5-4-2021

Data Efficient Learning: Towards Reducing Risk and Uncertainty of Data Driven Learning Paradigm

Krishanu Sarker
Georgia State University

Data Efficient Learning: Towards Reducing Risk and Uncertainty of Data Driven Learning Paradigm

by

Krishanu Sarker

Under the Direction of Saeid Belkasim, PhD

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2021

ABSTRACT

The success of Deep Learning in various tasks is highly dependent on the large amount of domain-specific annotated data, which are expensive to acquire and may contain varying degrees of noise. In this doctoral journey, our research goal is first to identify and then tackle the issues relating to data that causes significant performance degradation to real-world applications of Deep Learning algorithms.

Human Activity Recognition from RGB data is challenging due to the lack of relative motion parameters. To address this issue, we propose a novel framework that introduces the skeleton information from RGB data for activity recognition. With experimentation, we demonstrate that our RGB-only solution surpasses the state-of-the-art, all exploit RGB-D video streams, by a notable margin.

The predictive uncertainty of Deep Neural Networks (DNNs) makes them unreliable for real-world deployment. Moreover, available labeled data may contain noise. We aim to address these two issues holistically by proposing a unified density-driven framework, which can effectively denoise training data as well as avoid predicting uncertain test data points. Our plug-and-play framework is easy to deploy on real-world applications while achieving superior performance over state-of-the-art techniques. To assess effectiveness of our proposed framework in a real-world scenario, we experimented with x-ray images from COVID-19 patients.

Supervised learning of DNNs inherits the limitation of a very narrow field of view in terms of known data distributions. Moreover, annotating data is costly. Hence, we explore self-supervised Siamese networks to avoid these constraints. Through extensive experimentation, we demonstrate that self supervised method perform surprisingly comparative to its supervised counterpart in a real world use-case. We also delve deeper with activation mapping and feature distribution visualization to understand the causality of this method.

Through our research, we achieve a better understanding of issues relating to data-driven learning while solving some of the core problems of this paradigm and expose some novel and intriguing research questions to the community.

INDEX WORDS: Computer Vision, Deep Learning, Data Efficient Learning, Data Denoising, Data Abstention, Human Action Recognition, Self-Supervised Learning.

Data Efficient Learning: Towards Reducing Risk and Uncertainty of Data Driven Learning Paradigm

by

Krishanu Sarker

Committee Chair:	Saeid Belkasim
Committee:	Shihao Ji
	Rajshekhar Sunderraman
	Mustaque Ahamad

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

May, 2021

DEDICATION

I dedicate my PhD to my beloved and beautiful wife, Sharbani! I have put her through a lot during this long journey of my PhD. She endures my annoying whining, gives me hope during my frustration spells, and she always believed in me during my lowest times. I could have not been here without her.

ACKNOWLEDGMENTS

First and foremost I would like to express my heartiest gratitude to my parents. They have spent their life building our future brick by brick. I know this achievement will make them proud. I cannot thank my wife, Sharbani, enough for always being there. Without their tremendous understanding and encouragement in the past few years, it would be impossible for me to complete my PhD.

I am extremely grateful to my advisors, Dr. Belkasim and Dr. Ji for their invaluable advice, continuous support, and patience during my PhD study. Their in depth knowledge and experience have guided me during tough times. I would also like to thank Dr. Sunderraman and Dr. Ahamad for their valuable suggestions and directions. I would like to thank all faculties, staffs and fellow grads for making my study and life a wonderful time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

The success of Deep Learning Models has led to the deployment of these models into real-world use-cases. However, often the success achieved in a controlled environment does not translate well into real-world scenarios. In this research path, we aim to thoroughly investigate this issue and propose solutions to mitigate it. In this chapter, we present the general introduction of the research we embark upon to provide a summary of the work, our motivation and contributions, and the challenges we faced along the way.

1.1 Data Driven Learning

Deep Learning has been highly successful in a variety of tasks, including but not limited to, Computer Vision [8, 9, 10], Audio Processing [11], Natural Language Processing [12], Medical Imaging [13], etc. One of the most notable successes can be observed in Computer Vision; from simple handwritten digit detection [14] to complex Human Activity Recognition [10]. Deep Models have outperformed humans in complex visual tasks [8, 15]. However, this success did require an extensive amount of high-quality data, e.g., ImageNet [8], one million youtube videos[16], etc. Deep Neural Networks (DNN) heavily rely on data to learn meaningful patterns or features. One of the reasons that traditional shallow Machine Learning was not as successful as Deep Learning was that the features were required to be hand-crafted depending on the task; hence these models did not translate well if presented with complex data. The automated feature extraction capability of DNNs has caused a revolution in the data-driven learning paradigm.

The success of Deep Learning inspired wide deployment of DNNs in practice. Without even realizing, we are utilizing technologies powered by Deep Learning in our day to day life [17]. From search engines to voice assistants like Alexa or Siri, face unlocking smartphones to cashier-less shopping, we experience the convenience of Deep Learning powered Artificial Intelligence (AI). However, researchers face significant issues when translating DNN models, that are proven successful with benchmark datasets, to real-world scenarios. Very large DNNs require heavy hardware support, blocking deployment of such high performing systems in computationally constrained environments, e.g., IoT and mobile devices. Data-driven learning methods often fail to extract useful features if the data size is too small or not clean. In some cases, traditional DNNs fail to extract sufficient information to identify subtle differences from available limited training data. Researchers have been working on bypassing these issues of labeled data for supervised training and working on alternative approaches to training Deep Neural Networks (DNNs). Unsupervised, semi-supervised, and self-supervised learning have been proposed and a great effort has resulted in significant improvement in these fronts. However, there is still a lot of research to be done in this regard.

1.2 Explainable Artificial Intelligence

The lack of transparency and explainability of AI systems has created a vast open area for potential research [18]. Very recently, the term, “Explainable Artificial Intelligence” has gained popularity among researchers and general users. Not long ago, people used to refer to Deep Learning as magic and got used to treating DNNs as a magical black box, where given enough data and a task, it can generate good performance. In some use-cases, this approach worked well (e.g., face recognition). However, this trend hit a brick wall when, in some scenarios, high performing DNNs failed to deliver good enough results to a similar yet real-world task. Researchers tried to understand the rationale behind and figured that treating DNNs as black boxes can lead to several pitfalls, hence the performance degradation.

1.3 Issues faced by Data-Driven Learning

Explainable AI systems aim towards associating causality and rationale behind the decisions made by DNNs, which can explain the misled predictions from them. In this research path, we intend to exploit the insights we gain from the explainability to empower the DNNs to differentiate unknowns from knowns and better utilize available data.

1.3.1 Perfect Data does not Exist

The availability of desired data can dramatically improve the performance of DNNs, yet often such data is scarce and expensive in real-world scenarios [10]. Hence, the optimal system should be able to make the best use of the available data.

For example, Human Activity Recognition (HAR) is inherently complex due to the inter-class affinity and intra-class diversity. Hence, recognizing activity is a difficult task, which has attracted numerous researchers' attention [19, 20, 21, 22]. Most of the high performing methods utilize RGB-D video data or sensor data to track movements accurately [3, 5, 7, 6]. However, in many real-world use-cases of human action recognition, the use of depth enabled cameras or wearable sensors are not practical due to economic factors and the complexity of such specialized equipment.

Even though state-of-the-art image classification methods have surpassed human-level accuracy [8], the performance of methods based on the RGB video stream proposed in the literature for activity recognition is still unsatisfactory. One of the reasons behind this is that the multi-modal data provide a higher quantity of information, and the depth information provides precise detection of movement in the scene, which is not the case for RGB data. The usability issue of specialized equipment led surveillance systems to use simple RGB cameras. Therefore, a solution that efficiently leverages more widely available RGB videos to detect and classify human motion would benefit the real-life applications and the users.

1.3.2 Data can be Noisy

With better utilization of available data, we may increase the performance of a real-world AI system. Still, even these systems fall short of delivering reliable performance due to noisy training data. Large amounts of manually annotated data often pose a gridlock constraint towards the success of DNNs. Meta information based automated data collection has been explored as an alternative to manual annotation [23]. However, both types of data acquisition methods are susceptible to error and can introduce noise to the dataset, which results in performance degradation of deep models [24].

We observe that noise in training data often throw DNNs off the rail. Even though state-of-the-art DNNs are somewhat robust to a small amount of data noise, a high degree of noise may significantly disrupt learning. Noise in training data makes learning harder, and with time, large DNNs tend to overfit the data, learning the noise as patterns. This phenomena is defined as Memorization [25].

Noise in image or video domain can be of different types; random pixel noise, predominantly crafted noise to fool DNNs (also known as Adversarial Examples [26]), label noise while data collection, etc. In this work, we mainly focus on label noise in training data. This type of noise can be especially damaging for deep models, as feature learning in DNNs highly depends on associated labels for supervised learning. Noisy labels can easily confuse deep models and hamper the learning significantly. Other types of noise mentioned above are often introduced intentionally during training to improve the robustness of DNNs.

Being one of the major issues, training data noise has received a lot of attention from the research community. There are several methods proposed in the state-of-the-art (SOTA) literature to avoid noise during training. These methods can be categorized into two major types; filtering noise in the preprocessing step and handling noise during training [27]. The first type of solution tries to model the noise and filter it out before the training process, while the second type utilizes specialized loss to create a robust model that can sustain noisy data. However, existing solutions in the literature require significant modification to SOTA DNNs, making it difficult to deploy in practice.

1.3.3 Data can be Confusing

Besides having noise in the training data, the in-the-wild data samples can be problematic as well. In theory, researchers assume samples of the training set and test set are drawn from very similar if not the same data distribution, which is not a realistic assumption. In real world, data samples may come from a completely different distribution than the training phase. High performing DNNs do not possess the capability to differentiate between in and out of distribution data samples, resulting in erroneous predictions [28]. Often data samples from the edge of data distribution with affinity to other distributions cause confusion, but lack of proper detection capabilities force DNNs to make predictions on these confusing samples.

1.3.4 Data Annotation is Expensive

In most real world scenarios, available annotated data are very limited. Noise-free data annotation is expensive in terms of time and resources and this process is highly error-prone. More importantly, labeling life-critical data, e.g. medical data requires expert domain knowledge and high precision, which is even more resource-intensive. Hence, if the process of annotating large amount of data can be avoided, we can reduce cost significantly while reducing the risk of noisy annotation.

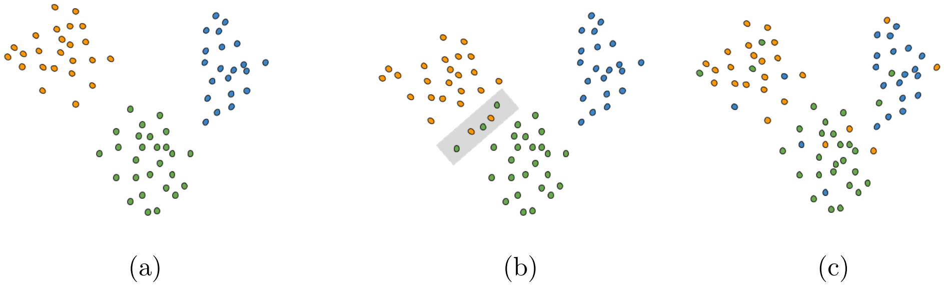


Figure 1.1 Dummy data distributions simulating different learning scenarios: (a) in an ideal scenario, (b) in presence of confusing samples that lie in the border of two distributions (highlighted in gray), and (c) in presence of label noise, where a fraction of samples are mislabeled.

In a data-driven learning paradigm, it is not feasible to learn all possible data distribution in an open world scenario, which marks a crucial issue with DNNs: Uncertainty or Risk, reducing the trust in and the verifiability of an AI system entrusted with critical decision making, e.g., healthcare systems, autonomous vehicle, secure authentication systems, etc. It is natural to make errors, even by a human, but we have a critical capability to differentiate between things we know and things we do not know. We often look for an expert's opinion when we are unsure about something. We go to doctors when we feel sick; we go to a mechanic when our vehicle breaks down. In short, we humans can always refer to experts when in confusion. However, even the most successful DNNs do not know what it does not know, forcing themselves to decide with the limited knowledge it has from the training. It is crucial for a reliable and robust system to differentiate between the known and the unknown.

We envision that a perfect AI system is not the one that has very high accuracy in a particular task. Rather a perfect AI system should be the one that may make wrong predictions due to the lack of knowledge or due to confusion, yet be able to identify such instances and refrain from making a decision. There exists no perfect system that can achieve optimum performance in a given task, but there should be systems that can identify where it can potentially fail to make the right decision.

1.4 Mitigation of Shortcomings of Data-Driven Learning

In the previous section, we presented different issues with data. Now, we aim to explore solutions to mitigate each of these issues.

1.4.1 Data Enhancing

In subsection 1.3.1, we have discussed how desired data can be unrealistic in real-world scenarios. Hence, extracting important features from available sub-optimal data is crucial. In the Human Activity Recognition scenario, we aim to optimize learning from more available RGB data instead of costly RGB-D data.

Traditionally, the works on RGB video stream are based on handcrafted features [29, 30, 31, 32, 33]. These approaches are highly data domain-dependent. Due to this problem, these methods are hard to deploy in real life despite the higher accuracy they achieve. With the advent of deep learning, methods were proposed where features could be automatically extracted [34, 35, 36]. Successful use of deep learning with image classification inspired researchers to deploy such methods in video classification [22]. These methods use raw RGB frames, often coupled with motion, to learn the temporal features. But the complex background and partial occlusion of subjects often cause these methods to perform poorly compared to the handcrafted solutions.

Approaches based on multiple modalities of data [3, 5, 7, 6], however, achieves higher accuracy even with complex actions. Skeleton information extracted from depth images is proven very efficient in extracting important features of action. Inspired by that, in this paper, we propose a technique that aims at separating salient features from the scene by extracting skeleton key-points rather than using raw frames. We propose to use Openpose API [37] as a black box to extract the skeleton key-points from each frame. These key-point features are then fed into a Bidirectional Long Short Term Memory (BLSTM) based model to learn the spatio-temporal representations, which are subsequently classified by a softmax classifier.

We use RGB-only modality for our experimental evaluations, whereas SOTA methods utilized multiple available modalities (RGB, depth, inertia, and skeleton data). This essentially reduces training data to one forth for our experiments compared to SOTA. Hence, we are dealing with one of the key challenges of deep learning, i.e., training with limited labeled data. To train the deep network effectively, we explore data augmentation and a few additional algorithmic approaches. Experiments on two popular and challenging benchmark datasets validate the effectiveness of these techniques and these approaches help to boost the performance of our RGB-only solution even higher than that of state-of-the-art, which all exploits RGB-D videos. We provide evidence that the proposed scheme is more cost-effective and highly competitive than RGB-D based solution and, therefore, widely deployable.

1.4.2 Data Denoising and Abstaining

Noise in training data and confusing data samples during inference often coincide in real-world scenarios. In our work, we aim to tackle both these issues in a unified manner.

For detecting and filtering label noise from data, recent works have utilized a scheme that reduces misclassification loss by incurring penalty while training the model [38, 1]. SelectiveNet [1] proposes a specialized rejection model that learns to reject any sample that produces high cross-entropy loss under the constraint of user-specified coverage. The authors show that with different training coverage, inference performance can be improved with corresponding calibrated coverage. DAC [38] proposed by Sunil Thulasidasan et al. utilizes a similar scheme that abstains hard to learn samples by learning the coverage constraint while training. After introducing artificial noise to data, they show that their method can identify that noise, and after filtering them, DNNs can achieve State-Of-The-Art (SOTA) performance. However, these prior works on detecting noisy samples do not evaluate their performance in a unified manner on training data noise detection and abstention of noisy test data. To the best of our knowledge, no unified framework is known to perform well on both tasks in an end-to-end manner.

To address this, we propose a simple yet effective framework, which can be applied to any SOTA models for both training and test data filtering without any alteration to the model architectures or loss functions. The high-level idea is to model the underlying data distribution in such a way that any sample lying outside known data distribution or sample that is equally distanced from any two or more distributions will be regarded as noise.

Deep models are proven to learn from dominant features at the beginning of the training process before memorization occurs [25, 39]. Hence, deep models, trained with best-practice choices to reduce overfitting, learn robust features even with the presence of noise. We assume that such pre-trained models can learn the underlying data distribution reliably but at the cost of a higher error rate. Under this assumption, we utilize the class-specific density of samples in the feature space to identify noisy samples during training and utilize this training data density to identify uncertain samples during inference. With empirical analysis, we observe a strong correlation between the distance of samples from data distribution and the noise associated with them.

We demonstrate our proposed framework's effectiveness in both tasks of denoising training data and test data abstention with widely used DNNs on benchmark datasets. Our proposed method outperforms the state-of-the-art method, DAC, on denoising training data. Moreover, we demonstrate the superior performance of our method over SelectiveNet on test data abstention given different coverage calibrations. Through visualization of data samples in feature space, we further justify the effectiveness of our proposed framework.

1.4.3 A Use-case of Data Abstention

As the main goal of this research is to bridge the gap between prototyping and real-world deployment, we seek to test the capabilities of our proposed framework on reducing risk and uncertainty of DNNs in life-critical applications. The use of DNNs has been immensely increasing in the medical imaging field [13], e.g., Computer-aided diagnosis (CAD) [40], medical image analysis, etc. Diagnostic tasks in medical imaging essentially require "learning from data" for a large and complex deep model. Even though CAD has been studied widely, the uncertainty of DNNs in medical imaging is remarkably understudied. Hence, we aim to provide a comprehensive study of existing methods for mitigating uncertainty in a life-critical decision-making scenario. There exist several medical imaging datasets, e.g., Lung Xray and Lung CT scan images for pneumonia detection, Brain MRI for Alzheimer disease detection, etc. The latest addition to these repositories is the COVID-19 lung Xray and ct scan dataset.

The COVID-19 caused by the SARS-CoV-2 virus has been declared a pandemic, and it is causing worldwide devastation and taking a heavy human toll [41]. The gold standard rtPCR test to detect COVID-19 suffers from complicated sample preparation, low detection efficiency, and high false-negative rate [42]. Often, critical COVID patients develop lung infection, and it can be visible through chest X-ray and CT scan [43]. A DNN enabled CAD system can potentially be utilized to diagnose COVID through medical image inspection and provide noninvasive detection solutions that would prevent medical personnel from contracting infections.

A lot of research is being conducted in this particular field, and some high performing models have been proposed [42]. COVID-Net [2] is such a model that achieves 94% positive predictive value (PPV) for detecting COVID positive Xray samples. Even though this is a very high accuracy, the predictive uncertainty problem still persists.

In this work, we evaluated two state-of-the-art methods from the literature that helps mitigate uncertainty by abstaining samples during inference; SelectiveNet, and our proposed method. We demonstrate that our proposed method outperforms the SelectiveNet in reducing uncertainty. We also propose a statistical testing based feature selection method to improve the abstaining framework to achieve higher PPV.

1.4.4 Self-Supervised Learning on a real-world usecase

To further investigate data driven learning and to reduce load on data annotations, we explored state-of-the-art self-supervised learning on real-world usecase, specifically, on COVID-19 detection from CXR images. As proven in the literature, proper representation learning can boost the accuracy and robustness of the DNNs. However, most of these successful models rely on a very large amount of labeled training data, i.e. ImageNet [8], Youtube8M [44], etc. Transfer Learning often helps in this regard, however, pretraining datasets often require to be of a similar domain to avoid the issue of domain-mismatch. While transfer learning has been proven successful across different domains, researchers mostly utilized annotated datasets from a similar domain, which is not always available or easily attainable for a variety of tasks. Self-supervised learning has proven to perform very similarly on several complex tasks with Supervised Learning, not relying on large annotated datasets [45, 46, 47, 48]. Self-supervised learning methods, hence, can reduce the cost of annotation by only using a fraction of labeled data to finetune on a specific task at hand.

1.5 Objectives

The final goal of this research work is to ensure data-efficient reliable learning of deep models. To break it down, the objective of this research is three folds.

- Efficient use of Data towards better performance.
- Denoising data to avoid distraction while learning.
- Mitigate risk and uncertainty of Deep models by abstaining from inferring.
- Minimize load on data annotation by utilizing self-supervised training.

As we discussed in the previous section, we have touched all three of our research objectives with the completed work. However, we strongly feel that there are a number of open research questions around our third objective. We only covered how to abstain confusing samples; however, there still remains the issue of tackling out-of-distribution samples and the adversarial samples. Though several methodologies exist to address these issues, similar to the other SOTA works, these methods require extensive modification to the model itself, making them less suitable for real-world deployment.

1.6 Motivations

From the dawn of AI, researchers have made significant progress on a wide variety of tasks. Yet a major issue persists when transferring models from controlled experimental settings to reality. Noisy environments, the wide gap between training and in-the-wild samples, data annotation bottleneck, and unreliability still bar DNNs to be efficiently deployed in the real world. A significant amount of researches are being conducted to address these issues. Yet, still, we have a long way to go before we can safely and with confidence deploy Deep Models in critical decision-making tasks. This is the driving force of this research. We aim to close this aforementioned gap with our research journey.

1.7 Challenges

While undergoing research to tackle the aforementioned issues, we have encountered several challenges, especially associated with data. In this section, we try to summarize the most critical challenges we faced during our research.

- Very limited amount of labeled data is readily available. Though extensive datasets are made available by researchers in some domains, such a large amount of data collection is very costly and error-prone.

- Large datasets often contain label noise. Moreover, as mostly these noises are unintentional, no information regarding them are present. The detection of such noise, thus, mandate human verification, which is expensive and time-consuming.

- Large DNNs require high computational power, and training them is a lengthy process. With limited resources, training large DNNs on extensive datasets were challenging and often infeasible.

- Medical data are often privacy protected and scarce.

- Annotating medical data with the highest precision is even costlier, hence, the scarcity.

- Several potential projects could not be completed due to limited resources and time. We have explored automated augmentation techniques that are learned from data itself. We also have explored facial expression recognition, small object detection in the heterogeneous field of view, and distress detection from video feeds. However, carrying out these works demanded resources that were scarce, resulting in them being abandoned prematurely. Competing with larger research teams with an abundance of resources is challenging, especially in the fast moving field of Deep Learning and Computer Vision.

1.8 Contributions

In this research journey, we have delved deep into different issues associated with data in Deep Learning. In the process, we have made core contributions to the data-driven learning paradigm. We list the contributions below.

1.8.1 Human Action Recognition from Limited labeled data

- We proposed methodologies to efficiently extract important features from RGB-only videos for human activity recognition.

- We leveraged data augmentation to tackle the problem of limited labeled data in deep learning and compensate for the data sparsity issue caused by using RGB-only modality.

- Additionally, we explore a few algorithmic approaches such as Dynamic Frame Dropout (DFD) and Gradient Injection (GI) to train the deep architecture effectively.

- We evaluate our proposed framework and demonstrate for the first time that using RGB-only streams we can surpass the state-of-the-arts RGB-D based solution and make our RGB-only solution widely deployable.

1.8.2 Unified Framework to handle Data Noise and Confusing data samples

- A novel approach to filter noise from both training and test data samples. We propose a density-driven approach for data denoising and abstaining. We introduce modality analysis and adaptive thresholding to differentiate between noise and clean data.

- End-to-end data filtering framework to improve deep models' reliability in a realistic noisy environment.

- Easy to incorporate in any real-world image classification applications as the framework works without modifying existing SOTA deep models.

- Through extensive experimentation and performance analysis, we demonstrate the proposed framework's performance benefit over existing SOTA methods.

1.8.3 Mitigating Uncertainty in Life-Critical applications of DNNs: COVID-19

- Investigation of uncertainty estimation methods to detect confusing cases on COVID diagnosis. To the best of our knowledge, we are the first to comprehensively study the uncertainty of CAD systems on COVID diagnoses.

- Utilization of feature filtering algorithm to augment uncertainty estimation framework in order to improve positive predictive value.

- Validation of the abstained samples by the best performing framework with medical professionals. Expert opinion on confusing samples abstained by the framework further validates the usability of the framework on screening COVID patients.

- Through extensive experimentation and performance analysis, we provide proof of efficacy of the SOTA uncertainty estimation methods on COVID-19 diagnosis.

1.8.4 Uncertainty Aware Self-Supervised Contrastive Learning towards Efficient Representation Learning on COVID-19 detection

- In order to understand the efficacy of siamese-based representation learning, we study SimSiam Network [48] on the real-world scenario of detecting CODIV-19 from chest x-ray images.

- Visualized gradient activation to evaluate the causality of representation learning by SimSiam network.

- Incorporated DbFF framework [28] to reduce the uncertainty of predictions by SimSiam network.

1.9 Thesis outline

The remainder of this thesis is organized as follows. Chapter 2 reviews existing approaches to Human Action Recognition, Data noise and uncertainty, and risk mitigation of DNNs and COVID diagnose. Chapters 3, 4, 5 and 6 present our proposed approaches for Human action recognition, data denoiser, COVID diagnosis and uncertainty aware self-supervised representation learning. Finally, Part 7 concludes the thesis and discusses directions for future work.

CHAPTER 2
Related Works

In this chapter, we explore some state-of-the-art literature that are relevant to our research. First, we look into the literature relating to Human Action Recognition. Then, we go through how researchers have proposed different methodologies to combat noise in data and handle confusing data during inference. We look into how deep learning is being utilized in Medical Image analysis for disease detection, especially, detection of COVID-19 from xray and CT scan images. Lastly, we explore self-supervised learning and siamese networks. These literature reviewing assisted us not only to better understand the problem we hope to tackle, but also to deliver more sophisticated solutions for them.

2.1 HAR

Human activity recognition has been extensively studied in the recent years [21, 22, 49]. Most of state-of-the-art methods trying to solve this problem using RGB data are based on handcrafted feature [29, 30, 31, 32, 33]. Schuldt et al. [29] present a method that identifies spatio-temporal interest points and classifies action by using SVMs. Zhang et al. [30] introduce the concept of motion context to capture spatio-temporal structure. Liu and Shah [31] considered correlation among features. Bregonzio et al. [32] proposed to calculate the difference between subsequent frames to estimate the focus of attention. These methods often achieve very high accuracy, however, hand-crafted features are highly data dependent, hence not viable in real world.

Baccouche et al. [34] propose to use deep learning based Convolutional Neural Network

(CNN) to extract spatial features and then use LSTM to learn the temporal features. Ji et al. [35] present 3D CNN to classify actions which learns inherent temporal structure among the consecutive frames. A two-stream CNN based method is proposed in [36]. These methods, however, fail to achieve higher accuracy primarily because of the raw RGB frames being used as input. Higher level feature vectors from CNN feature extractor fails to capture sharp changes due to deep convolution and pooling.

Skeleton information from RGB-D video is being widely studied to improve recognition accuracy. Liu et al. [50] propose a CNN based approach leveraging the skeleton data. In [51] the authors propose hierarchical bidirectional Recurrent Neural Network (RNN) to classify the human actions. Inspired from this work, we adopt bidirectional LSTM in our method. However, we have not used skeleton data hierarchically and we extract skeleton keypoints from RGB frames unlike [51], where the skeleton data is extracted from depth information. Methods proposed in [6] and [5] utilize skeleton data on three CNN streams that are pretrained on large ImageNet Dataset [52]. Li et al. [7] use view invariant features from skeleton data to improve over [6] and [5], and they used similar four stream pretrained models. All these methods utilize skeleton data, either extracted from depth data or kinect.

Evidently, methods leveraging skeleton data, extracted from depth information, edges over methods that simply take raw frames as input. This inspired us to look deeper into how skeleton extracted from RGB frames can be utilized to robustly classify actions. To the best of our knowledge, we are the first to leverage skeleton key-points extracted from RGB-only videos for human activity recognition. Although there exist a few CNN and LSTM based approaches for activity recognition, none address the issue of redundant information from raw frames. We also emphasize more on algorithmic approaches to address the training issues of deep networks, such as limited training data, overfitting and gradient vanishing. Enhanced by these techniques, our RGB-only solution is able to surpass the state-of-the-arts that all exploit RGB-D streams.

2.2 Denoising and Abstention

2.2.1 Selective Prediction

Classification with a reject option has been explored by researchers to tackle the prediction uncertainty of deep models [53, 54, 55, 56, 57]. One idea of implementing selective prediction is to define a threshold on posterior probabilities [54, 55]. Various SVMs-style variants have been developed to incorporate reject option with classification tasks [56, 57].

Another more recent trend in this domain is to learn the prediction and the selection parameter jointly [1]. SelectiveNet [1] proposes a user-defined coverage constraint to learn to abstain samples with high classification loss. By minimizing overall loss, the model learns to abstain from test samples that are difficult to predict. However, in order to set a coverage constraint for the model, users need to have information about the magnitude of noise present in training data, which is infeasible in a real-world scenario. The authors have not demonstrated the effect of noise in training data, which will throw the whole system off the rail, as the model will learn to abstain noisy samples from the training data and potentially misclassify the test samples. One more issue we observed with SelectiveNet is that through the auxiliary head, the system is learning from all examples during the training process even with the presence of noise; this makes the method unsuitable to handle noisy training data. The proposed model requires heavy modifications to existing models and loss functions, which increases overhead.

2.2.2 Label Noise Abstention

Label noise in training data has received less attention than the selective prediction problem. Yet there are a number of interesting works proposed in the literature to tackle this problem. The authors of [58] have proposed to use two-stream DNN that jointly learns from a large noisy dataset and a small clean dataset. In [59], the authors first train an ensemble of classifiers on data with noisy labels using cross-validation and then the predictions from the ensemble are used as soft labels to train the final classifier. DAC [38] introduces a more light-weight solution to handle label noise. Unlike SelectiveNet [1], DAC proposes to automated learning of noise level while training and use abstention class to determine if a training sample is abstained or not. The authors have empirically shown that adding artifact (smudge) to images results in abstention. However, that might lead to misclassify samples with similar occlusion pattern as noisy, even though the features of the point of interest is still prominent. This might lead to degraded performance with adversarial examples. The authors also have not demonstrated how their proposed model performs when test data were abstained in the presence of label noise.

2.2.3 Out-of-Distribution and Adversarial Example Detection

Out of Distribution detection is another aspect of detecting noise in test data which attracted a lot of attention in recent years [60, 61, 62, 63]. ODIN [61] and its variants [60, 62] are proven to be very successful in detecting OOD samples. One of the common themes of these methods is the input preprocessing step: adding adversarial noise to test data to increase the difference between in and out-of-distribution data. The authors of [60] have proposed a similar framework to ODIN by adding Gaussian discriminant analysis of samples. They empirically show that Mahalanobis distance can be effective in detecting OOD samples. We adopt the use of distance in detecting noisy data samples, but with key differences from them. For example, we do not employ input preprocessing and we introduce the concept of automated thresholding of distance to differentiate between the noisy and clean examples. However, OOD and adversarial sample detection are out of the scope of the current work. We will address these issues in the future.

2.3 Uncertainty on Computer-Aided Diagnosis of COVID-19

2.3.1 CAD on COVID-19 Detection

Deep Learning powered Artificial Intelligence (AI) has been widely used in different aspects of healthcare. Detection and diagnosis of COVID-19 are also not exceptions to that. COVID-19 medical image analysis mainly involves examining Chest X-ray (CXR) and Lung CT imaging. However, compared with CT images, CXR images are easier to obtain and cost effective [42]. Hence, in this work we focus on COVID-19 CXR image data.

There are several research that aims to make diagnosis of COVID-19 based on CXR images [42], that utilizes ensemble of multiple high performing DNNs [2, 64]. Though, they achieve satisfactory performance, these methods require extensive training procedure for each of the DNNs and computationally expensive. Another school of researchers propose specialized DNNs for COVID classification [2, 65]. COVID-Net [2] utilizes a projection-expansion-projection design pattern along with human-machine collaboration. The authors also publish COVIDx dataset, which is the largest publicly available COVID-19 dataset. Although a large number of methods have been proposed in the literature for detection of COVID-19, to the best of our knowledge, none of them address the uncertainty issue of their proposed method.

2.3.2 Uncertainty Mitigation on Benchmark Datasets

Classification with a reject option has been explored by researchers to tackle the prediction uncertainty of deep models [53, 55, 56]. Threshold on posterior probabilities [54, 55] and various SVMs-style variants have been developed to incorporate reject option with classification tasks [56, 57]. Though these methods are widely used in practice because of their simplicity, they have a major drawback. Research shows that DNNs often make wrong predictions with a very high confidence [66]. Another school of researchers propose methods to jointly learn rejection and classification from data itself. SelectiveNet [1] proposes a user-defined coverage constraint to learn to abstain samples with high classification loss. By minimizing overall loss, the model learns to abstain from test samples that are difficult to predict. However, this method is more complex and require extensive repetitive training, which makes it difficult to deploy in practice.

2.3.3 Uncertainty Mitigation on CAD

Uncertainty mitigation on CAD is greatly understudied in the literature. Christian Leibig et. al. [67] proposed one of the first uncertainty estimation methods utilizing stochastic Monte-Carlo Dropout (MCDO) during testing to approximate the aleatoric uncertainty of Bayesian CNN. However, the estimated uncertainty of correct and wrong predictions from their proposed method overlap significantly. In [66] Mirat et. al. proposed an intuitive framework based on test-time augmentation for quantifying the diagnostic uncertainty of Bayesian CNNs. However, even though Bayesian statistics provide simpler ways to estimate the uncertainty, these methods are intractable in most of the real-world scenarios. A very few works exist on handling uncertainty in medical image segmentation [68]. The authors propose uncertainty aware training to learn abstaining confusing segmentation on a data sample.

2.4 Self Supervised Siamese Networks

2.4.1 Self Supervised Learning

Self supervised learning refers to learning methods that are trained with supervisory signals that are generated from the data itself by leveraging its structure. Self supervised learning methodologies can be categorized into Generative and Discriminative approaches. Generative methods learn visual features through the process of image generation. This type of methods includes image super resolution [69], image inpainting [70], image generation with

Generative Adversarial Networks (GANs) [71, 72], etc. However, pixel-level generation is computationally expensive and this may not actively boost representation learning. On the other hand, discriminative approaches learn representations using specific objective function, but the labels are derived from an unlabeled dataset. Many discriminative approaches utilize heuristics to design the training flow [73, 74, 75], which could limit the generalization of the models. Contrastive learning, another discriminative approach, has been proven successful recently, achieving state-of-the-art results [46, 47, 45, 48].

2.4.2 Siamese networks

Siamese networks [76] compare inputs with identical backbone networks. These networks are extensively explored in signature [76] and face [77] verification, one-shot learning [78], and tracking [79]. Recent advancements in self supervised learning are based on siamese based networks [46, 47, 45, 48]. However, contrastive learning base siamese networks often require very large number of negative examples and require especial techniques for successful training, e.g. momentum, memory bank, etc. [46, 47, 45]. Authors in [48] provide proof that none of these especial techniques are required to train a siamese network though they may help boost the performance. They propose to use simple siamese network with stop gradient instead and show that their proposed method is simple and does not require large batch size as other methods do.

CHAPTER 3

Towards Robust Human Activity Recognition fromRGB Video Stream with Limited Labeled Data

Human activity recognition based on video streams has received numerous attentions in recent years. Due to lack of depth information, RGB video based activity recognition performs poorly compared to RGB-D video based solutions. On the other hand, acquiring depth information, inertia etc. is costly and requires special equipment, whereas RGB video streams are available in ordinary cameras. Hence, our goal is to investigate whether similar or even higher accuracy can be achieved with RGB-only modality. In this regard, we propose a novel framework that couples skeleton data extracted from RGB video and deep Bidirectional Long Short Term Memory (BLSTM) model for activity recognition. The biggest challenge of training such a deep network is the limited labeled training data, and exploring RGB-only stream significantly exaggerates the difficulty. We therefore propose a set of techniques to train this model effectively, e.g., data augmentation, video frame dropout and gradient injection. The experiments demonstrate that our RGB-only solution surpasses the state-of-the-arts, all exploit RGB-D video streams, by a notable margin.

3.1 Methodology

In this section, we present a novel end-to-end framework for human activity recognition from RGB video containing human silhouette. We review some important concepts in the following subsections, that are used in our proposed methodology, to make it self-contained.

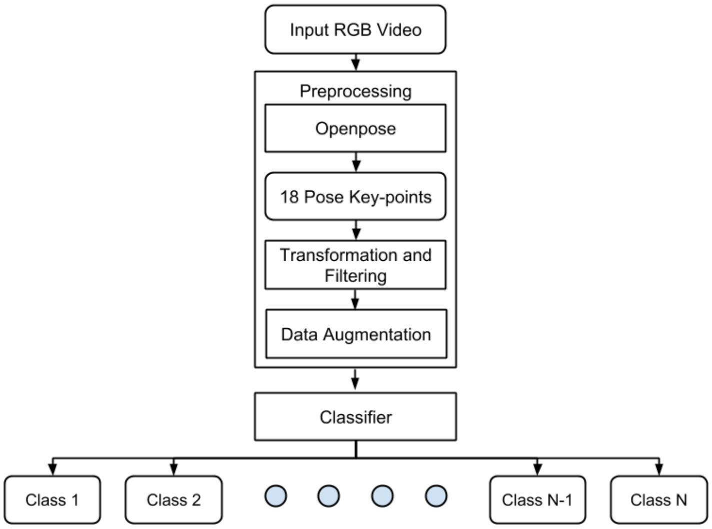


Figure 3.1 Overview of proposed method.

3.1.1 Overview

Our proposed architecture aims to detect, extract and classify human actions from RGB-only data. We formulate our problem as learning the mapping, $\mathbf{F} : x \rightarrow \ell$, where x is the raw video and ℓ is the collection of action label. After learning, $\mathbf{F}$ is used to classify the test samples.

Fig. 3.1 shows the overall pipeline of our proposed method. First, we extract pose key-points of human silhouette from input raw RGB video using Openpose API [37]. We perform preprocessing on the extracted pose key-points. After preprocessing, we use Data Augmentation on the extracted keypoints to mitigate the problem of data scarcity. Then we feed the key-points into our classifier. We used deep BLSTM [80] network coupled with MLP [81] as our classifier. Overfitting is a major drawback for LSTM when dealing with small dataset. Therefore, in addition to data augmentation, we deployed Dropout and L2 Regularization to introduce stochasticity, which prevent our model from overfitting. We

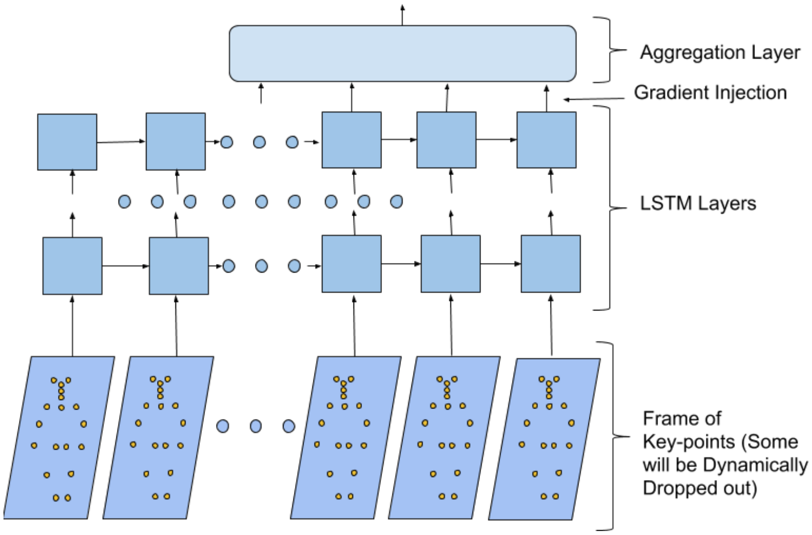


Figure 3.2 High level concept of Gradient Injection.

propose Dynamic Frame Dropout to reduce the redundant frames and improve the robustness of the BLSTM classifier. We also introduce Gradient Injection to improve gradient flow to mitigate the vanishing gradient problem. Overview of Gradient Injection is presented in fig. 3.2. We will discuss each of these components in details in later subsections.

3.1.2 Openpose

Openpose [37] is an opensource API, providing applications that can be used to detect the 2D poses of multiple human subjects in an image. The API leverages a novel two stream multi-stage CNN, which facilitates it to work on real time. The methodology proposed in [37] was ranked number one in COCO 2016 keypoints challenge. The input of the architecture is raw RGB image and the output of the system is 15 or 18 pose key-points along with the part joining edges. More details about the architecture and working principle can be found in [37]. In our work, we treat Openpose as a black box with raw video frames as inputs and

18 pose key-points per person as output (fig. 3.3 shows 15 key-points that were detected by openpose with higher confidence, remaining three key-points were excluded).

3.1.3 LSTM

Long Short-Term Memory (LSTM) [80] is a descendant of Recurrent Neural Network (RNN) especially designed to adapt long range dependencies when modeling sequential data. RNN, in general, has been proven very successful in modeling sequences that has strong temporal dependency. However vanishing gradient problem makes Vanilla RNN hard to train [82]. LSTM solves the problem by introducing non-linear gates regulating the information flow. However, Vanilla LSTM can only learn from past contexts, whereas Bidirectional RNN (BRNN) [83] can learn both past and future by utilizing feed forward and backward layers. Bidirectional LSTM (BLSTM) network can be obtained just by replacing the BRNN nodes with LSTM. BLSTM can efficiently incorporate long term dependency in both directions

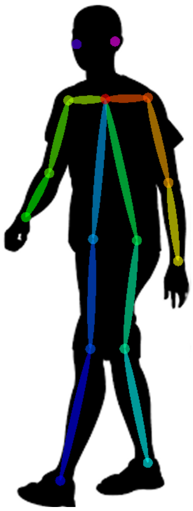


Figure 3.3 Output of Openpose: Rendered pose on silhouette.

which helps improve learning of temporal data.

3.1.4 Preprocessing

The preprocessing step represents the first step of our end-to-end pipeline where the raw video frames are fed into the Openpose API. The output of Openpose for each video frame is a matrix of shape $(n_{pose}, (a, b), c)$. Here, n_{pose} is the number of pose key-points, (a, b) is the coordinates of the key-points in Cartesian plane and c is the confidence score of the respective key-point. To simplify our problem, we put a constraint that each frame can contain at most one person, hence, the value of n_{pose} here is 18. When all pose key-points are extracted from a video, we use a filter to set the pose keypoints values that has confidence lower than a threshold value, Θ, to zero. Later, we mask these zero valued keypoints in order to avoid learning from these points as well as to prevent gradient calculation. Afterwards, the pose matrix is flattened and converted into a vector, Λ, of size $n_{pose} * 2$, excluding the confidence value. We concatenate each pose frame into a 2 dimensional matrix of shape (n_{frame}, v), where n_{frame} is the number of frames in the video and v is the length of pose vector, Λ.

3.1.5 Dynamic Frame Dropping

We propose to utilize Dynamic Frame Dropout (DFD) to reduce data redundancy. As different actions require different time span, and often there are redundant informations in consecutive frames, taking all frames into account actually occludes crucial information and hampers the learning. Techniques like randomly dropping frames or dropping each n frames etc. are often used in state-of-the-art methods to avoid redundancy. However, doing so may result in loss of important information. Hence, Dynamic Frame Dropout (DFD) based on information redundancy is a more sensible solution. Moreover, DFD helps data normalization which introduces stochasticity in data.

Pairwise euclidean distance, d, between key-points of two consecutive frames indicates how different these frames are; lower distance corresponds to similarity and higher distance means these frames actually have meaningful differences. Empirically, we set a cutoff threshold, $\hat{c} = 15$. If d is distance between $frame_1$ and $frame_2$ and $d < \hat{c}$, then we drop $frame_2$. This setup of $\hat{c}$ drops 20 to 25 frames per video that carries information with minimal significance.

3.1.6 Data Augmentation

Training a deep networks with limited amount of labeled training data is a major challenge in supervised learning paradigm. Our goal of achieving state-of-the-art performance with RGB-only data modality faces the same brick wall: insufficient training data. According

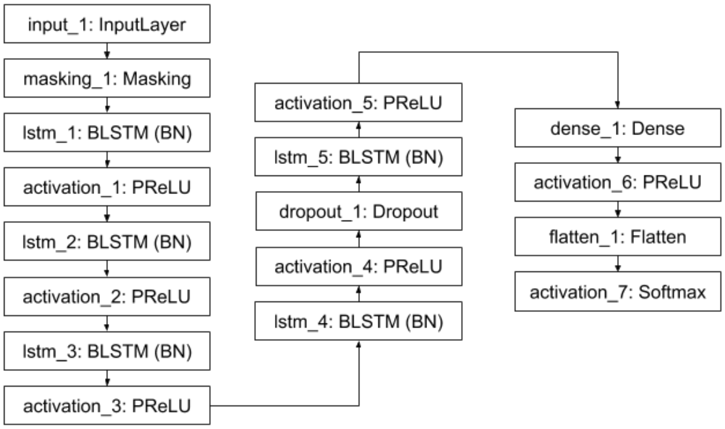


Figure 3.4 Proposed BLSTM network architecture.

to our problem formulation, we only leverage RGB data modality. Data augmentation has been proven very successful in supervised learning for image analysis. Inspired by this, we have explored several data augmentation techniques to solve the data scarcity problem. Translation, scaling and random noise are used to augment data, equation (3.1), (3.2) and (3.3) represent these augmentation techniques.

$$X' = X + b_{translate} \tag{3.1}$$

$$X' = A_{scale} * X \tag{3.2}$$

$$X' = X + b_{rand} \tag{3.3}$$

Here, X and X' are input data and new manipulated data. $b_{translate}$ and A_{scale} are constant translate and scale factor, which are tunable hyper-parameters. b_{rand} is a random sample drawn from normal distribution with mean, μ, and standard deviation, σ, which again are tunable hyper-parameters. We can also combine translation and scaling together to generate data variation in a more generic way as shown in Eq. (3.4). Our experimental results 3.2 reflect the significance of data augmentation for training deep networks using limited training data.

$$X' = A_{scale} * X + b_{translate} \tag{3.4}$$

3.1.7 Proposed Network Architecture

Our proposed deep architecture combines deep BLSTM layers and MLP (Fig. 3.4). We use five consecutive BLSTM layers with dropout layers to regularize the model training. We utilize Batch Normalization (BN) after each BLSTM layer to keep the data normalized throughout the pipeline. We feed the output of the Deep BLSTM layers to the MLP consisting of two Dense layers. For intermediate hidden BLSTM and Dense layers, we have utilized the Parametric Rectified Linear Unit (PReLU) [84] activation layer. We used softmax activation function for the final output layer to produce probabilistic score for each class. Categorical cross-entropy is used to measure the loss of our proposed network. We utilized RMSprop optimizer [85] to minimize the loss function.

3.1.8 Gradient Injection

Although LSTM serves as the solution of vanilla RNN for gradient vanishing problem, it itself faces this issue in some degree when training deep model [86]. LSTM many to one architecture is often used as the final layer of network for video classification. This creates a bottleneck dependency on the whole video sequence, but often a video can be clearly classified before having to see all the frames till the end. Hence, to avoid gradient vanishing problem and to reduce dependency, we propose to use Gradient Injection (GI) technique. In concrete terms, we utilize many to many architecture of LSTM at the top layer to allow gradients flow from multiple time steps, consequently, reducing the problem of vanishing gradient. Moreover, as outputs from multiple time steps are now available, it creates an

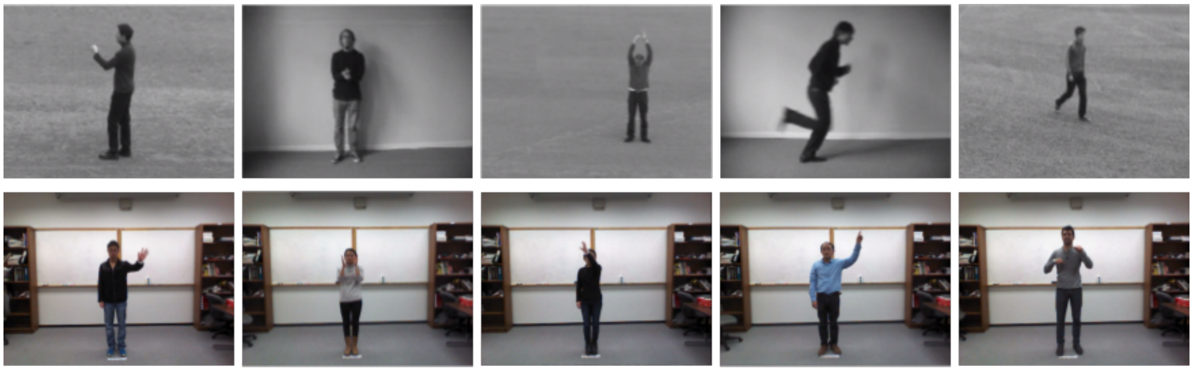


Figure 3.5 Sample frames from KTH (top row) and UTD-MHAD (bottom row) datasets.

ensemble of multiple outputs and reduces dependency on all the video frames. The high level scheme of this concept is presented in fig. 3.2.

3.2 Experimental Results

The principle goal of this paper is to show that by only using RGB data modality with limited training data, we can achieve similar or higher accuracy on action recognition task than the state-of-the-arts that use RGB-D video streams. We have tested our proposed method with two widely used datasets, KTH [29] and UTD-MHAD [3]. We focus on UTD-MHAD as this is a complex dataset offering multiple modalities and current state-of-the-art methods utilize data modalities consisting depth information to classify actions. Empirically, we show that with data augmentation combined with Dynamic Frame Dropout and Gradient injection, our proposed method surpasses state-of-the-art works. We also show that our proposed method performs better with RGB-only dataset such as KTH, compared to current literature.

We implemented our system in Python with Tensorflow backend on a GPU cluster with Intel(R) Xeon(R) CPU E5-2667 v4 @ 3.20GHz with 504 GB of RAM and NVIDIA TI-

TAN Xp with 12 GB of RAM and 3840 cuda cores. In our experiments, we empirically set learning rate, $lr = 0.00005$ for RMSprop optimizer. We report confidence interval based on 50 bootstrap trials. More details about datasets we evaluated our model on and comparative experimental studies with state-of-the-art literatures are presented in the following subsections.

3.2.1 Dataset

KTH [29] is an RGB-only benchmark action dataset containing six action classes (walking, running, boxing, hand-waving, and hand-clapping), performed by 25 subjects in various conditions. KTH dataset provides full silhouette figure in all the sequences, which satisfies our constraint. We have followed the original experimental setup stated in [29].

UTD-MHAD [3] is a multi-modal action dataset containing 27 actions performed by 8 subjects (4 males and 4 females) performing same action 4 times, a total 861 sequences. This dataset provides four temporally synchronized data modalities; RGB videos, depth videos, skeleton positions, and inertial signals from Kinect camera and a wearable inertial sensor. We follow 50-50 train-test split similar to [3]. In the experiments we are only using RGB modality to evaluate our proposed approach. Fig. 3.5 visualizes some example frames from KTH and UTD-MHAD datasets in the first and second row respectively.

3.2.2 Comparative Results on KTH dataset

Comparative results of our proposed method on KTH dataset with the state-of-the-arts are presented in Fig. 3.6. CNN based hybrid model proposed by Lei et al. [87] achieves 91.41%

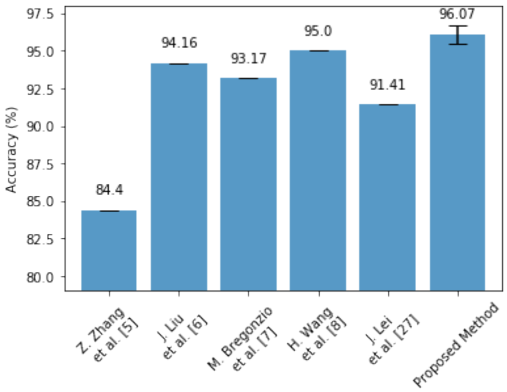


Figure 3.6 Accuracy comparison on KTH dataset with state-of-the-arts (confidence interval of our method is also shown above).

accuracy. Other compared methods [33],[88],[30], and [32] use hand-crafted features. These methods achieve high accuracy, but are extremely data dependent, hence not suitable for real world deployment. Our proposed method with data augmentation and dynamic frame dropout achieves 96.07% accuracy, outperforming all the others.

3.2.3 Comparative Results on UTD-MHAD dataset

We begin our experiments on UTD-MHAD dataset using our baseline model which takes dynamic length video as input, but without Dynamic Frame Dropout (DFD), Gradient Injection or Data Augmentation (DA). We then update our model by adding these features in a cumulative fashion. In other words, the second model includes DFD; the third one includes both DFD and GI; in the fourth model we use random jittering to augment data, and finally in the fifth and last model we use affine transformation as data augmentation. Fig. 3.7 shows the comparison among all these models on accuracy and F1 score. As can be seen, by using DFD on baseline, we surpass state-of-the-art accuracy (89.06%). An interesting phenomena to observe here is that, although Gradient Injection does not have much effect on larger training data, it helps gaining performance over DFD. Utilizing data augmentation we gain 2% accuracy over DFD model. However, using random jittering to augment data does not improve accuracy.

Table 3.1 Effect of Data Augmentation on the UTD-MHAD dataset.

Augment Size	Top-1 Error (%)	Top-3 Error (%)
0	10.94	4.01
430	9.75	3.75
860	9.35	3.68
1290	9.09	3.69
1720	9.05	3.66

To investigate the effect of data augmentation on the predictive accuracy, we experimented with incremental data augmentation. The results are summarized in Table 3.1. As can be seen, data augmentation significantly helps model to regularize when we essentially doubled the training data. Afterwards, the effect of data augmentation is less impressive yet every iteration shows downwards trend on error. Another phenomena we observed is that data augmentation did not have much affect on top-3 categorical accuracy, which means that data augmentation mainly boosts correct answers from top-3 positions to top-1 position.

We also have explored our choices of depth of the network. We tested our baseline model with three settings: 3 Layer, 5 Layer and 7 Layer model. Fig. 3.8 presents the accuracy of these models on top 1, 3 and 5 categories. The performance gain of the 5 layer model can be observed here. Notice that the 3 layer model has shown comparative accuracy on top 3 and 5 categories with other two models, and this indicates that deeper models mainly boost the top-1 accuracy.

Finally, we present the comparison results of our proposed method with state-of-the-

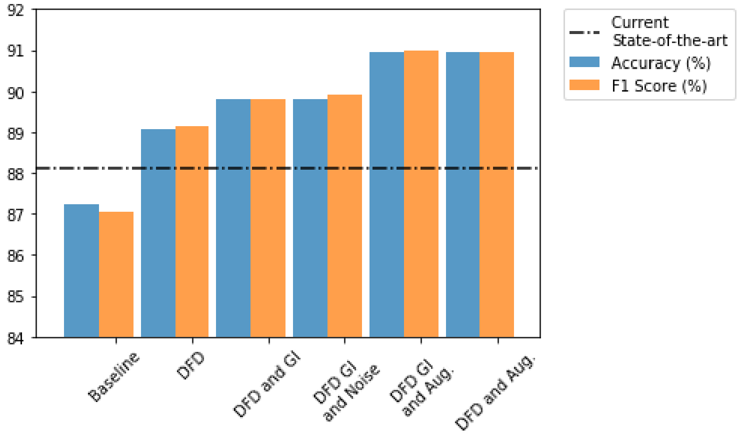


Figure 3.7 Accuracy comparison of the different design choices on the UTD-MHAD dataset.

arts (Fig. 3.9). Most of these existing methods [3, 5, 6, 7, 4], evaluated using UTD-MHAD dataset, use depth or inertia data modalities or both (section 2.1). These data modalities are only available from depth enabled camera and provide more precise information of motions related to actions. On the contrary, we use RGB modality only to train our model from scratch. As presented in fig. 3.9, our method achieves 90.95% accuracy which outperforms the state-of-the-art methods.

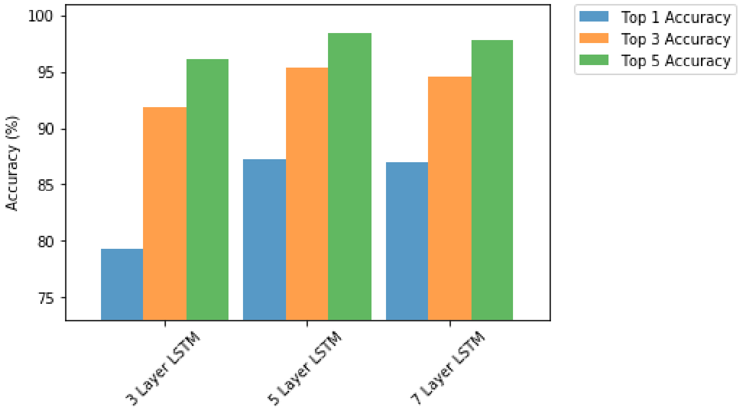


Figure 3.8 Accuracy comparison on the UTD-MHAD dataset on our models with different number of LSTM layers.

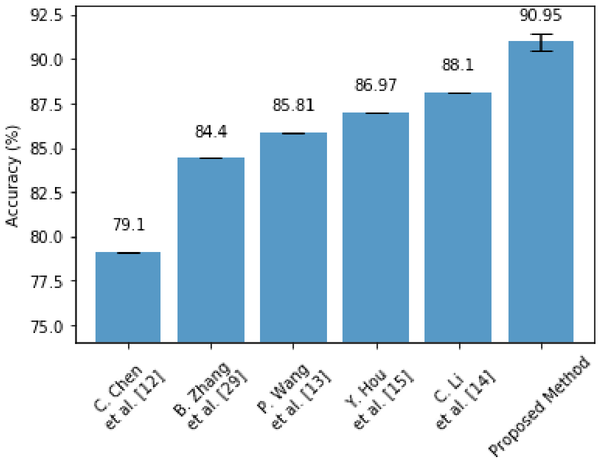


Figure 3.9 Accuracy comparison on the UTD-MHAD dataset. [3],[4],[5],[6] and [7] use depth enabled modalities, while our method use RGB-only modality (confidence interval of our method is also included).

3.3 Discussion and Future Work

The state-of-the-art methods achieve accuracy as high as 88.1%, utilizing depth data modalities on the UTD-MHAD dataset. However, RGB is the only data-modality we leverage. Our proposed system successfully achieves significantly better performances with data augmentation. Moreover, using 5-fold cross validation (80-20 train-test split) we could achieve even better performance (above 94%). This essentially proves that even with RGB-only data modality, data augmentation is sufficient to mitigate the problem of data sparsity and we were successful to train our model with augmented data and achieve better accuracy.

However, our experiments were conducted on the KTH and UTD-MHAD datasets, where there is only one person present per action and whole silhouette is visible. Additionally, these datasets were collected in a more controlled environment which makes them less realistic. To further strengthen our claim, as a future work, we will extend our method for multi-person datasets where silhouette is not a constraint.

3.4 Conclusion

In this paper we proposed an end-to-end framework that utilizes Openpose, a library for extracting skeleton data from RGB video streams. We deployed BLSTM network coupled with MLP for action recognition from the extracted skeleton data. We applied a number of algorithmic techniques like Dynamic Frame Dropout, Gradient Injection and Data Augmentation to train our framework effectively. Through extensive experimental evaluation, we demonstrate that data augmentation can be a major regularizer tool for training deep networks. Our experimental results indicate the superiority of our RGB-only solution over the state-of-the-art methods that all exploit RGB-D streams. This makes our solution cost effective and widely deployable.

CHAPTER 4

A Unified Density-Driven Framework for Effective Data Denoising and Robust Abstention

Success of Deep Neural Networks (DNNs) highly depends on data quality. Moreover, predictive uncertainty makes high performing DNNs risky for real-world deployment. In this paper, we aim to address these two issues by proposing a unified filtering framework leveraging underlying data density, that can effectively denoise training data as well as avoid predicting uncertain test data points. Our proposed framework leverages underlying data distribution to differentiate between noise and clean data samples without requiring any modification to existing DNN architectures or loss functions. Extensive experiments on multiple image classification datasets and multiple CNN architectures demonstrate that our simple yet effective framework can outperform the state-of-the-art techniques in denoising training data and abstaining uncertain test data.

4.1 Methodology

Given a DNN architecture, we propose a simple yet effective framework for detecting noise in data. First, we present our intuition behind the core of the proposed framework. Then we define the algorithmic steps of the framework in details.

We present an analogy of human social behavior to explain our intuition of noise in data. A group of people who share common interests are more likely to spend more time together or do more interaction with each other. Conversely, a group of people who do not have common interests are less likely to be together. Similarly, data samples that share dominant features are more likely to belong to the same cluster or class and data samples that project contrasting features are less likely to belong to the same distribution. Based on this hypothesis, we construct a framework to differentiate noise from data.

Hypothesis: *Samples that are away from distribution are potentially noisy or mislabeled.*

DNN learns high-level features from data samples during training. In the case of supervised learning, these features follow class constraints provided by labels. These features are high dimensional vectors that represent each data sample. To simplify, let us consider a dummy dataset where each sample only consists of two features. In an ideal world, samples of each class would be clearly separated and all samples of a class would be clustered together. We define these two ideal situations as inter-class diversity and intra-class affinity respectively. However, in real-world data, there exist inter-class affinity and intra-class diversity often due to errors in labeling or noise in data samples.

Let us consider a training set consisting of input-target pairs, $D = (x_i, y_i)_{i=1}^{N}$, where $x_i \in \mathbb{R}^n$ belongs to one of the $k \in L = \{l_1, l_2, \cdots, l_k\}$ classes. Note that in this paper we will state "class" and "cluster" interchangeably, where both of them are semantically similar. A DNN classifier consists of a feature extractor and a classifier. Feature extractor is a parameterized function $f_\theta : \mathbb{R}^n \rightarrow \mathbb{R}^\delta$ that learns to map n dimensional observed data x_i to feature space v_{x_i} of δ dimensions under the $l_j \in L$ class constraint. Classifier is a simple mapping function, $f_c : f_\theta(x_i) \rightarrow y_i$, which can be a softmax classifier.

Typically, parameter θ is optimized using the off-the-shelf cross-entropy loss. In an ideal scenario, f_θ thus learns to cluster semantically-similar inputs x_i to k different clusters corresponding to k classes. For any given new sample s, f_θ maps it to a feature vector v_s, which ideally should lie under any of k distributions observed during training. However, in real-world data, there might be $p \geq k$ clusters formed by observed data samples of k classes. Our goal here is to define each of these k data distributions robustly, such that even with presence of noise, definitions of these distributions hold. We propose to utilize density based clustering to identify which of these p clusters actually represent k classes and then we calculate centroids to represent these k clusters.

4.1.0.1 Density-based Clustering

To identify core classes from p clusters, we propose to use DBSCAN clustering [89], on feature space. Let us assume there are N_j samples that are bounded by the same class constraint l_j. Feature vectors $v_{x_i^j}$ extracted by f_θ are utilized with $DBSCAN$ algorithm to identify hidden clusters within class l_j. Any sample that is not affine to $MinPts$ number of density-reachable samples are treated as noise, and samples that are affine to at least $MinPts$ number of samples form a cluster. Though, there can be multiple clusters detected by $DBSCAN$ within a given class constraint, we define the cluster with the highest samples as the core cluster. The rational behind is that only the most populous cluster can be representative enough of a particular class. Other clusters with less samples may potentially be label noise occurred during data acquisition. However, in scenarios where there is no label noise presents, we expect to see a single cluster from the $DBSCAN$ algorithm. Identifying the most representative density distribution is crucial for our proposed framework, as we will define this cluster as the reference point.

4.1.0.2 Calculation of Centroid

We calculate the centroid of each class constrained core cluster yielded from $DBSCAN$ by calculating the median of feature vectors v_x extracted from trained DNN for sample x. Let us assume there are m samples in class j, and $DBSCAN$ returns a core cluster with m_{core} samples, where $m_{core} \leq m$. Then the centroid is defined as

$$c_j = \text{median}([v_{x_i^j}]_{i=0}^{m_{core}}), \quad \forall x_i^j \in l_j. \tag{4.1}$$

We collectively denote all k identified centroids as $C = \{c_1, c_2, \cdots, c_k\}$.

We take the approach of refining the data in multiple stages. Broadly, this can be divided into two stages. Firstly, we conduct denoising training data by utilizing a pretrained model and then we dive into abstaining from inferring noisy or confusing test data in inference time.

4.1.1 Denoising Training Data

In the first stage, we calculate distance between data samples and observed distributions and filter based on the derived distances. The first stage can be further granulated into five steps.

Step 1: We first train a DNN model with given training data (noisy or clean) with regularization. As demonstrated in [25], deep models learn from dominant features at the beginning of training. We confirm that claim empirically by training models with smaller number of epochs before memorizing starts. We also follow “best practice” to reduce overfitting.

Step 2: We employ $DBSCAN$ on m samples belong to each class to identify the core cluster with m_{core} samples. Then we calculate centroids for each class using Eq. 6.4.

Step 3: We calculate the distance $d^j_{x_i}$ between the feature vectors $v_{x^j_i}$ with label l_j and the corresponding centroid $c_j \in C$. In this step, we consider all m samples that belong to class j. We choose to use euclidean distance as our distance measure.

$$d^j_{x_i} = \text{euclid}(v_{x^j_i}, c_j) \tag{4.2}$$

Step 4: We propose a methodology to denoise any outliers by multimodality analysis, which will be discussed in details in Section 4.1.3.

Step 5: We train the model from scratch with denoised data we derived from the previous step.

4.1.2 Abstain from Inferring on Test Data

Second stage of our proposed framework takes place during the inference. At this stage, we already have a trained model on *cleaned* data. The second stage can be further divided into four steps.

Step 1: We calculate distance between all test samples and training data distributions under constraint of k classes. For $s \in S$, where S is the set of in-the-wild test samples, we calculate distance d_s between s and all $c_j \in C$ we derived in the previous stage:

$$d^j_s = \text{euclid}(v_s, c_j). \tag{4.3}$$

Here each sample s will have k distance values each corresponding to the distance from k classes. Note that the difference between Eq. 4.2 and Eq. 5.2: we do not have the class label information for test sample s, whereas we know the ground-truth label for training sample x_i.

Step 2: We propose to invoke our first filtering criterion on test data based on the distance we calculated in the previous step. It is expected that trained models can make better predictions when test data follows the similar distribution as the training data. However, for a state-of-the-art DNN, it is not possible to differentiate between samples that do or do not belong to the same distribution as it has observed during the training process. Hence, we calculate the maximum distance observed from respective centroids in training data to get the sense of data distribution. We utilize this maximum distance as a threshold τ for in-the-wild test samples so that the model can identify out-of-distribution samples. Specifically, we calculate τ as follows

$$\tau_j = \max([d^j_{x_i}]^{N_j}_{i=0}), \quad \forall x^j_i \in l_j$$

where $d^j_{x_i}$ is the distance between centroid c_j and train sample x_i that belong to class l_j.

Step 3: In the first phase of two layered filtering, we abstain test samples based on the threshold we calculate from training data. We first get the minimum of distances between each test sample s and all clusters $c_j \in C$. With this step, we abstain from classifying out-of-distribution samples.

$$d^{min}_s = \min([d^j_s]^k_{j=0}) \tag{4.4}$$

$$c^{min} = \arg\min([d^j_s]^k_{j=0}) \tag{4.5}$$

And we abstain samples if the following condition satisfies:

$$d_s^{min} > \tau_{c^{min}} \tag{4.6}$$

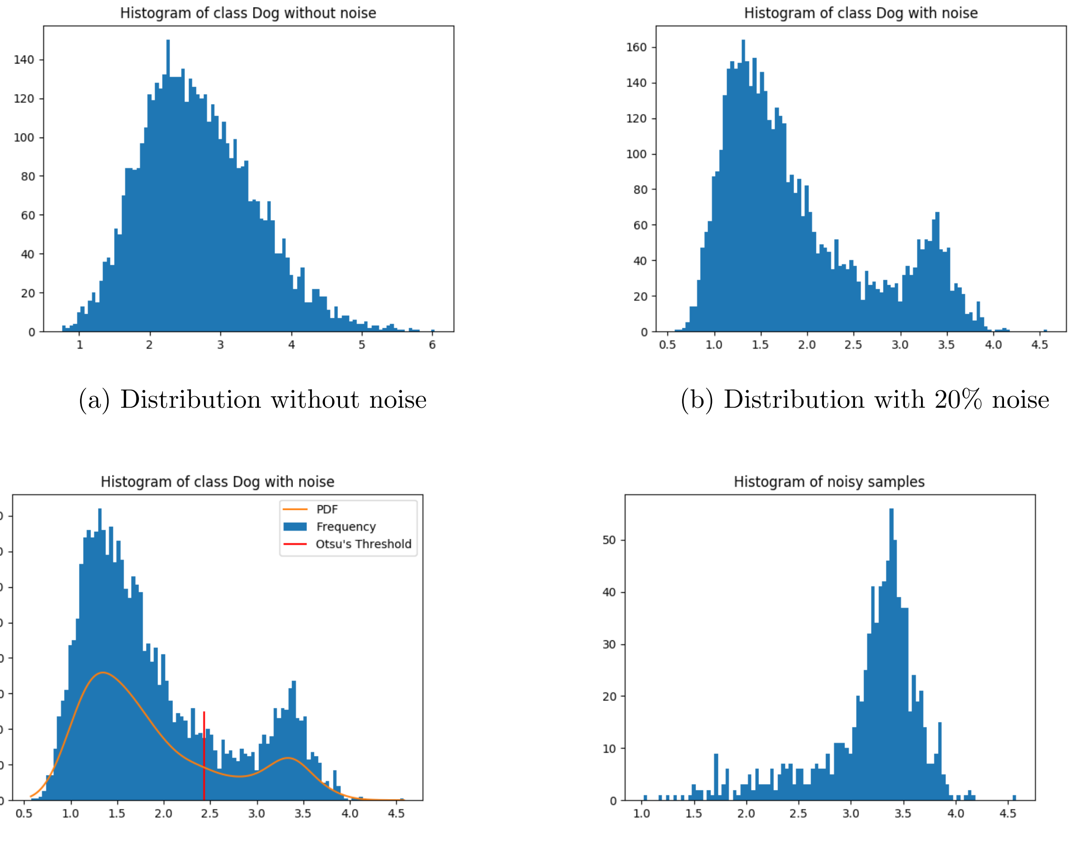
Step 4: Having out-of-distribution samples abstained, we here focus on the noisy or confusing samples. Samples that are similarly distanced from two or more class-constrained data distributions, are deemed as confusing samples. We only consider those distributions that are closest from the sample since samples belong to the distribution that they are closest to. We abstain sample s if the following condition holds:

$$|d_s^a - d_s^b| < \eta,$$

where a and b are the two nearest clusters from sample s. d_s^a and d_s^b are the distances between sample s and centroids c_a and c_b, respectively, and η is a tolerance parameter that we set empirically.

4.1.3 Modality Detection and Thresholding

We hypothesize that DNN features are closely clustered when samples share similar features, and they become scattered when there are less correlations between features. When noise is present in any classes of data, varying correlations between samples are observed. For example, as depicted in Fig. 4.1(b), multiple modalities in distance distribution of noisy samples from the same class are observed. We also observe from Fig. 4.1(a) single modality in distance distribution when data from the same class is free of noise, which supports the above hypothesis. Modality in distance distribution plays a key role in detecting noisy samples during training.



(a) Distribution without noise

(b) Distribution with 20% noise

(c) PDF and Otsu's threshold on distribution

(d) Distribution of 20% noise

Figure 4.1 Histograms of the distances between samples from CIFAR10 dataset and cluster centroids.

We utilize Kernel Density Estimation (KDE) to perform the modality test on distance distribution. Distance d here can be considered as a univariate sample that is drawn from some distribution with unknown density that we would like to model. With KDE, the Probability Density Function (PDF) of d can be approximated as

$$\mathrm{PDF}(d) \approx \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{d-d_i}{h}\right), \tag{4.7}$$

where K is the kernel function, e.g. the Gaussian kernel, and h is a smoothing parameter that we empirically set to 0.3 in order to avoid detecting false peaks.

We identify the number of peaks by calculating the gradient of the KDE curve. If we detect a single peak, we can define the distribution as unimodal, otherwise multimodal. Interestingly, in all our experiments we have observed that in presence of random label noise, distance distributions always follow bimodality. Hence, here we focus on bimodal distributions. But our proposed method can easily be generalized to multimodal distributions.

In case of a bimodal distance distribution, we propose to re-purpose Otsu's thresholding [90] to detect cut-off threshold in order to detect noisy samples from training data. In image processing, Otsu's method is widely used to perform automated image binarization. The algorithm returns a single intensity threshold to separate image pixels into two classes: foreground and background. The algorithm exhaustively searches for a threshold t that maximizes the inter-class variance σ_B^2 of the two classes, which is defined as

$$\sigma_B^2(t) = \omega_0(t)\omega_1(t)(\mu_1(t) - \mu_0(t))^2 \tag{4.8}$$

where ω_0 and ω_1 are the probabilities of the two classes separated by t and μ_0 and μ_1 are the means of two classes. We repurpose this algorithm to detect the cut-off threshold of bimodal distance distribution. We define two classes from Otsu's algorithm as clean and noisy data distributions. Fig. 4.1(c) shows the detection of bimodality by detecting peaks in distance distribution of noisy data; it also shows modality testing and Otsu's thresholding in practice. As we can see, Otsu's thresholding can effectively identify cut-off value to differentiate between noisy data and clean data. Compared to Fig. 4.1(d), which illustrates the ground truth distance distribution of randomized samples, we can observe that a very small number of noisy data samples fall below the Otsu's threshold, hence not excluded from training set. We deem this as expected since with 20% label noise introduced randomly, the probability of samples not being randomized within this 20% for a particular class is $\frac{1}{k}$ given that we are randomizing each of the k classes uniformly.

4.2 Experimental Analysis

In this section we demonstrate the performance of the proposed framework using various CNN architectures, e.g. VGGNet [91] and ResNet [92] on multiple image classification benchmarks: CIFAR10 [93], SVHN [94], and Fashion-MNIST [95]. We compare our method with the state-of-the-art algorithms SelectiveNet [1] and DAC [38]. To ensure a fair comparison, our experiments closely follow those of the competing methods. We plan to open source our code to facilitate the research in this area.

4.2.1 Detecting label noise

We aim at a problem of label noise that might occur on some fraction of data. Here, we assume that a fraction of labels have been corrupted by random assignment. Our proposed framework identifies the mislabeled samples and removes them as noisy samples from training set. To identify the corrupted samples, we first train an off-the-shelf DNN with best practice regularization to avoid overfitting using a validation set, which we assume to be clean. Our proposed framework utilizes the features extracted by the trained DNN to differentiate between noisy and clean training examples. We present our results by retraining the same DNN from scratch with cleaned training set.

We first compare our proposed framework with DAC [38], a state-of-the-art method that introduces an additional abstention class to learn to abstain noisy samples during training. We also present the performance of the bare baseline model, which is the same DNN utilised in both DAC and our proposed method. To ensure fairness, we report our results using similar setup as [38] and we use the numbers reported in their paper [38].

We conduct experiments on CIFAR10 [93] and Fashion-MNIST [95] with varying amount of arbitrary label noise. In our proposed approach, we use same CNN architectures for pre-training and retraining phases. We use the same hyperparameters, e.g., initial learning rate, learning rate decay and optimizer, as in DAC and baseline model for the retraining phase. We utilize ResNet18 and ResNet34 [92] without modifications as our DNN architecture for experiments presented in this section. For the $DBSCAN$ algorithm, we empirically set $MinPts = 300$ and $eps = 0.8$ for our experiments. We randomly choose a seed value (seed $= 1$) in our experiments to ensure reproduciblity.

Table 4.1 presents the comparative results of this experiment. Our proposed framework achieves improved accuracy in most of our experiments as compared to the state-of-the-art DAC [38]. Our framework could identify noisy data points reliably, even outperforming specialized learning model DAC when percentage of noise label is lower than 60%. When percentage of noise label is 80%, we observe that our framework does not perform as well as DAC. This is because with highly corrupted data our method's performance degrades as DNN struggles to learn class specific features, which results in scattered feature distribution of training data. In order to promote simplicity, our framework does not use the feedback

		Models		
Dataset	Noise Label	Baseline	DAC	Ours
CIFAR10 (ResNet34)	20%	88.64%	92.91% (0.24/0.01)	**93.03%** (0.25/0.03)
	40%	85.95%	90.71% (0.41/0.03)	**90.88%** (0.41/0.03)
	60%	80.92%	**86.30%** (0.56/0.07)	86.28% (0.56/0.05)
	80%	67.17%	**74.84%** (0.75/0.16)	69.7% (0.64/0.16)
Fashion-MNIST (ResNet18)	20%	93.92%	94.76% (0.25/0.01)	**94.95%** (0.21/0.01)
	40%	93.09%	94.09% (0.48/0.01)	**94.20%** (0.38/0.02)
	60%	91.83%	92.97% (0.66/0.03)	**93.05%** (0.58/0.01)
	80%	88.61%	**90.79%** (0.88/0.04)	89.77% (0.72/0.03)

Table 4.1 Comparative results (accuracy) with varying percentages of noise labels.

loop from the noise to the model, whereas DAC [38] models the noise explicitly while training, which helps them to learn more from the clean samples than from the noisy ones, yet DAC still suffers from the issue of memorization [38]. Nevertheless, we argue that presence of very high amount of noise (e.g., 80%) in dataset is not very realistic in real world as the label quality in this case is close to be random (e.g., random guess accuracy on 10 classes is already 10%), hence investing heavily to improve performance in this scenario is impractical. Despite that, our framework achieves improved results when percentage of noise label is lower than 60% even though DAC requires specialized loss function to learn the pattern of noise while training, whereas our proposed framework employs simple yet effective filtering approach on feature space extracted by DNNs.

Dataset	Coverage	Models		
		SelectiveNet (varying coverage)	SelectiveNet (100% coverage)	Ours
CIFAR10 (VGG16)	100%	93.21%	93.21%	93.21%
	95%	95.40%	95.44%	**95.64%**
	90%	97.27%	97.16%	**97.41%**
	85%	**98.40%**	98.19%	**98.40%**
	80%	**99.03%**	98.69%	98.97%
	75%	**99.31%**	98.83%	99.24%
	70%	**99.40%**	98.94%	**99.40%**
SVHN (VGG16)	100%	96.22%	96.22%	96.22%
	95%	**98.20%**	97.80%	97.88%
	90%	98.97%	98.74%	**99.07%**
	85%	99.25%	98.99%	**99.40%**
	80%	99.41%	99.10%	**99.49%**

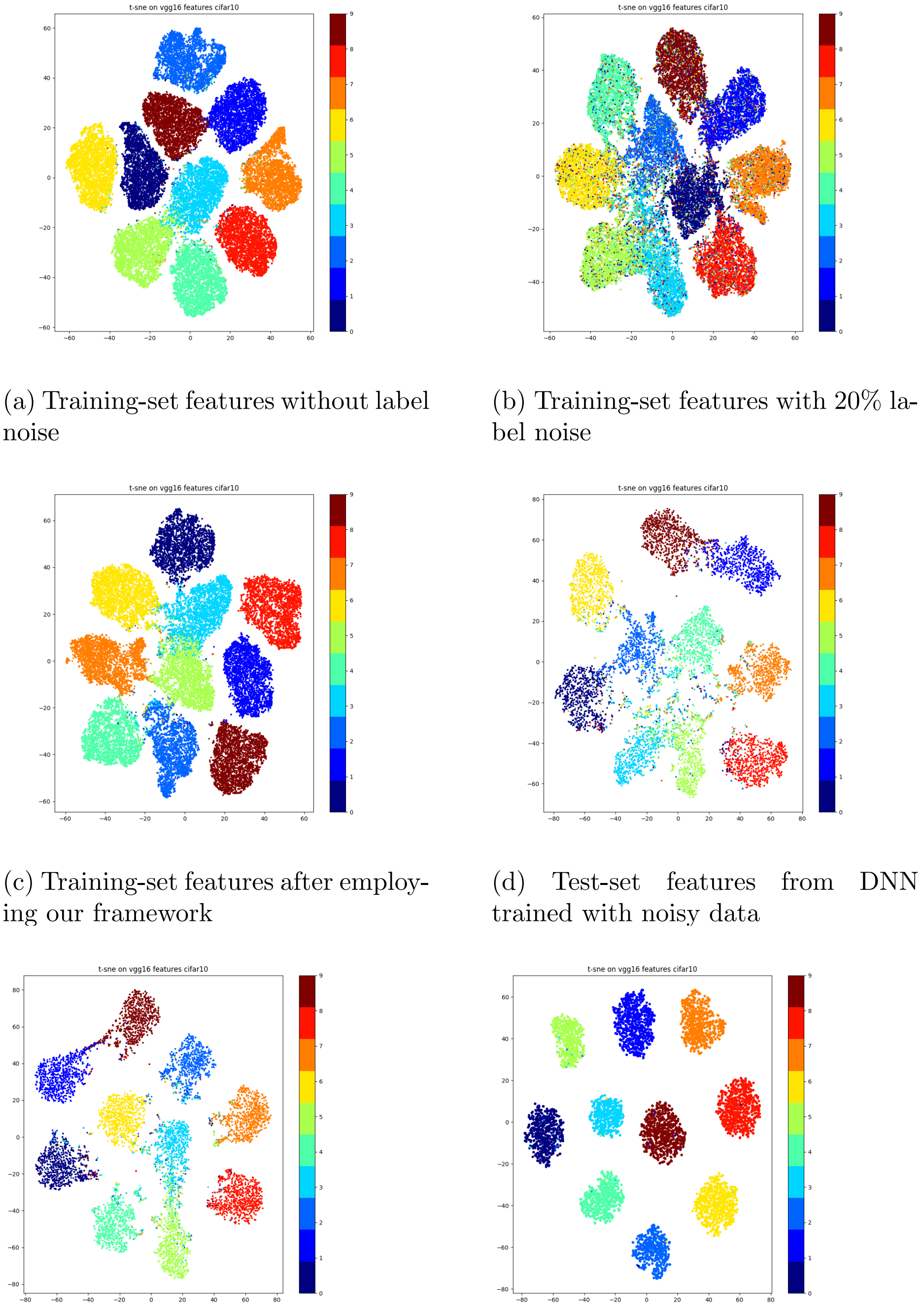
Table 4.2 Comparative results on CIFAR10 with varying calibrated coverages of our proposed framework and SelectiveNet [1].

4.2.2 Abstaining test samples

We now consider the predictive uncertainty problem during inference. For these particular experiments, we assume training data is free of noise, but in-the-wild test samples may be noisy or confusing. We aim to abstain such samples using our proposed framework to reduce predictive uncertainty. We first train an off-the-shelf DNN with given dataset and in the post training phase we employ our algorithm to filter out samples that are deemed confusing or out-of-distribution.

To demonstrate the advantages of our proposed framework, we compare its performance with state-of-the-art SelectiveNet [1], and report the results in Table 4.2. SelectiveNet [1] proposes a specialized rejection model that learns to reject any sample that produces high cross-entropy loss under the constraint of user-specified coverage. We use similar parameter settings reported in the paper [1] for a fair comparison. Note that reported numbers for "SelectiveNet (varying coverage)" are obtained by training with target coverage value and inferred on the same calibrated coverage as described in [1], whereas we report our performance by training DNN once and use varying calibrated coverage by tuning tolerance parameter η accordingly only during inference. To make a fair comparison, we also train SelectiveNet with 100% coverage only, similar to ours, and then use varying calibrated coverage to obtain results for "SelectiveNet (100% coverage)". Our proposed framework achieves reported results with greatly reduced complexity (both time and resource) compared to SelectiveNet.

We conduct experiments on CIFAR10 [93] and SVHN [94] with off-the-shelf VGG16 [91] architecture. Performance analysis is presented in Table 4.2. We observe that our proposed framework can outperform or achieve very similar performance compared to SelectiveNet [1] for both datasets. Moreover, our framework demonstrates a very clear advantage when compared with results from "SelectiveNet (100% coverage)". We found this observation very intriguing as our proposed method only takes advantage of feature learning capability of DNN coupled with intuitive filtering techniques. This would mean that using specialised loss functions to abstain samples has very small impact on the performance, and DNNs are robust enough to learn distinctive features but lack the ability to reject noisy or confusing samples. Overall, not only being more efficient (i.e., training once), our proposed framework also achieves better accuracies in most of the coverage levels, demonstrating the superiority of our method.



(a) Training-set features without label noise

(b) Training-set features with 20% label noise

(c) Training-set features after employing our framework

(d) Test-set features from DNN trained with noisy data

(e) Test-set features from DNN trained with denoised data

(f) Test-set features after abstaining from DNN trained with denoised data

Figure 4.2 t-SNE visualization of CIFAR10 training and test sets in feature space.

4.2.3 Visualizing Effectiveness of Proposed Framework

In order to demonstrate the effectiveness of our proposed framework in detecting noise in both training data as well as in-the-wild test data, we visualize the feature spaces of the trained model ResNet34 [92] on CIFAR10 [93] using T-distributed Stochastic Neighbor Embedding (t-SNE) [96]. We utilize color coding to annotate samples from different classes.

We visualize how data distribution is affected by noise in Fig. 4.2, where Fig. 4.2(a) presents the visualization of CIFAR10 training set features without any artificial noise, yet we can observe a very small amount of noise. We hypothesise that similar noise can be present across different annotated datasets, targeting a variety of tasks, available today. When we introduce 20% random label noise to the dataset, the samples get more scattered across the feature space (Fig. 4.2(b)). Our framework can identify these noisy samples and effectively clean them as demonstrated in Fig. 4.2(c). We also present the visualization of test samples from CIFAR10 in Figs. 4.2(d)-(f). Training with noisy data adversely affects the DNN's ability to extract features robustly (Fig. 4.2(d)). If data denoising is performed prior to training a DNN, we can minimize this adverse effect greatly (Fig. 4.2(e)). However, data distributions still cannot be very concise and often overlap. This phenomenon can be explained as even if there are no noise in a training set and the test set might still contain noise and confusing samples. Our framework can filter out most of the boundary samples from respective distributions, as demonstrated in Fig. 4.2(f). Yet, if we closely observe, our framework missed some samples what are well within the distribution but predicted labels do not match the ground-truth labels. We argue that these samples share dominant

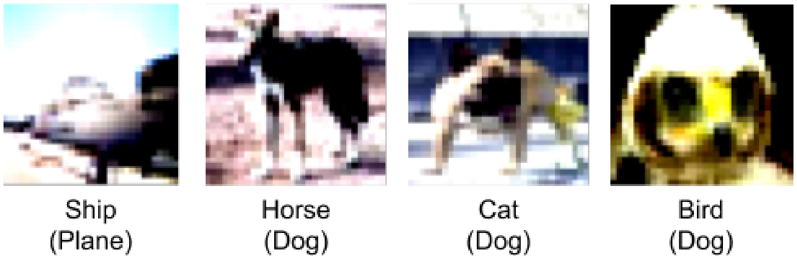


Figure 4.3 Sample images identified by our framework that are potentially mislabeled in CIFAR10 testset. Text below each image denotes the ground-truth label provided by CIFAR10 and text in parenthesis are the predicted labels by our framework.

features with samples from the closest distribution or may be mislabeled, as we can similarly observe in clean training data distribution (Fig. 4.2(a)). We have presented some examples of potentially mislabeled test samples of CIFAR10 in Fig. 4.3.

Fig. 4.4 provides further evidence of the effectiveness of our proposed framework. In this plot we show how denoising helps accelerate learning of DNNs. In this experiment, we train ResNet34 on CIFAR10 introducing heavy label noise (60%). Fig. 4.4 shows a stark difference between learning from original noisy data vs. denoised data by our framework. Our framework not only accelerates learning (left), but also improves accuracy on test data when learning from denoised data (right). Evidently, our framework can effectively clean data and expedite learning by eliminating noisy or confusing samples.

4.3 Conclusion

Noisy data is one of the most crucial hurdles for DNNs to achieve high accuracy and reliable performance. In this paper, with rigorous experimentation, we have shown that complicated, specialized training to filter out noise in data is not always effective and necessary. On the

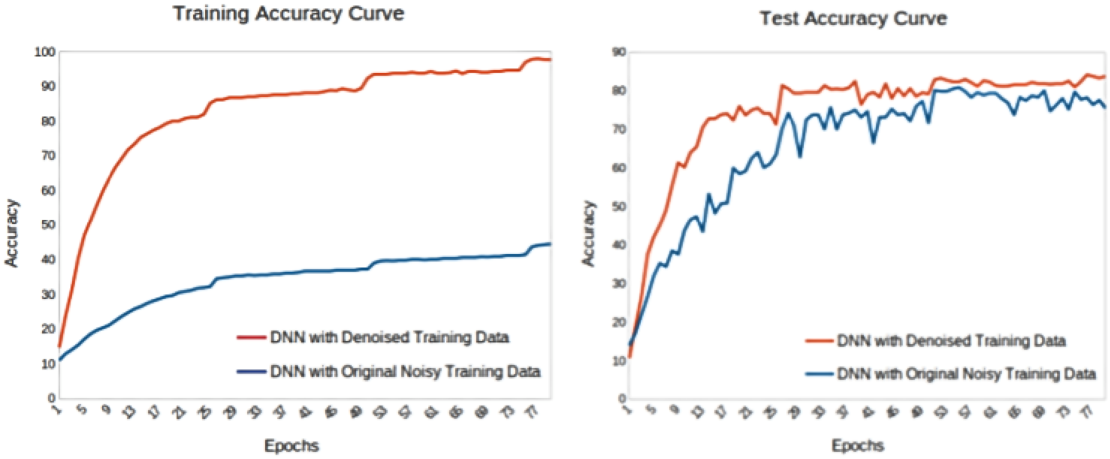


Figure 4.4 Effect of our proposed framework on training ResNet34 with the original noisy data (60% label noise) and denoised data. (left) Learning curves on CIFAR10 training data; (right) Learning curves on CIFAR10 test data.

contrary, we show features learned by off-the-shelf DNNs are quite robust. With a simple yet effective filtering mechanism, we can achieve competitive, often better, performance than these specialized models. However, we would like to point out some limitations and future work of our proposed framework. We consider threshold based on distance from distributions as a filtering criteria. While it has proven to be very successful, a distance threshold will limit data distribution to be spherical, but in reality data distributions can often be irregular. This can explain why our framework sometimes does not perform as expected. A more robust filtering method requires a more accurate model of distribution. One other pathway to address this issue would be learning more robust features along with filtering techniques. We leave these areas open for future research.

CHAPTER 5

Towards Reduced Risk and Uncertainty of Deep Neural Networks on Diagnosing COVID-19 Infection

Effective and reliable screening of patients via Computer Aided Diagnosis could play a crucial part in the battle against COVID-19. Most of the existing works focus on developing sophisticated methods that yield high detection performance, while not addressing the predictive uncertainty of their proposed systems. In this work, we propose to utilize density driven uncertainty estimation to detect confusing cases for further expert referral to address the unreliability of state-of-the-art (SOTA) DNNs on COVID-19 detection. To the best of our knowledge, we are one of the first to address this issue on COVID-19 detection problem. We also propose a novel feature denoising algorithm to further improve the Positive Predictive Value (PPV) of COVID positive cases. In collaboration with medical professionals, we further validate the results to ensure the viability of such systems in clinical practice. With extensive experimentation, we show that our proposed framework can effectively identify the confusing COVID-19 cases for further expert analysis, while outperforming the existing uncertainty estimation methods.

5.1 Methodology

Uncertainty is the source of risk in any decision making process, which leads to unreliability. Uncertainty associated with DNNs can be categorised into two types; aleatoric and epistemic uncertainty [97]. Aleatoric uncertainty is caused by the noise in data, whereas epistemic uncertainty is generated from the stochasticity of DNN models. Monte-Carlo methods are frequently used to estimate Epistemic uncertainty, and Aleatoric uncertainty can be reduced by gathering more knowledge, which is an expensive process and often infeasible in real-world scenarios.

In this paper, we aim to address the Aleatoric uncertainty by identifying noisy or confusing samples from data distribution using features learned by DNNs. In supervised learning, DNN learns high-level features from data dictated by class constraints. In an ideal world, samples of each class would be clearly separated and all samples of a class would be clustered together. However, in real-world data, there exist inter-class affinity and intra-class diversity often due to errors in labeling or noise in data samples.

5.1.1 Problem Formulation

A supervised prediction task is formulated as follows. Let us consider a training set consisting of input-target pairs, $D = (x_i, y_i)_{i=1}^{N}$, where $x_i \in \mathbb{R}^n$ belongs to one of the $k \in L = \{l_1, l_2, \cdots, l_k\}$ classes. A DNN classifier consists of a feature extractor and a classifier. Feature extractor is a parameterized function $f_\theta : \mathbb{R}^n \rightarrow \mathbb{R}^\delta$ that learns to map n dimensional observed data x_i to feature space v_{x_i} of δ dimensions under the $l_j \in L$ class constraint. Classifier is a simple mapping function, $f_c : f_\theta(x_i) \rightarrow y_i$, which can be a softmax classifier.

Typically, parameter θ is optimized using the off-the-shelf cross-entropy loss. In an ideal scenario, f_θ thus learns to cluster semantically-similar inputs x_i to k different clusters corresponding to k classes. For any given new sample s, f_θ maps it to a feature vector v_s, which ideally should lie under any of k distributions observed during training. However, in real-world data, there might be $p \geq k$ clusters formed by observed data samples of k classes.

As shown in [28], we can robustly define these underlying data distributions with DBSCAN clustering [89]. On the feature space, $DBSCAN$ algorithm identifies the core clusters given class constraint, then centroids can be utilized to define these clusters robustly. Our aim in this work is to identify samples that are nontrivial to classify within these k clusters utilizing their distance from cluster centroids.

Calculation of Centroid Let us assume there are m samples in class j, and $DBSCAN$ returns a core cluster with m_{core} samples, where $m_{core} \leq m$. Then the centroid is defined as

$$c_j = median([v_{x_i^j}]_{i=0}^{m_{core}}), \quad \forall x_i^j \in l_j. \tag{5.1}$$

We collectively denote all k identified centroids as $C = \{c_1, c_2, \cdots, c_k\}$.

5.1.2 Abstaining Confusing Samples

We hypothesize that DNN features are closely clustered when samples share similar features, and they become scattered when there are less correlations between features. Some samples often share features from different distributions and they exist near the boundary of either of these distributions. Hence, such samples that are similarly distanced from two or more class-constrained data distributions, are defined as confusing samples. We only consider those distributions that are closest from the sample since most of the cases samples belong to the distribution that they are closest to. During inference, we utilize the centroids calculated after training process to identify such confusing samples. We deem in-the-wild sample s to be confusing if the following condition holds:

$$|d_s^a - d_s^b| > \eta,$$

where a and b are the two nearest clusters from sample s. d_s^a and d_s^b are the distances between sample s and centroids c_a and c_b, respectively, and η is a tolerance parameter that we set empirically. By tuning the tolerance parameter, one can define the abstention rate of the framework, which provides additional control over the uncertainty of DNNs.

We calculate distance between test samples and training data distributions under constraint of k classes. For $s \in S$, where S is the set of in-the-wild test samples, we calculate distance d_s between s and centroid $c_j \in C$:

$$d_s^j = euclid(v_s, c_j). \tag{5.2}$$

Here each sample s will have k distance values each corresponding to the distance from k classes. Note that we do not have the class label information for test sample s.

5.1.3 Refinement of Feature Vector

Lack of information or misinformation often cause confusion for DNNs. Hence, DNNs often make prediction on samples based on sub-optimal features extracted from noisy samples. To some extent DNN classifiers (e.g. softmax or sigmoid) are robust to these feature noise due to the supervised feedback process. On the other hand, our proposed technique solely rely on the features learned by the DNN and unlike other complex methods, it doesnot require learning from the data. However, precise calculation of centroids is prerequisite to the success of our framework, as these centroids are utilized to determine the confusing samples. Hence, we propose a number of alternative ways to minimize the aleatoric uncertainty of the DNNs while filtering out the confusing samples. First, we propose to utilize the most simplistic idea of Monte-Carlo Dropout (MCDO) on feature vectors to randomly dropout features and calculate centroid. After T iterations we get T centroids for a particular data distribution and then we calculate the median to get the final centoid for the distribution. Using MCDO would reduce the dependency to a particular set of features while calculating centroid. Median of the centroids can be defined as follows.

$$c_j^{median} = median([c_j^t]_{t=0}^T) \tag{5.3}$$

Where c_j^t is the centroid of class j on t^{th} iteration.

MCDO is a naive approach as we decide to dropout random features on each run without considering their importance. To address this, we propose to utilize statistical analysis of the features rather than randomly choosing features to dropout. Specifically, we utilize chi test to obtain the scores on each features and based on an empirical threshold we filter-out the features with low statistical scores. However, chi test alone cannot quantify the importance of features. Variance of features is also calculated to identify very low variant and very high variant features. Low variance could mean these features are useless as they are present across different distributions. Again, very high variance could mean that these features are random noise in the data.

5.2 Experimental Analysis

In this section, we present and analyze the experimental study, which demonstrates the effectiveness of the framework described in section 5.1. We conducted our experiments on the COVIDx dataset [2], which is the largest publicly available COVID-19 dataset in terms of the number of COVID-19 positive patient cases.

5.2.1 Experimental Setup

For comparative studies of existing uncertainty estimation methods, we choose ResNet and VGGNet as our baseline DNNs as they are successful and popular in CAD systems. We utilized SelectiveNet [1] and incorporated our proposed method on these baseline DNNs to obtain the results. We plan to open-source our code to facilitate the research in this area.

We utilized SDG optimizer with a learning rate of 0.1 and used seed as 1 in all our experiments with ResNet and VGGNet to ensure reproducibility. We set the batch size to 32 for these experiments. We kept these settings constant to ensure a fair comparison. We also followed similar preprocessing steps mentioned in the COVID-Net paper. We measure the test accuracy for performance analysis, along with Positive Predictive Value (PPV) and Sensitivity for each class.

5.2.2 Comparative Study on Existing Methods

First, we present the experimental studies on the effectiveness of existing methods on the COVIDx dataset [2]. We compare our proposed method with state-of-the-art SelectiveNet [], and report the results in Table 5.1. Please note, for a fair comparison, we report SelectiveNet results when trained their model with 100% coverage and then calibrated to the desired abstention rate. It can be observed from Table 5.1 that our proposed method outperforms SelectiveNet in all abstention rate. Moreover, SelectiveNet utilizes a specialized loss function

Abstention Rate	Model	
	SelectiveNet	Ours
0%	92.90%	92.90%
5%	93.28%	**93.94%**
10%	93.95%	**95.01%**
15%	94.37%	**95.70%**
20%	95.26%	**96.21%**
25%	96.48%	**96.72%**
30%	96.81%	**97.12%**

Table 5.1 Comparative results on COVIDx with varying abstention rates of our proposed framework and SelectiveNet [1]. In these experiments we used VGG-16 as the baseline DNN.

which requires modification to the existing DNN, whereas our method can be utilized with any DNNs in a plug-and-play manner. This demonstrates the superiority of our method over the existing state-of-the-art.

5.2.3 Effect on COVID-Net

To further explore the effectiveness of our proposed method, we incorporated our framework with state-of-the-art COVID-Net. Although the authors only open-sourced the trained model, the required effort to integrate our method was minimal. We present the results of this experiment in Table 5.2. As we can observe, we can effectively reduce the error rate of COVID-Net with the abstention of confusing samples. Our method can identify 49.4% of the mistaken samples as confusing while only abstaining 10% of the data for referral. Our method also improves PPV and sensitivity of COVID-Net dramatically with higher abstention rate. In order to demonstrate the effectiveness of our proposed framework in detecting confusing samples, we visualize the feature spaces of the trained COVID-Net model on COVIDx [2] test-set using T-distributed Stochastic Neighbor Embedding (t-SNE) [96] in Figure 5.1 a. From the visualization, it is visible how our model can identify the confus-

Abstention Rate	Accuracy	Sensitivity			Positive Predictive Value		
		Normal	Pneumonia	COVID	Normal	Pneumonia	COVID
0%	94.82%	94.80%	94.90%	94.00%	96.30%	92.80%	94.00%
10%	97.16%	97.80%	96.60%	94.80%	97.30%	97.10%	95.70%
20%	98.81%	99.60%	98.30%	95.60%	98.60%	99.60%	96.60%
30%	99.18%	99.70%	99.00%	96.60%	99.00%	99.7%	97.70%

Table 5.2 Experimental results on COVIDx with varying abstention rate our proposed framework with COVID-Net [2] as the baseline DNN. Note that, the results presented here with 0% abstention rate represent the COVID-Net performance.

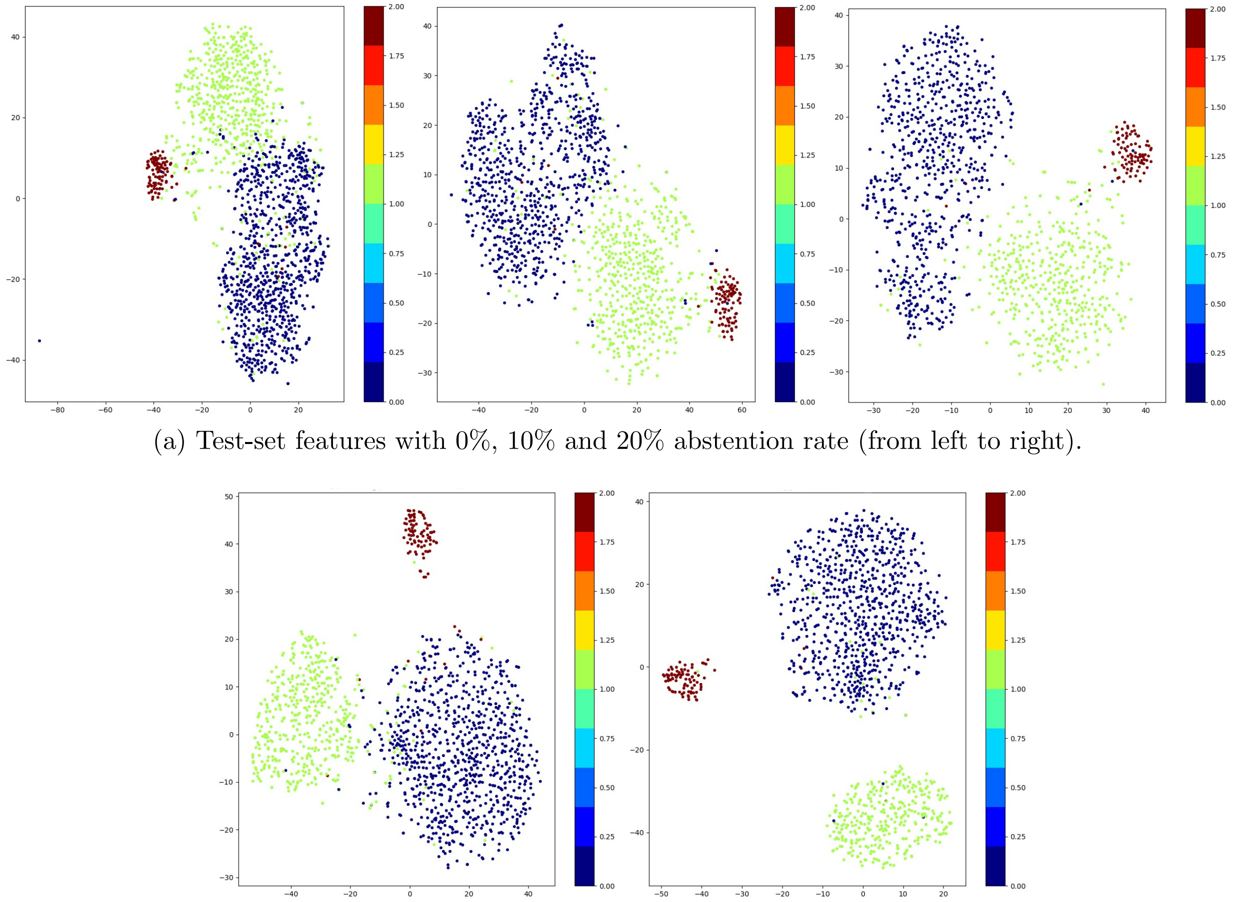
ing samples lying on the boarder of class distributions and with higher abstention rate well defined distributions for each class emerges. However, if closely observed, we can find few samples fall into wrong distributions. To address this issue, we proposed feature selection methods to be integrated with proposed uncertainty estimation framework.

5.2.4 Effects of Feature Selection on Uncertainty Estimation

We also experimented with the proposed feature selection methods on the COVIDx dataset and another benchmark dataset, CIFAR-10 [93]. The results are presented on Table 5.3. As we can observe, the chi-square test based selection method outperformed other methods, while both methods achieved better PPV and sensitivity for most cases. The feature selection method's effectiveness can be observed with the t-SNE visualization in Figure 5.1 a and b. If we consider the third plot of fig 5.1 a and fig 5.1 b, we can observe that the misclassification

Abstention Rate	Positive Predictive Value W/O Feature Selection			Positive Predictive Value W Feature Selection		
	Normal	Pneumonia	COVID	Normal	Pneumonia	COVID
10%	97.30%	97.10%	**95.70%**	**97.40%**	**97.50%**	**95.70%**
20%	**98.60%**	**99.60%**	96.60%	**98.60%**	99.50%	**98.80%**

Table 5.3 Experimental results on COVIDx demonstrating the effects of feature selection on our framework. In this experiment we empirically set the number of features to retain to 1024 from COVID-Net feature vector of length 259584.



(a) Test-set features with 0%, 10% and 20% abstention rate (from left to right).

(b) Test-set features with 10% and 20% abstention rate after feature selection (from left to right).

Figure 5.1 t-SNE visualization of COVIDx test-set in the feature space.

has been reduced. We argue that static or noisy features often create issues with the centroid calculation utilized in our proposed framework. After filtering these unwanted features, class constrained centroids become more robust to noise, hence improving performance.

5.3 Expert Analysis

To further analyze the experimental results we conducted on COVID-Net, we collaborated with medical professionals, including an Epidemiologist closely working with the COVID-19 outbreak. We set up the experiment as follows.

- First, we randomly sample x-rays from each class on the COVIDx test-set that are predicted correctly by the DNN, so that we get the samples where our model was confident enough to predict.

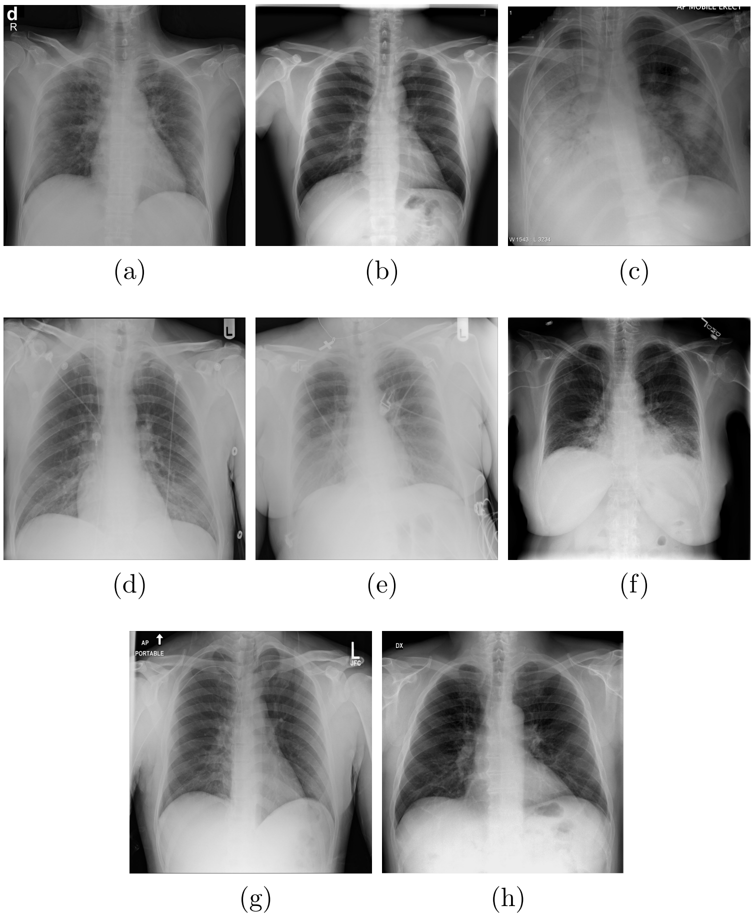
- Secondly, we sample x-rays from the abstained samples, while only 10% of data are abstained. The rationale behind this is that these are the most confusing samples to our model, chosen to be abstained for expert referral.

- Lastly, we sampled more x-rays from ones that were predicted wrong, yet not abstained by our model while abstaining 25% data. These are the samples where our model was pretty confident about the prediction but made mistakes.

We shared these three sets of samples with medical professionals without disclosing the sampling criteria to ensure unbiased analysis. Their analysis of each set is as follows.

- First set of samples were straight forward to diagnose (Fig 5.2 a-c).

- Samples from the second set were confusing, and medical professionals recommended lateral view x-ray or CT scan for further investigation. They mentioned that for some images, the x-ray quality was poor (Fig 5.2 d) as a reason for confusion. For some samples, the x-ray was not clear due to the obesity of the patients(Fig 5.2 e). There were a few samples where our model made mistakes, but the experts could diagnose. They pointed out that these x-rays had breast shadows as the patients were female (Fig 5.2 f).



(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 5.2 Samples from COVIDx dataset; (a)-(c) samples that were correctly classified by model; (d)-(f) samples that were deemed as confused when 10% data were abstained; (g)-(h) samples that were not abstained yet mistaken by model. Sample (g) and (h) were predicted as normal by the model while the ground truths are Pneumonia and COVID positive respectively.

- The last set were mostly identifiable expect a few poor quality samples. However, the experts agreed with the DNN's prediction over the ground truth on two samples (Fig 5.2 g and h). We argue that, these samples may be affected by label noise.

5.3.1 Recommendations from the Experts

X-rays are not a very reliable indicator of diagnosis. However, a CT scan or RT PCR test may not be available in remote parts of the world, where x-ray can be available. Hence, detecting critical patients via x-ray analysis could save their lives. However, our collaborating medical professionals suggested using a better quality x-ray for detection. They also recommended associating metadata with x-ray analysis, e.g., sex, BMI index, other clinical features, etc. for more reliable detection performance.

5.4 Conclusion

Since the beginning, COVID-19 has been causing devastation in every part of our lives. Detection and intervention are critical for patients who develop COVID-pneumonia. The research community has come together to create a reliable and accurate COVID-19 detection system with Deep Learning. On this consolidated effort, we intend to add our contribution. We are one of the first to address the predictive uncertainty issue of DNNs by proposing an uncertainty estimation framework on COVID-19 detection. Additionally, we also proposed a feature selection algorithm to improve PPV on the COVID data. Through extensive experimentation, we demonstrated that our framework could effectively improve the reliability of existing CAD systems. With expert collaboration, we further analyzed the samples to gain valuable insights regarding such CAD systems. Lastly, we came across a number of potential research areas that require further investigation; collecting high-quality x-ray data, handling potential label noise that may occur during data collection, incorporating clinical metadata with x-ray analysis, and handling obesity and sex bias on data. We leave these areas open for future research.

CHAPTER 6

Uncertainty Aware Self-Supervised Contrastive Learning towards Efficient Representation Learning: A Real-World Scenario Evaluation.

While supervised learning has proven to be very successful in many applications, limited noise-free annotated data has created a significant bottleneck. Self-supervised learning, on the other hand, can play an important role in avoiding stringent labeled data constraints. In this work, we explore SimSiam [48], a self-supervised learning method that can learn underlying representations from data without labels. While prior work have explored the SimSiam network on benchmark datasets, we evaluate the representations learned from data samples to understand the efficacy of this model on a real-world scenario, specifically, COVID-19 detection from Chest X-Ray (CXR) images. Through empirical evaluation, we demonstrate that the SimSiam network can learn useful representations achieving very comparable performance against the supervised counterpart. We also incorporate Uncertainty Estimation framework to compare the performance with a supervised model to better grasp the distribution of learned representations. These results demonstrate the promise of Siamese networks for generating robust representations while reducing the data annotation cost of medical data significantly, and establish future directions for employing such networks in real-world scenarios where annotating data is costly.

6.1 Methodology

Self-supervised learning though was first introduced in robotics, machine learning researchers further develop the idea into different aspects and applications. In self-supervised learning, models obtain labels from the data itself by using a semi-automatic process. One crucial part of self-supervised learning is data augmentation. The SimSiam Network [48] utilizes simple siamese architecture on these augmentations to learn representation from data samples. In the later sections, we explore the SimSiam Network and our proposed pipeline for evaluation in details.

6.1.1 Siamese Network

A Siamese Neural Network is a type of neural network architecture that contains two or more identical sub-networks [76]. These sub-networks share weights and parameters. Updates from each backward pass are mirrored across both sub-networks. It is used to find the similarity of the inputs by comparing its feature vectors, hence, these types of networks do not depend on traditional class labels.

Traditional DNNs learn to predict multiple classes. Moreover, these types of networks require a vast amount of labeled data, which in many use cases are hard to acquire. One other issue with these networks is that in case a new class is introduced or removed from the use case, the whole network requires to be retrained. To address these drawbacks, siamese networks learn from a similarity function.

For a given input pair, X_i and X_i', siamese architecture can be depicted as shown in the Figure 6.1. X_i and X_i' are fed into two identical DNNs, $f_w(.)$. Then, the outputs from the DNNs are fed into contrastive energy function as defined below.

$$E_w = ||f_w(X_i) - f_w(X_i')|| \tag{6.1}$$

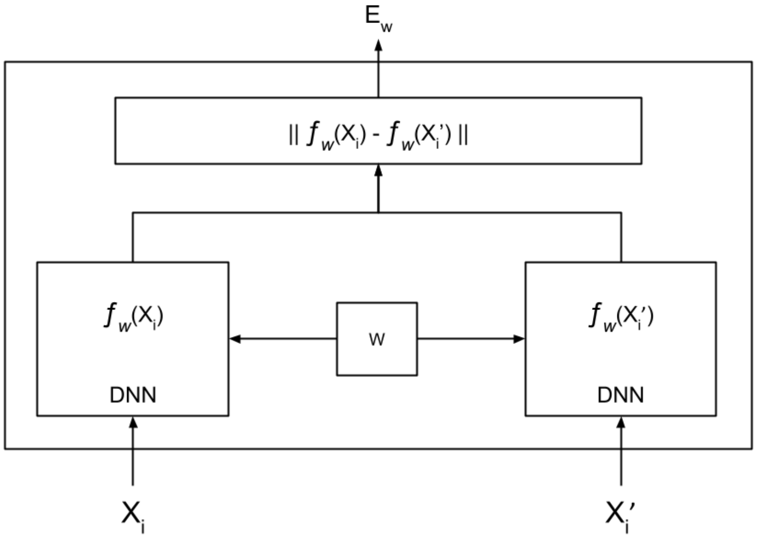


Figure 6.1 Siamese Network Architecture

In conventional use cases, the inputs to Siamese networks, X_i and X_i' are from different images, and the comparability is determined by supervision.

There are some advantages and disadvantages of Siamese architecture. These networks are more robust to class imbalance. Due to the fact that these networks do not rely on data labels, class imbalance does not affect representation learning. In most real-world scenarios, the data is imbalanced. Another advantage is these networks learn from Semantic Similarity. Siamese focuses on learning embedding that place the same classes/concepts close together.

Besides strong advantages, Siamese networks have some disadvantages. These networks require more training time than normal networks. Since Siamese Networks learn from quadratic pairs, it is slower than the normal classification type of learning.

6.1.2 SimSiam Network

In conventional use cases, the inputs to Siamese networks, X_i and X_i' are from different images, and the comparability is determined by supervision. However, as we are dealing with unlabeled data during train, recent self-supervised Contrastive Learning frameworks [45, 46] propose negative and positive samples for each iteration. Positive examples here are referred to as the transformed view, X_i and X_i', of the input image, X, whereas the negative examples are the samples that are not generated from X. With positive and negative samples in each iteration, the siamese model learns distinct features from the samples by contrasting between them [45, 46]. However, one major drawback to these contrastive learning methods is that they require a large number of negative samples, and the batch size required is very large. Such models require TPUs to train which are not feasible in most use cases.

Authors in [48] propose to use a simple siamese network and rather than using negative samples, they propose to only utilize positive samples during training. To avoid gradient collapse, they proposed a Stop-Gradient operation on the one branch of the siamese networks.

The SimSiam architecture takes two inputs, similar to siamese networks, X_i and X_i', which are two views generated from original image X. The two views are processed by an encoder network f_w consisting of a backbone DNN and a projection MLP layer. The encoder f_w shares weights between the two views. The projection head, h, transforms the output of one view and matches with the other. Figure 6.2 depicts the architecture.

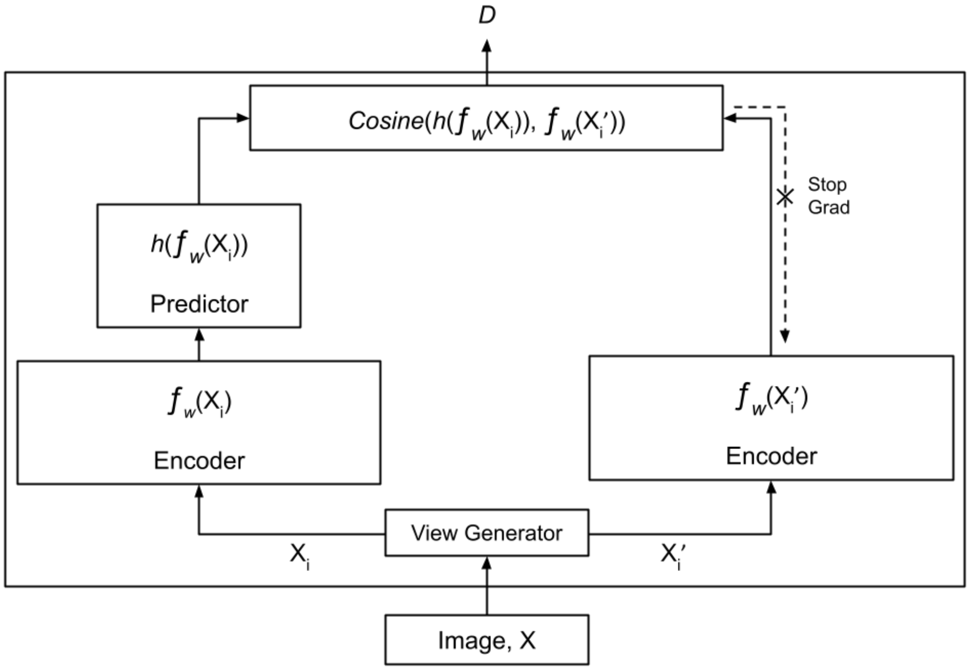


Figure 6.2 SimSiam Network Architecture

The authors [48] propose to use negative cosine similarity to measure the distance, $\mathcal{D}$.

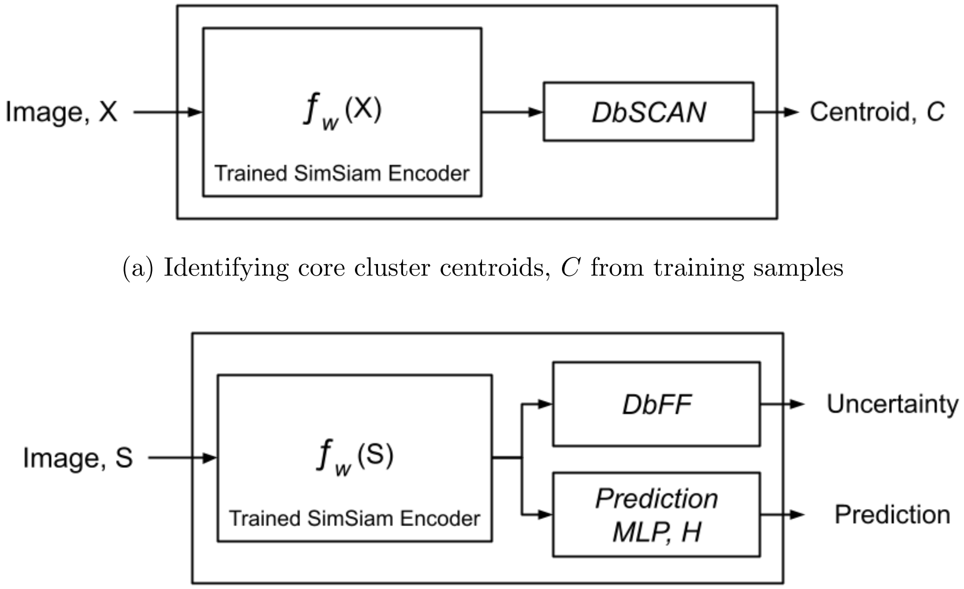
$$\mathcal{D}(p_i, z_i) = -\frac{p_i}{||p_i||_2} \cdot \frac{z_i'}{||z_i'||_2} \tag{6.2}$$

Where $||\cdot||_2$ is $l2$ norm, $p_i \triangleq h(f_w(X_i))$ and $z_i' \triangleq f_w(X_i')$.

The authors also propose symmetric loss with Stop-Gradient to calculate the loss, $\mathcal{L}$.

$$\mathcal{L} = \frac{1}{2}\mathcal{D}(p_i, \texttt{stopgrad}(z_i')) + \frac{1}{2}\mathcal{D}(p_i', \texttt{stopgrad}(z_i)) \tag{6.3}$$

Here the backbone DNN on X_i' receives no gradient from z_i' in the first term, but it receives gradients from p_i' in the second term and vice versa for X_i. Please refer to the paper [48] for further details of implementation.



(a) Identifying core cluster centroids, C from training samples

(b) Evaluating uncertainty and prediction for test samples

Figure 6.3 Incorporating DbFF with trained SimSiam Encoder.

6.1.3 Uncertainty Estimation and Confusing Sample Abstention

With the SimSiam network learned representations, the issue of uncertain prediction still persists. We aim to evaluate how SimSiam representations affect the uncertainty estimation methods, specifically, the DbFF framework. As this method heavily relies on representation learned from the data, the performance of the DbFF framework would heavily depend on the robustness of the SimSiam network. The overall flow of the algorithm is depicted in Figure 6.3.

First, utilizing trained encoder from SimSiam network, f_w, we generate the respective representations, v_X, for training samples, X. We then feed the v_X to $DbSCAN$ algorithm to generate core clusters and calculate respective centroids, $C = \{c_1, c_2, \cdots, c_j, \cdots\}$, using equation.

$$c_j = \text{median}([v_{x_i^j}]_{i=0}^{m_{core}}), \quad \forall x_i^j \in l_j. \tag{6.4}$$

Here, m_{core} is the number of core samples in cluster and l_j is the class label of sample x_i^j.

During inference, similar to the previous step, we generate representation, v_s for test sample, S. We feed v_s to DbFF framework to evaluate uncertainty of the samples and Prediction head, $\mathcal{H}$, to generate respective classification of the sample. Please refer to [28] for further details on how uncertainty score is generated and how a sample is abstained based on the score.

6.2 Experimental Analysis

In this section, we present and analyze the experimental study, which demonstrates the effectiveness of the SimSiam network, described in Sec. 6.1. We conducted our experiments on the COVIDx dataset [2], which is the largest publicly available COVID-19 dataset in terms of the number of COVID-19 positive patient cases. COVIDx is comprised of a total of 13,917 CXR images (Normal:7966, Pneumonia: 5462, COVID:489) for training and 1578 CXR images (Normal:885, Pneumonia: 593, COVID:100) as test-set. We visualize the gradient activation projected on CXR images to provide insight into how the SimSiam network decides on the prediction. Additionally, we present our empirical study on uncertainty estimation of SimSiam network by augmenting it with DbFF framework [28].

6.2.1 Experimental Setup

For all the experiments conducted, we utilize ResNet18 architecture as our backbone DNN as it is lightweight and popular in CAD systems. We use Vanilla ResNet18 for supervised learning as a baseline to ensure a fair comparison. To initialize the network weights, we used ImageNet pretrained weights for all our experiments. We also employ the same hyperparameters, e.g., initial LR, LR decay, batch size, and epochs, unless otherwise stated. Alongside that, we use the same optimizer, augmentations, and activation functions for all our experiments for a fair comparison. For hyperparameters and the optimizer, we follow a similar setting as presented in [48], except batch size. Due to computational constraints, we set batch size as 32 for all our experiments instead of 512. We run each experiment five times and present the mean values.

6.2.2 How well can SimSiam learn representations from CXR images?

To evaluate the efficacy of SimSiam network on COVIDx dataset, we trained the network with training data without the annotations. Then we freeze the encoder network and fine-tune a single layer perceptron network as prediction head. Finally, we feed test samples on to the $encoder + prediction\ head$ to calculate the accuracy. On the other hand, we train the same $encoder + prediction\ head$ in a supervised manner. Bar chart depicted in Figure 6.4 presents the accuracy from each experiment side by side. We included the untrained ResNet18 initialized with ImageNet pretraining weights to highlight how training on task domain data can help learning. As we can observe, the self-supervised SimSiam ResNet18

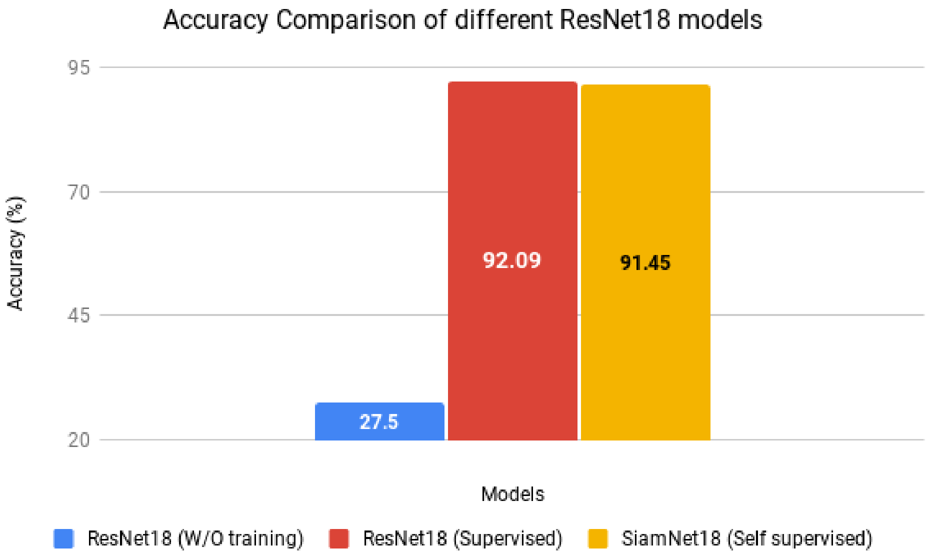


Figure 6.4 Performance comparison of *encoder + prediction head* model under different training scenarios on COVIDx Dataset. Please note, all models are initialized with ImageNet pretraining weights.

(SiamNet18) achieves very competitive performance compared to the supervised ResNet18, especially considering SiamNet18 was trained without any labels then only the prediction head was finetuned. Please note, we trained SiamNet18 for 800 epochs whereas we trained supervised ResNet18 for 200 epochs with other hyperparameters same. Self-supervised learning are proven to learn better with larger batch size and epochs, on the other hand, supervised learning tends to overfit with longer training. Due to limited computing resources we did not experiment with larger batch sizes.

6.2.3 Visualizing the Gradient Activations

This promising result inspires us to dig deeper into understanding the causality of the predictions made by SiamNet18. Hence, we investigated gradient mapping to locate the con-

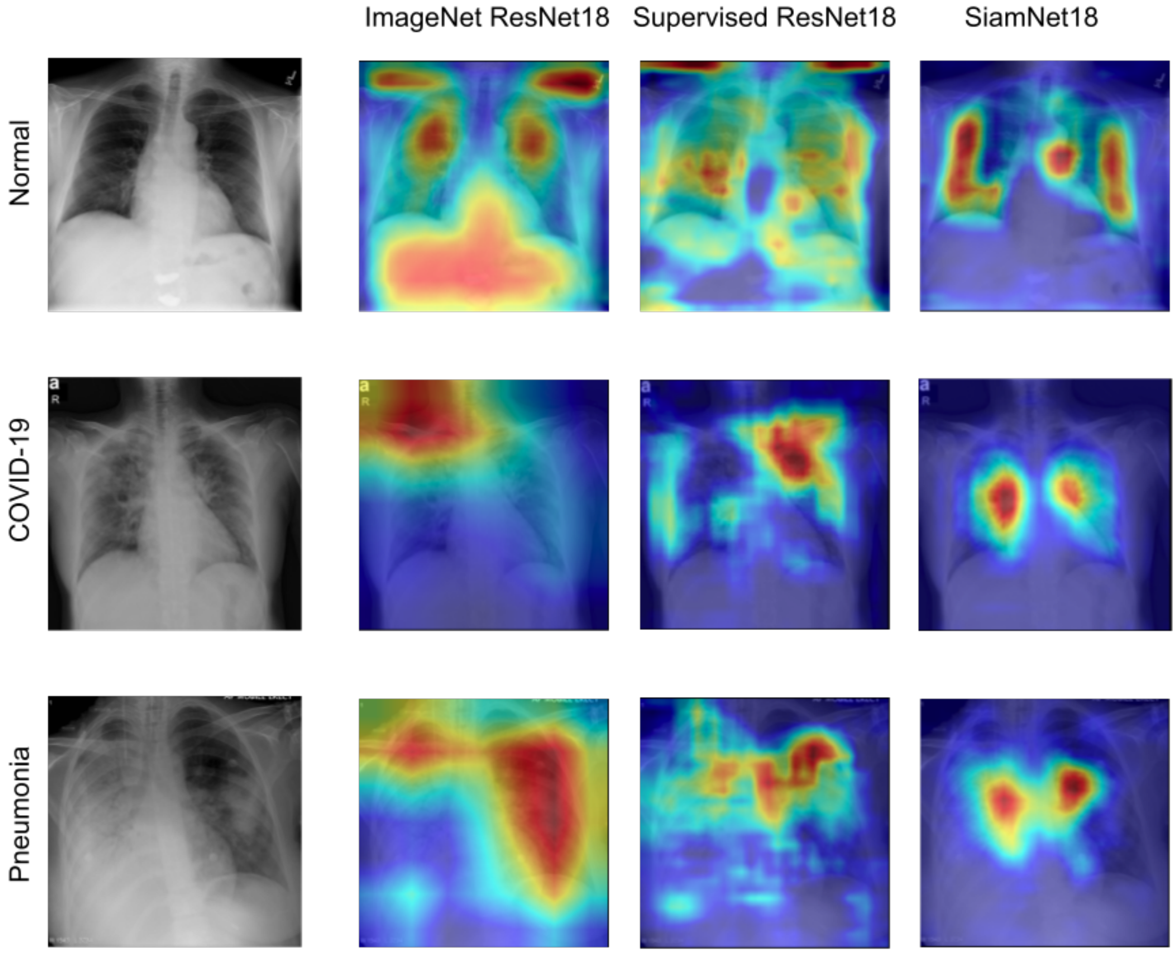


Figure 6.5 Visualization of gradient activation. Each row represent samples from each three classes of COVIDx dataset. Each columns from left to right present original images, activation plot from untrained ResNet18, supervised ResNet18, and the self-supervised SiamNet18 respectively.

tributing features of CXR images for a particular prediction. Figure 6.5 shows the gradient activation plotted on the corresponding CXR images by supervised and self-supervised methods (ResNet18 and SiamNet18). We employed GradCam++ [98] to generate these visualizations.

As shown in Figure 6.5, out-of-the-box ImageNet pretrained models cannot identify the

Table 6.1 Error (±confidence interval) on different abstention rate on COVIDx dataset. Please note, both networks are initialized with ImageNet pretraining weights.

	Models	
Abstention Rate	ResNet18 (Supervised)	SiamNet18 (Self-supervised)
0%	7.91 ±0.1%	8.64 ±0.06%
5%	6.04 ±0.07%	8.25 ±0.08%
10%	4.75 ±0.09%	7.93 ±0.09%
15%	3.38 ±0.08%	7.72 ±0.07%
20%	2.39 ±0.1%	7.64 ±0.08%
25%	2.04 ±0.05%	7.56 ±0.1%
30%	1.18 ±0.09%	7.41 ±0.08%

lung as the important part of the CXR images. Supervised ResNet18, on the other hand, can locate portions of the lung as a source of prediction. But, the gradient activation is scattered and noisy. An interesting observation we can make here is that for SiamNet18, each of the CXR images is highlighted to the portions that are the actual contributing factors. For example, if we observe the sample with ground truth as Normal, the SimSiam network actually highlighted the dark portions of both lungs which indicate that these lungs are healthy. Similarly, for the COVID-19 positive CXR image, the SimSiam network actually looks at the ground glass opacities on the lungs. Bacterial pneumonia infection can be identified when one of the lobes is opaque while the other is not. The SiamNet18 accurately highlights portions of both lungs to predict it as Pneumonia infection.

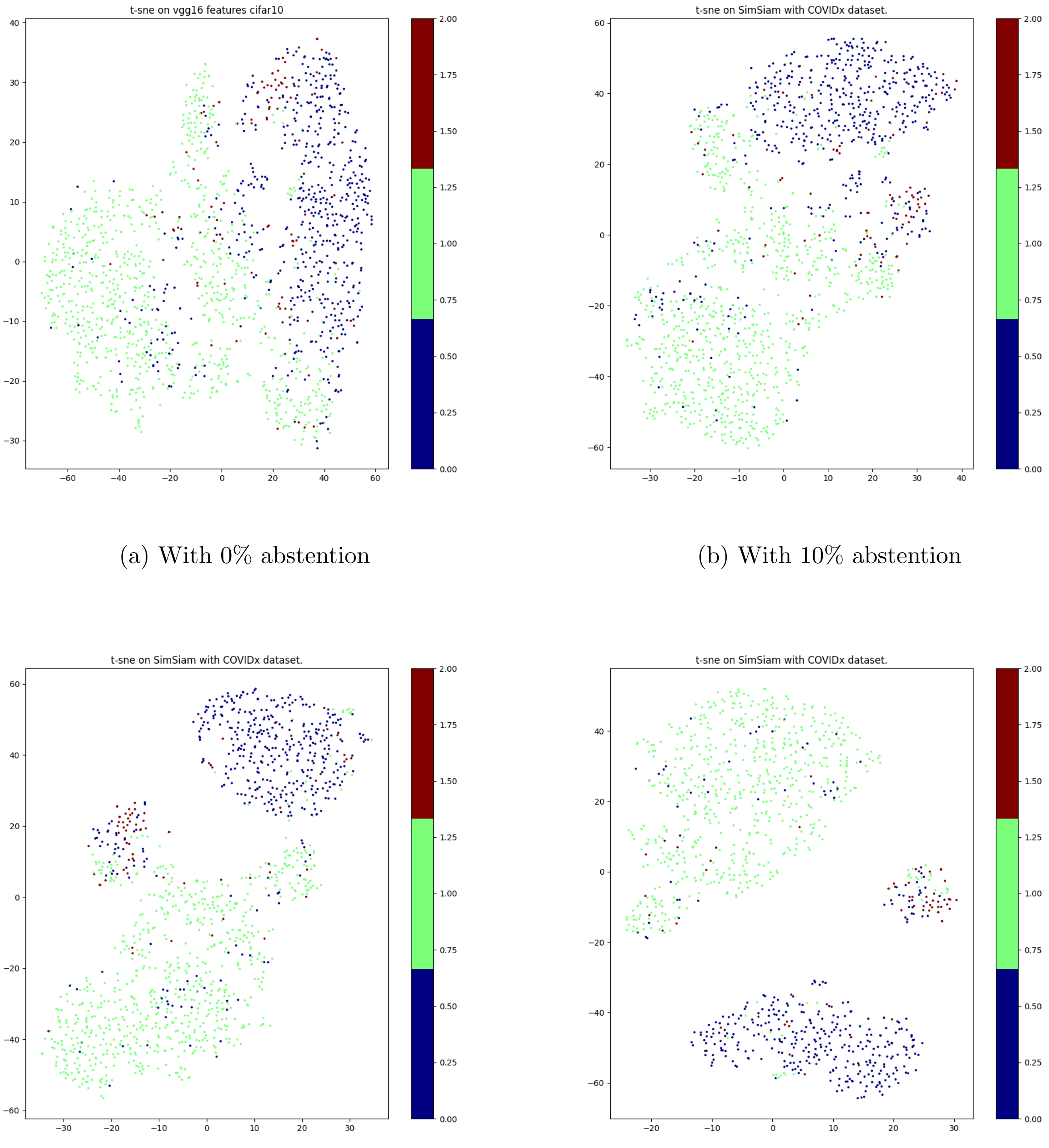
6.2.4 Abstaining Confusing Samples from SimSiam Predictions

The uncertainty of DNN predictions is a major contributing factor for Computer-Aided Systems not being deployed widely in practice. The SimSiam network also suffers from the issue, even though it can learn useful representations from data. To mitigate the issue of predictive uncertainty, we augment the SimSiam network with the DbFF framework [28] that abstains from uncertain predictions during inference. DbFF framework relies heavily on the representation learned from the data. Even though supervised ResNet18 and the SiamNet18 achieve similar performance on accuracy metric, the performance of DbFF contrasts highly among these two methods as shown in Table 6.1. DbFF could only achieve 14% improvement with 30% data abstention on SiamNet18, we can achieve close to 99% accuracy on supervised ResNet18.

This surprising observation leads us to investigate the representations learned by SiamNet18 visually. We have utilized tSNE [96] to plot the high dimensional representations into a Cartesian plane. Figure 6.6 depicts the data distribution over varying abstention rate. As we can observe, samples from different classes are disseminated throughout the plot which causes the degraded performance of DbFF framework on the SimSiam network. As we abstrain more, we start to observe more tight clusters with less contamination. This plot provides us insight that the representation learning still need to be more robust for SimSiam networks on real-world use cases even though we achieve comparable performance. We suspect that the class imbalance of the dataset may have lead to this issue. More deeper investigation is required to fully understand the discrepancy.

6.3 Conclusion

Self-supervised learning has proven to be competitive with its supervised counterpart on benchmark datasets in recent times. However, very limited work has been done on self-supervised learning on real-world use cases. Self-supervised learning has a significant advan-



(a) With 0% abstention

(b) With 10% abstention

(c) With 20% abstention

(d) With 30% abstention

Figure 6.6 tSNE plot of representations learned by SimSiam network from COVIDx dataset.

tage over supervised learning, it does not require a large annotated data pool to train on. In most real-world scenarios, annotated data is a crucial bottleneck, which can be overcome by utilized self-supervised learning. From this motivation, in this work we evaluated the state-of-the-art contrastive loss-based self-supervise learning network, SimSiam, on a life-critical real-world use case of COVID-19 detection from CXR images. We also visualized the activation map to observe the causality of the network. We augmented the SimSiam network by DbFF framework to abstain from confusing samples during test time. These experimentations have provided proof that SimSiam networks can learn useful representations for detecting pathogen infections even without the labels. These outcomes are very promising and demand further in-depth investigations to locate the contributing factors for SiamNet18 to learning representations that are scattered across. Augmentation can play a major factor for the SimSiam network. Generating domain-compatible augmentation is a major challenge that is still an open problem. We leave these as future research directions.

CHAPTER 7
Conclusion

In this research path, we focus on issues associated with the data-driven learning paradigm. Data represents the core of the success of DNNs. While investigating issues that hinder the performance of DNNs when dealing with real-world applications, we have come to realize the shortcomings of existing methodologies. We have faced several major challenges and proposed methods to overcome them. Specifically, we have proposed a method to utilize RGB data instead of RGB-D data to achieve state-of-the-art performance on Human Action Recognition. We have also investigated the issue of uncertainty and risk of DNNs and attempted to mitigate that by proposing a novel density-based framework to filter noisy training data while at the same time abstain from inferring on confusing samples. We also tested the effectiveness of uncertainty mitigation methodologies on the life-critical application of DNNs. We have proposed feature filtering methods to be utilized with our density-driven framework to further reduce the risk of uncertain predictions in detecting COVID-19 positive patients. To further investigate data efficient learning, we explored self-supervised siamese networks to learn representations without annotated data. Throughout this journey we have explored different aspects of data driven learning in depth, and solved some exciting challenges of most crucial aspect of Deep Learning, data. From our research, several new intriguing problems have surfaced, where a lot of attention is required.

- Augmentation in contrastive siamese networks is a crucial element for successful training of such networks. There have been a number of works proposed in the literature that generates augmentation based on the task at hand. However, majority of these automated augmentation methods rely on data labels [99]. There are other works that aim to utilize adversarial noise to augment data for contrastive learning [100]. More in depth investigation in this field is required to close the current performance gap between the supervised and self-supervised methods.

- The state-of-the-art contrastive learning frameworks utilize all available data during training. However, prior work has shown that all data at once may cause more harm than good. Some hard to learn samples may distract training at an early state, while there might be samples that does not carry valuable information for the network to learn from [101]. Hence, we can adopt active meta learning with contrastive siamese networks to boost data efficiency.

- We have explored denoising the training data pool and abstaining confusing samples on supervised settings. However, supervised methods are more vulnerable to adversarial attacks and that may reduce the representation learning capability of such networks. On the other hand, self-supervised learning methods are more robust to adversarial attacks and theoretically would perform better than supervised encoders. More research work in this aspect would help clarify unsolved questions in this regard.

7.1 List of Publications

7.1.1 Published

- Kamath, Goutham, Pavan Agnihotri, Maria Valero, Krishanu Sarker, and Wen-Zhan Song.“Pushing analytics to the edge.” In 2016 IEEE Global Communications Conference (GLOBECOM), pp. 1-6. IEEE, 2016.

- Masoud, Mohamed E., Krishanu Sarker, Saeid Belkasim, and Iman Chahine. “Automatically generated semantic tags of art images.” In IEEE International Conference on Signal and Image Processing Applications (ICSIPA). 2017.

- Sarker, Krishanu, Mohamed Masoud, Saeid Belkasim, and Shihao Ji. “Towards Robust Human Activity Recognition from RGB Video Stream with Limited Labeled Data.” In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 145-151. IEEE, 2018.

- Sarker, Krishanu, Sharbani Pandit, Anupam Sarker, Saeid Belkasim, and Shihao Ji, “Reducing Risk and Uncertainty of Deep Neural Networks on Diagnosing COVID-19 Infection”, Trustworthy AI for Healthcare (TAIH) workshop, AAAI, 2021.

- Sarker, Krishanu, Sharbani Pandit, Anupam Sarker, Saeid Belkasim, and Shihao Ji, “Towards Reliable and Trustworthy Computer-Aided Diagnosis Predictions: Diagnosing COVID-19 from X-Ray Images”, The ACM Conference on Health, Inference, and Learning (ACM CHIL), 2021.

7.1.2 Under Review

- Krishanu Sarker, Xiulong Yang, Yang Li,Saeid Belkasim, and Shihao Ji. “A unified plug-and-play framework for effective data denoising and robust abstention.” IEEE International Conference on Image Processing (ICIP), 2021.

7.1.3 Work in Progress

- Krishanu Sarker, Saeid Belkasim, and Shihao Ji. “Uncertainty Aware Self-Supervised Contrastive Learning towards Efficient Representation Learning: A Real-World Scenario Evaluation.” to be submitted.

REFERENCES

[1] Geifman, Y., El-Yaniv, R.: Selectivenet: A deep neural network with an integrated reject option. arXiv preprint arXiv:1901.09192 (2019)

[2] Wang, L., Wong, A.: Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. arXiv preprint arXiv:2003.09871 (2020)

[3] Chen, C., Jafari, R., Kehtarnavaz, N.: Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In: Image Processing (ICIP), 2015 IEEE International Conference on, IEEE (2015) 168–172

[4] Zhang, B., Yang, Y., Chen, C., Yang, L., Han, J., Shao, L.: Action recognition using 3d histograms of texture and a multi-class boosting classifier. IEEE Transactions on Image Processing **26** (2017) 4648–4660

[5] Wang, P., Li, Z., Hou, Y., Li, W.: Action recognition based on joint trajectory maps using convolutional neural networks. In: Proceedings of the 2016 ACM on Multimedia Conference, ACM (2016) 102–106

[6] Hou, Y., Li, Z., Wang, P., Li, W.: Skeleton optical spectra based action recognition using convolutional neural networks. IEEE Transactions on Circuits and Systems for Video Technology (2016)

[7] Li, C., Hou, Y., Wang, P., Li, W.: Joint distance maps based action recognition with convolutional neural networks. IEEE Signal Processing Letters **24** (2017) 624–628

[8] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115** (2015) 211–252

[9] Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. (2015)

[10] Sarker, K., Masoud, M., Belkasim, S., Ji, S.: Towards robust human activity recognition from rgb video stream with limited labeled data. In: ICMLA. (2018)

[11] Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S.Y., Sainath, T.: Deep learning for audio signal processing. IEEE Journal of Selected Topics in Signal Processing **13** (2019) 206–219

[12] Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)

[13] Erickson, B.J., Korfiatis, P., Akkus, Z., Kline, T.L.: Machine learning for medical imaging. Radiographics **37** (2017) 505–515

[14] Bellili, A., Gilloux, M., Gallinari, P.: An hybrid mlp-svm handwritten digit recognizer. In: Proceedings of sixth international conference on document analysis and recognition, IEEE (2001) 28–32

[15] Gibney, E.: Google ai algorithm masters ancient game of go. Nature News **529** (2016) 445

[16] Jansen, A., Gemmeke, J.F., Ellis, D.P., Liu, X., Lawrence, W., Freedman, D.: Large-scale audio event discovery in one million youtube videos. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE (2017) 786–790

[17] Elliott, A.: The culture of AI: Everyday life and the digital revolution. Routledge (2019)

[18] Das, A., Rad, P.: Opportunities and challenges in explainable artificial intelligence (xai): A survey. arXiv preprint arXiv:2006.11371 (2020)

[19] Turaga, P., Chellappa, R., Subrahmanian, V.S., Udrea, O.: Machine recognition of human activities: A survey. IEEE Transactions on Circuits and Systems for Video technology **18** (2008) 1473

[20] Poppe, R.: A survey on vision-based human action recognition. Image and vision computing **28** (2010) 976–990

[21] Cheng, G., Wan, Y., Saudagar, A.N., Namuduri, K., Buckles, B.P.: Advances in human action recognition: A survey. arXiv preprint arXiv:1501.05964 (2015)

[22] Herath, S., Harandi, M., Porikli, F.: Going deeper into action recognition: A survey. Image and Vision Computing **60** (2017) 4–21

[23] Li, W., Wang, L., Li, W., Agustsson, E., Van Gool, L.: Webvision database: Visual learning and understanding from web data. arXiv preprint arXiv:1708.02862 (2017)

[24] Nettleton, D.F., Orriols-Puig, A., Fornells, A.: A study of the effect of different types of noise on the precision of supervised learning techniques. Artificial intelligence review **33** (2010) 275–306

[25] Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al.: A closer look at memorization in deep networks. arXiv preprint arXiv:1706.05394 (2017)

[26] Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533 (2016)

[27] Frénay, B., Verleysen, M.: Classification in the presence of label noise: a survey. IEEE transactions on neural networks and learning systems **25** (2013) 845–869

[28] Sarker, K., Yang, X., Li, Y., Belkasim, S., Ji, S.: A unified plud-and-play framework for effective data denoising and robust abstention. (2020)

[29] Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local svm approach. In: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. Volume 3., IEEE (2004) 32–36

[30] Zhang, Z., Hu, Y., Chan, S., Chia, L.T.: Motion context: A new representation for human action recognition. Computer Vision–ECCV 2008 (2008) 817–829

[31] Liu, J., Shah, M.: Learning human actions via information maximization. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE (2008) 1–8

[32] Bregonzio, M., Gong, S., Xiang, T.: Recognising action as clouds of space-time interest points. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009) 1948–1955

[33] Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. International journal of computer vision **103** (2013) 60–79

[34] Baccouche, M., Mamalet, F., Wolf, Christian Garcia, C., Baskurt, A.: Sequential deep learning for human action recognition. In: International Workshop on Human Behavior Understanding, Springer (2011) 29–39

[35] Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence **35** (2013) 221–231

[36] Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. (2014) 568–576

[37] Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: CVPR. (2017)

[38] Thulasidasan, S., Bhattacharya, T., Bilmes, J., Chennupati, G., Mohd-Yusof, J.: Combating label noise in deep learning using abstention. arXiv preprint arXiv:1905.10964 (2019)

[39] Liu, S., Niles-Weed, J., Razavian, N., Fernandez-Granda, C.: Early-learning regularization prevents memorization of noisy labels. arXiv preprint arXiv:2007.00151 (2020)

[40] Doi, K.: Overview on research and development of computer-aided diagnostic schemes. In: Seminars in Ultrasound, CT and MRI. Volume 25., Elsevier (2004) 404–410

[41] Zumla, A., Niederman, M.S.: The explosive epidemic outbreak of novel coronavirus disease 2019 (covid-19) and the persistent threat of respiratory tract infectious diseases to global health security. Current Opinion in Pulmonary Medicine (2020)

[42] Chen, J., Li, K., Zhang, Z., Li, K., Yu, P.S.: A survey on applications of artificial intelligence in fighting against covid-19. arXiv preprint arXiv:2007.02202 (2020)

[43] Cleverley, J., Piper, J., Jones, M.M.: The role of chest radiography in confirming covid-19 pneumonia. bmj **370** (2020)

[44] Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: Youtube-8m: A large-scale video classification benchmark. arXiv preprint arXiv:1609.08675 (2016)

[45] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning, PMLR (2020) 1597–1607

[46] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 9729–9738

[47] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. arXiv preprint arXiv:2006.09882 (2020)

[48] Chen, X., He, K.: Exploring simple siamese representation learning. arXiv preprint arXiv:2011.10566 (2020)

[49] Chen, C., Jafari, R., Kehtarnavaz, N.: A survey of depth and inertial sensor fusion for human action recognition. Multimedia Tools and Applications **76** (2017) 4405–4425

[50] Liu, M., Liu, H., Chen, C.: Enhanced skeleton visualization for view invariant human action recognition. Pattern Recognition **68** (2017) 346–362

[51] Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 1110–1118

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

[53] Liu, Z., Wang, Z., Liang, P.P., Salakhutdinov, R.R., Morency, L.P., Ueda, M.: Deep gamblers: Learning to abstain with portfolio theory. In: NIPS. (2019) 10623–10633

[54] Cordella, L.P., De Stefano, C., Tortorella, F., Vento, M.: A method for improving classification reliability of multilayer perceptrons. IEEE Transactions on Neural Networks **6** (1995) 1140–1147

[55] De Stefano, C., Sansone, C., Vento, M.: To reject or not to reject: that is the question-an answer in case of neural classifiers. IEEE Transactions on Systems, Man, and Cybernetics **30** (2000) 84–94

[56] Bartlett, P.L., Wegkamp, M.H.: Classification with a reject option using a hinge loss. Journal of Machine Learning Research **9** (2008) 1823–1840

[57] Grandvalet, Y., Rakotomamonjy, A., Keshet, J., Canu, S.: Support vector machines with a reject option. In: NIPS. (2009)

[58] Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., Belongie, S.: Learning from noisy large-scale datasets with minimal supervision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 839–847

[59] Ostyakov, P., Logacheva, E., Suvorov, R., Aliev, V., Sterkin, G., Khomenko, O., Nikolenko, S.I.: Label denoising with large ensembles of heterogeneous neural networks. In: ECCV. (2018)

[60] Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In: Advances in Neural Information Processing Systems. (2018) 7167–7177

[61] Liang, S., Li, Y., Srikant, R.: Enhancing the reliability of out-of-distribution image detection in neural networks. arXiv preprint arXiv:1706.02690 (2017)

[62] Shafaei, A., Schmidt, M., Little, J.J.: A less biased evaluation of out-of-distribution sample detectors. arXiv preprint arXiv:1809.04729 (2018)

[63] Hendrycks, D., Mazeika, M., Dietterich, T.: Deep anomaly detection with outlier exposure. arXiv preprint arXiv:1812.04606 (2018)

[64] Castiglioni, I., Ippolito, D., Interlenghi, M., Monti, C.B., Salvatore, C., Schiaffino, S., Polidori, A., Gandola, D., Messa, C., Sardanelli, F.: Artificial intelligence applied on chest x-ray can aid in the diagnosis of covid-19 infection: a first experience from lombardy, italy. medRxiv (2020)

[65] Zhang, J., Xie, Y., Li, Y., Shen, C., Xia, Y.: Covid-19 screening on chest x-ray images using deep learning based anomaly detection. arXiv preprint arXiv:2003.12338 (2020)

[66] Ayhan, M.S., Kuehlewein, L., Aliyeva, G., Inhoffen, W., Ziemssen, F., Berens, P.: Expert-validated estimation of diagnostic uncertainty for deep neural networks in diabetic retinopathy detection. Medical Image Analysis (2020) 101724

[67] Leibig, C., Allken, V., Ayhan, M.S., Berens, P., Wahl, S.: Leveraging uncertainty information from deep neural networks for disease detection. Scientific reports **7** (2017) 1–14

[68] Ding, Y., Liu, J., Xu, X., Huang, M., Zhuang, J., Xiong, J., Shi, Y.: Uncertainty-aware training of neural networks for selective medical image segmentation. In: Medical Imaging with Deep Learning. (2020)

[69] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 4681–4690

[70] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 2536–2544

[71] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. arXiv preprint arXiv:1406.2661 (2014)

[72] Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. (2017) 2223–2232

[73] Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European conference on computer vision, Springer (2016) 649–666

[74] Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE international conference on computer vision. (2015) 1422–1430

[75] Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: European conference on computer vision, Springer (2016) 69–84

[76] Bromley, J., Bentz, J.W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., Säckinger, E., Shah, R.: Signature verification using a "siamese" time delay neural network. International Journal of Pattern Recognition and Artificial Intelligence **7** (1993) 669–688

[77] Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2014) 1701–1708

[78] Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: ICML deep learning workshop. Volume 2., Lille (2015)

[79] Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: European conference on computer vision, Springer (2016) 850–865

[80] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9** (1997) 1735–1780

[81] Sarle, W.S.: Neural networks and statistical models. (1994)

[82] Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks **5** (1994) 157–166

[83] Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Networks **18** (2005) 602–610

[84] He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. (2015) 1026–1034

[85] Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning **4** (2012) 26–31

[86] Hsu, W.N., Zhang, Y., Lee, A., Glass, J.: Exploiting depth and highway connections in convolutional recurrent deep neural networks for speech recognition. cell **50** (2016) 1

[87] Lei, J., Li, G., Li, S., Tu, D., Guo, Q.: Continuous action recognition based on hybrid cnn-ldcrf model. In: Image, Vision and Computing (ICIVC), International Conference on, IEEE (2016) 63–69

[88] Liu, L., Shao, L., Li, X., Lu, K.: Learning spatio-temporal representations for action recognition: A genetic programming approach. IEEE transactions on cybernetics **46** (2016) 158–170

[89] Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd. Volume 96. (1996) 226–231

[90] Otsu, N.: A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics **9** (1979) 62–66

[91] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[92] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778

[93] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. (2009)

[94] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. (2011)

[95] Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)

[96] Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9** (2008) 2579–2605

[97] Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in neural information processing systems. (2017) 5574–5584

[98] Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE (2018) 839–847

[99] Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:1805.09501 (2018)

[100] Tamkin, A., Wu, M., Goodman, N.: Viewmaker networks: Learning views for unsupervised representation learning. arXiv preprint arXiv:2010.07432 (2020)

[101] Al-Shedivat, M., Li, L., Xing, E., Talwalkar, A.: On data efficiency of meta-learning. In: International Conference on Artificial Intelligence and Statistics, PMLR (2021) 1369–1377