Effective Prior Selection and Knowledge Transfer for Deep Learning Applications

by

Sameeksha Katoch

A Dissertation Presented in Partial Fulfillment
of the Requirement for the Degree
Doctor of Philosophy

Approved July 2022 by the
Graduate Supervisory Committee:

Andreas Spanias, Co-Chair
Pavan Turaga, Co-Chair
Jayaraman J. Thiagarajan
Cihan Tepedelenlioglu

ARIZONA STATE UNIVERSITY

August 2022

ABSTRACT

In the recent years, deep learning has gained popularity for its ability to be utilized for several computer vision applications without any apriori knowledge. However, to introduce better inductive bias incorporating prior knowledge along with learned information is critical. To that end, human intervention including choice of algorithm, data and model in deep learning pipelines can be considered a prior. Thus, it is extremely important to select effective priors for a given application.

This dissertation explores different aspects of a deep learning pipeline and provides insights as to why a particular prior is effective for the corresponding application. For analyzing the effect of model priors, three applications which involve sequential modelling problems i.e. Audio Source Separation, Clinical Time-series (Electroencephalogram (EEG)/Electrocardiogram(ECG)) based Differential Diagnosis and Global Horizontal Irradiance Forecasting for Photovoltaic (PV) Applications are chosen. For data priors, the application of image classification is chosen and a new algorithm titled,"Invenio" that can effectively use data semantics for both task and distribution shift scenarios is proposed. Finally, the effectiveness of a data selection prior is shown using the application of object tracking wherein the aim is to maintain the tracking performance while prolonging the battery usage of image sensors by optimizing the data selected for reading from the environment. For every research contribution of this dissertation, several empirical studies are conducted on benchmark datasets. The proposed design choices demonstrate significant performance improvements in comparison to the existing application specific state-of-the-art deep learning strategies.

ACKNOWLEDGEMENTS

Over the past several years, I have grown both professionally and personally, and I would like to thank a number of people for my growth. I owe an outstanding debt of gratitude to my advisors Dr. Andreas Spanias, Dr. Jayaraman Thiagarajan, Dr. Pavan Turaga, Dr. Cihan Tepedelenlioglu and Dr. Suren Jayasuriya, for their constant support and guidance. This Ph.D. would not have been possible without their encouragement and feedback. I am eternally thankful to them for inscribing good research skills in me. I am grateful to them for their valuable time serving on my defense committee and their insightful comments and helpful feedback.

I would like to recognize the invaluable assistance provided to me by the graduate advisors and the School of Electrical, Computer and Energy Engineering at ASU. I would like to thank my friends, peers, and colleagues at the SenSIP center for their kind support, frequent discussions, and memorable days in the lab. I would also like to recognize several research grants that supported this research including NSF CPS award (1646542), NSF I/UCRC award (1540040), NSF IRES award (1854273), NSF MRI award (2019068) and SenSIP Center. I would also like to recognize Lawrence Livermore National Labs (LLNL) and Prime Solutions Group, Inc. (PSG) for providing me with the opportunity to intern and learn at these amazing institutions.

Most importantly, none of this could have happened without my family and friends. Their unconditional love and support have helped me achieve a great deal of success. In particular, I would like to thank my mother Dr. Meenu Katoch and my father Dr. Rajesh Katoch for motivating me to accomplish my goals. I would also like to thank my friends Ankit R., Deepak C., Deepak M., Dheeraj B., Sambhav M., Anup R., Kowshik T., Aseem C., Rahul C., Prateek S., Karan B., Ashra B., Amogh G., Magesh M., Ashish V., Parth S., Vrushti M. and Nikhil D. for their pep-talks, light-hearted humor and good advice throughout the process.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

Table LIST OF TABLES Page

Figure LIST OF FIGURES Page

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

In the previous decade, deep learning has achieved massive success in a wide variety of AI applications with varied data modalities including 1-D time series data (Stoller *et al.*, 2018), image data (Khan *et al.*, 2019) as well as video data (Li *et al.*, 2021). However, instead of directly adopting popular approaches from the computer vision/AI literature for different applications, we interpret learning from the neuroscience literature point of view which suggests, the most effective learning is an outcome of the innate biological machinery which has the capability to learn and improve based on the experience (Marcus, 2004, 2018). The idea of cognition is a combination of innate knowledge (which can be task specific or general) which improves along with experience in any effective learning mechanism. The patterns that are formed in the brain to arrive at a logical conclusion for any problem are due to a fusion of prior information from the innate machinery including innate algorithms, representation formats and knowledge along with the experience of solving related tasks. Learning by utilizing this prior information and arriving at an action which is a combination of variety of tactics that have been used to solve the similar task involves heavy cross-talk between different regions of brain.

We mimic similar learning mechanism through modern transfer learning pipelines wherein we want to fuse the prior knowledge to arrive at a rich initialization for a new task. In other words, if we restrict ourselves to current transfer learning pipelines, the prior can be thought as the distribution that is imputed in the model before it is exposed to any new data. Furthermore, the common assumption in most of the modern AI approaches is the choice of coherent set of tasks for which the existing

transfer learning techniques perform well however, much of this success relies on human selection strategy in terms of task design, data collection mechanism and model design such as architectural choices. Consequently, this constricts the scope of a deep learning model such that it becomes limited to a specific set of problems. Hence, in this work, we explore a more holistic view wherein we consider not only the model, but the implementation of a deep learning pipeline as a whole, including the choice of model and the data. Therefore, the prior includes anything from the choice of the algorithm, model as well as the data. The primary intuition is that the priors are analogous to the innate knowledge whereas the data from any new tasks adds to the improved experience and performance. From a statistical viewpoint, it becomes extremely important to select which **priors (model, tasks and domains)** are the most suitable for a given set of problems.

In this dissertation, we systematically study how each of these priors are effective in improving the performance for problem at hand. Firstly, we explore architectural designs that are able to form robust feature representations and analyze the effect of that prior information (model priors) on sequential modelling problems. Secondly, we also explore the data selection mechanism and data as a prior in this work. We propose an algorithm to utilize the data semantics in a systematic manner such that we are able to create powerful data priors which can improve task performance.

1.1 Motivation

The increased integration of deep learning tools in mainstream applications has triggered the need to microscopically analyze each component of a deep learning pipeline. Human intervention is a primary selection method for both aspects of a deep learning scheme including data selection and model selection. Unless these priors are selected carefully, it can significantly degrade the application performance while also

increasing the computational complexity. In this dissertation, we isolate each prior and empirically analyze the performance of an effective prior for a chosen application. We also provide insights as to why we chose a particular prior for a given application based on the recurring effective design choices made in the literature. In the next section, we state the problem and briefly address different applications considered and what are the design choices for these applications in terms of effective priors.

1.2 Problem Statement

Over the last decade, community-wide research efforts have led to the design of several effective transfer learning solutions, in particular based on deep neural networks (DNNs) (Zamir *et al.*, 2018; Devlin *et al.*, 2019; Yang *et al.*, 2019b; Khan *et al.*, 2019). However, there exists an information gap in terms of whether the deep learning based solution's success can be attributed to effective feature-extraction techniques, modern representation learning paradigms, the ability of model to align itself to novel tasks and operating environments or the combination of the three (Rajan *et al.*, 2019; Teijeiro *et al.*, 2017; Finn *et al.*, 2017a; Li *et al.*, 2017a). To this end, in this dissertation, we systematically study the effect of each component of a typical deep learning pipeline to get a better understanding of how efficient priors can aid the performance in different deep learning applications. We consider different paradigms of knowledge transfer and study the importance of priors in every paradigm.

Knowledge Transfer Paradigms

In recent years, transfer learning has gained huge momentum by improving the performance of several deep learning applications. Transfer learning has become commonplace in many computer vision tasks in which a model uses pretrained representation as an initialization to learn a more useful representation for the specific task in hand. To learn a good initialization, in a typical generic supervised learning

setting, we are given a dataset $\mathcal{D} = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^{N}$ with inputs $\mathrm{x} \in \mathcal{X}$ and labels $\mathrm{y} \in \mathcal{Y}$. The goal is to infer a model $\mathcal{F}$ with parameters ϕ that maps from inputs x to the outputs y. The parameters ϕ are inferred using the following MAP formulation:

$$\arg\max_{\phi} \log p(\phi|\mathcal{D}) = \arg\max_{\phi} \log p(\mathcal{D}|\phi) + \log p(\phi). \tag{1.1}$$

In practice, the first term is approximated using the empirical risk $\sum_i \log p(\mathrm{y}_i|\mathrm{x}_i, \phi)$, while the second term is often an appropriately chosen regularizer. A learning task can be specified by a dataset $\mathcal{D}$ or by the surrogate function $\mathcal{F}(\phi)$ or it can also be specified by other datasets.

Another promising approach for transfer learning from multiple tasks/datasets is exploited in multi-task representation learning (MTRL) paradigms. The intuition of such multi-task learning strategies is to prefer hypothesis which explains more than one task by introducing the inductive bias. In a typical MTRL setting, we are given a set of observed tasks/datasets $\{\mathcal{D}_k\}_{k=1}^{K}$. Each task is assumed to be sampled i.i.d. from a distribution $p(\mathcal{D})$. The assumption is that all the tasks are related and hence a shared feature representation ϕ can be trained jointly on these tasks. In terms of generalization to novel tasks, the idea is that since the hypothesis space performs well for multiple tasks, it will perform well while learning novel tasks. Following the underlying principles of MTRL paradigm, another approach called meta learning has emerged for learning information from multiple tasks and generalizing to unseen tasks proficiently. The primary difference between the two approaches is that while MTRL methods are typically solved by a simple joint optimization, meta-learning algorithms use a bi-level optimization procedure. Similar to MTRL, in a typical meta learning strategy, we have access to multiple observed tasks/datasets $\{\mathcal{D}_k\}_{k=1}^{K}$. These observed

tasks can be used when inferring the parameters ϕ for a new task $\mathcal{D}$,

$$\arg\max_{\phi} \log p(\phi \mid \mathcal{D}, \mathcal{D}_{\text{obs}}) \tag{1.2}$$

$$= \log \int_{\Theta} p(\phi \mid \mathcal{D}, \theta) p(\theta \mid \mathcal{D}_{\text{obs}}) d\theta. \tag{1.3}$$

The primary intuition is that we first infer meta-parameters θ, and subsequently adapt ϕ using both θ and $\mathcal{D}$.

In a multiple task setting, (Baxter, 2000) was one of the pioneer works which presented a bound on the expected loss on the unseen task of any hypothesis space. The method showed that learning multiple related tasks reduces the sampling burden required for good generalization. Following (Baxter, 2000), several works analyzed the generalization bounds in MTRL paradigm. In (Maurer *et al.*, 2016), authors obtained a bound in the order of $O(1/\sqrt{}(N) + 1/\sqrt{}(K))$, where K represents number of observed tasks and N represents number of samples per observed task. However, this bound is limited to the observed task samples only and not on the number of samples per novel unseen task implying that increasing the number of samples per training task cannot improve generalization on new tasks. On the contrary, there are several research works including (Du *et al.*, 2020; Tripuraneni *et al.*, 2021), that have developed bounds on generalization in MTRL which are dependent on the number of novel task samples. In these works, authors have obtained transfer learning guarantees in the order of $O(poly(1/MK))$ where M is the number of novel task samples. However, these guarantees are specified under certain assumptions including well behaved data generating distributions and notion of task diversity. Moreover, these bounds are restrictive in the sense that they have been primarily limited to linear task maps. While many of these existing works assume the task-similarity beforehand to implement versions of meta learning algorithms, a generalization error bound that fully utilize all training data by exploiting the proposed task relatedness is reported

in (Guan and Lu, 2021).

Based on these learning paradigms, in this work we explore knowledge transfer from the perspective of both data selection as well as model selection. It is evident that any learning task can be specified by the dataset or by the model parameters. Acknowledging the information sources in these paradigms, we propose more powerful data and model based priors that can produce better feature representations and utilize data semantics effectively.

1.2.1 Model Prior for Audio Source Separation, Clinical Time-series based Differential Diagnosis and Global Horizontal Irradiance Forecasting for Photovoltaic Applications

Audio source separation refers to the problem of extracting constituent sound sources from a given audio mixture. Clinical time-series (EEG/ECG) based differential diagnosis involves interpreting time-varying recordings of electrical activity from brain and heart to understand and predict the onset of disease conditions like seizure activity and myocardial infarction. Global horizontal irradiance forecasting refers to the prediction of irradiance values in future based on the meteorological parameters to predict the power output from a photovoltaic array. For such sequence modelling applications, a recurring design objective in the proposed solutions is to build feature representations that are robust to inherent data variabilities and can effectively represent complex, multi-scale dependencies in the data. We propose the idea of a robust model prior which can automatically infer multi-scale features (Lea *et al.*, 2017; Bai *et al.*, 2018a) and serve as an effective design choice from the standpoint of feature reuse and combating the vanishing gradient problems for sequential data.

1.2.2 Task and Domain Prior for Data Efficient Learning

Utilizing transfer learning for data efficient image classification has gained significant interest in last decade among the vision community. There are several approaches which achieve unparalleled performance gains by exploiting pre-training strategies (Devlin *et al.*, 2018; Yang *et al.*, 2019a). However, such task specific performance control can be shattered with the variability in testing scenarios wherein there are unforeseen tasks and domain shifts. Hence instead of assuming similar task semantics at the training and testing time, we propose an optimization protocol titled `Invenio` which creates a semantic space of tasks and domains and extracts meaningful prior information (task/domain prior) which is able to quantify how difficult it is to transfer information from one scenario to another.

1.2.3 Data Selection as a Prior for Energy-Efficient Object Tracking

The analog readout circuitry of image sensors which capture the image/video data can consume 50-70% of the total energy in most modern mobile system designs (LiKamWa *et al.*, 2016; Buckler *et al.*, 2017). Hence, to prolong battery life and address inefficient energy expenditure recent research is looking for methods to improve the image sensors such that the required power consumption can be reduced. To this end, we propose to use desired data selection from the image sensors as a way to optimize the power consumption. Hence, we study data selection mechanism as a prior in an online setting. We propose an adaptive subsampling strategy which selectively performs data capture based on the objects of interest. Effective data selection has proved several benefits in terms of reduced data quantization, faster bandwidth and improved energy efficiency.

1.3 Statement of Contributions

1.3.1 Development of Robust Model Priors

Audio Source Separation Audio source separation refers to the problem of extracting constituent sound sources from a given audio mixture. Modern audio source separation techniques rely on optimizing sequence model architectures such as, 1D-CNNs, on mixture recordings to generalize well to unseen mixtures. We use Wave-U-Net as our base architecture which exploits temporal context by extracting multi-scale features (Narayanaswamy *et al.*, 2019b; Stoller *et al.*, 2018). However, the optimality of the feature extraction process in these architectures has not been well investigated. In this work, we examine and recommend critical architectural changes that forge an optimal multi-scale feature extraction process. To this end, we replace regular 1D convolutions with adaptive dilated convolutions that have innate capability of capturing increased context by using large temporal receptive fields. We also investigate the impact of dense connections on the extraction process that encourage feature reuse and better gradient flow. The dense connections between the downsampling and upsampling paths of a U-Net architecture capture multi-resolution information leading to improved temporal modelling.

Clinical Time-series based Differential Diagnosis Interpreting time-varying recordings of electrical activity from the heart and brain, is central to understanding a gamut of abnormalities or detecting the onset of disease conditions. Processing short- and long-term EEG recordings is imperative for predicting the neurological state of a subject, such as detecting seizure events or early onset of abnormalities. Similarly, ECG interpretation is essential for detecting a variety of cardiac abnormalities, namely: atrial fibrillation (AF), myocardial infarction (MI) and arrhythmia. In this dissertation,

we demonstrate that an effective model prior can improve the performance for a wide-range of diagnostic tasks (Thiagarajan *et al.*, 2020). We find that adaptively dilated causal convolutions are an effective choice for the foundational computational unit to process clinical time-series data, and when coupled with dense architectures, it can outperform state-of-the-art diagnostic solutions.

Global Horizontal Irradiance Forecasting in Photovoltaic (PV) Applications In order to ensure the stability of a photovoltaic array output, we propose a weather feature based solar irradiance forecasting strategy which can aid in regulating and planning the operation of a grid integrated solar array (Duverger *et al.*, 2017; Rao *et al.*, 2020). This idea was inspired by shading based topology reconfiguration works which have shown to improve the PV power output performance previously (Katoch *et al.*, 2018b; Narayanaswamy *et al.*, 2019a). In this application, we utilize a model prior, which uses dilated causal 1-D convolutional layers to extract robust temporal features (Bai *et al.*, 2018a). The dilations lead to increased receptive field which aids to see global horizontal irradiance data much farther in the past and hence improve the forecasting performance.

1.3.2 Development of Efficient Data Priors

Image Classification For the problem of image classification, several sophisticated transfer learning approaches are often found to be brittle when applied in scenarios characterized by challenging domain and task shifts. Several transfer learning methods operate under the assumption that the observed set of tasks or domains are realizations from a common distribution. However, in practice, the degree of similarity between tasks or domains are unknown a priori, and hence the assumption of finding a single base learner could be restrictive. To this end, in this work, we present `Invenio`, a

scalable model agnostic protocol, that can effectively infer the semantic structure of the space of tasks (or domains) (Katoch *et al.*, 2019b). By extracting useful prior information from the inferred semantic space, most relevant train tasks (domains) can be used to drive the adaptation for a test task (domain). We show that `Invenio` can produce significant performance improvements in both task shift and domain shift scenarios, when compared to popular modeling choices that do not leverage the inherent semantic structure.

1.3.3 Development of Data Selection Mechanism as a Prior

Energy-Efficient Object Tracking In the field of video object tracking, recent research has turned to embedded or energy-efficient object tracking where system constraints on power and latency are critical for extended deployment in the wild. To address inefficient energy expenditure in the processing of real-time video data, we propose an important data selection mechanism for image sensors i.e. adaptive subsampling (Katoch *et al.*, 2019a). Adaptive subsampling is the selective readout of regions of interest (ROIs) in sequential frame capture while turning off other pixels in the image. Energy per pixel and spatiotemporal resolution of the streaming images are inversely proportional, i.e. lower frame rates and image resolutions consume less energy (LiKamWa *et al.*, 2013). Using this adaptive subsampling mechanism, the proposed method outperforms the state-of-the-art tracking methods in terms of tracking efficiency and reduces power consumption during image sensing.

1.4 Organization of the Dissertation

In this dissertation, we present the importance of prior selection in performance of deep learning based methods. We show that due to varied applicability of deep learning based solutions, they will be put in environments wherein the human intervention

for model design and data handling is not trivial. Hence such constraints should be considered during algorithm design so that there is inherent knowledge that when a new problem occurs, which priors can be effectively used. In Chapter 2, we emphasize on the model prior in terms of making significant architectural choices and how those choices improve the performance across several applications such as, Audio Source Separation, Clinical Time-Series (EEG/ECG) based Differential Diagnosis and Global Horizontal Irradiance Forecasting. In Chapter 3 and 4, the priority is given to understanding the task and domain priors. In Chapter 3, we introduce `Invenio`, a data conscious optimization technique, which is able to infer semantic similarities between a given set of tasks/domains. In Chapter 4, we explore how `Invenio`, can be modified based on the problem at hand i.e. it can leverage the inferred semantic structure to generalize to unobserved domains, construct semantic space of task distributions as well as utilize task similarity knowledge for effective multi-task learning. In Chapter 5, we discuss about data selection mechanism in terms of adaptive subsampling for video object tracking. We compare the tracking performance in the presence of fully sampled data, random subsampling as well as adaptive subsampling and show how the proposed method is able to compete with the state-of-the-art. Finally in Chapter 6, we summarize our work and conclude the dissertation.

Chapter 2

MODEL PRIORS

In this chapter, we study the unique challenges which are presented in dealing with time series data which has warranted the design of several novel neural network architectures. Existing solutions for time-series data are specialized to the task under consideration or the modality utilized, and rarely are they actually tested across different problems. Consequently, it is cumbersome to choose from the vast array of existing modelling choices and find an appropriate solution for new problems. A recurring design objective in these solutions is to build feature representations that are robust to inherent data variabilities and can effectively represent complex, multi-scale dependencies in the data. Taking this design objective into account, in this chapter we demonstrate effective architectural choices that work on a wide-range of sequential data processing tasks. We show that in the context of sequential data processing, the use of dilated convolutions is key to obtain robust multi-scale features and hence a rich temporal context. Along with dilated convolutions, utilizing the dense connections in the architecture helps in modelling long range dependencies and hence provides significant modelling power. We study the impact of these model priors in three areas i.e. Audio Source Separation, Clinical Time-series (EEG/ECG) based Diagnostics and Global Horizontal Irradiance Forecasting for PV Applications as shown in the following sections.

2.1 Audio Source Separation

Audio source separation refers to the problem of extracting constituent sound sources from a given audio mixture. Despite being a critical component of several

audio enhancement and retrieval systems (Spanias *et al.*, 2006), the task of source separation is severely challenged in practice due to variabilities in acoustic conditions. Mathematically, this is posed as an inverse problem, and classical regularized optimization techniques such as independent component analysis (ICA) (Makino *et al.*, 2004) and matrix factorization are often employed (Thiagarajan *et al.*, 2013). However, such unsupervised approaches are known to be effective only under specific conditions (e.g. fully determined) and hence several state-of-the-art solutions (Grais *et al.*, 2018), (Pascual *et al.*, 2017), (Stoller *et al.*, 2018), (Jansson *et al.*, 2017) increasingly rely on supervisory deep learning techniques, that directly learn the inverse mapping using mixture-source pairs. This was motivated by the success of deep learning in solving several highly ill-conditioned inverse tasks in computer vision, such as image completion and super-resolution (Ulyanov *et al.*, 2018). A recurring idea in the broad class of recent source separation techniques is to adopt an encoder-decoder style architecture, powered by convolutional or generative adversarial networks, for end-to-end optimization of the inversion process. While these data-driven solutions have produced unprecedented success in audio source separation, their performance depends heavily on the choice of data processing strategies and network architectures.

Until recently, majority of source separation techniques operated in the spectral domain, in particular based on the magnitude spectra. However, by ignoring the crucial phase information, these methods required extensive tuning of the front end spectral transformation for producing accurate separation results. Recently, in (Stoller *et al.*, 2018), Stoller *et. al.* argued that the need for optimizing spectral transformations can be entirely eliminated by directly operating in the time domain, and that the source recovery quality can be significantly improved by not rejecting the phase information. On the other hand, such a fully time-domain approach necessitates the need to deal with very long temporal contexts at high sampling rates, thus making the network

training quite challenging. Stoller *et. al.* addressed this critical limitation by proposing the *Wave-U-Net* that leverages multi-scale features obtained using a combination of 1D-convolutions and resampling strategies in a U-Net, which is a fully convolutional network widely adopted in semantic segmentation (Ronneberger *et al.*, 2015). In general, U-Nets are comprised of a *downstream* and an *upstream* module, wherein the former module produces multi-scale features by successively downsampling the audio signals while the latter utilizes resampling in order to produce appropriate context information for subsequent layers. In order to obtain meaningful gradients at different temporal scales, the network allows information propagation between the *downstream* and *upstream* layers using skip connections. Though Wave-U-Net outperformed several existing baselines, the optimality of the multi-scale feature extraction process has not been studied yet. Further, conventional upsampling was found to produce undesirable aliasing artifacts, thus requiring the design of an adaptive interpolation scheme.

2.1.1 Related Work

In this section, we briefly review existing approaches in the literature that utilize deep neural networks for audio source separation. There exists a large body of prior work for source separation using time-frequency representations typically, short-time Fourier transforms (STFTs) (Uhlich *et al.*, 2017; Liutkus *et al.*, 2017; Luo *et al.*, 2017). While (Uhlich *et al.*, 2017) and (Liutkus *et al.*, 2017) operated with spatial covariance matrices for source separation in the STFT domain, Luo et al. (Luo *et al.*, 2017) used the magnitude spectrogram as the representation for a mixture and its constituent sources. Due to inherent challenges in phase spectrum modification, much of existing literature has focused on the magnitude spectrum, while including an additional step for incorporating the mixture phase information, which often leads to inaccurate determination of source signals (Stoller *et al.*, 2018). Furthermore, with low-latency

systems, large window lengths are needed for effective separation in the STFT domain.

A common approach to address these drawbacks is to entirely dispense the spectral transformation step and build the estimation algorithm in the time-domain directly. Popular instantiations of this idea include the MultiResolution Convolutional Auto-Encoder (MRCAE) (Grais *et al.*, 2018), TasNet (Luo and Mesgarani, 2018) and the Wave-U-Net (Stoller *et al.*, 2018). MRCAE (Grais *et al.*, 2018) is an autoencoder-style architecture comprised of multiple convolution and transpose convolution layers, wherein each layer supports filters of different sizes. Note that, this is analogous to capturing audio frequencies with multiscale resolutions. A crucial limitation of this approach is its inability to deal with long temporal sequences - results reported were with 1024-length sequences, which is often insufficient to model the complex dependencies at high sampling rates. On the other hand, TasNet (Luo and Mesgarani, 2018), which is also an encoder-decoder style framework, represents an audio mixture as a weighted sum of basis signals, wherein the estimated weights indicate the contribution of each source signal and the filters from the decoder form the basis set. However, given that the architecture is designed for low-latency scenarios, similar to MRCAE, it deals with only short sequences.

In order to support the use of long temporal sequences, Stoller *et.al.* (Stoller *et al.*, 2018) proposed the Wave-U-Net model, which uses a U-Net based architecture and can deal with even 80,000- sample long sequences. While the contracting *downstream* part captures features at different scales, the expanding *upstream* part successively produces high-resolution features. Furthermore, skip connections are used between *downstream* and *upstream* layers, in order to obtain meaningful gradients at different temporal scales. However, as we show in this chapter, the design of the multi-layer feature extraction process plays a critical role in the performance of this architecture. Furthermore, the training of such multi-scale feature learning networks, particularly

with very deep *downstream* and *upstream* modules, can be significantly challenging. We propose to incorporate dense connections, that are known to implicitly encourage feature-reuse (Huang *et al.*, 2017b), to alleviate this challenge.

2.1.2 Proposed Approach

The task of audio source separation involves separating a given mixture waveform $M \in \mathbb{R}^{L_m \times C}$ into K constituent source waveforms $[S_1, \ldots S_K]$, where each $S_i \in \mathbb{R}^{L_n \times C}$. Here, L_m and L_n denote the lengths of the mixture and the sources respectively, and C represents the number of channels. In our formulation, we consider $L_m = L_n$, $C = 2$ implying stereo and the mixing process is a unweighted sum of sources.

Background: The Wave-U-Net Model

The proposed approach is based on the Wave-U-Net architecture in (Stoller *et al.*, 2018), which utilizes an encoder-decoder style architecture. This model follows a standard U-Net design and is comprised of 12 convolutional layers, in both the *downstream* and *upstream* parts. Each convolutional layer is followed by a factor 2 decimation to obtain successively higher resolution information along the *downstream* path. Similarly, in the *upstream* path, bilinear interpolation coupled with an 1D convolution layer is used to perform upsampling. In addition, skip connections are included between every convolutional layer in the *downstream* and *upstream* paths.

The number of filters in the first downstream layer is fixed at $f = 15$ and is increased in the subsequent layers as $f + f \times (i-1)i$ where i represents the layer index. The kernel size for the filters was chosen to be 15 in all layers. The *upstream* path also has similar filter configurations except that the kernel size was chosen to be 5. Finally, the model contains a bottleneck block consisting of a convolution layer with $f + f \times (i)$ filters and kernel size 15. Note that all convolutional layers included

the LeakyReLU activation function. The final source prediction layer uses the tanh activation. The loss function for training the model includes the Mean Squared Error (MSE) for each of the sources. Furthermore, an energy conservation constraint is imposed by directly estimating only $K-1$ sources and obtaining the Kth source as the difference between the input mixture and the sum of estimates for $K-1$ sources.

Dilated U-Net

As discussed earlier, the performance of source separation approaches that operate directly in the time-domain rely heavily on the quality of the feature extraction process. In particular, building a generalizable model requires the ability to model a wide-range of temporal dependencies, which in turn requires effective multi-scale feature extraction. Furthermore, it was found in (Stoller *et al.*, 2018) that the choice of resampling scheme was very sensitive. Hence, we propose to employ dilated convolutions to seamlessly incorporate multi-scale features, thereby dispensing the need for explicit resampling. The proposed Dilated U-Net architecture is illustrated in Fig. 2.1.

This model consists of 6 convolutional blocks in the downstream path, where every block contains 3 dilated convolutions with filter configurations similar to that of Wave-U-Net. Within each block, the dilation rate of the layers increases exponentially by a factor of 2. We have chosen the dilation rate of the first layer in the consecutive block to be the same as the dilation rate of the last layer in preceding block. This strategy results in providing a wide range of dilation rates from [1...4096] which increases the effective receptive field, thereby producing improved multi-scale features from the audio excerpt. Note that, all layers perform convolution with a stride of 1 and employ same padding. The bottleneck block consists of three 1D convolution layers with dilation rate 1, stride 1 and same padding. Correspondingly, the *upstream* path also consists of 6 blocks of transposed dilated convolutions, wherein the configurations were

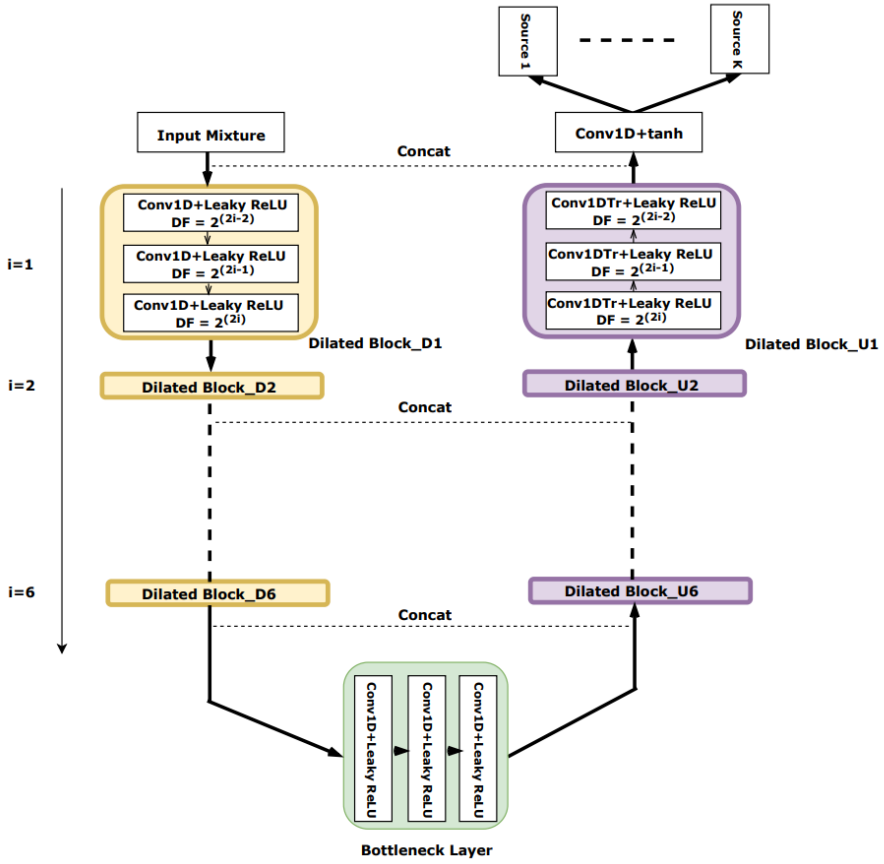


Figure 2.1: Dilated U-Net - Each Convolutional Block Consists Of Three 1D Convolutions Wit Exponentially Increasing Dilation Factors. Note That, The `Upstream` Part Utilizes Dilated Transposed Convolutions to Recover the Sources.

chosen to reflect the downstream path. The use of skip connections, and the process of source estimation follow (Stoller *et al.*, 2018). By retaining the training protocol and loss functions, we hope to quantify the impact of the proposed architectural changes.

Dilated Dense U-Net

While the Dilated U-Net enables seamless incorporation of multi-scale features, with increasing depths in downstream and upstream paths, the network training becomes very challenging. We propose to improve this by employing dense connections in the networks, that supports feature reuse and protects against vanishing gradients. The

Dilated Dense U-Net architecture proposed in this work is illustrated in Fig. 2.2. The architecture is very similar to the previous case, with the key difference that each block (a.k.a dense block), contains dense connections between the dilated convolutional layers. More specifically, within every dense block, the feature maps produced by each layer are concatenated to the subsequent layers in the block to exploit the advantages of feature reuse and improved gradient flow. This can however lead to a large number of feature maps which may be computationally infeasible to process. In order to control the growth of the number of feature maps, we include a transition block which performs dimensionality reduction at the end of every dense block.

The bottleneck block consists of three 1D convolution layers that are densely connected with the dilation rates and stride equal to 1 with same padding. Correspondingly, the upstream path consists of 6 dense blocks where each block contains 3 transposed convolution layers with dilation rates same as the corresponding block in the downstream path. Furthermore, in this model, the skip connections between the respective blocks along the paths are made dense, implying that the feature maps from the block in the downstream path are concatenated to all following layers in the corresponding dense block at the upstream path. Finally, the process of source extraction is identical to the Dilated U-Net.

2.1.3 Experiments

In this section, we evaluate the proposed approaches using the publicly available MUSDB18 dataset and present comparisons to the state-of-the-art Wave-U-Net model (Stoller *et al.*, 2018). Before presenting the performance evaluation, we will first discuss the impact of different design choices on the overall performance. This study provides important insights into the behavior of source separation approaches that operate directly in the time-domain.

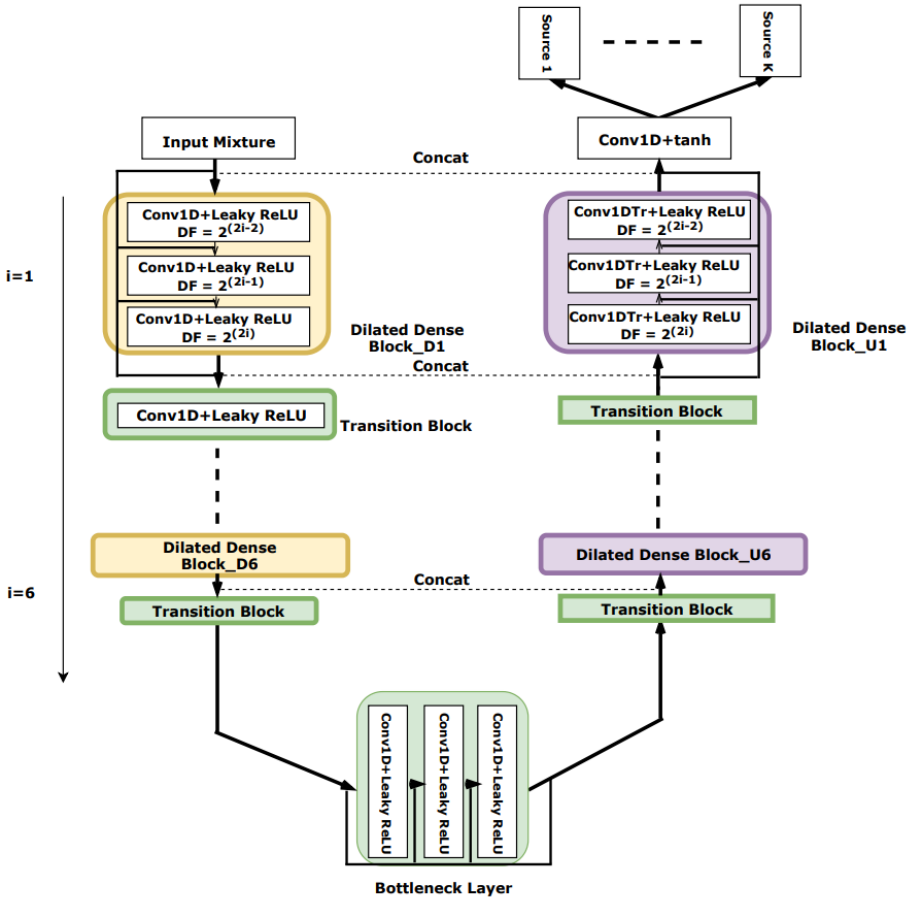


Figure 2.2: Dilated Dense U-Net - Similar to Fig. 2.1, Every Convolutional Block Is Comprised of Three 1 D Convolutions With Exponentially Increasing Dilation Factors. In Addition, We Allow Dense Connections Between Convolutions Within Each Block as Well as Across the *Downstream* And `Upstream` Paths.

Setup We use the MUSDB18 dataset (Rafii *et al.*, 2017) for our experiments, which is comprised of 75 tracks for train, 25 for validation and 50 for testing. The dataset is encoded in the stems format, and contains multi-stream files of separate sources i.e. bass, drums, other and vocals and resampled to 22050 Hz. In our experiment setup, we use segments of $16,384$ samples each (1sec) and adopt a simple additive mixing process, following current practice. Note that, in (Stoller *et al.*, 2018), the authors found that using much larger input contexts ($L_m > L_s$) produces improved results. However, to measure the effective performance of the architectural choices

alone, we benchmark without the additional input context. We also performed data augmentation similar to (Stoller *et al.*, 2018), wherein the source signals are scaled using a randomly chosen factor in the interval [0.7, 1]. All models reported in this work were trained using the Adam optimizer with a learning rate of 0.0001 and a batch size of 16. While the results for the initial study were obtained by training for only 30 epochs, the actual performance metrics were obtained by training for longer (80 epochs). The mean and median signal-to-distortion ratio (SDR) for each of the sources over the entire dataset are computed. The SDR metric takes into account the noise arising from interference and other artifacts in the estimated audio sources (Vincent *et al.*, 2006). The mean SDR is computed after removing silence regions. Since the mean value can be affected by outliers from near-silence regions we also report the median SDR which is known to be more unbiased.

Impact of Design Choices As discussed earlier, the source separation performance depends heavily on the architecture choices for multi-scale feature extraction. Hence, we first study the impact of different dilation schemes in the proposed architecture, wherein we entirely eliminate the resampling process using dilated convolutions. As described in the previous section, our architecture is comprised of 6 blocks of convolutional layers. In its simplest form, we use conventional 1D convolutions with the dilation rate fixed at 1 in all layers. In addition, we consider the case where it was fixed at a constant value (512) and the case with the proposed adaptive dilation scheme. As observed in Fig. 2.3(a), the sub-optimal performance of conventional 1D convolution clearly shows the importance of leveraging multi-scale features. Furthermore, the proposed adaptive dilation scheme provides a significant performance boost compared to using fixed dilation in all layers. Similarly, we analyzed the impact of using dense connections on the separation performance. For this experiment, we fixed the dilation

rate at a constant value of 512 and the number of convolutional blocks at 1 and 3 respectively. As showed in Fig. 2.3(b), as the depth of the network increases, using dense connections provides significant gains.

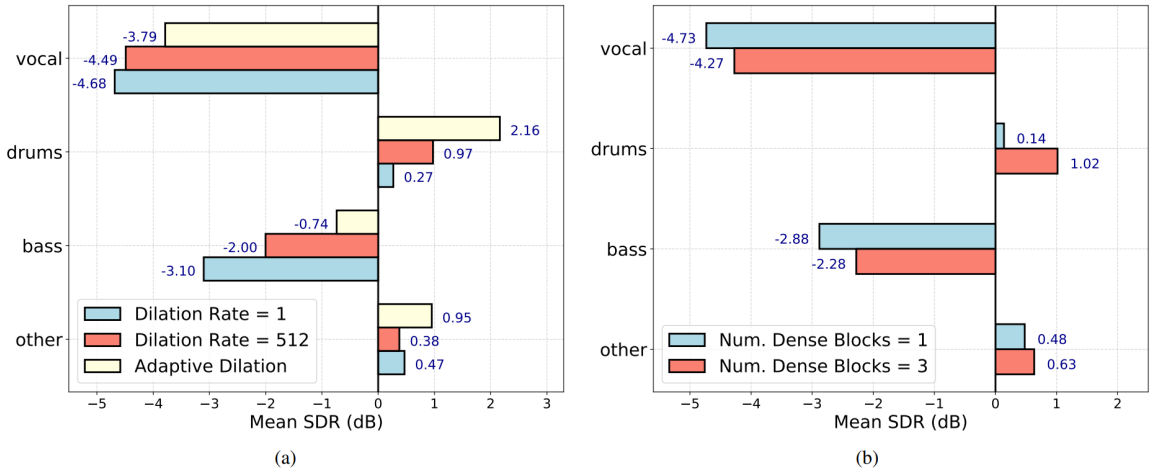


Figure 2.3: Effect of Design Choices on the Source Separation Performance (Mean SDR (dB)) - (a) Impact of the Choice of Dilation Rates in Different Layers of the Model. An Adaptive Learning Provides a Significant Performance Boost. (b) Impact of the Use of Dense Connections as the Depth of the Architecture Increases.

2.1.4 Performance Evaluation

In this section, we report the overall performance of the proposed approaches, namely Dilated U-Net and Dilated Dense U-Net, on the MUSDB18 dataset. Though a number of baseline techniques exist for time-domain source separation, we chose to compare against the state-of-the-art Wave-U-Net architecture from (Stoller *et al.*, 2018). Table 2.1 compares the mean/median SDR (dB) for each of the constituent sources for the testing set in MUSDB18. The first striking observation is that by improving the multi-scale feature extraction process, we could obtain significant performance improvements over the baseline in all cases. In particular, our approaches provide improvements between 0.2dB and 1.2dB. While the dilated variant eliminates the need for explicit resampling by capturing information from exponentially increasing

Source	Wave-U-Net		Dilated U-Net		Dilated Dense U-Net	
	Mean SDR	**Median SDR**	**Mean SDR**	**Median SDR**	**Mean SDR**	**Median SDR**
Vocal	-3.292	2.643	-3.787	2.561	**-2.986**	**2.83**
Drums	1.435	3.310	2.163	**3.977**	**2.449**	**3.934**
Bass	-1.935	1.942	**-0.738**	**3.08**	-1.023	2.711
Other	0.986	1.911	0.953	1.945	**1.187**	**2.039**

Table 2.1: Source Separation Performance Obtained Using Different Architectures on the MUSDB18 Corpus. We Show the Mean and Median Signal-To-Distortion Ratio (In dB) And in Each Case the Best Results Are Highlighted in Bold.

receptive fields, the inclusion of dense connections improves the robustness of the training process. This performance gain clearly evidences the dependence of these complex data-driven solutions for audio processing on the inherent feature extraction mechanism, and the need for improved architecture design.

In the next section, we look at the application of Clinical Time-series based Differential Diagnosis. With the similar design choice of using dilated dense convolutions, we could improve the performance for a given diagnostic task.

2.2 Clinical Time-series based Differential diagnosis

Precise differential diagnosis plays a crucial role in enabling robust decision-making and realizing effective patient care. In particular, interpreting time-varying recordings of electrical activity from the heart and brain is required frequently for clinical diagnosis and is central to understanding a gamut of abnormalities or detecting the onset of disease conditions. More importantly, a wide range of non-invasive diagnostic modalities, e.g. ECG (electrocardiogram) and EEG (electroencephalogram), have become highly prevalent because of their cost efficiency, thus leading to a deluge in diagnostic data being generated. Consequently, in the recent years, there has

been a rapid growth in automation approaches for performing effective coarse/fine characterization of these measurements. Since it is almost impossible to recreate the complex decision-making process of clinical experts, when it comes to discriminating between multiple seemingly-similar conditions and handling inherent variations in the data, these automation methods rely almost entirely on data-driven pattern discovery (Faust *et al.*, 2018). There are several AI-powered solutions that have produced unprecedented successes for a variety of challenging tasks in heart/brain health monitoring. However, in practice, choosing an appropriate solution for a new problem from the vast array of existing modelling choices is a cumbersome process. Consequently, in this work we explore the architectural choices that can be generic and hence, can operate across different scenarios effectively. The different scenarios considered in this work include EEG-based abnormality detection, ECG-based arrhythmia classification and ECG-based myocardial infarction detection. For all the applications considered in this work, we present `DDxNet`, a novel architecture composed of dilated convolutions, which can automatically infer multi-scale features (Lea *et al.*, 2017; Bai *et al.*, 2018a) without the need for explicit feature pooling operations, and hence can be used for processing any sort of clinical time-series data. In this work, we also employ *adaptively dilated causal convolutions* which prove to be an effective choice for the foundational computational unit to process clinical time-series data, and when coupled with dense architectures can provide significant modelling power. Our empirical studies clearly evidence that our approach, with no additional architecture tuning, outperforms state-of-the-art solutions specifically designed for different benchmark tasks. In addition to producing highly effective predictive models, `DDxNet` enables rapid prototyping of solutions in practice.

2.2.1 Related Work

Over the last decade, community-wide research efforts have led to the design of several predictive modelling solutions, in particular based on deep neural networks (DNNs), that can perform an accurate characterization, while being resilient to the inherent data challenges including sampling discrepancies, low-fidelity of measurements, class imbalances etc. A formal introduction to this large body of work can be found in broader survey articles such as (Faust *et al.*, 2018; Miotto *et al.*, 2017; Esteva *et al.*, 2019). These AI-powered solutions have produced unprecedented successes for a variety of challenging tasks in heart/brain health monitoring (Clifford *et al.*, 2017; Rajpurkar *et al.*, 2017; Schirrmeister *et al.*, 2017; de Diego, 2017; Rajan and Thiagarajan, 2018; Rajan *et al.*, 2019; Kachuee *et al.*, 2018; Roy *et al.*, 2018; Sourkov, 2018). The research in this space has also been accelerated by the curation of large-scale, open-source databases such as the TUH-corpus (Temple University) (Obeid and Picone, 2016), PhysioNet (Goldberger *et al.*, e 13) and Mimic (Harutyunyan *et al.*, 2019).

Though it might appear natural to directly adopt popular approaches from the computer vision/AI literature, e.g. recurrent models and convolutional neural networks (CNNs), the unique challenges in dealing with clinical data has warranted the design of novel network architectures and improved training strategies. Consequently, state-of-the-art solutions often rely on a carefully chosen combination of classical feature-extraction techniques as well as modern representation learning paradigms (Rajan *et al.*, 2019; Kachuee *et al.*, 2018; Teijeiro *et al.*, 2017; Hong *et al.*, 2017; Acharya *et al.*, 2017; Strodthoff and Strodthoff, 2018). A recurring design objective in these solutions is to build feature representations that are robust to inherent data variabilities and can effectively represent complex, multi-scale dependencies in the measurements (Roy *et al.*, 2018; Sourkov, 2018). Driven by the need for portable and rapid patient care,

researchers have also explored the inclusion of additional constraints such as using only subset of the measurements, e.g. a single-lead ECG in lieu of 12 channels, to perform diagnosis, and found data-driven methods to be surprisingly effective (Rajpurkar *et al.*, 2017). However, the primary drawback is the difficulty to choose an appropriate design choice that works across different problems. In this work, we build an architecture that requires both the design of computing units that are suitable for different modalities and task characteristics. Furthermore, we also perform rigorous empirical validation that the proposed architecture can perform competitively against specialized approaches.

2.2.2 Proposed Approach

We present a multi-specialty diagnosis model, `DDxNet`, that we demonstrate to be an effective architecture for a wide-range of diagnosis tasks, while enjoying the computational benefits of state-of-the-practice solutions. `DDxNet` is a fully convolutional architecture, similar to several existing state-of-the-art (Miotto *et al.*, 2017; Rajan *et al.*, 2019; Rajpurkar *et al.*, 2017) solutions in clinical diagnosis, which are known to be superior to conventional methods relying on hand-engineered features. While resnet-style solutions have produced unprecedented success with challenging problems in clinical diagnosis, e.g. atrial fibrillation detection from ECG and seizure onset prediction using EEG measurements, `DDxNet` utilizes dense connections between stacked convolutional layers (Huang *et al.*, 2017b), along with multi-scale feature extraction, for improved modelling. While a systematic control of how the network expands provides meaningful abstractions of patterns in time-varying data, including dense connections ensures maximal information flow between layers in the network. In contrast to resnet-style architectures (Rajpurkar *et al.*, 2017), which combine features through summation before they are forwarded to the next layer, we combine features by concatenating them. Through this connectivity pattern, the multi-scale abstraction

can be entirely accessed by the later layers, thus leading to much improved feature representations. Further, as argued in (Huang *et al.*, 2017b), using dense connections will ensure that the features are reused, thus avoiding redundancy in the learned feature maps.

Architecture We now describe the architectural choices required to design `DDxNet`, illustrated in Fig. 2.4. `DDxNet` is built by stacking blocks of convolutional layers designed to capture multi-scale patterns in temporal data through dilated causal convolutions, and to compound meaningful abstractions from those patterns.

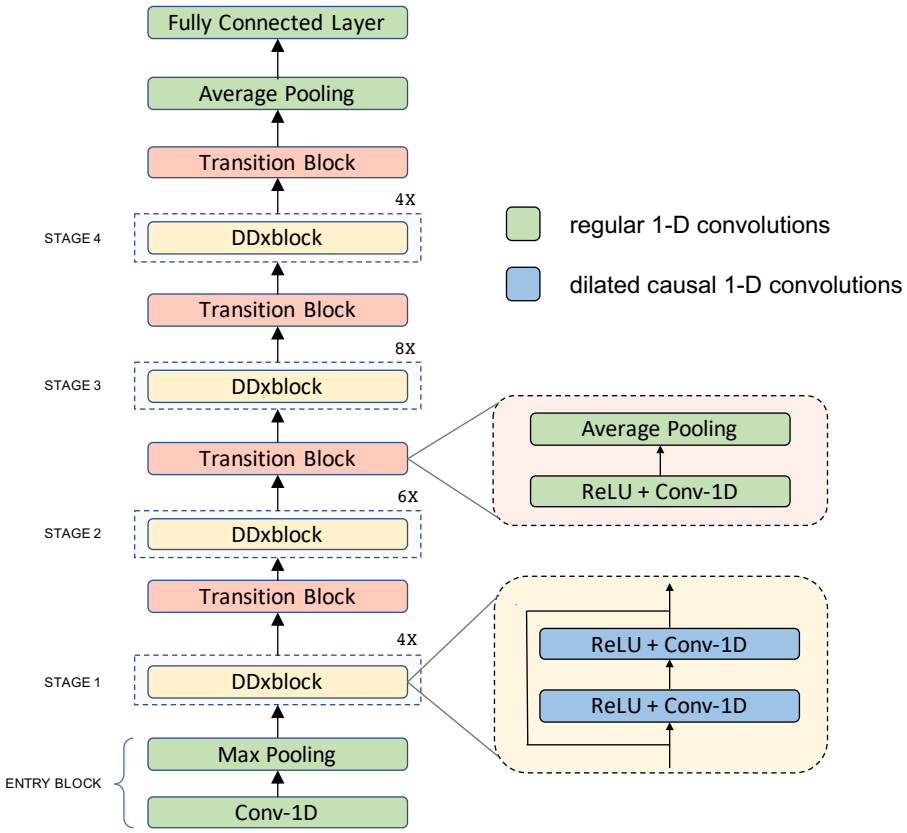


Figure 2.4: An Illustration of the Proposed `DDxNet` Architecture. `DDxNet` Builds a Densely Connected Networks With Dilated Causal Convolutions, Wherein the Dilation Factor Is Adaptively Adjusted for Extracting Multi-Scale Features. Each `DDxblock` Is Comprised of a Bottleneck Layer and a Convolutional Layer Designed According to the *Growth Rate* Hyper-Parameter. Each Processing Stage Is Followed by a Transition Block Which Performs Temporal Aggregation Prior to Invoking the Next Stage.

A Multi-stage Fully Convolutional Model The architecture is comprised of a sequence of repeating convolutional blocks – the outputs of a block are concatenated with the input to that block, before being forwarded to the next convolutional block. `DDxNet` is based entirely on $1-D$ convolutions that computes filter activations only in the temporal dimension. Referred to as a `DDxblock`, each block contains a bottleneck convolution layer with kernel size $k = 3$, followed by another convolution layer. Each `DDxblock` block produces the same number of output features, and the *growth rate* parameter controls the rate at which the network expands. Our network contains a total of 4 processing stages, wherein each of the stages contains 2, 6, 8 and 4 `DDxblock`s respectively. Prior to invoking the multiple stages of densely connected convolution layers, the input is processed using a convolutional layer with $k = 7$ and a max pooling layer.

Note that, after each stage, we include a *transition block*, that performs temporal aggregation through average pooling with stride $s = 2$, and then carries out channel-wise dimensionality reduction using a bottleneck convolution layer with $k = 1$. The resulting features from the transition block in the final stage are processed through an average pooling layer to produce a single feature vector for the entire sequence. The final classification layer is implemented using a single fully connected layer with *softmax* activation. In contrast to CNN architectures used in vision applications, we do not perform batch normalization in any stage of the network. We found in our empirical studies that using batch normalization resulted in a poorer convergence behavior during training.

Adaptive Dilation Dilated convolutions have become an integral part of several successful sequence-data processing approaches – examples include *Wavenet* (Van Den Oord *et al.*, 2016), segmentation networks (Lea *et al.*, 2017) and temporal convo-

lutional networks (Bai *et al.*, 2018a). Basically, dilated convolutions are convolutions with expanded receptive fields. As a result, stacking dilated convolutions with increasing dilation factors amounts to obtaining a multi-scale abstraction of the input signal. In a standard 1-D convolution with $k = 3$, the feature activations are obtained based on the adjacent time-steps. When the dilation factor is set at d, a dilated convolution with the same kernel size applies the filter to every $(d-1)^{th}$ instead of the directly adjacent time-steps. This effectively captures a larger context without the need to explicitly perform down-sampling of the signal. Every `DDxblock` is implemented using dilated convolutions, wherein the dilation factor is increased exponentially within each stage as follows:

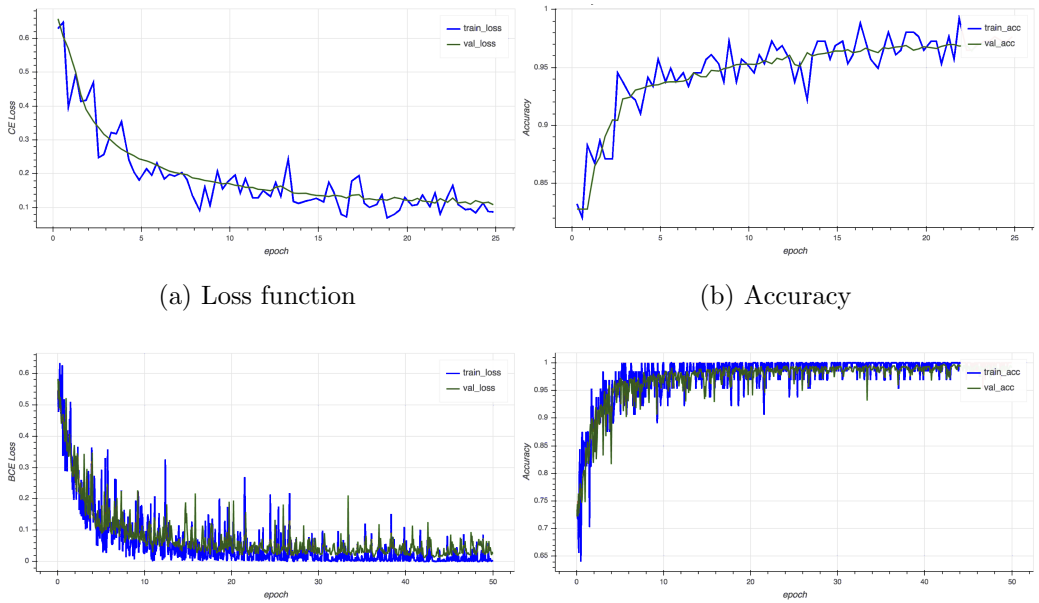
$$d_i = \min(128, 2^{i+2}), \tag{2.1}$$

where i denotes the index of the `DDxblock` in each stage. This enables a principled way to extract multi-scale features from the time-varying clinical data and effectively back-propagate gradients through the multiple stages. Note that, we set the maximum dilation factor 128, since for the most commonly adopted sampling rates in clinical recordings, larger contexts produce features that do not generalize well. This observation corroborates well with the results in a recent work (Song *et al.*, 2018a), where the authors designed an attention-only architecture for clinical modelling, and found that larger contexts while computing the attention weights led to poorer generalization.

Causal Convolutions Another important feature of `DDxNet` is that it employs causal convolutions. With causal convolutions, the activations that a layer produces at time-step t of the signal depends only on data obtained before t. While this is commonly used in speech synthesis models such as the Wavenet (Van Den Oord *et al.*, 2016), their importance in diagnostic models has not been well-studied yet. In particular, when there are repetitive patterns in the measurements, enforcing causality

while computing the receptive field will not lead to improved representations. However, when the data under consideration is highly non-stationary, enforcing causality can be highly beneficial. Hence, `DDxNet` employs causal convolutions in all stages, while retaining normal convolutions (without causality or dilation) in the entry and transition blocks.

Training While the proposed architecture is expected to demonstrate desirable convergence behavior even when the networks are very deep, we observed that additional regularization can be highly beneficial, particularly when training data sizes are small. `DDxNet` uses a weight regularization of 0.01 for all network parameters, applies gradient clipping and performs label smoothing (Pereyra *et al.*, 2017) on top of the loss function. Interestingly, we found that using dropout in the network resulted in lower performance, and hence none of the layers in `DDxNet` uses dropout. For binary classification problems, we used the binary cross entropy loss, and the categorical cross entropy for multi-class problems. `DDxNet` is implemented in PyTorch and the networks were trained using the Adam optimizer. Furthermore, we found that performing warm restarts during model training, following the idea in (Loshchilov and Hutter, 2016), was beneficial in all our experiments. On the other hand, choosing different schedules for learning rate decay did not have a significant effect. All results reported were obtained by starting at the learning rate of $1e-5$, batch size of 64 and training for 100 epochs. Fig. 2.5 shows the training behavior of `DDxNet` on two different ECG interpretation problems. As it can be observed, without any additional architectural modification, `DDxNet` achieves a stable convergence, even when the network is quite deep for smaller datasets.



(a) Loss function (b) Accuracy

(c) Loss function (d) Accuracy

Figure 2.5: Training Behavior – Convergence Characteristics of the Proposed `DDxNet` Model, in Terms of the Cross Entropy Loss and Accuracy Scores, for the Arrhythmia Classification (a-b) And Myocardial Infarction Detection (c-d) Datasets.

2.2.3 Experiments

In pursuit of designing a generic predictive modelling architecture for different clinical diagnosis tasks, we consider a set of benchmark problems, which vary in the data modality utilized, the required degree of characterization, and the assumptions on data fidelity. These benchmarks broadly represent challenges commonly encountered in cardiovascular/neurological diagnosis, and are typically solved using highly specialized deep learning solutions. With `DDxNet`, our goal is to provide an all-encompassing approach that can seamlessly transition across different scenarios, and to quantitatively evaluate its effectiveness against strong problem-specific baselines from the literature. Fig. 2.6 illustrates examples from each of the benchmark datasets.

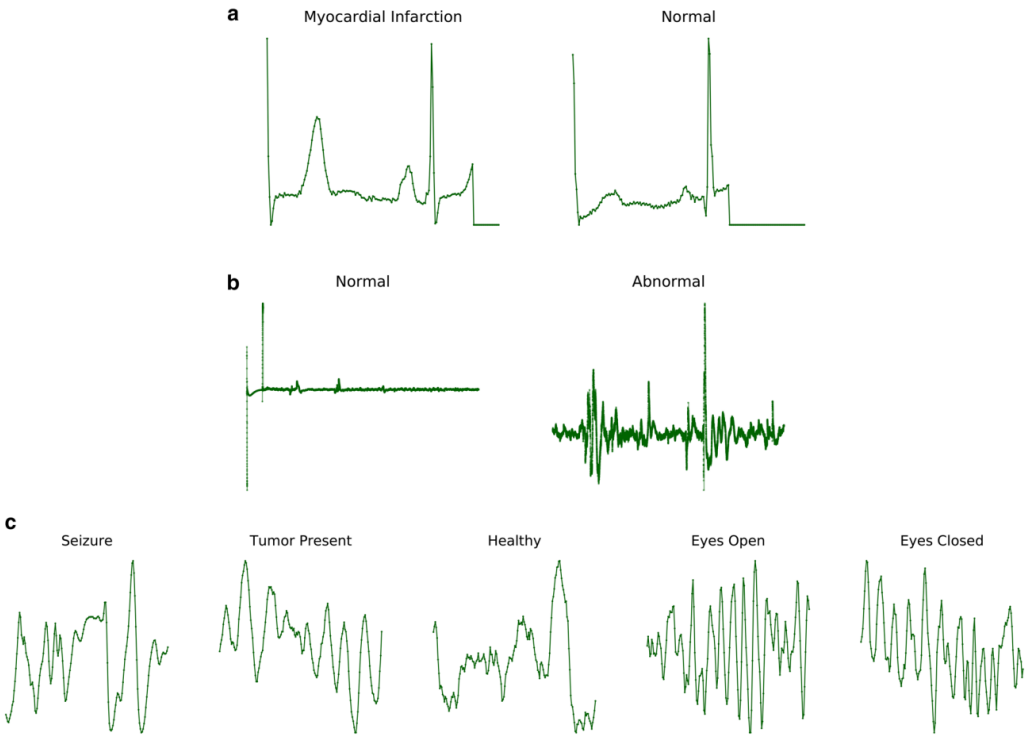


Figure 2.6: Example Data From the Different Benchmark Diagnosis Problems Considered in This Work: (a) Myocardial Infarction Detection; (b) Arrhythmia Classification; (c) EEG Abnormality Detection. Note That, for the Case of EEG Abnormality Detection, We Show Only One of the 22 Channels.

EEG-based abnormality detection The complex dynamics of a brain system can be viewed through EEG signals recorded by non-invasively placing electrodes on the scalp. Clinical studies focus on determining correlations between these signals and observable action during abnormal responses. Typically, an EEG technician reviews numerous montages looking for discrepancies in various frequency bands, starting with the alpha (7.5 - 12 Hz) component responsible for relaxation. The multi-dimensional nature of EEG signals carry critical frequency information, but similar content could mean different outcomes depending on which locations in the brain they are being picked up from. Consequently, feature representations that preserve both frequency and spatial characteristics have proven to be successful in prediction tasks. Further, lengthy recordings tend to be highly noisy due to active patient

movements such as eye blinking. Finally, the inherent variabilities in measurements based on patient demographics such as age and gender play a significant role in the overall interpretation of EEG patterns. For example, an infant showing flat lines and other turbulent regions in the recordings is normal, while interpreting the same patterns if it came from an older adult could mean abnormal. A typical process of EEG interpretation involves analyzing potential epileptic activity, spikes, sharp waves, background slowing, asymmetrical behavior and natural occurrences of awake, sleep and drowsy states. This process is known to be cumbersome, given the volume and long durations of recordings, as subjects are monitored for 48 to 72 hours on average. Consequently, a first step in automating EEG screening is to accurately detect abnormalities.

Data and Pre-processing: Building effective abnormality detection approaches has remained a challenge due to the lack of well-annotated datasets, owing to the labor-intensive annotation process. Recently, tremendous effort has been made in curating an open-source dataset, the TUH abnormal corpus, to enable machine learning research in EEG interpretation (Obeid and Picone, 2016). This dataset is comprised of EEG sessions from 2993 patients, out of which 1521 are normal and 1472 are abnormal. Each EEG record includes 22 channels, sampled at 250 Hz in an average reference (AR) configuration spanning at least 15 minutes long. It is widely accepted in the community that earlier segments of an EEG session are less likely to be corrupted by changes in recording conditions, when compared to the later stages (López *et al.*, 2017). Consequently, in our setup, each sample in our data is constructed as the first 5 minute snapshot of a subject's EEG. Subsequently, we convert the snapshot into a transverse central parietal (TCP) montage configuration, which is a common protocol for abnormality detection. The pre-processing step involves computing Mel Frequency

Cepstral Coefficients (MFCC) (McFee *et al.*, 2015) for each channel, a popular feature extraction technique used in time-varying signal processing – we first resample the data from 250 Hz to 100 Hz, and use a 2000-point FFT with 40 filters and an overlap of 100 timesteps, thus resulting in a sample of 880 channels and 301 timesteps.

ECG-based arrhythmia classification A crucial step towards monitoring cardiovascular health is to detect the presence or absence of abnormal rhythms in ECG. This typically involves delineation of different wave segments (P-wave, QRS-complex, T-wave) and their relationships from a standard 12-channel ECG recording. Several cardiac diseases including atrial fibrillation, ventricular flutter, tachycardias and left/right bundle branch blocks (BBB) manifest as anomalous deviations in ECG channel configurations, where each lead provides a unique perspective on the electrical activity of the heart. For example, leads II, III, and aVF are used to detect inferior myocardial infarction, while leads V1 and V6 are used for bundle branch block (Rajan and Thiagarajan, 2018). Despite that, in telemetry and other ambulatory settings only a subset of channels are accessible, making the task of abnormality detection more challenging (Rajan *et al.*, 2019). Consequently, the problem of arrhythmia detection is often formulated as classifying heartbeats, using only single-channel ECG, into one of the 5 categories prescribed in the association for advancement of medical instrumentation (AAMI) EC57 standard. The list of abnormalities are {N, S, V, F, Q} that can be broadly mapped to {BBB, atrial pre-mature, ventricular premature, fusion beats, paced beats} respectively.

Data and Pre-processing: The MITBIH database (Moody and Mark, 2001) contains 47 subjects, both inpatients and outpatients from Boston's Beth Israel Hospital. Each recording is a two-channel ECG, 48 hours long and sampled at 360 Hz. However, for the task of arrhythmia classification, the ECG data is initially pre-processed by

creating a representation for every heartbeat using the following protocol (Kachuee *et al.*, 2018): the lead II signal is resampled to 125 Hz and split into normalized windows of 10 seconds. Subsequently, for each snapshot, we detect the R-peaks and R-R intervals, essentially generating a heartbeat signal, which is finally classified into one of the five categories described earlier.

ECG-based myocardial infarction detection Detecting myocardial infarction (MI) is one of the most crucial problems within the computerized ECG interpretation community. There have been several solutions proposed to solve this task using both 12 ECG channels as in (Kojuri *et al.*, 2015; Sharma *et al.*, 2015) and (Strodthoff and Strodthoff, 2018), as well as a limited subset of channels as shown in (Acharya *et al.*, 2017; Rajan and Thiagarajan, 2018). The standard 12-lead ECG depicts evidence of ischemic heart diseases that predominantly occur due to the narrowing of blood vessels caused by atherosclerosis. Abnormalities in ECG segments such as the T-wave and Q-wave in addition to ST-elevation are typical signs of myocardium damage leading to a myocardial infarction (MI), commonly referred as a heart attack. Infarction could occur in different regions of the heart, and are picked up by the corresponding ECG leads (Rajan *et al.*, 2019). In this problem, we consider the challenging task of detecting all variants of MI (including lateral, septal and posterior MI) using only a single channel of ECG (lead II).

Data and Pre-processing: The Physionet PTB database (Bousseljot *et al.*, 1995) includes 148 subjects with MI and 52 subjects with normal heart rhythms. Each ECG record is 30 seconds long, sampled at 1 KHz. Similar to the case of arrhythmia classification, we use a single channel ECG (lead II) resampled to 125 Hz.

Method	Acc (%)	Rec (%)	Prec (%)
(López *et al.*, 2017)	78.8	75.4	77.8
(Andreotti *et al.*, 2017)	82.2	73.8	85.32
(Bai *et al.*, 2018a)	82.2	77.8	82.35
(Roy *et al.*, 2018)	**86.6**	83.46	**86.7**
`DDxNet`	**86.6**	**87.3**	84.0

Table 2.2: `EEG Abnormality Detection` - Performance of `DDxNet` On the Publicly Available TUH Data Corpus. For Comparison, We Report the Results Obtained Using Several State-Of-The-Art Baselines. The Best Numbers Are Shown in Bold.

2.2.4 Performance Evaluation

Through rigorous empirical analysis with the benchmark problems, we find that, with the same underlying network architecture, `DDxNet` produces high detection rates, often outperforming even problem-specific state-of-the-art solutions, thus motivating its adoption as a generic approach for clinical diagnosis from time-varying measurements. Note that, all results reported in this section were obtained using the standard train-test data splits prescribed in each of the benchmark datasets.

Typically, in EEG-based abnormality detection systems, a high-quality solution is characterized by a high recall of the abnormal cases, while producing a satisfactory overall accuracy. Hence, we utilize accuracy, recall and precision for the abnormal cases as the evaluation metrics.

$$\text{Acc} = \frac{tp + tn}{tp + fp + fn + tn}; \quad \text{Rec} = \frac{tp}{tp + fn}; \quad \text{Prec} = \frac{tp}{tp + fp},$$

where tp, fp, fn, tn correspond to the number of true positives, false positives, false negatives and true negatives respectively. Table 2.2 reports the abnormality detection performance on the challenging TUH corpus obtained using `DDxNet` in comparison to state-of-the-art baselines.

Method	Acc (%)	f1-score
(Kachuee *et al.*, 2018)	93.4	-
(Bai *et al.*, 2018a)	97.7	0.864
(Acharya *et al.*, 2017)	93.5	-
(Liu *et al.*, 2015)	94.4	-
`DDxNet`	**98.5**	**0.927**

Table 2.3: `Arrhythmia Classification` - Performance of `DDxnet` With Single-Lead ECG. Best Numbers Are Shown in Bold.

Most importantly, `DDxNet` provides the highest recall so far on this dataset, with an improvement of 4% over the current state-of-the-art *ChronoNet*, without compromising on the overall accuracy. Note that, this performance improvement can be attributed to the key architectural innovations in our approach. Although approaches such as temporal convolution networks (TCN) (Bai *et al.*, 2018a) and the ResNet (Andreotti *et al.*, 2017; Rajpurkar *et al.*, 2017) are expected to be effective in clinical diagnosis tasks as a general solution, `DDxNet` outperforms the former by $\sim 9.5\%$ improvement in recall, while significantly improving over the latter (13.5% improvement in recall).

In the arrhythmia classification task, ECG signal abnormalities are assigned to one of the 5 types of abnormalities elaborated in the 2.2.3. `DDxNet` was evaluated and compared to several competitive baseline models that have used both traditional domain expert-based feature engineering as well as convolutional and recurrent style neural networks. To obtain a holistic evaluation of the prediction quality, we consider the overall accuracy and $f1$-score metrics. Note, the f1-score can be measured as

$$\text{f1} = 2.\frac{\text{Prec.Rec}}{\text{Prec} + \text{Rec}}.$$

Despite an appreciable imbalance in the label distribution of the MIT-BIH dataset, the performance of `DDxNet` exceeds that of existing approaches, achieving an improvement

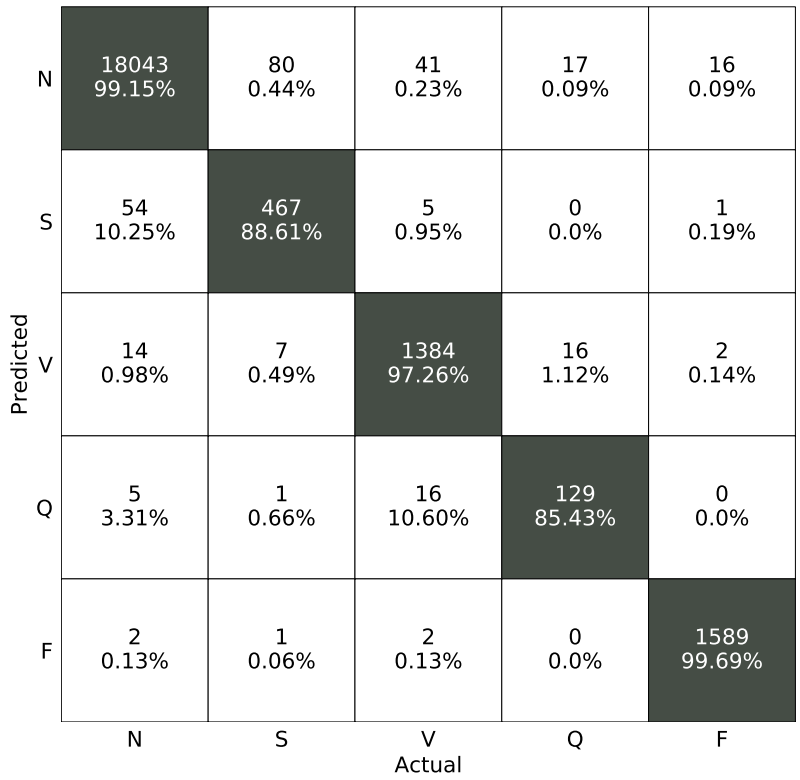


Figure 2.7: Confusion Matrix Obtained Using `DDxNet` For the 5-Category Arrhythmia Classification Task.

of 5% in terms of prediction accuracy over the state-of-the-art baseline (Kachuee *et al.*, 2018). The striking observation is that, the performance reported were obtained using a single channel ECG (lead II), which clearly emphasizes the effectiveness of `DDxNet`, even when the data fidelity is low. Further, as showed in Table 2.3, `DDxNet` leads to major improvements over existing convolutional neural network solutions (Kachuee *et al.*, 2018; Acharya *et al.*, 2017). The confusion matrix from our approach, showed in Fig. 2.7, evidences the ability of `DDxNet` in handling severe class imbalances.

Given the amount of variabilities within the manifestations of myocardial infarction, using the entire 12 channel ECG has been long considered to be essential. However, using the pre-processing described earlier that involves simple heartbeat signal extraction, `DDxNet` accomplishes near-perfect detection using just a single channel ECG

Method	Acc (%)	Rec (%)	Prec (%)
(Kachuee *et al.*, 2018)	95.9	95.1	95.2
(Acharya *et al.*, 2017)	93.5	93.7	92.8
(Kojuri *et al.*, 2015)	95.6	93.3	97.9
(Sharma *et al.*, 2015)	96	93	99
(Bai *et al.*, 2018a)	98	98.7	98.6
(Strodthoff and Strodthoff, 2018)	-	93.3	93.6
(Rajan and Thiagarajan, 2018)	86	96	-
(Reasat and Shahnaz, 2017)	84.54	85.33	-
`DDxNet`	**99.7**	**99.7**	**99.9**

Table 2.4: `Myocardial Infarction Detection` - Performance of `DDxNet` In Detecting Myocardial Infarction From ECG Recordings. Interestingly, Even With a Single Lead ECG, `DDxNet` Achieves Near-Perfect Detection.

(lead II) as shown in Table 2.4, outperforming all other competing solutions including those that use all 12 channels (Kojuri *et al.*, 2015; Sharma *et al.*, 2015). Note, similar to the EEG-based abnormality detection experiment, we use the accuracy, precision and recall metrics.

In the next section, we elaborate on the problem of Global Horizontal Irradiance Forecasting for PV applications and show how robust model priors obtained using causal dilated convolutions resulted in improved forecasting performance.

2.3 Global Horizontal Irradiance Forecasting for Photovoltaic Applications

Solar power has emerged as one of the most popular renewable energy sources in recent years. With increased integration of utility-scale solar arrays in energy grid, solar irradiance forecasting has become significantly important to quantify the photovoltaic array power generation efficiency. Global irradiance incident at ground

depends on the sun's position as well as atmospheric components such as humidity, pressure, dew point, etc. In photovoltaic (PV) systems, power output depends on the sun's intensity which is directly dependent on the cloud cover and other meteorological parameters such as humidity and pressure. The cloud pattern and type in particular affects the way the sun's intensity reaches the horizontal surface on ground and hence cloud shading is one of the major reasons for intermittency and uncertainty in photovoltaic power output (Katoch *et al.*, 2018a; Bosch *et al.*, 2013; Mueller *et al.*, 2004; Chow *et al.*, 2011). Solar irradiance has been shown in literature as a good indicator of PV array power production (El Mghouchi *et al.*, 2016; Duverger *et al.*, 2017).

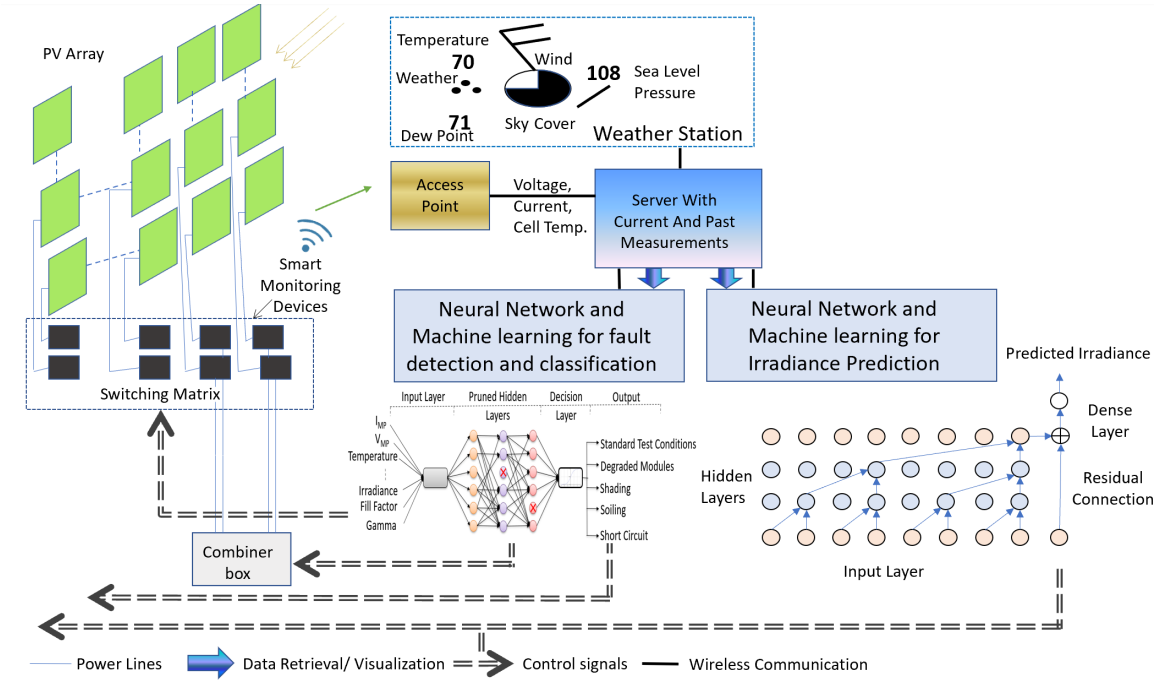


Figure 2.8: Illustration of a PV Array Facility. Smart Monitoring Devices (SMDs) Provide Panel-Level Features (e.g. Current). Weather Station Data Is Used for Irradiance Prediction Which Is Used to Reconfigure PV Arrays to an Efficient Topology by Bypassing Panels Producing Lower Power.

In order to ensure the stability of a PV array output, in this work we propose a weather feature based solar irradiance forecasting strategy which can aid in regulating

and planning the operation of a grid integrated solar array. As shown in Fig. 2.8, we illustrate our cyber-physical approach to a utility scale solar array wherein, we use smart monitoring devices (SMDs) that can read panel-wise voltage, current and temperature features. These panel-wise features aid in automating the fault detection in PV panels and hence, timely repair when needed (Rao *et al.*, 2020). Furthermore, we illustrate the process of irradiance forecasting using the data from a weather station. The proposed irradiance forecasting method will serve as an input to a PV topology reconfiguration setup which aims at maintaining consistent power output as shown in Fig. 2.9. In this setup, we use SMDs with relay connections that provide the capability to switch between several PV panel topologies such as Series Parallel, Total Cross Tied and Bridge Link (Narayanaswamy *et al.*, 2019a). Based on the predicted irradiance, the PV array can switch between the connection topologies to maximize the power yield. Hence, automated irradiance forecasting in conjunction with topology reconfiguration pipeline leads to a smart cyber-physical system approach for maintaining the power yield of a solar array.

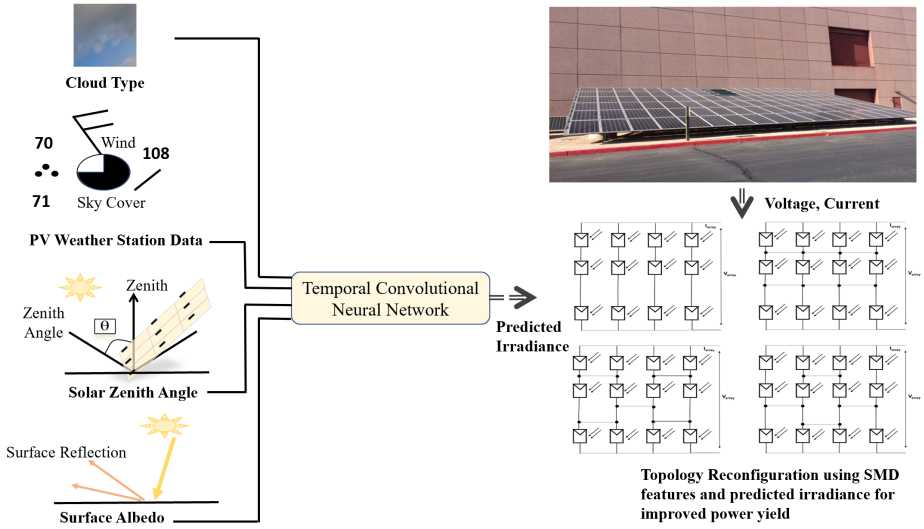


Figure 2.9: Illustration of a PV Weather Station Data Being Used for Irradiance Prediction Which Is Used to Reconfigure PV Arrays to an Efficient Topology by Bypassing Panels Producing Lower Power.

To develop and automate the irradiance forecasting method with robust model priors, we again employ advanced tools from machine learning aimed for the task of time-series forecasting. In order to train the chosen machine learning model, we utilize multivariate meteorological weather feature data (from NSRDB (Sengupta *et al.*, 2018; Zang *et al.*, 2020; Wang *et al.*, 2020)) in conjunction with Global Horizontal Irradiance (GHI) data to perform an hour ahead GHI forecasting. However, we show that with the proposed method, with only 1−day data look back and 30−min data resolution, we are able to achieve improvements or be at par in prediction performance compared to the state-of-the-art baselines. Furthermore, we perform a feature ranking experiment using leave-one out strategy and determine the features that show highest correlation to the irradiance. Hence, the contributions of this work are threefold:

1. We develop a 1−day look back irradiance forecasting model that can be utilized for power output prediction based on the weather features.

2. We identify the weather features which are strongly correlated with the irradiance output by performing a feature ranking experiment.

3. We perform an ablation study based on the data look-back regarding the performance of four regression models for the task of global horizontal irradiance forecasting.

2.3.1 Related Work

There are several models that utilize sky-cameras and satellite images for GHI forecasting (Mueller *et al.*, 2004; Jang *et al.*, 2016; Le Guen and Thome, 2020) and have achieved improved performance compared to physical models (Dolara *et al.*, 2015). However, obtaining such images can be expensive due to the capturing instruments and hence cumbersome in terms of dataset availability and processing. In recent

years, there has been extensive work on weather based irradiance forecasting using deep learning particularly for both opaque and non-opaque overcast skies since such shading conditions cause large fluctuations of irradiance (Katoch *et al.*, 2018a). There are several artificial neural network (ANN) based methods which use meteorological parameters to develop GHI based forecasting models (Wang *et al.*, 2012; Pazikadin *et al.*, 2020). In recent years, long short term memory (LSTM) based architectures with capability to model past history well due to their hidden states are able to improve the solar irradiance prediction accuracy and have become commonplace (Srivastava and Lessmann, 2018; Huang *et al.*, 2020; Bhattacharjee and Chowdhury, 2020). Methods also derive direct normal irradiance using the GHI and compare it to multimodel-ensembles and bootstrap method (Kim *et al.*, 2019). Different variations of combination architectures that include a simple convolutional neural network (CNN) with LSTM model have been previously proposed in literature (Shi *et al.*, 2015; Zang *et al.*, 2020). Attention-based architectures to improve the irradiance forecasting performance (Sharda *et al.*, 2020) have also been explored.

In the recent years, temporal convolutional network (TCN) has become a state-of-the-art method for sequence modelling problems (Song and Brown, 2019). This is due to the fact that for feature extraction as compared to conventional neural network based architectures such as, ANNs and CNNs, TCNs exhibit longer memory and hence capture the data history more effectively. Secondly, due to the dilated convolution usage in TCN, they are also able to extract robust multi-scale temporal features from the data. Furthermore, in practice TCNs are more stable compared to LSTMs since the latter are shown to be harder to converge and hence difficult to train (Bai *et al.*, 2018b). While other methods have utilized TCNs with longer look-backs for irradiance prediction, to the best of our knowledge, this is the first study that utilizes only $1-$day look-back to history alongwith a dilated convolution based architecture i.e. TCN (Bai

et al., 2018b) to get mutli scale temporal features to perform short-term irradiance forecasting. The intuition for using 1−day look-back is to reduce the data storage and computation cost for the forecasting in a utility scale PV array setting.

2.3.2 Proposed Approach

In this work, the task of GHI forecasting is implemented using a TCN architecture. Let the input GHI and meteorological feature data which has 9 features be defined as $\mathrm{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_T)$, $\mathbf{x}_t \in \mathrm{R}^N$ where $N = 10$ is the input data dimension and T represents number of time steps in the input data. Based on our data resolution (1 hour = 2 data points), we choose $T = 48$ time steps i.e. 1 day worth of data. The proposed network produces a mapping from the input to the output which is given by,

$$\hat{\mathrm{y}}_{T+1}, \hat{\mathrm{y}}_{T+2}, \ldots \hat{\mathrm{y}}_{T+h} = \mathcal{F}(\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_T) \tag{2.2}$$

where $\hat{\mathrm{y}}_i$ is the estimated irradiance values, h is number of predicted time steps and $\mathcal{F}$ represents a custom TCN model. In our work, we are forecasting 2 time steps i.e. 1 hour to the future. The network does apply causal constraints on this mapping i.e. output $\hat{\mathrm{y}}_{t+1}$ only depends on inputs until time step t. The TCN network $\mathcal{F}$ minimizes the loss $\mathcal{L}(\mathrm{y}_{T+1}, \mathrm{y}_{T+2}, \ldots \mathrm{y}_{T+h}, \mathcal{F}(\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_T))$ between the actual and predicted outputs.

A temporal convolutional network architecture used for GHI forecasting is illustrated in Fig. 2.11. We customize the TCN architecture such that it is composed of 3 TCN modules wherein each module has dilated causal 1-D convolutional layers. We implement TCN module such that it is composed of 2 convolution layers with dilation rate varying up to 2. The primary advantage of TCNs over regular convolutional networks is the use of dilations. Unlike conventional convolution, dilated convolutions increase the receptive field size (as shown in Fig. 2.10) based on the dilation factor.

Increased receptive field can see data much farther in the past and hence extract more accurate features. This aids in improved and robust mapping of temporal dependencies and hence richer context. A dilated convolution on element s of our 48 time step input sequence X is given by,

$$D(s) = \sum_{i=0}^{k-1} f(i)\mathbf{x}_{s-di} \tag{2.3}$$

where $\mathrm{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_T)$ represents the input sequence, d is the dilation factor, f is the filter, k is the filter size, and $s - di$ are the past input samples.

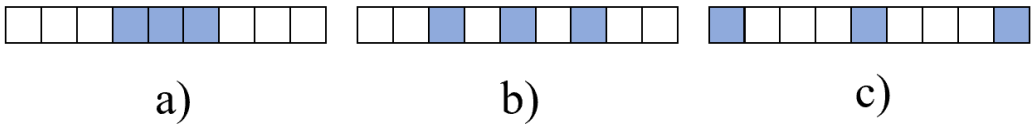


Figure 2.10: Illustration of Dilated Convolution With Filter Size 3 And Increasing Dilation Rates a) $d = 1$, b) $d = 2$ and c) $d = 4$.

Causality in TCNs prevents future data leakage and is achieved by choosing non-negative values of i. The dilated convolution with filter size 3 and increasing dilation rate is represented in Fig. 2.10. We also experimented with dilation rate 4 however adding a higher dilation rate than 2 did not improve the performance significantly specifically when input sequence length is limited to 1 day. Apart from dilated causal convolutions, as shown in Fig. 2.11, we also use residual connections in the TCN architecture. The output of the two convolutional layers will be added to the current input to produce the input for the next module which is given by,

$$\mathrm{X}^{j+1} = \mathcal{T}(\mathrm{X}^j) + \mathrm{X}^j \tag{2.4}$$

where X^j represents the input to the current TCN module, $\mathcal{T}(\mathrm{X}^j)$ represents the transformed input and X^{j+1} represents the input to the next module. To make sure that $\mathcal{T}(\mathrm{X}^j)$ and X^j have same dimensions, we apply an optional linear transformation to the input. The residual connections aid the model to learn the input distribution

as well as the transformed versions of the input distribution. Finally, throughout the TCN architecture, we use Rectified Linear Unit (ReLU) activation function and the dropout is fixed to 0.01. We limit the architecture to 3 TCN modules to avoid overfitting to the training data.

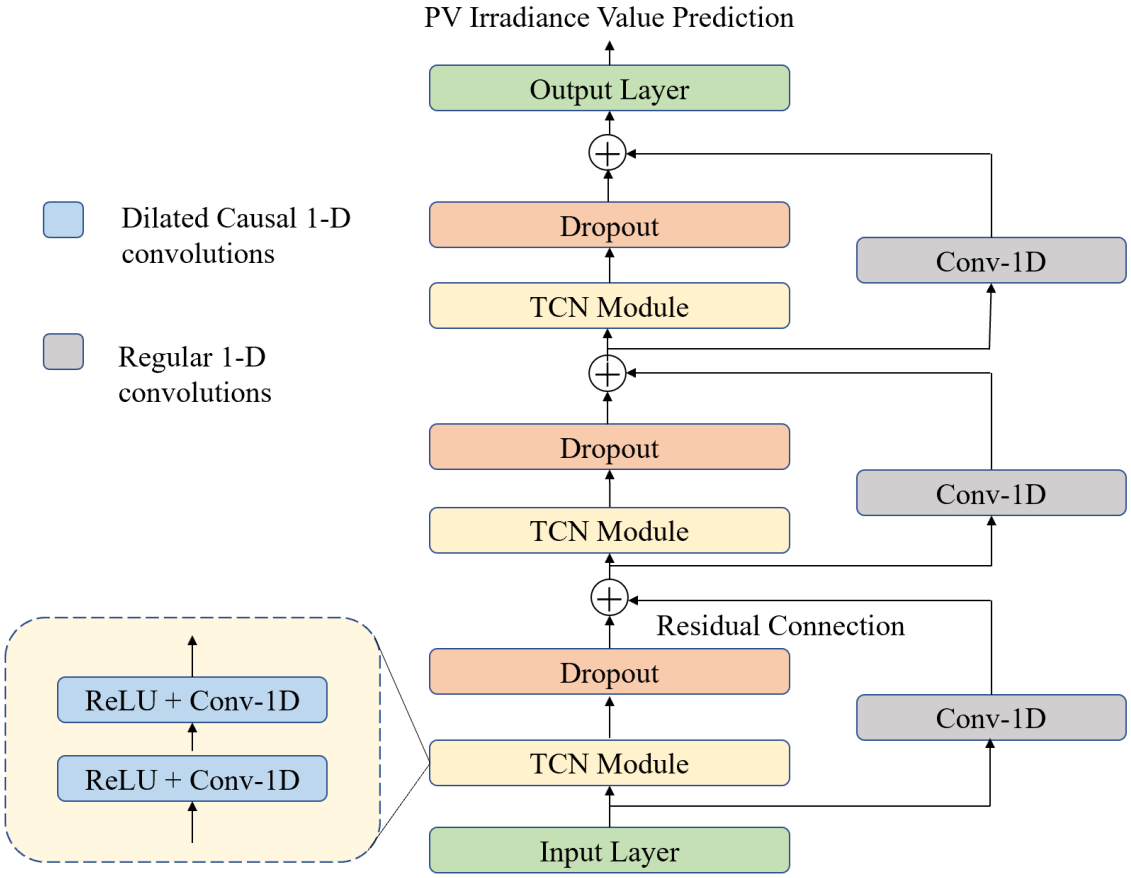


Figure 2.11: Illustration of Proposed Temporal Convolutional Architecture With Three TCN Modules.

2.3.3 Experiments

For GHI forecasting, we utilize the multivariate weather feature data from National Solar Radiation Database (NSRDB) as our input to predict the target GHI. The end-to-end GHI forecasting pipeline is illustrated in Fig. 2.12. The first step in the pipeline after obtaining the weather data is data cleaning and pre-processing. We

follow standard data preparation steps i.e. imputation wherein we substitute the missing data values followed by normalizing the data. We then split the dataset into training, validation and test sets and create 3−D data tensor with batch, time steps and channels. The proposed TCN model is trained and evaluated on training and test splits, respectively. The hyperparameter tuning of TCN architecture is performed using validation split of the data. The final step of the pipeline is data denormalization which is used to obtain the forecasted GHI values from the predicted test data.

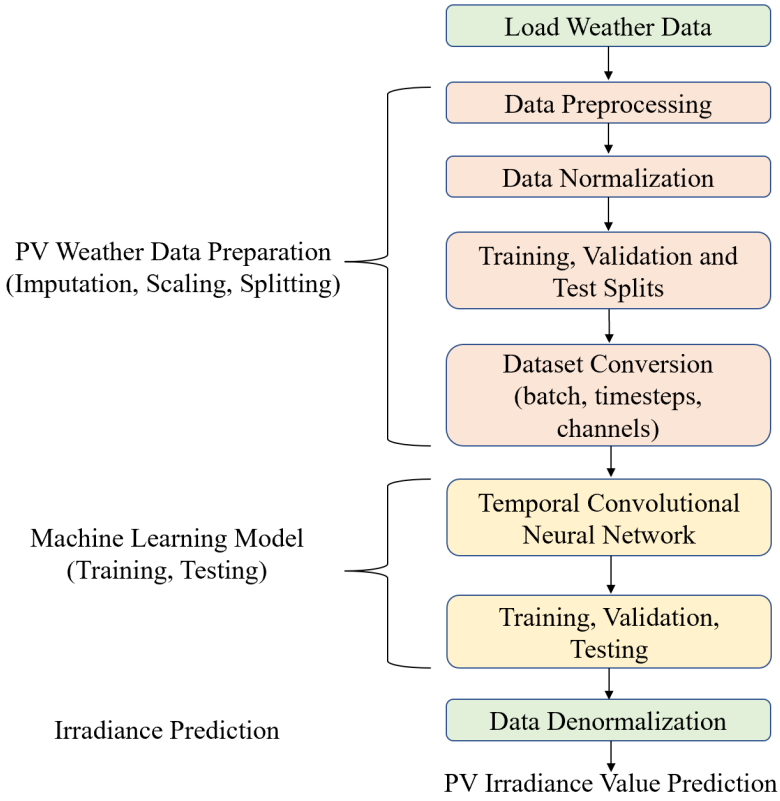


Figure 2.12: Illustration of Proposed Global Horizontal Irradiance Forecasting Pipeline.

In the following sub-sections, we elaborate on the input data features and how to pre-process the data to make it suitable for TCN architecture. We also discuss the baselines and the chosen hyperparameters for the proposed TCN model.

Data Preprocessing The proposed method is trained and evaluated on 1 year long data from the NSRDB database which has been collected at a 30-min time resolution (Sengupta *et al.*, 2018). The data provides several markers for identification such as, 'city', 'state', 'country', 'latitude', 'longitude', 'time zone', 'elevation' and 'local time zone'. The specified data includes 9 major features which are utilized to predict the GHI value. The features include 'dew point' (temperature below which water droplets begin to condense), 'solar zenith angle' (angle between sun's rays and the vertical), 'cloud type', 'surface albedo' (fraction of the sunlight reflected by the surface of the Earth), 'wind speed', 'precipitable water' (total atmospheric water vapor contained in a vertical column of unit cross-sectional area extending between any two specified levels), 'relative humidity' (a present state of absolute humidity relative to a maximum humidity given the same temperature), 'temperature' and 'pressure'. The dataset identifies cloud type into 13 categories including clear, probably clear, fog, water, super-cooled water, mixed, opaque ice, cirrus, overlapping, overshooting, unknown, dust and smoke types.

Based on the resolution of data capture, every hour has 2 data points which amounts to 17520 samples available in the dataset for 365 days in an year. Across all methods, we perform a train-test split of 70% and 30% and testing data is further split by 0.5 validation ratio. We perform standard scaling on all the features and the categorical cloud type data is converted using one hot encoding to be used as an input feature. The $10-$D input dataset including the GHI feature is reshaped to have 48 timesteps in every batch (1 day data) and univariate GHI output is reshaped to have 2 timesteps (1 hr forecast).

Metrics and Hyperparameters We compare with three baseline methods which involve a $2-$layer, $1-$D CNN architecture (Ghimire *et al.*, 2019), a $2-$layer artifi-

cial neural network (Wang *et al.*, 2012) and an LSTM architecture with 16 hidden units (Srivastava and Lessmann, 2018). We chose state-of-the-art neural architectures used in literature for irradiance forecasting as our baselines. This helps us to determine whether the specific design choices of dilated convolution and residual connection aid TCN network over the baselines. We use Adam optimizer with a learning rate of $2e-3$ and a batch size of 32 for training all the neural network based models. We use mean squared error (MSE) as a loss function to train TCN and all baseline methods. Finally, we evaluate the performance of all the methods on test set using root mean squared error (RMSE) and mean absolute error (MAE) along with mean squared error. The equations for these error metrics are given by,

$$MSE = \frac{1}{n}\sum_{i=0}^{n}(y_t^i - \hat{y}_t^i)^2 \tag{2.5}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(y_t^i - \hat{y}_t^i)^2} \tag{2.6}$$

$$MAE = \frac{1}{n}\sum_{i=0}^{n}(|y_t^i - \hat{y}_t^i|) \tag{2.7}$$

where n is the number of datapoints in the data batch. All the training of TCN and baseline methods is performed on NVIDIA GTX 1080.

2.3.4 Performance Evaluation

In this section, we show the evaluation results using the TCN and baseline methods on the test dataset. As mentioned in Section. 2.3.3, we use the MSE, RMSE and MAE as the error metrics for performance evaluation. Note that all the methods are trained on standard scaled data.

It is evident from Table. 2.5, that LSTM and TCN model provides significant improvement over other methods. With LSTM, the improvement is attributed to

Method	MSE	RMSE	MAE
Conv 1D	0.16	0.4	0.23
Dense	0.12	0.35	0.18
LSTM	0.0068	0.082	0.052
TCN	**0.0056**	**0.075**	**0.051**

Table 2.5: Error Metrics on Test Data. It Is Evident That TCN Provides the Lowest Error Compared to the Baselines.

the fact that the hidden state is able to capture the temporal information quite well. With TCN, the improvements are primarily due to the efficient feature extraction and rich temporal context attributed to the dilated convolutions. Please note that all the comparisons in Table. 2.5 use all the input features while training and testing.

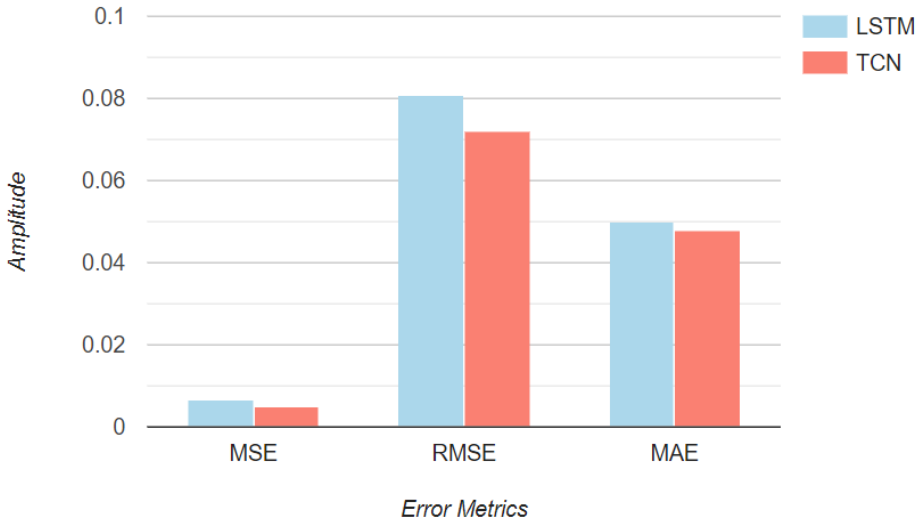


Figure 2.13: Error Metrics on Test Data With 3−Day Look-Back.

We also perform an ablation study based on the past look-back length using TCN and LSTM models. When the models are exposed to longer sequences as shown in the results plotted in Figs. 2.13 and 2.14, both the methods are able to improve their relative performance. This can be attributed to the fact that due to longer input context, the models are able to better capture the inherent patterns in the temporal

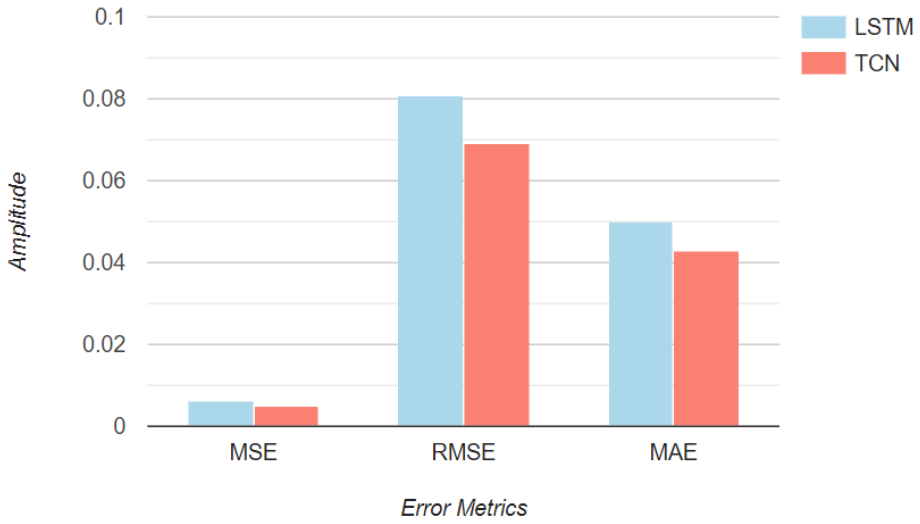


Figure 2.14: Error Metrics on Test Data With 5−Day Look-Back.

data. However, an interesting result is shown in Fig. 2.15 i.e. as the sequence length of the input data increases, the difference in mean square error between TCN and LSTM also increases. We infer this behavior to the fact that due to dilations, TCNs exhibit longer memory and capture data history effectively.

Futhermore, we also performed feature ranking experiment to determine which features exhibit a strong correlation with GHI using the root mean squared error. We performed leave one out experiment for all 9 input features except GHI itself to determine the drop in prediction performance in absence of a certain feature. Based on careful empirical analysis, Solar Zenith Angle (SZA), Cloud Type and Surface Albedo represent the feature set to be most strongly correlated with GHI. For these three features, the RMSE on the test set increased significantly as much as 0.090 in the absence of Solar Zenith angle, upto 0.087 in the absence of Cloud Type feature and 0.079 without the Surface Albedo as the feature.

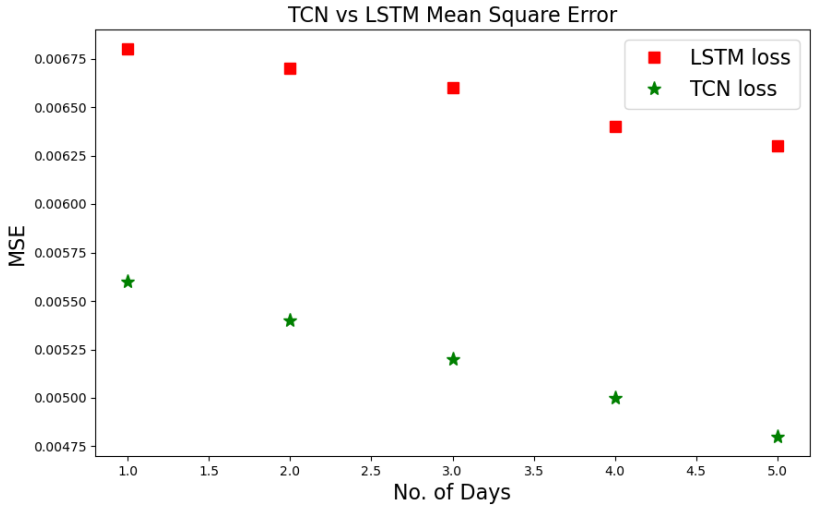


Figure 2.15: Mean Squared Error for the TCN and LSTM Model as Sequence Length Increases. It Is Evident That as Number of Days Increase, TCN Performs Consistently Better Than LSTM Due to Efficient Data History Capture.

2.4 Summary

In this section, we investigated the use of strong model prior in sequence modelling problems. We considered applications such as, Audio Source Separation, Clinical Time-series based Diagnosis and Global Horizontal Irradiance Forecasting for PV Applications. For all the applications, we make similar architectural design choices i.e. using dilated convolutions. This was motivated by the fact that robust multi-scale feature extraction achieved by dilated convolutions is able to provide richer context and give a bird's eye view of longer temporal histories. Based on the empirical evidence, we show that robust features obtained using such model priors resulted in better task performance. In the following chapter, we will investigate how data priors are significantly effective for image classification based tasks.

Chapter 3

TASK AND DOMAIN PRIORS FOR DATA EFFICIENT LEARNING

In this chapter, we show how due to varied applicability of deep learning techniques, data handling for deep learning based methods is not trivial. We argue that data should be utilized in a more exploratory sense such that we can determine which priors can be effectively used. We primarily work with task (information from multiple classification tasks) and domain priors (information from multiple data distributions) and develop a mechanism which will learn a semantic space from these data priors and will aid in learning meaningful attributes from the inferred semantic space. The structure of this space will assist in computing semantic similarities and will then be used to select powerful priors. This semantic space can give insights in terms of: 1) Finding new apparent connections which were not comprehended by humans, 2) sharing knowledge across disparate set of problems. The motivation for this research is also based on the success of recent model agnostic approaches which exploit semantic relationships between fine-grained tasks/domains. There are currently successful methods and complex optimization paradigms like multi-task multi-domain learning (Yang and Hospedales, 2015) and meta learning (Finn *et al.*, 2017b) which leverage supervised data for model development for related tasks. These approaches often rely on meta optimization to make a model robust to systematic task or domain shifts. However, in practice, the performance of these methods can suffer, when there are no coherent semantic relationships between the tasks (or domains). In this work, we present `Invenio`, an optimization protocol which can account for covariate shifts, can infer semantic similarities between a given set of tasks and can provide insights into the complexity of transferring knowledge between different tasks or different domains.

More specifically, we introduce `Invenio` as an optimization protocol using a specific problem setting. In the following chapter, we elaborate on how `Invenio` can be re-purposed for different applications and show the algorithm setting for each application followed by extensive empirical analysis.

3.1 Knowledge Transfer

In a typical generic supervised learning setting, we are given a dataset $\mathcal{D} = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^{N}$ with inputs $\mathrm{x} \in \mathcal{X}$ and labels $\mathrm{y} \in \mathcal{Y}$. The goal is to infer a model $\mathcal{F}$ with parameters ϕ that maps from inputs x to the outputs y. The parameters ϕ are inferred using the following MAP formulation:

$$\arg\max_{\phi} \log p(\phi|\mathcal{D}) = \arg\max_{\phi} \log p(\mathcal{D}|\phi) + \log p(\phi). \tag{3.1}$$

In practice, the first term is approximated using the empirical risk $\sum_i \log p(\mathrm{y}_i|\mathrm{x}_i, \phi)$, while the second term is often an appropriately chosen regularizer. While a learning task can be specified by a dataset $\mathcal{D}$ or by the surrogate function $\mathcal{F}(\phi)$, the same task can also be specified by other datasets. This implies that the complexity of a task depends on the underlying structure shared among data points, and the variability that samples exhibit with respect to the shared structure. Formally, the complexity of a task can be expressed as (Achille *et al.*, 2019b):

$$\mathcal{C}(\mathcal{D}) \coloneqq \min_{p(\mathrm{y}|\mathrm{x})} \sum_i - \log p(\mathrm{y}_i|\mathrm{x}_i) + \mathrm{K}(p), \tag{3.2}$$

where the minimum is defined over all computable probability distributions $p(\mathrm{y}|\mathrm{x})$ and $\mathrm{K}(p)$ denotes the Kolmogorov complexity of $p(\mathrm{y}|\mathrm{x})$. By effectively characterizing the complexity of tasks, one can quantify how well one can transfer between two given tasks. Instead of solving pairwise adaptation between tasks A and B, our goal is to learn from K different observed tasks, $\{\mathcal{D}_k \in p(\mathcal{D})\}$, in order to generalize to novel unseen tasks from the unknown distribution $p(\mathcal{D})$. This setup can be modelled into

several scenarios i.e. when all datasets use same output space or each dataset has a different output space. In this section, we present `Invenio` from the viewpoint when we are given a set of observed datasets $\{\mathcal{D}_k\}_{k=1}^K$, where all datasets use the same output space $\mathcal{Y}$. Since the classification task is the same across datasets (e.g. MNIST and SVHN for digit recognition), we take into account all parameters θ^k for each dataset while quantifying the transferability. In the next section, we provide the algorithmic description for `Invenio`.

3.1.1 Proposed Approach

We begin by presenting an overview of meta optimization which serves as a backbone to `Invenio`.

Backbone: Meta Optimization Assume we have access to multiple observed datasets $\mathcal{D}_{\text{obs}} = \{\mathcal{D}_1, \cdots, \mathcal{D}_K\}$. Each of these datasets are characterized by differences in the marginal feature distribution with identical conditional distributions (domain shifts). These observed datasets can be used when inferring the parameters ϕ for a given task from a new dataset $\mathcal{D}$,

$$\arg\max_{\phi} \log p(\phi \mid \mathcal{D}, \mathcal{D}_{\text{obs}}) \tag{3.3}$$

$$= \log \int_{\Theta} p(\phi \mid \mathcal{D}, \theta) p(\theta \mid \mathcal{D}_{\text{obs}}) d\theta. \tag{3.4}$$

As it can be observed, instead of using the observed datasets directly to optimize for ϕ , we first infer meta-parameters θ, and subsequently adapt ϕ using both θ and $\mathcal{D}$. More specifically, we need to solve the following two problems:

$$\text{Meta learning} : \theta^* = \arg\max_{\theta} \log p(\theta | \mathcal{D}_{\text{obs}}), \tag{3.5}$$

$$\text{Adaptation} : \arg\max_{\phi} \log p(\phi | \theta^*, \mathcal{D}). \tag{3.6}$$

In order to solve the problem of generalizing to the same task from the unobserved dataset, meta learning techniques first split the given set of observed datasets/domains $\mathcal{D}_{\text{obs}}$ into meta-train and meta-test domains, $\mathcal{D}^{tr}$ and $\mathcal{D}^{ts}$, and directly parameterize $\phi^* = f(\mathcal{D}^{tr}; \theta^*)$. More importantly, by learning θ such that the corresponding ϕ can be effective for $\mathcal{D}^{ts}$ enables us to define this general bi-level optimization objective:

$$\begin{aligned} \theta^* = &\max_{\theta} \sum_{\mathcal{D} \in \mathcal{D}^{ts}} \log p(\phi | \mathcal{D}) \\ &\text{where } \phi = f(\mathcal{D}^{tr}; \theta^*). \end{aligned} \tag{3.7}$$

While a broad class of solutions exist for solving (3.7), we focus on optimization-based inferencing methods such as, MLDG (Li *et al.*, 2017a), meta-SGD (Li *et al.*, 2017b) and model agnostic meta learning (MAML) (Finn *et al.*, 2017a). In this approach, instead of learning the parameterization f, we obtain ϕ by fine-tuning the meta parameters θ using $\mathcal{D}^{ts}$. More specifically, we begin by defining the meta-train step as follows: We optimize for the meta parameters θ using aggregated losses from all the $K^\dagger$ meta-train domains $\mathcal{D}^{tr}$:

$$\mathcal{L}(\theta) = \frac{1}{K^\dagger} \sum_{k=1}^{K^\dagger} \frac{1}{N_k} \sum_{(\mathrm{x},\mathrm{y}) \in \mathcal{D}_k^{tr}} \ell\Big(\mathrm{y}, \mathcal{F}(\mathrm{x}; \theta)\Big). \tag{3.8}$$

Here, ℓ denotes an appropriate loss function, e.g. cross entropy and N_k is the number of examples in each of the meta-train datasets. This loss function is parameterized using θ and hence the gradients $\nabla_\theta \mathcal{L}(\theta)$ can be used to update the meta parameters:

$$\hat{\theta} = \theta - \alpha \nabla_\theta \mathcal{L}(\theta), \tag{3.9}$$

where α denotes the step size. In the meta-test step, the estimated parameters are evaluated on the $K^\ddagger = K - K^\dagger$ meta-test domains to virtually measure the generalization performance. Consequently, the aggregated loss function obtained using

the updated parameters on the test domains can be written as

$$\mathcal{G}(\hat{\theta}) = \mathcal{G}(\theta - \alpha\nabla_\theta\mathcal{L}(\theta)) = \frac{1}{K^\ddagger}\sum_{j=1}^{K^\ddagger}\frac{1}{N_j}\sum_{(\mathrm{x},\mathrm{y})\in\mathcal{D}_j^{ts}}\ell\Big(\mathrm{y}, \mathcal{F}(\mathrm{x};\hat{\theta})\Big). \tag{3.10}$$

MLDG updates the parameters θ such that the best ϕ for each of the meta-test domains are only a few gradient descent steps away from θ. Hence, the overall objective is:

$$\arg\min_\theta \mathcal{L}(\theta) + \beta\mathcal{G}(\theta - \alpha\nabla_\theta\mathcal{L}(\theta)). \tag{3.11}$$

To intuitively understand this objective, we can perform first-order Taylor expansion on the second term to obtain

$$\mathcal{G}(\theta - \alpha\nabla_\theta\mathcal{L}(\theta)) = \mathcal{G}(\theta) - \alpha\nabla_\theta\mathcal{L}(\theta).\nabla_\theta\mathcal{G}(\theta), \tag{3.12}$$

where the expansion is carried out around θ. The meta optimization process amounts to minimizing the losses on meta-train domains while maximizing the dot product between parameter sensitivities from meta-train and meta-test domains. When such a strategy is applied to domains that are significantly dissimilar, the resulting meta-parameters can be ineffective during generalization. Hence, we propose a structured meta learning protocol `Invenio`, which can jointly infer the inherent semantic structure and build meta parameters that can effectively generalize to unseen domains.

`Invenio` - A Structured Meta Learning Protocol Learning invariant feature representations and identifying redundancies across domains to support cross domain knowledge transfer has been of key research interest, e.g. (Wang and Deng, 2018). We argue that incorporating such a characterization into meta optimization is critical to achieving significant performance improvements in applications such as domain adaptation. We propose `Invenio`, a structured meta learning protocol, that jointly infers an implicit distance across same task from different datasets, and at the same time leverages the similarities while optimizing for the meta parameters.

In our formulation, we begin by assuming a different set of meta parameters for each of the domains in $\mathcal{D}^{tr}$, i.e., $\mathrm{y} = \mathcal{F}_k(\mathrm{x}; \theta_k), \forall(\mathrm{x}, \mathrm{y}) \in \mathcal{D}_k^{tr}$. Consequently, the posterior on the meta parameters is modeled as a mixture of posteriors from all meta-train datasets:

$$p(\theta|\mathcal{D}_{obs}) = \bigcup_{k=1}^{K} p(\theta_k|\mathcal{D}_k). \tag{3.13}$$

$$p(\theta|\mathcal{D}^{tr}) = \bigcup_{k=1}^{K^\dagger} p(\theta_k|\mathcal{D}_k^{tr}). \tag{3.14}$$

The inference of parameters for a task from a novel unseen dataset/domain $\mathcal{D}$ can be formulated as

$$p(\phi|\mathcal{D}, \mathcal{D}^{tr}) = \log \int_{\Theta} p(\phi|\mathcal{D}, \{\theta_k\})\ p(\theta|\mathcal{D}^{tr}). \tag{3.15}$$

Here, the first term performs fine-tuning from $K^\dagger$ different sets of meta parameters, while the second term is the posterior on θ, described using the mixture in eq. (3.14). Conceptually, this can be viewed as identifying the mixture probabilities conditioned on the task from the unseen domain $\mathcal{D}$, such that the most relevant meta parameters can be used to drive the adaptation. Such an optimization naturally induces a semantic structure and the mixture associations can be used to construct embeddings that reveal intricate relationships between domains. To the best of our knowledge, currently there exists no approach to infer such embeddings from the data, while also supporting adaptation to task from unseen domains. In the next section, we describe how `Invenio` can be used for creating the embeddings between the datasets.

3.1.2 `Invenio` for Constructing Semantic Space of Domains

Similar to the classical meta learning setup, we also split the set of observed datasets into meta-train ($\mathcal{D}^{tr}$) and meta-test ($\mathcal{D}^{ts}$) sets. Following the structured meta learning formulation in Section 3.1.1, we compute the loss for each of the meta-train

datasets $\mathcal{D}_k^{tr}$ using the corresponding meta parameters θ_k as follows:

$$\mathcal{L}_k(\theta_k) = \frac{1}{N_k} \sum_{(\mathrm{x},\mathrm{y}) \in \mathcal{D}_k^{tr}} \ell\Big(\mathrm{y}, \mathcal{F}_k(\mathrm{x}; \theta_k)\Big). \tag{3.16}$$

Here, the term $\mathcal{L}_k(\theta_k)$ is the empirical risk for samples from $\mathcal{D}_k$ obtained using model parameters θ_k. Updating the parameters θ_k using a gradient descent step can be expressed as:

$$\hat{\theta}_k = \theta_k - \alpha \nabla_{\theta_k} \mathcal{L}_k(\theta_k). \tag{3.17}$$

Since the posterior on meta parameters is modeled as a mixture of posteriors $p(\theta_k|\mathcal{D}_k^{tr})$, performing adaptation requires identifying which meta parameters should we attend to for generalizing to a meta-test dataset $\mathcal{D} \in \mathcal{D}^{ts}$. Loosely speaking, this is akin to estimating assignment probabilities for each of the components in the mixture $p(\theta|\mathcal{D}^{tr})$, conditioned on the test dataset. Interestingly, as we will show later, this mixture component association carries crucial information to construct embeddings. Following the intuition in eqn. 3.12, we propose to quantify the transferability between a meta-train dataset $\mathcal{D}_k^{tr}$ and a meta-test dataset $\mathcal{D}_j^{ts}$ as the dot product between gradients with respect to the weights θ_k relative to the losses evaluated on both $\mathcal{D}_k^{tr}$ and $\mathcal{D}_j^{ts}$. In other words,

$$\eta_{kj} = \sum \nabla_{\theta_k} \mathcal{L}_k(\theta_k) . \nabla_{\theta_k} \mathcal{L}_j(\theta_k). \tag{3.18}$$

Here, we measure the similarity between sensitivites of θ_k with respect to the two tasks from different distributions, while learning the model ϕ_j for $\mathcal{D}_j^{ts}$. The summation in the above expression is over all parameters in the set θ_k, and the gradient estimates are obtained by summing over all mini-batches. Note, other similarity metrics and ranking statistics can be used in lieu of simple dot products; however, we find this simple metric to be effective in practice. This formulation corroborates with existing formulations in (Achille *et al.*, 2019a,b) in treating gradients of weights of a neural

network relative to a task-specific loss to characterize tasks. Given these mixture component associations, $\eta^{'s}$, we can define the objective for the meta-test step as follows:

$$\mathcal{G}(\hat{\theta}_k) = \mathcal{G}(\theta_k - \alpha \nabla_{\theta_k} \mathcal{L}_k(\theta_k)) = \sum_{j=1}^{K^{\ddagger}} \bar{\eta}_{kj} \frac{1}{N_j} \sum_{(\mathrm{x},\mathrm{y}) \in \mathcal{D}_j^{ts}} \ell\Big(\mathrm{y}, \mathcal{F}_j(\mathrm{x}; \hat{\theta}_i)\Big). \tag{3.19}$$

Note, the weights $\bar{\eta}_{kj}^{'s}$ are obtained normalizing $\bar{\eta}_{kj}, \forall j$ to sum to 1. By systematically controlling, which meta-test task distributions each θ_k generalizes to, an understanding of the inherent semantic structure between all observed datasets $\mathcal{D}_{obs}$ can be obtained. The overall objective for updating each meta-parameter set θ_k using the structured meta-optimization can thus be written as,

$$\arg\min_{\theta_k} \mathcal{L}_i(\theta_k) + \beta \mathcal{G}(\theta_k - \alpha \nabla_{\theta_k} \mathcal{L}_k(\theta_k))$$

A detailed algorithm of our approach is outlined in Algorithm 1.

Algorithm 1: `Invenio` On Multiple Observed Datasets for the Same Task.

Input: Set of observed datasets $\mathcal{D}_{obs}$

Output: Meta-parameters for the learning task on all observed datasets $\mathcal{D}_{obs}$

Initialization: Parameters θ_k for each $\mathcal{F}_k, \forall k \in 1, \cdots, |\mathcal{D}_{obs}|$. Set hyper-parameters α, β, δ ;

Random Split: $\mathcal{D}^{tr}$ and $\mathcal{D}^{ts} \leftarrow \mathcal{D}_{obs}$;

for *iter* ***in*** n_{iter} **do**

  for k *in* $1 \cdots K^{\dagger}$ **do**

    Compute loss $\mathcal{L}_k(\theta_k)$ and gradients $\nabla_{\theta_k}\mathcal{L}_k(\theta_k)$ using $\mathcal{D}_k^{tr}$;

    Update $\hat{\theta}_k = \theta_k - \alpha\nabla_{\theta_k}\mathcal{L}_k(\theta_k)$;

  end

  for k *in* $1 \cdots K^{\dagger}$ **do**

    for j *in* $1 \cdots K^{\ddagger}$ **do**

      Estimate $\eta_{kj} = \sum \nabla_{\theta_k}\mathcal{L}_k(\theta_k).\nabla_{\theta_k}\mathcal{L}_j(\theta_k)$;

    end

    Obtain normalized scores $\bar{\eta}$ and compute meta-test loss $\mathcal{G}(\hat{\theta}_k)$ in (3.19) ;

    Update θ_k:

$$\theta_k = \theta_k - \delta\frac{\partial(\mathcal{L}_k(\theta_k) + \beta\mathcal{G}(\theta_k - \alpha\nabla_{\theta_k}\mathcal{L}_k(\theta_k))}{\partial\theta_k} \ ;$$

  end

  New random split: $\mathcal{D}^{tr}$ and $\mathcal{D}^{ts} \leftarrow \mathcal{D}_{obs}$

end

3.1.3 Experiments

We consider a set of covariate shifts arising from image transformations on Cifar-10 and study the semantic space of different domains (same task). In this experiment,

we consider 53 different variants of the CIFAR-10 dataset (Krizhevsky and Hinton, 2010), obtained using a broad class of image transformations, while solving the same task of multi-class classification (10 classes). Here is the complete list of domain shifts considered: (i) *Rotation*: 7 variants were generated by rotating the images, where the degree of rotation was varied between 0 to 90; (ii) *Flip*: We generated 2 datasets by applying horizontal and vertical flips to the images. These transformations can be viewed as special cases of *Rotation*; (iii) *Affine*: We constructed 14 domains by applying different affine transformations to images and this was carried out by varying the settings for scale and shear; (iv) *Color*: 20 different datasets were created by manipulating parameters pertinent to color transformations, namely brightness, saturation, contrast and hue; and (v) *Filter*: We used blurring and Gaussian smoothing techniques to create 10 variants of the base domain. While Gaussian smoothing produces blurring by applying Gaussian function based transformation on image pixels, the Box Blur filter replaces each pixel by the average of its neighboring pixels. Intuitively, we expect geometric transformations such as *Affine*, *Rotation* and *Flip* to be related among themselves and can benefit by shared feature representations. On the other hand, transformation such as hue, saturation, contrast and brightness are expected to be strongly related.

Each dataset is comprised of 300 randomly chosen samples from each class and we use the following architecture to design $\mathcal{F}_k$ - Conv(3,20,5,1), ReLU, MaxPool, Conv(20,50,5,1), ReLU, MaxPool, Linear(2450,500), ReLU, Linear(500,10), ReLU. The hyperparameter values α, δ and β were set to 1e-4, 1e-3 and 0.1 respectively.

3.1.4 Performance Evaluation

Figure 3.1(a) provides a 2D visualization of the semantic space obtained by applying truncated SVD on the η matrix $\mathrm{S} \in \mathbb{R}^{53\times53}$ between the set of domains. With two

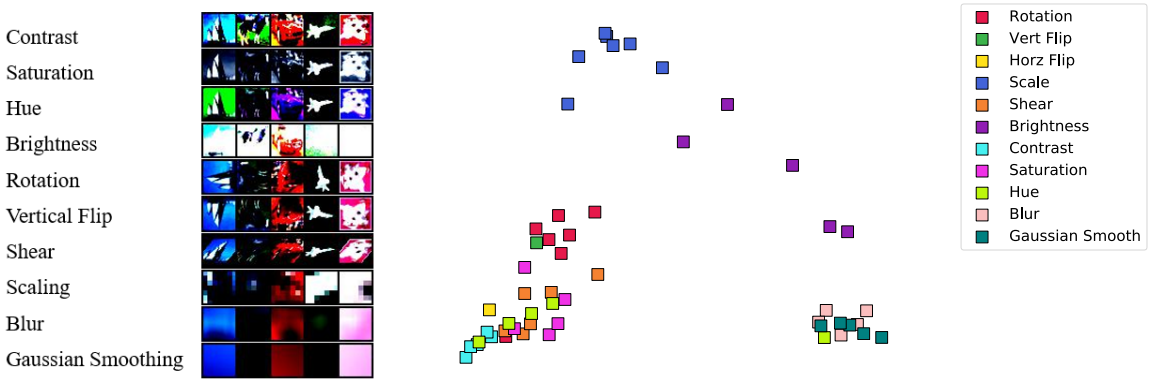


Figure 3.1: A Semantic Space of Domains for the Cifar-10 Classification Task. We Provide a 2D Visualization of the Domain Embeddings Obtained Using `Invenio`.

domains that are in a close neighborhood, we expect the task of adapting a model from one case to another to be effective, even with very few examples. As it can be observed, the structure largely aligns with our hypothesis, i.e., the geometric transforms such as, rotation, flip and shear are closely related to each other. An interesting outcome is that the scale transformation does not belong in the same part of the semantic space as the other geometric transformations. Similar observations can be made about domains constructed based on color transformations to the original images. It is evident from Figure. 3.1(a) that the datasets generated by manipulating hue, saturation and contrast respectively, are closely related to each other. However, brightness changes manifest as being completely unrelated to other standard color transformations. As illustrated in Figure 3.1, this may be partly due to the high degree of brightness change that we applied, which caused the shadows/darker regions to mask the crucial features like edges. On the other hand the *Contrast* transformation makes separation between dark and bright regions more prominent. Finally, the two filtering transformations that we considered are found to carry shared knowledge about the images, since both of them produce low-pass variants of the original images.

3.2 Summary

In this chapter, we studied the effect of data priors for knowledge transfer in image classification applications. In terms of data priors, we primarily considered task and domain priors. Domain Priors imply the prior information that can be leveraged from different data distributions for the same task. Unlike conventional transfer learning, where the assumption is that all the available data distributions contribute equally such that model can perform well on the new distribution we proposed a new optimization paradigm i.e. `Invenio`, which shows that systematic knowledge transfer is the key. It was motivated by the literature that extremely disparate distributions have been shown to lead to inferior generalization on new data. `Invenio` achieves this systematic knowledge transfer through a semantic space that is obtained using the gradients during the model training step. In this chapter, we exhibit that the inferred semantic space is able to quantify the relationships between different data distributions to show which distribution is more closely related to the other distribution. In next chapter, we elaborate on different applications using `Invenio` paradigm and how the semantic space weightings are used to achieve improved generalization (on both domains and tasks).

Chapter 4

LEVERAGING TASK AND DOMAIN PRIORS FOR COMPUTER VISION APPLICATIONS

In the last chapter, we showed how data priors are significant and can be used to map the semantics between different domains. We also presented `Invenio`, a generic optimization scheme which incorporates a machinery to infer semantic relationships between different sets of data. Furthermore, we showed a proof of concept experiment where we add different kinds of transformations to the data and during training `Invenio` is able to cluster the data which has been transformed in a similar way or shares same semantics. In this chapter, we explore how to leverage the data priors obtained using `Invenio` for different computer chapter include test time multi-source domain adaptation, semantic space of tasks, adaptable transfer learning and multi-task learning.

4.1 Test Time Multi-Source Domain Adaptation

Multi-source domain adaptation is a sub-category of transfer learning in which we aim at learning from multiple observed data distributions and use the learnt parameters on a different unlabelled data distribution. In our application, we are considering test time adaptation i.e. we assume that the model must generalize to new data distribution using only the limited number of test samples from that distribution at test time. Upon the execution of `Invenio` on the set of observed datasets as shown in Section. 3.1.1, we can then leverage the inferred semantic structure to better adapt to unobserved datasets. As shown in Algorithm 2, we adopt a procedure similar to the meta-test phase (from Algorithm 1) and first compute similarities between the

test distribution (unobserved) and meta models from all observed datasets. We then rank the models based on their relevance to the new dataset, as determined by the η values. The model parameters ϕ_{unobs} are then obtained by performing transfer learning from the most relevant models. We use an ensembling approach which independently fine-tunes S most-relevant models and obtains the final predictions by aggregating the ensemble.

Algorithm 2: Generalize to a New Dataset for the Same Task.

Input: Meta parameters from `Invenio`, θ_k for all observed datasets $\mathcal{D}_{obs}$, Ensemble size S, Unobserved few-shot dataset $\mathcal{D}_{unobs}$

Output: Ensemble of model parameters ϕ for $\mathcal{D}_{unobs}$

for k *in* $1 \cdots |\mathcal{D}_{obs}|$ **do**

  Estimate $\eta_k = \sum \nabla_{\theta_k} \mathcal{L}_k(\theta_k) . \nabla_{\theta_k} \mathcal{L}_{unobs}(\theta_k)$;

end

Select the top S tasks from $\mathcal{D}_{obs}$, based on η ;

Perform transfer learning from θ_k for the selected tasks to obtain ensemble $\{\phi^s\}_{s=1 \cdots S}$ for $\mathcal{D}_{unobs}$;

4.1.1 Experiment: Test Time Multi-Source Domain Adaptation

We demonstrate the effectiveness of `Invenio` in generalizing to novel unseen datasets (domains) for the same task. We consider a standard dataset used for domain generalization - PACS database (Li *et al.*, 2017a), which comprises four different domains namely photos(P), art-painting(A), i.e., paintings of objects, cartoons(C) and sketches(S). Each of these domains have the same 7 classes. We choose two domains namely, A and S as the observed domains while P and C as the unobserved domains. To introduce more complex domain shifts, we apply the transformations

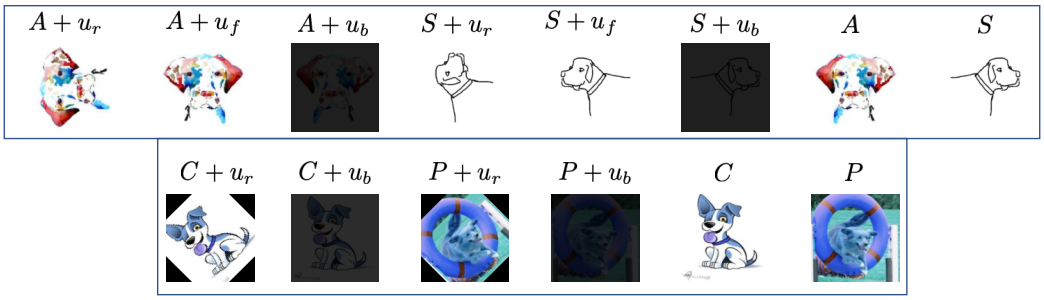


Figure 4.1: We Illustrate an Example Image From the Dog Class for Each of the Domains in the Observed (Top) And Unobserved (Bottom) Sets.

often encountered in real-world image models such as different illuminating/brightness conditions(u_b), flip(u_f) and rotations (u_r). In effect we consider 8 observed domains defined by the set $\{A,\ A + u_b,\ A + u_f\ A + u_r,\ S,\ S + u_b,\ S + u_f,\ S + u_r\}$, while we consider 6 unobserved domains given by $\{C,\ C + u_b,\ C + u_r,\ P,\ P + u_b,\ P + u_r\}$. To capture the complexity of the resulting domain shifts visually, in Figure 4.1, we show a representative sample of the dog class from each of the observed and unobserved domains. The transformation functions u_b reduces the brightness to $\frac{1}{8}$ of its original intensity while u_f and u_r flips the image in vertical and horizontal directions, respectively. We report the performance on the test set for each of the unobserved domains.

Baselines We consider the following baselines: (i) *MDL:* This is a multi-domain learning approach wherein a single model is trained by using the training data from all the observed domains. This common model is then optionally fine tuned on each of the unobserved domains using their corresponding train and validation data; (ii) *MLDG:* A common model is trained through a meta learning algorithm such as MLDG (Li *et al.*, 2017a). Note, the meta model is fine tuned on each of the unobserved domains using their corresponding train and validation data. For all experiments, we use ResNet-18 (He *et al.*, 2016) pretrained on imagenet (Deng *et al.*, 2009a). Across all experiments, the hyperparameter values α, δ and β were set to 1e-4, 1e-3 and 0.1

	Accuracy						
	$C + u_r$	$C + u_b$	$P + u_r$	$P + u_c$	C	P	Mean Performance
MTL	21.89	38.57	15.21	42.75	37.33	47.43	33.86
MTL + finetuning	53.97	57.68	52.16	57.49	54.52	58.80	55.77
MLDG + finetuning	80.25	86.01	84.55	85.69	86.01	85.87	84.73
Invenio(S=1)	82.59	85.41	87.78	90.48	85.41	92.69	87.40
Invenio(S=2)	**84.30**	**88.10**	**91.14**	**94.01**	**87.71**	**94.19**	**89.91**

Table 4.1: Results of Our Proposed Method, Compared Against Several Alternatives, Evaluated for Test Time Domain Adaptation. We Report Both Task Specific and Average Performance.

respectively.

In Table 4.1, we compare the performance achieved using baselines to that of `Invenio` as described in the Algorithm 2, with $S = 1$ and $S = 2$. The proposed approach provides a strong improvement on the generalization performance with a boost of more than 5% points on average. We attribute this boost in performance to exploiting the inherent semantic similarities between the different datasets.

4.2 Invenio for Constructing Semantic Space of Task Distributions

Similar to creating a semantic space for domains, we also consider a scenario where we modify `Invenio` to learn the semantic relationships between different tasks. We define this problem scenario as follows: Given a set of observed datasets $\{\mathcal{D}_k\}_{k=1}^{K}$, wherein each task contains N_k labeled examples, $\mathcal{D}_k = \{(\mathrm{x}_i^k, \mathrm{y}_i^k)\}_{i=1}^{N_k}$, and can have a different output space $\mathcal{Y}^k$. Since the actual tasks are different across the datasets (e.g. *Dogs* vs *Cats* and *Kingfishers* vs *Pigeons*), the redundancies across tasks are identified using a common set of parameters (i.e. feature extractor) $\theta_{\mathrm{f}}^{\mathrm{k}}$, while we allow task-specific parameters (i.e. classifier) $\theta_{\mathrm{c}}^{\mathrm{k}}$ as well.

In this case, we assume each of the observed tasks $\mathcal{D}_k \in \mathcal{D}_{obs}$ has a separate label space $\mathcal{Y}_k$ and `Invenio` is extended to support this. Inspired by modern transfer learning approaches, we view a model $\mathcal{F}_k$ as a composition of a feature extractor and a task-specific classifier, i.e., $\theta_k = [\theta_k^f; \theta_k^c]$. Note that, our formulation is generic enough to allow different number of output classes in each of the tasks. While the meta-train step is identical to the one in Algorithm 1, there are crucial differences in the meta-test phase. During the computation of the similarity score $\eta_k j$, for a meta-test task $\mathcal{D}_j^{ts}$, we first construct a model $\bar{\theta}_k = [\theta_k^f; \phi_j^c]$. In other words, we emulate a transfer learning scenario where the classifier layer is redesigned to support the task at hand. Since this classifier layer is initialized randomly, we also refine that layer (with θ_k^f fixed) using only $\mathcal{D}_j^{ts}$. We use a similar strategy even while computing the meta-test loss in eqn. (3.19). The modified algorithm can be found in Algorithm 3.

4.2.1 Experiment: Semantic Space of Tasks

Since `Invenio` identifies key relationships between tasks, in terms of the ease of transferring knowledge from a meta-train task to a meta-test task, we believe the η estimates can be used to construct a meaningful semantic space. In this experiment, we follow the setup in *Task2Vec* (Achille *et al.*, 2019a), wherein we consider a suite binary classification tasks (400 in our case) sampled from four different datasets namely CUB (Wah *et al.*, 2011), DeepFashion (Liu *et al.*, 2016), iMaterialist (Guo *et al.*, 2019) and iNaturalist (Van Horn *et al.*, 2018). While the positive class in each task corresponds to a specific image class from one of the datasets, the negative class contains images (randomly chosen) from all datasets - (i) *CUB200* (Wah *et al.*, 2011): We use 12 randomly selected classes from the Caltech-UCSD Birds dataset; (ii) *Deep*

Algorithm 3: `Invenio` On Multiple Observed Tasks.

Input: Set of observed datasets $\mathcal{D}_{obs}$

Output: Meta-parameters for the learning task on all observed datasets $\mathcal{D}_{obs}$

Initialization: Parameters $\theta_k = [\theta_k^f; \theta_k^c]$ for each $\mathcal{F}_k, \forall k \in 1, \cdots, |\mathcal{D}_{obs}|$. Set hyper-parameters α, β, δ ;

Random Split: $\mathcal{D}^{tr}$ and $\mathcal{D}^{ts} \leftarrow \mathcal{D}_{obs}$;

for *iter* ***in*** n_{iter} **do**

  for k *in* $1 \cdots K^{\dagger}$ **do**

    Compute loss $\mathcal{L}_k(\theta_k)$ and gradients $\nabla_{\theta_k}\mathcal{L}_k(\theta_k)$ using $\mathcal{D}_k^{tr}$;

    Update $\hat{\theta}_k = \theta_k - \alpha\nabla_{\theta_k}\mathcal{L}_k(\theta_k)$;

  end

  for k *in* $1 \cdots K^{\dagger}$ **do**

    for j *in* $1 \cdots K^{\ddagger}$ **do**

      For task $\mathcal{D}_j^{ts}$, construct model $\bar{\theta}_k = [\theta_k^f, \phi_j^c]$ and learn the classifier ϕ_j^c using $\mathcal{D}_j^{ts}$;

      Estimate $\eta_{kj} = \sum \nabla_{\theta_k^f}\mathcal{L}_k(\theta_k^f).\nabla_{\theta_k^f}\mathcal{L}_j(\theta_k^f)$;

    end

    Obtain normalized scores $\bar{\eta}$ and compute meta-test loss $\mathcal{G}(\hat{\theta}_k)$ in (3.19) ;

    Update θ_k:

$$\theta_k = \theta_k - \delta\frac{\partial(\mathcal{L}_k(\theta_k) + \beta\mathcal{G}(\theta_k - \alpha\nabla_{\theta_k}\mathcal{L}_k(\theta_k))}{\partial\theta_k} \ ;$$

  end

  New random split: $\mathcal{D}^{tr}$ and $\mathcal{D}^{ts} \leftarrow \mathcal{D}_{obs}$

end

Fashion (Liu *et al.*, 2016): We use 13 randomly selected clothing categories from this benchmark fashion dataset; (iii) *iMaterialist* (Guo *et al.*, 2019): We also chose 33 categories from this large-scale fashion database; (iv) *iNaturalist* (Van Horn *et al.*, 2018): From this large-scale species detection dataset, we randomly sampled 342 categories from broad taxonomical classes such as Mammalia, Reptilia, Aves etc. By design, there is partial overlap in tasks between iNaturalist and CUB datasets, and similarly between iMaterialist and DeepFashion, while simultaneously there is a clear disconnect between fashion and species datasets. Such a design enables us to evaluate our approach and reason about the discovered semantic structure between tasks. Each binary classification problem contains 100 positive samples, while another 100 randomly chosen samples for the negative class. Note that, the architecture and the training hyper-parameters are same as that from the domain experiment.

Upon execution of `Invenio` on this large database of binary classification tasks, we compute the overall task similarity matrix by computing η between every pair of tasks (treating one of them as the train and other as the test). We perform truncated singular value decomposition (SVD) on this similarity matrix to infer an $8-$dimensional embedding space which characterizes the inferred semantic structure. We also generated $2-D$ embeddings for visualization in Figure 4.2. As it can be clearly seen, even in the 2D embedding, the disparate tasks (bird type recognition vs clothing type detection) are well separated. Furthermore, we generated Table 4.2 which shows the results from doing a nearest neighbor search in the semantic space for different query tasks. We find that our approach produces highly meaningful relationships between tasks and hence we expect such a semantic space to be beneficial to understand which models can be re-purposed for which tasks.

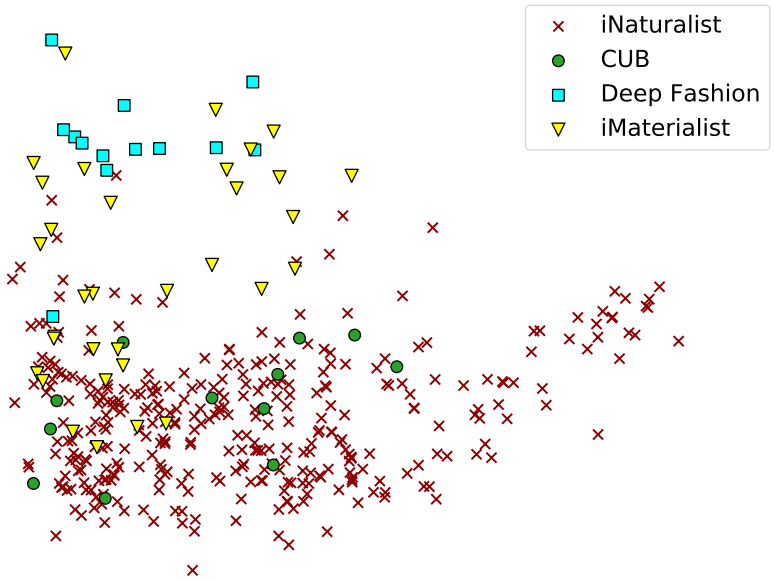


Figure 4.2: 2D Visualization of the Semantic Space Obtained Using `Invenio` On 400 Diverse Binary Classification Tasks.

Query	**Most Semantically Relevant Tasks**
Jeans	Shorts, Pants, Jackets, Dress, Kimonos
Kingfisher	Oreothlypis, Cuckoo, Hetaerina, Spizella, Zonotrichia
Crow	Cormorant, Hyles, Calypte, Blackbird, Latrodectus
Dress	Cocktail dresses, Party dresses, Jumpsuits, Jackets, Prom_Dresses
Rabdotus	Cipangopaludina, Oxidus gracilis, Leccinum scabrum, Fistularia commersonii

Table 4.2: Examples of Nearest Neighbors for Query Tasks in the Semantic Space Inferred Using `Invenio`.

4.3 Generalization to New Tasks

In this application, we show that meta parameters learnt using observed tasks can be used to transfer learn unobserved tasks at test time similar to domain adaptation paradigm. A crucial benefit of leveraging structure while inferring the task-specific models in the meta optimization process is that we obtain distinct sets of meta parameters which are appropriate for generalizing to tasks of different complexity. The

transfer learning protocol to unobserved tasks in this case is similar to Algorithm 2 with the additional step of adapting the model $\bar{\theta}_k$ prior to computing η. The proposed transfer learning protocol is given in detail in Algorithm 4.

Algorithm 4: Generalize to New Tasks.

Input: Meta parameters from `Invenio`, $\theta_k = [\theta_k^f, \phi_k^c]$ for all observed datasets $\mathcal{D}_{obs}$, Ensemble size S, Unobserved dataset $\mathcal{D}_{unobs}$

Output: Ensemble of model parameters ϕ for $\mathcal{D}_{unobs}$

for k *in* $1 \cdots |\mathcal{D}_{obs}|$ **do**

For task $\mathcal{D}_j \in \mathcal{D}_{unobs}$, construct model $\theta_k = [\theta_k^f, \phi_j^c]$ and learn the classifier ϕ_j^c using $\mathcal{D}_j$;

Estimate $\eta_{kj} = \sum \nabla_{\theta_k^f} \mathcal{L}_k(\theta_k^f) . \nabla_{\theta_k^f} \mathcal{L}_j(\theta_k^f)$;

end

Select the top S tasks from $\mathcal{D}_{obs}$, based on η ;

Perform transfer learning from θ_k^f and learn a new classifier ϕ_j^c for the selected task to obtain ensemble $\{\phi^s\}_{s=1 \cdots S}$ for $\mathcal{D}_j \in \mathcal{D}_{unobs}$;

4.3.1 Experiment: Generalization to New Tasks

We consider the benchmark MiniImagenet dataset (Vinyals *et al.*, 2016) to create 10 observed and 10 unobserved tasks, wherein each task is a $5-$way $30-$shot classification task. Note that, the classes in the unobserved/new tasks are different from those in the observed set, while we allowed overlaps between classes in the observed set. We report generalization performance on the test set for each of the held-out unobserved tasks. We consider the following baselines: (i) *MTL:* This is a multi-task learning approach wherein a single model is trained by using all observed tasks. This common model is then optionally fine tuned on each of the unobserved tasks using their corresponding

	Accuracy										
	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Task 10	Mean
MTL	23.5	22	18	18	16.5	25.5	20	25.5	12	23.5	18.75
MTL+finetuning(last layer)	26.5	27	35	28.5	43.5	31	31	39.5	27.5	41.5	33.1
MAML+finetuning	32.5	51.5	44	50.5	59.5	39	47.5	48	52.5	56.5	48.15
ERM	**61.5**	55	69	58.5	66	50.5	65	61	54.5	**79.5**	62.05
Invenio (S=1)	56	62	67	64	66.5	57.5	61.5	64	58	76	63.25
Invenio (S=2)	58	**67**	**70**	**64.5**	**74.5**	**60.5**	**69.5**	**65**	**61.5**	76	**66.65**

Table 4.3: Results of Our Proposed Method, Compared Against Several Baselines, Evaluated for Task Generalization on the Miniimagenet Dataset.

data. Note, that we use task specific classifier layers in this paradigm; (ii) *MAML:* A common model is trained through a meta learning algorithm such as MAML (Finn *et al.*, 2017a). Note, the meta model is fine tuned on each of the unobserved tasks using their corresponding train and validation data; (iii) *ERM:* A single model is trained using single objective function by merging all observed tasks. For all experiments, we use ResNet-18 (He *et al.*, 2016) pretrained on imagenet (Deng *et al.*, 2009a). Similar to domain experiments, here the hyperparameter values α, γ and β were set to 1e-4, 1e-3 and 0.1 respectively. From the results in Table 4.3, we find that the `Invenio` provides significant performance gains in almost all cases, with an average improvement of 180% in comparison to MAML. This clearly evidences the importance of exploiting the structure among tasks.

4.4 Multi-task Learning

In a typical multi-task learning pipeline, the goal is to leverage information learned by one task to aid the training of other tasks (Zhang and Yang, 2021; Ruder, 2017). The intuition of such multi-task learning strategies is to prefer hypothesis which explains more than one task by introducing the inductive bias, however, choosing

which tasks should be trained jointly is still a cumbersome process. While there are existing multi-task learning strategies which perform joint task modeling (Kendall *et al.*, 2018; Yu *et al.*, 2020), the primary drawback is that the model maybe unable to learn a good shared representation due to semantic discrepancies between multiple observed tasks. Using `Invenio`, we are able to model these sematic discrepancies and hence, learn a good shared feature representation.

We define the problem scenario as follows: Given a set of observed tasks $\{\mathcal{D}_k\}_{k=1}^{K}$, wherein each task contains N_k labeled examples, $\mathcal{D}_k = \{(\mathrm{x}_i^k, \mathrm{y}_i^k)\}_{i=1}^{N_k}$, and can have a different output space $\mathcal{Y}^k$. The redundancies across tasks are identified using a shared set of parameters (i.e. feature extractor) θ_{f}, while we also allow task-specific parameters (i.e. classifier) $\theta_{\mathrm{k}}^{\mathrm{c}}$. We further introduce task-specific parameters $\theta_{\mathrm{k}}^{\mathrm{l}}$ which are primarily used for semantic space computation in meta-test step. The primary advantage of having shared feature extractor and separate parameters for semantic space computation is in terms of scalability. As the number of observed tasks increase, we are still able to compute the semantic structure with less number of task specific parameters $\theta_{\mathrm{k}}^{\mathrm{l}}$.

For this scenario, `Invenio` implementation is shown in Algorithm 5. Inspired by modern transfer learning approaches, we view a model $\mathcal{F}_k$ as a composition of a shared feature extractor θ^f and task-specific parameters θ_k. In our paradigm, we further characterize task-specific parameters as a combination of task-specific convolutional layer and a task-specific classifier, i.e., $\theta_k = [\theta_k^l; \theta_k^c]$. In this paradigm, we split the each task $\mathcal{D}_k$ into training $\mathcal{D}_k^{tr}$, validation set $\mathcal{D}_k^v$ and test set $\mathcal{D}_k^{ts}$.

In the meta-train step, similar to Algorithm 1, we update task-specific parameters $\theta_k = [\theta_k^l; \theta_k^c]$ and save task-specific gradients for shared feature extractor θ^f. In meta validation phase, during the computation of the similarity score η_{kj}, for a meta-validation set $\mathcal{D}_j^v$, we use a model $\bar{\theta}_k = [\theta_f; \theta_k^l; \theta_j^c]$ where θ_f is original shared

feature extractor, θ_k^l is the updated train task specific parameters and θ_j^c is the updated validation task's classifier. We use a similar strategy while computing the meta-validation loss as in eqn. (3.19), however, the update is applied on task-specific convolutional parameters i.e θ_k^l. Once all the task-specific parameters are updated, we update the shared feature extractor using the gradients from meta-train step. Note that, our formulation is generic enough to allow different number of output classes in each of the tasks. The modified algorithm can be found in Algorithm 5.

Once `Invenio` training is concluded, we obtain meta-parameters for each dataset $\mathcal{D}_k$. To show the performance on test set of the observed task, we use the paradigm as shown in Algorithm 6. For an observed task's test set denoted by $\mathcal{D}_j^{ts}, \forall j \in 1, \cdots, |\mathcal{D}|$, we estimate η_{kj} and select top S tasks from $\mathcal{D}$, based on η. Finally, we perform transfer learning from shared feature extractor θ^f, ensemble $\{(\theta_k^l)^s\}_{s=1\cdots S}$, and θ_j^c for test set. From the ensemble, one set of parameters will belong to the observed task itself which were obtained using the training process. The remaining $S-1$ set of parameters are from the most related tasks and hence aid in improving the task performance.

4.4.1 Experiment: Multi task Learning

We consider the benchmark CelebA dataset (Liu *et al.*, 2018) to create 9 observed binary classification tasks. Note that, each task is created by selecting an attribute from the 40 possible attributes from the CelebA dataset. Similar to other multi-task learning baselines such as (Fifty *et al.*, 2021), we choose `5 o Clock Shadow`, `Black Hair`, `Blond Hair`, `Brown Hair`, `Goatee`, `Mustache`, `No Beard`, `Wearing Hat` as our task attributes for fair comparison. We utilize the provided splits for training, validation and test sets. We report the task specific and total loss performance on the held-out test set for each of the observed tasks.

We consider the following baselines: *MTL:* This is a multi-task learning approach

wherein a single model is trained by using all observed tasks. (ii) *MAML:* A common model is trained through a meta learning algorithm such as MAML (Finn *et al.*, 2017a). (iii) *TAG:* In this method, authors use inter-task affinity by training all tasks together in a single multi-task network and quantifying the effect to which one task's gradient update would affect another task's loss (Fifty *et al.*, 2021). (iv) *FLUTE:* In this method, a universal template is created which is composed of shared model parameters along with task-specific batch normalization layers. The shared model is trained using all observed tasks, whereas scaled and shifted task-specific batch norm layers are trained using corresponding task. For all experiments, we use ResNet-18 (He *et al.*, 2016) pretrained on imagenet (Deng *et al.*, 2009a). Similar to domain experiments, here the hyperparameter values α, γ and β were set to 1e-4, 1e-3 and 0.1, respectively. From the results in Table 4.4, we find that the `Invenio` provides significant performance gains in all cases.

Algorithm 5: `Invenio` For Multitask Learning.

Input: Set of datasets $\mathcal{D}$

Output: Meta-parameters for the test set of all datasets $\mathcal{D}$

Initialization: Parameters θ^f shared; $\theta_k = [\theta_k^l; \theta_k^c]$ for each $\mathcal{F}_k, \forall k \in 1, \cdots, |\mathcal{D}|$. Set hyper-parameters α, β, δ ;

Split each dataset to train and validation set: $\mathcal{D}^{tr}$ and $\mathcal{D}^v \leftarrow \mathcal{D}_k$;

for *iter* ***in*** n_{iter} **do**

  for k *in* $1 \cdots K$ **do**

    Compute loss $\mathcal{L}_k(\theta^f; \theta_k)$ and gradients $\nabla_{\theta^f;\theta_k}\mathcal{L}_k(\theta^f; \theta_k)$ using $\mathcal{D}_k^{tr}$;

    Update $\hat{\theta}_k = \theta_k - \alpha\nabla_{\theta_k}\mathcal{L}_k(\theta^f; \theta_k)$;

  end

  for k *in* $1 \cdots K$ **do**

    for j *in* $1 \cdots K$ **do**

      For task $\mathcal{D}_j^v$, ;

      Estimate $\eta_{kj} = \sum \nabla_{\theta_k^l}\mathcal{L}_k(\theta^f; \theta_k^l, \theta_k^c).\nabla_{\theta_k^l}\mathcal{L}_j(\theta^f; \theta_k^l, \theta_j^c)$;

    end

    Obtain normalized scores $\bar{\eta}$ and compute meta-validation loss $\mathcal{G}(\hat{\theta}_k^l)$;

    Update θ_k^l:

$$\theta_k^l = \theta_k^l - \delta\frac{\partial(\mathcal{L}_k(\theta^f; \theta_k) + \beta\mathcal{G}(\theta_k^l - \alpha\nabla_{\theta_k^l}\mathcal{L}_k(\theta^f; \theta_k))}{\partial\theta_k^l} ;$$

  end

  Update $\hat{\theta}^f = \theta^f - \alpha\nabla_{\theta^f}\mathcal{L}_k(\theta^f; \theta_k)$;

end

	Loss									
	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Total Loss
MTL	6.91	11.23	4.31	12.51	2.57	3.01	4.59	4.81	**0.64**	50.57
MAML	6.78	11	4.13	12.12	2.87	3.1	4.32	4.54	0.84	49.7
TAG	6.39	11.08	4.07	12.20	2.63	2.99	4.79	4.62	0.722	49.49
FLUTE	6.45	10.23	3.67	13.46	1.94	2.98	3.99	4.23	1.02	47.97
Invenio (S=2)	**1.83**	**2.27**	**2.64**	**2.38**	**0.75**	**0.78**	**0.85**	**2.12**	0.8	**14.42**

Table 4.4: Results of Our Proposed Method, Compared Against Several Baselines, Evaluated Multi-Task Learning on the CelebA Dataset.

Algorithm 6: Generalize to Test Data.

Input: Meta parameters from `Invenio`, $\theta^f; \theta_k = [\theta_k^l, \theta_k^c]$ for all datasets $\mathcal{D}$,

Ensemble size S, Test set from one of the dataset $\mathcal{D}_j^{ts}$

Output: Ensemble of model parameters for $\mathcal{D}_j^{ts}$

for k *in* $1 \cdots |\mathcal{D}|$ **do**

For task $\mathcal{D}_j^{ts}$, Estimate $\eta_{kj} = \sum \nabla_{\theta_k^l} \mathcal{L}_k(\theta^f; \theta_k^l, \theta_k^c).\nabla_{\theta_k^l} \mathcal{L}_j(\theta^f; \theta_k^l, \theta_j^c)$;

end

Select the top S tasks from $\mathcal{D}$, based on η ;

Perform transfer learning from θ^f, ensemble $\{(\theta_k^l)^s\}_{s=1\cdots S}$, θ_j^c for $\mathcal{D}_j^{ts}$;

4.5 Summary

In this chapter, we showcase how `Invenio` is a generic optimization paradigm and with small modifications can be re-purposed for several different computer vision applications. We utilize `Invenio` for applications such as test time domain adaptation, constructing semantic space of tasks, generalizing to new tasks and multi-task learning. We show how `Invenio` exploits the task and domain priors and achieves the performance improvements on several benchmarks. In next chapter, we explore data capture mechanism as a prior and how that prior can be effectively utilized for energy

efficient object tracking.

Chapter 5

DATA SELECTION MECHANISM AS A PRIOR

5.1 Introduction

In this chapter, we introduce the data selection mechanism as a prior. Similar to model, task and domain priors, we empirically identify whether data selection mechanism has a significant effect on the final task performance. Primarily we investigate data capture in the context of video object tracking. Object tracking is one of the most ubiquitous applications for computer vision with a rich history in robotics, surveillance, and autonomous vehicles. In recent years, deep learning-based neural networks have accelerated progress in object tracking to state-of-the-art performance. However along with advanced architectural design, the vastly available high resolution video data is key for significant improvements in the tracking performance. Note that, despite the huge success, recent research has turned to embedded or energy-efficient object tracking where system constraints on power and latency are critical for extended deployment in the wild. In this work, we develop a mechanism termed as adaptive subsampling which systematically captures relevant video data such that it is enough to maintain object tracking task performance while also achieving energy efficiency by reading less redundant data during data capture. We achieve adaptive subsampling by two specific strategies 1) Adaptive subsampling using frame intensity 2) Adaptive Subsampling using policy gradient method. The two strategies are discussed in further detail in following sections.

5.2 Energy Efficient Video Object Tracking

5.2.1 Introduction

Imaging sensors are one of the major sources of energy expenditure for embedded vision platforms, particularly for continuous object tracking. The analog readout circuitry of image sensors can consume 50-70% of the total energy in most modern mobile system designs (LiKamWa *et al.*, 2016; Buckler *et al.*, 2017). Furthermore, always-on vision cameras which duty cycle their sensing to save battery life are used in several applications (Naderiparizi *et al.*, 2017; Sadasivam *et al.*, 2017; Yi *et al.*, 2020). Finally, surveillance cameras employ uninterrupted data capture and hence need to be energy efficient for prolonging battery life.

To address inefficient energy expenditure in the processing of real-time video data, one important mechanism for image sensors is adaptive subsampling. Adaptive subsampling is the selective readout of regions of interest (ROIs) in sequential frame capture while turning off other pixels in the image. Cameras in the market that can read out selective ROIs yield the resulting benefits: reduced image quantization, faster bandwidth and improved energy efficiency. Energy per pixel and spatiotemporal resolution of the streaming images are inversely proportional, i.e. lower frame rates and image resolutions consume less energy (LiKamWa *et al.*, 2013).

ROIs are particularly useful for object tracking, where only a small ROI around a moving object is necessary for tasks including surveillance and autonomous driving. The problem of adaptive subsampling is to determine this correct ROI. In this work, we consider the issue of performing robust object tracking for an image sensor performing adaptive ROI subsampling. This means that frames of the image sensor will be subsampled to only a small ROI rather than a full visual frame while sensing, and the algorithm needs to perform tracking with this limited information. To this end, we

propose two adaptive subsampling strategies achieved using frame intensity as well as using a policy gradient method respectively. Our primary contributions in this area are as follows:

- We develop a frame intensity based adaptive subsampling technique which devoid of a neural network, is able to perform energy efficient object detection at test time.

- We develop a policy gradient method for learning image subsampling patterns which aid in ROI prediction and can be simultaneously integrated with ROI-capable cameras to improve image sensor energy efficiency. We propose a loss function based on target location and image subsampling which captures the dissimilarities between network predictions and corresponding target labels.

- We also show the efficacy of our network in the context of energy optimization by reporting potential energy savings and computational efficiency at test time.

The proposed techniques are evaluated on a variety of datasets and against conventional state-of-the-art object trackers. Note that we are developing it as a proof of concept to show that detection based trackers can be utilized to maintain energy efficiency by tracking with subsampling. We also compare against a number of baseline algorithms developed in (Iqbal *et al.*, 2021) coupled with Kalman filtering to endow them with predictive capability. Our method outperforms both state-of-the-art and baselines in terms of Area under the Receiver Operating characteristic curve (AUC) and Mean Average Precision (mAP) (Bradley, 1997) and achieves significant energy savings owing to the adaptive subsampling component.

5.2.2 Related Work

Object Detection and Tracking There has been extensive work in the field of object detection and tracking in recent years. Due to the advent of deep learning, several deep learning trackers have been proposed which give significant performance improvements (Fan and Ling, 2017; Nam and Han, 2016; Song *et al.*, 2018b; Wang *et al.*, 2015) compared to correlation filter based tracking (Bolme *et al.*, 2010; Danelljan *et al.*, 2015; Henriques *et al.*, 2014; Danelljan *et al.*, 2016; Valmadre *et al.*, 2017) and Kalman filter based tracking (Li *et al.*, 2010; Black *et al.*, 2002; Marcenaro *et al.*, 2002; Kim and Jeon, 2014). However, a major drawback of these methods is in terms of computational complexity leading to issues when they are applied at very small frames per second (FPS). This has led to a series of deep Siamese based trackers and its variants (Tao *et al.*, 2016; He *et al.*, 2018; Wang *et al.*, 2019a,b; Guo *et al.*, 2017; Bertinetto *et al.*, 2016; Zhang and Peng, 2019; Li *et al.*, 2019, 2018; Zhu *et al.*, 2018).

Several recent works also utilize regression based target tracking methods (Bertinetto *et al.*, 2016; Held *et al.*, 2016). However, some of these methods are unable to utilize long-term temporal information efficiently. Recently, there has been a surge in utilization of recurrent neural networks for target tracking (Kahou *et al.*, 2017; Ning *et al.*, 2017; Gan *et al.*, 2015). In (Ning *et al.*, 2017), authors utilize the regression capability of Long Short Term Memory (LSTM) to predict the target location. A similar model was proposed in (Gan *et al.*, 2015), wherein authors utilized a recurrent neural network to predict top-left and bottom-right corners of a bounding box. However this method utilizes localization error as the final cost function unlike (Kahou *et al.*, 2017), wherein a classification error averaged over all the frames is used. The object tracking methods most close to ours include (Zhang *et al.*, 2017; Sun *et al.*, 2020) and (Mnih *et al.*, 2014) in which authors utilize the CNNs to extract image information and the RNN

and RL techniques are used to perform visual target tracking. In (Li *et al.*, 2021), the authors introduce a new benchmark and baseline for predictive visual tracking that accounts for both performance and latency. While several authors have also proposed RL algorithm based video tracking solutions (Choi *et al.*, 2017; Supancic III and Ramanan, 2017; Mnih *et al.*, 2016; Yun *et al.*, 2017; Huang *et al.*, 2017a; Chen *et al.*, 2018; Dunnhofer *et al.*, 2019), none of them to the best of our knowledge are developed and analysed keeping in mind the energy efficiency needed for image sensing in embedded systems.

The recent boom in the autonomous and mobile platforms has triggered a need for a major design choice to make the object detection and tracking pipeline more energy efficient. In (Casares and Velipasalar, 2011), the authors present an adaptive methodology wherein the embedded camera state duration is determined based on the speed of the tracked object. However, this method is not able to work well with strongly shadowed videos. In (Apicharttrisorn *et al.*, 2019), the authors propose a software framework titled MARLIN which enables content driven real time tracking by switching between deep learning and light-weight techniques. But the method fails in instances when the neural network based tracker is not triggered in time.

ROI Adaptive Subsampling In the majority of embedded and mobile platforms, image sensing is one of the major sources of energy expenditure which leads to inefficient battery usage. In (LiKamWa *et al.*, 2013), authors show how image sensor energy expenditure has an inversely proportional response to changes in pixel resolution and frame rate. Several algorithms for adaptive spatial subsampling have been proposed in the context of image and video compression (Lin and Dong, 2006; Dong and Ye, 2013; Belfor *et al.*, 1994). However, we are concerned with saving energy while performing object detection on real-time video, and thus cannot rely on video

compression algorithms which rely on the full video being captured first. To address this problem, we develop a content driven adaptive subsampling strategy wherein the algorithm learns to read specific regions-of-interest (ROIs) to save energy using spatial subsampling. In a similar vein, the authors have developed an adaptive subsampling strategy in (Iqbal *et al.*, 2020) which utilizes a YOLO network for object detection and a Kalman filter for ROI prediction. In our proposed intensity based adaptive subsampling method (Katoch *et al.*, 2019a), we propose an algorithm which detects ROIs by employing the objectness feature. However, the reference frame subsampling mask is used for ROI detection in consecutive frames. This fails to account for changes in the appearance of the object, which leads to erroneous tracking at least until the next reference frame comes in. Keeping this erroneous tracking into consideration, we have investigated and implemented another adaptive subsampling method that has greater predictive power. We employ an LSTM network as our agent in the proposed policy gradient method to make our future location predictions. The LSTM network benefits from past temporal information encoded in its hidden units and is able to anticipate future trajectories with a great degree of accuracy. In the following sections, we elaborate on both intensity based adaptive subsampling and policy gradient based adaptive subsampling in detail.

5.2.3 Proposed Approach: Frame Intensity based Adaptive Subsampling

In this method, to enable energy-efficient video object detection, we propose an adaptive algorithm to subsample video frames that uses a metric for objectness (Alexe *et al.*, 2010, 2012) and intensity-based segmentation. This algorithm utilizes semantic information from a previous key frame in the video to determine subsampling patterns for future frames. We show that this adaptive algorithm achieves better trade-offs in energy savings to object detection accuracy as compared to naive subsampling

methods of uniform and random sampling.

Algorithm Description In this section, we present our frame intensity based adaptive subsampling algorithm for video object detection and tracking. We require that our algorithm operates at run-time by determining the future subsampling patterns based only on prior frames (i.e. causal system) so that it can work on incoming video frames. This algorithm is simple conceptually in nature as we wanted to reduce the amount of overhead computation it takes to allow for adaptive sampling. However, we show that this method achieves only a slight degradation in object detection and tracking performance while saving energy. The whole algorithm is summarized in Figure 5.1.

One constraint we placed on our method is that it had to work on embedded platforms that have limited resources. This included the assumption that there is no GPU on the platform, and thus no way to retrain the object detection neural network to adapt to the subsampling pattern. Of course doing so could yield even better object detection accuracy for the same energy savings. The advantage of this method is that it is immediately deployable to existing systems such as UAVs and robotic platforms and requires no training or GPUs on-board.

Objectness as semantic information The first key question we consider is how to extract semantic information from previous frame(s). While there are several techniques that could be used as generic visual features including CNN features (Sharif Razavian *et al.*, 2014), we utilize an algorithm that trains a measure for objectness for a given image (Alexe *et al.*, 2010, 2012). This makes our algorithm highly tuned for object detection, and does not require an additional neural network to be stored on the embedded device to extract visual features. This algorithm quantifies

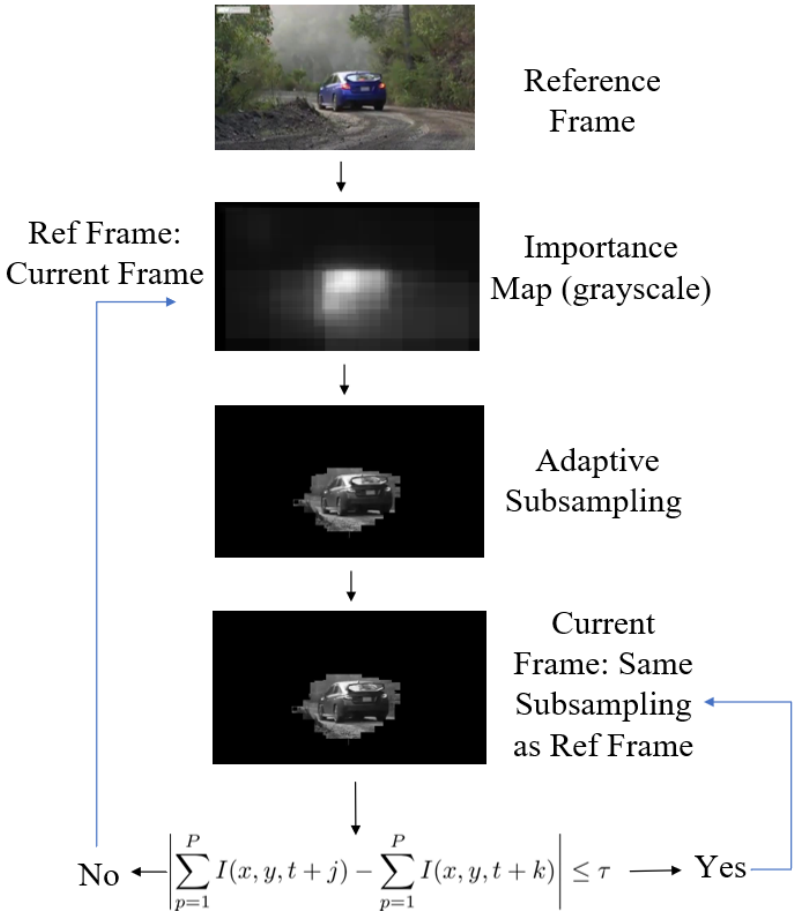


Figure 5.1: Flowchart Explaining the Intensity Based Adaptive Video Subsampling Algorithm.

how likely it is for an image window to cover an object of any class. It does so by considering four image cues: multi-scale saliency, color contrast, edge density and straddleness (Alexe *et al.*, 2010, 2012). Combining different image windows, the algorithm produces an objectness map that can be seen in Figure 5.2. In this figure, we show how the objectness map still can identify primary objects even when operating on different types of subsampled imagery. We will utilize these objectness maps to help determine our spatial subsampling in the video.

Adaptive Subsampling Algorithm We now present our adaptive algorithm, which couples this objectness map with intensity changes in the video to help determine

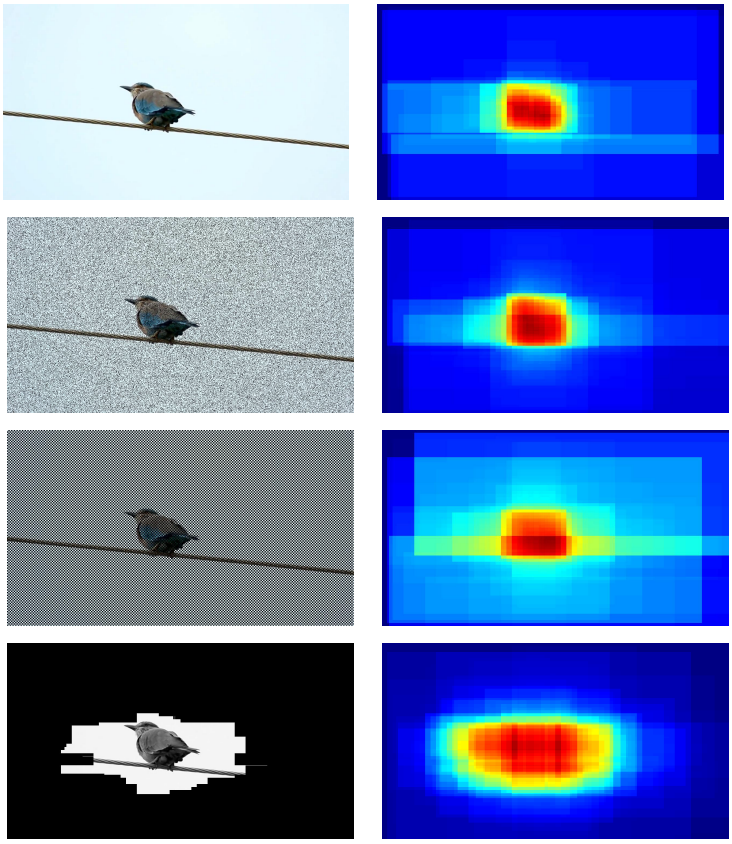


Figure 5.2: The First Row Shows the Original Image and Its Resulting Objectness Map. The Next Three Rows Show the Same Process for Three Different Forms of Image Subsampling on the Original Image: Random Pixelation, Checkerboard Mask, and Adaptive Video Sampling.

a spatial sampling pattern. Let $I(x, y, t)$ represent a video where (x, y) represents the locations of the pixels and t represents the frame index in time. Let N_1 and N_2 represent number of rows and columns in a given frame, respectively. Consequently, the number of pixels in a given frame is given by $P = N_1 N_2$ for a gray-scale image. Let $\mathbf{M}_i$ for $1 < i < T$ represent the objectness-maps as described above, and T represents total number of frames in the video.

The algorithm begins by considering the importance map of the first frame (reference frame) $\mathbf{M}_1$. We calculate a histogram of the importance map, and based on this histogram and an empirically-chosen threshold, we convert the gray-scale objectness map to a binary mask B_1. This threshold is called the objectness threshold, and is

determined either by empirically-chosen values (γ) or by Otsu's method (δ) (Otsu, 1979).

The object blobs in the binary mask are labelled based on their neighboring pixel connections. Once these object blobs in the binary mask are identified, we compute the area of these blobs and only select objects with area greater than a threshold of 2000 pixels to obtain an updated binary mask B_1^u. This binary mask is used for our subsampling for the next consecutive frame.

In other words, the updated binary image is the final mask which is used to turn off pixels in the reference frame. We utilize this mask to subsample the consecutive frames in the video. However, the underlying assumption is that the objects in the scene do not move significantly, so that the mask is still relevant in the subsampling. To check the continued validity of this assumption, we calculate the absolute mean intensity difference between the reference frame and the current subsampled frame, as shown in (5.1):

$$\left| \sum_{(x,y)} I(x, y, t + j) - \sum_{(x,y)} I(x, y, t + k) \right| \leq \tau, \tag{5.1}$$

where $I(x, y, t + j)$ represents reference frame and $I(x, y, t + k)$ represents the current frame. Note that the choice of the Frame Intensity threshold τ is critical for determining whether to update the reference frame and whether the binary mask may overlap only partially with objects in the reference image. A smaller threshold means less energy-savings as more reference frames need to be fully sampled, but the resulting subsampling will more accurately track object motion.

To further validate this object motion assumption, we use another constraint obtained using optical flow between two frames. We use Lucas-Kanade optical flow (Lucas and Kanade, 1981), and if the mean magnitude of the optical flow is less than a fixed threshold ϕ, we use the same subsampling binary mask as the

previous frame. If the constraint is not satisfied, we capture a new reference frame. In Section 5.2.4, we compare the performance using the frame intensity threshold versus the optical flow magnitude threshold.

5.2.4 Experiments

Dataset For the video subsampling algorithm, we use the ILSVRC2015 Image Vid Dataset (Deng *et al.*, 2009b) which has 555 video snippets with 30 classes. For our experiments, we consider videos of 6 classes namely, Bird, WaterCraft, Car, Dog, Horse and Train. We performed object detection using an implementation of Faster RCNN, an object classification algorithm (Yang *et al.*, 2017). The accepted metric of object detection, mean Average Precision (mAP), per classification is obtained based on the bounding boxes from the video frames.

We compare four types of subsampling: (1) random subsampling where each pixel has a probability α of being turned off, (2) our adaptive sampling algorithm using Otsu's method for objectness threshold (δ) and values of $0.1, 0.3, 0.5$ for the frame intensity threshold (τ), (3) adaptive subsampling algorithm with Otsu's method for objectness threshold (δ) and an optical flow magnitude threshold (ϕ) with values $0.0015, 0.005, 0.015$, and (4) Adaptive subsampling with the tuned parameters of 0.15 for the objectness threshold (γ) and 0.1 for the frame intensity threshold. These parameters were tuned on separate videos from the dataset not included in the test set we consider.

Energy modeling For our energy modeling, we assume that the proportion of pixels that are turned off are proportional to the savings in readout energy (LiKamWa *et al.*, 2013). As described in Section 3, τ (i.e. the frame intensity threshold) is one of the most important parameters to control the energy savings while keeping the accuracy of

object detection at almost the same level. If the optimization constraint is too strong (i.e τ is really low), it will lead to subsampling calculation of every consecutive frame which will result in high computation time. It will make the algorithm inefficient for use in camera sensors. However, if this threshold τ is very big, it can lead to conditions where the subsampling strategy neglects the changes due to object motion. The choice of ϕ (i.e. Flow Magnitude threshold) can be justified similarly.

5.2.5 Performance Evaluation

Qualitative Results In Fig. 5.3, we show some visuals of detected objects from adaptive subsampling strategy. For the shown result, a frame is chosen from the Car video and bounding box generated on each subsampled frame is shown. It is evident that even after turning off a large number of pixels, Faster RCNN is able to detect the object in most cases. These benefits coupled with energy savings make it a decent approach for video subsampling.

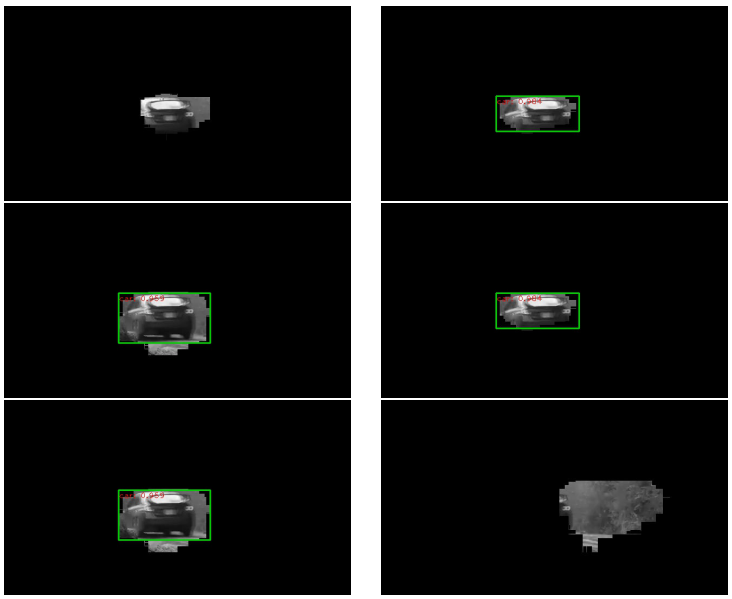


Figure 5.3: The First Column Shows the Object Detection for Three Different Frame Intensity Thresholds ($\tau = 0.1, 0.3$ and 0.5). The Next Column Shows the Same Process for Three Different Optical Flow Magnitude Thresholds ($\phi = (1.5, 5.0, 15.0) \times 10^{-3}$).

Subsampling Strategies	**Fully Sampled**	**Random Subsampling** (α)			**Adaptive Subsampling** ($\delta + \tau$)			**Adaptive Subsampling** (ϕ ($\mathbf{10^{-3}}$))			**Adaptive Subsampling** ($\gamma + \tau$)
		0.15	**0.25**	**0.35**	**0.1**	**0.3**	**0.5**	**1.5**	**5.0**	**15.0**	**0.15 + 0.1**
mAP	**55.5**	15.4	5.9	0.9	40.1	37	38	41.8	41.7	28.6	**50.1**

Table 5.1: mAP Scores for Different Subsampling Strategies.

Quantitative Results To test whether the proposed subsampling strategy achieves the desired energy savings along with the computer vision task accuracy, we show the results of mean Average Precision (mAP) scores of fully sampled, randomly subsampled and adaptive subsampled videos presented in Table. 5.1. It is evident that random subsampling results in the worst mAP scores compared to adaptive subsampling strategy. As mentioned in Section. 5.2.3, in adaptive subsampling strategy, a binary mask is used to obtain the subsampled frames. This binary mask is obtained using the objectness threshold (δ) obtained from Otsu's method. As shown in Table. 5.1, the empirical objectness threshold (γ) resulted in better mAP score compared to Otsu's objectness threshold. Among the two thresholding methods i.e. optical flow magnitude and frame intensity, the frame intensity threshold performed slightly better with an empirically-chosen objectness threshold which gives a mAP score of 50.1% which is closest to fully sampled video mAP score of 55.5%.

In Table 5.2, we show the percentage of pixels turned off for each subsampling strategy. Note that the strategy that received the best mAP score (Adaptive Subsampling with objectness threshold (δ) and frame intensity threshold (τ)) saves 18 - 67% of energy. As mentioned in 5.2.2, this adaptive subsampling method doesn't account for changes in the appearance of the object, which leads to erroneous tracking at least until the next reference frame comes in. Consequently, in the next section, we propose the policy gradient based adaptive sampling approach which uses neural

Subsampling Strategies	**Random Subsampling** (α)			**Adaptive Subsampling** ($\delta+\ \tau$)			**Adaptive Subsampling** (ϕ ($\mathbf{10^{-3}}$))			**Adaptive Subsampling** ($\gamma\ +\ \tau$)
	0.15	**0.25**	**0.35**	**0.1**	**0.3**	**0.5**	**1.5**	**5.0**	**15.0**	**0.15 + 0.1**
Bird	14.16	22.75	30.80	87.43	86.64	86.43	92.23	92.24	92.22	54.04
Watercraft	13.26	22.32	31.62	79.80	79.91	80.09	83.15	83.01	88.30	50.04
Dog	17.71	29.39	40.59	11.83	11.87	11.86	68.76	68.76	68.76	18.44
Car	18.13	30.66	42.86	30.42	30.10	30.18	90.44	90.72	88.94	67.87
Horse	21.21	34.96	48.01	25.82	26.26	26.46	75.65	75.65	75.90	38.85
Train	22.24	29.60	37.41	21.05	21.02	21.07	71.19	71.19	71.19	55.97

Table 5.2: Energy Efficiency in Terms of Turned off Pixel Percentage in a Video for Different Subsampling Strategies.

network based method to achieve energy efficient object tracking.

5.2.6 Proposed Approach: Adaptive Subsampling using Policy Gradient Method

Policy gradient based adaptive subsampling method is similar to predictive object tracking (Li *et al.*, 2021) where an object's future position is inferred from previous frames. While there has been an ample amount of research on object tracking, predictive tracking with adaptive subsampling is less studied in the literature. Previous methods such as (Bertinetto *et al.*, 2016; Kahou *et al.*, 2017; Zhang *et al.*, 2017; Mnih *et al.*, 2014) utilize deep neural networks such as RNNs and LSTMs to perform regression-based location prediction for predictive object tracking. However, all these methods rely on fully sensed frames to perform predictive tracking, and do not robustly track objects while images are being subsampled into ROIs.

We show that existing state-of-the-art trackers such as DIMP (Bhat *et al.*, 2019) and ATOM (Danelljan *et al.*, 2019) which adaptively generate the search region (in next frame) by roughly extending the tracking bounding box in current frame, degrade in performance when operating on subsampled images. Thus, we consider

the objective of jointly performing adaptive subsampling of the image while doing predictive object tracking. This allows us to tradeoff between energy savings via adaptive subsampling and object tracking performance. Most existing ROI selection techniques are performed offline and without the image sensor in the loop. In contrast, our method considers a programmable sensor in the loop which is reconfigured by the network-determined ROI, and we design an algorithmic pipeline tailored to such programmable sensors.

While tracking by detection is not a recent methodology, the novelty is that in particular, we propose a neural network architecture with predictive capabilities to enable preemptive ROI sampling based on the estimated object trajectory. We utilize the tiny YOLO network for feature representation (Redmon *et al.*, 2016) followed by an LSTM network (Hochreiter and Schmidhuber, 1997) for the adaptive subsampling prediction and tracking. Novel to our design is the use of reinforcement learning to train the network, specifically the REINFORCE algorithm (Williams, 1992; Mnih *et al.*, 2014), also known as the Monte Carlo Policy Differentiation, to converge to the optimal tracking and subsampling policy.

Algorithm Description The proposed algorithm implements a dual architecture to perform predictive object tracking and subsampling. Subsampling refers to the selection of only the region of interest (ROI) in the image based on the computer vision task. The algorithm exploits a policy gradient (Mnih *et al.*, 2014) strategy with a cost based on target object location and subsampling pattern (explained in Section. 5.2.7) to guide the model training. The final objective is to obtain a trained network with predictive ROI and subsampling mask capabilities.

The key idea of **keyframing** is a critical design choice in our subsampling and tracking pipeline. Keyframing implies that visual feature extraction on fully sampled

frames in the video occurs at only specific intervals termed as keyframing interval. The underlying premise is that for the non keyframes, the image sensor samples specific pixels based on the ROI predicted by the network architecture for maximal energy efficiency. Keyframing has been used extensively in video compression algorithms (Le Gall, 1991; Sullivan *et al.*, 2012) to reduce image storage. In Section 5.2.8 we present the tracking performance of our architecture for different keyframing intervals.

Network Architecture The dual architecture is composed of (1) a pre-trained Tiny YOLO network to extract feature representations from fully sampled and subsampled frames (Redmon *et al.*, 2016), and (2) an LSTM layer for ROI and subsampling prediction. This dual architecture is inspired from (Zhang *et al.*, 2017) wherein the authors have an observation network to obtain feature information followed by the recurrent network used for location regression. However, to the best of our knowledge, ours is the first work which focuses on developing subsampling masks using regression to perform object detection. The two kinds of subsampling masks we use for guiding the training include 1. ROI based subsampling mask and 2. subsampling mask created by using static grids of 7×7 on a video frame and turning them on or off depending on whether a portion of ground truth object lies in the grid or not (coarse-grained ROI).

In Figure 5.4, we show the network architecture and inference pipeline for a trained network. For the incoming frames, visual feature extraction is conducted using Tiny YOLO, a state-of-the-art real-time object detector which has been used extensively for tracking applications. Tiny YOLO was chosen based on our goal of implementing the proposed pipeline on hardware in future. We will use Tiny YOLO/YOLO interchangeably throughout the section. The extracted features are regressed using an LSTM to predict the bounding box and a coarse-grained ROI

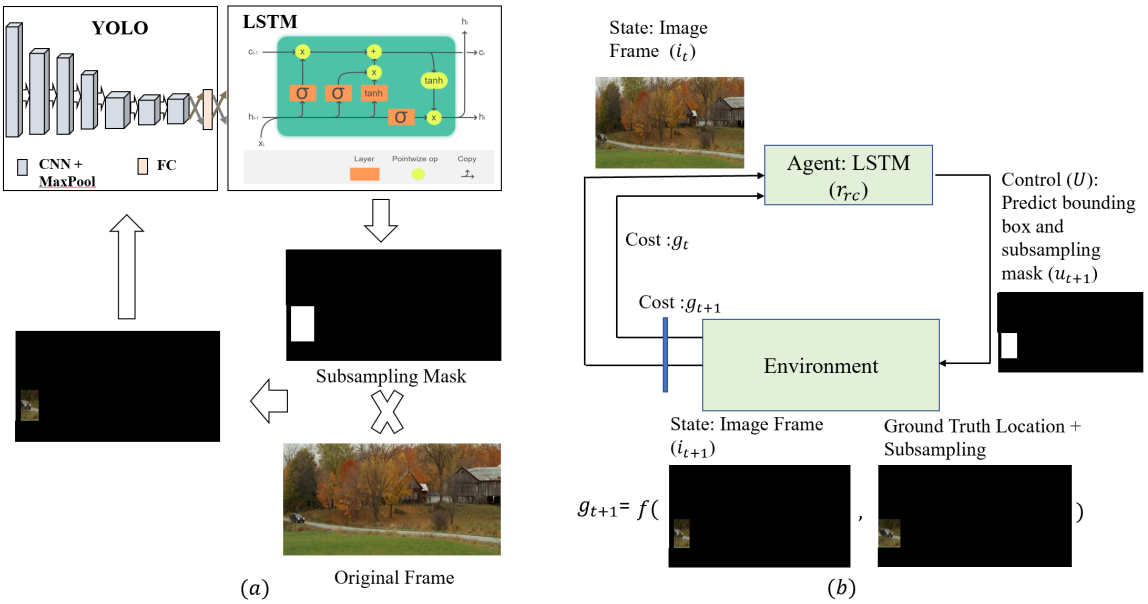


Figure 5.4: A Pretrained Tiny YOLO Extracts Feature Maps From the Images Which Is Then Fed to the LSTM. The LSTM Aims to Learn the Optimal Sensor Mask Generation Strategy Based on Joint Bounding Box and Coarse-Grained Subsampling Pattern Prediction and Uses the Mask to Obtain Subsampled Frame.

(subsampling matrix) for the next frame. At the next time step, a non-keyframe is subsampled according to the location information/ coarse-grained ROI predicted by the LSTM leading to pixels outside of the ROI getting switched off. The tiny YOLO extracts features from this subsampled image which are then passed through the LSTM along with the updated hidden state in which past bounding box and coarse ROI information remains embedded. The prediction phase persists until the next keyframe comes in, at which point the LSTM network once again gets access to the fully sampled image features.

Network Training via Reinforcement Learning The proposed architecture leverages the YOLO based deep learning pipeline for extracting improved quality visual features. Furthermore, it employs state-of-the-art LSTM model to predict the object location. However since our application at the inference requires sequential subsampled

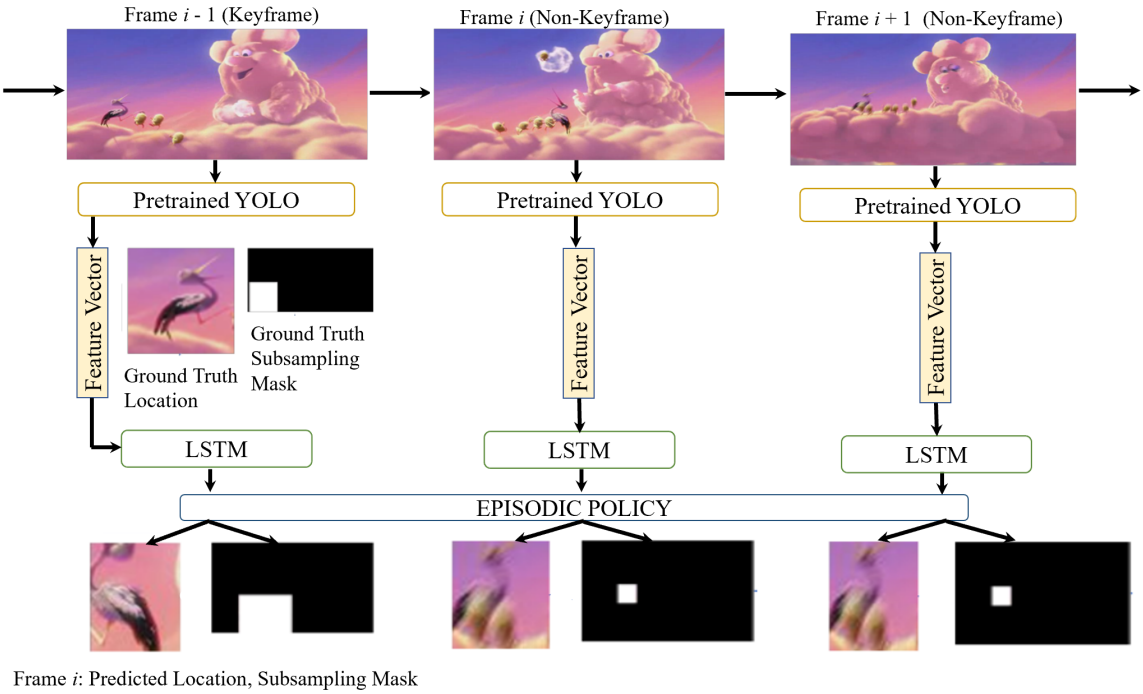


Figure 5.5: At the Training Time, Tiny YOLO Extracts Feature Vector From Incoming Image Frames. Ground Truth Object Location and Coarse Subsampling Mask Corresponding to Keyframe Are Fed Explicity to An LSTM. LSTM Operates on These Inputs and Previous Hidden States and Outputs the Hidden State for the Consecutive Time Step. ROI and Subsampling Predictions Are Extracted From This Output Hidden State.

frame capture for longer durations, we train our network using REINFORCE algorithm to improve the tracking and subsampling performance in the long run. We formulate the problem of joint tracking and adaptive subsampling as a reinforcement learning task and utilize the REINFORCE algorithm (Williams, 1992; Mnih *et al.*, 2014) to perform the prediction step. This REINFORCE algorithm based training procedure is depicted in Figure 5.5. As presented in Figure 5.5, the fully sampled frames i.e. keyframe's feature vectors are concatenated with groundtruth location and subsampling mask for the consecutive frame. Unlike keyframes, non-keyframe feature vectors are not provided with any informative ground truth bounding box and subsampling mask in order to guide the network towards accurate target estimation.

In the terms of reinforcement learning for this framework, the LSTM plays the role of the RL agent, the environment is visual world sensed through the image sensor, and the state is the current image frame (which may or may not be subsampled). The policy function is defined by the network weights ($r = \{r_o, r_{rc}\}$) where r_o are the YOLO network parameters and r_{rc} represents the LSTM parameters. The control set U represents all possible bounding boxes and subsampling masks that an agent can choose to minimize the cost. Thus the network's goal is to learn a policy function $\mu(u_k \mid z_{1:k}; r)$ characterized by network parameters r to determine a $u_k \in U$ (i.e. a bounding box and subsampling mask pair). This function depends on the past state-control trajectories up to time step k,

$$z_{1:k} = \{(i_0, u_0), (i_1, u_1), \ldots, (i_k, u_k)\}$$

where $u \in U$ denotes the predicted bounding box and subsampling mask and i denotes the image frame/state for a given time step. The LSTM encodes the past information $z_{1:k}$ in its hidden states encoded by r_{rc}. Hence, the policy function relies on past interactions between the agent and the environment.

The policy function induces a probability distribution p over all possible state-control trajectories z_k, and the optimization problem is restricted to a parametrized subset $\tilde{\mathcal{P}}_z \subset \mathcal{P}_z$ of distributions and can be reduced to $p(z; r)$ where $\mathcal{P}_z$ represents a set of probability distributions over $\mathcal{Z}$. Hence, the final optimization problem that needs to be solved is as follows (Zhao *et al.*, 2012):

$$L(r) = \mathbb{E}_{p(z_{1:K};r)}[\sum_{k=1}^{K}[g_k] = \mathbb{E}_{p(Z_K;r)}[F(z)] \tag{5.2}$$

where, K is the end time step, g_k is the cost at timestep k, $F(Z)$ represents the cumulative cost till the last time step. As shown in Equation. 5.2, the expectation of the cost generating function with respect to the control space probability distribution is the primary optimization problem.

For our problem of joint object tracking and adaptive subsampling, we have formulated the following cost function:

$$\begin{aligned} g_k = \text{mean}(\mid u_{bb_k} - gt_{bb_k} \mid) + \\ \max(\mid u_{bb_k} - gt_{bb_k} \mid) + \\ (1 - \cos(u_{sm_k}, gt_{sm_k})) \end{aligned} \tag{5.3}$$

where, $u_{bb_k} = \{x, y, w, h\}$ denotes the bounding box predicted by the algorithm and is sampled from a multivariate Gaussian distribution with mean ϕ_{loc_k} (output of tracking algorithm) and fixed variance, gt_{bb_k} is the ground truth bounding box corresponding to state k, u_{sm_k} is the subsampling prediction sampled from a multivariate Gaussian distribution with mean ϕ_{sm_k} and fixed variance, and gt_{sm_k} is the corresponding ground truth subsampling pattern.

The $(1 - \cos(u_{sm_k}, gt_{sm_k}))$ term in the cost function guides the network to learn the accurate subsampling patterns based on the ground truth object location. The ground truth subsampling provides supervision for coarse grained subsampling in order to zoom in on the frame region containing the object. The $\text{mean}(\mid u_{bb_k} - gt_{bb_k} \mid) + \max(\mid u_{bb_k} - gt_{bb_k} \mid)$ term guides the network to focus in on the zoomed in region of the frame and get the finer object location with stronger precision.

Since, the expectation operation requires an integral over a probability distribution defined by the inaccessible policy, we make the assumption that $p(z; r_k)$ is a discrete probability distribution. Furthermore, since the control space $U(i)$ is defined by a probability distribution, we can construct an episodic algorithm. This results in the following simplification in the gradient (Zhao *et al.*, 2012):

$$\nabla_r L \approx \tfrac{1}{K} \sum_{n=1}^{N} \sum_{k=1}^{K} \nabla_r \log \mu(u_k^n \mid z_k^n; r)(F_k^n(z) - b_k) \tag{5.4}$$

The variable b_k is used to compensate for the high variance exhibited by the episodic outputs and is the expectation of the reward function.

Testing At test time, the LSTM receives feature vectors of fully sampled frames from the YOLO only during the beginning of the keyframing interval. Once the prediction phase is activated, the YOLO network no longer receives fully sampled image frames and starts receiving non keyframes i.e. subsampled frames (Figure. 5.4). The LSTM trained on image features is able to track the target and produce the subsampling mask for consecutive frames partly due to the partial subsampled image features from the YOLO network along with the object trajectory the network has learnt to track implicitly. As it transpires, our network is capable of making reliable predictions without receiving fully sampled image features at every time step.

5.2.7 Experiments

Datasets We have evaluated our algorithm on three different datasets: 1) TB-100 (Wu *et al.*, 2013), 2) LaSOT (Fan *et al.*, 2019) and 3) TrackingNet (Muller *et al.*, 2018). The video sequences comprising these datasets feature a wide variety of objects in motion including people, animals, vehicles, etc. We randomly split up the TB100 dataset into training and testing sets like in (Zhang *et al.*, 2017). However, instead of creating the split within video sequences as in (Zhang *et al.*, 2017), we chose a random set of 81 videos for training. We also use 30 and 100 randomly sampled videos for training from LaSOT and TrackingNet datasets, respectively. The reasoning for these spilts and also for using TB-100 for training is that the primary goal of our proposed method is not to compare the methods for conventional object tracking rather it is to analyze the performance on subsampled video sequences and hence energy-accuracy trade-off. For fair comparison we also used the similar splits for the baseline methods as well. We use only a subset of the main datasets during training in order to account for the complexity of the LSTM network on top of the computational complexity of the REINFORCE algorithm. We were further motivated to down-select the training

videos to investigate the generalizability of our RL-trained LSTM network. Even with sparse training data, our model manages to anticipate the state-control trajectories remarkably well at test time.

Although there are instances where multiple objects are present per frame, we use the ground truth labels to perform single object tracking. Furthermore, to develop ground truth subsampling pattern masks needed for stronger supervision, we resized all videos to 448×448 and gridded the frames into 7×7 grids forming a total of 4096 patches in each image frame. Each patch is assigned a binary label of 0 or 1 depending on whether a portion of the ground truth object lies in the patch or not. Consequently, the 4096-D vector is used as a ground truth subsampling mask during training.

Baselines To validate the effectiveness of the proposed tracker, we compare the performance of our network against two types of baselines: (1) predictive trackers as well as (2) state-of-the art tracking architectures deployed at test time on adaptive subsampled videos.

For predictive trackers, we utilize baseline systems with similar structure to ours, wherein we couple an object detector with a Kalman filter (Kalman, 1960) as shown in (Iqbal *et al.*, 2020), but we introduce variation by swapping out the tiny YOLO with various detectors in the pipeline. This approach was also developed by (Li *et al.*, 2021) as a type of new baseline for visual tracking algorithms. The various object detectors we utilize include the YOLO architecture (Iqbal *et al.*, 2020), a Kernalized Correlation Filter (KCF) (Henriques *et al.*, 2014), a Distractor-Aware Tracker (DAT) (Possegger *et al.*, 2015), and Efficient Convolution Operators for Tracking (ECO) (Danelljan *et al.*, 2017).

For state-of-the-art object trackers, we use two recent methods: Accurate tracking

by overlap maximization (ATOM) (Danelljan *et al.*, 2019) and Learning discriminative model prediction for tracking (DiMP) (Bhat *et al.*, 2019). ATOM determines the target using high-level information during offline learning and then a dedicated component for classification is trained online to maximize the discerning capabilities of the network while dealing with distractors in the input scene (Danelljan *et al.*, 2019). DiMP is an end-to-end tracking architecture wherein both foreground and background information are leveraged for target prediction (Bhat *et al.*, 2019). While both of these trackers are state-of-the-art, we show in our experimental results they are not well-suited for adaptively subsampled images.

Implementation Details We implement the proposed algorithm using the PyTorch framework. A learning rate of $\lambda = 0.0001$ was selected after careful empirical analysis during the initial training phase and the Adam optimizer was chosen. To train our network, we chose a keyframing interval of 11 based on the fact that a duration of 11 frames doesn't signify a huge change in motion trajectory of the object while simultaneously being a good choice for maintaining energy savings/accuracy trade-off. We have also conducted an ablation study demonstrating the effect of increasing the keyframing interval for the proposed method as well as the baselines.

On average, the network needs at least three days of training on the Nvidia GeForce RTX 2080 Ti graphics card in order to converge on a dataset. However, the test time implementation can be performed in real time. Per frame computation time on GPU using the proposed method during testing is approximately 3.4 ms.

5.2.8 Performance Evaluation

Benchmarking Tables 5.3 and 5.4 illustrate the effectiveness of the proposed tracker in terms of achieving high object tracking precision (mean average precision -

Method	TB100	LaSOT	TrackingNet
YOLO (Redmon *et al.*, 2016)+KF	0.0626	0.1311	0.2857
KCF (Henriques *et al.*, 2014)+KF	0.2156	0.1780	0.2865
DAT (Possegger *et al.*, 2015)+KF	0.1905	0.1469	0.2463
ECO (Danelljan *et al.*, 2017)+KF	0.4172	0.3293	0.2936
ATOM (Danelljan *et al.*, 2019)	0.3063	0.3043	0.3654
DiMP (Bhat *et al.*, 2019)	0.3155	0.3223	0.3452
Ours	**0.5113**	**0.4979**	**0.5177**

Table 5.3: Results. Our Method vs. Baselines. We Report the AUC Scores With IoU@[0:0.05:1] and Keyframing Interval of 11 on the Three Benchmarking Datasets - TB100, LaSOT and TrackingNet.

mAP) while maintaining satisfactory energy efficiency in terms of image resolution. Tables 5.3 shows that our method outperforms all the baselines and achieves an AUC score of 0.5113, 0.4979 and 0.5177 on the TB100, LaSOT and TrackingNet datasets, respectively. Furthermore, we achieve high energy savings in terms of ratio of pixels turned off per frame on all three datasets, as has been shown in Table 5.4. Note that our method is outperformed in terms of energy savings by most of the other methods with a keyframing interval of 11. This can be attributed to the fact that the other methods are not trained for adaptive subsampling and, therefore, are prone to missing target objects and switching pixels off inside the region of interest. This provides higher energy savings at the cost of deterioration of tracking performance. Hence maintaining the energy-accuracy trade-off is the key which is well achieved by our method.

Ablation Study of Subsampling Loss We formulate our loss as a function of both the bounding box prediction loss as well as the subsampling loss. Analyzing the

Method	Dataset	Ratio of Pixels Off	B1/B2/B3 Power(mW)
Ours	TB100	0.7839	9143 / 2514 / 510
	LaSOT	0.8736	5113 / 1406 / 285
	TrackingNet	0.7770	9212 / 2533 / 514
YOLO (Redmon *et al.*, 2016)+KF	TB100	0.6489	15531 / 4270 / 867
	LaSOT	0.8085	4835 / 1329 / 270
	TrackingNet	0.6195	14106 / 3879 / 787
KCF (Henriques *et al.*, 2014)+KF	TB100	0.9732	4957 / 1363 / 276
	LaSOT	0.8432	6247 / 1717 / 348
	TrackingNet	0.7697	9993 / 2747 / 557
DAT (Possegger *et al.*, 2015)+KF	TB100	0.9741	4894 / 1345 / 273
	LaSOT	0.8443	6193 / 1703 / 345
	TrackingNet	0.7721	9896 / 2721 / 552
ECO (Danelljan *et al.*, 2017)+KF	TB100	0.9712	5076 / 1395 / 283
	LaSOT	0.8281	7228 / 1987 / 403
	TrackingNet	0.7437	10993 / 3023 / 613
ATOM (Danelljan *et al.*, 2019)	TB100	0.9758	4853 / 1334 / 270
	LaSOT	0.8589	4839 / 1330 / 270
	TrackingNet	0.8270	7139 / 1963 / 398
DiMP (Bhat *et al.*, 2019)	TB100	0.9850	4195 / 1153 / 234
	LaSOT	0.8677	4657 / 1280/ 260
	TrackingNet	0.8499	6181 / 1699 / 345

Table 5.4: Energy Results for Adaptive Subsampling With a Keyframing Interval of 11. Our Method vs. The Baselines.

network performance on the test data by training the network both with and without the subsampling loss, we observe the advantage of implementing the proposed loss function. After having trained the network for roughly the same number of epochs, we obtain a test mAP (IoU@0.5) of 0.3388 on the TB100 dataset without the subsampling prediction loss and a test mAP of 0.5262 with the subsampling prediction loss. This can be attributed to the fact that at the instances when the network is not able to converge to the correct ROI, the subsampling information may capture the correct location information. Essentially, the image frame's grid-wise division helps encode the moving objects' correct localization even for erroneous bounding box predictions.

Keyframing We refer to the fully sampled images that the network receives as the keyframes and the rest are referred to as subsampled frames. After the first fully sampled frame is processed through the Tiny YOLO network + LSTM (update phase), we get the object location and subsampling mask prediction. For the consecutive non-keyframes, the LSTM accepts a feature vector extracted from a subsampled frame wherein the scene content inside the previously predicted bounding box and subsampling mask is read out from the sensor as the new input, with all other pixels switched off (prediction phase). A user-defined interval triggers the next update phase, i.e. the reception of the next keyframe. Note that a longer interval will result in higher energy savings at the expense of tracking precision. The interval can be adapted as per the fidelity needs of the application.

Figure 5.6 depicts the effect of increasing the interval between the update stage and the prediction stage of the tracking algorithm. Comparing the effect of increasing the keyframing interval, it is evident that our method is able to maintain significant precision even at higher keyframing intervals. On the contrary, techniques like the KCF+KF and especially the ECO+KF, which are shown to perform noticeably well at

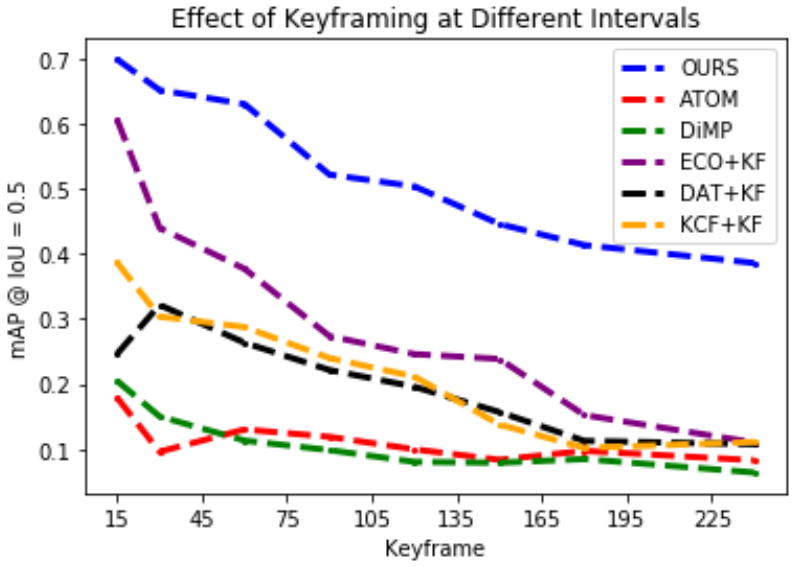


Figure 5.6: Results for the Keyframing Experiment. We Have Swept the Keyframing Interval From 15 to 240 for Our Method and All of the Baselines on the TB100 Dataset and Reported The mAP (IoU@0.5). It Is Evident That Our Method Is Able to Maintain the Tracking Fidelity for a Longer Duration.

lower intervals, cannot sustain that same performance at higher intervals. Further, the ATOM and DiMP methods start breaking down as the keyframing interval increases. The reason being, the frequency of fully sampled frame information has decreased with the increase of keyframing interval, and these methods don't work very well when there is dearth of target specific information. This proves the potential efficacy of our technique for applications where both tracking accuracy and energy efficiency are of prime importance. Note that the keyframing experiment was conducted on the TB100 test data. We attribute our network's improved performance even with a prolonged prediction phase on its ability to zero in on (even coarsely for longer keyframing intervals) the object's region.

Figure 5.6 shows the effect of keyframing on mAP (IoU@0.5) for intervals of up to 240 frames. For most computer vision based applications, an effective frame rate of 60 FPS affords satisfactory latency. Therefore, it is promising that our network retains

it's performance for the most part even at the 60 frames interval. This implies that it will not require the next fully sampled frame for an entire second when integrated with a real camera system. Thus extremely high degree of object motion for long durations would degrade our network performance similar to the baselines.

Power Analysis Adaptive subsampling offers an energy efficient solution whereby pixels are switched off outside of the ROI for non-keyframes, thus saving energy. To estimate the energy savings we achieve with our RL tracking algorithm, we characterize the energy requirements of several CMOS image sensors based on analysis from (LiKamWa *et al.*, 2013; Iqbal *et al.*, 2021). Using the ROIs generated from the proposed algorithm, we assert that the image sensor can skip certain columns during the frame read out and read only the pixels from the predicted regions. Since fewer pixels would be read out, it would result in substantial power savings while sensing. We model power for sensors B1, B2 and B3 from (LiKamWa *et al.*, 2013) with resolution 3264x2448, 2592x1944 and 752x480 respectively.

Then the model equations show that the average power consumption is proportional to the image resolution (LiKamWa *et al.*, 2013):

$$P = \frac{P_{idle}T_{exp} + P_{active}T_{active}}{T_{frame}} \tag{5.5}$$

$$P = \alpha_1.R.T_{exp}.f + \frac{R.c_2.N}{f} \tag{5.6}$$

where R represents frame rate (fixed at 30 fps), T_{exp} is the exposure time (fixed at 0.05ms), N represents frame resolution, c_2 denotes static power consumption (fixed at for every sensor: B1: 159.0, B2: 93.0 and B3: 13.1), α_1 (fixed at for every sensor: B1: $4.0E-06$, B2: $8.2E-07$ and B3: $3.35E-06$) is a sensor intrinsic independent of resolution and f represents the optimal clock frequency dependent on resolution $(\frac{c_2.N}{\alpha_1.T_{exp}})^{1/2}$.

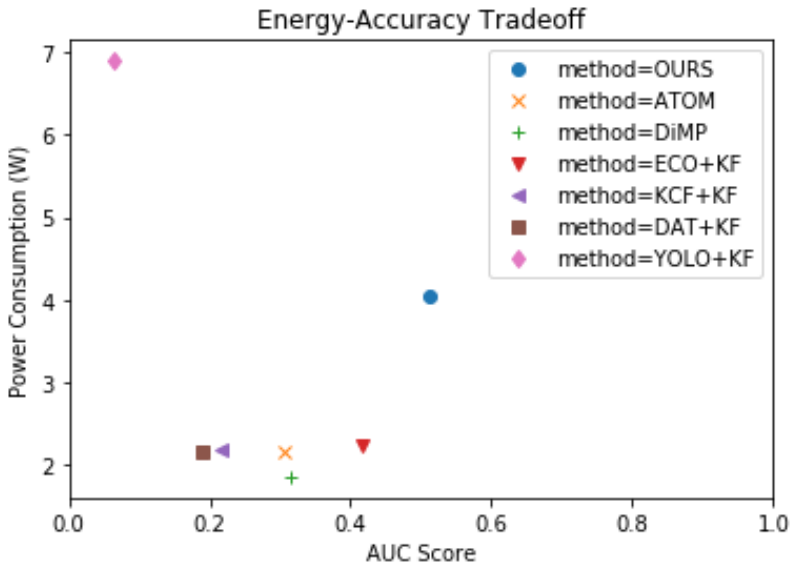


Figure 5.7: Scatter Plot Demonstrating the Accuracy vs. Energy Savings Tradeoff for the TB100 Dataset (With a Keyframing Interval of 11). Our Method Provides the Highest AUC Score With Satisfactory Energy Savings.

The power consumption in milliWatts is presented in Table 5.4. We see more savings in sensors B2 and B3 since, they have a higher resolution. We fixed parameter T_{exp} to 0.05ms which works typically for outdoor settings and a frame rate R of 30fps.

Figure 5.7 visualizes the power-accuracy tradeoff of the various methods, where accuracy is denoted by the AUC score on the TB100 dataset and corresponding power consumption is given in watt. As is evident from the tradeoff plot, other methods rank higher in terms of energy savings but at the expense of tracking performance. On the other hand, our network, although slightly more power-hungry, offers superior tracking performance because of it having learnt the implicit subsampling information during training. Therefore, in terms of energy-accuracy tradeoff, our method strikes the right balance and sustains good tracking accuracy with reasonable energy savings.

5.3 Summary

In this chapter, we considered data selection/capture mechanism as a prior. We argued that for efficient resource utilization, it is important to understand the data priority in several applications. We address this from the view point of object detection and tracking. We developed two adaptive subsampling paradigms which aim to read specific regions from a video frame while it is being captured through an image sensor. The first paradigm was based on frame intensity i.e. we keep capturing the same region of interest as long as the change in frame intensity in consecutive frames is below an empirically chosen threshold. The second adaptive subsampling paradigm is a predictive method which uses REINFORCE algorithm to predict the region of interest masks for consecutive frames and uses these masks for sequential frame capture. Using both the paradigms, we show that the algorithms are able to capture regions of interest with high accuracy and hence, they can save the energy and bandwidth in terms of capturing the data in real time using sensors. We believe this research is the first step towards superior embedded object tracking algorithms deployed in the wild while maintaining energy-efficiency.

Chapter 6

CONCLUSIONS

Understanding of different priors which altogether lead to the current success of deep learning models for several applications is critical. The main challenge lies in understanding that human intervention during the deployment of any deep learning systems in terms of data capture or model selection serves as a constraint which will govern the final task performance. Hence, in this work we studied the effect of several priors on different applications ranging from sequence modelling to image classification and object tracking. In sequence modelling problem, we consider applications such as, audio source separation, clinical time-series based diagnosis and global horizontal irradiance forecasting. For all three applications, we choose similar architectural design choices i.e. using dilated dense convolutions. This was motivated by the fact that robust multi-scale feature extraction achieved by dilated dense convolutions is able to provide richer context and give a bird's eye view of longer temporal histories. Based on the empirical evidence, we show that robust features obtained using such model priors resulted in better task performance.

Apart from model prior, we also study the effect of data priors for image classification applications. In terms of data priors, we primarily considered task and domain priors. Domain priors imply the prior information that can be leveraged from different data distributions for the same task. In conventional transfer learning, the assumption is that all the available data distributions contribute equally such that model can perform well on the new distribution. However, in this work through our proposed optimization paradigm i.e. `Invenio`, we show that systematic knowledge transfer is the key as extremely disparate distributions have been shown to lead to

inferior generalization on new data. `Invenio` achieves this systematic knowledge transfer through a semantic space that is obtained using the gradients during the model training step. Similar semantic space unfolds for multiple observed tasks as well. The semantic space is able to quantify the relationships between different data distributions (or tasks) to show which distribution is more closely related to the other distribution and uses these weightings to achieve improved generalization (domains and tasks).

Lastly, we also consider data selection/capture mechanism as a prior. We argue that for efficient resource utilization, it is important to understand the data priority in several applications. We address this from the view point of object detection and tracking. We develop two adaptive subsampling paradigms which aim to read specific regions from a video frame while it is being captured through an image sensor. The first paradigm is based on frame intensity i.e. we keep capturing the same region of interest as long as the change in frame intensity in consecutive frames is below an empirically chosen threshold. The second adaptive subsampling paradigm is a predictive method which uses REINFORCE algorithm to predict the region of interest masks for consecutive frames and uses these masks for sequential frame capture. Using both the paradigms, we show that the algorithms are able to capture regions of interest with high accuracy and hence, they save the energy and bandwidth in terms of capturing the data in real time using sensors.

Acharya, U. R., H. Fujita, S. L. Oh, Y. Hagiwara, J. H. Tan and M. Adam, "Application of deep convolutional neural network for automated detection of myocardial infarction using ecg signals", Information Sciences **415**, 190–198 (2017).

Achille, A., M. Lam, R. Tewari, A. Ravichandran, S. Maji, C. C. Fowlkes, S. Soatto and P. Perona, "Task2vec: Task embedding for meta-learning", ArXiv **abs/1902.03545** (2019a).

Achille, A., G. Paolini, G. Mbeng and S. Soatto, "The information complexity of learning tasks, their structure and their distance", arXiv preprint arXiv:1904.03292 (2019b).

Alexe, B., T. Deselaers and V. Ferrari, "What is an object?", in "2010 IEEE computer society conference on computer vision and pattern recognition", pp. 73–80 (IEEE, 2010).

Alexe, B., T. Deselaers and V. Ferrari, "Measuring the objectness of image windows", IEEE transactions on pattern analysis and machine intelligence **34**, 11, 2189–2202 (2012).

Andreotti, F., O. Carr, M. A. Pimentel, A. Mahdi and M. De Vos, "Comparing feature-based classifiers and convolutional neural networks to detect arrhythmia from short segments of ecg", Computing **44**, 1 (2017).

Apicharttrisorn, K., X. Ran, J. Chen, S. V. Krishnamurthy and A. K. Roy-Chowdhury, "Frugal following: Power thrifty object detection and tracking for mobile augmented reality", Proceedings of the 17th Conference on Embedded Networked Sensor Systems pp. 96–109 (2019).

Bai, S., J. Z. Kolter and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling", arXiv preprint arXiv:1803.01271 (2018a).

Bai, S., J. Z. Kolter and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling", arXiv:1803.01271 (2018b).

Baxter, J., "A model of inductive bias learning", Journal of artificial intelligence research **12**, 149–198 (2000).

Belfor, R. A., M. P. Hesp, R. L. Lagendijk and J. Biemond, "Spatially adaptive subsampling of image sequences", IEEE Transactions on Image Processing **3**, 5, 492–500 (1994).

Bertinetto, L., J. Valmadre, J. F. Henriques, A. Vedaldi and P. H. Torr, "Fully-convolutional siamese networks for object tracking", European conference on computer vision pp. 850–865 (2016).

Bhat, G., M. Danelljan, L. V. Gool and R. Timofte, "Learning discriminative model prediction for tracking", Proceedings of the IEEE/CVF International Conference on Computer Vision pp. 6182–6191 (2019).

Bhattacharjee, A. D. and A. R. Chowdhury, "Short-Term Solar Irradiance Forecasting Using Long Short Term Memory Variants", in "Proceedings of International Conference on Data Science and Applications,", pp. 227–243 (Springer, 2020).

Black, J., T. Ellis and P. Rosin, "Multi view image surveillance and tracking", Proceedings. Workshop on Motion and Video Computing pp. 169–174 (2002).

Bolme, D. S., J. R. Beveridge, B. A. Draper and Y. M. Lui, "Visual object tracking using adaptive correlation filters", 2010 IEEE computer society conference on computer vision and pattern recognition pp. 2544–2550 (2010).

Bosch, J. L., Y. Zheng and J. Kleissl, "Deriving cloud velocity from an array of solar radiation measurements", Solar Energy **87**, 196–203 (2013).

Bousseljot, R., D. Kreiseler and A. Schnabel, "Nutzung der ekg-signaldatenbank cardiodat der ptb über das internet", Biomedizinische Technik/Biomedical Engineering **40**, s1, 317–318 (1995).

Bradley, A. P., "The use of the area under the roc curve in the evaluation of machine learning algorithms", Pattern recognition **30**, 7, 1145–1159 (1997).

Buckler, M., S. Jayasuriya and A. Sampson, "Reconfiguring the imaging pipeline for computer vision", Proceedings of the IEEE International Conference on Computer Vision pp. 975–984 (2017).

Casares, M. and S. Velipasalar, "Adaptive methodologies for energy-efficient object detection and tracking with battery-powered embedded smart cameras", IEEE Transactions on Circuits and Systems for Video Technology **21**, 10, 1438–1452 (2011).

Chen, B., D. Wang, P. Li, S. Wang and H. Lu, "Real-time'actor-critic'tracking", in "Proceedings of the European conference on computer vision (ECCV)", pp. 318–334 (2018).

Choi, J., J. Kwon and K. M. Lee, "Visual tracking by reinforced decision making", (2017).

Chow, C. W., B. Urquhart, M. Lave, A. Dominguez, J. Kleissl, J. Shields and B. Washom, "Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed", Solar Energy **85**, 11, 2881–2893 (2011).

Clifford, G. D., C. Liu, B. Moody, L.-w. H. Lehman, I. Silva, Q. Li, A. Johnson and R. G. Mark, "Af classification from a short single lead ecg recording: the physionet/computing in cardiology challenge 2017", Computing **44**, 1 (2017).

Danelljan, M., G. Bhat, F. S. Khan and M. Felsberg, "Atom: Accurate tracking by overlap maximization", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 4660–4669 (2019).

Danelljan, M., G. Bhat, F. Shahbaz Khan and M. Felsberg, "Eco: Efficient convolution operators for tracking", Proceedings of the IEEE conference on computer vision and pattern recognition pp. 6638–6646 (2017).

Danelljan, M., G. Hager, F. Shahbaz Khan and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking", Proceedings of the IEEE international conference on computer vision pp. 4310–4318 (2015).

Danelljan, M., A. Robinson, F. S. Khan and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking", in "European conference on computer vision", pp. 472–488 (Springer, 2016).

de Diego, S. L., *Automated Interpretation of Abnormal Adult Electroencephalograms* (Temple University, 2017).

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", in "2009 IEEE conference on computer vision and pattern recognition", pp. 248–255 (Ieee, 2009a).

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", in "Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on", pp. 248–255 (Ieee, 2009b).

Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", (2018).

Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", in "NAACL-HLT", (2019).

Dolara, A., S. Leva and G. Manzolini, "Comparison of different physical models for pv power output prediction", Solar energy **119**, 83–99 (2015).

Dong, J. and Y. Ye, "Adaptive downsampling for high-definition video coding", IEEE Transactions on Circuits and Systems for Video Technology **24**, 3, 480–488 (2013).

Du, S. S., W. Hu, S. M. Kakade, J. D. Lee and Q. Lei, "Few-shot learning via learning the representation, provably", arXiv preprint arXiv:2002.09434 (2020).

Dunnhofer, M., N. Martinel, G. Luca Foresti and C. Micheloni, "Visual tracking by means of deep reinforcement learning and an expert demonstrator", in "Proceedings of The IEEE/CVF International Conference on Computer Vision Workshops", pp. 0–0 (2019).

Duverger, E., C. Penin, P. Alexandre, F. Thiery, D. Gachon and T. Talbert, "Irradiance forecasting for microgrid energy management", in "2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)", pp. 1–6 (IEEE, 2017).

El Mghouchi, Y., E. Chham, M. Krikiz, T. Ajzoul and A. El Bouardi, "On the prediction of the daily global solar radiation intensity on south-facing plane surfaces inclined at varying angles", Energy conversion and management **120**, 397–411 (2016).

Esteva, A., A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun and J. Dean, "A guide to deep learning in healthcare", Nature medicine **25**, 1, 24 (2019).

Fan, H., L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao and H. Ling, "Lasot: A high-quality benchmark for large-scale single object tracking", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 5374–5383 (2019).

Fan, H. and H. Ling, "Sanet: Structure-aware network for visual tracking", Proceedings of the IEEE conference on computer vision and pattern recognition workshops pp. 42–49 (2017).

Faust, O., Y. Hagiwara, T. J. Hong, O. S. Lih and U. R. Acharya, "Deep learning for healthcare applications based on physiological signals: a review", Computer methods and programs in biomedicine (2018).

Fifty, C., E. Amid, Z. Zhao, T. Yu, R. Anil and C. Finn, "Efficiently identifying task groupings for multi-task learning", Advances in Neural Information Processing Systems **34** (2021).

Finn, C., P. Abbeel and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks", in "ICML", (2017a).

Finn, C., P. Abbeel and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks", in "Proceedings of the 34th International Conference on Machine Learning", edited by D. Precup and Y. W. Teh, vol. 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135 (PMLR, International Convention Centre, Sydney, Australia, 2017b), URL `http://proceedings.mlr.press/v70/finn17a.html`.

Gan, Q., Q. Guo, Z. Zhang and K. Cho, "First step toward model-free, anonymous object tracking with recurrent neural networks", arXiv preprint arXiv:1511.06425 (2015).

Ghimire, S., R. C. Deo, N. Raj and J. Mi, "Deep solar radiation forecasting with convolutional neural network and long short-term memory network algorithms", Applied Energy **253**, 113541 (2019).

Goldberger, A. L., L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals", Circulation **101**, 23, e215–e220, circulation Electronic Pages: http://circ.ahajournals.org/content/101/23/e215.full PMID:1085218; doi: 10.1161/01.CIR.101.23.e215 (2000 (June 13)).

Grais, E. M., D. Ward and M. D. Plumbley, "Raw multi-channel audio source separation using multi-resolution convolutional auto-encoders", in "2018 26th European Signal Processing Conference (EUSIPCO)", pp. 1577–1581 (IEEE, 2018).

Guan, J. and Z. Lu, "Task relatedness-based generalization bounds for meta learning", in "International Conference on Learning Representations", (2021).

Guo, Q., W. Feng, C. Zhou, R. Huang, L. Wan and S. Wang, "Learning dynamic siamese network for visual object tracking", in "Proceedings of the IEEE international conference on computer vision", pp. 1763–1771 (2017).

Guo, S., W. Huang, X. Zhang, P. Srikhanta, Y. Cui, Y. Li, S. Belongie, H. Adam and M. Scott, "The imaterialist fashion attribute dataset", arXiv preprint arXiv:1906.05750 (2019).

Harutyunyan, H., H. Khachatrian, D. C. Kale, G. Ver Steeg and A. Galstyan, "Multitask learning and benchmarking with clinical time series data", Scientific data **6**, 1, 96 (2019).

He, A., C. Luo, X. Tian and W. Zeng, "A twofold siamese network for real-time object tracking", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition pp. 4834–4843 (2018).

He, K., X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", in "CVPR", pp. 770–778 (2016).

Held, D., S. Thrun and S. Savarese, "Learning to track at 100 fps with deep regression networks", European Conference on Computer Vision pp. 749–765 (2016).

Henriques, J. F., R. Caseiro, P. Martins and J. Batista, "High-speed tracking with kernelized correlation filters", IEEE transactions on pattern analysis and machine intelligence **37**, 3, 583–596 (2014).

Hochreiter, S. and J. Schmidhuber, "Long short-term memory", Neural computation **9**, 8, 1735–1780 (1997).

Hong, S., M. Wu, Y. Zhou, Q. Wang, J. Shang, H. Li and J. Xie, "Encase: An ensemble classifier for ecg classification using expert features and deep neural networks", in "Computing in Cardiology (CinC), 2017", pp. 1–4 (IEEE, 2017).

Huang, C., S. Lucey and D. Ramanan, "Learning policies for adaptive tracking with deep feature cascades", in "Proceedings of the IEEE International Conference on Computer Vision", pp. 105–114 (2017a).

Huang, G., Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely connected convolutional networks", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 4700–4708 (2017b).

Huang, X., C. Zhang, Q. Li, Y. Tai, B. Gao and J. Shi, "A comparison of hour-ahead solar irradiance forecasting models based on LSTM network", Mathematical Problems in Engineering **2020** (2020).

Iqbal, O., V. I. T. Muro, S. Katoch, A. Spanias and S. Jayasuriya, "Adaptive subsampling for roi-based visual tracking: Algorithms and fpga implementation", arXiv preprint arXiv:2112.09775 (2021).

Iqbal, O., S. Siddiqui, J. Martin, S. Katoch, A. Spanias, D. Bliss and S. Jayasuriya, "Design and fpga implementation of an adaptive video subsampling algorithm for energy-efficient single object tracking", 2020 IEEE International Conference on Image Processing (ICIP) pp. 3065–3069 (2020).

Jang, H. S., K. Y. Bae, H.-S. Park and D. K. Sung, "Solar power prediction based on satellite images and support vector machine", IEEE Transactions on Sustainable Energy **7**, 3, 1255–1263 (2016).

Jansson, A., E. Humphrey, N. Montecchio, R. Bittner, A. Kumar and T. Weyde, "Singing voice separation with deep u-net convolutional networks", (2017).

Kachuee, M., S. Fazeli and M. Sarrafzadeh, "Ecg heartbeat classification: A deep transferable representation", arXiv preprint arXiv:1805.00794 (2018).

Kahou, S. E., V. Michalski, R. Memisevic, C. Pal and P. Vincent, "Ratm: recurrent attentive tracking model", 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) pp. 1613–1622 (2017).

Kalman, R. E., "A new approach to linear filtering and prediction problems", Journal of Basic Engineering **82**, 32–45 (1960).

Katoch, S., D. Mohan, S. Jayasuriya, P. Turaga and A. Spanias, "Adaptive video subsampling for energy-efficient object detection", Asilomar Conference on Signals, Systems, and Computers (2019a).

Katoch, S., G. Muniraju, S. Rao, A. Spanias, P. Turaga, C. Tepedelenlioglu, M. Banavar and D. Srinivasan, "Shading prediction, fault detection, and consensus estimation for solar array control", in "2018 IEEE Industrial Cyber-Physical Systems (ICPS)", pp. 217–222 (IEEE, 2018a).

Katoch, S., K. Thopalli, J. J. Thiagarajan, P. Turaga and A. Spanias, "Invenio: Discovering hidden relationships between tasks/domains using structured meta learning", arXiv preprint arXiv:1911.10600 (2019b).

Katoch, S., P. Turaga, A. Spanias and C. Tepedelenlioglu, "Fast non-linear methods for dynamic texture prediction", in "2018 25th IEEE International Conference on Image Processing (ICIP)", pp. 2107–2111 (IEEE, 2018b).

Kendall, A., Y. Gal and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 7482–7491 (2018).

Khan, N. M., N. Abraham, M. Hon and L. Guan, "Machine learning on biomedical images: Interactive learning, transfer learning, class imbalance, and beyond", in "2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)", pp. 85–90 (IEEE, 2019).

Kim, C. K., H.-G. Kim, Y.-H. Kang, C.-Y. Yun and S. Y. Kim, "Probabilistic prediction of direct normal irradiance derived from global horizontal irradiance over the Korean Peninsula by using Monte-Carlo simulation", Solar Energy **180**, 63–74 (2019).

Kim, D. Y. and M. Jeon, "Data fusion of radar and image measurements for multi-object tracking via kalman filtering", Information Sciences **278**, 641–652 (2014).

Kojuri, J., R. Boostani, P. Dehghani, F. Nowroozipour and N. Saki, "Prediction of acute myocardial infarction with artificial neural networks in patients with nondiagnostic electrocardiogram", Journal of Cardiovascular Disease Research **6**, 2, 51–59 (2015).

Krizhevsky, A. and G. Hinton, "Convolutional deep belief networks on cifar-10", Unpublished manuscript **40**, 7, 1–9 (2010).

Le Gall, D., "Mpeg: A video compression standard for multimedia applications", Communications of the ACM **34**, 4, 46–58 (1991).

Le Guen, V. and N. Thome, "A deep physical model for solar irradiance forecasting with fisheye images", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops", pp. 630–631 (2020).

Lea, C., M. D. Flynn, R. Vidal, A. Reiter and G. D. Hager, "Temporal convolutional networks for action segmentation and detection", in "proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 156–165 (2017).

Li, B., Y. Li, J. Ye, C. Fu and H. Zhao, "Predictive visual tracking: A new benchmark and baseline approach", arXiv preprint arXiv:2103.04508 (2021).

Li, B., W. Wu, Q. Wang, F. Zhang, J. Xing and J. S. Yan, "Evolution of siamese visual tracking with very deep networks", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA", pp. 16–20 (2019).

Li, B., J. Yan, W. Wu, Z. Zhu and X. Hu, "High performance visual tracking with siamese region proposal network", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 8971–8980 (2018).

Li, D., Y. Yang, Y.-Z. Song and T. M. Hospedales, "Learning to generalize: Meta-learning for domain generalization", in "AAAI", (2017a).

Li, X., K. Wang, W. Wang and Y. Li, "A multiple object tracking method using kalman filter", The 2010 IEEE International Conference on Information and Automation pp. 1862–1866 (2010).

Li, Z., F. Zhou, F. Chen and H. Li, "Meta-sgd: Learning to learn quickly for few-shot learning", arXiv preprint arXiv:1707.09835 (2017b).

LiKamWa, R., Y. Hou, J. Gao, M. Polansky and L. Zhong, "Redeye: analog convnet image sensor architecture for continuous mobile vision", ACM SIGARCH Computer Architecture News **44**, 3, 255–266 (2016).

LiKamWa, R., B. Priyantha, M. Philipose, L. Zhong and P. Bahl, "Energy characterization and optimization of image sensing toward continuous mobile vision", Proceeding of the 11th annual international conference on Mobile systems, applications, and services pp. 69–82 (2013).

Lin, W. and L. Dong, "Adaptive downsampling to improve image compression at low bit rates", IEEE Transactions on Image Processing **15**, 9, 2513–2521 (2006).

Liu, B., J. Liu, G. Wang, K. Huang, F. Li, Y. Zheng, Y. Luo and F. Zhou, "A novel electrocardiogram parameterization algorithm and its application in myocardial infarction detection", Computers in biology and medicine **61**, 178–184 (2015).

Liu, Z., P. Luo, S. Qiu, X. Wang and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 1096–1104 (2016).

Liu, Z., P. Luo, X. Wang and X. Tang, "Large-scale celebfaces attributes (celeba) dataset", Retrieved August **15**, 2018, 11 (2018).

Liutkus, A., F.-R. Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono and J. Fontecave, "The 2016 signal separation evaluation campaign", in "International Conference on Latent Variable Analysis and Signal Separation", pp. 323–332 (Springer, 2017).

López, S., I. Obeid and J. Picone, "Automated interpretation of abnormal adult electroencephalograms", MS Thesis, Temple University (2017).

Loshchilov, I. and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts", arXiv preprint arXiv:1608.03983 (2016).

Lucas, B. D. and T. Kanade, "An iterative image registration technique with an application to stereo vision", (1981).

Luo, Y., Z. Chen, J. R. Hershey, J. Le Roux and N. Mesgarani, "Deep clustering and conventional networks for music separation: Stronger together", in "2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)", pp. 61–65 (IEEE, 2017).

Luo, Y. and N. Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation", in "2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)", pp. 696–700 (IEEE, 2018).

Makino, S., S. Araki, R. Mukai and H. Sawada, "Audio source separation based on independent component analysis", in "2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)", vol. 5, pp. V–V (2004).

Marcenaro, L., M. Ferrari, L. Marchesotti and C. S. Regazzoni, "Multiple object tracking under heavy occlusions by using kalman filters based on shape matching", Proceedings. International Conference on Image Processing **3**, III–III (2002).

Marcus, G., "Innateness, alphazero, and artificial intelligence", arXiv preprint arXiv:1801.05667 (2018).

Marcus, G. F., *The birth of the mind: How a tiny number of genes creates the complexities of human thought* (Basic Civitas Books, 2004).

Maurer, A., M. Pontil and B. Romera-Paredes, "The benefit of multitask representation learning", Journal of Machine Learning Research **17**, 81, 1–32 (2016).

McFee, B., C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg and O. Nieto, "librosa: Audio and music signal analysis in python", in "Proceedings of the 14th python in science conference", pp. 18–25 (2015).

Miotto, R., F. Wang, S. Wang, X. Jiang and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges", Briefings in bioinformatics (2017).

Mnih, V., A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning", in "International conference on machine learning", pp. 1928–1937 (PMLR, 2016).

Mnih, V., N. Heess, A. Graves *et al.*, "Recurrent models of visual attention", Advances in neural information processing systems pp. 2204–2212 (2014).

Moody, G. B. and R. G. Mark, "The impact of the mit-bih arrhythmia database", IEEE Engineering in Medicine and Biology Magazine **20**, 3, 45–50 (2001).

Mueller, R., K.-F. Dagestad, P. Ineichen, M. Schroedter-Homscheidt, S. Cros, D. Dumortier, R. Kuhlemann, J. Olseth, G. Piernavieja, C. Reise *et al.*, "Rethinking satellite-based solar irradiance modelling: The SOLIS clear-sky module", Remote sensing of Environment **91**, 2, 160–174 (2004).

Muller, M., A. Bibi, S. Giancola, S. Alsubaihi and B. Ghanem, "Trackingnet: A large-scale dataset and benchmark for object tracking in the wild", Proceedings of the European Conference on Computer Vision (ECCV) pp. 300–317 (2018).

Naderiparizi, S., P. Zhang, M. Philipose, B. Priyantha, J. Liu and D. Ganesan, "Glimpse: A programmable early-discard camera architecture for continuous mobile vision", Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services pp. 292–305 (2017).

Nam, H. and B. Han, "Learning multi-domain convolutional neural networks for visual tracking", Proceedings of the IEEE conference on computer vision and pattern recognition pp. 4293–4302 (2016).

Narayanaswamy, V. S., R. Ayyanar, A. Spanias, C. Tepedelenlioglu and D. Srinivasan, "Connection topology optimization in photovoltaic arrays using neural networks", in "2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)", pp. 167–172 (IEEE, 2019a).

Narayanaswamy, V. S., S. Katoch, J. J. Thiagarajan, H. Song and A. Spanias, "Audio source separation via multi-scale learning with dilated dense u-nets", arXiv preprint arXiv:1904.04161 (2019b).

Ning, G., Z. Zhang, C. Huang, X. Ren, H. Wang, C. Cai and Z. He, "Spatially supervised recurrent convolutional neural networks for visual object tracking", 2017 IEEE International Symposium on Circuits and Systems (ISCAS) pp. 1–4 (2017).

Obeid, I. and J. Picone, "The Temple university hospital eeg data corpus", Frontiers in neuroscience **10**, 196 (2016).

Otsu, N., "A threshold selection method from gray-level histograms", IEEE Transactions on Systems, Man, and Cybernetics **9**, 1, 62–66 (1979).

Pascual, S., A. Bonafonte and J. Serrà, "Segan: Speech enhancement generative adversarial network", arXiv preprint arXiv:1703.09452 (2017).

Pazikadin, A. R., D. Rifai, K. Ali, M. Z. Malik, A. N. Abdalla and M. A. Faraj, "Solar irradiance measurement instrumentation and power solar generation forecasting based on Artificial Neural Networks (ANN): A review of five years research trend", Science of The Total Environment **715**, 136848 (2020).

Pereyra, G., G. Tucker, J. Chorowski, Ł. Kaiser and G. Hinton, "Regularizing neural networks by penalizing confident output distributions", arXiv preprint arXiv:1701.06548 (2017).

Possegger, H., T. Mauthner and H. Bischof, "In Defense of Color-based Model-free Tracking", Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015).

Rafii, Z., A. Liutkus, F.-R. Stöter, S. I. Mimilakis and R. Bittner, "Musdb18-a corpus for music separation", (2017).

Rajan, D., D. Beymer and G. Narayan, "Generalization studies of neural network models for cardiac disease detection using limited channel ecg", arXiv preprint arXiv:1901.03295 (2019).

Rajan, D. and J. J. Thiagarajan, "A generative modeling approach to limited channel ecg classification", arXiv preprint arXiv:1802.06458 (2018).

Rajpurkar, P., A. Y. Hannun, M. Haghpanahi, C. Bourn and A. Y. Ng, "Cardiologist-level arrhythmia detection with convolutional neural networks", arXiv preprint arXiv:1707.01836 (2017).

Rao, S., S. Katoch, V. Narayanaswamy, G. Muniraju, C. Tepedelenlioglu, A. Spanias, P. Turaga, R. Ayyanar and D. Srinivasan, "Machine learning for solar array monitoring, optimization, and control", Synthesis Lectures on Power Electronics **7**, 1, 1–91 (2020).

Reasat, T. and C. Shahnaz, "Detection of inferior myocardial infarction using shallow convolutional neural networks", in "Humanitarian Technology Conference (R10-HTC), 2017 IEEE Region 10", pp. 718–721 (IEEE, 2017).

Redmon, J., S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition pp. 779–788 (2016).

Ronneberger, O., P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation", in "International Conference on Medical image computing and computer-assisted intervention", pp. 234–241 (Springer, 2015).

Roy, S., I. Kiral-Kornek and S. Harrer, "Chrononet: A deep recurrent neural network for abnormal eeg identification", arXiv preprint arXiv:1802.00308 (2018).

Ruder, S., "An overview of multi-task learning in deep neural networks", arXiv preprint arXiv:1706.05098 (2017).

Sadasivam, S., A. Swaminathan and M. Ramachandran, "Always-on camera sampling strategies", US Patent 9,661,221 (2017).

Schirrmeister, R., L. Gemein, K. Eggensperger, F. Hutter and T. Ball, "Deep learning with convolutional neural networks for decoding and visualization of eeg pathology", in "Signal Processing in Medicine and Biology Symposium (SPMB), 2017 IEEE", pp. 1–7 (IEEE, 2017).

Sengupta, M., Y. Xie, A. Lopez, A. Habte, G. Maclaurin and J. Shelby, "The national solar radiation data base (nsrdb)", Renewable and sustainable energy reviews **89**, 51–60 (2018).

Sharda, S., M. Singh and K. Sharma, "RSAM: Robust Self-Attention Based Multi-Horizon Model for Solar Irradiance Forecasting", IEEE Trans. on Sustainable Energy **12**, 2, 1394–1405 (2020).

Sharif Razavian, A., H. Azizpour, J. Sullivan and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition", in "Proceedings of the IEEE conference on computer vision and pattern recognition workshops", pp. 806–813 (2014).

Sharma, L., R. Tripathy and S. Dandapat, "Multiscale energy and eigenspace approach to detection and localization of myocardial infarction", IEEE transactions on biomedical engineering **62**, 7, 1827–1837 (2015).

Shi, X., Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting", Advances in neural information processing systems **28** (2015).

Song, H., D. Rajan, J. J. Thiagarajan and A. Spanias, "Attend and diagnose: Clinical time series analysis using attention models", in "Thirty-Second AAAI Conference on Artificial Intelligence", (2018a).

Song, Y., C. Ma, X. Wu, L. Gong, L. Bao, W. Zuo, C. Shen, R. W. Lau and M.-H. Yang, "Vital: Visual tracking via adversarial learning", Proceedings of the IEEE conference on computer vision and pattern recognition pp. 8990–8999 (2018b).

Song, Z. and L. E. Brown, "Multi-dimensional evaluation of temporal neural networks on solar irradiance forecasting", in "2019 Innovative Smart Grid Technologies-Asia (ISGT Asia)", pp. 4192–4197 (IEEE, 2019).

Sourkov, V., "Igloo: Slicing the features space to represent long sequences", arXiv preprint arXiv:1807.03402 (2018).

Spanias, A., T. Painter and V. Atti, *Audio signal processing and coding* (John Wiley & Sons, 2006).

Srivastava, S. and S. Lessmann, "A comparative study of LSTM neural networks in forecasting day-ahead global horizontal irradiance with satellite data", Solar Energy **162**, 232–247 (2018).

Stoller, D., S. Ewert and S. Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation", arXiv preprint arXiv:1806.03185 (2018).

Strodthoff, N. and C. Strodthoff, "Detecting and interpreting myocardial infarctions using fully convolutional neural networks", arXiv preprint arXiv:1806.07385 (2018).

Sullivan, G. J., J.-R. Ohm, W.-J. Han and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard", IEEE Transactions on circuits and systems for video technology **22**, 12, 1649–1668 (2012).

Sun, M., J. Xiao, E. G. Lim, B. Zhang and Y. Zhao, "Fast template matching and update for video object tracking and segmentation", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition", pp. 10791–10799 (2020).

Supancic III, J. and D. Ramanan, "Tracking as online decision-making: Learning a policy from streaming videos with reinforcement learning", in "Proceedings of the IEEE International Conference on Computer Vision", pp. 322–331 (2017).

Tao, R., E. Gavves and A. W. Smeulders, "Siamese instance search for tracking", Proceedings of the IEEE conference on computer vision and pattern recognition pp. 1420–1429 (2016).

Teijeiro, T., C. A. García, D. Castro and P. Félix, "Arrhythmia classification from the abductive interpretation of short single-lead ecg records", Comput. Cardiol **44**, 1–4 (2017).

Thiagarajan, J. J., D. Rajan, S. Katoch and A. Spanias, "Ddxnet: a deep learning model for automatic interpretation of electronic health records, electrocardiograms and electroencephalograms", Scientific reports **10**, 1, 1–11 (2020).

Thiagarajan, J. J., K. N. Ramamurthy and A. Spanias, "Mixing matrix estimation using discriminative clustering for blind source separation", Digital Signal Processing **23**, 1, 9–18 (2013).

Tripuraneni, N., C. Jin and M. Jordan, "Provable meta-learning of linear representations", in "International Conference on Machine Learning", pp. 10434–10443 (PMLR, 2021).

Uhlich, S., M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi and Y. Mitsufuji, "Improving music source separation based on deep neural networks through data augmentation and network blending", in "2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)", pp. 261–265 (IEEE, 2017).

Ulyanov, D., A. Vedaldi and V. Lempitsky, "Deep image prior", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 9446–9454 (2018).

Valmadre, J., L. Bertinetto, J. Henriques, A. Vedaldi and P. H. Torr, "End-to-end representation learning for correlation filter based tracking", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 2805–2813 (2017).

Van Den Oord, A., S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior and K. Kavukcuoglu, "Wavenet: A generative model for raw audio", CoRR abs/1609.03499 (2016).

Van Horn, G., O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona and S. Belongie, "The inaturalist species classification and detection dataset", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 8769–8778 (2018).

Vincent, E., R. Gribonval and C. Févotte, "Performance measurement in blind audio source separation", IEEE transactions on audio, speech, and language processing **14**, 4, 1462–1469 (2006).

Vinyals, O., C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning", in "Advances in neural information processing systems", pp. 3630–3638 (2016).

Wah, C., S. Branson, P. Welinder, P. Perona and S. Belongie, "The caltech-ucsd birds-200-2011 dataset", (2011).

Wang, F., Z. Mi, S. Su and H. Zhao, "Short-term solar irradiance forecasting model based on artificial neural network using statistical feature parameters", Energies **5**, 5, 1355–1370 (2012).

Wang, G., C. Luo, Z. Xiong and W. Zeng, "Spm-tracker: Series-parallel matching for real-time visual object tracking", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition pp. 3643–3652 (2019a).

Wang, H., R. Cai, B. Zhou, S. Aziz, B. Qin, N. Voropai, L. Gan and E. Barakhtenko, "Solar irradiance forecasting based on direct explainable neural network", Energy Conversion and Management **226**, 113487 (2020).

Wang, L., W. Ouyang, X. Wang and H. Lu, "Visual tracking with fully convolutional networks", Proceedings of the IEEE international conference on computer vision pp. 3119–3127 (2015).

Wang, M. and W. Deng, "Deep visual domain adaptation: A survey", Neurocomputing **312**, 135–153 (2018).

Wang, N., Y. Song, C. Ma, W. Zhou, W. Liu and H. Li, "Unsupervised deep tracking", Proceedings of the IEEE conference on computer vision and pattern recognition pp. 1308–1317 (2019b).

Williams, R. J., "Simple statistical gradient-following algorithms for connectionist reinforcement learning", Machine learning **8**, 3-4, 229–256 (1992).

Wu, Y., J. Lim and M.-H. Yang, "Online object tracking: A benchmark", IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2013).

Yang, J., J. Lu, D. Batra and D. Parikh, "A faster pytorch implementation of faster r-cnn", https://github.com/jwyang/faster-rcnn.pytorch (2017).

Yang, Y. and T. M. Hospedales, "A unified perspective on multi-domain and multi-task learning", in "International Conference on Learning Representations (ICLR)", (2015).

Yang, Z., Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding", arXiv preprint arXiv:1906.08237 (2019a).

Yang, Z., Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding", ArXiv **abs/1906.08237** (2019b).

Yi, J., S. Choi and Y. Lee, "Eagleeye: wearable camera-based person identification in crowded urban spaces", Proceedings of the 26th Annual International Conference on Mobile Computing and Networking pp. 1–14 (2020).

Yu, T., S. Kumar, A. Gupta, S. Levine, K. Hausman and C. Finn, "Gradient surgery for multi-task learning", Advances in Neural Information Processing Systems **33**, 5824–5836 (2020).

Yun, S., J. Choi, Y. Yoo, K. Yun and J. Young Choi, "Action-decision networks for visual tracking with deep reinforcement learning", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 2711–2720 (2017).

Zamir, A. R., A. Sax, W. B. Shen, L. J. Guibas, J. Malik and S. Savarese, "Taskonomy: Disentangling task transfer learning", 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 3712–3722 (2018).

Zang, H., L. Liu, L. Sun, L. Cheng, Z. Wei and G. Sun, "Short-term global horizontal irradiance forecasting based on a hybrid CNN-LSTM model with spatiotoral correlations", Renewable Energy **160**, 26–41 (2020).

Zhang, D., H. Maei, X. Wang and Y.-F. Wang, "Deep reinforcement learning for visual object tracking in videos", arXiv preprint arXiv:1701.08936 (2017).

Zhang, Y. and Q. Yang, "A survey on multi-task learning", IEEE Transactions on Knowledge and Data Engineering (2021).

Zhang, Z. and H. Peng, "Deeper and wider siamese networks for real-time visual tracking", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition", pp. 4591–4600 (2019).

Zhao, T., H. Hachiya, G. Niu and M. Sugiyama, "Analysis and improvement of policy gradient estimation", Neural Networks **26**, 118–129 (2012).

Zhu, Z., Q. Wang, B. Li, W. Wu, J. Yan and W. Hu, "Distractor-aware siamese networks for visual object tracking", in "Proceedings of the European Conference on Computer Vision (ECCV)", pp. 101–117 (2018).

BIOGRAPHICAL SKETCH

Sameeksha Katoch is currently pursuing her Ph.D. degree with the school of Electrical, Computer and Energy Engineering at Arizona State University. She received her Master's degree in electrical engineering from Arizona State University (ASU) in 2018 and a Bachelor's degree in electronics and communication engineering from the National Institute of Technology, Srinagar, India, in 2015. Her research interests include developing robust priors in terms of data and model priors effective in varied deep learning applications. She also has experience in energy data analytics for photovoltaic array monitoring. Her internship with Prime Solutions Group Inc.(2018) focused on algorithm development for photovoltaic weather and energy data. She also interned with Lawrence Livermore National Laboratory during the summer of 2020 where she built models pertaining to healthcare AI.