analysis. This emerging field encompasses both reconstruction-based methods [1, 46] and contrastive learning approaches [12, 73] that aim to learn meaningful signal representations without requiring extensive labeled datasets.

Early work focused on autoencoder-based reconstruction for narrowband modulation classification [46], while recent advances have explored contrastive learning frameworks that leverage domain-specific augmentations to improve sample efficiency [12, 73]. The most recent developments have introduced generative approaches using masked modeling techniques for dense prediction tasks on spectrogram data [1], marking the first attempts at foundation models for RFML.

Despite these advances, existing SSL approaches in RFML primarily target narrowband signal classification tasks rather than the complex challenges of wideband signal recognition, where multiple overlapping signals must be jointly detected, localized, and classified across time-frequency representations. The following sections examine the key contributions in each approach and their implications for RF signal analysis.

Semi-Supervised Radio Signal Identification. An early example of reconstruction based SSL in the RF domain was presented by O'Shea et al. [46], the authors explored how unlabeled raw radio signals could be leveraged for modulation classification through feature learning by reconstruction. The authors proposed using a convolutional autoencoder to learn a latent space representations from raw time-series RF data without relying on class labels.

In this approach, the autoencoder learns to reconstruct the input signal by passing it into a bottleneck layer, performing nonlinear dimensionality reduction. The bottleneck layer represents, a lower dimensional latent space representation of extracted features of the input signals. Such representation vector allows signals with similar features, for example similar modulation schemes, to cluster together. The authors show that this feature space, trained without labels, reaches a high level of discrimination for modulations of signals with distinct temporal or spectral features.

The use of reconstruction loss as a pretext task is consistent with the fundamental ideas of self-supervised learning, which states that solving pretext tasks, in this case input reconstruction,learns meaningful representations. They made a significant contribution to the development of data-efficient RFML systems by proving that these representations are suitable for downstream tasks like classification or clustering.

Exploring Self-supervised Learning for Radio Signal Recognition. In the study from Yun et al. [73], the authors shifted the focus from reconstruction-based methods to explore joint-embedding based SSL for RF signals with their Self-RadioNet framework. Inspired by SimCLR [10], their approach applies contrastive learning directly to raw time-domain RF samples.

To generate invariant training pairs, the authors used domain-specific augmentations, that simulate real-world impairments such as AWGN and Carrier Frequency Offset (CFO). The architecture integrates a ResNet-based encoder, which is followed by a projection head composed of three dense layers. The model is pretrained using the Normalized Temperature-Scaled Cross-Entropy loss function, and its performance is evaluated using linear probing on a synthetically generated dataset that includes ten different modulation schemes.

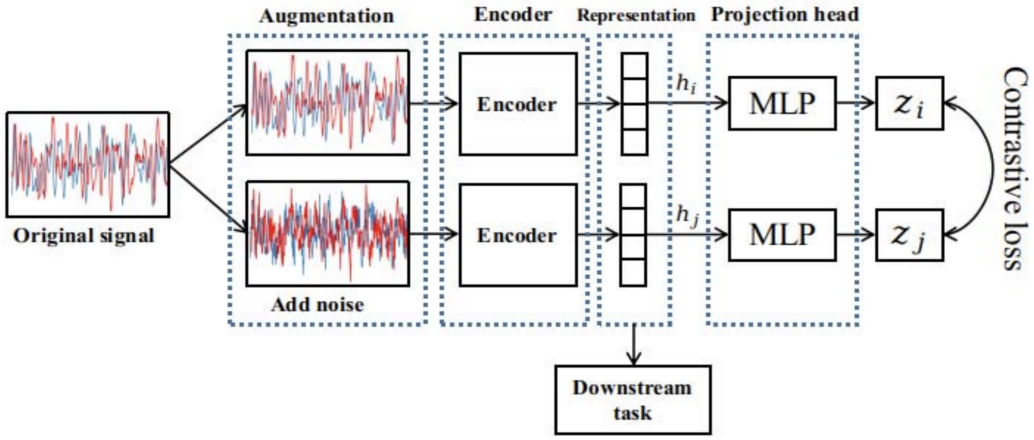


Figure 12: The Self-RadioNet framework proposed by Yun et al. [73] for contrastive representation learning on raw RF signals. Two augmented views of the same input are created through Additive White Gaussian Noise and Carrier Frequency Offset augmentations. The encoder extracts features from raw RF signals, which are then projected to a lower-dimensional space for contrastive loss calculation.

Figure 12 shows the two-branch structure of Self-RadioNet, which is a common design in many contrastive learning approaches. Each input signal is impaired by a set of randomized augmentations to create two views of the same data. These views are independently processed through identical encoders with shared weights, followed by projection heads that map the features to a lower-dimensional space where the contrastive loss is applied. This approach forces the network to learn representations that are invariant to channel effects like noise while remaining discriminative for different modulation or signal types.

Self-RadioNet demonstrates improved performance compared to the supervised baseline model when using only 10% of the available labeled training data. This

shows the potential of SSL for data-efficient RF signal classification. However, the approach has limitations regarding diversity and strength of the applied data augmentations, which is important factor [3]. Only applying the AWGN and CFO augmentations may not be sufficient to capture the full range of variations in real-world RF signals.

Self-Supervised RF Signal Representation Learning for NextG Signal Classification With Deep Learning. Kemal et al. [12] advanced contrastive learning for RF signals by addressing the limited augmentation diversity in previous approach. Building on the MoCo-v3 framework [11], they introduced a more comprehensive set of RF-specific transformations, as illustrated in Figure 13.

The authors implemented in total five data augmentations, DC shift, time shift, amplitude scaling, zero-masking, and AWGN each designed to preserve signal semantics while enhancing the robustness of the representation. This builds on top of Self-RadioNets [73] augmentations limited AWGN and CFO transformations. The architecture employs a modified ResNet50 backbone with momentum encoders and maintains a queue of negative samples for InfoNCE loss computation.

Experiments on the RadioML2016.10a [45] dataset demonstrate that this approach improves sample efficiency. When using only 0.5% of available labeled data (880 samples), their MoCo-v3 approach achieved an impressive 50.4% classification accuracy, compared to just 14.9% for ImageNet pre-training and 9.1% for random initialization. This represents a remarkable 35-41% accuracy improvement in extremely low-data regimes. The performance gap is particularly striking at 1% labeled data, where MoCo-v3 reached 53.1% accuracy while transfer learning required 5% of the data (10× more labeled samples) to achieve similar performance. The authors approach consistently outperformed both SimCLR variants from [73] and transfer learning, with advantages as labeled data increased, demonstrating a 1.3% improvement even with 90% labeled data. Their ablation studies showed that using all five RF-specific transformations was crucial, with their SimCLR-5TX implementation outperforming the AWGN-only version from [73] by 12.2% at 0.5% data. This comprehensive evaluation conclusively demonstrates the effectiveness of their enhanced augmentation strategy and momentum-based contrastive learning for RF signal classification under label-scarce conditions.

Building 6G Radio Foundation Models with Transformer Architectures. Recent work by Aboulfotouh et al. [1] presents a novel direction in RFML by introducing the first self-supervised pretrained foundation model designed for dense prediction tasks on spectrogram data. Their approach leverages a ViT architecture, pretrained using a generative self-supervised objective called Masked Spectrogram Modeling (MSM), which is inspired by masked image modeling strategies in computer vision [21].

The core methodology is illustrated in Figure 14. The upper section shows the pretraining phase, where spectrograms are divided into fixed-size patches and 75% of them are masked. The ViT encoder learns lower dimensional representations by

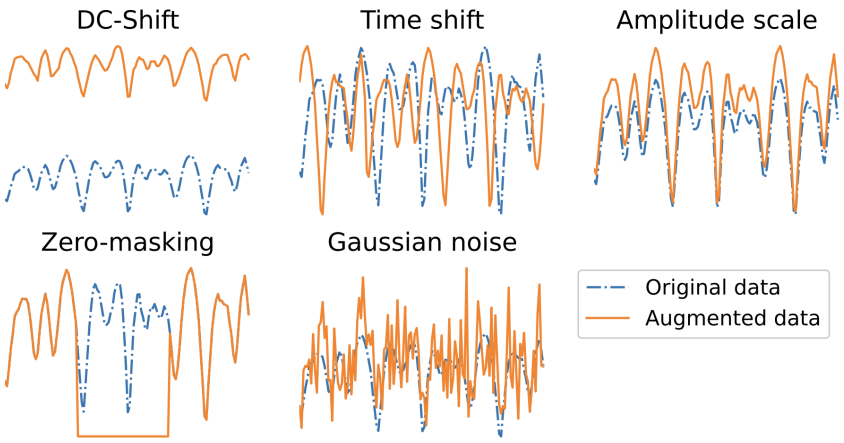


Figure 13: The five signal augmentations implemented by Kemal et al. [12] for contrastive learning on RF signals: DC shift, time shift, amplitude scaling, zero-masking, and AWGN. These domain-specific transformations create different views of the same signal while preserving semantic information.

reconstructing the masked patches using the remaining visible patches. A decoder reconstructs the full spectrogram, but is discarded after pretraining. In the lower section, the pretrained encoder is transferred to downstream tasks such as spectral segmentation. The encoder weights are frozen, and a task-specific head is fine-tuned on labeled data.

While this represents the first published attempt to pretrain an RFML model using SSL for spectrogram-based dense prediction tasks, the evaluation has critical limitations. The authors only pretrain the ViT via their MSM on radio spectrograms from scratch, without evaluating against established computer vision backbones pretrained on ImageNet using MAE or other SSL methods. This fundamental omission makes it impossible to determine whether the domain-specific pretraining effort actually provides benefits over existing pretrained models, or whether the observed improvements stem solely from having any pretraining versus no pretraining at all. Additionally, the dataset includes only three signal classes (noise, LTE, NR), which does not reflect the complexity of real-world RF environments with heterogeneous emitters and overlapping transmissions.

Compared to contrastive SSL approaches in RFML [12, 73], which targeted signal classification and rely on domain-specific augmentations, this generative approach avoids manual augmentation design and could potentially scale better with the amount of unlabeled data. As such, it sets a foundation for future research in RFML on spectrogram data, though the lack of proper baselines limits its impact.

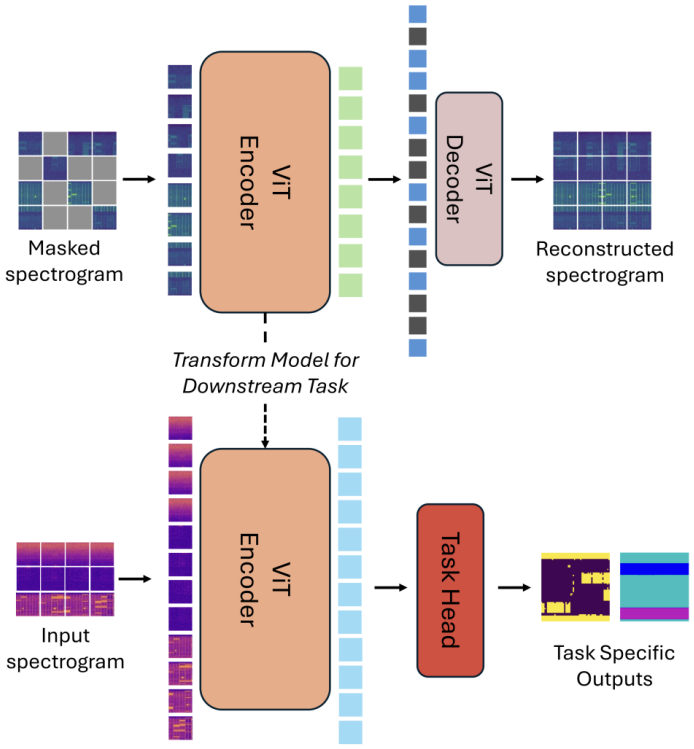


Figure 14: Overview of the self-supervised foundation model framework introduced by Aboulfotouh et al. [1]. Top: Pretraining with Masked Spectrogram Modeling. Bottom: Transfer to downstream tasks such as spectral segmentation.

3.3. Identified Gaps in the Literature

Although recent work has advanced wideband RF signal recognition and explored SSL in RFML, three key limitations persist:

First, there is no established self-supervised, domain-specific pretraining strategy for *wideband* signal recognition. Existing SSL work in RFML largely targets narrowband modulation classification, rather than dense tasks that jointly detect, localize, and classify signals across time–frequency spectrograms. As a consequence, current systems are typically initialized from scratch or rely on computer-vision backbones whose statistics do not reflect the RF domain. Existing work only covers related spectral segmentation tasks [1], but does not address the specific challenges of wideband signal recognition, such as heterogeneous emitters and overlapping transmissions. Furthermore, existing SSL approaches in RFML fail to establish proper evaluation baselines by comparing domain-specific pretraining against established computer vision models pretrained on ImageNet, making it unclear whether observed improvements stem from domain adaptation or simply from having any pretraining at all.

Second, realistic and densely annotated wideband datasets remain scarce. Public resources with rich time–frequency annotations, heterogeneous emitters, overlapping transmissions, and realistic channel and impairment conditions are limited. With appropriate SSL pretraining, the need for large labeled datasets could be substantially reduced.

Third, robustness and generalization remain insufficiently explored in the literature. Few studies systematically evaluate cross-domain transfer, such as adapting models from synthetic to real-world deployments, or assess performance under extreme low-SNR conditions and interference-heavy environments. There is also a lack of understanding regarding how well models generalize across different features of the electromagnetic spectrum, whether they can transfer from one signal type to another, or adapt across different frequency bands.

The core objective of this thesis is to address the first gap by developing and evaluating a self-supervised, domain-specific pretraining strategy tailored to wideband signal recognition. In doing so, it also contributes toward mitigating the data scarcity issue and evaluating how pretraining can improve robustness and generalization of wideband signal recognition models.

4. Methodology

This chapter details the methodological framework developed to evaluate domain-specific self-supervised learning for wideband signal recognition. We begin by formally defining the wideband signal recognition problem and outlining our two-stage approach combining SSL pretraining with supervised fine-tuning. The chapter describes the three datasets used in our evaluation: TorchSig Narrowband for pretraining, TorchSig Wideband variants for data efficiency assessment, and RadDet for cross-domain transfer evaluation. We then present the spectrogram preprocessing pipeline that transforms raw In-phase and Quadrature (IQ) data into computer vision-compatible representations, followed by detailed descriptions of our three SSL methods—DenseCL, VICRegL, and MAE—adapted for RF spectrograms. Finally, we outline the training protocols for both pretraining and fine-tuning phases, and define the evaluation metrics used to assess model performance across different scenarios.

4.1. Problem Formulation

Wideband signal recognition is the task of jointly detecting, localizing, and classifying multiple co-occurring signals in a broad radio-frequency band [66]. Unlike narrowband scenarios, where signals are pre-isolated by channelization or prior knowledge of center frequency and bandwidth, wideband settings must account for unknown signal placements in time and frequency, varying bandwidths, and potential overlap or interference among signals[7].

For a formal definition of the problem we will closely follow [7]. Let $r(t) = [r_1, r_2, \ldots, r_T]^\top$ with $r_i \in \mathbb{C}$ denote a wideband time-domain signal consisting of T complex-valued time samples. Each sample r_i is a complex number of the form $r_i = I_i + jQ_i$, where I_i and Q_i represent the IQ components, respectively. This complex baseband form enables the signal to encode both amplitude and phase information, which is essential for tasks such as modulation classification and signal recognition.

We assume $r(t)$ may contain zero, one, or multiple distinct signals $\{s_n(t)\}_{n=1}^{N}$ occurring at different times/frequencies or overlapping within the capture. After transforming $r(t)$ into a time-frequency representation through spectrogram computation, our objective is to learn a function that maps spectrograms to signal detections. This function can be decomposed into two main components: a backbone encoder ϕ_θ that extracts feature representations from the spectrogram, and a detection head h_ψ that processes these features to produce the final predictions:

$$f_{\theta,\psi} : h_\psi(\phi_\theta(\text{Spectrogram}(r(t)))) \rightarrow \{\hat{b}_1, \hat{b}_2, \ldots, \hat{b}_M\} \tag{6}$$

where M represents the number of detected signals, θ are the backbone parameters, ψ are the detection head parameters, and each predicted output $\hat{b}_m$ is a tuple (c, t_c, f_c, d, B) encoding both the signal classification and its time-frequency localization within the spectrogram:

- $c \in \mathcal{C}$ is the class label from a predefined set of signal types $\mathcal{C}$
- $t_c \in \{0, 1, ..., T-1\}$ denotes the temporal center of the signal in sample indices
- $f_c \in [0, F_s/2]$ represents the signal's center frequency in Hz, where F_s is the sampling frequency
- $d > 0$ specifies the duration of the signal in samples
- $B > 0$ indicates the bandwidth of the signal in Hz

This definition directly maps to object detection in computer vision, where bounding boxes are typically represented as (c, x_c, y_c, w, h) with class and localization in pixel-space.

As discussed in Section 2.2.3, a key challenge in this domain is the limited availability of labeled real-world RF datasets. Annotating wideband signals is resource-intensive and often constrained by regulatory and privacy limitations. To address this, we adopt SSL to pretrain the backbone encoder ϕ_θ using large collections of unlabeled spectrograms.

SSL aims to learn meaningful feature representations by pretraining the backbone parameters θ through solving a *pretext task* that does not require human annotations. During this pretraining phase, the backbone encoder, along with SSL-specific components such as projection heads, is trained end-to-end on unlabeled spectrogram data. The goal is to learn generalizable feature representations that capture the underlying structure and patterns in the data, which can then be effectively transferred to downstream tasks.

After pretraining, the learned backbone encoder ϕ_θ, with pretrained parameters, is integrated with a randomly initialized detection head h_ψ to form the complete wideband signal recognition model. The SSL-specific components, such as projection heads, are discarded during this transfer. The complete model is then fine-tuned on labeled data for the supervised task of detecting, localizing, and classifying multiple co-occurring signals.

This two-stage approach, SSL pretraining followed by supervised fine-tuning, enables effective learning of transferable representations from large amounts of unlabeled spectrogram data, which can then be adapted to the specific requirements of wideband signal recognition.

4.2. Datasets

This thesis leverages three distinct datasets tailored to address different research objectives: the *TorchSig Narrowband dataset* for SSL pretraining, several *TorchSig Wideband datasets* for data-efficiency evaluations, and the *RadDet dataset* for assessing cross-domain transferability.

Both the TorchSig Narrowband and TorchSig Wideband datasets used in this work are generated using the TorchSig library (version 1.1.0) [6, 56], which provides

extensive flexibility in creating realistic RF environments, signal classes, bandwidth variations, and impairment conditions. Due to significant updates in TorchSig, direct comparison to earlier benchmarks [7] is no longer applicable. Since the primary goal is to evaluate relative performance improvements from SSL, we focus on the new datasets generated with the updated library and do not attempt to replicate and compare to previous results.

The RadDet dataset [25] is an independent, publicly available resource specifically designed for wideband radar spectrum recognition. It is synthetically generated using existing narrowband radar classification datasets and transformed into spectrograms.

Following sections describe the datasets in detail, including their generation parameters and characteristics.

4.2.1. Narrowband Dataset

The narrowband dataset acts as the foundational dataset for SSL pretraining, analogous to ImageNet in computer vision. It consists of 50,000 synthetic samples, each with 262,144 complex I/Q samples recorded at a sampling rate of 10 MHz, equating to approximately 26.2 ms of capture duration per sample.

This dataset is specifically designed for clarity in SSL representation learning, with each sample containing exactly one signal to avoid the complexity of overlapping transmissions during pretraining. The signals vary in bandwidth between 200 kHz and 2 MHz, representative of typical narrowband communication scenarios. To enhance robustness, Level 2 impairments including IQ imbalance, phase offset, and fading are applied to simulate realistic channel conditions. The signal-to-noise ratio ranges between 5 and 50 dB to encourage robust feature extraction while maintaining sufficient signal quality for meaningful representation learning.

The dataset includes all signal types available in the TorchSig library, organized into 57 distinct classes across 10 modulation families. A full list of signal types is provided in Appendix A.1.

For the usage of pretraining the dataset is converted into spectrograms using the preprocessing pipeline described in Section 4.3. The resulting spectrograms have a resolution of 512×512 pixels, providing a square format that is compatible with standard computer vision models.

Detailed parameters are summarized in Table 1.

4.2.2. Wideband Datasets

To assess the effectiveness of SSL pretraining across different data regimes, multiple wideband datasets with varying numbers of training samples (1k, 5k, 10k, 25k, 50k, and 100k) were synthesized. All datasets use a common validation set of 10,000 samples extracted from the largest (100k) dataset, enabling fair comparisons across experiments.

Table 1: Parameters for the TorchSig Narrowband Dataset

Parameter	**Value**
Number of Samples	50,000
Sample Rate	$10\,\mathrm{MHz}$
IQ Samples per Sample	262,144
Signals per Sample	1
SNR Range	5–50 dB
Impairment Level	2
Signal Classes	57
Class Distribution	Uniform
Bandwidth Range	$200\,\mathrm{kHz}$–$2\,\mathrm{MHz}$
FFT Size	512
Signal Duration	$1\,\mathrm{ms}$–$26.2\,\mathrm{ms}$
Seed	123456789

Each wideband sample comprises 262,144 complex I/Q samples captured at $100\,\mathrm{MHz}$, providing about $2.62\,\mathrm{ms}$ of data. The wideband datasets present significantly more challenging scenarios compared to the narrowband pretraining data. Signal density ranges from 1 to 5 signals per sample, including overlapping transmissions that reflect real-world spectrum occupancy patterns. Individual signal bandwidths span $5\,\mathrm{MHz}$ to $20\,\mathrm{MHz}$, representing typical wideband communication scenarios. The SNR conditions are more demanding, ranging from 0 to 50 dB, including signals at the noise floor that test the model's ability to detect weak transmissions in challenging environments.

Similar to the narrowband dataset, the wideband datasets utilize all available TorchSig signal types across multiple communication families (see Appendix A.1 for the complete list). This diversity ensures that the evaluation encompasses a wide range of signal characteristics and modulation schemes commonly encountered in real-world wideband scenarios. For training and evaluation, the 57 distinct signal classes are grouped into 10 broader categories based on their modulation families rather than using individual signal types.

To detect and localize signals in the spectrograms, the wideband datasets are transformed into 512×512 pixel spectrograms using the preprocessing pipeline described in Section 4.3. This transformation preserves the time-frequency characteristics of the signals while providing a format compatible with computer vision models.

Detailed parameters are presented in Table 2.

4.2.3. RadDet Dataset

The RadDet dataset [25] is utilized to evaluate cross-domain transfer capabilities of the learned RF representations. Specifically designed for wideband radar spectrum

Table 2: Parameters for the TorchSig Wideband Datasets

Parameter	**Value**
Number of Samples	1k, 5k, 10k, 25k, 50k, 100k
Validation Set Size	10,000 samples (subset of 100k)
Sample Rate	100 MHz
IQ Samples per Sample	262,144
Signals per Sample	1–5
SNR Range	0–50 dB
Impairment Level	2
Signal Classes	57 (10 families)
Class Distribution	Uniform
Bandwidth Range	5 MHz–20 MHz
FFT Size	512
Signal Duration	0.16 ms–2.62 ms
Seed	123456789

detection in real-time scenarios across varying operational conditions.

RadDet introduces 11 radar classes, including 6 new LPI polyphase codes (P1, P2, P3, P4, Px, Zadoff-Chu) and a wideband frequency-modulated continuous wave (FMCW), all coexisting across a 500 MHz frequency band. The dataset contains a total of 40,000 radar frames divided into training (20,000 frames), validation (14,000 frames), and test (6,000 frames) sets.

To investigate wideband spectrum detection in different scenarios, RadDet provides two distinct radar environments. The sparse dataset (RadDet-1T) provides at most a single radar instance per frame with a 50% probability of radar presence. The dense dataset (RadDet-9T) contains up to 9 radar instances per frame where the probability of background noise-only frames is 10%, representing a dense radar environment.

For this thesis, we utilize the RadDet-9T subset in 512×512 resolution to evaluate transfer learning performance. The SNR is sampled from a uniform distribution ranging from -20 to 20 dB at 8 dB resolution intervals, generating more than 6,500 unique signals per SNR level. This challenging setup with multiple overlapping radar signals per frame provides a realistic testbed for evaluating how well SSL-pretrained models transfer from communication signals to radar signals.

This dataset comes as a precomputed set of spectrograms, eliminating the need for additional preprocessing steps.

A summary of RadDet's parameters is provided in Table 3.

Table 3: Summary of the RadDet Dataset

Attribute	Specification
Number of Samples	40,000 spectrograms
Training Samples	20,000 frames
Validation Samples	14,000 frames
Test Samples	6,000 frames
Frequency Bandwidth	$500\,\mathrm{MHz}$
Signal Classes	11
SNR Range	-20 to 20 dB
Signals per Sample	0–9
Resolution	512×512
Background Probability	10% (noise-only frames)

4.3. Spectrogram Preprocessing

This section describes the preprocessing steps applied to convert raw IQ data into spectrograms suitable for computer vision-based SSL methods. The transformation pipeline is designed to produce high-quality, normalized spectrogram representations that preserve signal characteristics while being compatible with standard image processing techniques.

The spectrogram transformation is implemented using PyTorch's torchaudio library, specifically the `torchaudio.transforms.Spectrogram` [49] class. This choice ensures compatibility with the broader PyTorch ecosystem and enables efficient GPU acceleration during training.

We follow [7] and use these configuration parameters for the spectrogram transformation:

- **FFT size**: 512 points for balanced time-frequency resolution
- **Window function**: Blackman window (length 512) to reduce spectral leakage
- **Hop length**: 512 samples (no overlap) ensuring exactly 512 time frames
- **Power spectrum**: Magnitude squared (`power=2`) for proper dB scaling later
- **Two-sided spectrum**: (`onesided=False`) to preserve complex IQ frequency content

$$S = \text{Spectrogram}(r(t), parameters) \tag{7}$$

For the wideband dataset with 262,144 IQ samples, this configuration produces spectrograms with dimensions 512×512 pixels, providing a convenient square format that matches standard computer vision model expectations. To ensure consistent

input characteristics across different signals and improve training stability, we apply a multi-stage normalization process.

Each spectrogram is first normalized by its maximum value to prevent numerical overflow:

$$S_{\text{norm}} = \frac{S}{\max(S) + \epsilon} \tag{8}$$

where $\epsilon = 10^{-12}$ prevents division by zero.

We apply FFT shift along the frequency axis and vertical flip to align with conventional spectrogram visualization where lower frequencies appear at the bottom and higher frequencies at the top:

$$S_{\text{shifted}} = \text{FFTShift}(S_{\text{norm}}) \tag{9}$$

$$S_{\text{flipped}} = \text{Flip}(S_{\text{shifted}}) \tag{10}$$

where FFTShift rearranges the frequency bins from $[0, \pi, -\pi, 0]$ to $[-\pi, 0, \pi]$ ordering, and Flip vertically flips the spectrogram so that lower frequencies appear at the bottom of the image.

The power spectrum is converted to decibel scale using:

$$S_{\text{dB}} = 10 \log_{10}(S_{\text{flipped}} + \epsilon) \tag{11}$$

This compression of the dynamic range helps balance weak and strong signal components.

Finally, we apply linear scaling to map values to the range [0,1]:

$$S_{\text{final}} = \frac{S_{\text{dB}} - \min(S_{\text{dB}})}{\max(S_{\text{dB}}) - \min(S_{\text{dB}}) + \epsilon} \tag{12}$$

The resulting spectrograms are single-channel grayscale images with pixel values normalized to $[0, 1]$. This format is compatible with standard computer vision architectures and allows us to leverage existing SSL methods designed for image data. All transformations are applied consistently during both pretraining and downstream fine-tuning to ensure feature compatibility across training phases.

4.4. Self-Supervised Learning Methods

This subsection details the specific SSL methods employed to pretrain the backbone networks described previously. The primary goal of applying these SSL techniques is to learn robust and transferable feature representations directly from large quantities of unlabeled spectrogram data. By leveraging these self-supervised pretraining strategies, we aim to enhance the performance of the downstream wideband signal recognition task, particularly when labeled data is scarce. We explore three distinct SSL paradigms: a local contrastive method (DenseCL), a non-contrastive method with regularization (VICRegL), and a masked autoencoder approach (MAE).

4.4.1. DenseCL

DenseCL [65] extends the global contrastive learning framework of MoCo [22] by moving the contrastive objective from a single global embedding per image to local embeddings at the feature map level, respectively a patch of pixels. The DenseCL pipeline is visualized in Figure 15. The first step is to create two different augmented views of the same input spectrogram: a query view q and a key view k_+. These views are generated by applying different data augmentations to the original spectrogram, ensuring that while they represent the same underlying signal, they have different visual characteristics that the model must learn to associate as semantically equivalent representations. Following the MoCo framework, DenseCL maintains a large memory bank of negative samples k_- from previously processed spectrograms. This memory bank enables contrastive learning by providing a diverse set of negative examples without requiring them all to be present in the current batch, significantly improving training efficiency.

The core innovation of DenseCL is its dual global and local contrastive learning approach. It employs two parallel projection heads: a global projection head that produces a single feature vector per view, and a dense projection head that generates spatial feature maps preserving local information.

For global contrastive learning, following the InfoNCE [61] loss, the loss for each encoded query q is:

$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+/\tau)}{\exp(q \cdot k_+) + \sum_{k_-} \exp(q \cdot k_-/\tau)} \tag{13}$$

where k_+ is the positive key from the corresponding key view of the same spectrogram, k_- are negative keys sampled from the memory bank, and τ is a temperature parameter. The loss function pulls q and k_+ closer together in the embedding space while pushing q away from all k_-, effectively learning a global representation that captures the overall characteristics of the spectrogram.

For dense contrastive learning, DenseCL extends this concept to local features. Each spatial location in the dense feature maps from the query view performs contrastive learning by matching corresponding local regions in the key view. The dense projection head generates spatial feature maps of dimensions $S_H \times S_W$, where S_H and S_W represent the height and width of the feature maps, respectively. For simplicity, the authors use the notation S^2 to represent the total number of spatial locations, assuming square feature maps where $S_H = S_W = S$. The dense contrastive loss is defined as:

$$\mathcal{L}_r = \frac{1}{S^2} \sum_s -\log \frac{\exp(r^s \cdot t^s_+/\tau)}{\exp(r^s \cdot t^s_+) + \sum_{t^s_-} \exp(r^s \cdot t^s_-/\tau)} \tag{14}$$

where r^s denotes the s-th out of S^2 encoded queries from the dense feature maps of the query view, t^s_+ is the positive key from the corresponding spatial location in the key view, and t^s_- are negative keys sampled from the memory bank.

The correspondence between spatial locations across the query and key views is found by computing cosine similarity between backbone feature maps and selecting the most similar local regions. Specifically, for each feature vector f_i in the query view, the correspondence with the key view is obtained by:

$$c_i = \arg\max_j \text{sim}(f_i, f'_j) \tag{15}$$

where $\text{sim}(u, v) = u^T v/(\|u\|\|v\|)$ is the cosine similarity.
The total loss combines both global and dense objectives:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_q + \lambda\mathcal{L}_r \tag{16}$$

where $\lambda \in [0, 1]$ is a hyperparameter that balances the contribution of the global contrastive loss $\mathcal{L}_q$ and the dense contrastive loss $\mathcal{L}_r$. When $\lambda = 0$, the method reduces to standard global contrastive learning, focusing only on learning global representations of the entire spectrogram. When $\lambda = 1$, the method performs purely dense contrastive learning. A value of $\lambda = 0.5$ is often used to balance the two objectives, allowing the model to learn both global and local representations simultaneously.

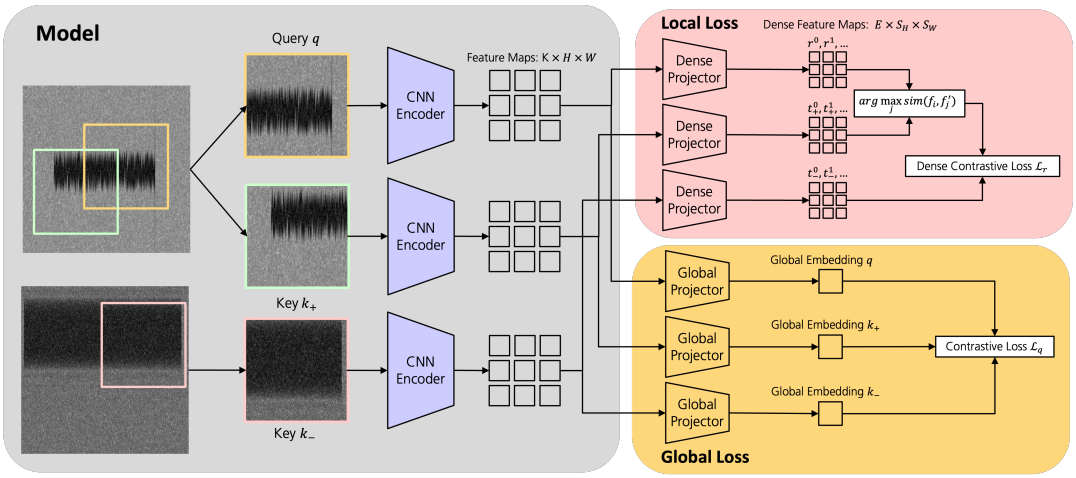


Figure 15: Conceptual illustration of DenseCL applied to spectrogram data. The method encourages local features from corresponding regions (patches or pixels) in two augmented views of the same input spectrogram to be aligned, promoting fine-grained and spatially-aware representations.

The DenseCL model was implemented using the Lightly SSL library [36], which provides modular building blocks, like loss functions, projection heads and image transformations, for self-supervised learning methods. This framework enabled efficient implementation, while maintaining compatibility with standard PyTorch training pipelines.

In order to generate the query and key views required for contrastive learning, we apply a series of data augmentations to the input data. First, the RF-specific augmentations, discussed in Section 4.4.4, are applied to the raw IQ data.

On the resulting spectrograms, *RandomResizedCrop* [48] is additionally applied to generate the query and key views required for contrastive learning. This operation combines resampling and cropping in both time and frequency dimensions in a computationally efficient manner,simulating the effect of observing the same signal at different temporal scales and frequency resolutions. While similar transformations, such as temporal windowing and frequency-domain filtering could theoretically be applied directly to the IQ data, operating on spectrograms offers significant computational advantages. RandomResizedCrop on spectrograms provides direct control over time-frequency regions of interest and is orders of magnitude faster than equivalent operations on raw IQ data, making it practical for large-scale SSL training. Traditional image augmentations such as color jittering and gaussian blur were not included in the pipeline, as spectrograms are single-channel intensity maps where such transformations would not preserve the underlying signal characteristics.

4.4.2. VICRegL

VICRegL [5] is a non-contrastive self-supervised learning method that extends the principles of VICReg [4] to learn rich local feature representations from spectrograms. Unlike contrastive methods that rely on explicit positive and negative pairing, VICRegL regularizes the embeddings of augmented views of an input to prevent representational collapse while encouraging the learning of useful features.

VICRegL builds upon the VICReg loss function, which consists of three complementary terms that work together to learn meaningful representations without negative samples. The VICReg loss ℓ applied to pairs of feature vectors is defined as:

$$\ell(u, v) = \lambda s(u, v) + \mu[v(u) + v(v)] + \nu[c(u) + c(v)] \tag{17}$$

where λ, μ, and ν are hyperparameter that control the relative importance of each term:

- **Invariance term** $s(u, v)$ ensures that corresponding features from augmented views of the same spectrogram remain similar, weighted by λ
- **Variance terms** $v(u)$ and $v(v)$ maintain sufficient variability in each feature dimension to prevent representational collapse, weighted by μ
- **Covariance terms** $c(u)$ and $c(v)$ decorrelate different feature dimensions to encourage diverse and informative representations, weighted by ν

A detailed mathematical overview of these loss components is provided in the original VICReg paper [4]. Following the original VICReg formulation, these hyperparameter are set to $\lambda = 25$, $\mu = 25$, and $\nu = 1$. The higher weights for the invariance

and variance terms, λ and μ, emphasize the importance of learning consistent representations while preventing collapse, while the lower weight for the covariance term, ν, provides a regularization effect to decorrelate features without overwhelming the primary objectives.

VICRegL operates on spectrograms and creates two different augmented views of the same input. Two projection networks are used: a local projector $h^l_\phi : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^{D\times H\times W}$ that embeds the feature maps while preserving spatial structure, and a global expander $h^g_\psi : \mathbb{R}^C \to \mathbb{R}^D$ that maps globally pooled representations to embeddings. For feature maps y and y' from the two views, the local projector produces spatial embeddings $z = h^l_\phi(y)$ and $z' = h^l_\phi(y')$, while the global expander produces pooled embeddings $z_\oplus = h^g_\psi(y_\oplus)$ and $z'_\oplus = h^g_\psi(y'_\oplus)$. The full VICRegL pipeline is visualized in Figure 16.

Similar to DenseCL, VICRegL matches local and global feature vectors between the two augmented views. The global feature vectors follow the same processing as in VICReg, where the global representations $z_\oplus$ and $z'_\oplus$ are compared using the VICReg loss $\ell(z_\oplus, z'_\oplus)$. For local matching, VICRegL uses two strategies: *location-based matching* and *feature-based matching*.

Location-based matching matches feature vectors based on their spatial correspondence in the original spectrogram by tracking the transformations applied during augmentation to maintain correspondence between views. For each feature vector z_p at position p in the query view, the method computes the absolute position in the original spectrogram that this feature represents, then finds the corresponding position in the key view that represents the same time-frequency region. The loss for spatially matched pairs is defined as:

$$\mathcal{L}_s(z, z') = \sum_{p\in P} \ell(z_p, z'_{NN(p)}) \tag{18}$$

where P represents all spatial positions in the feature map, $NN(p)$ denotes the spatially closest coordinate to p based on the transformation between views, and ℓ is the VICReg loss applied to the matched pair.

Feature-based matching matches feature vectors based on their similarity in the embedding space, capturing long-range interactions not covered by spatial matching:

$$\mathcal{L}_d(z, z') = \sum_{p\in P} \ell(z_p, NN'(z_p)) \tag{19}$$

where $NN'(z_p)$ denotes the closest feature vector to z_p in terms of ℓ^2-distance in the embedding space.

To eliminate poorly matched pairs, only the top-γ pairs with the smallest distances are retained for both matching strategies. This filtering prevents the inclusion of mismatched feature vectors that likely represent different signal components or noise.

The complete VICRegL loss combines both local matching strategies with the global VICReg criterion:

$$\mathcal{L}(z, z') = \alpha\ell(z_{\oplus}, z'_{\oplus}) + (1-\alpha)[\mathcal{L}_s(z, z') + \mathcal{L}_s(z', z) + \mathcal{L}_d(z, z') + \mathcal{L}_d(z', z)] \quad (20)$$

where $\alpha \in [0, 1]$ controls the trade-off between global and local learning. Following the original VICRegL paper [5], we set $\alpha = 0.75$ to emphasize local feature learning while still maintaining global context. The top-γ filtering retains approximately 75% of feature pairs, effectively removing poorly matched correspondences while preserving meaningful local relationships in the time-frequency domain.

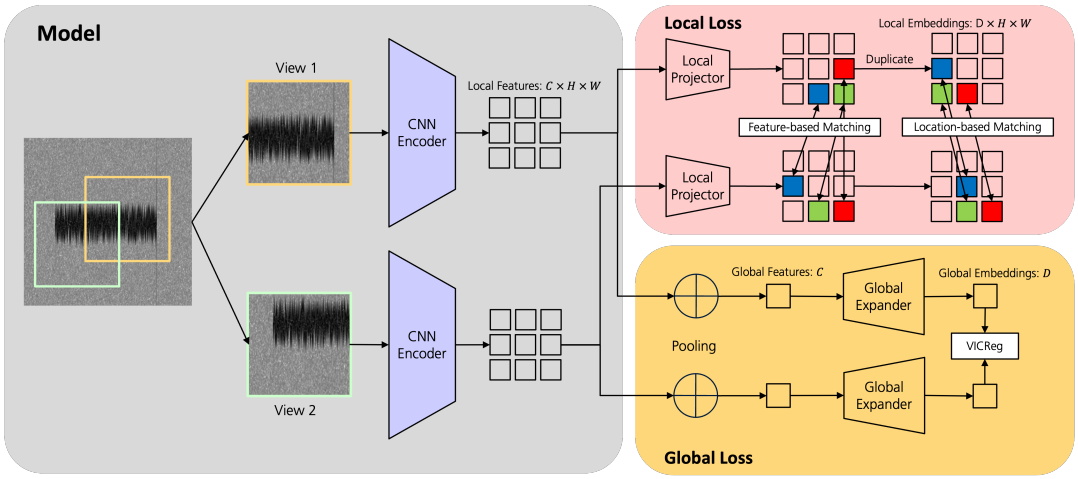


Figure 16: Conceptual illustration of VICRegL applied to spectrogram data. The method matches local features between two augmented views using both spatial proximity and embedding similarity, applying VICReg regularization to each matched pair while also learning global representations.

The VICRegL model was implemented using the Lightly SSL library [36]. By using the provided building blocks for loss functions, projection heads, and data transformations, we were able to efficiently implement the VICRegL method while ensuring compatibility with standard PyTorch training pipelines. The data augmentation pipeline was extended to include RF-specific augmentations, as described in Section 4.4.4, followed by a RandomResizedCrop operation to generate the two augmented views required for VICRegL. Any color jittering or gaussian blur operations were omitted, as they are not applicable to spectrograms.

The key advantage of VICRegL over contrastive methods like DenseCL is its elimination of negative sampling and memory banks while still learning detailed local representations. This makes it computationally efficient and particularly suitable for large-scale spectrogram datasets where maintaining extensive memory banks of negative samples becomes impractical.

4.4.3. MAE

MAE [21] is a self-supervised learning method that learns representations by reconstructing randomly masked patches of the input spectrogram. Unlike contrastive methods that require multiple augmented views, MAE operates on a single input and learns by predicting missing information, making it conceptually simple and computationally efficient.

The MAE approach follows an asymmetric encoder-decoder architecture, as illustrated in Figure 17. The encoder is based on a ViT architecture but processes only the visible unmasked patches of the spectrogram. Following the ViT paradigm, the input spectrogram is divided into regular non-overlapping patches, which are then linearly projected with added positional embeddings. However, unlike standard ViT, the MAE encoder operates on only a small subset of patches of typically 25% of the total, with masked patches completely removed from the input sequence, no mask tokens are used during encoding.

The lightweight decoder reconstructs the original spectrogram from two inputs: the encoded visible patches from the encoder and learnable mask tokens representing the missing patches. Each mask token is a shared, learned vector that indicates the presence of a missing patch to be predicted. Positional embeddings are added to all tokens in the full sequence to provide spatial location information to the mask tokens. The decoder consists of a series of Transformer blocks but is designed to be much smaller than the encoder, requiring less than computation per token compared to the encoder.

For spectrogram data, patches correspond to rectangular time-frequency regions. Random sampling with a high masking ratio of typically 75%, is applied, removing three-quarters of the patches following a uniform distribution. This aggressive masking eliminates redundancy and prevents the model from solving the reconstruction task through simple extrapolation from neighboring visible patches, forcing it to learn rich representations that capture both local signal patterns and global time-frequency relationships.

The MAE training objective is to minimize the mean squared error between the original and reconstructed pixel values for the masked patches:

$$\mathcal{L}_{\text{MAE}} = \frac{1}{|M|} \sum_{p \in M} \|x_p - \hat{x}_p\|_2^2 \tag{21}$$

where M represents the set of masked patches, x_p is the original patch, and $\hat{x}_p$ is the reconstructed patch. The loss is computed only on masked regions, similar to BERT's masked language modeling objective.

For our implementation, we adapt the standard MAE architecture to work with single-channel spectrogram data. Following the original MAE paper [21] and recent work by Aboulfotouh et al. [1], we resize the 512×512 spectrograms to 224×224 pixels to match standard input dimensions. The ViT-B/16 encoder processes 16×16 pixel patches from these resized spectrograms, resulting in 14×14 = 196 patches total. The

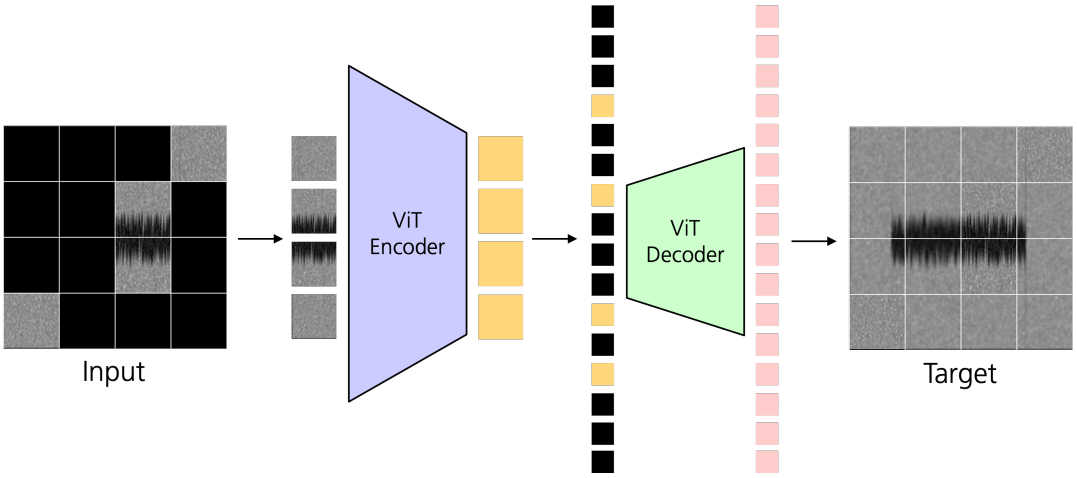
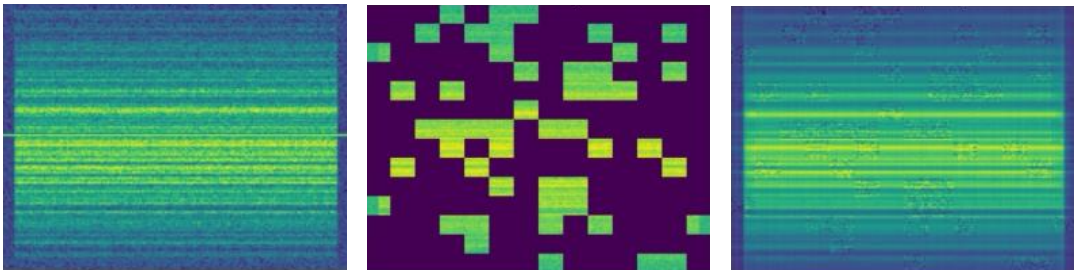


Figure 17: Conceptual illustration of MAE applied to spectrogram data. The method randomly masks patches of the input spectrogram and trains the model to reconstruct the missing regions, learning rich representations of time-frequency patterns without requiring augmented views or negative samples.

asymmetric design significantly reduces computational cost during training since the encoder processes only the visible subset of patches while the lightweight decoder handles the complete reconstruction task.

The MAE model was implemented using the Lightly SSL library [36], similar to the other SSL methods. By leveraging the provided building blocks for masked autoencoder architectures, loss functions, and data transformations, we were able to efficiently implement the MAE method while ensuring compatibility with standard PyTorch training pipelines. During pretraining, only the reconstruction loss is used, with no additional augmentations applied to the input spectrograms beyond the masking operation itself.

Figure 18 visualizes the MAE reconstruction process for spectrogram data. The left panel shows the original spectrogram, representing the full time-frequency content of the signal. The middle panel displays the same spectrogram after random masking, where 75% of the patches have been removed—these regions are set to zero and provide no information to the model. The right panel shows the output of the MAE model, which reconstructs the missing regions based only on the visible patches. The reconstructed spectrogram closely matches the original, recovering both global structure and fine signal details, but is noticeably blurrier than the original because the model must infer missing regions from limited context, and the pixel-wise loss encourages averaging over possible solutions, which smooths out fine details. This demonstrates that the MAE learns meaningful representations of RF signals, even when most of the input is missing.



(a) Original (b) Masked (c) Reconstructed

Figure 18: MAE reconstruction example from TorchSig Narrowband spectrogram. (a) The original spectrogram shows the full time-frequency content of the signal. (b) The masked spectrogram has 75% of its patches removed, providing only sparse information to the model. (c) The reconstructed spectrogram is generated by the MAE model using only the visible patches.

4.4.4. Data Augmentations

The augmentation strategy for our SSL methods combines RF-specific transformations with computer vision techniques to create meaningful contrastive views. The pipeline operates in two stages: first on the raw IQ data, followed by computer vision techniques on the resulting spectrograms.

For the IQ-level augmentations, we leverage TorchSig's [6] domain-specific transforms to simulate realistic channel conditions and hardware impairments. Our approach builds upon the foundational work of Kemal et al. [12], who identified key signal augmentations for contrastive learning on RF signals. These domain-specific transformations are applied differently on IQ data and create different views of the same signal while preserving semantic information. We extend this approach using the following TorchSig transforms:

- **AWGN**: Adds white gaussian noise to simulate varying SNR conditions
- **TimeReversal**: Reverses the time sequence with a specified probability, creating temporal variations while preserving signal characteristics, similar to a horizontal flip in images
- **SpectralInversion**: Inverts the frequency spectrum, simulating different receiver configurations and improving frequency-domain robustness, similar to a vertical flip in images
- **CutOut**: Randomly masks portions of the IQ data to simulate signal dropouts
- **MagnitudeScale**: Applies random scaling to the signal magnitude, simulating varying signal strengths

The choice of TimeReversal and SpectralInversion is motivated by their direct correspondence to fundamental image augmentations, horizontal and vertical flips, used extensively in computer vision SSL methods. This design philosophy ensures compatibility with established SSL frameworks while maintaining the domain-specific benefits of RF signal processing. By mimicking these standard image transformations at the IQ level, we create a bridge between RF signal processing and computer vision approaches.

Since this thesis focuses on SSL for wideband signal recognition, the augmented IQ data is transformed into spectrograms by using the transform described previously in 4.3. This transformation is applied to both the original and augmented IQ data, resulting in two distinct spectrogram views for each input signal. The spectrograms are then used as input to the SSL models.

Notably, we do not implement the time shift augmentation identified by Kemal et al. This is because our SSL methods rely on additional image-based transformations applied at the spectrogram level that inherently provide temporal variability. These transformations are described in the respective sections for each SSL method.

4.5. Training Protocol

After defining our backbone architectures and self-supervised learning (SSL) variants, we now turn to the training protocols used for both pretraining and downstream fine-tuning on the wideband signal recognition task.

All trainings and experiments were constructed on a single Ubuntu 24.04 LTS workstation equipped with an NVIDIA GeForce RTX 5090 GPU with 32 GB VRAM, an AMD Ryzen 9 9950X CPU and total of 64 GB system RAM.

Compared to the original SSL papers [5, 21, 65], this setup is significantly more limited in terms of available computational resources, which necessitated several adaptations to the training protocols. Especially the batch sizes were reduced to fit within the available GPU memory, which in turn required adjustments to the learning rate to maintain stable training dynamics. Despite these constraints, we carefully preserved the core principles and training dynamics of each SSL method to ensure valid comparisons and meaningful results.

The following sections detail the pretraining protocols for each SSL method and the subsequent fine-tuning process on the wideband signal recognition task.

4.5.1. Pretraining

The pretraining of the backbone networks is performed using the self-supervised methods described in the previous sections. The pretraining process involves training the backbone networks on large, unlabeled spectrogram datasets to learn rich feature representations that can be transferred to downstream tasks.

DenseCL. We used DenseCL to pretrain the ResNet-50 backbone using the TorchSig narrowband dataset. The training configuration follows the original DenseCL

approach [65] with several adaptations for our computational constraints and spectrogram data. An overview of the pretraining hyperparameter is provided in Table 4.

The optimizer used for DenseCL pretraining is Stochastic Gradient Descent (SGD) with momentum. Contrastive learning requires a large batch size to ensure sufficient negative samples for effective training. However, due to the limited GPU memory, the batch size was reduced to 64 from originally a total batch size of 512 across 8 GPUs. The memory bank, needed for storing negative samples, was also reduced from originally 65,536 to only 4096. This reduction made adjustments to the learning rate necessary to maintain stable training.

To improve optimization with small batch sizes, linear learning rate scaling [18] was applied, where the base learning rate of 0.03 is multiplied by the ratio of the batch size to 256 (i.e., $\text{LR}_{\text{eff}} = 0.03 \times \frac{\text{batch size}}{256}$, resulting in $\text{LR}_{\text{eff}} = 0.0075$ for a batch size of 64). Differential weight decay is utilized, with a decay rate of 1×10^{-4} applied to convolutional and linear layer weights, while biases and batch normalization parameters are exempted to prevent over-regularization. The learning rate schedule follows a cosine annealing strategy, decaying the learning rate to zero over 100 epochs, preceded by a 10-epoch linear warm-up phase to stabilize early training. The temperature parameter τ is set to 0.1, which is the default choice for DenseCL. The balance parameter λ is set to 0.5, allowing equal contribution from global and local contrastive losses.

Table 4: DenseCL Pretraining Hyperparameter

Parameter	**Value**
Epochs	300
Batch size	64
Optimizer	SGD
Learning rate	0.0075
Momentum	0.996
Weight decay	1×10^{-4}
Temperature (τ)	0.1
Balance (λ)	0.5
Memory bank size	4096
Warm-up epochs	10

Reducing the batch size and memory bank was a direct result of limited GPU memory and compute budget. These constraints reduce the number of negatives per query and samples per update, which increases InfoNCE gradient variance and makes the loss sensitive to mini-batch composition. Smaller batches also produce noisier batch-norm statistics. As a result, the loss in Fig. 19 shows clear step-to-step fluctuations and slower, more gradual decline than the original DenseCL setup, with the smoothed trace leveling off above the baseline. This behavior reflects undersampling of negatives rather than a failure of the method, improvements would

likely require larger effective batches and a larger queue.

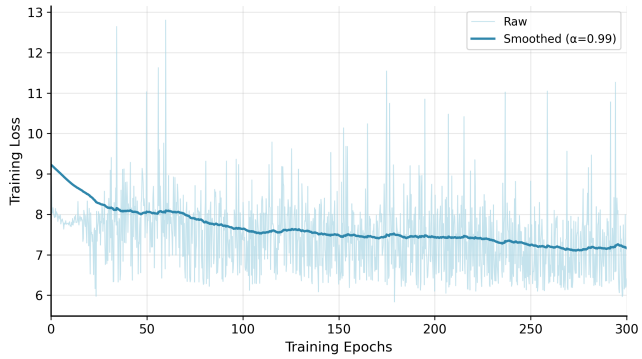


Figure 19: DenseCL training loss under compute constraints. The raw loss (light blue) shows high variance, while the EMA-smoothed curve (dark blue) reveals slower convergence and an early plateau due to limited negatives and small batches.

VICRegL. For VICRegL pretraining, we employ the same ResNet-50 backbone and the TorchSig narrowband dataset. Since VICRegL do not need negative sampling and memory banks, the computational requirements in terms of GPU-memory are significantly lower than DenseCL.

Following the original VICRegL [5] recommendations, the base learning rate is determined dynamically based on the global batch sizes: 0.8 for batch sizes $\leq$128, 0.5 for 256, 0.4 for 512, and 0.3 for larger batches.

The LARS [72] optimizer uses momentum of 0.9 and applies a differential weight decay strategy: 1×10^{-4} for applicable parameters (backbone, pooling, and projection heads) while excluding bias and batch normalization parameters. The learning rate schedule employs cosine annealing with 10-epoch warm-up, decaying from the learning rate to 1% of the base rate over the training duration.

The VICReg loss hyperparameter follow the original formulation with $\lambda = \mu = 25$ for invariance and variance terms, and $\nu = 1$ for covariance. The global-local balance parameter $\alpha = 0.75$ emphasizes local feature learning, while top-γ filtering retains 75% of feature correspondences.

Table 5: VICRegL Pretraining Hyperparameter

Parameter	**Value**
Epochs	300
Batch size	128
Optimizer	LARS
Learning rate	0.8
Momentum	0.9
Weight decay	1×10^{-4}
Warm-up epochs	10
Invariance (λ)	25
Variance (μ)	25
Covariance (ν)	1
Balance (α)	0.75
Top-γ filtering	75%

Unlike DenseCL, VICRegL fits our compute budget, it requires no negatives or memory bank, so GPU memory limits do not degrade the learning signal. With batch size 128, LARS optimizer, a 10-epoch warm-up, and cosine decay, optimization remained stable throughout training. As shown in Fig. 20, the loss decreases rapidly and then settles into a low-variance plateau, indicating smooth, predictable convergence.

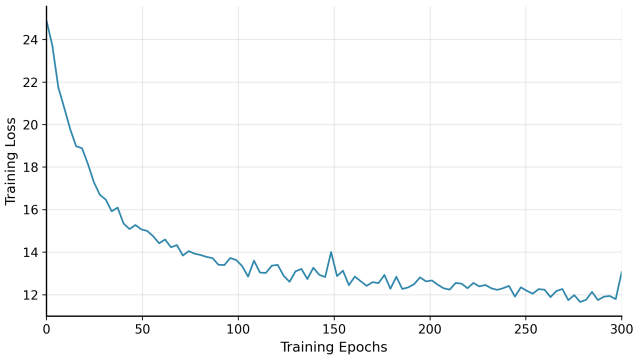


Figure 20: VICRegL training loss. The curve shows a rapid initial decrease followed by a stable, low-variance plateau, showing smooth optimization and reliable convergence without negative sampling.

MAE. The MAE pretraining uses a ViT-B/16 backbone on the narrowband dataset. The pretraining configuration is adapted from the original MAE paper [21] with adjustments for our computational constraints. An overview of the pretraining hyperparameters is provided in Table 6. The original MAE model was pretrained on ImageNet-1K, which contains 1.28 million images, while our dataset contains only

50,000 spectrograms. Due to poor downstream results when using only narrowband data, both TorchSig Narrowband and wideband datasets were used for pretraining, and results are reported separately for each to assess their impact. In order to fit more images into the GPU memory, the spectrograms were resized to 224×224 pixels, which is the standard input size for ViT-B/16. This way the model was trained with a batch size of 512 spectrograms and 600 epochs, instead of 1600 epochs and a batch size of 4096 reported on original MAE paper.The masking ratio was set to 0.75, meaning that three-quarters of the patches are randomly masked during training, this follows both the original MAE paper and the recent work by Aboulfotouh et al. [1].

Table 6: MAE Pretraining Hyperparameters

Parameter	**Value**
Epochs	600
Spectrogram size	224×224
Batch size	512
Decoder embedding dim	512
Decoder depth	8
Decoder attention heads	16
Mask ratio	0.75
Patch size	16
Optimizer	AdamW
Learning rate	3.0×10^{-4}
Weight decay	0.05
Scheduler	CosineWarmupScheduler
Warm-up epochs	40

The MAE training loss curves for both narrowband and wideband pretraining are shown in Fig. 21. Both runs exhibit a rapid initial decrease in loss, followed by a stable, low-variance plateau, indicating smooth and predictable convergence. The wideband run shows higher and more variable loss throughout training, likely due to the increased complexity and signal overlap in the wideband dataset. In contrast, the narrowband run is smoother and reaches a lower final loss, reflecting the simpler single-signal structure of the narrowband data. This behavior is consistent with the expected difficulty of reconstructing masked patches in more densely populated spectrograms.

Figure 21: MAE training loss for narrowband and wideband pretraining runs. The loss decreases rapidly at first, then settles into a low-variance plateau for both datasets. Wideband pretraining shows higher and more variable loss, likely due to increased signal density and overlapping transmissions, while narrowband training is smoother and reaches lower final loss values.

4.5.2. Fine-tuning

The fine-tuning stage adapts the general-purpose representations learned during self-supervised pretraining to the specific downstream task of wideband signal recognition. We leverage the Detectron2 framework [71], a well-established library for object detection, which provides a robust implementation of state-of-the-art detection architectures. Integrating our custom RF datasets required developing a dedicated dataset loader to handle spectrograms and their time-frequency annotations, as well as implementing a model converter to remap pretrained weights to match Detectron2's layer naming conventions.

For the Faster R-CNN architecture with ResNet-50 backbone and FPN, we employ SGD optimization with a base learning rate of 0.02, batch size of 16, momentum of 0.9, and weight decay of 1×10^{-4}. The training schedule incorporates a linear warm-up period over the first 1,000 iterations to stabilize early optimization, followed by multi-step learning rate decay that reduces the learning rate by a factor of 10 at 90% and 97% of total training iterations. Data augmentation includes random horizontal flipping and multi-scale training with input spectrograms resized to 512–1024 pixels on the shorter edge to improve model robustness across different signal bandwidths.

For ViTDet, a Faster R-CNN model with ViT-B/16 backbone and a SFP is used. The architecture requires different training configurations due to the distinct characteristics of vision transformers. High-resolution spectrograms of 1024×1024 pixels are necessary to provide sufficient detail for the patch-based ViT encoder, though this constrains GPU memory and limits batch size to 2 images per iteration. The optimization follows Detectron2's ViTDet configuration using AdamW with layer-wise learning rate decay, where deeper transformer layers receive proportionally lower learning rates. Positional embeddings are excluded from weight decay to preserve learned spatial relationships. The learning rate schedule employs multi-step

decay with approximately 250 warm-up iterations. Due to transformers' weaker inductive biases compared to CNNs, aggressive data augmentation is essential, including random horizontal flipping, resize scaling with factors from 0.1 to 2.0, and fixed-size cropping to 1024×1024 pixels. This augmentation strategy provides the scale and translation invariance that CNN architectures inherit naturally through their structure.

All fine-tuning experiments scale training iterations according to dataset size, as detailed in Section 5. Model performance is evaluated periodically on validation sets using COCO-style metrics, as described in Section 4.6. Automatic mixed precision is enabled across all experiments to reduce memory footprint while maintaining numerical stability.

4.6. Evaluation Metrics

In order to evaluate the performance of our wideband signal recognition models, we adopt metrics from the COCO object detection benchmark [38]. Since our task involves both localizing and classifying signals within the time-frequency domain, we need metrics that assess both spatial accuracy and classification performance. We use three complementary metrics: IoU for localization accuracy, AP for combined localization and classification performance, and mAP for overall multi-class performance.

Intersection over Union. To evaluate how well a predicted signal instance $\hat{b}_m = (c, t_c, f_c, d, B)$ matches a ground-truth signal, we use the IoU metric. IoU measures the similarity between two time-frequency bounding boxes by comparing the area of their overlap with the area of their union [78]:

$$\text{IoU}(B_p, B_g) = \frac{|B_p \cap B_g|}{|B_p \cup B_g|} = \frac{\text{Area of intersection}}{\text{Area of union}} \tag{22}$$

A detection is considered correct if the predicted class label matches the ground-truth and the IoU exceeds a predefined threshold, typically from 0.5 to 0.95.

Average Precision. Average Precision (AP) evaluates the model's detection performance by combining both classification accuracy and localization precision at a specific IoU threshold. For a given signal class c and IoU threshold τ, AP is computed as the area under the precision-recall curve:

$$\text{AP}_c(\tau) = \int_0^1 P(R; \tau)\, dR \tag{23}$$

where a detection is considered a true positive only if both the predicted class matches the ground truth class c and the IoU exceeds the threshold τ. The precision P and recall R are defined as:

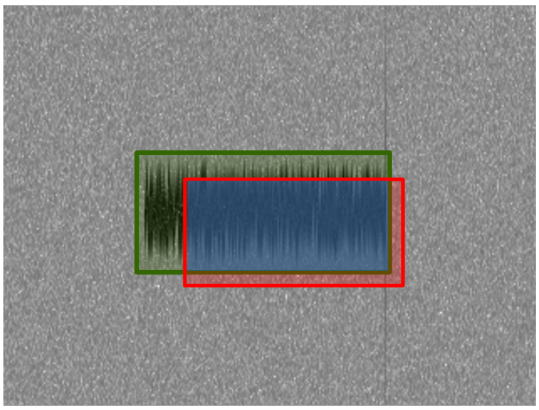


Figure 22: Illustration of IoU calculation for signal detection in time-frequency domain. The IoU metric measures the overlap between predicted (red) and ground-truth bounding boxes (green), with higher values indicating more precise localization. In RF signal detection, this corresponds to accurate temporal and spectral boundary estimation.

$$P(\tau) = \frac{T_p(\tau)}{T_p(\tau) + F_p(\tau)} \tag{24}$$

$$R(\tau) = \frac{T_p(\tau)}{T_p(\tau) + F_n(\tau)} \tag{25}$$

where $T_p(\tau)$, $F_p(\tau)$, and $F_n(\tau)$ count true positives, false positives, and false negatives at IoU $\geq \tau$.

Mean Average Precision@IoU. For multi-class signal recognition at a single IoU threshold, we compute the mean Average Precision (mAP) by averaging the per-class AP at that threshold:

$$\text{mAP@}\tau = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \text{AP}_c(\tau),$$

where

$$\tau \in [0, 1]$$

is the chosen IoU threshold (e.g., $\tau = 0.50$ for mAP@0.5), and $|\mathcal{C}|$ is the number of signal classes.

Mean Average Precision@0.50:0.95. To capture performance over a range of localization strictness, we also report the COCO-style mAP@0.50:0.95, i.e. the mean AP averaged over 10 IoU thresholds from 0.50 to 0.95 in steps of 0.05:

$$\text{mAP@0.50:0.95} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \left[\frac{1}{10} \sum_{t=0.50:0.05:0.95} \text{AP}_c(t) \right] \tag{26}$$

COCO Evaluation Protocol. Following the COCO object detection benchmark [38], we report three primary evaluation metrics that comprehensively assess both detection and localization performance:

- $\mathbf{mAP_{50}}$: Mean AP at IoU threshold 0.5 (mAP@0.5); evaluates coarse detection and classification.
- $\mathbf{mAP_{75}}$: Mean AP at IoU threshold 0.75 (mAP@0.75); enforces stricter localization and precise boundary estimation.
- **mAP**: Mean AP averaged across IoU thresholds from 0.50 to 0.99in increments of 0.05. This is also referred to as mAP@0.50:0.95 and serves as the primary evaluation metric, providing a comprehensive assessment of both detection accuracy and localization precision.

These multi-threshold evaluations are particularly valuable for wideband signal recognition, where signal boundaries may exhibit significant variability due to overlapping transmissions, multipath fading, and spectrogram resolution limitations. The graduated IoU thresholds enable differentiation between models that achieve rough signal detection versus those capable of precise temporal and spectral localization.

5. Experiments & Results

This section presents the experimental setup and results of our conducted experiments. The focus is on evaluating the effectiveness of SSL methods for wideband signal recognition, particularly in terms of data efficiency and transfer learning capabilities across different signal types.

5.1. Experimental Setup

One of the key challenges in RFML is the scarcity of labeled data required for training effective models. To address this limitation, we design different experimental setups to evaluate the performance of SSL methods under varying conditions of labeled data availability and transfer learning scenarios. The experiments are structured around two main axes: data efficiency and transfer learning. Data efficiency experiments assess how well models perform with limited labeled data, while transfer learning experiments evaluate the ability of models pretrained on one signal type to generalize to another signal type. These experiments are crucial for understanding the practical applicability of SSL methods in real-world scenarios where labeled data is often scarce or expensive to obtain.

The experimental pipeline is illustrated in Figure 23. It consists of a single pretraining phase followed by two distinct downstream evaluation experiments. The pretraining phase involves training backbone networks on the unlabeled TorchSig Narrowband dataset, mimicking the pretraining phase from computer vision [5, 65], where models are pretrained on large datasets like ImageNet and then fine-tuned on object detection tasks. For the downstream experiments, the pretrained backbones are fine-tuned on labeled datasets for wideband signal recognition.

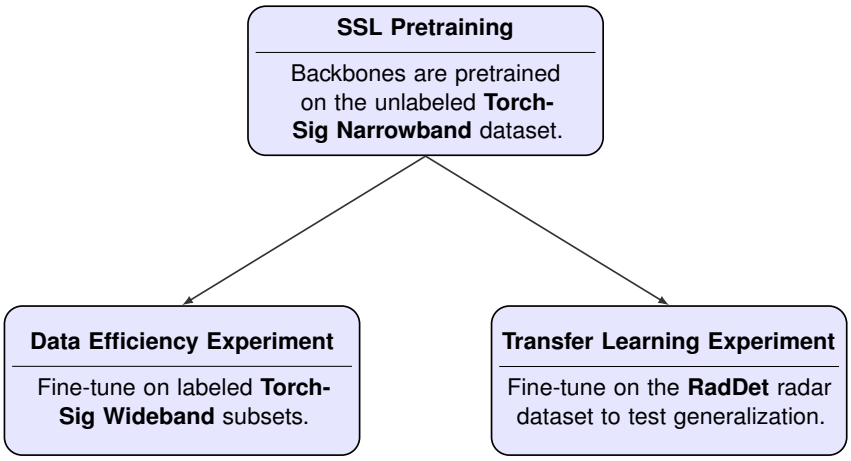


Figure 23: The experimental pipeline, consisting of a single pretraining phase followed by two distinct downstream evaluation experiments: data efficiency and transfer learning.

All experiments evaluate three SSL methods (DenseCL, VICRegL and MAE) against baselines trained from scratch and ImageNet-pretrained backbones. Performance is measured with standard detection metrics (mAP, mAP_{50} and mAP_{75}). To ensure robustness and guard against cherry-picking, each configuration is run three times with different random seeds and we report the average for all metrics. This repeated-runs protocol is particularly important for small labeled subsets, where the larger number of effective training epochs increases variance and overfitting risk, reporting averages ensures results reflect stable trends rather than single-run fluctuations.

5.2. ViTDet & MAE - Exploratory Results and Limitations

This subsection reports exploratory runs of ViTDet with MAE pretraining on the 100k TorchSig Wideband dataset for signal recognition and explains why ViTDet is excluded from the official evaluations due to compute constraints. We evaluated four strategies: training from scratch, MAE pretraining on narrowband data, MAE pretraining on wideband data, and MAE pretraining on ImageNet. All experiments use high-resolution 1024×1024 spectrograms, which constrained the batch size to only two spectrograms per iteration. To keep experiments feasible, we reduced the training schedule to 250,000 iterations; however, to match the original ViTDet training protocol, approximately 5,000,000 iterations (100 epochs) would be required, which would take multiple weeks of continuous training on our available hardware.

The results, shown in Figure 24, demonstrate progressive improvements with different pretraining strategies. The model trained from scratch achieves 35.18 mAP, while MAE pretraining on narrowband data improves performance to 37.01 mAP (+1.83 mAP, 5.20% improvement). MAE pretraining on wideband data shows further improvement to 38.06 mAP (+2.88 mAP, 8.19% improvement). Most notably, MAE pretraining on ImageNet achieves the best performance at 41.14 mAP (+5.96 mAP, 16.94% improvement), demonstrating that large-scale ImageNet pretraining provides substantial benefits even for RF signal recognition tasks.

Despite the computational constraints, the ViTDet models achieve competitive performance with the best CNN-based approaches. The MAE-ImageNet pretrained ViTDet achieves 41.14 mAP, which is comparable to the best CNN results (VICRegL: 43.50 mAP for recognition). This suggests that transformer-based architectures have significant potential for RF signal processing, but their scalability advantages—the ability to effectively utilize larger datasets and computational resources—cannot be fully realized under typical computational constraints.

With access to larger computational resources, higher batch sizes, and more extensive pretraining datasets, ViTDet models would likely outperform CNN-based approaches, as demonstrated in computer vision tasks. Given evidence from computer vision that transformers scale favorably with data and compute, we hypothesize that RF transformers may surpass CNNs under larger data and compute availability, validating this requires future work. However, given these computational constraints

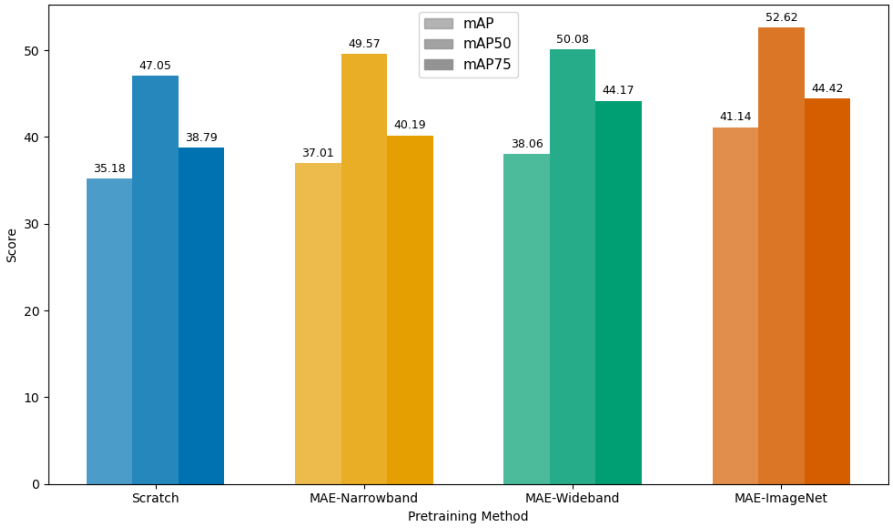


Figure 24: Recognition performance of ViTDet models with different MAE pretraining strategies on TorchSig Wideband. Bars show mAP, mAP_{50}, and mAP_{75} for models trained from scratch, with MAE pretraining on narrowband, wideband, and ImageNet datasets.

and the need for comprehensive experimentation across different data regimes and transfer learning scenarios, transformer-based models prove impractical for our current experimental setup.

Therefore, the subsequent experiments in this thesis focus on CNN-based approaches, which demonstrate strong performance improvements with more reasonable computational requirements. This decision enables more comprehensive experimentation while maintaining practical applicability for typical RF signal processing applications. However, it is important to note that this represents a computational trade-off rather than a fundamental limitation of transformer architectures, and future work with larger computational budgets should revisit transformer-based approaches for RF signal processing.

5.3. Data-Efficiency Experiment

The following experiment evaluates how effectively SSL pretrained models perform when fine-tuned on varying amounts of labeled data from the TorchSig Wideband dataset. By systematically reducing the available labeled training data from 100k down to 1k samples, we assess the data efficiency gains provided by different pretraining strategies compared to training from scratch. The experiment covers both signal detection (localization only) and signal recognition (joint localization and classification) tasks to provide a comprehensive evaluation of SSL benefits across different levels of task complexity. We first outline the experimental design, then

present the results, and conclude with a discussion.

5.3.1. Design

The data efficiency experiment aims to evaluate how well pretrained models perform when fine-tuned on limited labeled data from the TorchSig Wideband dataset. The experiment systematically varies the amount of labeled training data available, allowing us to assess how much pretraining helps in scenarios where labeled data is scarce. We use subsets of the TorchSig Wideband dataset, ranging from 1k to 100k labeled wideband spectrograms, to fine-tune the pretrained models. The experiment compares the performance of backbones pretrained with different SSL methods against a baseline model trained from scratch and against a backbone pretrained on ImageNet.

To ensure a fair comparison, all models are fine-tuned using the same training procedure and hyperparameters, as discussed in Section 4.5.2. Each ResNet-50 training run uses a fixed number of 75k iterations, which follows the default COCO training schedule and equals a total of 12 epochs for the 100k subset at a batch size of 16. This approach keeps the training time constant across all experiments, allowing for a fair comparison of model performance based on the amount of labeled data used.

For final evaluation, the model with the highest mAP on the validation dataset from each run is selected and evaluated on the test dataset of the TorchSig Wideband. We use the official validation split of TorchSig Wideband 100k strictly as a held-out test set for final reporting, without any use for hyperparameter selection or model tuning.

The experiments are performed for both signal detection (single-class object detection) and signal recognition tasks (multi-class object detection). This follows the evaluation protocol from [7], where the authors also evaluate performance on both tasks.

5.3.2. Signal Detection Results

This section presents the results of the data efficiency experiments on signal detection. Signal detection focuses solely on finding and localizing signals within the spectrograms; the classification of signal type is not considered. The experiments were conducted using Faster R-CNN with a ResNet-50 backbone, pretrained on the TorchSig Narrowband dataset with different SSL methods. Each backbone was fine-tuned on the TorchSig Wideband dataset with varying amounts of labeled data and trained for a fixed number of 75k iterations. The results are summarized in Table 7 and visualized in Figure 25. Complete metrics can be found in Appendix A.2.

All pretrained models outperform the model trained from scratch, indicating that pretraining is beneficial for signal detection tasks. The model pretrained on ImageNet also increases the mAP, showing that general-purpose pretraining can be beneficial for wideband signal detection tasks. In low-data regimes, the ImageNet pretrained backbone shows strong improvements, coming close to or even surpassing

the improvements from domain-specific pretraining methods. With more labeled data, the improvements of the ImageNet pretrained backbone decrease. Overall, it improves performance by 3.90% or +2.52 mAP on the 100k subset compared to the model trained from scratch.

The performance of the backbone pretrained with DenseCL improves over the model trained from scratch, only slightly outperforming the ImageNet pretrained backbone. On experiments with 1k, 5k, and 10k labeled samples, it even performs worse than the ImageNet pretrained backbone but shows strong improvement on 25k+ subsets. In total, it improves performance by 5.1% or +3.30 mAP on the 100k subset compared to the model trained from scratch.

The VICRegL pretrained backbone shows the best performance across all subsets, demonstrating consistent improvements from 12.89% on the 1k subset to 10.36% on the 100k subset compared to the model trained from scratch. Overall, it improves performance by +6.70 mAP on the 100k subset with an impressive mAP of 71.39%. This demonstrates that domain-specific pretraining is highly effective and outperforms general-purpose pretraining methods on computer vision datasets like ImageNet.

Dataset Size	**ImageNet**		**VICRegL**		**DenseCL**		**Random**
	mAP	Δ	mAP	Δ	mAP	Δ	mAP
1k	53.48	+5.77	**53.86**	+6.15	50.65	+2.94	47.71
5k	62.32	+7.15	**62.89**	+7.72	58.85	+3.68	55.17
10k	65.67	+7.27	**67.38**	+8.98	62.02	+3.62	58.40
25k	66.33	+3.02	**70.72**	+7.41	67.08	+3.77	63.31
50k	67.04	+3.31	**71.13**	+7.40	67.39	+3.66	63.73
100k	67.21	+2.52	**71.39**	+6.70	67.99	+3.30	64.69

Table 7: Detection results of Faster R-CNN with ResNet-50 and FPN on the TorchSig Wideband 100k validation dataset after fine-tuning for 75,000 iterations. Results are shown for models trained from scratch, pretrained on ImageNet, and pretrained with DenseCL and VICRegL on TorchSig Narrowband.

Figure 25 shows the detection performance as a function of labeled training set size for different pretraining strategies. For all experiments, the mAP increases with the amount of labeled data available, demonstrating that more labeled data leads to better performance. The performance increases significantly up to 25k labeled samples, while the improvements decrease with additional labeled data. The ImageNet pretrained backbone and VICRegL pretrained backbone perform comparably until the 25k subset, where the VICRegL pretrained backbone shows significant improvement over other pretrained backbones. Around 25k labeled samples appears to be the optimal point for fine-tuning, as improvements decrease with additional labeled data. Most importantly, the VICRegL pretrained backbone almost matches the mAP of the randomly initialized model trained on 100k labeled samples using only 5k labeled samples. This demonstrates that the VICRegL pretrained backbone is

highly data-efficient and can achieve strong performance with limited labeled data.

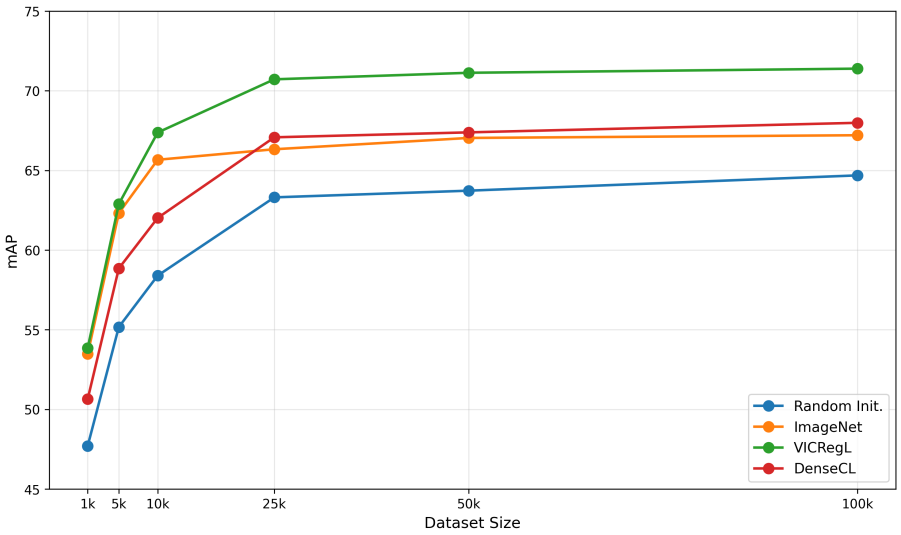


Figure 25: Detection performance (mAP) as a function of labeled training set size for different pretraining strategies on TorchSig Wideband. Results are shown for models trained from scratch, pretrained on ImageNet, and pretrained with VICRegL and DenseCL on TorchSig Narrowband.

5.3.3. Signal Recognition Results

This section presents the results of the data efficiency experiments on signal recognition. Signal recognition focuses on classifying detected signals into different classes, which is a more complex task than signal detection. The experiments were conducted using Faster R-CNN with a ResNet-50 backbone, pretrained on the TorchSig Narrowband dataset with different SSL methods. Each backbone was fine-tuned on the TorchSig Wideband dataset with varying amounts of labeled data and trained for a fixed number of 75k iterations. The results are summarized in Table 8 and visualized in Figure 26. Complete metrics can be found in Appendix A.3.

The overall results are similar to those from signal detection. Because the recognition task involves joint detection and classification of signal type, the reported mAP is lower than the mAP from the signal detection task. However, the relative improvements are almost twice as high, demonstrating that pretraining is highly effective for signal recognition.

The pattern of decreasing improvements with more labeled data is also present for signal recognition, as observed in Figure 26. For the randomly initialized model, performance stagnates around 25k labeled samples. For the pretrained models, this effect appears around 50k labeled samples, showing that pretrained models benefit from additional labeled data for signal recognition.

Again, all pretrained models outperform the model trained from scratch, with much higher improvements compared to signal detection. DenseCL performs the worst but still shows significant improvement of +6.35 mAP or 19.34% on the 100k subset, achieving a total mAP of 39.18. The ImageNet pretrained backbone outperforms the DenseCL pretrained backbone, improving performance by +7.86 mAP or 23.94% on the 100k subset to a total mAP of 40.69 compared to the baseline. The best approach is the VICRegL pretrained backbone, which improves performance by +10.67 mAP or 32.50% on the 100k subset to a total mAP of 43.50.

For low-data experiments, both the VICRegL and ImageNet backbones show nearly identical performance, demonstrating impressive improvements up to 40.99% relative to the baseline model trained from scratch.

Dataset Size	**ImageNet**		**VICRegL**		**DenseCL**		**Random**
	mAP	Δ	mAP	Δ	mAP	Δ	mAP
1k	**24.18**	**+7.03**	23.07	+5.92	21.84	+4.69	17.15
5k	30.03	+5.84	**31.09**	**+6.90**	29.19	+5.00	24.19
10k	**33.25**	**+5.66**	33.14	+5.55	31.29	+3.70	27.59
25k	37.95	+5.55	**39.83**	**+7.43**	35.75	+3.35	32.40
50k	40.46	+7.85	**42.75**	**+10.14**	38.97	+6.36	32.61
100k	40.69	+7.86	**43.50**	**+10.67**	39.18	+6.35	32.83

Table 8: Recognition results of Faster R-CNN with ResNet-50 and FPN on the TorchSig Wideband 100k validation dataset after fine-tuning for 75,000 iterations. Results are shown for models trained from scratch, pretrained on ImageNet, and pretrained with DenseCL and VICRegL on TorchSig Narrowband.

5.3.4. Discussion

The results of the data efficiency experiments clearly demonstrate the effectiveness of SSL methods for wideband signal recognition tasks. Both signal detection and recognition tasks show significant improvements when using pretrained backbones compared to models trained from scratch.

Our analysis reveals that the best pretrained model nearly matched the performance of a randomly initialized model trained on 100k labeled samples using only 5k labeled samples. This finding indicates that SSL methods can dramatically reduce the amount of labeled data required for effective training. Since performance improvements plateaued at approximately 25k labeled samples, we conclude that only around 20% of the labeled data is needed to achieve comparable results to the random baseline.

The performance of the ImageNet pretrained backbone demonstrates that general-purpose pretraining can be beneficial for wideband signal recognition tasks, particularly in low-data regimes. This suggests that features learned from ImageNet

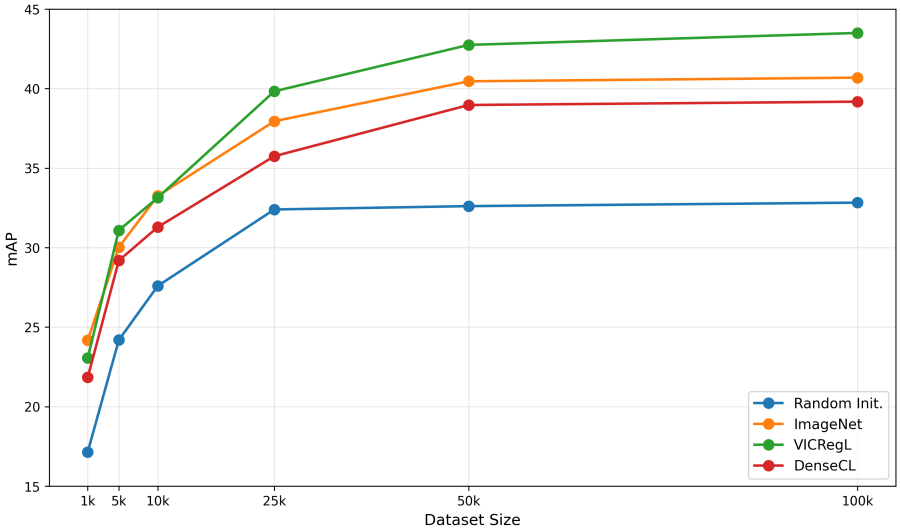


Figure 26: Recognition performance (mAP) as a function of labeled training set size for different pretraining strategies on TorchSig Wideband. Results are shown for models trained from scratch, pretrained on ImageNet, and pretrained with VICRegL and DenseCL on TorchSig Narrowband.

transfer well to the wideband domain, although domain-specific pretraining methods like VICRegL outperform it in most scenarios. This finding is significant because pretrained backbones capture general features such as edges and shapes that are also relevant for wideband signals. Moreover, these pretrained backbones are widely available and do not require additional pretraining time, making them practically valuable.

The VICRegL pretrained backbone consistently outperforms both the ImageNet backbone and DenseCL across both experimental settings. This demonstrates that domain-specific pretraining is highly effective for wideband signal recognition tasks, as it captures features that are more relevant to the specific characteristics of wideband signals. However, this approach comes with the cost of additional pretraining time, as it requires training on the unlabeled TorchSig Narrowband dataset. It is important to note that effective pretraining often requires substantial computational resources, which can be a limiting factor for some applications. Nonetheless, VICRegL demonstrates that this can be accomplished with a single GPU within a reasonable timeframe.

In contrast, DenseCL lags behind the other methods, highlighting how limited computational resources can lead to suboptimal results. The pretraining process was unstable due to reduced batch size and memory bank size, which is a known issue for DenseCL [65]. This observation underscores that the choice of pretraining method is crucial for achieving optimal performance in wideband signal recognition tasks.

Overall, this experiment demonstrates the significant potential of SSL methods to

improve data efficiency in wideband signal recognition tasks. The results indicate that with limited labeled data, pretrained models can achieve strong performance, making them highly suitable for practical applications where labeled data is scarce or expensive to obtain. While general-purpose pretraining methods can provide effective initialization, successful domain-specific pretraining outperforms these approaches as it captures more task-relevant features.

5.4. Transfer-Learning Experiment

This subsection evaluates the data efficiency of SSL pretraining for wideband RFML. We fine-tune a Faster R-CNN with a ResNet-50 backbone on TorchSig Wideband using labeled subsets from 1k to 100k spectrograms, evaluating two tasks—signal detection (localization only) and signal recognition (joint localization and classification). Backbones pretrained with VICRegL and DenseCL on TorchSig Narrowband are benchmarked against ImageNet pretraining and random initialization under a fixed 75k-iteration schedule. By holding the training budget constant and varying label count, we isolate the label-efficiency gains of domain-specific SSL relative to general-purpose and no pretraining. The following sections detail the experimental design, present the results, and discuss their implications.

5.4.1. Design

The transfer learning experiment evaluates the cross-domain generalization capabilities of SSL pretrained models by assessing their performance when transferred from communication signals to radar signals. This experiment addresses a critical question in RFML: whether representations learned from one type of RF signal can effectively transfer to fundamentally different signal types with distinct characteristics and operational contexts.

The experimental design follows a three-stage pipeline. First, backbone networks are pretrained on the unlabeled TorchSig Narrowband dataset using different SSL methods. Second, these pretrained backbones are fine-tuned on the RadDet radar dataset, which contains 11 distinct radar signal classes including LPI polyphase codes and FMCW signals. Third, the fine-tuned models are evaluated on the RadDet test set to assess transfer learning performance.

We compare the performance of backbones pretrained with VICRegL and DenseCL against baseline models trained from scratch and models pretrained on ImageNet. All models use the same Faster R-CNN architecture with ResNet-50 backbone to ensure fair comparison. The RadDet dataset provides a challenging transfer learning scenario due to the fundamental differences between communication and radar signals in terms of waveform characteristics, spectral occupancy patterns, and operational environments.

To ensure robust evaluation, all models are fine-tuned using identical training procedures and hyperparameters. Each training run uses 75k iterations with a batch size of 16, following the same protocol established in the data efficiency experiments.

The RadDet-9T subset is used for training and evaluation, providing up to 9 radar instances per frame in dense radar environments.

The experiment evaluates radar signal recognition. While the primary objective is to assess whether pretrained features can generalize to detect signals with different spectral and temporal characteristics, the multi-class nature of the RadDet dataset inherently requires joint detection and classification. This provides a comprehensive evaluation of cross-domain transfer capabilities across both localization and classification tasks.

Performance is evaluated using standard object detection metrics including mAP, mAP_{50}, and mAP_{75}. The transfer learning capability is assessed by comparing the absolute performance of different pretraining strategies and analyzing the relative improvements over the baseline model trained from scratch.

5.4.2. Results

Table 9 presents the transfer learning performance when transferring pretrained backbones on communication signals from TorchSig Narrowband dataset to radar signals from RadDet dataset. The results demonstrate varying degrees of success across different pretraining strategies, with all pretrained models achieving substantial improvements over the baseline model trained from scratch.

The baseline model trained from random initialization achieves a baseline performance of 27.80 mAP, 36.08 mAP_{50}, and 31.24 mAP_{75} on the RadDet radar signal recognition task. This establishes the lower bound for transfer learning performance and serves as the reference point for evaluating the effectiveness of different pretraining strategies.

ImageNet pretraining demonstrates the strongest transfer learning performance across all metrics, achieving 43.31 mAP, 49.70 mAP_{50}, and 45.98 mAP_{75}. This represents a substantial improvement of +15.51 mAP (55.83%) over the random baseline, indicating that general-purpose visual features learned from natural images transfer remarkably well to radar signal recognition tasks. The ImageNet pretrained model outperforms all other approaches by a significant margin across all evaluation metrics.

VICRegL pretraining achieves moderate transfer learning performance with 37.42 mAP, 46.17 mAP_{50}, and 42.18 mAP_{75}. This corresponds to an improvement of +9.62 mAP (34.64%) over the baseline model. While VICRegL demonstrated superior performance in the data efficiency experiments on communication signals, its effectiveness is reduced when transferring to the fundamentally different radar signal domain.

DenseCL pretraining shows the most limited transfer learning capability among the pretrained methods, achieving 33.69 mAP, 42.49 mAP_{50}, and 31.24 mAP_{75}. This represents an improvement of +5.89 mAP (21.19%) over the baseline, which is substantially lower than both ImageNet and VICRegL pretraining.

Figure 27 visualizes the transfer learning performance across different pretraining

strategies. The figure clearly illustrates the ranking of methods, with ImageNet pretraining showing the largest improvement, followed by VICRegL, DenseCL, and finally the random baseline. The substantial gap between ImageNet pretraining and the SSL methods highlights the challenge of cross-domain transfer in RF signal processing, where domain-specific features may not generalize as effectively as general visual features learned from huge amount of images.

All pretrained models demonstrate positive transfer, with improvements ranging from 21.19% to 55.83% over the baseline. However, the magnitude of improvements varies significantly, with general-purpose ImageNet pretraining substantially outperforming domain-specific SSL pretraining in this cross-domain transfer scenario.

Pretraining	**Performance**			**Improvement**	
	mAP	**mAP$_{50}$**	**mAP$_{75}$**	**Δ mAP**	**Δ %**
Random	27.80	36.08	31.24	–	–
ImageNet	**43.31**	**49.70**	**45.98**	**+15.51**	**55.83%**
DenseCL	33.69	42.49	37.74	+5.89	21.19%
VICRegL	37.42	46.17	42.18	+9.62	34.64%

Table 9: Transfer learning results of Faster R-CNN with ResNet-50 and FPN on the RadDet radar signal recognition task. All models were pretrained on TorchSig Narrowband 50k and fine-tuned on RadDet for 75,000 iterations with batch size 16. Results are shown for models trained from scratch, pretrained on ImageNet, and pretrained with DenseCL and VICRegL.

5.4.3. Discussion

The transfer learning results reveal several important insights about the cross-domain generalization capabilities of different pretraining strategies in the RF domain. The findings highlight both the potential and limitations of domain-specific SSL pretraining when transferring across fundamentally different signal types.

The poor performance of DenseCL in the transfer learning scenario is consistent with the training instabilities observed during pretraining. As discussed in Section 4.5.1 and 5.3, DenseCL suffered from unstable training due to computational constraints that required reduced batch sizes and memory bank sizes. These limitations, which are known issues for DenseCL [65], appear to have resulted in suboptimal feature representations that do not transfer effectively to the radar domain. However, DenseCL does show improvement over the random baseline across all metrics, though this improvement is modest compared to other pretraining strategies.

More surprising is the finding that VICRegL, despite its superior performance in the data efficiency experiments, fails to outperform ImageNet pretraining in the transfer learning scenario. This represents a significant limitation of domain-specific pretraining and suggests that features learned from communication signals may be

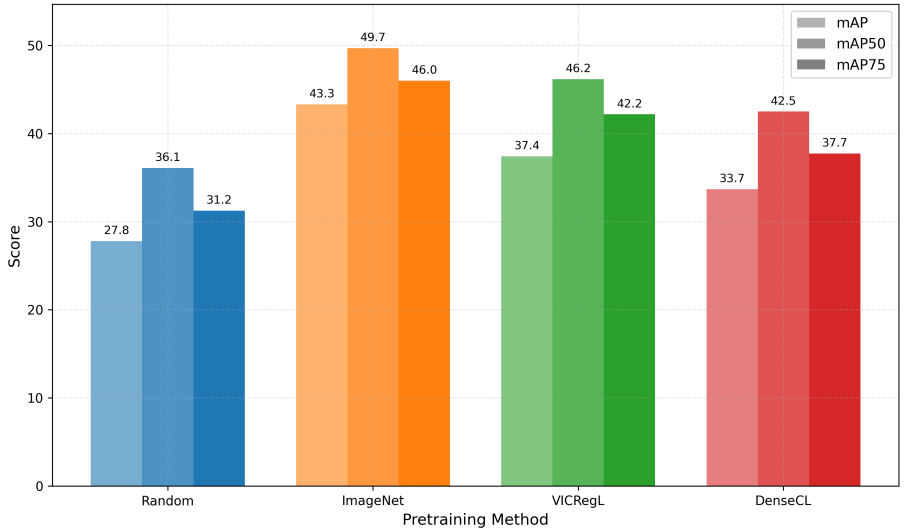


Figure 27: Transfer learning performance of Faster R-CNN with ResNet-50 and FPN on the RadDet radar signal recognition task. Bars show mAP, mAP_{50}, and mAP_{75} for models trained from scratch, pretrained on ImageNet, and pretrained with VICRegL and DenseCL on TorchSig Narrowband.

too specialized to transfer effectively to the fundamentally different characteristics of radar signals.

The remarkable success of ImageNet pretraining in this transfer learning scenario provides important insights into the nature of transferable features in RF signal processing. ImageNet models are trained on massive datasets with million of natural images and substantial computational resources, learning general visual features such as edges, textures, and spatial patterns that appear to be broadly applicable across different domains. The fact that these general-purpose features transfer more effectively than domain-specific RF features from communication signal suggests that low-level visual patterns may be more universally applicable than initially anticipated.

However, this finding must be interpreted within the context of domain similarity. The transfer scenario evaluated here represents an extreme case of cross-domain transfer, moving from communication signals to radar signals, two fundamentally different signal types with distinct operational principles, frequency characteristics, and waveform patterns. The key insight is that domain-specific pretraining effectiveness is highly dependent on domain similarity. While communication-to-radar transfer proves to be challenging, radar-to-radar pretraining would likely demonstrate significant advantages over general-purpose pretraining, as demonstrated by the excellent performance of VICRegL on communication signals when achieving comparable results with only 20% labeled data.

These findings point toward an important trade-off in RFML: domain-specific

pretraining offers exceptional data efficiency and performance within specific signal domains using modest computational resources, while general-purpose pretraining provides better cross-domain transferability at the cost of massive computational resources. Understanding this trade-off is crucial for determining the most effective pretraining strategy for different RFML applications.

5.5. Key Findings

The experimental results demonstrate significant advances in SSL for wideband signal recognition while revealing important limitations in cross-domain transfer scenarios and computational constraints for transformer-based approaches. This section summarizes the key findings from all experiments and discusses critical directions for future research.

The data efficiency experiments demonstrate that domain-specific SSL pretraining is highly effective for wideband signal recognition tasks. VICRegL pretraining enables models to achieve comparable performance using only 20% of the labeled data compared to training from scratch, representing a significant reduction in labeling requirements. The method consistently outperforms both random initialization and ImageNet pretraining across all data regimes, achieving improvements of up to +10.67 mAP (32.50%) for signal recognition tasks. Importantly, this level of performance can be achieved with modest computational resources, pretraining requires only a single GPU and reasonable timeframes, making it accessible to practitioners with limited computational budgets.

However, the transfer learning experiments reveal a critical limitation: poor generalization across fundamentally different RF signal types. When transferring from communication signals to radar signals, domain-specific pretraining (VICRegL: +9.62 mAP, 34.64%) is substantially outperformed by general-purpose ImageNet pretraining (+15.51 mAP, 55.83%). This finding challenges the assumption that domain-specific features should inherently transfer better within the RF domain and suggests that the spectral and temporal differences between communication and radar signals may be too significant for effective transfer using current small scale SSL methods.

The ViTDet experiments with MAE pretraining reveal both the potential and limitations of transformer-based architectures in RF signal processing. Despite significant computational constraints—including severely limited batch sizes (only 2 spectrograms), extended training times (250,000 vs. 75,000 iterations), and high-resolution requirements (1024×1024 pixels), the ViTDet models achieve competitive performance with CNN-based approaches. The MAE-ImageNet pretrained ViTDet achieves 41.14 mAP, which is comparable to the best CNN results. This suggests that transformer architectures have significant potential for RF signal processing, but their scalability advantages, the ability to effectively utilize larger datasets and computational resources, cannot be fully realized under computational constraints.

The experiments reveal multiple important trade-offs in RFML. First, there is a trade-off between computational requirements and transferability: domain-specific

SSL pretraining can be accomplished with modest resources while providing excellent performance within specific domains, whereas general-purpose pretraining requires massive computational resources and datasets but offers superior cross-domain transfer capabilities. Second, there is a trade-off between architectural sophistication and computational practicality: transformer-based models show promise for superior performance with adequate resources, but CNN-based approaches remain more practical for systematic evaluation under computational constraints. These trade-offs have practical implications for resource allocation in RFML projects: for applications within specific signal domains with limited computational budgets, domain-specific CNN pretraining offers the best return on investment, while cross-domain applications or those with substantial computational resources may benefit from general-purpose pretraining or transformer architectures.

6. Conclusion and Future Work

To the best of our knowledge,this thesis has presented the first systematic evaluation of domain-specific self-supervised learning pretraining strategies tailored for wideband signal recognition. Through systematic experimentation across data efficiency and cross-domain transfer scenarios, we have demonstrated both the significant potential and inherent limitations of SSL methods in addressing the fundamental challenges facing radio frequency machine learning.

The experimental evaluation reveals that domain-specific SSL pretraining, particularly VICRegL, delivers substantial improvements in wideband signal recognition and detection. For signal recognition, VICRegL achieves 43.50 mAP with improvements of +10.67 mAP (32.50%) over baseline models trained from scratch. For signal detection, VICRegL reaches 71.39 mAP with gains of +6.70 mAP (10.36%) compared to the baseline. These results demonstrate that SSL methods can reduce labeled data requirements, with pretrained models achieving comparable performance using only 20% of the labeled data compared to training from scratch.

The data efficiency experiments consistently show VICRegL outperforming both random initialization, ImageNet and DenseCL pretraining across all data regimes, with particularly striking improvements in low-data scenarios where labeled samples are most scarce. This represents a significant breakthrough for practical RFML applications, where data collection and annotation remain expensive and time-consuming processes requiring specialized equipment and expert knowledge.

However, the cross-domain transfer learning experiments reveal critical limitations when transferring knowledge across fundamentally different RF signal types. Domain-specific pretraining on TorchSig Narrowband communication signals (VICRegL: +9.62 mAP, 34.64%) is substantially outperformed by general-purpose ImageNet pretraining (+15.51 mAP, 55.83%) when fine-tuned on radar signal recognition. This finding challenges the assumption that RF-specific features should inherently transfer better within the RF domain and reveals the critical importance of domain similarity in determining transfer learning effectiveness. The superior performance of ImageNet pretraining suggests that the learned visual features, such as edge detection, texture recognition, and spatial patterns, may be more universally applicable across different spectrogram types than the specialized temporal and spectral features learned from communication signals. This counterintuitive result indicates that the spectral characteristics, temporal dynamics, and signal structures of communication and radar signals may be fundamentally different enough that domain-specific pretraining on one does not effectively benefit the other, despite both being RF signals represented as spectrograms.

The exploratory ViTDet experiments demonstrate that transformer-based architectures achieve competitive performance (41.14 mAP) with CNN-based approaches despite severe computational constraints. This suggests significant potential for transformer architectures in wideband signal recognition, though their full capabilities could not be realized given the computational resources available for this study.

This work makes several significant contributions to the RFML field by providing the first systematic evaluation of domain-specific SSL pretraining for wideband signal recognition and establishing comprehensive baselines comparing domain-specific pretraining against ImageNet-pretrained models. The implications for RFML, particularly for wideband signal detection and recognition, are substantial. The findings demonstrate that SSL pretraining can significantly reduce the labeled data requirements that have limited RFML development, potentially enabling more widespread adoption of ML-based spectrum sensing systems. The successful adaptation of VICRegL to RF spectrograms establishes a new paradigm for leveraging unlabeled RF data and demonstrates the feasibility of effective SSL pretraining with modest computational resources.

The most promising direction for future research involves developing an RF-specific foundation model that could bridge the gaps identified in this work. Such a model would be pretrained on large, diverse datasets spanning multiple signal types such as communication, radar and IoT, potentially combining the domain relevance of RF-specific pretraining with the broad transferability demonstrated by ImageNet models. While training such a foundation model would require larger datasets and significantly higher compute compared to the modest resources needed for domain- and task-specific pretraining, the resulting model could serve as a universal foundation for diverse RF tasks and signal types.

Additionally, future work should revisit transformer-based approaches with larger computational budgets, as the competitive performance achieved under severe constraints suggests that properly scaled transformer models could outperform CNN-based approaches in RF signal processing. Finally, evaluation on real-world datasets and operational deployments remains crucial for validating the practical applicability of RFML models. Future work should prioritize the development of realistic benchmarks and the transition from synthetic to real-world datasets to ensure advances translate into practical improvements in operational RF systems.

This thesis establishes a foundation for future developments in RFML by systematically demonstrating both the strengths and limitations of current SSL approaches. The insights gained from data efficiency, cross-domain transfer, and architectural comparisons provide a clear roadmap for advancing the field toward more robust and generalizable RF signal processing systems. By highlighting the practical impact of SSL pretraining, the challenges of domain adaptation, and the promise of transformer architectures, this work sets the stage for the next generation of RFML research—focused on scalable, adaptable, and data-efficient models that can meet the demands of real-world spectrum sensing and signal recognition applications.

A. Appendix

The appendix provides supplementary materials referenced in the main text: a complete list of TorchSig signal types used in the experiments and the full detection and recognition result tables from the data-efficiency study.

A.1. TorchSig Signal Types

Table 10 provides a list of all signal types available in the TorchSig [6] library, organized by signal family. These signals were used in the wideband dataset for training and evaluation.

Table 10: TorchSig signal families and constituent signal types used in the wideband dataset.

Family	Signal Type	Family	Signal Type	Family	Signal Type
AM	am-dsb	FSK	4gmsk	OFDM	ofdm-1024
AM	am-dsb-sc	FSK	4msk	OFDM	ofdm-1200
AM	am-lsb	FSK	8fsk	OFDM	ofdm-2048
AM	am-usb	FSK	8gfsk	OOK	ook
ASK	4ask	FSK	8gmsk	PSK	bpsk
ASK	8ask	FSK	8msk	PSK	qpsk
ASK	16ask	FSK	16fsk	PSK	8psk
ASK	32ask	FSK	16gfsk	PSK	16psk
ASK	64ask	FSK	16gmsk	PSK	32psk
Chirp	lfm_data	FSK	16msk	PSK	64psk
Chirp	chirps	OFDM	ofdm-64	QAM	16qam
Chirp	lfm_radar	OFDM	ofdm-72	QAM	32qam
FM	fm	OFDM	ofdm-128	QAM	32qam_cross
FSK	2fsk	OFDM	ofdm-180	QAM	64qam
FSK	2gfsk	OFDM	ofdm-256	QAM	128qam_cross
FSK	2gmsk	OFDM	ofdm-300	QAM	256qam
FSK	2msk	OFDM	ofdm-512	QAM	512qam_cross
FSK	4fsk	OFDM	ofdm-600	QAM	1024qam
FSK	4gfsk	OFDM	ofdm-900	Tone	tone

A.2. Detailed Signal Detection Results for Data Efficiency Experiment

Dataset Size	ImageNet			VICRegL			DenseCL			Random		
	mAP	AP_{50}	AP_{75}	mAP	AP_{50}	AP_{75}	mAP	AP_{50}	AP_{75}	mAP	AP_{50}	AP_{75}
1k	53.48	79.00	57.55	**53.86**	**80.08**	**58.43**	50.65	78.71	54.33	47.71	74.90	50.83
5k	62.32	86.21	68.55	**62.89**	**86.18**	**68.82**	58.85	84.18	64.41	55.17	81.55	59.54
10k	65.67	88.44	71.95	**67.38**	**88.45**	**74.07**	62.02	86.97	69.27	58.40	84.55	63.08
25k	66.33	88.86	73.59	**70.72**	**90.61**	**77.55**	67.08	88.96	73.42	63.31	87.77	68.25
50k	67.04	89.59	73.74	**71.13**	**90.74**	**77.83**	67.39	89.15	74.21	63.73	87.88	69.59
100k	67.21	89.61	73.93	**71.39**	**91.22**	**77.88**	67.99	89.62	74.67	64.69	88.83	70.98

Table 11: Detection performance of Faster R-CNN on TorchSig Wideband 100k. All experiments are pretrained on TorchSig Narrowband 50k and fine-tuned for 75k iterations with batch size 16.

A.3. Detailed Signal Recognition Performance Results

Dataset Size	ImageNet			VICRegL			DenseCL			Random		
	mAP	AP_{50}	AP_{75}	mAP	AP_{50}	AP_{75}	mAP	AP_{50}	AP_{75}	mAP	AP_{50}	AP_{75}
1k	**24.18**	**38.13**	**25.11**	23.07	38.58	25.71	21.84	35.38	23.43	17.15	29.94	16.90
5k	30.03	43.67	32.06	**31.09**	**43.65**	**32.47**	29.19	41.71	31.67	24.19	36.97	26.56
10k	**33.25**	**46.76**	**36.72**	33.14	44.69	36.59	31.29	43.81	34.34	27.59	40.54	29.74
25k	37.95	51.93	41.74	**39.83**	**55.56**	**43.94**	35.75	48.65	39.24	32.40	44.40	35.46
50k	40.46	54.75	44.03	**42.75**	**55.65**	**47.19**	38.97	51.81	43.13	32.61	44.90	35.49
100k	40.69	55.03	44.18	**43.50**	**56.31**	**48.01**	39.18	51.91	43.28	32.83	45.06	35.86

Table 12: Recognition performance of Faster R-CNN on TorchSig Wideband 100k. All experiments are pretrained on TorchSig Narrowband 50k and fine-tuned for 75k iterations with batch size 16.

References

[1] Ahmed Aboulfotouh, Ashkan Eshaghbeigi, and Hatem Abou-Zeid. Building 6g radio foundation models with transformer architectures, 2024.

[2] Saleh Albelwi. Survey on self-supervised learning: Auxiliary pretext tasks and contrastive learning methods in imaging. *Entropy*, 24(4), 2022.

[3] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning, 2023.

[4] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning, 2022.

[5] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicregl: Self-supervised learning of local visual features, 2022.

[6] Luke Boegner, Manbir Gulati, Garrett Vanhoy, Phillip Vallance, Bradley Comar, Silvija Kokalj-Filipovic, Craig Lennon, and Robert D. Miller. Large scale radio frequency signal classification, 2022.

[7] Luke Boegner, Garrett Vanhoy, Phillip Vallance, Manbir Gulati, Dresden Feitzinger, Bradley Comar, and Robert D. Miller. Large scale radio frequency wideband signal detection & recognition, 2022.

[8] Danijela Cabric, Artem Tkachenko, and Robert W. Brodersen. Experimental study of spectrum sensing based on energy detection and network cooperation. In *Proceedings of the First International Workshop on Technology and Policy for Accessing Spectrum*, TAPAS '06, page 12–es, New York, NY, USA, 2006. Association for Computing Machinery.

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

[11] Xinlei Chen*, Saining Xie*, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021.

[12] Kemal Davaslioglu, Serdar Boztas, Mehmet Ertem, Yalin Sagduyu, and Ender Ayanoglu. Self-supervised rf signal representation learning for nextg signal classification with deep learning. *IEEE Wireless Communications Letters*, PP:1–1, 01 2022.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[15] Mark Everingham et al. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 2010.

[16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations, 2018.

[17] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2014.

[18] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour, 2018.

[19] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020.

[20] S. Haykin. Cognitive radio: brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, 23(2):201–220, 2005.

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021.

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[24] Ben Hilburn, Nathan West, Tim O'Shea, and Tamoghna Roy. SigMF: The signal metadata format. In *Proceedings of the GNU Radio Conference*, volume 3, 2018.

[25] Zi Huang, Simon Denman, Akila Pemasiri, Terrence Martin, and Clinton Fookes. Raddet: A wideband dataset for real-time radar spectrum detection, 2025.

[26] International Telecommunication Union (ITU). Managing the radio-frequency spectrum for the world, 2024. Backgrounder on ITU-R and the Radio Regulations.

[27] W. H. Clark IV and A. J. Michaels. Training from zero: Forecasting of radio frequency machine learning data quantity. *Telecom*, 5:632–651, 2024.

[28] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning, 2022.

[29] Yann LeCun. I-jepa: The first ai model based on my vision for more human-like ai, 2023. Accessed: 2025-04-03.

[30] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[31] Yann LeCun and Ishan Misra. Self-supervised learning: The dark matter of intelligence, March 2021. Accessed: 2025-04-03.

[32] Shancang Li, Li Da Xu, and Shanshan Zhao. 5g internet of things: A survey. *IEEE Vehicular Technology Magazine*, 13(4):28–41, 2018.

[33] Weihao Li, Keren Wang, and Ling You. A deep convolutional network for multitype signal detection and classification in spectrogram. *Mathematical Problems in Engineering*, 2020(1):9797302, 2020.

[34] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection, 2022.

[35] Marc Lichtman. *PySDR: A Guide to SDR and DSP using Python*. Self-published, 2020. Accessed: 2025-03-21.

[36] Lightly.ai. Lightly, 2025. Accessed: 2025-06-04.

[37] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017.

[38] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[39] Antonio Mejias and Francisco Ochando Terreros. Simple detection and classification of spectrogram rf signals using a four-layer perceptron. In *2023 IEEE International Conference on Communications (ICC)*, 11 2023.

[40] R. D. Miller, S. Kokalj-Filipovic, G. Vanhoy, and J. Morman. Policy-based synthesis: Data generation and augmentation methods for rf machine learning. In *IEEE MILCOM 2019*, pages 123–128, 2019.

[41] Raluca Nelega, Romulus Valeriu Flaviu Turcu, Bogdan Belean, and Emanuel Puschita. Radio frequency-based drone detection and classification using deep learning algorithms. In *2023 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–6, 2023.

[42] Hai N. Nguyen, Marinos Vomvas, Triet D. Vo-Huu, and Guevara Noubir. Wrist: Wideband, real-time, spectro-temporal rf identification system using deep learning. *IEEE Transactions on Mobile Computing*, 23(2):1550–1567, 2024.

[43] Tim O'Shea, Tamoghna Roy, and T. Charles Clancy. Over-the-air deep learning based radio signal classification. *IEEE Journal of Selected Topics in Signal Processing*, 12(1):168–179, 2018.

[44] Tim O'Shea, Tamohgna Roy, and T. Charles Clancy. Learning robust general radio signal detection using computer vision methods. In *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pages 829–832, 2017.

[45] Timothy O'Shea and Nathan West. Radio machine learning dataset generation with gnu radio. *Proceedings of the GNU Radio Conference*, 1(1), 2016.

[46] Timothy J. O'Shea, Nathan West, Matthew Vondal, and T. Charles Clancy. Semi-supervised radio signal identification, 2017.

[47] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting, 2016.

[48] PyTorch. torchvision.transforms.randomresizedcrop, 2025. Accessed: 2025-06-04.

[49] PyTorch Audio Development Team. torchaudio.transforms.Spectrogram — torchaudio documentation, 2025. Accessed: 2025-06-05.

[50] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

[51] Farheen Ramzan, Muhammad Usman Khan, Asim Rehmat, Sajid Iqbal, Tanzila Saba, Amjad Rehman, and Zahid Mehmood. A deep learning approach for automated diagnosis and multi-class classification of alzheimer's disease stages using resting-state fmri and residual neural networks. *Journal of Medical Systems*, 44, 12 2019.

[52] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016.

[53] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[54] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[55] TeraSense. Radio frequency bands, 2025. Accessed: 2025-04-04.

[56] TorchDSP. Torchsig v1.0.0, Mar 2025. GitHub release.

[57] U.S. Federal Communications Commission. Radio spectrum allocation, 2022. OET page on spectrum allocation and use.

[58] Adela Vagollari, Martin Hirschbeck, and Wolfgang Gerstacker. An end-to-end deep learning framework for wideband signal recognition. *IEEE Access*, 11:52899–52922, 2023.

[59] Adela Vagollari, Viktoria Schram, Wayan Wicke, Martin Hirschbeck, and Wolfgang Gerstacker. Joint detection and classification of rf signals using deep learning. In *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pages 1–7, 2021.

[60] P. Vallance, E. Oh, J. Mullins, M. Gulati, J. Hoffman, and M. Carrick. Torchsig: A gnu radio block and new spectrogram tools for augmenting ml training. In *Proceedings of the GNU Radio Conference*, volume 9, September 2024.

[61] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019.

[62] Johanna Vartiainen, Janne Lehtomäki, Harri Saarnisaari, Markku Juntti, and Kenta Umebayashi. Two-dimensional signal localization algorithm for spectrum sensing. *IEICE Transactions*, 93-B:3129–3136, 11 2010.

[63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[64] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning (ICML 2008)*, pages 1096–1103, 2008.

[65] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training, 2021.

[66] Nathan West, Timothy O'Shea, and Tamoghna Roy. A Wideband Signal Recognition Dataset, 2021.

[67] L. J. Wong, S. McPherson, and A. J. Michaels. An analysis of rf transfer learning behavior using synthetic data. *Preprints*, pages 1–25, 2022.

[68] L. J. Wong and A. J. Michaels. Transfer learning for radio frequency machine learning: A taxonomy and survey. *Sensors*, 22:1416, 2022.

[69] Lauren J. Wong, William H. Clark IV, Bryse Flowers, R. Michael Buehrer, Alan J. Michaels, and William C. Headley. The rfml ecosystem: A look at the unique challenges of applying deep learning to radio frequency applications, 2020.

[70] Dongming Wu, Junpeng Shi, Zhihui Li, Mingyang Du, Fangzheng Liu, and Fangling Zeng. Contrastive semi-supervised learning with pseudo-label for radar signal automatic modulation recognition. *IEEE Sensors Journal*, PP:1–1, 10 2024.

[71] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2, 2019.

[72] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks, 2017.

[73] Xingtong Yun and Xin Zhou. Exploring self-supervised learning for radio signal recognition. In *2021 IEEE 23rd International Conference on High Performance Computing and Communications; 7th International Conference on Data Science and Systems; 19th International Conference on Smart City; 7th International Conference on Dependability in Sensor, Cloud and Big Data Systems and Applications (HPCC/DSS/SmartCity/DependSys)*, pages 2425–2430, 2021.

[74] Chaoning Zhang, Chenshuang Zhang, Junha Song, John Seon Keun Yi, and In So Kweon. A survey on masked autoencoder for visual self-supervised learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)*, pages 6810–6816, 2023.

[75] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization, 2016.

[76] Tianyi Zhao, Benjamin W. Domae, Connor Steigerwald, Luke B. Paradis, Tim Chabuk, and Danijela Cabric. Drone rf signal detection and fingerprinting: Uavsig dataset and deep learning approach. In *MILCOM 2024 - 2024 IEEE Military Communications Conference (MILCOM)*, pages 431–436, 2024.

[77] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection, 2024.

[78] Dingfu Zhou, Jin Fang, Xibin Song, Chenye Guan, Junbo Yin, Yuchao Dai, and Ruigang Yang. Iou loss for 2d/3d object detection. In *2019 International Conference on 3D Vision (3DV)*, pages 85–94, 2019.