

UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Ciencias Matemáticas y Computacionales

TÍTULO: Computer-Assisted Mispronunciation Detection System for L2 Kichwa Speech

Trabajo de integración curricular presentado como requisito para la obtención del título de Ingeniero en Tecnologías de la Información

Autor:

Velasco Silva Ricardo Isaías

Tutor:

PhD - Fonseca Delgado Rigoberto Salomón

Co-Tutor:

PhD - Morales Navarrete Diego Fabián

Urcuquí, Marzo 2024

Autoría

Yo, VELASCO SILVA RICARDO ISAÍAS, con cédula de identidad 175027344-1,

declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, defini-

ciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos

y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el/la au-

tor/a del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos

de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, Marzo 2024.

Velasco Silva Ricardo Isaías

CI: 175027344-1

Autorización de publicación

Yo, VELASCO SILVA RICARDO ISAÍAS, con cédula de identidad 175027344-1,

cedo a la Universidad de Investigación de Tecnología Experimental Yachay, los derechos

de publicación de la presente obra, sin que deba haber un reconocimiento económico por

este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser

cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente

con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este

trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto

en el Art. 144 de la Ley Orgánica de Educación

Urcuquí, Marzo 2024.

Velasco Silva Ricardo Isaías

CI: 175027344-1

Resumen

Se ha realizado una evaluación inicial y experimental de un sistema de detección de errores de pronunciación para la lengua kichwa. Se empleó arquitecturas de redes neuronales convolucionales preentrenadas para clasificar espectrogramas de palabras pronunciadas con exactitud y palabras pronunciadas con inexactitud. El modelo inicial, conocido como un modelo basado en características de redes convolucionales, extrae características de las capas totalmente conectadas. A continuación, emplea una técnica de selección de características para separar las características discriminativas de las no discriminativas. Por último, estas características se clasifican mediante un clasificador KNN. El segundo modelo, basado en el aprendizaje por transferencia con redes neuronales convolucionales (CNN), utiliza el conocimiento de las capas convolucionales y adapta la capa clasificadora para la clasificación binaria, distinguiendo entre audios bien pronunciados y mal pronunciados. En cuanto al conjunto de datos utilizado, se construyeron dos conjuntos de datos que se utilizaron en este estudio: un conjunto de datos con palabras en Kichwa y palabras sintéticas, y el mismo pero con palabras sintéticas para el entrenamiento. En conclusión, el método basado en el aprendizaje por transferencia es superior al método basado en características en ambos conjuntos de datos. Concretamente, AlexNet con ajuste de hiperparámetros alcanza 0,90 y 0,92 en la métrica de valor predictivo equilibrado en ambos conjuntos de datos, respectivamente.

Palabras Clave:

modelo basado en características, modelo basado en aprendizaje por transferencia, redes neuronales convolucionales, ajuste de hiperparámetros, recopilación de conjuntos de datos, lengua Kichwa

Abstract

An initial and experimental evaluation of a mispronunciation detection system was developed for the Kichwa language. The study implemented pretrained convolutional neural network architectures to classify spectrograms of accurately pronounced and inaccurately pronounced words. The initial model, known as the CNN feature-based model, extracts features from the fully connected layers. It then employs a feature selection technique to separate discriminative features from non-discriminative ones. Finally, these features are classified using a KNN classifier. The second model, which is based on transfer learning with convolutional neural networks (CNNs), uses the knowledge from convolutional layers and adapts the classifier layer for binary classification, distinguishing between well-pronounced and mispronounced audios. When referring to the used dataset, two datasets were constructed and used in this study: a dataset with Kichwa words and synthetic words, and the same but with synthetic words for training. In conclusion, the CNN transfer learning-based method is superior to the CNN feature-based method in both datasets. Concretely, AlexNet with hyperparameter tuning achieves 0.90 and 0.92 in the balanced predictive value metric in both datasets, respectively.

Keywords:

features-based model, transfer learning-based model, convolutional neural networks, hyperparameter tuning, dataset collection, Kichwa language

Contents

Resumen							
Abstract Contents			iv				
			v				
Li	st of	Tables	vii				
Li	st of	Figures	х				
1	Intr	roduction	1				
	1.1	Background	1				
	1.2	Problem statement	4				
	1.3	Objectives	4				
2	The	oretical Framework	5				
	2.1	Artificial Intelligence	5				
	2.2	Machine Learning	7				
		2.2.1 Data	8				
		2.2.2 Types	10				
		2.2.3 Tasks	10				
3	Stat	te of the Art	17				
4	Methodology						
	4.1	Available Datasets	20				
	4.2	Dataset Construction	22				
	43	Dataset Labelling	23				

	4.4	Datase	et Splitting and Audio Distribution	24
	4.5	Prepro	ocessing and Spectrogram Conversion	25
	4.6	Selecte	ed CNN Architectures	25
		4.6.1	AlexNet	25
		4.6.2	VGG19	27
		4.6.3	ResNet50	27
	4.7	Data A	Augmentation	28
	4.8	CNN I	Features-Based Model	29
		4.8.1	Feature Extraction	29
		4.8.2	Dataset Discretization	30
		4.8.3	Statistical-based Feature Selection Methods	30
		4.8.4	Classification and Hyperparameter Tuning	31
	4.9	CNN T	Transfer learning-based Model	31
		4.9.1	Importing Pre-trained CNN Network	31
		4.9.2	Modifying Final Layers	32
		4.9.3	Hyperparameter Tuning	32
	4.10	Model	Performance Metrics	32
		4.10.1	PPV (Positive Predictive Value)	33
		4.10.2	NPV (Negative Predictive Value)	33
		4.10.3	Custom Metric: BPV (Balanced Predictive Value)	34
5	Res	ults an	nd Discussion	35
	5.1	Datase	et Construction	35
	5.2	Syntet	ic Data	36
	5.3	Datase	et Labelling	37
	5.4	Model	's Testing	37
		5.4.1	CNN Features-based Model	38
		5.4.2	CNN Transfer Learning-based Model	39
		5.4.3	Observations from Table 5.4	39
6	Con	clusio	ns	42
	6.1	Observ	vations and Limitations	42

Bi	Bibliography		
Aı	Appendices		
\mathbf{A}	CNN Features-based Model Hyperparameter Tuning and Testing Results: Diagrams and Tables		
В	B CNN Transfer Learning-based Model Hyperparameter Tuning and Test- ing Results: Diagrams and Tables		
\mathbf{C}	Exemplary Code		
	C.1	Dataset Splitting	77
	C.2	Audio Dataset Augmentation, Resampling, and Conversion	78
	C.3	CNN Features-based Model	82
	C.4	CNN Transfer Learning-based AlexNet Hyperparameter Tuning	91
	C.5	CNN Transfer Learning-based AlexNet Testing	98

List of Tables

4.1	Range of Parameters to be Chosen for Hyperparameter Tuning	33
5.1	Number of participants in the dataset	36
5.2	Distribution of words in Kichwa dataset	36
5.3	Syntetic Participants Details	37
5.4	Best Models' Performance on Testing Datasets. Each model has been se-	
	lected according to validation metric.	41
A.1	CNN Features-based Hyperparameter Tuning on k and Testing in Original	
	Dataset	55
A.2	CNN Features-based Hyperparameter Tuning on k and Testing in Synthetic	
	Dataset	56
A.3	CNN Features-based Hyperparameter Tuning with Feature Selection and	
	Testing in Original Dataset	56
A.4	CNN Features-based Hyperparameter Tuning with Feature Selection and	
	Testing in Synthetic Dataset	56
B.1	AlexNet Hyperparameter Tuning on Original Dataset: best results per trial	
	using validation loss as metric	62
B.2	AlexNet Hyperparameter Tuning on Original Dataset: best results per trial	
	using BPV as metric	63
В.3	AlexNet Hyperparameter Tuning on Synthetic Dataset: best results per trial	
	using validation loss as metric	64
B.4	AlexNet Hyperparameter Tuning on Synthetic Dataset: best results per trial	
	using BPV as metric.	65

B.5	VGG19 Hyperparameter Tuning on Original Dataset: best results per trial	
	using validation loss as metric	66
B.6	VGG19 Hyperparameter Tuning on Original Dataset: best results per trial	
	using BPV as metric	67
B.7	VGG19 Hyperparameter Tuning on Synthetic Dataset: best results per trial	
	using validation loss as metric	68
B.8	VGG19 Hyperparameter Tuning on Synthetic Dataset: best results per trial	
	using BPV as metric	69
B.9	ResNet50 Hyperparameter Tuning on Original Dataset: best results per trial	
	using validation loss as metric	70
B.10	ResNet50 Hyperparameter Tuning on Original Dataset: best results per trial	
	using BPV as metric	71
B.11	ResNet50 Hyperparameter Tuning on Synthetic Dataset: best results per	
	trial using validation loss as metric	72
B.12	ResNet50 Hyperparameter Tuning on Synthetic Dataset: best results per	
	trial using BPV as metric	73
B.13	AlexNet Mispronunciation Results in Original Dataset	74
B.14	AlexNet Mispronunciation Results in Synthetic Dataset	74
B.15	VGG19 Mispronunciation Results in Original Dataset	75
B.16	VGG19 Mispronunciation Results in Synthetic Dataset	75
B.17	ResNet50 Mispronunciation Results in Original Dataset	76
B.18	ResNet50 Mispronunciation Results in Synthetic Dataset	76

List of Figures

2.1	Artificial intelligence Diagram [1]	6
2.2	Multi-layer Perceptron Diagram with 16 inputs, 2 hidden layers and 3 outputs	14
2.3	CNN Architecture Diagram	15
2.4	The transfer learning process involves transferring knowledge from a pre-	
	trained model to a new deep learning model $[1]$	16
4.1	Textgrid with manual annotations from a L2-ARCTIC's sample audio file	
	[2]. Top to bottom: speech waveform, spectrogram, words, phonemes, error	
	tags, and comments from the annotator	20
4.2	Textgrid with manual annotations from a TIMIT's sample audio file [3].	
	Top to bottom: speech waveform, spectrogram, phonemes, words	21
4.3	Folder directory labelling	24
4.4	Spectrogram representation of the word "allku" pronounced by a non-native	
	female speaker	25
4.5	AlexNet Architecture Diagram [4]	26
4.6	VGG19 Architecture Diagram [5]	27
4.7	ResNet50 Architecture Diagram	28
A.1	KNN Classifier Hyperparameter Tuning in Original Dataset	57
A.2	KNN Classifier Hyperparameter Tuning in Synthetic Dataset	58
A.3	Feature Selection Hyperparameter Tuning in Original Dataset	59
Δ 4	Feature Selection Hyperparameter Tuning in Synthetic Dataset	60

Chapter 1

Introduction

1.1 Background

The global population of indigenous people is estimated to be approximately 370 million individuals, residing in more than 90 nations worldwide [6]. Despite constituting just 5% of the global population, this group owns, inhabits, and/or uses 22% of the world's land area. A significant portion of the world's cultural diversity is represented by indigenous people, who speak a total of 7000 languages [7]. However, the enduring effects of historical inequality and marginalization have resulted in indigenous populations representing approximately 19% of the global impoverished demographic. These individuals experience a significantly reduced life expectancy, which has been calculated to be 20 years less than their non-indigenous counterparts worldwide. They encounter challenges such as the absence of official recognition regarding their lands, territories, and natural resources, delayed access to public investments in essential services and infrastructure, and numerous obstacles hindering their full engagement in the formal economy, access to justice, and participation in political processes and decision-making [8].

Alongside the challenges faced by indigenous communities, their languages suffer from similar issues. Many languages are currently facing the imminent threat of extinction, and various factors contribute to this pressing concern [9]:

• Implementation of state policy. Several governmental institutions have implemented initiatives with the objective of eradicating indigenous languages with legal regulations that prohibit their usage. As a result, numerous nations deny the exis-

tence of indigenous populations and classify their languages as mere dialects, thus undervaluing them, devaluing them compared to national languages.

- Lack of support and attention. Because of their minority status, indigenous populations frequently encounter a lack of recognition and support from governmental initiatives aimed at preserving languages, resulting in the marginalization of their linguistic heritage. Furthermore, even though there may be initiatives, they may not be well executed and consequently be overlooked. In the Philippines, the government has started to implement mother-tongue-based instruction in the classrooms [9]. Nonetheless, due to the lack of available resources, including qualified teachers and appropriate learning materials, the effective instruction of indigenous children in their respective mother tongues may is hindered, and consequently, individuals acquire proficiency in another language rather than their own, ultimately leading to the deterioration of their native language skills.
- Obsolescence. Due to a long history of discriminatory policies and practices, a significant number of indigenous parents have made the decision to teach and speak in current languages with their children [9]. This is done with the intention of fostering an environment that is beneficial to their children's development in a globalized society. In most cases, indigenous children are currently unable to take part in conversations with their grandparents, who commonly use their mother tongue to communicate with each other, inevitably leading to the loss of oral traditions and expressions.

Nevertheless, the increasing global recognition of indigenous knowledge systems has restored optimism concerning the preservation and growth of indigenous languages in both oral and written formats. Numerous indigenous communities have already implemented autonomous mechanisms aimed at restoring their respective languages. Moreover, legislative provisions and measures have been proposed. For instance, in Ecuador, the following articles support the right to education for indigenous people:

• "Article 29 of the Constitution declares that 'the State shall guarantee the freedom of teaching, academic freedom in higher education, and the right of individuals to learn in their own language and cultural environment....." [10]

• "Article 57, number 14, of the Constitution agrees that the State must 'develop, strengthen and enhance the system of bilingual intercultural education, on the basis of quality criteria, from early childhood stimulation to the highest level, in accordance with cultural diversity, for the safeguarding and preserving identities in harmony with teaching and learning methodologies....." [10]

These articles guarantee the rights of communities, peoples, and nationalities, emphasizing the state's obligation to integrate education with an intercultural environment [10]. By doing so, Ecuador acknowledges the significance of fostering inclusivity and demonstrating respect for the diverse cultural backgrounds of its communities.

In the end, this legislative provision resulted in Ecuador having 1,834 bilingual intercultural educational institutions that educated 156,027 students in 2015–2016, as well as 13 educational institutions designated as guardians of the language and staffed by teachers who speak one of the indigenous languages [6]. Similarly, there are other countries that have implemented similar legislation, such as Argentina, Australia, Bolivia, Brazil, Chile, Guatemala, Honduras, Iran, Mexico, New Zealand, Norway, Sweden, and Venezuela [6].

Unfortunately, indigenous peoples must still overcome several challenges before breaking free from marginalization. One-third of the world's impoverished population are indigenous, and in many countries, indigenous rights laws conflict with laws pertaining to agriculture, land, conservation, forestry, mining and other sectors [9].

7% of the indigenous people in Ecuador use Kichwa and quite frequently it has been reduced to a peasant language [11]. In order to contribute to the preservation of the Kichwa language, Yachay Tech University proposed a community service project called "Kichwa Yachay App" whose head chairman is Sophia L. Cadoux from the Yachay Tech English Department. Formerly, this was a software engineering project aimed at developing a mobile application for learning and practicing the Kichwa language. As a result, all the students gave a presentation presented prototypes of their applications. For one of the projects, there was an open thesis statement to develop a computer-assisted mispronunciation system. The development of such system will provide valuable insights into the capabilities and limitations of the developed models and the developed data, allowing for further improvements and advancements in the field of language learning for Kichwa.

1.2 Problem statement

It is important to know that recently, the adaptability of Computer-Aided Language Learning (CALL) has drawn a lot of attention because it lets students improve their language skills at their own pace. To achieve this cause, it is necessary to use CALL technologies. By definition, CALL involves using computers and computer-based resources like the Internet, to present, reinforce, and assess learning. Specifically, for this work, we focus on Computer-Aided Pronunciation Training (CAPT), a subset of CALL that focuses on finding and correcting pronunciation mistakes made by non-native Kichwa language learners [12].

Deep learning technology is considered one of the hot topics within the areas of machine learning, artificial intelligence, data science, and analytics due to its learning capabilities from the given data [1]. The problem begins with the scarcity of data for the Kichwa language (see Section 4.1), to which two models for mispronunciation detection based on convolutional neural networks and spectrograms were developed. In addition, two Kichwa datasets were collected and proposed for the sake of the project and future research. These models will be subjected to testing and measurement using various evaluation metrics, such as positive and negative predictive values, to evaluate their performance.

1.3 Objectives

- Generate two datasets of Kichwa words from both native and non-native speakers.
- Implement two models for Kichwa mispronunciation detection by means of convolutional neural network architectures (AlexNet, VGG19, ResNet): CNN features-based and transfer learning-based models.
- Compare the performance of the proposed models with adequate metrics: positive, negative and balanced predictive value.
- Evaluate how the model performs whenever a word is well-pronounced or mispronounced, without restricting only to evaluating the accuracy of the model.

Chapter 2

Theoretical Framework

2.1 Artificial Intelligence

The renowned author Karel Capek first used the term "robot" in his play "R.U.R." (Rossum's Universal Robots) in 1921, and it originally meant a factory of biomechanical machines used as forced labor [13]. During the mid-20th century, Isaac Asimov popularized the term "robot" through a compilation of contemporary science-fiction short stories [13]. Even though the word was slightly new, during the Renaissance period, Leonardo da Vinci performed a comprehensive study of human anatomy with the purpose of guiding the design of his humanoid robot, Leonardo's robot. It was a mechanized knight automaton capable of assuming many postures, including standing, sitting, and gesturing with its arms, as well as articulating its head and jaw by using a combination of pulleys and cables [14]. Subsequently, the term "Artificial Intelligence", derived from the idea of a robot, appeared years later. John McCarthy coined the term "artificial intelligence" (AI) in 1955, defining it as "the science and engineering of making intelligent machines". He was very influential in the early development of AI. With his colleagues, he founded the field of AI in 1956 at a Dartmouth College conference on artificial intelligence. The conference gave birth to what developed into a new interdisciplinary research area. It provided an intellectual framework for all subsequent computer research and development efforts [15].

Given the general historical background, Artificial Intelligence (AI) focuses on the replication of cognitive abilities to address real-world challenges and the development of systems capable of learning and reasoning in a manner akin to human cognition [16]. More and

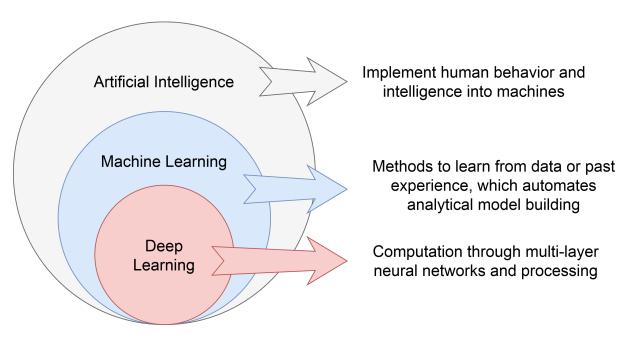


Figure 2.1: Artificial intelligence Diagram [1]

more studies show how important it is to give artificial intelligence systems the ability to build causal models of the world, giving explanation and understanding more weight than just solving pattern recognition problems [17]. Finally, the branches of artificial intelligence can be directly and informally represented in Figure 2.1.

With the advancement of technical elements such as data, algorithms, and computing capabilities reaching a relatively mature stage, artificial intelligence (AI) has begun to efficiently address problems and generate real economic advantages [18]. Thus, it is not surprising that AI is currently being applied in [19]:

- Automotive Industry.
- Financial Markets.
- Health Industry.
- Retailing Industry.
- Media Industry.
- Smart Payment Systems.
- Smart Homes

2.2 Machine Learning

The formulation of a singular, comprehensive definition for machine learning (ML) is challenging due to its broad scope, which covers several methodologies derived from computer science and multivariate statistics [20]. However, machine learning is precisely defined as "a set of methods that can automatically detect patterns in data and then use the uncovered patterns to predict future data or to perform other kinds of decision-making under uncertainty (such as planning how to collect more data!)" [21]. Given this definition, machine learning is defined as a subfield of artificial intelligence and a group of methods and algorithms with the capacity to learn continuously from accumulated or past experience and hone their skills through knowledge extraction from data without being specifically programmed. On top of that, it is the most famous and modern technology in the fourth industrial revolution [22], [23]. For example, in a real-world scenario where ML is used, in the context of a sophisticated system like a self-driving vehicle, machine learning algorithms must process extensive sensor data and utilize their acquired knowledge to identify potential hazards and subsequently choose appropriate control actions [24].

At the moment, the discipline of machine learning is structured around three main areas of research [25]:

- Task-Oriented Studies. It employs the engineering approach, which refers to the process of designing and evaluating learning systems with the aim of enhancing performance in a certain set of tasks. This involves the development and analysis of these systems.
- Cognitive Simulation. The study and computational modeling of human learning processes.
- Theoretical Analysis. The present study focuses on the theoretical investigation of the expansive range of potential learning methods and algorithms, absent any specific application domain.

When considering applications, machine learning is utilized across a diverse range of domains, including robotics, virtual personal assistants such as Google, computer games, pattern recognition, natural language processing, data mining, traffic prediction, online

transportation networks, product recommendation systems, share market prediction, medical diagnosis, online fraud detection, agriculture advisory services, search engine result optimization like Google's search engine, chatbots for online customer support, e-mail spam filtering, crime prediction through video surveillance systems, and face recognition in social media [26].

In order to properly summarize the previously mentioned field of inquiry, it is necessary to follow the appropriate structure suggested by [27], [28], [20] and [29]. Thus, in general, the following sections are going to consist of data, types of algorithms, and tasks related to machine learning.

2.2.1 Data

Because the machine learns by using specific data sets for training and then applying algorithms to those data sets, the availability of data is the main factor in the development of machine learning models in practical applications.

In the real world, data may appear in diverse formats, including structured, semi-structured, and unstructured forms [30]:

- Structured. The data exhibits a clearly defined structure and follows a data model that belongs to a standardized sequence, thereby enabling efficient organization and accessibility machine learning algorithms. Structured data is commonly stored in well-defined schemes, such as relational databases, where it is organized in a tabular fashion.
- Semi-Structured. Semi-structured data, in contrast to the structured data discussed earlier, is not often stored within a relational database. However, it possesses particular organizational characteristics that make its analysis feasible. Semi-structured data incorporates several forms, such as HTML, XML, JSON documents, and NoSQL databases.
- Unstructured. Unstructured data lacks a predetermined format or organizational structure, thereby making it considerably more challenging to gather, handle, and analyze. This type of data mostly includes textual and multimedia content. There

are various forms of unstructured data, such as sensor data, emails, blog entries, wikis, word processing documents, photographs, presentations, web pages, and other business documents, can be classified as examples of unstructured data.

Furthermore, since datasets are made up of data objects (e.g. rows of the dataset, observations) and attributes/features (e.g. columns of the dataset), it is important to be clear with which types of attributes we are dealing with. Overall, there are four types of attributes [31]:

- Binary Attributes. A binary attribute represents a nominal attribute that includes merely two categories or states, namely 0 or 1. In this context, 0 signifies the absence of the attribute, while 1 denotes its presence. Binary attributes are commonly known as booleans when the two states they represent correspond to true and false.
- Nominal Attributes. The values associated with a nominal attribute consist of symbolic representations or designations of entities. Values that represent nominal attributes, also referred to as categorical attributes, correspond to various categories, codes, or states. The values lack a recognizable or significant order. If *haircolor* is an attribute, its possible nominal values may be black, brown, yellow, or blonde. Besides, we can assign a code to such values, for instance, 0 for black, 1 for brown, and so on.
- Ordinal Attributes. An ordinal attribute refers to an attribute that contains potential values that indicate some sort of order or ranking, but the exact magnitude between consecutive values remains unknown. If drinksize is an attribute, its possible ordinal values may be small, medium, and big. These values follow a sequence by nature, but we cannot tell how much bigger "small" is than "medium."
- Numerical Attributes. It is possible to classify a numerical attribute as quantitative, indicating that it is a measurable quantity that has an integer or real value as its representation.

Machine learning classification tasks often categorize attributes as discrete or continuous. In this context, discrete attributes have a finite or countably infinite set of values; otherwise, continuous attributes can be represented as floating-point numbers [31]. In machine learning, it is essential to determine whether the attributes are discrete or continuous, as it affects the choice of appropriate algorithms and techniques.

2.2.2 Types

Algorithms for machine learning fall into the following categories [32]:

- Supervised learning. In a commonplace supervised situation, a teacher or supervisor gives the agent a precise error measure that is equivalent to output values. When referring to algorithms, actual methods use a training set of input-output pairs to provide this function. Based on this knowledge, the agent may adjust its settings to lower its minimize loss function. If the method is flexible and the data components are coherent, accuracy improves, and the difference between predicted and expected values approaches zero after each iteration. The applications of supervised learning can be in the fields of predictive analysis, spam detection, pattern detection, natural language processing, and sentiment analysis, among others.
- Unsupervised learning. This approach relies on the lack of a supervisor and, consequently, the absence of absolute error measures. Cluster analysis is a valuable tool for grouping elements based on their similarity or distance measure. In fact, unsupervised learning provides an implicit descriptive analysis of the data in order to obtain a complete understanding of the dataset. This approach can be applied to object segmentation, similarity detection, automatic labeling, and recommendation engines.

2.2.3 Tasks

Classification

Classification is a machine learning task that is employed to classify data into two or more distinct classes. The classification of potato diseases based on deep learning and computer vision methods has the potential to serve as an interesting instance of multi-class classification [33]. From the last example, it can be inferred that there are commonplace types of classification problems based on the label assigned to each instance.

- Binary Classification. The term "binary classification" refers to classification tasks in which there are two class labels that can be expressed by either numerical values (0 or 1) or textual descriptors (true or false).
- Multi-Class Classification. Multi-class classification involves the classification of more than two classes. In the context of a given set of classes, examples are categorized as members of a specific class.

For instance, some popular classification algorithms used are Naive Bayes, logistic regression, K-nearest neighbors (KNN), support vector machines, decision trees, random forests, adaptive boosting (AdaBoost), extreme gradient boosting (XGBoost), etc [27].

Dimensionality Reduction

The objective of dimensionality reduction is to address the issue of high dimensionality in data, also referred to as the curse of dimensionality. The primary purpose is to enhance the quality of data by lowering its complexity [34]. In the fields of machine learning and data science, processing high-dimensional data poses a significant challenge that both researchers and application developers must address. Dimensionality reduction plays a crucial role as it encourages improved human interpretations, reduces computational costs, and decreases problems such as overfitting and redundancy by simplifying models [27]. Dimensionality reduction can be categorized into feature selection and feature extraction. The main difference between feature selection and feature extraction lies in their respective approaches: feature selection involves keeping a subset of the original features [35], whereas feature extraction involves creating whole new features [36]. More precisely, in the former, discriminative features are chosen to create a new subset of them, which allows us to delete irrelevant or redundant features present in the data. The selection of an appropriate and effective subset of features inside a given problem domain has the potential to reduce the issue of overfitting by simplifying and generalizing the model, hence enhancing its accuracy [35]. In the latter, a better understanding of the data is provided, which results in enhanced predictive accuracy and a reduction in computing costs or training duration.

Deep Learning

The Fourth Industrial Revolution, also known as Industry 4.0, primarily centers around technology-driven automation and intelligent systems, which are applied in various areas, such as smart healthcare, business intelligence, smart cities, and cybersecurity intelligence [37]. Due to its excellent learning capabilities from historical data, DL techniques can play a key role in building intelligent data-driven systems that meet the needs of today and potentially alter the world and the way people live in it through their automation potential and their ability to learn from experience [1].

By definition, deep learning is a branch of machine learning that uses hierarchical structures to try to find complex abstractions or architectures in high-dimensional data [1], [38]. This branch of machine learning has become more popular and has been used in many areas, such as semantic parsing, transfer learning, natural language processing, computer vision, and many more. Furthermore, the current growth of DL can be attributed to three key factors: the notable advancements in chip processing capabilities, including GPU processing and Moore's Law; the substantial reduction in hardware construction costs; and the enormous progress made in machine learning algorithms [29]. Surprisingly, there are already many areas where deep learning has exceeded human-level capability and performance, e.g., predicting movie ratings, decisions to approve loan applications, and estimating car delivery times [39].

In order to employ deep learning techniques, it is crucial to consider the following properties and potential requirements [1]:

- Amount of Data. Deep learning relies on a substantial volume of data to construct a data-centric model for a specific problem domain. Deep learning algorithms tend to exhibit suboptimal performance when the dataset size is small [40].
- Hardware Dependency. Deep learning algorithms require significant computational resources when training models with extensive datasets. Due to the increased computational demands, GPUs are commonly employed to boost running time efficiency, as they offer significant advantages over CPUs. GPU hardware is essential for effective deep learning training. Accordingly, deep learning algorithms require

high-performance computing machines, such as the HPC CEDIA Cluster, to be run optimally [41].

- Feature Engineering Efficiency. Feature engineering involves extracting features from raw data by applying domain knowledge. One key difference between DL and other machine learning methods lies in the focus on extracting high-level features directly from data [27], [42]. Fortunately, deep learning reduces the time and effort needed to create a feature extractor for each problem.
- Training and Execution Time. Deep learning algorithms typically require a substantial amount of time for training due to the high number of parameters involved. Deep learning models can require several weeks to complete a training session, while training with machine learning algorithms typically takes significantly less time, ranging from seconds to hours [43], [41].
- Blackbox Model. Interpretability is a crucial consideration when comparing DL with ML. The process of obtaining a DL result can be challenging to explain due to its "black-box" nature.

We can generally specify several technologies used in deep learning and intended to be employed in this work [44], [1]:

• Multi-layer Perceptron (MLP). Multi-layer Perceptron (MLP) is a supervised learning method [45] and the foundational structure of deep neural networks (DNN). This type of architecture falls under the category of feedforward artificial neural networks (ANNs). A standard multi-layer perceptron (MLP) is a fully connected neural network consisting of an input layer for getting input data, an output layer for making decisions or predictions, and one or more hidden layers that serve as the network's computational engine (See Figure 2.2) [31], [46]. The MLP network's output is determined by employing various activation functions, also referred to as transfer functions, including ReLU (Rectified Linear Unit), Tanh, Sigmoid, and Softmax [45], [47]. The most commonly used algorithm for training MLP is "backpropagation", which is a supervised learning technique and considered the fundamental component of a neural network [31]. Different optimization approaches, including Stochastic

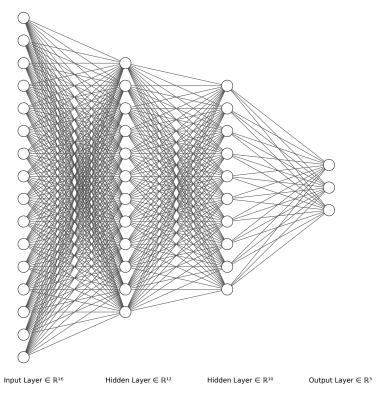


Figure 2.2: Multi-layer Perceptron Diagram with 16 inputs, 2 hidden layers and 3 outputs

Gradient Descent (SGD), Limited Memory BFGS (L-BFGS), and Adaptive Moment Estimation (Adam), are utilized in the training process. The MLP model needs hyperparameter tuning or adjusting the parameters, including the number of hidden layers, neurons, and iterations, which can result in a computationally demanding solution for complex models. MLP has the advantage of learning non-linear models in real-time or online through partial fit [45].

• Convolutional Neural Networks (CNN). CNN are widely used deep learning architectures that learn directly from input data without requiring human feature extraction. Each layer in a CNN considers optimal parameters to produce a meaningful output and also reduces the complexity of the model, thus improving the architecture of conventional ANNs, such as regularized MLP networks. Moreover, on top of that improvement, CNNs employ a 'dropout' technique [48], which involves randomly dropping out a certain percentage of nodes during training, which helps prevent the model from memorizing the training data and generalizing better, to address the issue of overfitting. CNNs are commonly used in various fields such as

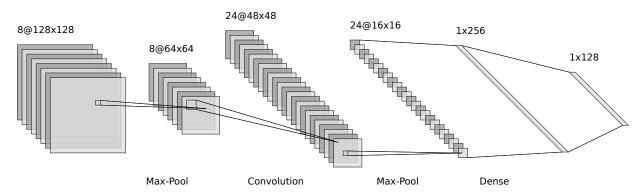


Figure 2.3: CNN Architecture Diagram

visual recognition, medical image analysis, image segmentation, and natural language processing since they are intended to deal with 2D shapes such as images (See Figure 2.3) [49], [47]. The autonomous identification of crucial input features without human intervention enhances the potency of this network in comparison to traditional networks. Various CNN variants have been developed in the field, such as VGG [50], AlexNet [51] and ResNet [52].

• Deep Transfer Learning (DTL). Transfer learning is a method that leverages preexisting model knowledge to efficiently address a new task, requiring minimal training
or fine-tuning. Training DL models requires a large amount of data. However, this
can present a challenge for specific applications, such as medical image analysis. The
collection and annotation of large datasets of patient data are not only challenging
and expensive but also require significant computational resources for model training.
Hopefully, transfer learning could help address this issue by transferring knowledge
from a pretrained DL model to a specific or desired model to be trained. Because
of its ability to successfully train deep neural networks with minimal input data,
transfer learning has become a hot topic in the field of DL [53]. The process can be
visualized in Figure 2.4.

There are many other deep learning methods that deserve mention, such as Recurrent Neural Networks (RNNs), Generative Adversarial Networks (GANs), Auto-Encoders (AEs), Self-Organizing Maps (SOMs), Deep Belief Networks (DBNs), Hybrid Deep Neural Networks, and Deep Reinforcement Learning, all of which are reviewed in detail in [1], but not in this work since those architectures aren't used at all.

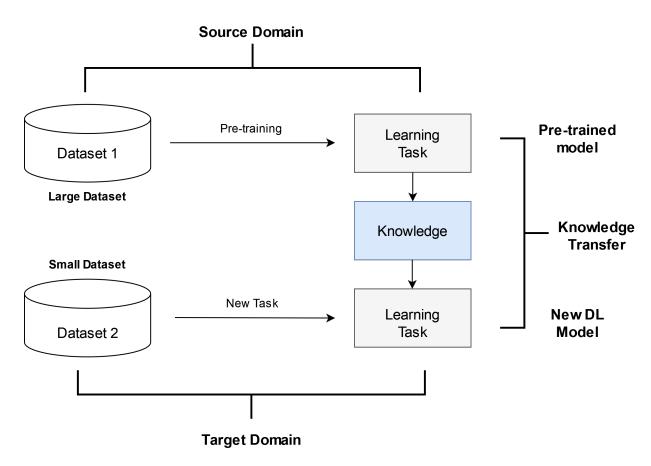


Figure 2.4: The transfer learning process involves transferring knowledge from a pre-trained model to a new deep learning model [1]

Chapter 3

State of the Art

There is no official state-of-art document for mispronunciation detection system. However, in [54], a literature review detailed three main techniques: posterior probability-based methods, classifier-based methods, and deep belief network-based methods. Probability-based techniques have the capability to identify the quality of pronunciation; yet, these scoring computations alone are insufficient to differentiate the type of error and accurately pinpoint its position. To solve this, classifier-based methods are used [54]. These methods can be summarized as follows:

• Posterior-Probability-based Methods

- The Goodness of Pronunciation (GOP) score was introduced by Witt and Young [55], who based it on a standardization of log-probability values by the duration of individual phone sections. A criterion was established for evaluating the accuracy of pronunciation for each phone.
- In their study, Franco et al. [56] developed three recognition models that used data from different levels of nativeness. The researchers also took into account the log-likelihood-based scores in order to determine the proportionality of these scores.
- The application of acoustic-phonetic parameters, such as log root mean square, energy, and zero-crossing rate, for the recognition of velar fricative and velar plosive was examined by Strik et al. [57].

- Minematsu et al. [58] introduced a widely recognized acoustic framework in speech that effectively eliminates non-semantic information.
- Wang and Lee [59] additionally incorporated GOP scores with error pattern identifiers to enhance the performance of identifying errors within the gathering of students from 36 nations learning Mandarin Chinese.
- Zhang et al. [60] presented scaled log-posterior probability (SLPP) and weighted phone SLPP to enhance measures of pronunciation quality.

• Classifer-based Methods

- The clusters of pronunciation rules were produced by Ito et al. [61], who also established a threshold for each cluster. The researchers proposed a clustering technique that is based on decision-making in order to improve the accuracy of error detection.
- The features obtained by Georgoulas et al. [62] by the utilization of a discrete wavelet transform. Wavelet analysis is a distinctive signal processing technique that uses a support vector machine (SVM) classification algorithm to identify instances of mispronunciation.
- In their work, Strik et al. [57] conducted a comparative analysis of four different methods, specifically GOP, decision tree, LDA_APF, and LDA_MFCC. The results of this study show that the LDA-APF and LDA-MFCC techniques did a better job of classifying the velar fricative /x/ and the velar plosive /k/ than the GOP scores and decision tree.
- Amdal et al. [63] conducted a study in which they made distinctions between short and long vowels in speech. The researchers employed acoustic-phonetic characteristics and conducted training on a linear discriminant analysis (LDA) classifier.
- Li et al. [64] introduced a detection strategy that is based on GLDS-SVM (Generalized Linear Discriminant Sequence-Support Vector Machine) and has been proven to be effective. The researchers integrated the GLDS-SVM method with

the UBM-GMM (universal background model) framework in order to improve performance.

- In their study, Wei et al. [65] employed log-likelihood ratios obtained from acoustic models as features for a support vector machine (SVM) classifier. Multiple acoustic models were used for each phone in order to determine the variety of pronunciation variations of that phone across various degrees of capability. This method facilitated the framework's achievement of superior performance compared to conventional likelihood-based strategies.
- In their study, Maqsood et al. [66] employed the sequential floating-forward selection method to identify the discriminative features for the purpose of detecting Arabic mispronunciations. Additionally, a technique based on grouping was presented by the researchers to identify distinctive properties of Arabic phonemes.

• Deep Learning

Nazir et al. and Akhtar et al. in [54] and [67] propose similar mispronunciation detection systems based on deep learning and machine learning. Most of them work by extracting features from spectrograms. In the former, it proposes hand-crafted features-based, CNN features-based, and transfer learning-based models subjected to feature selection and feature extraction (depending on the case). In the latter, it employs the same structure. The main differences between these articles are the different locations from which features are extracted.

Chapter 4

Methodology

4.1 Available Datasets

There are several classical datasets used for state-of-the-art mispronunciation system detection in Chapter 3. An example of big speech corpora designed for research purposes are L2-ARCTIC and TIMIT for English. Every speech corpora implements Textgrid as a tool for segmentation and labeling of the phonemes, words, and errors in an audio file as in Figure 4.1 and 4.2.

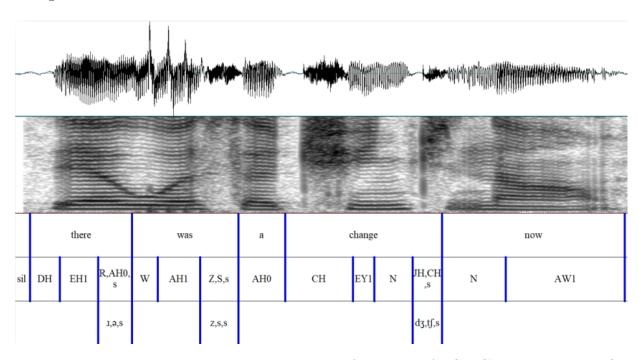


Figure 4.1: Textgrid with manual annotations from a L2-ARCTIC's sample audio file [2]. Top to bottom: speech waveform, spectrogram, words, phonemes, error tags, and comments from the annotator.

L2-ARCTIC [2] targets non-native English speakers. It consists of recordings from a total of 24 individuals whose first languages (L1s) are Hindi, Korean, Mandarin, Spanish, Arabic, and Vietnamese. Each L1 group includes two male and two female speakers, and each speaker contributed approximately one hour of recorded speech, specifically reading from CMU's ARCTIC prompts. From these recordings, orthographic and forced-aligned phonetic transcriptions were made, and each speaker's 150 utterances were manually annotated to look for substitutions, deletions, and additions, which are three different types of mispronunciation errors.

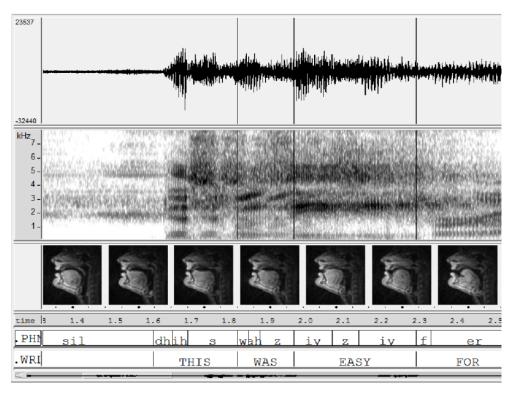


Figure 4.2: Textgrid with manual annotations from a TIMIT's sample audio file [3]. Top to bottom: speech waveform, spectrogram, phonemes, words

TIMIT [3] is a speech dataset corpus that supplies speech data that can be used for acoustic-phonetic research as well as for the advancement and assessment of automatic speech recognition systems. It is also a collection of broadband recordings of 630 people who speak eight of the most common types of American English. Each person was asked to read ten phonetically dense sentences. The TIMIT corpus consists of orthographic, phonetic, and word transcriptions that are aligned with time, along with a speech waveform file for each utterance, which is encoded in 16-bit format and has a sampling rate of 16 kHz.

The development of the corpus was a collaborative endeavor between the Massachusetts Institute of Technology (MIT), SRI International (SRI), and Texas Instruments, Inc. (TI). The lecture was recorded at TI, transcribed at the MIT, and then checked and prepared for CD-ROM manufacturing by the National Institute of Standards and Technology (NIST).

There are other datasets like TIMIT and L2-ARCTIC that target other languages, such as CU-CHLOE [68] and iCALL [69] for Chinese.

4.2 Dataset Construction

It was developed a simple and effective way to encourage participation and collect audio files for the dataset. The desired user (native or non-native) is required to record short audio clips using their own devices and send them to our number as WhatsApp audio files. In this way, by implementing state-of-the-art deep learning methods, the audio file is only required to generate its corresponding spectrogram to create a dataset, thus dealing with data scarcity and a lack of resources. To expand the dataset, the following text was sent to several test subjects in order to obtain different audios:

Hello, I need you to do me a favor, help me recording some audios of some words in Kichwa, it is for a thesis. Send me as Whatsapp audios right here.

Here are the requirements for the recordings:

- Record at a safe distance from the microphone.
- Speak at normal speed and timbre.
- Avoid at all costs blowing into your phone's microphone to avoid generating noise, being in a drafty environment, or being in an environment with background noise.
- One audio per word. If there are 5 words, 5 audios.
- Record the word completely.
- Record the word as you think it is pronounced (depending on whether or not you speak Quechua).

These are the words: yana, yanami, yurak, yurakmi, killu, killumi, puka, pukami, misi, misika, allku, allkuka, wakra, wakraka, atallpa, atallpaka, kuchi, kuchika, challwa, challwaka, amaru, amaruka, wiwa, Maykan, Kan.

Thank you for your cooperation and please take the requirements into account.

After that, they immediately send the audio files, and then the dataset collector can easily download the audio files in a single batch by using WhatsApp Web.

4.3 Dataset Labelling

For the labeling process, a procedure is followed in order to save the tags corresponding to the audio file. In this project, there are two labels: either well-pronounced or mispronounced, which can also be represented with the integers 1 or 0, respectively. Therefore, it is employed a folder structure in order to extract the labels from each audio file. This method consists of creating a root directory for the Kichwa dataset, creating subdirectories with the names of the labels, and saving the corresponding audio files to each label according to some criteria (see Figure 4.3). This type of labeling enables us to extract the labels and preprocess them later.

The criteria employed for labeling each audio file were partly based on the fluency in Kichwa of the participant in the non-synthetic case. This means that if the participant was a native Kichwa speaker, the label of its audio files is 1. Otherwise, if the participant is a non-native Kichwa speaker, the data collector, in this case the author himself, takes a native Kichwa speaker as a model and compare the spoken word. If the word pronounced by the non-native speaker matches up with the word pronounced by the other counterpart, it was categorized as 1; otherwise, it was categorized as 0. Finally, in the case of synthetic data (see Section 5.2), since all the speakers are pronouncing Kichwa words according their languages, their respective audio files will be labelled as 0.

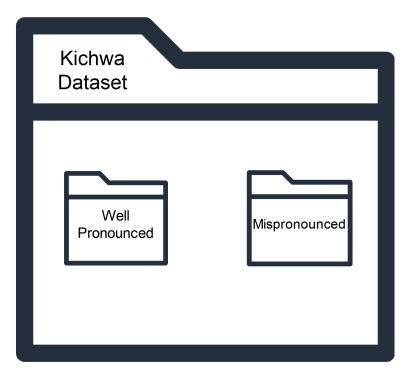


Figure 4.3: Folder directory labelling

4.4 Dataset Splitting and Audio Distribution

Given the non-synthetic and synthetic audio files of various speakers, it raises the question of how the data is distributed over the dataset. For this work, it was decided to split and distribute the data in two ways, therefore creating two datasets.

- Original Dataset. The synthetic data is considered only for the training set. Since the dataset is imbalanced at the beginning, stratified dataset splitting is used on the original dataset to obtain metrics such as PPV and NPV. In other words, the training, validation, and testing sets have the same proportion of well-pronounced and mispronounced words. After the splitting, all the synthetic audio samples were added to the mispronounced training set. Only non-synthetic voices were added to the validation and testing datasets.
- Synthetic Dataset. The synthetic data is considered for training, testing, and validation sets. In this way, a balanced dataset is obtained and, in consequence, split the data into 70%, 20%, and 10% for training, validation, and testing sets, respectively.

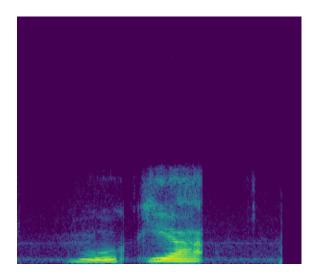


Figure 4.4: Spectrogram representation of the word "allku" pronounced by a non-native female speaker

4.5 Preprocessing and Spectrogram Conversion

Since a CNN-based deep learning model is used, it is necessary to decide how the audio dataset is represented in a 2D input way. The traditional spectrogram (see Figure 4.4), like other time-frequency representations, offers a visual representation of the temporal and frequency distribution of energy in a signal, specifically in relation to Fourier's frequency [70].

After deciding the representation of the data, each audio file is resampled before the spectrogram conversion. In the case of non-synthetic data, audio files are sampled at 48000 Hz, and in the case of non-synthetic data, at 24000. Then both audios are resampled to 44100 Hz. Finally, the duration of all audio files is adjusted to 3 seconds.

4.6 Selected CNN Architectures

4.6.1 AlexNet

Alexnet [71], [72] is a convolutional neural network (CNN) that achieved success in the 2012 Imagenet Large Scale Visual Recognition Challenge, specifically in the tasks of object recognition and image categorization. The model performed training using ImageNet dataset, which includes over 1.2 million photos belonging to 1000 distinct categories. It includes a total of eight trainable layers. Those layers have been divided into a total of five

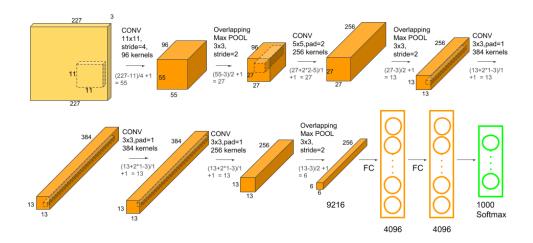


Figure 4.5: AlexNet Architecture Diagram [4]

convolutional layers and three fully connected layers. The Rectified Linear Unit (ReLu) activation function is employed in all trainable layers except for the last fully connected layer, which adopts a softmax function. The architectural design also has non-trainable layers, including three pooling layers, two normalization layers, and one dropout layer, which fulfill the purpose of preventing overfitting (See Figure 4.5). Researchers choose to employ the Rectified Linear Unit (ReLU) function. It was shown that deep convolutional neural networks implementing rectified linear units (ReLUs) exhibited much faster training speeds compared to their counterparts employing hyperbolic tangent (tanh) units. When compared to the tanh activation function, the use of the ReLU activation function solely generated a substantially faster training process, achieving a 25% error rate on the training set six times more quickly. Stochastic Gradient Descent has been used with a learning rate of 0.01, a momentum of 0.9, and a weight decay of 0.0005. The training process takes place using two GPUs, specifically GTX 580, in order to achieve parallelism. The graphics processing units (GPUs) employed in the system have a memory capacity of 3 gigabytes (GB) each. The network is divided into two equal halves distributed across the two graphical processing units (GPUs).

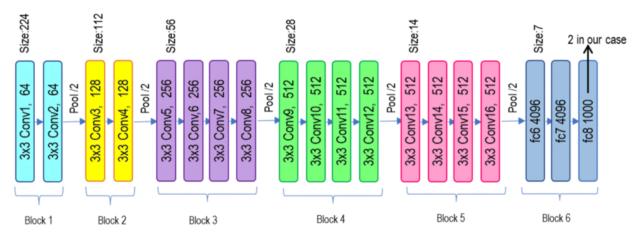


Figure 4.6: VGG19 Architecture Diagram [5]

4.6.2 VGG19

At the 2014 ILSVRC, A. Zisserman and K. Simonyan from the University of Oxford's convolutional neural network model, known as the VGG model or VGGNet, took first and second place in the categories of object detection and image classification, respectively [73]. The proposed approach substitutes the use of big kernel-sized filters with a sequence of several 3×3 kernel-sized filters, resulting in a notable improvement compared to the AlexNet architecture [73]. The VGG network is designed using convolutional filters of reduced dimensions. The VGG-16 architecture consists of a series of 13 convolutional layers, three fully connected layers, and a total of 16 layers in all. In contrast, VGG-19 shows an identical architecture to VGG-16, although with the inclusion of three more convolutional layers (See Figure 4.6) [73].

4.6.3 ResNet50

ResNets make it possible to train up to hundreds or even thousands of layers and still achieve outstanding performance, which is the case with ResNet-152, while still having lower complexity than VGGNet [74]. In general, training a deep neural network with lots of layers involves the vanishing gradient problem. The backpropagation algorithm is employed to iteratively adjust the weights of a neural network by using the chain rule of derivatives. As the process progresses, the repeated multiplication of gradients can result in the weights becoming exceedingly small as they propagate towards earlier layers [75]. To avoid this problem, ResNets use the concept of skip connections. In a conventional network

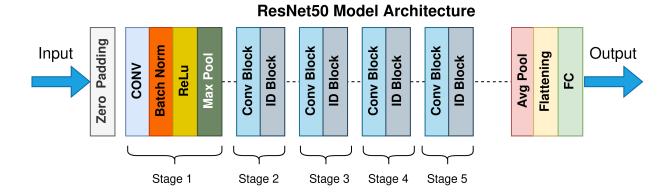


Figure 4.7: ResNet50 Architecture Diagram

architecture, the convolutional layers are sequentially arranged, but in skip connections, the conventional layers are sequentially arranged, with the original input being added into the output of the convolutional block [75].

Finally, ResNet50 can be described as a CNN architecture that consists of 50 layers that are divided into five residual blocks. Each residual block contains two convolutional layers, a batch normalization unit, and a ReLu unit after it. The output of the second convolutional layer is added to the input of the residual block and then passed to the next residual block [76]. The final layer is a fully connected layer that retrieves a flattened output from the last residual block and applies a softmax activation function to produce the final output probabilities.

4.7 Data Augmentation

First of all, in order to standardize all the audios in the dataset, the duration of all the audios was adjusted to 3 seconds. Then shifting time, shifting pitch, stretching time, adding Gaussian noise, and masking time were considered as audio data augmentation techniques. The following is a brief explanation of each audio augmentation technique [77]:

- Noise Injection. It inserts a random number into the data.
- **Time Shift.** It moves the audio to the left or right at an arbitrary second interval. When The audio is moved to the left with x seconds, the first x seconds will be

marked as 0 (which indicates quiet) or rollover. When the audio is moved to the right after x seconds, the final x seconds will be marked as 0 or rollover.

- Pitch Shift. It changes pitch randomly or within a range of semitones.
- **Time Stretch.** It stretches an audio file by a fixed rate, or in other words, changes the speed of the audio.
- Time Masking. It deletes a portion of audio.

For this work, each audio is augmented 100 times, and the probability that each type of audio augmentation is applied to such audio is 0.5.

4.8 CNN Features-Based Model

4.8.1 Feature Extraction

Abstract characteristics and concepts can be taught to CNN by feeding them with raw image pixels. Features like edges and basic textures are taught to the first convolutional layer(s); complex textures and patterns are examples of the types of characteristics that later convolutional layers can learn; and the final convolutional layers are responsible for learning features such as objects or parts of things [78]. Finally, the fully linked layers are trained to establish associations between feature activations and target classes.

Thus, for the current work, knowledge of low-level features and high-level features acquired from pretrained CNN architectures are used to extract the deep learning features from the fully connected layers inspired by S. Akhtar et al. [67].

After obtaining the audio spectrograms and resampling the images to pass them through the CNNs, pre-trained models from AlexNet, VGG19, and ResNet50 are imported, and the point of extraction for the features are defined. For AlexNet, the features are extracted from the output of the first and second fully connected layers; for VGG19, similarly, from the output of the first and second fully connected layers; and for ResNet50, from the input of the first and only fully connected layer. The dimensions of the array of features extracted (number of features) are 4096 in both layers for AlexNet and VGG19 and 2048 in ResNet50.

On top of that, in order to optimize the process of feature extraction, extraction was performed in batches. Particularly, the batch size used was 1024 for every model specified.

4.8.2 Dataset Discretization

The majority of classification tasks in the field of machine learning often involve the process of acquiring the ability to differentiate between different nominal class values. However, it is important to note that these tasks can include features that possess ordinal or continuous attributes in addition to their nominal characteristics. Discretization is a widely employed method for achieving this objective. Discretization refers to the procedure of converting attributes with continuous values into nominal values.

For the purpose of discretizing numerical features, Fayyad and Irani [79] proposed a discretization technique that uses the minimum entropy heuristic. The approach employs the concept of class entropy to determine the optimal cut point for discretization among the candidate divisions. The aforementioned approach can be iteratively implemented on the two intervals resulting from the previous split until certain halting criteria are met. This process generates several intervals for the feature. Since a number of studies have shown that this discretization technique is superior to any other technique overall [80], this technique is used in this work.

4.8.3 Statistical-based Feature Selection Methods

Statistical-based feature selection methods originate from different statistical measures. The majority of these methods are filter-based, utilizing statistical measures rather than learning algorithms to evaluate feature relevance. Furthermore, the majority of statistical algorithms tend to analyze features in isolation, and therefore, feature redundancy is neglected during the selection phase [81].

For our purpose, we are using the Chi-square score, proposed in [82] as a feature selection method, that utilizes the test of independence to assess whether the feature is independent of the class label. Since this feature selection method can only be used in discrete or ordinal values [81], dataset discretization must be used before as a preprocessing step.

4.8.4 Classification and Hyperparameter Tuning

For simplicity, KNN is used. In particular, KNN is a non-generalizing learning algorithm, also referred to as an instance-based or lazy learning algorithm. The approach does not prioritize the construction of a comprehensive internal model. Instead, it retains all instances associated with the training data in an n-dimensional space. KNN uses data and classifies new data points based on similarity measures (e.g., the Euclidean distance function) [45]. The classification process involves determining the class label of a point by taking into account the majority vote of its k nearest neighbors. The model exhibits resilience to noisy training data, and its accuracy is dependent upon the quality of the data [27]. A major challenge in K-nearest neighbors (KNN) is determining the optimal number of neighbors to consider. KNN is applicable for both classification and regression tasks [27].

In order to evaluate some possible effects between selecting features and not selecting features, the best parameter for the number of neighbors (k) is searched in the dataset without any preprocessing or feature selection methods. Then, using the same obtained hyperparameter, preprocessing, feature selection and hyperparameter tuning in the best parameter for the number of selected features is performed using validation datasets. Finally, KNN is tested on feature-selected data and non-feature-selected data using the testing dataset and searched hyperparameters. For hyperparameter tuning the HPC CEDIA Cluster was used. More precisely, a "gpu" partition with 64 CPU cores, 128000 MB of memory, and 1 Nvidia A100 SXM4 with 40 GB of memory was reserved for 48 hours.

4.9 CNN Transfer learning-based Model

4.9.1 Importing Pre-trained CNN Network

The transfer learning model incorporates pre-trained versions of the AlexNet, VGG19, and ResNet networks. After preprocessing the audio datasets, the models trained on the ImageNet database were loaded, which comprises a large number of images. PyTorch provides pre-trained models within its API.

4.9.2 Modifying Final Layers

The final three fully connected layers of both the pre-trained AlexNet and VGG models are designed to classify 1000 different classes. ResNet employs a single, fully connected layer consisting of 1000 classes. The layers in question acquire high-level features for data classification, while others acquire low-level features. Consequently, these layers were optimized to conduct classification on our dataset. In this scenario, it was opted to maintain consistency in the number of fully connected layers across different architectures. However, the output of the last layer were modified to two outputs, enabling binary classification. Following the reset and modification of the final layers in each model, the parameters for a fully connected layer were defined using new data. Cross-entropy was employed as the loss function.

4.9.3 Hyperparameter Tuning

In order to perform hyperparameter tuning and training, it is need to define hardware specifications and parameters to be tuned.

The HPC CEDIA Cluster for hyperparameter tuning was used. More precisely, it was reserved a "gpu" partition for 48 hours, 64 CPU cores, 128000 MB of memory, and 1 Nvidia A100 SXM4 with 40 GB of memory. Additionally, the learning rate and training batch size were first set as hyperparameters to be tuned during the hyperparameter process in all models. The learning rate determines the step size at each iteration, while the batch size defines the number of samples processed in each training iteration. The optimizers employed were Stochastic Gradient Descent (SGD) with 0.09 momentum for Alexnet, Adam for VGG19, and ResNet. In the case of Adam, we also set the weight decay parameter to be tuned during the hyperparameter process. In the case of Alexnet and VGG19, the number of output features of the first two fully connected layers were set to be tuned. Table 4.1 summarizes the range of values to be chosen for hyperparameter tuning.

4.10 Model Performance Metrics

In order to access the mispronunciation system, distinctive metrics are introduced that evaluate performance based on a particular class or label, such as PPV, NPV, and BPV.

Parameter/Model	AlexNet	VGG19	${ m ResNet50}$	
	Logarithmic	Logarithmic	Logarithmic	
Learning Rate	Random Value	Random Value	Random Value	
	from 1e-4 to 1e-1	from 1e-4 to 1e-1	from 1e-4 to 1e-1	
Training Batch	64 100 056 510	64 100 056 510	64 100 056 510	
Size	64, 128, 256, 512	64, 128, 256, 512	64, 128, 256, 512	
First Fully	32, 64, 128, 256,	32, 64, 128, 256,		
Connected Output	512, 1024, 2048,	512, 1024, 2048,		
Features	4096	4096, 8192, 16384		
Second Fully	32, 64, 128, 256, 512,	32, 64, 128, 256,		
Connected Output	1024, 2048, 4096	512, 1024, 2048,		
Features	1024, 2040, 4090	4096, 8192, 16384		
Weight Decay		Uniform Float Value	Uniform Float Value	
		from 1e4 to 1e-2	from 1e4 to 1e-2	

Table 4.1: Range of Parameters to be Chosen for Hyperparameter Tuning

4.10.1 PPV (Positive Predictive Value)

It estimates the ratio of accurately predicted positive occurrences (true positives) to the total number of cases that the model identified as positive (See Formula 4.1). Intuitively, PPV is the ability of the classifier or model not to label as well-pronounced an audio that is mispronounced [83].

$$PPV = \frac{TP}{TP + FP} \tag{4.1}$$

where TP and FP stands for true positives and false positives.

4.10.2 NPV (Negative Predictive Value)

It estimates the ratio of accurately predicted negative occurrences (true negatives) to the total number of cases that the model identified as negative (See Formula 4.2). Intuitively, NPV is the ability of the classifier or model not to label as mispronounced an audio that is well-pronounced [83].

$$NPV = \frac{TN}{TN + FN} \tag{4.2}$$

where TN and FN stands for true negatives and false negatives.

4.10.3 Custom Metric: BPV (Balanced Predictive Value)

It is the arithmetic mean of the positive and negative predictive values (See Formula 4.3). This metric aims to assess how well-balanced the model's predictive power is. A high value for BPV indicates that the model does a good job of not confusing well-pronounced words with mispronounced words and vice versa. If not, the model does confuse a lot when trying to distinguish well-pronounced words from mispronounced ones, and vice versa.

$$BPV = \frac{PPV + NPV}{2} \tag{4.3}$$

where PPV and NPV stands for positive and negative predictive values, respectively.

Chapter 5

Results and Discussion

5.1 Dataset Construction

When considering building a mispronunciation detection system taking into account the state-of-the-art used datasets (See Section 4.1) for Kichwa, there were several problems.

- There are no Kichwa language speech corpora for mispronunciation system detection.
- In order to test and evaluate a mispronunciation system's detection, it is necessary to have a dataset like L2-ARCTIC and TIMIT, but for the Kichwa language.
- Since L2-ARCTIC, TIMIT, and other similar datasets use Textgrid and audio files to align the transcriptions and label the phonemes and mispronunciation errors, it would be ideal to have a similar format for the Kichwa language dataset, but this is clearly outrageously difficult due to the lack of personnel.
- The process of text-gridding audio files in the Kichwa language requires extensive resources, expertise, and lots of time.
- It is extremely hard to find participants who are fluent in Kichwa and willing to contribute to the dataset. This case also applies even when finding participants who are not fluent in Kichwa, which is completely disheartening.
- The process of collecting data can discourage Kichwa native speakers and non-native speakers from participating in the process due to the duration of the process, lack of incentives, potential privacy concerns, transportation issues, lack of time, etc.

For this work, to overcome these problems, the previously described method in Section was employed. Then a Kichwa dataset that consists of a total of 500 recorded words from various Kichwa speakers and non-speakers was constructed. Each participant in the collection recorded and sent 25 audio files corresponding to the words in Kichwa: yana, yanami, yurak, yurakmi, killu, killumi, puka, pukami, misi, misika, allku, allkuka, wakra, wakraka, atallpa, atallpaka, kuchi, kuchika, challwa, challwaka, amaru, amaruka, wiwa, Maykan, Kan. We summarize the dataset in Tables 5.1 and 5.2.

Number of Participants							
Fluency	Adult Male	Adult Female	Total				
Native	3	6	9				
Non-Native	6	5	11				
Total	9	11	20				

Table 5.1: Number of participants in the dataset

Words	
Label	Total
Well-pronounced	100
Mispronounced	400
Total	500

Table 5.2: Distribution of words in Kichwa dataset

5.2 Syntetic Data

As in Table 5.2, there is an imbalanced proportion of well-pronounced words. and mispronounced words. When developing transfer learning models, or, to be more precise, hyperparameter tuning, this imbalance can pose a challenge to accurately predict the mispronounced words. In fact, the value of the negative predictive value was always 0, and the positive predictive value was greater than it. This means that the model does not have enough data to learn how to distinguish mispronounced words from pronounced words.

As a solution, it was decided to use synthetic data in the dataset. To generate such data, a website called "Coqui Studio" was used. Initially, the website allows users to input text and have it converted into audio, providing a large amount of data for the model to learn from. Thus, by inputting Kichwa words into the generator, a person's voice and

Name	Gender	Language
Aaron Dreschner	Male	English
Szofi Granger	Female	English
Ige Behringer	Male	English
Claribel Dervla	Female	German
Ilkin Urbano	Male	French
Camilla Holmstrom	Female	Portuguese
Baldur Sanjin	Male	Portuguese
Tanja Adelina	Female	French
Wulf Carlevaro	Male	German
Annmarie Nele	Female	German
Marcos Rudaski	Male	Portuguese
Vjollca Johnnie	Female	English

Table 5.3: Syntetic Participants Details

language can be selected to generate mispronounced Kichwa audio outputs. The synthetic participants' names and their respective languages used in this work can be detailed in Table 5.3.

In total, 300 Kichwa audios were collected from each artificial participant. It is important to clarify that the website's model used in this work was XTTS, which is a newer and more expensive model. Therefore, it was expected that the voices generated by XTTS would be of higher quality and more natural-sounding compared to other models, which, in fact, they were.

5.3 Dataset Labelling

In fact, it should be ideal to have a voting mechanism in which if two of some experts in Kichwa language agree on the same label, then that label is assigned to that data as in [67], but that is not clearly possible given the circumstances.

5.4 Model's Testing

The best models' testing metrics can be shown in Table 5.4. Therefore, it might be found appropriate to expose the results and discussion for this section by exposing how the aforementioned table was constructed in addition to relevant observations in their respective stages of construction.

5.4.1 CNN Features-based Model

- From Figure A.1 and A.2, it can be observed the hyperparameter best selection given the number of neighbours k. The highest peak in the figure indicates the best k selection. It is important to note that the original dataset was transformed in another dataset given the feature extraction location in AlexNet, ResNet50 and VGG19. Moreover, the hyperparameter tuning was done given the training set and validation set.
- From Tables A.1 and A.2, it can be detailed better the model, its feature extraction point, the best k hyperparameter selection and its corresponding validation balanced predictive value metric given the dataset. Thus, from A.1 and A.2, it is seen that "ResNet50: AvgPool" and "AlexNet: FC1" are the best models without feature selection taking into account the validation BPV for the original and synthetic dataset, respectively. Hereby, those models are also tested using the testing dataset and measured using PPV, NPV and BPV, and added to Table 5.4.
- After establishing the baseline models without feature selection, it is fixed k for each model and then perform dataset discretization and use Chi-square statistical-based feature selection method to obtain the best features. In Figures A.3 and A.4 we plot the possible improvement of performance based on BPV metric against the number of best important features taken into account. In Chi-square feature selection, the greater the Chi-square score is, the better the feature is [81]. The red line is the best BPV score without feature selection. Thus, the purpose of feature selection here, similarly, is to search for the optimal number of features to be taken into account. It can be observed that in some models, the feature selection method presuppose a greater performance in the validation dataset, while in others, the performance is not evident.
- In Tables A.3 and A.4, the best selection for the number of features according to the validation BPV is extracted for each model, and then obtain the PPV, NPV and BPV metrics using the testing dataset. Then, the best model given its validation BPV is added to Table 5.4.

• When comparing Tables A.1, A.2, A.3 and A.4, it is noted that: in the original dataset, feature selection improves the NPV value for the first three models, on the other hand, in the synthetic dataset, the NPV performance is worsened in some models;

5.4.2 CNN Transfer Learning-based Model

- For the hyperparameter tuning, it is selected a metric in order to optimize it, in this case, validation loss. However, Ray Tune library allows to eventually save the best models per trial given validation loss and validation BPV. This permits to compare the performance between different metric targets when searching a metric for optimization.
- In Tables B.1 to B.12, it is shown the best models given validation loss or balanced predictive value. Moreover, we record some other valuable metrics, for instance, PPV, NPV and BPV for training and validation sets. In the columns named "config", it can be seen the hyperparameters used in that trial.
- In Tables B.13 to B.18, it is extracted the TOP-5 best models by its corresponding best validation metric (loss or BPV), and tested them by using the testing set. In these tables, it is important to note that there are NPV perfect values, i.e. with a value of 1, which can be a symptom of an inappropriate size of the dataset. By the size of the dataset or the quality of the dataset, they may be deceivingly values. Nevertheless, finally, we extract the best models and add them to the general table.

5.4.3 Observations from Table 5.4

- In general, transfer learning-based models achieve better performance in BPV metric.
- Feature selection technique did not show any improvement performance against models without feature selection. On the contrary, the testing metric shows that the best models with feature selection are worse than the ones without it.
- It is not necessary to use deeper convolutional neural networks such as VGG19 or Resnet50 to achieve the best results. For instance, AlexNet achieves the best results

in CNN Tranfer Learning-based Model for both original and synthetic datasets.

• In terms of running time, in CNN Transfer Learning-based model, VGG19 took roughly 24 hours to complete hyperparameter search using original dataset and synthetic datasets. Meanwhile, the other two architectures took, in average, 4 hours to complete hyperparameter search for both datasets. On the other hand, in CNN Features-based model, running time was negligible since we performed feature extraction in batches, and the KNN Classifier used 64 cores.

Best Models' Performance on Testing Set							
Original Dataset							
CNN Features-Based Model							
Model	Test	Test	Test				
Wiodei	PPV	NPV	BPV				
ResNet50: AvgPool	0.84	0.50	0.67				
Without Feature Selection	0.04	0.90	0.01				
VGG19: FC2	0.85	0.37	0.61				
with Feature Selection	0.00	0.51	0.01				
CNN Transfer Learnin	g Base	d Mode	el				
Model	Test	Test	Test				
Wiodei	PPV	NPV	BPV				
AlexNet Loss	0.87	0.61	0.74				
AlexNet BPV	0.80	1.00	0.90				
VGG19 Loss	0.83	0.64	0.73				
VGG19 BPV	0.81	0.40	0.60				
ResNet50 Loss	0.80	0.00	0.40				
ResNet50 BPV	0.80	0.00	0.40				
Synthetic Dataset							
CNN Features-Based Model							
Model	Test	Test	Test				
Model	PPV	NPV	BPV				
AlexNet: FC1	0.88	0.90	0.89				
Without Feature Selection	0.00	0.90	0.69				
AlexNet: FC1	0.01	0.86	0.94				
with Feature Selection	0.81	0.80	0.84				
CNN Transfer Learnin	g-Base	d Mode	el				
Model	Test	Test	Test				
Wodel	PPV	NPV	BPV				
AlexNet Loss	0.83	1.00	0.92				
AlexNet BPV	0.83	1.00	0.92				
VGG19 Loss	0.83	0.94	0.88				
VGG19 BPV	0.79	0.91	0.85				
ResNet50 Loss	0.52	1.00	0.76				
ResNet50 BPV	0.65	1.00	0.82				

Table 5.4: Best Models' Performance on Testing Datasets. Each model has been selected according to validation metric.

Chapter 6

Conclusions

This work represents an initial attempt at developing a mispronunciation system for the Kichwa language. In this study, pretrained CNN architectures were employed to classify spectrograms of both well-pronounced and mispronounced words. The first model, the CNN features-based model, extracts the features from the fully connected layers, applies a feature selection method technique in order to extract the discriminative features from the non-discriminative ones, and classifies them with a KNN classifier. The second model, CNN transfer-learning-based, uses knowledge from convolutional layers and modifies the classifier layer for binary classification (well-pronounced or mispronounced). Overall, we can conclude that the CNN transfer learning-based method is better than the CNN feature-based method, but nevertheless, the CNN feature-based method achieves similar performance to the CNN transfer learning-based method in the synthetic dataset, With a larger dataset, the transfer learning-based method may outperform the CNN feature-based method in the original and synthetic datasets.

6.1 Observations and Limitations

Drawing significant conclusions from a small dataset is nearly impossible, especially
from a small and imbalanced dataset. However, a larger and more balanced dataset
would allow for more meaningful conclusions, and it would facilitate a comprehensive
study of feature selection techniques and deep learning architectures with increased
reliability.

- Since the majority of Kichwa words proposed were actually similar in pronunciation to Spanish words, most of the audios went to the majority class of well-pronounced words, leaving the minority class of poorly pronounced words underrepresented in the recordings. Thus, it is probable that the selection of words may be unfavorable for our work. In other words, it should have been chosen words with much more difficult pronunciation.
- The lack of participants and an official project for Kichwa preservation makes it
 difficult to gather a diverse range of audio samples. Additionally, the absence of
 standardized pronunciation guidelines further complicates the accuracy and consistency of the recordings.
- The use of synthetic data may ultimately distort the results and limit the reliability of any conclusions drawn from the study, particularly in the use of the synthetic dataset.
- The way words were collected may not be appropriate due to their individual recordings. In fact, it could be better if a person recorded different sentences from different sources. Hence, it should have been better to follow recording and organizational guidelines from official mispronunciation datasets such as TIMIT or L2-Arctic.
- It might be a risky decision to construct a mispronunciation model based on Kichwa first rather than a model based on official mispronunciation datasets such as TIMIT or L2-Arctic. In this way, it could have drawn better conclusions from the model, given the official datasets and learning how such datasets are constructed, in order to implement a similar one for Kichwa.
- CNN architectures may not be suitable for phrases or sentences. Instead, recurrent neural networks or transformer models are often employed due to their ability to capture sequential dependencies.
- The work only uses a KNN classifier and a feature selection method technique to evaluate performance, thus probably ignoring better alternatives.

It is necessary	o mibiemem	and Halll	the models	m a mgn-	репоппансе	companing
cluster.						

Bibliography

- [1] I. H. Sarker, "Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions," *SN Computer Science*, vol. 2, no. 6, p. 420, 2021.
- [2] G. Zhao, S. Sonsaat, A. Silpachai, I. Lucic, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna, "L2-arctic: A non-native english speech corpus," in *Proc. Interspeech*, 2018, p. 2783–2787. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1110
- [3] J. S. Garofolo, "Timit acoustic phonetic continuous speech corpus," *Linguistic Data Consortium*, 1993, 1993.
- [4] A. Pujara, "Concept of alexnet:- convolutional neural network," Jun 2021. [Online]. Available: https://medium.com/analytics-vidhya/concept-of-alexnet-convolutional-neural-network-6e73b4f9ee30
- [5] A. Khattar and S. Quadri, "Generalization of convolutional network to domain adaptation network for classification of disaster images on twitter," *Multimedia Tools and Applications*, vol. 81, no. 21, pp. 30437–30464, 2022.
- [6] UNESCO, "Indigenous peoples' right to education: overview of the measures supporting the right to education for indigenous peoples reported by member states in the context of the ninth consultation on the 1960 convention and recommendation against discrimination in education," 2019.
- [7] "Indigenous peoples UNESCO," https://www.unesco.org/en/indigenous-peoples, (accessed Aug 28, 2023).

- [8] "Indigenous peoples overview," https://www.worldbank.org/en/topic/indigenouspeoples#1, (accessed Aug 28, 2023).
- [9] UNESCO, "The UNESCO Courier," in 2019: the International Year of Indigenous Languages, 2019.
- [10] M. de Educación, "Informe en relación a la "convención relativa a la lucha contra las discriminaciones en la esfera de la enseñanza (la convención) y la recomendación relativa a la lucha contra las discriminaciones en la esfera de la enseñanza (la recomendación), aprobadas por la conferencia general de la unesco en 1960."
- [11] J. G. Armijos Monar, B. N. Fuertes Lopez, J. E. Delgado Altamirano, and V. M. V. Villa, "University indigenous students' perceptions towards kichwa, spanish and english." *English Language Teaching*, vol. 11, no. 2, pp. 131–148, 2018.
- [12] M. A. Peabody, "Methods for pronunciation assessment in computer aided language learning," Ph.D. dissertation, Massachusetts Institute of Technology, 2011.
- [13] T. R. Kurfess et al., Robotics and automation handbook. CRC press Boca Raton, FL, 2005, vol. 414.
- [14] M. Rosheim, Leonardo's Lost Robots. Springer Science & Business Media, 2006.
- [15] P. Hamet and J. Tremblay, "Artificial intelligence in medicine," Metabolism, vol. 69, pp. S36–S40, 2017.
- [16] A. Holzinger, G. Langs, H. Denk, K. Zatloukal, and H. Müller, "Causability and explainability of artificial intelligence in medicine," Wiley Interdisciplinary Reviews:

 Data Mining and Knowledge Discovery, vol. 9, no. 4, p. e1312, 2019.
- [17] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and brain sciences*, vol. 40, p. e253, 2017.
- [18] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 249–270, 2020.

- [19] C. Zhang and Y. Lu, "Study on artificial intelligence: The state of the art and future prospects," *Journal of Industrial Information Integration*, vol. 23, p. 100224, 2021.
- [20] M. Bertolini, D. Mezzogori, M. Neroni, and F. Zammori, "Machine learning for industrial applications: A comprehensive literature review," Expert Systems with Applications, vol. 175, p. 114820, 2021.
- [21] C. Robert, "Machine learning, a probabilistic perspective," 2014.
- [22] I. H. Sarker, A. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: an overview from machine learning perspective," *Journal of Big data*, vol. 7, pp. 1–29, 2020.
- [23] S. Das, A. Dey, A. Pal, and N. Roy, "Applications of artificial intelligence in machine learning: review and prospect," *International Journal of Computer Applications*, vol. 115, no. 9, 2015.
- [24] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, "Machine learning and the physical sciences," *Reviews of Modern Physics*, vol. 91, no. 4, p. 045002, 2019.
- [25] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Machine learning: An artificial intelligence approach. Springer Science & Business Media, 2013.
- [26] S. Ray, "A quick review of machine learning algorithms," in 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon). IEEE, 2019, pp. 35–39.
- [27] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN computer science*, vol. 2, no. 3, p. 160, 2021.
- [28] D. Dhall, R. Kaur, and M. Juneja, "Machine learning: a review of the algorithms and its applications," *Proceedings of ICRIC 2019: Recent Innovations in Computing*, pp. 47–63, 2020.

- [29] L. Deng, "A tutorial survey of architectures, algorithms, and applications for deep learning," APSIPA transactions on Signal and Information Processing, vol. 3, p. e2, 2014.
- [30] A. Arya and S. Sridhar, "Overview of big data analytics technologies in smart grid," 2023.
- [31] J. Han, J. Pei, and H. Tong, *Data mining: concepts and techniques*. Morgan kaufmann, 2022.
- [32] G. Bonaccorso, Machine learning algorithms. Packt Publishing Ltd, 2017.
- [33] A. Arshaghi, M. Ashourian, and L. Ghabeli, "Potato diseases detection and classification using deep learning methods," *Multimedia Tools and Applications*, vol. 82, no. 4, pp. 5725–5742, 2023.
- [34] F. Anowar, S. Sadaoui, and B. Selim, "Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne)," Computer Science Review, vol. 40, p. 100378, 2021.
- [35] I. H. Sarker, Y. B. Abushark, F. Alsolami, and A. I. Khan, "Intrudtree: a machine learning based cyber security intrusion detection model," *Symmetry*, vol. 12, no. 5, p. 754, 2020.
- [36] I. H. Sarker, Y. B. Abushark, and A. I. Khan, "Contextpca: Predicting context-aware smartphone apps usage based on machine learning techniques," *Symmetry*, vol. 12, no. 4, p. 499, 2020.
- [37] I. H. Sarker, "Data science and analytics: an overview from data-driven smart computing, decision-making and applications perspective," SN Computer Science, vol. 2, no. 5, p. 377, 2021.
- [38] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.
- [39] A. Ng, "Machine learning yearning: Technical strategy for ai engineers in the era of deep learning. retrieved online at h ttps," 2019.

- [40] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, no. 7553, pp. 436–444, 2015.
- [41] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *Ieee access*, vol. 6, pp. 35365–35381, 2018.
- [42] L. Deng, D. Yu et al., "Deep learning: methods and applications," Foundations and trends® in signal processing, vol. 7, no. 3–4, pp. 197–387, 2014.
- [43] I. H. Sarker, A. Kayes, and P. Watters, "Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage," *Journal of Big Data*, vol. 6, no. 1, pp. 1–28, 2019.
- [44] L. Deng, "Three classes of deep learning architectures and their applications: a tutorial survey," APSIPA transactions on signal and information processing, vol. 57, p. 58, 2012.
- [45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., "Scikit-learn: Machine learning in python," the Journal of machine Learning research, vol. 12, pp. 2825–2830, 2011.
- [46] I. H. Sarker, M. H. Furhad, and R. Nowrozy, "Ai-driven cybersecurity: an overview, security intelligence modeling and research directions," SN Computer Science, vol. 2, pp. 1–18, 2021.
- [47] I. H. Sarker, "Deep cybersecurity: a comprehensive overview from neural network and deep learning perspective," *SN Computer Science*, vol. 2, no. 3, p. 154, 2021.
- [48] A. Géron, Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. "O'Reilly Media, Inc.", 2022.
- [49] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

- [50] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine* intelligence, vol. 37, no. 9, pp. 1904–1916, 2015.
- [51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.
- [52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
- [53] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.
- [54] F. Nazir, M. N. Majeed, M. A. Ghazanfar, and M. Maqsood, "Mispronunciation detection using deep convolutional neural network features and transfer learning-based model for arabic phonemes," *IEEE Access*, vol. 7, pp. 52589–52608, 2019.
- [55] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech communication*, vol. 30, no. 2-3, pp. 95–108, 2000.
- [56] H. Franco, L. Neumeyer, M. Ramos, and H. Bratt, "Automatic detection of phonelevel mispronunciation for language learning," in Sixth European Conference on Speech Communication and Technology, 1999.
- [57] H. Strik, K. Truong, F. De Wet, and C. Cucchiarini, "Comparing different approaches for automatic pronunciation error detection," *Speech communication*, vol. 51, no. 10, pp. 845–852, 2009.
- [58] N. Minematsu, S. Asakawa, and K. Hirose, "Structural representation of the pronunciation and its use for call," in 2006 IEEE Spoken Language Technology Workshop. IEEE, 2006, pp. 126–129.

- [59] Y.-B. Wang and L.-S. Lee, "Improved approaches of modeling and detecting error patterns with empirical analysis for computer-aided pronunciation training," in 2012 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2012, pp. 5049–5052.
- [60] F. Zhang, C. Huang, F. K. Soong, M. Chu, and R. Wang, "Automatic mispronunciation detection for mandarin," in 2008 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2008, pp. 5077–5080.
- [61] A. Ito, Y.-L. Lim, M. Suzuki, and S. Makino, "Pronunciation error detection method based on error rule clustering using a decision tree," in *Ninth European Conference* on Speech Communication and Technology, 2005.
- [62] G. Georgoulas, V. C. Georgopoulos, and C. D. Stylios, "Speech sound classification and detection of articulation disorders with support vector machines and wavelets," in 2006 International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 2006, pp. 2199–2202.
- [63] I. Amdal, M. H. Johnsen, and E. Versvik, "Automatic evaluation of quantity contrast in non-native norwegian speech," in *International Workshop on Speech and Language* Technology in Education, 2009.
- [64] H. Li, J. Liang, S. Wang, and B. Xu, "An efficient mispronounciation detction method using glds-sym and formant enhanced features," in 2009 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2009, pp. 4845–4848.
- [65] S. Wei, G. Hu, Y. Hu, and R.-H. Wang, "A new method for mispronunciation detection using support vector machine based on pronunciation space models," Speech Communication, vol. 51, no. 10, pp. 896–905, 2009.
- [66] M. Maqsood, "Feature selection for arabic mispronunciation detection based on sequential floating forward selection and data mining classifiers," Pakistan Journal of Science, vol. 68, no. 4, 2016.
- [67] S. Akhtar, F. Hussain, F. R. Raja, M. Ehatisham-ul haq, N. K. Baloch, F. Ishmanov, and Y. B. Zikria, "Improving mispronunciation detection of arabic words for non-

- native learners using deep convolutional neural network features," *Electronics*, vol. 9, no. 6, p. 963, 2020.
- [68] M. Wu, K. Li, W.-K. Leung, and H. Meng, "Transformer based end-to-end mispronunciation detection and diagnosis." in *Interspeech*, 2021, pp. 3954–3958.
- [69] N. F. Chen, R. Tong, D. Wee, P. Lee, B. Ma, and H. Li, "icall corpus: Mandarin chinese spoken by non-native speakers of european descent," in *Sixteenth Annual Conference* of the International Speech Communication Association, 2015.
- [70] S. A. Fulop and K. Fitz, "A spectrogram for the twenty-first century," *Acoustics today*, vol. 2, no. 3, pp. 26–33, 2006.
- [71] B. Akera, "Alexnet: A brief review," May 2020. [Online]. Available: https://medium.com/ai-research-lab-kampala/alexnet-a-brief-review-14979ce7cc84
- [72] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari, "The history began from alexnet: A comprehensive survey on deep learning approaches," arXiv preprint arXiv:1803.01164, 2018.
- [73] G. Boesch, "Vgg very deep convolutional networks (vggnet) what you need to know viso.ai," Oct 2021. [Online]. Available: https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/
- [74] Jan 2019. [Online]. Available: https://iq.opengenus.org/resnet/
- [75] S. R. Sapireddy, "Resnet-50: Introduction," Jul 2023. [Online]. Available: https://srsapireddy.medium.com/resnet-50-introduction-b5435fdba66f
- [76] wisdomml, "Understanding resnet-50 in depth: Architecture, skip connections, and advantages over other networks wisdom ml," Mar 2023. [Online]. Available: https://wisdomml.in/understanding-resnet-50-in-depth-architecture-skip-\connections-and-advantages-over-other-networks/
- [77] E. Ma, "Data augmentation for audio," Jun 2019. [Online]. Available: https://medium.com/@makcedward/data-augmentation-for-audio-76912b01fdf6

- [78] C. Molnar, "Interpretable machine learning," Aug 2023. [Online]. Available: https://christophm.github.io/interpretable-ml-book/cnn-features.html
- [79] U. Fayyad and K. Irani, "Multi-interval discretisation of continuous-valued attributes,[in:] proceedings of the xiii international joint conference on artificial intelligence," 1993.
- [80] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, The University of Waikato, 1999.
- [81] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM computing surveys (CSUR)*, vol. 50, no. 6, pp. 1–45, 2017.
- [82] H. Liu and R. Setiono, "Chi2: Feature selection and discretization of numeric attributes," in *Proceedings of 7th IEEE international conference on tools with artificial intelligence*. Ieee, 1995, pp. 388–391.
- [83] [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn. metrics.precision_score.html

Appendices

Appendix A

CNN Features-based Model Hyperparameter Tuning and Testing Results: Diagrams and Tables

Hyperparameter Tuning (k) and Testing without Feature Selection in Original Dataset								
Best Results								
Model	1.	Validation	Test	Test	Test			
wiodei	k	BPV	PPV	NPV	BPV			
AlexNet: FC1	9	0.62	0.82	0.35	0.59			
AlexNet: FC2	4	0.58	0.82	0.26	0.54			
VGG19: FC1	9	0.60	0.84	0.38	0.61			
VGG19: FC2	5	0.63	0.86	0.45	0.66			
ResNet50: AvgPool	5	0.64	0.84	0.50	0.67			

Table A.1: CNN Features-based Hyperparameter Tuning on k and Testing in Original Dataset

Hyperparameter Tuning (k) and Testing without Feature Selection in Synthetic Dataset								
Best Results								
Martal L Validation Test Test Test								
Model k	K	BPV	PPV	NPV	BPV			
AlexNet: FC1	9	0.90	0.88	0.90	0.89			
AlexNet: FC2	9	0.88	0.83	0.94	0.88			
VGG19: FC1	10	0.89	0.82	0.89	0.85			
VGG19: FC2	1	0.90	0.82	0.91	0.87			
ResNet50: AvgPool	9	0.89	0.84	0.92	0.88			

Table A.2: CNN Features-based Hyperparameter Tuning on k and Testing in Synthetic Dataset

Hyperparameter Tuning (number of features) and Testing with Feature Selection in Original Dataset								
Best Results								
Model	Number of Features	Validation BPV	Test PPV	Test NPV	Test BPV			
AlexNet: FC1	245	0.69	0.82	0.43	0.62			
AlexNet: FC2	88	0.62	0.83	0.29	0.56			
VGG19: FC1	188	0.60	0.84	0.45	0.65			
VGG19: FC2	47	0.72	0.85	0.37	0.61			
ResNet50: AvgPool	346	0.65	0.82	0.32	0.57			

Table A.3: CNN Features-based Hyperparameter Tuning with Feature Selection and Testing in Original Dataset

Hyperparameter Tuning (number of features) and Testing with Feature Selection in Synthetic Dataset							
$\begin{array}{ c c c c c c c c c c c c c c c c c c c$							
AlexNet: FC1	360	0.92	0.81	0.86	0.84		
AlexNet: FC2	249	0.91	0.84	0.94	0.89		
VGG19: FC1	249	0.91	0.80	0.82	0.81		
VGG19: FC2	166	0.90	0.80	0.82	0.81		
ResNet50: AvgPool	386	0.90	0.78	0.94	0.86		

Table A.4: CNN Features-based Hyperparameter Tuning with Feature Selection and Testing in Synthetic Dataset

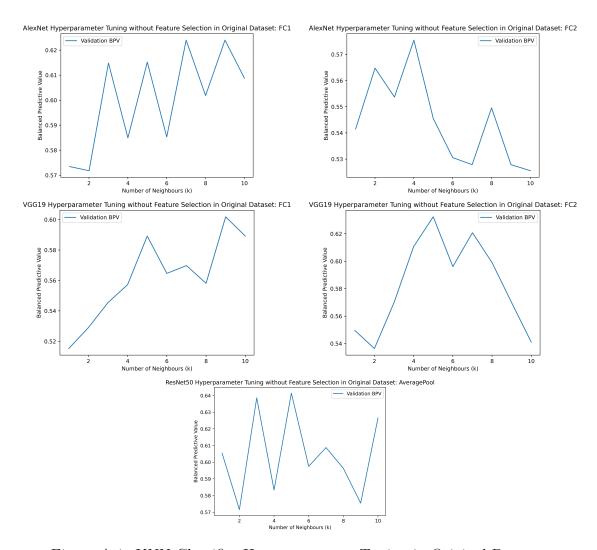


Figure A.1: KNN Classifier Hyperparameter Tuning in Original Dataset.

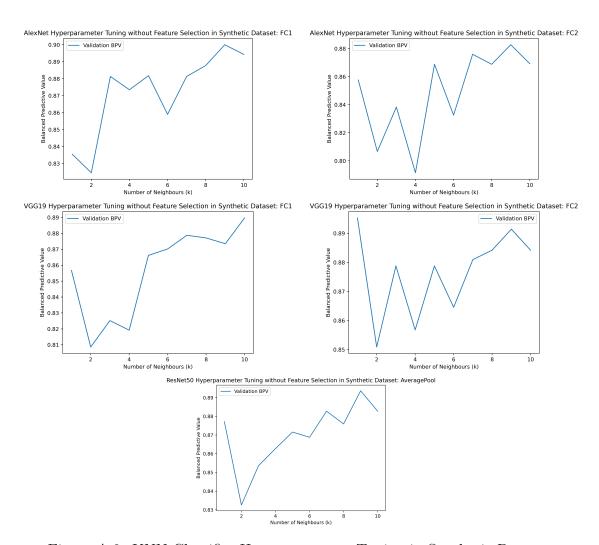


Figure A.2: KNN Classifier Hyperparameter Tuning in Synthetic Dataset.

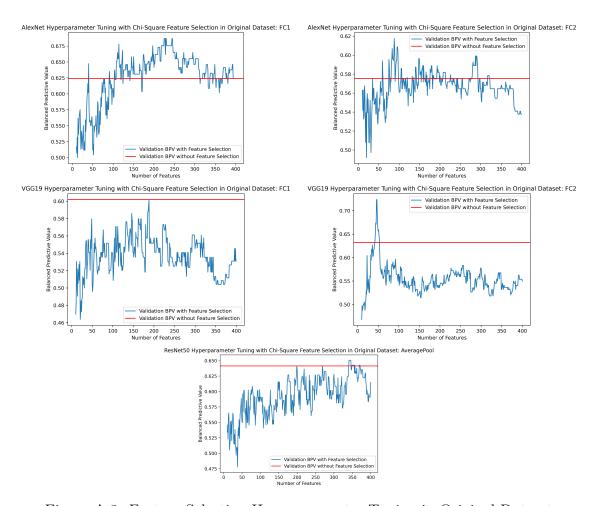


Figure A.3: Feature Selection Hyperparameter Tuning in Original Dataset

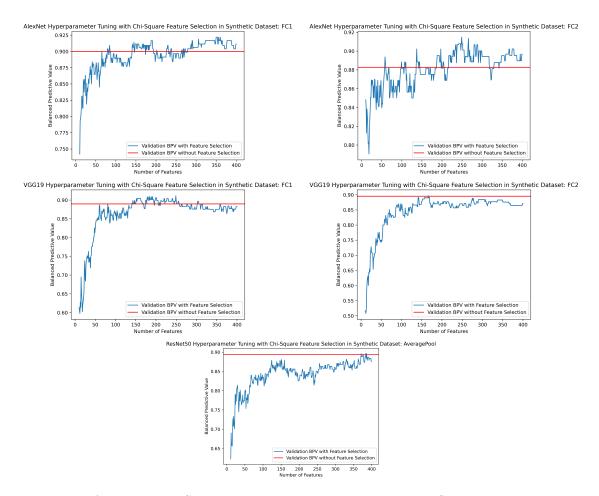


Figure A.4: Feature Selection Hyperparameter Tuning in Synthetic Dataset

Appendix B

CNN Transfer Learning-based Model Hyperparameter Tuning and Testing Results: Diagrams and Tables

	0.005245738771	0.003630359205	0.002247478746	0.001242772164	0.0007267619399	0.01363959793	0.06347851029	0.004649425052	0.01593369348	0.0005814781809	0.0001250668514	0.0004167142515	0.0004608885029	0.001341633506	0.001602820534	0.008691128326	0.04378308448	0.04755932478	0.0002767449059	0.04283669504	0.0006186200733	0.002102868451	0.001436858437	0.0003357943216	0.0002201257156	0.06447338379	0.002312216418	0.03815056208	0.001489122605	0.0008639901321	0.0005665775196	0.006938471816
config/hn2	1024	1024	1024	64	256	1024	20:48	20:48	128	20:48	20:48	256	512	4096	20:48	20:48	20:48	64	32	512	19	512	128	64	20.48	512	9601	1024	64	512	128	512
config/hn1	512	4096	1024	256	32	19	1024	19	25	512	512	2048	256	64	64	1024	4096	512	1024	512	2048	4096	256	19	4096	2048	4096	1024	19	9601	32	2048
config/batch_size	64	128	512	128	64	512	512	512	128	64	256	64	256	64	512	64	128	64	128	128	256	64	256	128	128	256	512	512	128	128	128	256
re iterations since restore	43	60	8	00	2	00	47	00	00	11	00	22	3	13	3	2	4	2	33	1	3	33	00	2	3	00	00	3	10	9	85	
time since restore	1212.169877	81.12207341	82.56837416	76.19372654	55.88073921	84.88647914	1347.283412	76.3599875	81.98223925	327.2882318	74.19143796	675.6061273	89.51412773	413.92855	90.84553957	219.5180748	128.6603851	-	90.05019069	31.12317562	90.96127868	1152,30874	89.98450422	63.59335828	98.20345831	100.8358257	107.781188	89.46723366	279.7443788	177.1907592	81.65955019	209.9061475
node_ip	201.159.221.20		201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20		201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20
hostname	1212.169877 2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	2557764 dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	1695055922 84.88647914 2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	2557764 dgx-node-0-0.cedia.edu.ec	2557764 dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	1695059393 675.6061273 2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	2557764 dgx-node-0-0.cedia.edu.ec	2557764 dgx-node-0-0.cedia.edu.ec	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	1695062659 219.5180748 2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	2557764 dgx-node-0-0.cedia.edu.ec	2557764 dgx-node-0-0.cedia.edu.ec	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	1695065209 63.59335828 2557764 dgx-node-0-0.cedia.edu.ec	2557764 dgx-node-0-0.cedia.edu.ec	1695065439 100.8358257 2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	2557764 dgx-node-0-0.cedia.edu.ec	2557764 dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec
time total s pid	12.169877 2557764	81.12207341 2557764	82.56837416 2557764	76.19372654 2557764	55.88073921 2557764	.88647914 2557764	1347.283412 2557764	76.3599875 2557764	.98223925 2557764	7.2882318 2557764	.19143796 2557764	5.6061273 2557764			.84553957 2557764	9.5180748 2557764	8.6603851 2557764	.97985411 2557764	.05019069 2557764	31.12317562 2557764	.96127868 2557764		.98450422 2557764	.59335828 2557764	98.20345831 2557764	0.8358257 2557764	7.781188 2557764	.46723366 2557764	279.7443788 2557764	177.1907592 2557764	.65955019 2557764	9.9061475 2557764
	1695054909 12	1695055595 81	1695055677 82	1695055753 76	1695055809 55	395055922 84	1695057269 13	1695057847 76	1695057929 81.98223925	395058256 32	395058717 74	395059393 67	1695060843 89.51412773	1695061257 413.92855	1695062440 90.84553957	395062659 21	395062946 12	1695063075 64.97985411	1695063195 90.05019069	1695063226 31	922063376 90	395064528 11	1695065146 89.98450422	395065209 63	1695065338 98	395065439 10	395065547 10	395065637 89	1695066006 27	1695066237 17	395066319 81	395066529 20
	2023-09-18_16-35-09	2023-09-18_16-46-35	2023-09-18_16-47-57	2023-09-18_16-49-13	2023-09-18_16-50-09	80559_00005 2023-09-18_16-52-02 14	80559_00006 2023-09-18_17-14-29 14	2023-09-18_17-24-07	2023-09-18_17-25-29	80559_00009 2023-09-18_17-30-56 1695058256 327.2882318 2557764 dyx-node-0-0.cedia_edu.ec 201.159.221.30	80559_00010 2023-09-18_17-38-37 1695058717 74_19143796 2557764 dgx-node-0-0-cedia.edu.ec 201.159_221_20	2023-09-18_17-49-53	2023-09-18_18-14-03	2023-09-18_18-20-57	2023-09-18_18-40-40	80559 00015 2023 09 18 18 44 19 16	80559_00016 2023-09-18_18-49-06 1695062946 128.6603851	2023-09-18_18-51-15	2023-09-18_18-53-15	2023-09-18-18-53-46	80559_00020_2023-09-18_18-56-16_1695063376_90.96127868_2557764_dgx-node-0-0.cedia.edu.ec_201.159.221.20	21 2023-09-18_19-15-28 1695064528 1152.30874	2023-09-18_19-25-46	2023-09-18_19-26-49	2023-09-18_19-28-58	2023-09-18_19-30-39	80559,00026 2023-09-18,19-32-27 1695065547 107.781188	27 2023-09-18_19-33-57 1695065637 89.46723366	2023-09-18_19-40-06	2023-09-18_19-43-57	80559_00030 2023-09-18_19-45-19 1695066319 81.65955019 2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	80559,00031 2023-09-18_19-48-49 1695066529 209.9061475 2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.30 209.9061475
trial id	80559_00000	80559 00001	80559_00002	80559_00003	80559_00004	80559_000	80559_000	80559_00007	80229-00008	80559,000	80559,000	80559,00011	80559,00012	80559,00013	80559_00014	80559_000	80559_000	80559_00017	80229 00018	80559 00019	80559_000	80559_00021	80559,00022	80559,00023	80559,00024	80559,00025	80559,000	80559_00027	80559,00028	80559_00029	80559_000	80559_000
training iteration	43	3	3	3	2	8	47	80	80	11	89	22	3	13	3	2	4	2	3	1	3	33	8	2	3	80	8	3	10	9	33	7
nt done	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE
time this iter s should checkpoint	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
time this iter	28.27	26.58	25.07	25.46	24.47	26.89	30.83	26.33	26.99	26.79	24.34	30.32	26.94	30.61	28.81	29.92	33.35	29.60	27.02	31.12	32.97	33.47	34.49	36.30	31.68	32.18	30.16	30.51	25.64	28.49	27.43	30.60
v-npv	0.55	0.71	0.32	29.0	0.50	0.45	0.59	0.58	0.43	0.63	0.21	0.59	0.19	0.50	0.37	0.56	0.43	0.45	0.42	0.64	0.32	0.55	0.35	0.35	0.21	0.42	0.28	0.57	0.50	0.58	0.41	0.59
v v-ppv	0.85	0.82	0.82	0.83	0.82	0.82	0.84	0.83	0.81	0.82	0.81	0.84	08.0	0.83	0.82	0.84	0.81	0.84	0.83	0.83	0.82	0.88	0.82	0.84	08:0	0.82	0.83	0.83	0.82	0.85	0.82	98.0
v tr_npv	0.91	Н	0.84	98'0	0.84	0.87	06'0	0.87	0.87	88.0	0.73	Т	0.82	0.89	0.84	0.88	0.87	0.80	0.84	0.82	0.83	0.91	0.85	08'0	0.82	98'0	0.84	0.87	88.0	0.87	0.84	0.88
pv tr.ppv	0.82	0.76	7 0.74	0.75	0.73	1.00	0.80	0.72	2 0.67	82.0	0.70	82.0	0.72	0.75	0.73	0.77	92.0	Н	0.71	8 0.69	7 0.72	0.82	0.73	0.70	0.72	0.76	0.74	0.75	9.70	92.0	0.73	0.73 0.77
pv v-bpv	0.70	2 0.77	0.57	0.75	9970	9.04	5 0.72	0.71	7 0.62	3 0.72	0.51	0.72	0.20	9970	90'0	3 0.70	1 0.62	970	7 0.62	0.73	8 0.57	0.71	0.59	90'0	7 0.51	0.62	0.55	0.70	99.0	0.71	8 0.62	0.73
oss tr_bpv	0.86	5 0.82	3 0.79	2 0.80	9 0.79	0.79	0.85	1 0.79	4 0.77	9 0.83	0.71	9 0.84	0.77	2 0.82	3 0.78	2 0.83	5 0.81	1 0.74	2 0.77	0.76	8 0.78	98'0	3 0.79	1 0.75	1 0.77	2 0.81	2 0.79	3 0.81	3 0.82	7 0.81	4 0.78	0.43 0.83
tr_loss v_loss	25 0.40	34 0.45	39 0.53	96 0.45	89 0.49	02.0	27 0.40	15.0 78	9 0.54	11 0.39	0.80	68 0.39	13 0.59	2 0.42	89 0.53	25 0.42	35 0.45	Г	11 0.52	13 0.50	11 0.58	25 0.39	88 0.53	19.0	19.0	96 0.52	9 0.62	35 0.43	3 0.43	74.0 47	0.54	
tr_	0.25	1 0.34	2 0.39	3 0.36	4 0.39	5 0.38	6 0.27	7 0.37	8 0.39	9 0.31	10 0.52	11 0.29	12 0.43	13 0.32	14 0.39	15 0.32	16 0.35	17 0.45	18 0.41	19 0.43	20 0.41	21 0.25	22 0.38	23 0.45	24 0.43	25 0.36	26 0.39	27 0.35	28 0.33	29 0.34	30 0.40	31 0.31

Table B.1: AlexNet Hyperparameter Tuning on Original Dataset: best results per trial using validation loss as metric.

conng/ir	0.005245738771	0.003630359205	0.002247478746	0.001242772164	0.0007267619399	0.01363959793	0.06347851029	0.004649425052	0.01593369348	0.0005814781809	0.0001250668514	0.0004167142515	0.0004608885029	0.001341633506	0.001602820534	0.008691128326	0.04378308448	0.04755932478	0.0002767449059	0.04283669504	0.0006186200733	0.002102868451	0.001436858437	0.0003357943216	0.0002201257156	0.06447338379	0.002312216418	0.03815056208	0.001489122605	0.0008639901321	0.0005665775196	0.006938471816	
conngy nnz	1024	1024	1024	64	256	1024	2048	2048	128	2048	2048	256	512	4096	2048	2048	2048	64	25	512	64	512	128	19	2048	512	4096	1024	64	512	128	512	
conng/nm1	512	4096	1024	256	32	64	1024	64	32	512	512	2048	256	64	64	1024	4096	512	1024	512	2048	4096	256	64	4096	2048	4096	1024	64	4096	32	2048	
ang batch_size	19	128	512	128	64	512	512	512	128	64	256	64	256	19	512	- 64	128	64	128	128	256	19	256	128	128	256	512	512	128	128	128	256	
regardons since restore config batch size config/hit config/hiz	6	3	2	3	2	-	10	3	1	*	3	9	2	12	3	1	-	2	3	-	3	13	3	2	2	1	3	3	8	9	3	2	
time since restore	213.0833924	81.12207341	57.4969461	76.19372654	55.88073921	31.12829971	290,5979261	76.3599875	25.8458662	239.579946	74.19143796	186.8286638	62.5782845	383,3221371	90.84553957	31.01267457	33.10567164	64.97985411	90.05019069	31.12317562	90.96127868	452.9366724	89.98450422	63.59335828	66.51873994	35.62668085	107.781188	89.46723366	227.0733752	177.1907592	81.65955019	57.02800298	
n drapou	201.159.221.20	201.159.221.20	201.159.221.20	01.159.221.20	201.159.221.20	201.159.221.20	01.159.221.20	01.159.221.20	201.159.221.20	201.159.221.20	01.159.221.20	01.159.221.20	201.159.221.20	201.159.221.20	01.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	01.159.221.20	01.159.221.20	201.159.221.20	01.159.221.20	01.159.221.20	201.159.221.20	201.159.221.20	01.159.221.20	01.159.221.20	01.159.221.20	201.159.221.20	01.159.221.20	01.159.221.20	01.159.221.20	
nostname	dgx-node-0-0-cedia.ed 2	dgx-node-0-0.cedia.edu.ec 2	dgx-node-0-0.cedia.edu.ec 2	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.codia.odu.oc	dgx-node-0-0.cedia.edu.ec 2	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0-cedia.ec 2	dgx-node-0-0.cedia.edu.ee 2	dgx-node-0-0.cedia.edu.ec 201.159.221.20	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec 2	dgx-node-0-0.cedia.edu.ec 2	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0-cedia.ec 2	dgx-node-0-0.cedia.edu.ee 2	dgx-node-0-0.cedia.ed 2	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec 2	dgx-node-0-0.cedia.edu.ec 201.159.221.20	2557764 dgx-node-0-0.cedia.edu.ee 201.159.221.20	dgx-node-0-0-cedia.ed 2	dgx-node-0-0.cedia.edu.ec 2	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	2557764 dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec 2	1695066237 177.1907592 2557764 dgx-node-0-0.cedia.edu.ee 201.159.221.20	dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0 cedia.edu.ee 201.159.221.20	
OH .	4 dgx-node-0			4 dgx-node-6	4 dgx-node-6		4 dgx-node-0	4 dgx-node-0	4 dgx-node-6		4 dgx-node-0	4 dgx-node-0	4 dgx-node-0		4 dgx-node-0	4 dgx-node-0		4 dgx-node-6	4 dgx-node-6	4 dgx-node-0	4 dgx-node-0	4 dgx-node-0	4 dgx-node-6	4 dgx-node-6		4 dgx-node-6	4 dgx-node-0	4 dgx-node-0		4 dgx-node-6	4 dgx-node-6	4 dgx-node-0	
als pid	3924 2557764	7341 2557764	461 2557764		3921 2557764	9971 2557764			662 2557764	946 2557764	3796 2557764		845 2557764	1371 2557764		7457 2557764	7164 2557764	5411 2557764	3069 255776		7868 2557764	3724 2557764	M22 255776	5828 2557764	3994 2557764	3085 255776		3366 2557764	3752 2557764	7592 255776	5019 2557764	1298 2557764	
mestamp time_total_s	909 213.0833924	595 81.12207341	652 57.4969461	753 76.19372654	809 55.88073921	868 31.1282997.	1695056213 290.5979261	847 76.3599875	873 25.8458662	169 239.579946	717 74.19143796		816 62.5782845	227 383.3221371	1695062440 90.84553957	471 31.01267457	851 33.10567164	075 64.97985411	195 90.05018	226 31.12317562	1695063376 90.96127868	829 452.9366724	146 89.98450	209 63.59335828	307 66.51873994	1695065374 35.62668085	547 107.781188		953 227.0733752	237 177.1907	1695066319 81.65955019	376 57.02800298	
timesta	8-29 1695053909	6-35 1695055595	7-32 1695055652	9-13 1695055753	0-09 1695055809	1-08 1695055868	S-53 1695056	1-07 1695057847	4-33 1695057873	9-29 1695058169	8-37 1695058717	1-44 1695058904	3-36 1695060816	0-27 1695061227	0-40 1695062	1-11 1695062471	7-31 1695062851	1-15 1695063075	3-15 1695063	3-46 1695063		3-49 1695063829	5-46 1695063	6-49 1695065209		9-34 1695065	2-27 1695065547	3-57 1695065637	9-13 1695065953		9-19 1695066	6-16 1695066376	
date	2023-09-18,16-18-29	2023-09-18_16-46-35	2023-09-18_16-47-32	80559 00003 2023-09-18 16-49-13	80559 00004 2023-09-18 16-50-09	2023-09-18_16-51-08	80559_00006 2023-09-18_16-56-53	80559,00007 2023-09-18,17-24-07	80559,00008 2023-09-18,17-24-33	80559,00009 2023-09-18,17-29-29	80559 00010 2023-09-18 17-38-37	80559 00011 2023-09-18 17-41-44	80559 00012 2023-09-18 18-13-36	2023-09-18_18-20-27	80259,00014 2023-09-18,18-40-40	80559_00015 2023-09-18_18-41-11	80559,00016 2023-09-18,18-47-31	2023-09-18_18-51-15	80559 00018 2023-09-18 18-53-15 1695063195 90.05019069	80559 00019 2023-09-18 18-53-46 1695063226	80559 00020 2023-09-18 18-56-16	2023-09-18,19-03-49	80559,00022 2023-09-18,19-25-46 1695065146 89.98450422	80559,00023 2023-09-18,19-26-49	80559 00024 2023-09-18 19-28-27	80559 00025 2023-09-18 19-29-34	80559 00026 2023-09-18 19-32-27	80559 00027 2023-09-18 19-33-57	2023-09-18_19-39-13	80559_00029 2023-09-18_19-43-57	80559,00030 2023-09-18,19-45-19	2023-09-18_19-46-16	
trialid	80229,00000	80559,00001	80559,00002	80229,00003	80559 00004	80559,00005	90000762208	20229,00007	80229,00008	80229,00009	80559,00010	80559 00011	80559,00012	80259,00013	80229,00014	80229,00015	80229,00016	80559,00017	80229,00018	80229 00019	80559,00020	80559,00021	80559,00022	80559,00023	80559,00024	80559 00025	80229 00026	80559,00027	80229,00028	80229,00029	80229,00030	80559,00031	
training_iteration	6	3	2		2	-	10	3	-	×	3	9	2	12	3	-	_	2		-		13	3	2	2	-	3	3	8	9	3	2	
done	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	
r s snould checkpoint	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	
time_time_iter_s	28.67480755	26.58422947	28.46396589	25.46013546	24.47234631	31.12829971	33.1827395	26,32916808	25.8458662	29.57932353	24.33757663	32.65491033	30.74627256	33.465307	28.80972838	31.01267457	33.10567164	29.60378909	27.02413082	31.12317562	32.97423911	34.83539414	34.49463463	36.29548645	33.86753559	35.62668085	30.15644932	30,51266694	30.04001307	28.48843503	27.43429375	26.14257097	
v-npv	0.75	0.7142857143	0.3142857143	19999999999	0.5	0.4545454545	0.8181818182	0.5833333333		19999999999	0.206185567	0.72727273	0.2641509434	199999999990	0.3684210526	0.6363636364	-	0.4545454545	0.4210526316	0.6363636364	0.3214285714	7.0	0.347826087	0.3548387097	0.2558139535	0.5454545455	0.2765957447	0.5714285714	0.8571.428571	0.5789473684	0.4117647059	0.6666666667	
^dd-	0.8431372549	0.8227848101	0.8307692308	0.8269230769	0.8235294118	0.8181818182	0.8441558442	0.8300653595	0.8310810811	0.8176100629	0.8088235294	0.8376623377	0.8303571429	0.8366013072	0.8219178082	0.8311688312	0.8148148148	0.8391608392	0.8287671233	0.8311688312	0.8248175182	0.8322580645	0.823943662	0.8358208955	0.8196721311	0.8246753247	0.8305084746	0.8344370861	0.8291139241	0.8493150685	0.8243243243	0.8269230769	
tr_npv	0.8824493956	0.8716079476	0.8286182813	0.8603353604	0.8416131276	0.7944450107	0.8786987384	0.8676304524	0.8099031968	0.8764511222	0.7307591042	0.8716113731	0.7838471014	0.884433269	0.8432451042	0.8342725941	0.8364943446	0.8041228229	0.8376598951	0.8245213056	0.8349864788	0.8928773911	0.8535040844	0.8032009818	0.7907173904	0.8334204542	0.8430650116	0.8676877498	0.871379751	0.8720471892	0.8404321671	0.8633023397	
v_opv tr_npv	0.775773264	0.7592238938	0.7235352472	0.7460528517	0.7323094013	0.4340101308	0.751748979	0.7223003459	0.5813347924	0.7654991386	0.6959600989	0.7544963018	0.7094305852	0.7490864874	0.7261341634	0.6979782408	0.7043209872	0.6780250533	0.7075634388	0.6890882695	0.7185264154	0.7899993222	0.7347964206	0.7021438766	0.7170441638	0.6847255811	0.7396718501	0.7518351449	0.7560662426	0.7561103929	0.7252340079	0.7459236389	
v-opv	0.7965686275	0.7685352622).5725274725	0.7467948718	0.6617647059	3,6363636364	3311688312	7.066993464	0.6508346582).7421383648),5075045482	0.7824675325),5472540431	0.7516339869).5951694304	0.7337662338	3.9074074074	3.6468531469	3.6249098774	3.7337662338),5731230448	0.7661290323).5858848745).5953298026),5377430423	0.6850649351	0.5535521096	0.7029328288	3.8431283906	0.7141312185	0.6180445151	3.7467948718	
rr-pbv	0.4246509075 0.8291113298 0.7965686275 0.775773264 0.8824493956 0.8431372549	0.8154159207 0.7685352622 0.7592238938 0.8716079476 0.8227848101 0.7142857143	0.7760767643 0.5725274725 0.7235352472 0.8286182813 0.8307692308 0.3142857143	3 0.3831292627 0.4531677663 0.803194106 0.7467948718 0.7460528517 0.8603353604 0.8269230769 0.6666666667 25.46013546 0.866666667 0.8666666667 0.8666666667 0.8666666667 0.8666666667 0.8666666667 0.8666666667 0.866666666667 0.8666666667 0.8666666667 0.8666666667 0.8666666667 0.86666666667 0.86666666667 0.8666666666667 0.8666666666666666666666666666666666666	0.3930536634 0.4869146049 0.7869612645 0.6617647059 0.7323094013 0.8416131276 0.8235294118	0.5453369617 0.6142275707 0.6363636364 0.4340101308 0.7944450107 0.81818182 0.4545454545	6 0.3319672907 0.4437686801 0.8152238687 0.8311688312 0.751748979 0.8786987384 0.8441558442 0.818181818 33.182739	7 + 0.3723849152 + 0.5103389621 + 0.7949653991 + 0.7096993464 + 0.7223003459 + 0.8676304524 + 0.8300653595 + 0.583333333 + 26.3291686	8 0.455712063 0.5528001189 0.6956189946 0.6508346382 0.5813347924 0.8099031968 0.8310810811 0.470582353	9 0.3272071854 0.4789555073 0.8209751304 0.7421383648 0.7654991386 0.8764511222 0.8176100629 0.6666666667	$10 \ 0.5225423263 \ 0.7308565226 \ 0.7133596015 \ 0.5075045482 \ 0.6956600989 \ 0.7307591042 \ 0.8088235294 \ 0.206185567 \ $	11 0.3425876969 0.4408787489 0.8130538374 0.7824675325 0.7544963018 0.8716113731 0.8376623377 0.7272727273 3.2.654910	12 0.4684987578 0.6545013189 0.7466388433 0.5472540431 0.7094305852 0.7838471014 0.8303571429 0.2641509434	0.4395739404 0.8167598782 0.7516339869 0.7490864874 0.884433269 0.8360013072 0.666666667	14 0.3943318177 0.526026739 0.7846896338 0.5951694304 0.7261341634 0.8432451042 0.8219178082 0.3684210526 28.8097285	15 0.4168942085 0.4872290492 0.7661254174 0.7337662338 0.6979782408 0.8342725941 0.8311688312 0.633636364 31.01267457 0.83168942085 0.8311688312 0.831688312 0.8363636364 0.831688312 0.831688311 0.831688312	16 0.4160106218 0.4622173905 0.7704076659 0.9074074074 0.7043209872 0.8364943446 0.8148148148148	$17 \\ 17 \\ 0.4503052427 \\ 0.6115829349 \\ 0.7410739381 \\ 0.6408531469 \\ 0.6780250533 \\ 0.8041228229 \\ 0.8041228229 \\ 0.8391608392 \\ 0.4545454545 \\ 0.8391608392 \\ 0.4545454545 \\ 0.8391608392 \\ 0.8391608$	18 0.411394223 0.5193086267 0.772611667 0.6249098774 0.7075634388 0.8376598961 0.8287671238 0.4210526316 27.02413082 0.8287671238 0.	$19 \ \ 0.4311596547 \ \ \ 0.4962902069 \ \ \ 0.7568047875 \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \$	20 0.4090576023 0.5800320506 0.7767564471 0.5731230448 0.7185264154 0.8349864788 0.8248175182 0.3214285714 32.974289	0.289671219 0.4582444131 0.8414383566 0.7661290323 0.7899993222 0.8928773911 0.8322580645	22 0.3819514037 0.5341044068 0.7941502525 0.5858848745 0.7347964206 0.8539040844 0.823943662 0.347826087 34.49463463	23 0.4482750384 0.6135058999 0.7726724292 0.5953298026 0.7021438766 0.8032009818 0.8358208955 0.3548387097 36.29548645	24 0.4610470675 0.6558214426 0.7538807771 0.5377430423 0.7170441638 0.7907173904 0.8196721311 0.2558139555	25 0.4154289185 0.5254890919 0.7590730177 0.6850649651 0.6847255811 0.8334204542 0.8240753247 0.545454545 35.62668085	26 0.3932550582 0.6157870253 0.7913684308 0.553551096 0.7396718301 0.8430650116 0.8303084746 0.2765957447 30.1564495	27 0.3515415344 0.4299615026 0.8007614473 0.7029328288 0.7518351449 0.8670877498 0.8344370861 0.5714285714 30.5129666	0.8137229968 0.8431283906 0.7560662426 0.871379751 0.8291139241	29 0.3425015912 0.4689917266 0.814078791 0.7141312185 0.7561103929 0.8720471892 0.8493150685 0.5789473684 28.4884350	30 + 0.3970871723 + 0.536584965 + 0.7228330875 + 0.6180445151 + 0.7252340079 + 0.8404321671 + 0.8243243243 + 0.4117647039 + 27.4342937 + 0.8404821671 + 0.8243243243 + 0.4117647039 + 27.4342937 + 0.8404821677 + 0.8243243243 + 0.4117647039 + 27.4342937 + 0.8404821677 + 0.8243243243 + 0.4117647039 + 27.4342937 + 0.840482167 + 0.840482167 + 0.840482167 + 0.840482167 + 0.8404821 + 0.8404821 + 0.8404821 + 0.8404821 + 0.8404821 + 0.840481 + 0	0.35813935 0.477889484 0.8046129893 0.7467948718 0.7459236389 0.86360866667	
V_088	7.4246509075	0.4452569187	9.606374383	.4531677663	.4869146049	.5453369617	1.4437686801	1.5103389621	.5528001189	.4789555073	7968565226	4408787489	.6545013189	.4395759404	9.526036799	.4872290492	.4622173905	.6115829349	.5193086267	.4962902069	.5800320506	7.4582444131	.5341044068	.6135058999	.6558214426	.5254890919	0.0157876253	7.4299615026	0.443955332	.4689917266	1,5365844965	.4778894484	
TI_DSS	0.313257294 0.	0.3408493166 0.	0.4205935128 0.606374383	0.3631292627 0	0.3930536634 0	5 0.4678936601 0.	0.3319672907 0	0.3723849152 0	0.4555712063 0	0.3272071854 0	0.5225423263 0	0.3425876969 0	0.4684987578 0	0.3257321263 0.	0.3943318177 (0.4168942085 0	0.4160106218 0	0.4503052427 0	0.411394223 0	0.4311596547 0	0.4090576023 0	0.289571219 0	0.3819514037 0	0.4482750384 0	0.4610470675 0	0.4154289185 0	0.3932550582 0	0.3515415344 0	28 0.3384910357 0	0.3425015912 0	0.3970871723 0	0.359313935 0	
	0	Ĺ	- 5	8	4	10	9	-	×	6	10	11	12	13	14 (15	16	17	18	19	20	21	22	23	24	22	36	27	28	29	30	31	

Table B.2: AlexNet Hyperparameter Tuning on Original Dataset: best results per trial using BPV as metric.

r	-	-	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
	config/fr	0.001565350872	0.01365161077	0.001005753959	0.0002466962916	0.0001092609267	0.009346488325	0.009948334746	0.03536332723	0.01141031954	0.0405644679	0.0009029105726	71789988717	0.003430193564	0.01022937972	0.02649460034	0.0001403612824	0.0006510680652	0.000451109904	0.00643691461	0.0001186458671	0.0005790499532	0.05227947972	0.08091986224	0.0002076881736	0.0007704250842	0.01478964646	0.0005035920872	0.004343401267	0.082250754	0.0007139516739	0.0519005377	0.00028/16090023
	config/hn2	128	1024	2048	256	128	128	19	1-9	64	1024	64	256	2048	128	64	64	512	2048	128	64	2048	2048	256	4096	128	1024	512	1024	2048	256	2048	1024
t	config/hnl c	128	35	19	256	256	64	256	1024	4096	256	4096	64	256	2048	1024	4096	33	512	128	4096	19	128	2048	256	4096	128	128	256	2048	28	256	512
	onfig/batch_size or	64	256	512	256	128	64	512	128	512	256	128	64	128	64	256	256	512	512	128	256	256	128	128	256	64	512	256	128	128	512	128	512
	iterations since restore config/batch size	45	3	8	8	8	4	4	P	8	9	9	3	13	13	2	3	8	3	8	8	e	1	3	8	1-	8	8	10	3	8	3	8
	time since restore	1578.546121	91.94040465	110.8148446	100,1379797	93.90857291	142.8299851	354.7448144	171.4342225	387,5320859	199.6390762	214.6894631	108.0156209	447.3827367	504.3631818	176.4129517	109.3657529	103.8092754	98.37933612	98.52707291	90.48249054	85.5575037	31.8970747	92.62602663	79.33283424	263,1995299	98,03679156	91.27638316	315.4047122	98.79877448	95.65932918	93.13302779	98.44754195
		201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20
	hostname	dgx-node-0-0.codia.edu.oc	dgx-node-0-0.cxtia.edu.ec	110.8148446 561706 dgx-node-0-0.cedia.edu.ec	561706 dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.codia.edu.oc	9792e_00006 2023-08-18_20-51-08 1695070268 354.7448144 561706 dgx-node-0-0.cedia.cdu.cc 201.159.221.20	561706 dgx-node-0-0.osdia.edu.ec	dgx-node-0-0.cedia.edu.ec	561706 dgx-node-0-0.cedia.edu.ec	561706 dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.codia.edu.oc	dgx-node-0-0.cedia.edu.ec	9792e_00013 2023-09-18_21-49-52 1695073792 504.3631818 561706 dgx-node-0-0.cedia.edu.ec 201.159.221.20	9792e_00014 2023-09-18_21-59-54 1695074394 176.4129517 561706 dgx-node-0-0-cedia-edu.ec 201.159.221.20	dgx-node-0-0.oedia.edu.ec	dgx-node-0-0.cedia.edu.ec	9792e_00017 2023-09-18_22-05-39 1695074739 98.37933612 561706 dgx-node-0-0-cedia-edu-ee 201.159.221.20	561706 dgx-node-0-0.codia.edu.oc 201.159.221.20	dgx-node-0-0.codia.edu.oc	dgx-node-0-0.cedia.edu.ec	561706 dgx-node-0-0.cedia.edu.ec 201.159.221.20	561706 dgx-node-0-0.codia.edu.oc	dgx-node-0-0.codia.edu.oc	dgx-node-0-0.cedia.edu.ec	9792e_00025 2023-09-18_22-25-14 1695075914 98.03679156 561706 dgx-node-0-0.cedia.echn.ec 201.159.221.20	9792e_00026 2023-09-18_22-26-45 1695076005 91.27638316 561706 dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.codia.edu.oc	9792c_00029 2023-09-18_22-36-16 1695076576 95-65932918 561706 dgx-node-0-0-cedia-celu-ce 201.159-221.20	9792e 00030 2023 09-18 22-37-50 1695076670 93.13302779 561706 dgx-node-0-0 oedia.edu.ec	9792e_00031 2023-09-18_22-39-28 1695076768 98.44754195 561706 dgx-node-0-0.cedia.ech.ze 201.159.221.20
L	bid	561706	561706	561706 d		261706	561706	561706 d		561706		561706 d	561706	561706	561706 d	561706 d	561706	561706 d	561706 d		561706 d	561706	561706 d		561706	561706	561706 d	561706 d	561706	561706	561706 d	561706 d	561706 d
	time total s	1578,546121	91.94040465	110.8148446	100.1379797	93.90857291	142.8299851	354,7448144	171.4342225	387,5320859	199.6390762	214.6894631	108.0156209	447.3827367	504.3631818	176.4129517	109.3657529	103.8092754 561706	98.37933612	98.52707291	90.48249054	85.5575037	31.8970747	92.62602663	79.33283424	263.1995299	98.03679156	91.27638316	315.4047122	98.79877448	95.65932918	93.13302779	98.44754195
	_	1695068632	1695069377	395069488	1695069588	1695069682	1695069825	395070268	395070532	1695071012	1695071927	395072141	1695072465	395072913	395073792	395074394	1695074536	1695074640	395074739	395074837	1695075019 90.48249054 561706	1695075105		1695075289	1695075368	1695075631	395075914	395076005	1695076321	1695076480	395076576	95076670	395076768
l	_		-	20-38-08	20-39-48 16			20-21-08	20-55-32 16			21-22-21 16		21-35-13 16	21-49-52 16	21-59-54 16		22-04-00 16	22-05-39 16	22-07-17 16			22-12-17 16	22-14-49 16		⊢	22-25-14 16	22-26-45 16			22-36-16 16	22-37-50 16	22-39-28 16
	date	2023-09-18_20-23-52	2023-09-18_20-36-17	2023-09-18_20-38-08 1695069488	2023-09-18	2023-09-18_20-41-22	2023-09-18_20-43-45	2023-09-18	2023-09-18	2023-09-18_21-03-32	2023-09-18_21-18-47	2023-09-18	2023-09-18_21-27-45	2023-09-18_21-35-13 1695072913	2023-09-18	2023-09-18	2023-09-18	2023-09-18	2023-09-18	2023-09-18	2023-09-18	2023-09-18_22-11-45	2023-09-18	2023-09-18	2023-09-18_22-16-08	2023-09-18_22-20-31	2023-09-18	2023-09-18	2023-09-18_22-32-01	2023-09-18_22-34-40	2023-09-18	2023-09-18	2023-09-18
	trial_kt	9792e_00000	9792e_00001	9792e_00002	9792e_00003 2023-09-18_20-39-48	9792e_00004	9792e_00005	792e_00006	9792e.00007 2023-09-18 20-55-32 1695070532	9792e_00008	9792e_00009	9792e_00010 2023-09-18_21-22-21 1695072141 214.6894631	9792e_00011	9792e_00012	792e_00013	792e_00014	9792e_00015 2023-09-18_22-02-16	9792e_00016 2023-09-18_22-04-00	792e_00017	9792e 00018 2023 09-18 22-07-17 1695074837 98.52707291	9792e 00019 2023 09-18 22-10-19	9792e_00020	9792e_00021 2023-09-18_22-12-17 1695075137	9792e 00022 2023 09-18 22-14-49	9792e_00023	9792e_00024	792e_00025	792e_00026	9792e_00027	9792e_00028	792e_00029	792e_00030	792e_00031
Ļ	training_iteration	45 9	3 8	3	3 8	3	4 9	4 9	4 9	3	6 9	9	3	13	13 8	5 9	3 8	3	3	3	3	3 8	1 9	3	3	7	3 8	3	10	3	3	3 8	3 8
t	_	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
	hould_checkpoint	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
	time this iter's should checkpoint	37.07819438	30.44120598	36.61903071	35.85677791	31.82855487	40.6467998	47.02904797	43.17577934	248.6930933	33.58177257	35.52800012	35.1895082	36.12916946	38,0562489	36.47043586	34.12981153	31.29855347	31.21441579	33.72485995	30.99985385	27.93542719	31.8970747	32.17606115	25.20838594	36.27578807	33,0221889	29.36053109	31.79359388	34.37928581	30.410218	29,28661895	33.68008924
	v_npv	0.92	0.9538461538	.9154929577	0.8375	0.835443038	.9692307692	.9565217391	.9714285714	0.9710144928	.9692307692	.9315068493	.9428571429	.9571428571	.9315068493	.9701492537	799999997	.8205128205	.8192771084	.9558823529	STTTTTTT18	0.9552238806	0	0	0.8125	0.9565217391	79999999967	0.858974359	0.9558823529	0.5	.8783783784	0.5	.7804878049
	vqpv	0.8705882353	0.8105263158 (0.8314606742	0.8375	0.8271604938	0.8210526316	0.8461538462 (0.866666667	0.8571428571 (0.8210526316	0.8620689655 (0.844444444 (0.8555555556	0.8620689655 (0.8387096774 (0.8428571429 (0.8048780488 (0.8441558442 (0.8369565217	0.7848101266	0.8279569892	0.5	0.5	0.8125	0.8461538462 (0.8588235294 (0.8414634146	0.8369565217	0	0.8255813953 (0	0.7948717949 (
l	tr_npv).8543908251 ().8570644118 (7694008983	0.7165775601 0.7337447598 0.8375	0.7321818275	38257610795 (38152018527	3374596026	8190313133	8237834235 (38166990832	0.8126126149 (3396645654	0.857752138 (8312803068 (7.292340862	0.7516447083 (7323989648 (0.814956025 (7241032486	021169220	0.3513150387 0.6482278871	0.2830616337 0.2259927069	0.7200087859 0.7372145298	8289252615	7266211130	7283557627	3318320055	0.2556185788 0.2419912957	0.7537321278 (0.2698283165 0.3608525064	0.7146624199 (
	tr_ppv	021565731 (673969446	354704115 0	165775601	136281371	256409321 (577187746 (420646155 (558207145 0	298054689 (669507304 (410503215 (774283858 (755156442	599227912	039073971 0	119016847	19040981	467119975	055041654 0	339860466	513150387	830616337 (200087859	706349937	426384119 (35469436 (707928574 (556185788 (178062481	698283165 (69212983
ŀ	vdd.v	$0.2282496691 \mid 0.8282736991 \mid 0.8052941176 \mid 0.8021565731 \mid 0.8543008251 \mid 0.8705882353$	0.3732534647 0.7622306782 0.8821862348 0.6673969446 0.8570644118 0.8105263158	0.4985633866 0.3444455266 0.7524356549 0.873476816 0.7354704115 0.7694008983 0.8314606742 0.9154929677 36.61903 0.8314606742 0.9154929677 36.61903 0.8314606742 0.9154929677 0.915496967 0.9154967 0	0.8375 0.7	0.553678054 0.4779375196 0.7229009823 0.8313017659 0.7136281371 0.7321818275 0.8271600938 0.835443038 0.835443038 0.83648088 0.83688988 0.83688988 0.83688988 0.83688988 0.83688988 0.83688988 0.83688988 0.83688988 0.83688988 0.83688988 0.83688988 0.83688988 0.83688988 0.83688988 0.83688988 0.83688988 0.83688988 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.836888888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.836888888 0.8368888 0.836888888 0.83688888 0.83688888 0.836888888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.83688888 0.836888888 0.836888888 0.836888888 0.836888888 0.836888888 0.83688888 0.836888888 0.83688888888 0.836888888 0.836888888 0.836888888 0.836888888 0.836888888 0.836888888 0.836888888 0.836888888 0.83688888888 0.836888888 0.836888888 0.836888888 0.836888888 0.8368888888 0.836888888888 0.83688888888 0.83688888888 0.8368888888 0.836888888 0.8368888888 0.8368888888 0.83688	0.2943440974 0.7757010058 0.8051417004 0.7256409321 0.8257610795 0.8210526316 0.9692307692	6 0.4421813615 0.2457338572 0.7864003137 0.9013377926 0.7577187746 0.8152018527 0.8461538462 0.9565217391 47.02904 0.8461813617 0.84	0.4427764911 0.23304829 0.789762109 0.919047619 0.7420646155 0.8374596026 0.8666666667 0.9714285714 43.17577 0.8866666667 0.9714285714 43.17577 0.88666666667 0.986666667 0.9866666667 0.9866666667 0.9866666667 0.9866666667 0.9866666667 0.986666667 0.9866666667 0.9866666667 0.9866666667 0.9866666667 0.9866666667 0.9866666667 0.986666667 0.9866666667 0.986666667 0.986666667 0.9866666667 0.9866666667 0.9866666667 0.986666667 0.9866666667 0.9866666667 0.9866666667 0.9866666667 0.9866666667 0.9866666667 0.9866666667 0.9866666667 0.986666667 0.986666667 0.986666667 0.986666667 0.98666667 0.98666667 0.986666667 0.986666667 0.98666667 0.98666667 0.98666667 0.98666667 0.98666667 0.98666667 0.98666667 0.98666667 0.98666667 0.986666667 0.98666667 0.98666667 0.98666667 0.98666667 0.986666667 0.98666667 0.986666667 0.986666667 0.986666667 0.986666667	0.23218593 0.7874260169 0.9140786749 0.7558207145 0.8190813193 0.8571428571	0.2875904739 0.7767944462 0.8951417004 0.7298054689 0.8237834235 0.8210526316 0.9692307692 0.821626316 0.9692307692 0.821626316 0.9692307692 0.821626316 0.9692307692 0.821626316 0.9692307692 0.969207692 0.969207692 0.969207692 0.969207692 0.969207692 0.969207692 0.969207692 0.969207692 0.969207692 0.969207692 0.969207692 0.969207692 0.969207692 0.969207692 0.969207692 0.969207692 0.969207692 0.969207692 0.969207692	10 0.4313159661 0.2650312781 0.7918249068 0.8967879074 0.7669507304 0.8166990832 0.8620689655 0.9315068493 3.552800012 0.8816990832 0.88169999832 0.8816999832 0.88169999832 0.88169999832 0.88169999832 0.88169999832 0.88169999832 0.881699999999999999999999999999999999999	0.4591784791 0.3054469526 0.7768314682 0.8936507937 0.7410503215 0.8126126149 0.844444444 0.9428571429 0.84444444 0.9428571429 0.844444444 0.9428571429 0.84444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.8444444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.8444444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9428571429 0.844444444 0.9488571429 0.844444444 0.9488571429 0.844444444 0.9488714444444 0.94887144444 0.948871444444 0.9488714444444444444444444444444444444444	0.2211053073 0.8085464756 0.9063492063 0.7774283858 0.8390645654 0.855555556 0.9571428571	0.3864851799 0.2319475114 0.8166289911 0.8067879074 0.7755156442 0.857752138 0.8620689655 0.9315068493 0.8620689655 0.9315068493 0.8620689655 0.8620689659 0.8620689659 0.8620689659 0.8620689659 0.8620689699 0.8620689999 0.862068999 0.862068999 0.862068999 0.862068999 0.862069999 0.862069999 0.862069999 0.862069999 0.862069999 0.8620699999 0.86206999999 0.8620699999999999999999999999999999999999	14 0.4214096078 0.2552779913 0.795751549 0.5044294656 0.7590227912 0.8315803068 0.8387096774 0.9701492537 36.47043	0.5746652566 0.5396606326 0.7165707416 0.8047619048 0.7039073971 0.7292340862 0.8428571429 0.7066666667 0.7066666667 0.7066666667 0.7066666667 0.7066666667 0.7066666667 0.70666666667 0.7066666667 0.70666666667 0.7066666667 0.7066666667 0.70666666667 0.7066666667 0.70666666667 0.70666666667 0.70666666667 0.70666666667 0.70666666667 0.70666666667 0.70666666667 0.70666666667 0.70666666667 0.70666666667 0.70666666667 0.7066666667 0.70666666667 0.70666666667 0.7066666667 0.7066666667 0.7066666667 0.706666667 0.706666667 0.706666667 0.706666667 0.706666667 0.706666667 0.706666667 0.706666667 0.706666667 0.706666667 0.706666667 0.70666667 0.70666667 0.706666667 0.706666667 0.706666667 0.706666667 0	16 0.5364619141 0.489495039 0.7317731965 0.8126954346 0.7119016847 0.7516447083 0.8048780488 0.8205128205 0.820616847 0.820616847 0.820616847 0.820616848 0.8206168888 0.820616888 0.820616888 0.820616888 0.820616888 0.820616888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.820616888 0.8206168888 0.820616888 0.820616888 0.820616888 0.820616888 0.820616888 0.820616888 0.820616888 0.820616888 0.8206168888 0.820616888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.82061688888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.82061688888888 0.82061688888 0.8206168888 0.8206168888 0.82061688888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.82061688888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.8206168888 0.82061688888 0.8206168888 0.820616888888 0.820616888888 0.82061688888 0.820616888888 0.82061	0.5465917346 0.4952264726 0.7257199765 0.8317164763 0.7190409881 0.7323089648 0.8441558442 0.8192771084 31.21441	18 0.4486616516 0.272023302 0.7808340112 0.8064194373 0.7467119975 0.814956025 0.8369565217 0.9558823529 33.72485	0.5684648453 0.543642117 0.714803707 0.7812939522 0.7055041654 0.7241032486 0.7848101266 0.77777778 30.99885 0.7848101266 0.777777778 0.99885 0.7848101266 0.777777778 0.99885	0.4914329208 0.3488953114 0.7554485385 0.8915904349 0.7339860466 0.7769110304 0.8279569892	П	П	0.8125 0.7	0.2331019193 0.7997801276 0.9013377926 0.7706349987 0.8289252615 0.8461538462	25 0.4571474606 0.3160237968 0.7768872048 0.882745098 0.7426384119 0.8111359977 0.858235294 0.9066666667	26 0.5004895647 0.397770077 0.7469125994 0.8502188868 0.735460436 0.7583557627 0.8414634146 0.858974359 29.36063	0.4082954239 0.2457100898 0.8013124314 0.8064194373 0.7707928574 0.8318320055 0.8669565217	0.25 0.2	29 0.5328529063 0.4273747504 0.735769188 0.8519798869 0.7178062481 0.753721278 0.8255813953 0.8783783784 30.4102	0.25 0.2	31 0.5844378433 0.5530332327 0.7033961249 0.7876797999 0.69212983 0.7146624199 0.7348717949 0.7804878049 33.68008
Ì	tr_bpv	8282736991 0.	7622306782 0.	7524356549 C	0.5468704064 0.4644676149 0.72516116 0.8375	7229049823 0.	7757010058 0.	7864603137 0.	.789762109 (7874260169 0.	7767944462 0.	7918249068 0.	7768314682 0.	8085464756 0.	8166338911 0.	.795751549 0.	7165707416 0.	7317731965 0.	7257199765 0.	7808340112 0.	.714803707 0.	7554485385 0.	4997714629	22 0.6938173293 0.6931587458 0.2545271703 0.25		7997801276 0.	7768872048 C	7469125994 0.	8013124314 0.	2488049372	.735769188 0.	30 0.6928990442 0.6932863593 0.3153404115 0.25	7033961249 0.
ŀ	v_loss	282496691 0.	732534647 0.	144455266 0.	344676149 C	73375196 0.	343440974 0.	157338572 0.	3304829 0.	23218593 0.	375904739 0.	50312781 0.	754469526 0.	311053073 0.	319475114 0,	522779913 0.	396606326 0.	89495039 0.	152264726 0.	730233302 0.	43643117 0.	188953114 0.	0.6903615014 0.6932321787 0.4997714629	31587458 0.	0.4733188748 0.7286116576	331019193 0.	160237968 0.	97770077 0.	157100898 0x	28 0.6943747917 0.6936026812 0.2488049372	273747504 0.	132863593 0.	530332327 0.
ŀ		0.3575793322 0.22	0.513154619 0.37	533866 0.34	704064 0.46	578954 0.45	0.4622522172 0.29	\$13615 0.24	764911 0.5		0.4611771021 0.28	159661 0.26	784791 0.36	474837 0.22	\$51709 0.25	99078 0.25	552506 0.55	519141 0.4k	917346 0.45	516516 0.25	548453 0.5-	329208 0.34	515014 0.65	173293 0.65	0.5397835687 0.47	0.4154469091 0.23	474606 0.31	895647 0.38	954239 0.24	747917 0.65	529063 0.42	990442 0.65	378433 0.55
	trioss	0 0.3575.	1 0.5131	2 0.49856	3 0.54685	4 0.55396	5 0.4622.	6 0.44218	7 0.44277	8 0.4401612867	9 0.46115	10 0.4313.	11 0.45915	12 0.3998474837	13 0.38648	14 0.42140	15 0.57460	16 0.53646	17 0.54658	18 0.44860	119 0.56846	20 0.49143	21 0.69030	22 0.69383	23 0.53978	24 0.41544	1 25 0.45714	26 0.50948	27 0.40828	28 0.69435	29 0.5328.	30 0.69285	31 0.58442
•	_			_	_	_				_	_			_	_	_		_	_	_	_	_		_	_	_	_	_	_		_		_

Table B.3: AlexNet Hyperparameter Tuning on Synthetic Dataset: best results per trial using validation loss as metric.

			<u> </u>	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_		_	_	_	_	_	_	$\overline{}$	т
conng/ ir	0.001565350872	0.01365161077	0.001005753959	0.0002466962916	0.0001092609267	0.009346488325	0.009948334746	0.03536332723	0.01141031954	0.0405644679	0.0009029105726	71789988717	0.003430193564	0.01022937972	0.02649460034	0.0001403612824	0.0006510680652	0.000451109904	0.00643691461	0.0001186458671	0.0005790499532	0.05227947972	0.08091986224	0.0002076881736	0.0007704250842	0.01478964646	0.0005035920872	0.004343401267	0.082250754	0.0007139516739	0.0519005377	P. CONDONAS ARCHORDER
conng/nnz	128	1024	2048	256 (128	128	64	64	64	1024	64	256	2048	128	64	64 (512	2048	128	64 (2048 (2048	256	4096 (128	1024	512	1024	2048	256	2048	1004
mng/nmt ox	128	35	64	256	256	64	256	1024	4096	256	4096	64	256	2048	1024	4096	35	512	128	4096	19	128	2048	256	4096	128	128	256	2048	35	256	519
conng/ paren_size conng/nm.	64	256	512	256	128	19	512	128	512	256	128	19	128	64	256	256	512	512	128	256	256	128	128	256	64	512	256	128	128	512	128	519
nerations, since restore to	2	3	23	3	3	3	3	4	3	3	~	3	12	12	4	3	3	8	3	2	3	-	1	2	9	3	3	2	1	3	1	6
age	76.85849118	91.94040465	110.8148446	100.1379797	93.90857291	102.1831853	307.7157664	171.4342225	387,5320859	100.6609488	285.2246866	108.0156209	411.2535672	466,3069329	139,9425159	109.3657529	103.8092754	98.37933612	98.52707291	59.48263669	85.5575037	31.8970747	30.716429	54.1244483	226.9237418	98.03679156	91.27638316	62.43827343	33.68402982	95.65932918	31.29935384	64 76745979
noce_np	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	00 100 001 100
nostname	dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.codia.edu.oc	dgx-node-0-0.codia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0, cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.codia.odu.oc 201.159.221.20	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.codia.edu.oc	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.codia.edu.oc 201.159.221.20	dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.codia.edu.oc 201.159.221.20	dgx-node-0-0.codia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.codia.edu.oc 201.159.221.20	dow node 0.0 codic odn oc
time_total_s pid	76.85849118 561706 0	_	110.8148446 561706 0	100.1379797 561706	93.90857291 561706 <		307.7157664 561706 o	171.4342225 561706	387,5320859 561706 0	100.6609488 561706	285.2246866 561706 c	108.0156209 561706 4	411.2535672 561706 of	466,3069329 561706 0	139,9425159 561706	109.3657529 561706	103.8092754 561706 4	98.37933612 561706 4	98.52707291 561706 0	59.48263669 561706	85.5575037 561706 0	31.8970747 561706 4	30.716429 561706 4	54.1244483 561706 o	226.9237418 561706 0	98.03679156 561706	91.27638316 561706 4	62.43827343 561706 4	33.68402982 561706 o	95.65932918 561706 0	31.29935384 561706	CONTRACTOR CONTRACTOR
timestamp	1695067131	1695069377	1695069488	1695069588	1695069682	1695069785	1695070221	1695070532	1695071012	1695071828	1695072212	1695072465	1695072876	1695073754	1695074357	1695074536	1695074640	1695074739	1695074837	1695074988	1695075105	1695075137	1695075227	1695075343	1695075595	1695075914	1695076005	1695076068	1695076415	1695076576	1695076608	1000000000
carre	2023-09-18_19-58-51	2023-09-18_20-36-17	2023-09-18_20-38-08	2023-09-18_20-39-48	2023-09-18_20-41-22	2023-09-18_20-43-05	2023-09-18_20-50-21	2023-09-18-20-55-32 1695070532	2023-09-18_21-03-32 1695071012	2023-09-18_21-17-08	2023-09-18_21-23-32	2023-09-18_21-27-45 1695072465	2023-09-18_21-34-36 1	2023-09-18_21-49-14	2023-09-18_21-59-17	2023-09-18_22-02-16 1695074536	2023-09-18_22-04-00	2023-09-18_22-05-39	9792e 00018 2023 09-18 22-07-17 1	2023-09-18_22-09-48 1695074988	2023-09-18_22-11-45	2023-09-18_22-12-17	2023-09-18_22-13-47 1695075227	2023-09-18_22-15-43	2023-09-18_22-19-55	2023-09-18_22-25-14	2023-09-18_22-26-45 1695076005	2023-09-18_22-27-48	2023-09-18_22-33-35	2023-09-18_22-36-16 1695076576	9792e 00030 2023 09-18 22-36-48 1695076608	F 2 00 00 10 00 00 00
trial_Ki		-	9792e_00002 20	9792e_00003 20	9792e_00004 20	9792e_00005 20	9792e_00006 20	9792e_00007 20	9792e_00008 20	9792e_00009 20	9792e_00010 20	9792e_00011 20	9792e_00012 20	9792e_00013 20	9792e_00014 20	9792e_00015 20	9792e_00016 20	9792e_00017 20	2c 00018 20	9792e_00019 20	9792e_00020 20	9792e_00021 20	9792e_00022 20	9792e_00023 20	9792e 00024 20	9792e_00025 20	9792e_00026 20	9792e_00027 20	9792e_00028 20	9792e_00029 20	72e_00030 20	0000=0000
traming iteration	2 979	3 979	3 97	3 97	3 979	3 97	3 97	4 979	3 979	3 979	8 979	3 97	12 979	12 979	4 979	3 979	3 97	3 979	3 97	2 979	3 979	1 979	1 979	2 979	9 97	3 979	3 979	2 97	1 975	3 97	1 979	0.00
GOING LEAIN	FALSE	TRUE	TRUE	TRUE	TRUE	PALSE	PALSE	PALSE	SALSE	PALSE	PALSE	TRUE	PALSE	PALSE	FALSE	TRUE	TRUE	TRUE	PALSE	SALSE	TRUE	PALSE	PALSE	PALSE	SALSE	TRUE	TRUE	PALSE	PALSE	TRUE	PALSE	CLO 1 V C
	3		TRUE	Dil.	TRUE	TRUE	TRUE	TRUE	-	TRUE F.	TRUE F	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE		TRUE F	TRUE	-	TRUE	TRUE	TRUE	TRUE F	TRUE	TRUE	TRUE	a anda
time time the smooth checkbomt	31.70634699	30.44120598	36.61903071	35.85677791	31.82855487	31.60537362	88.234869	43.17577934	248.6930933	36.4260869	33.02581811	35.1895082	37.49703932	38.99663877	33.16665983	34.12981153	31.29855347	31.21441579	33.72485995	30.15006375	27.93542719	31.8970747	30.716429	27.04557467	39.25414872	33.0221889	29.36053109	32.37938118	33.68402982	30.410218	31.29935384	31 41092502
v-npv	0.9577464789	0.9538461538	0.9154929577	0.8375		0.969696967	0.9846153846	0.9714285714	0.971014/928	0.9846153846	0.9565217391	0.9428571429	0.83672637 0.847826087 0.9705882353	0.9718309859			0.8205128205	0.8192771084	0.9558823529	0.7710843373	0.9552238806	0	0.5		0.971014/928	19999999906	0.858974359		0	0.8783783784	0.6666666667	00100100100
veppv	0.8651685393	0.8105263158	0.8314606742	0.8375	0.8271604938	0.829787234	0.8315789474	0.866666667	0.8571428571	0.8315789474	0.8461538462	0.844444444	0.847826087	0.8764044944	0.8494623656	0.8428571429	0.8048780488	0.8441558442	0.8369565217	0.7922077922	0.8279569892	0.5	0	0.8148148148	0.8571428571	0.8588235294	0.8414634146	0.8571428571	0.5	0.8255813953	0.5031847134	A 707 1 1 1 1 0 0 0 1
trapy.	0.802916375	0.8570644118	0.7694008983	0.7337447598	0.7321818275	0.819427553	0.809887419	0.8374596026	0.8190313193	0.82451761	0.8218795937	0.8126126149	0.83672637	0.8552554	0.8323806443	0.7292340862	0.7516447083	0.7323989648	0.814956025	0.7047307379	0.7769110304	0.3513150387 0.6482278871	0.3246398473 0.407147872	0.7187883578	0.8272448589	0.8111359977	0.7583557627	0.8016028418	0.3063225871 0.3020703012	0.7537321278	0.4051384	0.000110000
rr-ppv	.7405895953	.6673969446	.7354704115	.7165775601	.7136281371	.7159202576	.7543514639	.7420646155	.7558207145	.7034451928	.7707412841	.7410503215	.7726512299	.7734227055	.7591972514	.7039073971	7119016847	.719040981	.7467119975	6691668853	.7339860466	.3513150387	.3246398473	.6988263745	.7672356794	.7426384119	735469436	.7481008407	.3063225871	.7178062481	.3070737804	002022222
vdo.v	0.9114575091 6	0.8821862348 0	0.873476816 0	0.8375 0.7165775601 0.7337447598 0.8375	0.8313017659 0	0.8997421019 0	0.908097166 0	0.919047619 0	0.9140786749 0	0.908097166 0	0.9013377926 0	0.8936507937 0	0.9092071611 0	0.9241177401 0	0.9172684962 0	0.8047619048 0	0.8126954346 0	0.8317164763 0	0.8964194373 0	0.7816460648 0	0.8915904349 0	0.25	0.25	0.8187998125 0	0.9140786749 0	0.882745098 0	0.8502188868	0.9140786749 0	0.25	0.8519798869 0	0.58492569 0	21 0.6967650431 0.6710.409598 0.780196938 0.65675737353 0.6863419647 0.7874418605 0.9141635
Adulta Adulta Adulta Adulta Solita	0 - 0.466114445 - 0.2909484306 - 0.7717529852 - 0.9114575091 - 0.7405895953 - 0.802916375 - 0.8651685393 - 0.9577464789 - 31.70634699 - 0.771752985 - 0.9651685393 - 0.9651685393 - 0.9651685393 - 0.96516859 - 0.965168599 - 0.965168599 - 0.965168599 - 0.96516859 - 0.965168599 - 0.96516859 -	0.513154619 0.3732534647 0.7622306782 0.8821862348 0.6673969446 0.8570644118 0.8105263158	0.4985633866 0.3444455266 0.7524356549 0.873476816 0.7354704115 0.7694008983 0.8314606742 0.91549295777 0.8814606742 0.91549296777 0.91549296777 0.91549296777 0.91549296777 0.91549296777 0.91549296777 0.91549296777 0.91549296777 0.91549296777 0.91549296777 0.91549296777 0.91549296777 0.91549296777 0.91549296777 0.91549296777 0.91549296777 0.91549296777 0.91549296777 0.915492967 0.915492967 0.9154967 0.915492967 0.915492967 0.915492967 0.915492967 0.915492967 0.915492967 0.915492967 0.915492967 0.915492967 0.915492967 0.915492967 0.915492967 0.915492967 0.915492967 0.915492967 0.915492967 0.91549297 0.915492967 0.91549297 0.91549297 0.91549297 0.91549297 0.91549297	3 0.5468704064 0.4644676149 0.72516116	4 0.5539678954 0.4779375196 0.7229049823 0.813017659 0.7136281371 0.7321818275 0.8271604938 0.835443038	5 0.4718892515 0.2988026738 0.7676739053 0.8997421019 0.7159202576 0.819427553 0.829787234	0.4484381164 0.3031568527 0.7821194415 0.908097166 0.7545514639 0.809887419 0.8315789474 0.9846153846 0.98461544 0.98461544 0.98461544 0.9846153846 0.98461544 0.98461544 0.98461544 0.984615444 0.984615444 0.9846154444 0.9846154444 0.9846164444 0.98461644444 0.98461644444 0.9846164444444444444444444444444444444444	0.4427764911 0.23304829 0.789762109 0.919047619 0.7420646155 0.8374596026 0.866666667 0.9714285714 43.17577	8 0.4401612867 0.23218588 0.7874260169 0.9140786749 0.7558207145 0.8190313198 0.8571428571 0.9710144928 248.68099	9 0.4929809075 0.292770952 0.7639814014 0.908097166 0.7034451928 0.82451761 0.8315789474 0.9846153846	$10 \\ 10 \\ 10 \\ 10 \\ 10 \\ 10 \\ 10 \\ 10 \\$	11 0.4591784791 0.3054469526 0.7768314682 0.8036507937 0.7410503215 0.8126126149 0.844444444 0.9428571429 35.189508	12 0.4035279537 0.2474649847 0.8046887999 0.9092071611 0.7726512299	13 0.3894052265 0.2371092588 0.8143390528 0.9241177401 0.7734227055	14 0.4275277822 0.2892288268 0.7957889478 0.9172684962 0.7591972514 0.8323806443 0.8494623656 0.9850746269	15 0.5746652506 0.5396606326 0.7165707416 0.8047619048 0.7039073971 0.7292340802 0.8428571429 0.7066666667 0.8428671429 0.7066666667 0.8428671429 0.70666666667 0.8428671429 0.8428671	$16 \\ 16 \\ 10.5964619141 \\ 0.489496039 \\ 0.7317731965 \\ 0.8126954346 \\ 0.7119016847 \\ 0.7119016847 \\ 0.7516447083 \\ 0.8048780488 \\ 0.8205128205 \\ 0.8205128$	17 0.5455917346 0.4952264726 0.7257199765 0.8317164763 0.7190409881 0.7323089648 0.8441558442 0.8192771084	18 0.4486616516 0.2720233202 0.7808340112 0.8064194373 0.7467119975 0.814956025 0.8169565217 0.9558823529	1.9 0.6090345688 0.55422966437 0.6869488116 0.7816400648 0.6691668853 0.7047307379 0.7922077922 0.7710843373 30.150065	20 0.4914329208 0.3488953114 0.7554455385 0.8915904349 0.7339860466 0.7769110304 0.8279569892 0.9552238806	0.6903615014 0.6932321787 0.4997714629	22 0.8576620873 0.6960983872 0.3658938596	23 0.5704803611 0.5163700581 0.7088073661 0.8187998125 0.6988263745 0.7187883578 0.8148148148 0.8227848101	24 0.4209395811 0.2620297968 0.7972402692 0.9140786749 0.7672356794 0.8272448589 0.8571428571 0.9710144928	25 0.4571474606 0.3160237968 0.7768872048 0.82745098 0.7426384119 0.8111359977 0.8588235294 0.9066666667	26 0.5004895647 0.3977770077 0.7469125994 0.8502188868 0.735409436 0.7583557627 0.8414634146 0.858974359	27 0.4627689065 0.2906835079 0.7748518412 0.9140736749 0.7481008407 0.8016028418 0.8571428571 0.9710144928	0.69413203 0.3041964441	29 0.5328529063 0.4273747504 0.735769188 0.8519798869 0.7178062481 0.7537321278 0.8255813953 0.8783783784	30 0.6962281674 0.7011238337 0.3561060902 0.58492569 0.3070737804 0.4051384 0.5031847134 0.6666666667 31.29835	0.6710.007599
	484506	732534647	3444455266	4644676149	1,4779375196	3.2988026738	0.3031568527	0.23304829	0.23218593	0.292770952	0.2704665661	0.3054469526	0.2474649847	0.2371092588	0.2892288268	0.5396606326	0.489495039	0.4952264726	0.2720233202	0.5842966437	0.3488953114	0.6932321787	0.6960983872	0.5163700581	0.2620297968	0.3160237968	0.397770077	0.2906835079	0.69413203	0.4273747504	0.7011238337	0.576700141
v_loss	0.2909	0.3	ci.								-	-	Ľ	_	_	_	_	_	-	_	_	_	_	Ľ	Ĺ	Ĺ	\vdash	.4627689065 (_	_	ビ	۲

Table B.4: AlexNet Hyperparameter Tuning on Synthetic Dataset: best results per trial using BPV as metric.

config/wd	0.004061184702	0.001342279465	0.008916068564	0.0004985604437	0.007136375082	0.003269287767	0.004659795004	0.006356989462	0.00122228527	0.000648522356	0.003352476881	0.005863914571	0.003141911008	0.004888695606	0.002832183062	0.005584680213	0.007924700881	0.0002010142919	0.006799556385	0.00686528478	0.00165147235	0.008063947189	0.007654391635	0.004617847331	0.0054201634	0.009232470961	0.009995135962	0.003141916753	0.006609644365	0.003949123257	0.007955806194	0.008006468130
config/Ir	0.05559395377	0.02157092023	0.0002442402658	0.07744105697	0.06134134821	0.002766231285	0.0004385872335	0.01718461729	0.0008623078517	0.0004104941075	0.0001772317869	0.000262413112	0.01722764842	0.07723961246	0.002301789697	0.02796920797	0.0001706890204	0.03643194282	0.0013664823	0.0003064221439	0.0002993755697	0.01047111272	0.002935253947	0.0001218301799	0.003708952937	0.004011678439	0.08545917956	0.005450266819	0.0004455679841	0.0003882719365	0.02696476461	0.0191703018
config/fm2	64	128	64	1024	16384	1024	256	64	128	16384	1024	4096	128	512	1024	2048	1024	4096	1024	20:48	6-4	64	16384	256	1024	4096	32	1024	32	4096	1024	00.00
config/hn1 cc	80.08	16384	128	20:48	256	512	226	128	32	9600	8192	9600	128	16384	9600	1024	64	32	9600	256	1024	128	9600	8192	32	256	128	128	64	1024	226	10001
config/batch_size con	256	256	128	512	19	128	256	512	256	512	256	256	512	256	128	19	128	256	256	128	512	512	99	512	512	19	512	128	256	256	256	920
iterations since restore c	2	+	64	_		4	2	1		_	9		2		2	-	7	_	59	9	4	1	2	8			-	2	18			-
time_since_restore	116.064826	333.8755028	3327.2218	66.33523369	166.9644287	216.4648645	114.1887622	65.85715818	175.1872292	85.00689101	871.4799483	204.1417642	128.0351641	261.8102062	128.5145011	65.71394467	377.1746299	56.95391092	3838.212718	341.7987902	245.0734117	62.81446671	177.5650866	719.9921441	173.8016331	230.3586838	59.70610547	106.7182741	961.1425698	167.1029427	167.5149155	0220101000
node_ip ti	201.159.221.21	301.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	301.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201 120 001 01
hostname	12.122.02.102. ocho.do-0-1.cod	dgx-node-0-1.cedia.edu.ec 201.139.221.21	dgx-node-0-1.cedia.edu.ec	dgx-node-0-1.codin.edu.oc	dgx-node-0-1.cedia.edu.ec 201.159.221.21	1695061978 216.4648645 3054627 dgx-node-0-1.codin.edu.cc 201.159.221.21	2023-09-18_19-27-33 1695065253 114.1887622 3054627 dgx-node-0-1.cedla.edu.ec 201.159.221.21	2023-09-18_20-24-53 1695068693 65.85715818 3054627 dgx-node-0-1.cedia.edu.ec 201.159.221.21	dgx-node-0-1.cedia.edu.ec 201.159.221.21	dgx-node-0-1.codia.odu.oc 201.159.221.21	3054627 dgx-node-0-1.cedin.edu.ec	dgx-node-0-1.cedia.edu.ec 201.139.221.21	128.0351641 3054627 dgx-node-0-1.cedin.edu.ec 201.159.221.21	a45f3_00013 2023-09-18_22-30-35 1695076235 261.8102062 3054627 dgx-node-0-1.ceda.edu.ec 201.159_221.21	a45f3_00014 2023-09-18_222-32-44 1695076364 128.5145011 3054627 dgx-node-0-1.cedis.edu.cc 201.159_221_21	3054627 dgx-node-0-1.codin.cdu.oc 201.159.221.21	dgx-node-0-1.codia.edu.oc 201.159.221.21	dgx-node-0-1.codia.edu.oc	dgx-node-0-1.cedia.edu.ec 201.159.221.21	2023-09-19_00-05-42 1695-081942 341.7987902 305-4627 dgx-node-0-1.cedia.edu.cc 201.159.221.21	a453.00020 2023-09-19_01-02-42 1695085362 245.0734117 3054627 dgx-node-0-1.ceda.edu.ec 201.159.221.21	2023-09-19_01-11-39 1695085899 62.81446671 3054627 dgx-node-0-1.cedis.edu.cc 201.159.221.21	3054627 dgx-node-0-1.cedin.edu.ec 201.159.221.21	dgx-node-0-1.codia.edu.oc	dgx-node-0-1.codin.edu.oc	2023-09-19-02-00-32 169508832 230,3586838 3054627 dgx-node-0-1.cedts.edu.ec 201,159,221,21	2023-09-19_02-01-32 1695088892 59.706105-47 305-4627 dgx-node-0-1.cedia.edu.ec 201.159.221.21	2023-09-19_02-05-13 1695089113 106,7182741 3054627 dgx-node-0-1.cedis.edu.ec 201.159.221.21	1695090128 961.1425698 3054627 dgx-node-0-1.codia.edu.cc 201.159.221.21	dgx-node-0-1.codia.edu.oc	dgx-node-0-1.codia.odu.oc	described 0.1 codes color on 201 150 001 01
bid	3054627	3054627	3054627	3054627	3054627	3054627	3054627	3054627	3054627	3054627		3054627	3054627	3054627	3054627		3054627	3054627	3054627	3054627	3054627	3054627	3054627	3054627	3054627	3054627	3054627	3054627	3054627	3054627	3054627	m (000 4000
mp time_total_s	116.064826	911 333.8755028	.405 3327.2218	.472 66.33523369	.762 166.9614287	.978 216.4648645	253 114.1887622	1693 65.85715818	1181 175.1872292 305.4627	1695069780 85.00689101	1695071264 871.4799483	1695072819 204.1417642 3054627	917 128.0351641	1235 261.8102062	128.5145011	1695076492 65.71394467	985 377.1746299	306 56.95391092	1695081254 3838.212718 3054627	.942 341.7987902	362 245.0734117	1899 62.81446671	1199 177.5650866	1996 719.9921441	173.8016331	1832 230.3586838	1892 59.70610547	1113 106.7182741	1128 961.1425698	1612 167.1029427	.981 167.5149155	100000000 00 01000000
timestamp	16-17-56 1695053	17-25-11 1695057911	18-23-25 1695061405	18-24-32 1695061472	18-29-22 1695061762		19-27-33 1695065	20-24-53 1695068	20-33-01 1695069181	20-43-00 1695069			2023-09-18_22-25-17 1695075917	22-30-35 1695076	22-32-44 1695076		22-43-05 1695076985	22-48-26 1695077306		00-05-42 1695081	01-02-42 1695085	01-11-39 1692082	01-16-39 1695086199	01-29-56 1695086996	01-56-42 1695088002	02-00-32 1695088	02-01-32 1695088	02-05-13 1695089		02-30-12 1695090612	02-53-01 1695091981	
d date	2023-09-18	0001 2023-09-18_17-25-11	0002 2023-09-18 18-23-25	000 2023-09-18,18-24-32	0001 2023-09-18 18-29-22	000 2023-09-18_18-32-58	2023-09-18		0.008 2023-09-18,20-33-01	0009 2023-09-18 20-43-00	010 2023-09-18_21-07-44	011 2023-09-18 21-33-39	2023-09-18	2023-09-18	0.14 2023-09-18_	015 2023-09-18,22-34-52	0016 2023-09-18 22-43-05	017 2023-09-18,22-48-26	0.18 2023-09-18-23-54-14	0.19 2023-09-19	020 2023-09-19		022 2023-09-19-01-16-39	023 2023-09-19 01-29-56	024 2023-09-19-01-56-42		0.26 2023-09-19		028 2023-09-19-02-22-08	029 2023-09-19,02-30-12	0030 2023-09-19 02-53-01	NOT 0000 00 10 00 54 99
eration trialid	a45f3_00000	a45f3_00001	a45f3_00002	a45f3_00003	a45f3_00004	a45f3_00005	a45f3_00006	a45f3_00007	a45f3_00008	a45f3_00009	a45f3_00010	a45f3_00011	a45f3_00012	a45f3_00	945f3_0C	a45f3_00015	a45f3_00016	a45f3_00017	a45f3_00018	a45f3_00019	a45f3_00	a45f3_00021	a45f3_00022	a45f3_00023	a45f3_00024	a45f3_00025	a45f3_00026	a45f3_00027	a45f3_00028	a45f3_00029	a45f3_00030	1000000
training iteration	2	4	64		3	4	2	-	00	_	9	3	5	3	2	1	1		28	9	4	1	2	8	m	3		2	18	8		
nt done	EALSE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	EALSE	FALSE	EALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	100
s should_checkpoint	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	COLUMN TO SERVICE
time_this_iter_s	55.27797389	81.39263487	3 53.24762988	66.33523369	54.88680696	53.80975485	4 55,0990088	65.85715818	55.75530767	3 85.00689101	7 139.6667998	62,32388568	0.2277227723 57.38856196	96.16220379	5 62.21274686	3 65.71394467	4 53.98469543	56.95591092	7 65.56853223	5 52.53494477	6 57.91662455	62.81446671	7 86.1515882	7 7433621502	4 54.49460483	60.24510169	59.70610547	3 51.62087631	3 52.52617359	4 54.94650745	55.1375618	000 010 0000
v-npv	0.19375	0.2	0.53333333	0.2	0.2	0.5	0.571428571	0.225	0.5	0.58333333	0.434782608	0.4		0.2	0.352941176	0.213333333	0.615384615	0.2	0.285714285	0.538461538	0.368421052	0.1875	0.307692307	0.466666666	0.194444444	0.2	0.2	0.33333333	0.58333333	0.636363636	0.2	0.0000000
v-ppv	9'0	0	0.9356895132 0.833333333 0.5333333333	0	1 0.8	0.6564352345 0.8890869987 0.8036809816	1 0.8164556962	4 0.808	0.7250696755 0.9332364255 0.8152866242	3 0.8300653595	0.8222906932 0.9281933943 0.838028169 0.4347826087	0.82	0.84375	3	0.8175675676	0.7097071126 0.9333333333 0.2133333333	0.9400854399 0.8355263158 0.6153846154 53.98469543	0	0.7485999735 0.9000348353 0.8037974684 0.2857142857	5 0.8289473684	6 0.8219178082	9 0.7986577181	0.6009140745 0.8172433915 0.8092105263 0.3076923077	0.9341738206 0.826666667 0.466666667	0.7252608638 0.7984496124 0.194444444	0 8	0 5	2 0.8050314465	0.9325860055 0.8300653595 0.583333333	0.9185058642 0.8311688312 0.6363636364	0	0.0000000000000000000000000000000000000
tr_npv	0.01040372671 0.710644606	0.008361779854 0.7107351365	0.935689513;	0.703477082	0.7100395171	0.889086998;	0.910832722,	0.607632476 0.7793001844 0.808	0.9332364252	0.8685671172	0.928193394.	0.9203387756	0.5566276361 0.8394577556	0.7106816272	0.874110108	0.7097071120	0.9400854395	0.09535054448 0.6973385104	0.900034835.	0.919949882t	0.926341654k	0.791048956t	0.817243391:	0.934173820t		0.7192405776	0.1679514177 0.7094892385	0.9006659361	0.9325860052	0.918505864;	0.7146580647	0.1105164046 0.4070644101 0.4046915700 0.404947101
tr_ppv	0.01040372671	0.008361779854		0.07941682822	0.2487227353		0.2750825463 0.4526169896 0.8463941495 0.6839421338 0.781955769 0.9108327221 0.8164556962 0.5714285714 55.0990088 0.8868888 0.88688888 0.88688888 0.886888888 0.8868888888 0.8868888888 0.8868888888 0.88688888888 0.88688888888 0.886888888888 0.886888888888 0.88688888888 0.8868888888 0.8868888888 0.8868888888 0.8868888888 0.88688888888 0.8868888888 0.8868888888 0.886888888 0.886888888 0.8868888888 0.8868888888 0.8868888888 0.8868888888 0.886888888 0.8868888888 0.8868888888 0.886888888 0.8868888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.88688888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.8868888888 0.8868888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.88688888888 0.886888888 0.886888888 0.88688888 0.886888888 0.88688888 0.886888888 0.886888888 0.886888888 0.886888888888 0.886888888 0.886888888 0.88688888 0.88688888 0.88688888 0.886888888 0.886888888 0.886888888 0.88688888 0.88688888 0.886888888 0.886888888 0.886888888 0.8868888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.88688888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.88688888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888 0.886888888888 0.8868888888 0.8868888888 0.886888888 0.886888888 0.8868888888 0.8868888888 0.8868888888 0.8868888888 0.88688888888 0.886888888888 0.88688888888 0.88688888888 0.88688888888 0.8868888888888 0.886888888888888 0.886888888888888888888 0.8868888888888888888888888888888 0.8868888888888888888888888888888888888			0.4190690291 0.4645198882 0.7796060917 0.7066903464 0.6906450662 0.885671172 0.8300653505 0.5833333333		0.8068143352		0	0.5358481006 0.5183777809 0.7994145773 0.582545372 0.7247190467 0.874110108 0.8175675675676 0.552411765 62.21274686 0.817677676 0.8176776 0.8176776 0.8176776 0.8176776 0.8176776 0.81776776 0.817767777 0.81776777 0.8177777 0.8177777 0.8177777 0.8177777 0.8177777 0.8177777 0.8177777 0.8177777 0.8177777 0.8177777 0.8177777 0.8177777 0.8177777 0.817777 0	0.2673250536	0.7254554656 0.8051340335	0.09535054448		0.2523023475 0.4317631125 0.8570457363 0.6837044534 0.7941415901 0.9199498826 0.828473684 0.5384615385 52.5394477	0.233157798 0.4315789044 0.8706130902 0.5951694304 0.8148845259 0.9263416546 0.8219178082 0.3684210226 57.91662455	0.6738858236 0.5670912266 0.7035653053 0.4930788591 0.616082054 0.7010489566 0.708677181 0.1875	0.6009140745	0.826261467	0.1746174087	0.1816533964	0.1679514177	0.4335109434 0.6729678512 0.7363147159 0.5601823899 0.551963496 0.9006659362 0.8050314465 0.33333333333 5.162087631 0.66669362 0.66669292 0.66669362 0.6666960000000000000000000000000000000	0.7824017866	0.8010377095	0.1151341525	
v-bpv	0.396875	0.1	0.6833333333	0.1	0.5	0.6518404908	0.6939421338	0.5165	0.6576433121	0.7066993464	0.6364053889	0.61	0.5357363861	0.1	0.585254372	0.573333333	0.7254554656	0.1	0.544755877	0.6837044534	0.5951694304	0.4930788591	0.558451417	0.646666667	0.4964470284	0.1	0.1	0.5691823899	0.7066993464	0.7337662338	0.1	4000000000
tr.bpv	0.6080434915 0.957763638 0.3605241664	0.3595484582	0.4243549109 0.8750851216 0.6833333333	0.3914469551	1.112894654 0.4793811262	0.3646250922 0.4970176518 0.7727611166 0.6518404908	6 0.8463941495	0.6964398681 0.6066454649 0.6934663302 0.5165	0.2785683225 0.4443999231 0.8291530505 0.6576433121	2 0.7796060917	0.229542966 0.4339521527 0.8752420438 0.6364053889	0.2533299879 0.4562711716 0.8635765554	0.4529211644 0.6788284779 0.6980426958 0.5357363861	0.6020306488 1.012672424 0.3553408136	9 0.7994145773	7 0.4885160831	1 0.8726097367	0.9913259149 0.3963445274	0.3136558734 0.4631227851 0.8243174044 0.544755877	5 0.8570457363	4 0.8706130902	6 0.7035655053	0.6149633385 0.5381324291 0.709078733	0.2228845478 0.4498341978 0.8802176438 0.646666667	0.6903185844 0.4499391363 0.4964470284	0.5368707514 0.835752368 0.450446987	26 10.16420514 1.02954793 0.4387203281	2 0.7263147159	28 0.2476415726 0.4367242455 0.8574938961 0.7066993464	0.4189970493 0.8597717868 0.7337662338	0.9026520848 0.4148961086	59 00795097 0 920200000 0 4001004599
V_1088	0.955763638	1.02221489	0.424354910.	1.06342423	1.112894654	0.4970176518	0.452616989	0.606645464	0.444399923	0.464519888.	0.433952152.	0.4562711716	0.678828477	1.012672424	0.518377780	1.287734747	0.457841575.	0.991325914	0.463122785.	0.431763112	0.431578904	0.567091226	0.538132429	0.449834197.	0.690318584	0.835752368	1.02954793	0.672967851	0.436724245	0.418997049.	0.902652084	ACT OF THE PARTY O
tr_loss	6080434915	0.6098159354		1916.377738	17.87587802	3646250922	2750825163	18984398681	2785683225	.4190690291	12299542966	.2533299879	.4529211644	6020306488	3584481003	8.07808174	0.2319388805 0.4578415751	3.602631196	3136558734	2525023475	235155798	6738858236	6149633385	.2228845478	0.448867021	5368707514	10.16420514	.4335109434	2476415726	0.2531253421	0.6653174366	2000000000

Table B.5: VGG19 Hyperparameter Tuning on Original Dataset: best results per trial using validation loss as metric.

config/wd	0.004061184702	0.001342279465	0.008916068564	0.0004985604437	0.007156375082	0.003269287767	0.004659795004	0.006356989462	0.00122228527	0.000648522356	0.003352476881	0.005863944571	0.003141911008	0.004888695606	0.002832183062	0.005584680213	0.007924700881	0.0002010142919	0.006799556385	0.00686528478	0.00165147235	0.008063947189	0.007654391635	0.004617847331	0.0054201634	0.009232470961	0.009995135962	0.003141916753	0.006609644365	0.003949123257	0.007955806194	0.008996468139
config/lr	0.05559395377	0.02157092023	0.0002442402658	0.07744105697	0.06134134821	0.002766231285	0.0004385872335	0.01718461729	0.0008623078517	0.0004104941075	0.0001772317869	0.000262413112	0.01722764842	0.07723961246	0.002301789697	0.02796920797	0.0001706890204	0.03643194282	0.0013661823	0.0003064221439	0.0002993755697	0.01047111272	0.002935253947	0.0001218301799	0.003708952937	0.004011678439	0.08545917956	0.005450266819	0.0004455679841	0.0003882719365	0.02696476461	0.0121793018
config/hn2	64	128	64	1024	16384	1024	226	64	128	16384	1024	9601	128	512	1024	2048	1024	4096	1024	20:48	64	64	16384	256	1024	4096	32	1024	32	4096	1024	20.48
config/hn1	89:00	16384	128	2048	256	512	256	128	32	960#	8192	9600	128	16384	9600	1024	64	32	9600	256	1024	128	4096	2618	32	256	128	128	64	1024	256	16384
config/batch_size	256	256	128	512	19	128	256	512	256	512	256	256	512	256	128	64	128	256	256	128	512	512	64	512	512	19	512	128	256	256	256	256
time since restore iterations since restore	44	_	1	_		18	62	2	-	2	17	-		1	2	1	7	_	62		-	2	2	22	2	1	00	2	13		1	
ime since restore	2597.11408	89.21835852	55.54585266	66.33523369	166.9614287	948.9222898	3375,99932	315.9981761	405.5690389	190.8954971	1721.124011	78.79822731	184,5658765	84.6331315	128.5145011	65.71394467	377.1746299	56.95591092	4057,732144	289.2638454	66.58783937	123.218437	177.5650866	1949.738245	119.3070283	73.64326215	173.3535583	106.7182741	696.345192	167.1029427	57.89597607	90.31043553
node.ip	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	301.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21
hostname	dgx-node-0-1.cedia.edu.oc	dgx-node-0-1.cedia.edu.ec	a45f3_00002 2023-09-18_17-28-54 1695058134 55.54585266 3054627 dgx-node-0-1.cedis.edu.ec 201.159.221.21	3054627 dgx-node-0-1.cedin.edu.ec 201.159.221.	dgx-node-0-1.cedia.edu.oc	dgx-node-0-1.codin.edu.oc	dgx-node-0-1.cedia.edu.ec	3054627 dgx-node-0-1.codia.edu.oc 201.159.221.21	3054627 dgx-node-0-1.cedia.edu.ec 201.159.221.21	dgx-node-0-1.codia.odu.oc	dgx-node-0-1.codin.edu.oc	dgx-node-0-1.cedia.edu.ec	3054627 dgx-node-0-1.cedin.edu.ec	3054627 dgx-node-0-1.cedia.edu.ec 201.159.221.21	3054627 dgx-node-0-1.codis.edu.oc 201.159.221.21	3054627 dgx-node-0-1.codin.odu.oc	dgx-node-0-1.codia.odu.oc	dgx-node-0-1.cedin.edu.ec	3054627 dgx-node-0-1.cedia.edu.ec 201.159.221.21	3054627 dgx-node-0-1.cedin.edu.ec 201.159.221.21	dgx-node-0-1.cedia.edu.ec	dgx-node-0-1.codia.edu.ec	dgx-node-0-1.cedin.edu.ec	3054627 dgx-node-0-1.codia.edu.oc	a45f3_00024 2023-09-19_01-55-47 1695088547 119.3070283 3054027 dgx-node-0-1.ceda.edu.cc 201.159.221.21	3054627 dgx-node-0-1.cedia.edu.ec 201.159.221.21	3054627 dgx-node-0-1.cedin.edu.ec	dgx-node-0-1.cedia.edu.ec	dgx-node-0-1.codia.odu.oc	3054627 dgx-node-0-1.codin.edu.oc	a45f3 00030 2023-09-19 02-51-12 1695091872 57.89597607 3054627 dgx-node-0-1.cedia.edu.zc 201.159.22121	3054627 dgx-node-0-1.codin.edu.ec 201.159.221.21
pid	3054627	3054627	3054627 d		3054627	3054627	3054627			3054627	3054627 d	3054627 d		3054627 d			3054627	3054627		3054627 d	3054627	3054627	3054627	3054627 d	3054627 d	3054627 d		3054627 d	3054627		3054627 d	
time_total_s	2597.11408	89.21835852	25.54585266	66.33523369	166.9614287	1695062711 948.9222898	3375,99932	315.9981761	405,5690389	1695069886 190.8954971	1721.124011	78.79822731	1695075973 184.5658765	84.6331315	128.5145011	1695076492 65.71394467	377.1746299	56.95591092	4057.732144	289.2638454	1695085183 66.58783937	123.218437	177.5650866	1695088225 1949.738245	119.3070283	73.64326215	173.3535583	106.7182741	696.345192	167.1029427	57.89597607	90.31043553
timestamp	1695056357	1695057666	1695058134	1695061472	1695061762		1695068515	1695068943	1695069411		1695072113	1695072694		1695076058	1695076364		1695076985	1695077306	1695081473	1695081889		1695085960	1695086199		1695088547	9298802691	1695089006	1695089113	1695089863	1695090612	1695091872	1695092072
date	2023-09-18_16-59-17	2023-09-18_17-21-06	123-09-18 17-28-54	2023-09-18_18-24-32 1695061472 66.33523369	2023-09-18 18-29-22	2023-09-18_18-45-11	2023-09-18_20-21-55	2023-09-18_20-29-03 1695068943	2023-09-18_20-36-51 1695069411 405.5690389	2023-09-18_20-44-46	2023-09-18_21-21-53	2023-09-18_21-31-34	2023-09-18_22-26-13	2023-09-18_222-27-38 1695076058 84.6331315	2023-09-18_22-32-44 1695076364 128.5145011	2023-09-18_22-34-52	2023-09-18_22-43-05	2023-09-18_22-48-26	2023-09-18_23-57-53	2023-09-19_00-04-49 1695081889 289.2638454	2023-09-19,00-59-43	2023-09-19 01-12-40	2023-09-19_01-16-39	2023-09-19 01-20-25	23-09-19_01-55-47	2023-09-19-01-57-56 1695088676 73.64326215	2023-09-19_02-03-26	2023-09-19_02-05-13	2023-09-19_02-17-43	2023-09-19_02-30-12 1695090612 167.1029427	23-09-19 02-51-12	2023-09-19_02-54-32 1695092072 90.31043553
trialid	a45f3_00000 20	a45f3_00001 20	a45f3_00002 20	a45f3_00003 20	at5f3_00001 20	a45f3_00005 20	a45f3_00006 20	a45f3_00007 20	at5f3_00008 20	a45f3_00009 20	a45f3_00010 20	a45f3_00011 20	a45f3_00012 20	a45f3_00013 20	a45f3_00014 20	a45f3_00015 20	a45f3_00016 20	a45f3_00017 20	at5f3_00018 20	a45f3_00019 20	a45f3_00020 20	a45f3_00021 20	a45f3_00022 20	a45f3_00023 20	a45f3_00024 20	at5f3_00025 20	a45f3_00026 20	a45f3_00027 20	a45f3_00028 20	a45f3,00029 20	a45f3_00030 20	a45f3_00031 20
training iteration	44	-	1	-	3	18	62	.0	t	2	17	-		-	2	-	7	-	62	10	-	2	2	22	5	1		2	13	8	-	-
done	EALSE	FALSE	EALSE	EALSE	TRUE	EALSE	FALSE	EALSE	FALSE	EALSE	EALSE	FALSE	TRUE	FALSE	EALSE	EALSE	EALSE	EALSE	FALSE	EALSE	FALSE	EALSE	EALSE	EALSE	EALSE	FALSE	TRUE	FALSE	EALSE	FALSE	EALSE	FALSE
should_checkpoint	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
	98883901.09	89.21835852	22.54585266	66.33523369	54.88680696	52.11209178	56.16844606	59.67531204	58.40756512	105.8886061	76.75145006	78.79822731	56.53071237	84.6331315	62.21274686	65.71394467	53.98469543	56.95591092	64.63184619	52.94590569	66.58783937	60.40397024	86.1515882	80.86843491	56.94109631	73.64326215	57.43443751	51.62087631	54.89529061	54.94650745	57.89597607	90.31043553
v-npv	0.2012195122	0.2	1	0.2	0.2	-	0.625	0.2080536913	-	9'0	0.5	9.0	0.2692307692	0.2	0.3529411765	0.2133333333	0.6153846154	0.2	1	0.6666666667	-	0.2033023256	0.3076923077	0.487804878	0.2089552239	0.2	0.2024539877	0.8050314465 0.3333333333	0.0666666667	0.6363636364	0.2	0.202247191
v-ppv	1	0	0.8098159509	0	8.0	0.8148148148	0.8723404255	0.875	0.8098159509	0.8258064516	0.8510638298	0.8125	0.8129496403	0	0.8175675676	0.933333333	0.8355263158	0	0.80-48780-488	0.8176100629	0.8098159509	0.8101265823	0.8092105263	0.8951612903	0.806122449	0	-	0.8050314465	0.8176100629	0.8311688312	0	0.8026315789
tr_npv	0.7107169046	0.7094942763	0.8715684044	0.703477082	0.7100395171	0.9051338028 0.8148148148	0.9438112548 (0.771267908	0.9406031496	0.9180235179	0.9468387857	0.8841866369	0.7861650646 0.8129496403	0.7106499581	0.874110108	0.7097071126 0.933333333 0.2133333333	0.9400854399 0.8355263158 0.6153846154	0.6973385104	0.9010681742	0.9166799237	0.8714086461	0.8029992204	0.8172433915 0.8092105263 0.3076923077	0.9474730654	0.7124382331	0.7442907145	0.7105476383	0.9006659362	0.9302885861 0.8176100629 0.666666667	0.9185058642	0.7117434496	0.6978644121
tr_ppv	0.01086956522	0.02200111728	0.3557451782 0.4992086619 0.7991834981 0.3049079755 0.7267985918 0.8715684044 0.8098159509	0.07941682822 0.703477082	0.2487227353	0.6249445025	0.8018489787	0.645482389	0.2508392255 0.4603428245 0.8417791334 0.9049079755 0.7429551171 0.9406031496 0.8998159599	0.2538163999 0.4807957113 0.8583577341 0.712903228 0.7986919503 0.9180235179 0.8258064516	0.8620081246 0.9468387857	0.7349131738	0.479297028	0.004926162183 0.7106499581	0.3584481003 0.5183777809 0.7994145773 0.585254372 0.7247190467 0.874110108 0.8175675676 0.3529411765	0.2673250536	0.8051340335	0.09535054448	0.314313014 0.4915015101 0.823158116 0.9024390244 0.7452480577 0.9010681742 0.8048780488	0.2592527117 0.5279269218 0.8553217349 0.7421383648 0.7939635462 0.9166799237 0.8176100629 0.0666666667	0.806522052 0.9049079755 0.7416354579 0.8714086461 0.8098159509	0.7002556152 0.8029992204 0.8101265823 0.2093023256	0.6009140745	0.905676813 0.6914830842 0.8638805607 0.9474730654 0.8951612903 0.487804878	0.5164594845 0.6967394948 0.3751181906 0.5075388364 0.03779814801 0.7124382331 0.806122449 0.2089552239	0.2848443968 0.7442907145	0.116168437	0.5519634956	0.7735777448	0.2531253421 0.4189970459 0.8597717868 0.7337662338 0.8010377095 0.9185058642 0.8311688312 0.6563636364	0.1535053001 0.7117434496	0.1185164946 0.6978644121 0.8026315789 0.202247191
v-bpv	1,6006097561	0.1	1.9049079755	0.1	0.5		0.7486702128	1.5415268456	1.9049079755	1.7129032258		0.70625	1.5410902048	0.1	0.585254372	0.573333333	1.7254554656	0.1	1.9024390244	1.7421383648	1.9049079755	1.5097144539	0.558451417	1.6914830842	1.5075388364	0.1	1.6012269939	0.5691823899		1.7337662338	0.1	0.502439385
tr.bpv	1922609009'0 65:82862098'0	0.3657476968	0.7991834981	0.3914469551	0.4793811262	0.7650391526 0.9074074074	0.8728301168	0.4773871157 0.8774345517 0.7083751485 0.5415268456 0.645482389	0.8417791334	0.8583577341	0.9044234551 0.6755319149	0.8095499054	0.6952222586 0.6327310463 0.5410902048	1.057708025 0.3577880602	0.7994145773	0.4885160831	0.8726097367 0.7254554656	0.3963445274	0.823158116	0.8553217349		0.7516274178 0.5097144539	0.709078733		0.3751181906	0.5145675557	0.4133580376 0.6012269939	0.7263147159	0.8519331655 0.7421383648	0.8597717868	3.498751289 0.9807639122 0.4326243749	53.90725937 0.8707969785 0.4081904533 0.502439385
	0.9724339247	1.054414034	0.4992986619	1.06342423	1.112894654	0.552541256	0.5136245489	0.8774345517	0.4603428245	0.4807957113	0.5580316186	0.509873569	0.6952222586	1.057708025	0.5183777809	1.287734747	0.4578415751	0.9913259149	0.4915015101	0.5279269218	13483600208 0.467004776	0.6600763202	0.5381324291		0.6967394948	1.037828803	1.296965837	0.6729678512	0.4512567222	0.4189970493	0.9807639122	0.8707969785
tr_loss	0.6718499269	49.47073078	13557451782	1916.377738	17.87587802	0.3859260124	0.2100537904 (0.4773871157	0.2508395255	0.2538163999	0.1796877428 (0.3300239287	0.4488471957	2980.780831	0.3584481003	8.07808174	0.2319388805 (3.602631196	0.314313014	0.2592527117	0.3483600208	0.4225211348 (0.6149633385 (0.1788608463 0.4718012512	0.5164594845	0.5111475421	0.6443328786	0.4335109434 (0.2558941499 (0.2531253421	3.498751289	53.90725937

Table B.6: VGG19 Hyperparameter Tuning on Original Dataset: best results per trial using BPV as metric.

Γ	28	98	8	11	22	28	11	E	=	2	22	22	16	20	75	25	46	92	24	98	æ	=	81	23	54	23	8	80	28	12	12	82
pw/ginoo	0.004057259988	0.006160013460	0.00226787872	0.0066805428	0.00702227385	0.00111536956	0.0068365434	0.00428514341	0.006174156423	0.00380614338	0.00105566953	0.00137215483	0.0079366845	0.00836310120	0.00924748512	0.00657477398	0.0002543811846	0.0070694937.	0.00453451537	0.00246174074	0.005737552548	0.0032893309	0.00140273628	0.007304940458	0.000533930419	0.007143531658	0.005814579208	0.000407346420	0.009173115098	0.00972859251	0.0081815941	0.00609342998
config/lr	0.02291154008	0.003665737195	0.000144815453	0.01341145855	0.00272243137	0.0001348969613	0.0008859104983	0.07024536547	0.0004105288637	0.06361179329	0.03800694642	0.0002090720054	0.008028446355	0.02852980691	0.07405577337	0.004511012484	0.006142432526	0.0001969590185	0.0003367869903	0.0003984424533	0.07812642972	0.0001622058465	0.04100482089	0.01449432222	1.0003258297361	0.0001783647433	0.0931531612	0.03057269777	0.03946097237	0.0005373097407	0.002735066653	0.0001953925933
config/hn2	8192	2048	256	128	64	8192	1024	960#	256	128	16384	8192	32	1024	512	32	128	16384 (9600	512	2048	32	1024	256	128	8192	32	8192	8192	64	1024	256
	1024	128	16384	16384	8192	2048	4096	64	256	256	4096	512	4096	6-4	8192	1024	6-4	4096	1024	256	8192	4096	16384	256	1024	16384	1024	32	8192	1024	16384	1024
config/batch_size cor	512	256	64	64	128	128	512	128	256	256	64	64	64	128	256	256	512	64	256	512	512	128	512	512	128	128	512	256	128	128	256	256
time_since_restore iterations_since_restore config/batch_size config/lm1	11	+	1-		9	_	62	2	63		_	2	-			7	9	16	12	2	_		2			29	-		2	9	1	2
time since restore	798.3483758	278.6578188	816.1528029	343.2146995	492.5904169	71.23765755	4894.090337	130.1416068	4064.266164	192.208555	89.63877296	134.6616607	80.72840214	193,6992581	246.6340413	273.96945	400.697089	1380,158476	827.1856215	143,3706887	89.30738521	229.3269925	203.3732076	199.1109407	193.4771852	6675.402836	71.35978866	197.6078634	178.5386949	406.0547113	104.8331921	131.9834747
node,ip	159.221.21	159.221.21	201.159.221.21	159.221.21	159.221.21	159.221.21	159.221.21	159.221.21	159.221.21	159.221.21	159.221.21	159.221.21	159.221.21	129.221.21	159.221.21	159.221.21	129.221.21	159.221.21	159.221.21	129.221.21	159.221.21	159.221.21	129.221.21	159.221.21	159.221.21	159.221.21	159.221.21	159.221.21	159.221.21	159.221.21	159.221.21	129.221.21
-	n.ec 201.	n.ec 201.	u.ec 201.	n.ec 201.	n.ec 201.	n.ec 201.	n.ec 201.	n.ec 201.	u.ec 201.	и.ес 201.	n.ec 201.	u.ec 201.	и.ес 201.	n.ec 201.	u.ec 201.	n.ec 201.	n.ec 201.	n.ec 201.	n.ec 201.	n.ec 201.	n.ec 201.	n.ec 201.	n.ec 201.	n.ec 201.	u.ec 201.	n.ec 201.	n.ec 201.	n.ec 201.	n.ec 201.	u.ec 201.	u.ec 201.	и.ес 201.
hostname	0-1 cedia ed	476939 dgx-node-0-1.cedia.edu.ec 201.159.221.21	476939 dgx-node-0-1.cedia.edu.ec	476939 dgx-node-0-1.cedia.edu.ec 201.159.221.21	0-1 cedia ed	476939 dgx-node-0-1.cedia.edu.ec 201.159.221.21	dgx-node-0-1 codia odu.ec 201.159.221.21	476939 dgx-node-0-1.codia.odu.ec 201.159.221.21	0-1 codia od	476939 dgx-node-0-1.cedia.edu.ec 201.159.221.21	476939 dgx-node-0-1.cedia.edu.ec 201.159.221.21	476939 dgx-node-0-1 cedia.edu.ec 201.159.221.21	dgx-node-0-1.codia.odu.ec 201.159.221.21	0-1 cedia ed	0-1 codia od	476939 dgx-node-0-1.codia.odu.ec 201.159.221.21	476939 dgx-node-0-1.cedia.edu.ec 201.159.221.21	476939 dgx-node-0-1.codia.odu.oc 201.159.221.21	0-1 cedia ed	0-1 cedia ed	476939 dgx-node-0-1 cedia edu.ec 201.159.221.21	476939 dgx-node-0-1 codia.odu.ec 201.159.221.21	476939 dgx-node-0-1.cedia.edu.ec 201.159.221.21	0-1 codia od	0-1 cedia ed	0-1 cedia ed	476939 dgx-node-0-1 codia.odu.ec 201.159.221.21	476939 dgx-node-0-1.cedia.edu.ec 201.159.221.21	476939 dgx-node-0-1.cedia.edu.ec 201.159.221.21	0-1 codia od	476939 dgx-node-0-1 codia.odu.ec 201.159.221.21	476939 dgx-node-0-1.cedia.edu.ec 201.159.221.21
À	dgx-node-	-opou-xgp	-opou-x8p	dgx-node-	-opou-xgp	-opou-x8p	dgx-node-	-opou-xgp	-opou-x8p	dgx-node-	-opou-xgp	-opou-x8p	dgx-node-	-apou-xgp	-opou-xgp	-opou-x8p	dgx-node-	-opou-xgp	-opou-x8p	dgx-node-	-opou-xgp	-opou-x8p	dgx-node-	-opou-xgp	-opou-xgp	dgx-node-	-opou-xdp	-opou-x8p	dgx-node-	-opou-xgp	-opou-xgp	-epou-x8p
bid	8 476939			5 476939	9 476939		476939		476939	476939	8 476939		4.76939	1 476939	3 476939	-		8 476939	5 476939	7 476939			6 476939	476939	2 476939	6 476939	8 476939		9 476939	3 476939		476939
time_total_	798.348375	278.657818	816.152802	343,214699	492,590410	71.2376575	1891,09033	130,141606	1064.26616	192,208557	89.6387729	134.661660	80.72840214	193,699258	246.634041	273.96945	400.697089	1380.15847	827.185621	143,370688	89.3073852	229.326992	203.373207	199.110940	193.477185	8675.40283	71.3597886	197.607863	178.538694	406.054711	104.833192	131.983474
timestamp time_total_s	1695093049 798.3483758 476939 dgx-node-0-1.codia.edu.ec 201.159.221.21	1695097264 278.6578188	1695102108 816.1528029	192108921	895109446	1695109518 71.23765755	1695115158 4894,090337	1695115442 130.1416068	892113268	895119824	895119914	1695120220 134.6616607	1695120961	895121311	995121557	1695121831	1695122371	1695123752 1380.158476	892123269	892128988	895130584	1695130984 229.3269925	1695132714 203.3732076	895133013	995133207	1695140075 6675.402836 476939 dgx-node-0-1.codia.odu.ec 201.159.221.21	1695140701 71.35978866	1695141036 197.6078634	895141214	895141706	1695142210 104.8331921	1695142538 131.9834747
r	⊢			7-35-54	17-44-06				0-32-48	0-37-04	0-38-34		⊢	1-01-21	1-05-57	Н	⊢	1-42-32	2-07-49 1	3-09-48	3-36-24			4-16-53	4-20-07			┺	6-33-34	6-41-46		
date	2023-09-19_03-10-49	2023-09-19 04-21-04	2023-09-19-05-41-48	10H0_00003 2023-09-19_07-35-54 1695108954 343.2146995	3040 00004 2023-09-19 07-44-06 1695109446 492.3904169 476939 dgs-node-0-1.cedia.edu.ec 201.159.221.21	JOHO 00005 2023-09-19 07-45-18	2023-09-19,09-19-18	2023-09-19 09-24-02	1000008 2023-09-19_10-32-48 1095119568 4064.266164 470939 dgx-node-0-1.cedia.chi.ec 201.159.221.21	00H0_00009 2023-09-19_10-37-04 1695119824 192.208555	30H0 00010 2023-09-19 10-38-34 1695119914 89:63877296	10H0_00011 2023-09-19_10-43-40	2023-09-19,10-56-01	10460 00013 2023-09-19-11-01-51 1695121311 193-6992581 476939 dgx-node-0-1.cedia.edu.ec 201.159.221.21	40f0_00014 2023-09-19_11-05-57 1695121557 246.6340413 476939 dgx-node-0-1.cedia.chu.ec 201.159.221:21	JOHO 00015 2023-09-19,11-10-31	2023-09-19-11-19-31	JOHO 00017 2023-09-19 11-42-32	1040_00018 2023-09-19_12-07-49 1695125266 827.1856215 476939 dgx-node-0-1.cedia.chu.ec 201.159.221.21	1046 00019 2023-09-19-13-09-48 1695128988 143.3706887 476939 dgx-node-0-1.cedia.edu.ec 201.159.221.21	40f0 00020 2023-09-19-13-36-24 1695130584 89.30738521	2023-09-19_13-43-04	2023-09-19_14-11-54	40f0 00023 2023-09-19-14-16-53 1695133013 199.1109407 476939 dgx-node-0-1.cedia.edu.ec 201.159.221.21	40f0_00024 2023-09-19_14-20-07 1695133207 193-4771852 476939 dgx-node-0-1.cedia.edu.ee 201.159.221.21	10ff0 00025 2023-09-19 16-14-35	40HO 00026 2023-09-19 16-25-01	2023-09-19,16-30-36	Off0 00028 2023-09-19 16-33-34 1695141214 178.5386949	40f0 00029 2023-09-19 16-41-46 1695141706 406.0547113 476939 dgx-node-0-1.cedia.edu.ee 201.159.221.21	00f0_00030 2023-09-19_16-50-10	40ff0_00031 2023-09-19_16-55-38
trial_id	40ff0_0000	40ff0_00001	40ff0_00002	40ff0_00003	40ff0_00004	40ff0_00005	40ff0_00006	40ff0_00007	40ff0_00008	40ff0_00009	40ff0_00010	40ff0_00011	40ff0_00012	40ff0_00013	40ff0_00014	40ff0_00015	40ff0_00016	40ff0_00017	40ff0_00018	40ff0_00019	40ff0_00020	40ff0_00021	40ff0_00022	40ff0_00023	40ff0_00024	40ff0_00025	40ff0_00026	40f0_00027	40ff0_00028	40ff0_00029	40ff0_00030	40ff0_00031
training_iteration	11	4	t-	ee	9	_	79	2	89	ee	-	2	-			4	9	16	12	2	-	eo	2	eo	eo	627	-	00	2	9	_	2
done	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
should_checkpoint	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
time this iter s	68.88985372	68.50376415	112.0818956	111.0227704	80.96822143	71.23765755	74.14733648	62.43433332	63.31889129	63.14664245	89.63877296	65.75593257	80.72840214	63.09783888	81.26948452	69.10437942	65.09932709	87.20955706	68.88297629	69.8362031	89.30738521	74.59268498	97.48316526	65.04532671	63.81285501	110.7134373	71.35978866	65.04620504	86.46181059	66.24838829	104.8331921	63.82719517
ven-v	9032258065	0.96875	94444444	0.5	3838709677	8974358974	9705882353	4285714286	9705882353	0	5112781955	9285714286	0	0.5	0	9402985075	92727273	0.9583333333	8765432099	9285714286	5172413793	98484848	9787234043	8181818182	9577464789	9565217391	0	5094339623	0.5	9558823529	-	9189189189
add*a	968992248 0.	0.8125	0.8636363636 0.944444444	0	806122449 0.	1780487805 0.	847826087 0.	0 1920261 0	847826087 0.	0.5	55555556 0.	133333333 0.	0.5	0.5	0.5	172043011 0.	238095238 0.	0.875 0.	1860759494 0.	133333333 0.	1666666667 0.	404255319 0.	991150442 0.	234042553 0.	651685393 0.	Jel 538462 0.	0.5	185185185 0.	0.5	369565217 0.	766990291	1604651163 0.
tr_npv	3314143081 0.2	L		805594904	8547738511 0.	8419637444 0.8	8855144083 0.	1982223639 0.4	0.72614772 0.		5738944664 0.2	8692640381 0.5	9676318069			846017836 0.5	3301468543 0.7	800695235	892839731 0.5	8681504311 0.8	2200928038 0.6	873603469 0.5	3072890011 0.6	9094426337 0.3	805122744 0.8	0.13020541 0.8	235701284	1936920231 0.5	5051202442	8594061947 0.8	8227202146 0.3	8625323045 0.8
tr.ppv	956188631 0.3	0.6999750534 0.875872398	075627179 0.3	0.2907217284 0.2805594904	179978702 0.3	785176558 0.3	857218489 0.3	980307926 0.4	149891397 0.9	963924936 0.4	532986931 0.3	005972217 0.3	0.3300106849 0.3676318069	0.44544496 0.4694204401	486479363 0.3	897539236 0.	553677711 0.9	139067408 0.3	286234546 0.	986818863 0.3	199932101 0.3	868412622 0.3	234534078 0.8	449210141 0.8	114097591 0.3	489497296 0.9	0.3402908251 0.235701284	996109031 0.3	0.5023783427 0.5051202442	660143863 0.8	898537452 0.3	967321752 0.8
v.bpv	500625156 0.4	0.890625 0.6	0.904040404 0.8075627179 0.8728535941	0.25 0.2	949967084 0.6	87742339 0.7	7.0 1191000	826517274 0.4	092071611 0.8	0.25 0.3963924936 0.4167239866	334168755 0.5	80952381 0.8	0.25 0.3	0.5	0.25 0.2486479363 0.2976415549	787514043 0.6	255411255 0.6	8.0 29999991	813095796 0.8	80952381 0.7	01954023 0.4	126370084 0.7	389192243 0.6	707930368 0.6	114575091 0.8	013377926 0.8	0.25 0.3	139762404 0.4	0.5 0.5	964194373 0.7	883495146 0.6	896920176 0.7
tr_bpv	5135165856 0.7			ı	7673762765 0.8	3102407001 0.3	335181286 0.9	981265783 0.4	8011253084 0.9		5635965798 0.5	\$349306299 0.3	3488212459	1.4574327		678858798 0.8	927573127 0.8	3469881322 0.9	807315928 0.8	334161587 0.8	5250123524 0.3	\$274008046 0.9	7153712044 0.8	7271818239 0.7	8459610167 0.9	8751258919 0.9	ı	966514631 0.5	ı	8127102905 0.8	562869799 0.8	296322399 0.8
v_loss	3.514863875 0.6511535048 0.5135165856 0.57500625156 0.4956188631 0.534143081 0.5968992248 0.9032258065 0.9032258065 0.9032258065 0.9032258065 0.9032258065 0.9032258065 0.9032258065 0.9032258065 0.9032258065 0.9032258065 0.9032258065 0.9032258065 0.9032258065 0.9032258065 0.903268 0.9	1 0.4772234301 0.3113476634 0.7879237257	0.3512239006 0.2408688366 0.840208156	2.215335605 0.6913900111 0.2856406094	0.5291921793 0.3612131178 0.7673762765 0.8949967084 0.679978702 0.8547738511 0.806122449 0.9838709677	0.4029501144 0.2757805884 0.8102407001 0.887742389 0.7785176558 0.8419637444 0.8780487805 0.8974358974	0.359982976 0.2354445457 0.8356181286 0.9092071611 0.7357218489 0.8855144083 0.847826087 0.9705882343	0.8999739659 0.7411063313 0.4981265783 0.4626517274 0.4980307926 0.498223639 0.4967320261 0.4285714286	8 0.3103130045 0.2551254034 0.8611253084 0.9692071611 0.8149891397 0.9072614772 0.847826087 0.9705882353	1.25422367 0.6888334155 0.4065582401	804.0653514 0.757171452 0.5635965798 0.5334168755 0.5532986931 0.5738944664 0.555555556 0.5112781955	11 0.3551564495 0.2659260631 0.8349306299 0.880952381 0.8005972217 0.8692640381 0.835333333 0.9285714286	0.693005383 0.3488212459	13 0.7493309512 0.6949098706 0.4574327	14 2.461109757 0.6931510568 0.2731447456	15 0.5103963812 0.3395510018 0.7678858798 0.8787514043 0.6897539236 0.846017836 0.8172043011 0.9402985075	16 0.4785262029 0.3935081363 0.7927573127 0.8255411255 0.65536777711 0.3801468543 0.723809538 0.9272727273	17 0.3371580429 0.2440053225 0.8469881322 0.9166666667 0.8139067408 0.8800695235	18 0.3069913223 0.235785529 0.8607315928 0.8813095796 0.8286234546 0.892839731 0.886759494 0.8765432099	19 0.3606001521 0.2612317502 0.8334161587 0.880052381 0.7986818863 0.881504311 0.833333333 0.9285714286 69.8363031	20719.55086 3.480050564 0.5250123524 0.591954023 0.499932101 0.5500926038 0.60666666667 0.5172413733	21 0.3853331101 0.242866862 0.8274008046 0.9126370084 0.7868412622 0.8679603469 0.840425319 0.984848488	22 0.6921282489 0.4730080068 0.7153712044 0.8389192243 0.6234534078 0.8072890011 0.6991150442 0.9787234043	23 0.5849454864 0.4651616514 0.7271818239 0.7707936368 0.6449210141 0.8094426387 0.7234042553 0.8181818182 65.0453671	24 0.3382947485 0.2380371439 0.8459610167 0.9114575091 0.8114097591 0.8805122744 0.8651685393 0.9577464789 63.81285501	0.2825434078 0.2134927511 0.8751258919 0.9013377926 0.8489497296 0.9013020541 0.8461538462 0.9565217391 0.9562717391 0.9565217391 0.9565217391 0.9565217391 0.9565217391 0.9565217391 0.9565217391 0.9565217391 0.9565217391 0.9565217391 0.9565217391 0.9565217391 0.9565217391 0.9565217391 0.9565217391 0.9565217391 0.9565217391 0.9562717391 0.9565217391 0.95	26 86.96529479 0.6931672096 0.2873960546	0.7405436975 0.6954571009 0.4966514631 0.5139762404 0.4986109031 0.498920231 0.5185185185 0.5094339623 0.5185185185 0.5185185 0.	17.78943277 4.315719604 0.5037492934	29 0.4050840788 0.2685029507 0.8127102905 0.8964194373 0.7060143863 0.8594061947 0.8369565217 0.958823529 66.24838829	1.787232461 0.3650565743 0.7562869799 0.8883495146 0.6898537452 0.8227202146 0.7766990291	31 0.3686260943 0.2622787654 0.8296322399 0.8896920176 0.7967321752 0.8625323045 0.8604651163 0.9189189189 63.82719517
tr_loss	114865875 0.	772234301 0.	512239006 0.	15335605 0.	291921793 0.	029501144 0.	159982976 0.	999739659 0.	103130045 0.	25422367 0.	4.0653514 6	551564495 0.	0.7756399759 0	493309512 0.	161109757 0.	103963812 0.	785262029 0.	371580429 0.	069913223 0	506001521 0.	719.55086 3	853331101 0.	921282489 0.	849454864 0.	382947485 0.	825434078 0.	96529479 0.	405436975 0.	78945277 4	0.59840788 0.	87232461 0.	586260943 0.
H	0 3.5	1 0.4.	2	3 2.2	4 0.5	5 0.4	9	7 0.8	8	6	10 80	11 0.3	12 0.7.	13 0.7,	14 2.4	15 0.5	16 0.4.	17 0.3,	18 0.30	19 0.34	30	21 0.3	22 0.6	23 0.5	24 0.3	25 0.2	36.86	27 0.7.	28 17	29 0.40	08	31 0.30

Table B.7: VGG19 Hyperparameter Tuning on Synthetic Dataset: best results per trial using validation loss as metric.

pa/gjuoo	0.004057259988	0.006160013466	0.002267878726	0.006680542817	0.007022273852	0.001115369569	0.006836543417	0.004285143411	0.006174156421	0.003806143381	0.001055669552	0.001372154833	0.007936684594	0.008363101202	0.009247485124	0.006574773984	0.0002543811846	0.007069493716	0.004534515372	0.002461740746	0.005737552548	0.003289330911	0.001402736282	0.007304940458	0.0005339304194	0.007143531658	0.005814579208	0.0004073464208	0.009173115098	0.009728592512	0.008181594117	0.006093429983
config/lr	0.02291154008	0.003665737195	0.000144815453	0.01341145855	0.00272243137	0.0001348969613	0.0008859104983	0.07024536547	0.0004105288637	0.06361179329	0.03800694642	0.0002090720054	0.008028446355	0.02852980691	0.07405577337	0.004511012484	0.006142432526	0.0001969590185	0.0003367869903	0.0003984424533	0.07812642972	0.0001622058465	0.04100482089	0.01449432222	0.0003258297361	0.0001783647433	0.0931531612	0.03057269777	0.03946097237	0.0005373097407	0.002735066653	0.0001953925933
config/hn2	8192	2048	256	128	64	8192	102.4	960#	256	128	16384	8192	32	1024	512	32	128	16384 (9600	512	2048	32	1024	256	128	8192	32	8192	8192	64 (1024	226
config/hn1 c	1024	128	16384	16384	8192	20:48	4096	64	256	226	4096	512	4096	6-4	8192	1024	6-4	4096	1024	226	8192	4096	16384	256	1024	16384	1024	32	8192	1024	16384	1024
config/batch_size	512	256	64	6-4	128	128	512	128	256	256	64	64	64	128	256	256	512	64	256	512	512	128	512	512	128	128	512	256	128	128	256	256
iterations since restore on	10	41		-	9	9	28	3	09	-	2	_	-	3	_	1	20	16	23	20	_		2	_		2	_		-	10	1	_
time since restore i	729.4585221	2782.602919	356.5902636	122.4397545	492.5904169	410.5419674	2200.51964	192.1119113	3875.375894	66.18437147	174.9913652	68.9057281	80.72840214	193,6992581	85.11370444	69.75989079	335,5977619	1380.158476	1577.144829	9809297096	89.30738521	229.3269925	203.373.2076	69.8227706	193.4771852	225.087492	71.35978866	197.6078634	92.07688427	673,310,9946	104.8331921	68.15627956
node,ip t	01.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	01.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	01.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21	201.159.221.21
hostname	dgx-node-0-1 cedia.edu.ec 3	dgx-node-0-1.codia.odu.oc 20	dgx-node-0-1 codia odu.ec 2	dgx-node-0-1 cedia.edu.ec 2	dgx-node-0-1 codia odu.ec 3	dgx-node-0-1 cedia edu.ec 2	dgx-node-0-1.cedia.edu.ec 2	dgx-node-0-1 codia odu.ec 3	476939 dgx-node-0-1 codia odu.ec 2	dgx-node-0-1 codia odu.ec 2	dgx-node-0-1 codia.odu.ec 3	dgx-node-0-1 codia odu.ec 2	dgx-node-0-1 cedia edu.ec	dgx-node-0-1 cedia.edu.ec 2	476939 dgx-node-0-1.codia.odu.ec 20	dgx-node-0-1 cedia edu.ec 2	dgx-node-0-1 cedia.edu.ec 3	dgx-node-0-1 codia.odu.oc	dgx-node-0-1 cedia edu.ec 2	476939 dgx-node-0-1.cedia.edu.ec 2	476939 dgx-node-0-1 cedia edu.ec 201.159.221.21	dgx-node-0-1 cedia edu.ec 2	dgx-node-0-1 codia.odu.ec 20	dgx-node-0-1 codia odu.ec 3	dgx-node-0-1 codia odu.ec 2	dgx-node-0-1 cedia.edu.ec 3	476939 dgx-node-0-1 codia.odu.ec 201.159.221.21	dgx-node-0-1.codia.odu.ec 2	dgx-node-0-1.cedia.edu.ec 2	dgx-node-0-1 codia odu.ec 3	dgx-node-0-1 cedia.edu.ec 2	dgx-node-0-1.cedia.edu.ec
bid s	476939	476939	476939	476939	476939	476939	476939	476939	476939 c	476939	476939	476939	476939	476939	476939 c	476939	476939	476939	476939		476939	476939	476939	476939	476939	476939		476939	476939	476939	476939	476939
time total s	729.4585221	2782,602919	356.5902636	122.439754	492,5904169	410.5419674	2200.51964	192,1119113	3875.37589	66.18437145	174.9913652	68.9057281	80.72840214	193,6992581	85.1137044	1695121627 69.75989079	335,5977619	1695123752 1380.158476	1577.144829	360.6560868	89.30738521	229.3269925	203.3732076	69.8227706	193.4771852	225.087492	71.35978866	197.6078634	92.07688427	673.3109946	104.8331921	1695142474 68.15627956
timestamp	1695092980	1695099768	1695101648	1695108733	1695109446	1695109857	1695112465	1695115504	1695119379	1695119698	1695119999	1695120154	1695120961	1695121311	1695121396	1695121627	1695122306	1695123752	1695126019	1695129205	1695130584	1695130984	1695132714	1695132884	1695133207	1695133624	1695140701	1695141036	1695141128	1695141974	1695142210	1695142474
date	2023-09-19_03-09-40	2023-09-19-05-02-48	2023-09-19-05-34-08 1695101648	2023-09-19_07-32-13 1695108733 122,4397545	2023-09-19 07-44-06	2023-09-19_07-50-57	2023-09-19,08-34-25	2023-09-19-09-25-04 1695115504	2023-09-19_10-29-39 1695119379 3875.375894	2023-09-19_10-34-58 1695119698 66.18437147	2023-09-19-10-39-59	2023-09-19_10-42-34	2023-09-19,10-56-01	2023-09-19,11-01-51	30H0_00014 2023-09-19_11-03-16 1695121396 85.11370444	2023-09-19,11-07-07	2023-09-19-11-18-26	2023-09-19,11-42-32	2023-09-19_12-20-19	2023-09-19-13-13-25 1695129205 360,6560869	40H0_00020 2023-09-19_13-36-24 1695130584 89.30738521	2023-09-19_13-43-04	2023-09-19_14-11-54	2023-09-19_14-14-44	2023-09-19_14-20-07	2023-09-19_14-27-04 1695133624	2023-09-19_16-25-01 1695140701 71.35978866	2023-09-19_16-30-36 1695141036	2023-09-19,16-32-08	2023-09-19 16-46-14	2023-09-19_16-50-10 1695142210	2023-09-19,16-54-34
trial_id	00000 0HO	. 10000 OHOF	40ff0_00002	40ff0_00003	40ff0_00004 ;	40ff0_00005	30000 0HO	70000 OHOF	#00000_00008	40 ft0 00000	40ff0_00010	40ff0_00011	#0ff0_00012 ;	40ff0_00013 [3	40ff0_00014 ;	40ff0_00015 :	40ff0_00016	40ff0_00017	\$1000 0HO	40ff0_0019 :	40ff0_00020		#0ff0_00022 ;	40ff0_00023		#0ff0_00025	\$0000 QHOP	40ff0_00027	40ff0_00028	40ff0_00029 ;	#0ff0_00030	40ff0_00031
training_iteration	10	41		-	9	9	88	3	09	-	2	-	_		-	_	-01	16	23	10	-		2	-		2	-		-	10	-	-
done	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
should_checkpoint	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
time_this_iter_s	69.43802691	67.173908	116.506386	122.4397545	80.96822143	68.77669334	84.47188807	61.97030449	63.3950088	66.18437147	85.35259223	68.9057281	80.72840214	63.09783888	85.11370444	69.75989079	66.13054442	87.20955706	67.53002071	69.37031436	89.30738521	74.59268498	97.48316526	69.8227706	63.81285501	107.1484058	71.35978866	65.04620504	92.07688427	68.28673077	104.8331921	68.15627956
adu-a	1	1	-	0	0.9838709677	0.9565217391	0.9710144928	9.0	-	0	0.6363636364	-	0	0.5	0.5	0.9830508475	86'0	0.9583333333	0.9848484848	0.9210526316	0.5172413793	0.9848484848	0.9787234043	-	0.9577464789	1	0	0.5094339623	0.5555555556	0.9848484848		-
add*a	0.5194805195	0.8163265306	0.8421052632	0.5	0.806122449	0.9035954017 0.8461538462 0.9565217391	0.8571428571	0.5032258065	0.8333333333	0.5	0.5100671141	0.8333333333	0.5	0.5	0	0.7821782178	0.71818182	0.875	0.8404255319	0.880952381	79999999990	0.8404255319	0.6991150442	0.6896551724	0.8651685393	0.8421052632	0.5	0.5185185185	0.6153846154	0.8404255319	0.7766990291	0.8333333333
tr_npv	0.5161039604	P.1262061810	0.8669674199	0.2283436757	0.8547738511	0.9035954017	0.8806263868	0.4842742133	0.9050215357	0.4461166582 0.460609213	0.496573971	0.837942012	0.3676318069	0.44544496 0.4694204401	0.3881713093	0.7165391149	0.9379003335	0.8800695235	0.9073101346	0.8846322317	0.5500926038	0.8679603469	1100682208:0	0.7726614677	0.8805122744	0.8606418778	0.3402908251 0.235701284	0.4936920231	0.5509860513	0.8682923708	0.8227202146	0.8376186914
tr.ppv	0.5266719135	0.7891631703 0.9081632653 0.6992667696	0.9210526316 0.7819730418 0.8669674199 0.8421052632	0.3522144872 0.2283436757	0.679978702	0.862633957	0.7881516061	0.4887661826	0.8151943699	0.4461166582	0.4914913858	0.7741385952	0.3300106849	0.44544496	0.25 0.3557282785 0.3881713093	0.6470476006	0.6385753499	0.8139067408	0.8389667581	0.8290208737	0.499932101	0.7868412622	0.6234534078	0.632105135	0.8114097591	0.7869844141	0.3402908251	0.4996109031	0.5308828775	0.767570664	0.6898537452	0.7689401701
v.bpv	0.7597402597	0.9081632653	0.9210526316	ı	0.8949967084	0.9013377926	0.9140786749	0.5516129032	0.9166666667	0.25	0.5732153752	0.9166666667	0.25	0.5		0.8826145326	0.8490909091	0.9166666667	0.9126370084	0.9010025063	0.591954023	0.9126370084	0.8389192243	0.8448275862	0.9114575091	0.9210526316	0.25	0.5139762404	0.5854700855	0.9126370084	0.8883495146	0.9166666667
tr_bpv	0.5213879369	0.7891631703	0.8244702309	0.2902790815	0.7673762765	0.8831146794	0.8343889964	0.4865201979	0.8601079528	0.4533629356	41.02001572 0.4940326784 0.5732153752 0.4914913858 0.496573971 0.5100671141 0.6363636364	0.8060403036	0.3488212459	0.4574327	0.3719497939	0.6817933577	0.7882378417	0.8469881322	0.2453391105 0.8731384463 0.9126370084 0.8389667581 0.9073101346 0.8404255319 0.98484848888888888888888888888888888888	0.8568265527	0.5250123524	0.8274008046	0.7153712044	0.7023833013	0.8459610167	0.823813146	0.2879960546	0.4966514631	0.5109341614	0.2689034641 0.8179315174 0.9126370084 0.767570664 0.8682923708 0.844255319 0.9848484848	0.7562869799	0.8032794308
v_loss	0.7028210759	10021620050	12582319379	1,6932426691	3612131178	1,3269532323	2704241872	7803300619	12912024856	1,6931854486	41.02001572	12769338191	0.693005383	90286061691	1,693,427,9203	13939464688	A113613069	1.2440053225	12453391105	2788267434	3.480050564	1.2442866862	4730080068	1,5038048029	12330371439	1.2697630823	1,693,167,2096	1,6954571009	8.726045609	12689034641	3650565743	1.275.2068937
tr_loss	3.463201761 0	0.4955157647 0.	0.3835567451 0.2582319379 0.8244702309	3.27620601 0.6932426691 0.2902790815 0.25	0.5291921793 0.3612131178 0.7673762765 0.8949967084 0.679978702 0.8547738511 0.806122449 0.9838709677 0.8847738511 0.806122449 0.9838709677 0.8847738511 0.88477738511 0.88477738511 0.88477738511 0.88477738511 0.88477738511 0.8847778511 0.8847778511 0.8847778511 0.8847778511 0.8847778511 0.	0.2608482017 0.3269532323 0.8831146794 0.9013377926 0.862633957	0.3608697388 0.2704241872 0.8343889964 0.9140786749 0.7881516061 0.8806263868 0.8571428571 0.9710144928 0.971014928 0.971	1.170847256 0.7803300619 0.4865201979 0.5516129032 0.4887661826 0.4842742133 0.5032258065	0.3112348287 0.2912024856 0.8601079528 0.9166060667 0.8151943699 0.9050215357 0.8333333333	13.95370901 0.6931854486 0.4533629356	29.17925122 4	0.4124632752 0.2769338191 0.8060403036 0.9166666667 0.7741385952 0.837942012	0.7756399759 0	0.7493309512 0.6949098706 0.4574327	14 1631.394027 0.6934279203 0.3719497939	0.6149518029 0.3939464688 0.6817933577 0.8826145326 0.6470470006 0.7165391149 0.7821782178 0.9830508475	0.4887265327 0.4113613069 0.7882378417 0.8490909091 0.6385753499 0.937900335 0.7181818182	0.3371580429 0.2440053225 0.8469881322 0.9166666667 0.8139067408 0.8800695235	0.2842535949 0.	19 0.3142577028 0.2788267434 0.8568265227 0.901002063 0.8290208737 0.8846322317 0.80005331 0.9210536316	20 20719.55086 3.4800.50564 0.5250.123524 0.591954023 0.499922.101 0.5500928038 0.6666666667 0.5172413733 89.30738521	0.3853331101 0.2442866862 0.8274008046 0.9126370084 0.7868412622 0.8679603469 0.840425319 0.9948484888888 0.84679603469 0.84648888888888888888888888888888888888	0.6921282489 0.4730080068 0.7153712044 0.8389192243 0.6234534078 0.8072890011 0.6991150442 0.9787234043 0.8972890043 0.89787234044 0.89787234044 0.89787234044 0.89787234044 0.89787244044 0.89787244044 0.89787244044 0.8978724444 0.8978724444 0.8978244444 0.8978724444 0.89787244444 0.8978724444 0.89787244444 0.89787244444 0.89787244444 0.89787244444 0.897872444444 0.89787244444 0.897872444444 0.8978724444444444444444444444444444444444	1.496508522 0.5038048029 0.7023833013 0.8448275862 0.632105135 0.7726614677 0.6896551724	0.3382947485 0.2330371439 0.8456610167 0.9114575091 0.8114097591 0.805122744 0.8651685393 0.9577464789 0.9677464789 0.9677464789 0.9677464789 0.9677464789 0.9677464789 0.9677464789 0.9677464789 0.9677464789 0.9677464789 0.9677464789 0.9677464789 0.9677464789 0.9677464789 0.967748789 0.967748789 0.967748789 0.967748789 0.967748789 0.967748789 0.967749 0.9677489 0.9	25 0.3824843672 0.2697630823 0.823813146 0.9210526316 0.7869844141 0.8606418778 0.8421052632	86.96529479 0.6931672096 0.2879960546 0.25	0.7405436975 0.6954571009 0.4966514631 0.5139762404 0.4966109031 0.4936920231 0.5185185185 0.5094339623 0.5185185185 0.5185185 0.5185185185 0.5185185	2943.892982 8.726045609 0.5409344644 0.5854700855 0.5308928775 0.5509860513 0.6153846154 0.555555556	0.3977393407 0.	1.787232461 0.3650565743 0.7562869799 0.8883495146 0.6898537452 0.8227202146 0.7766990291	0.4220852857 0.2752068937 0.8032794308 0.9166066667 0.7689401701 0.8376186914 0.8333333333
	0	Ε	C-1	65	4	10	9	t-	∞	6	10	=	12	13	14	121	91	12	18	19	30	51	- 22	23	24	22	98	27	88	8	8	E5

Table B.8: VGG19 Hyperparameter Tuning on Synthetic Dataset: best results per trial using BPV as metric.

config/wd	0.001510451824	01140190987	01978035796	0.003386165854	0.009949709149	0.00850181588	0.0086421597	0.001056055423	0.007654271044	0.008537335751	0.007435148118	02074727611	0.001008242634	0.006012991941	0.0008896224369	0.00576383982	0.003248522892	0.004451056956	0.0007651831475	0.006405663003	0.0003600208533	0.003790434832	0.0002516977864	0.001439345557	0.007384154559	0.002874425557	0.00309630442	0.007304670286	0.004578114407	0.009887470506	0.005096968715	006421349858
	0.0002047653346 0.0	0.0001927539469 0.001140190987	0.0001261700242 0.001978035796	0.00378248955 0.0	0.002114775439 0.0	0.001262158737 0.0	0.01588512397 0.0	0.000232623984 0.0	0.0003361220842 0.0	0.01853113801 0.0	0.005768671246 0.0	0.0001678445014 0.00207472761	0.002083973911 0.0	0.01748139827 0.0	0.00015132965 0.0	7	0.007547896257 0.0	-	0.007588661567 0.0	0.01435163916 0.0	0.05458950306 0.0	0.005348451252 0.0	0.003154670321 0.0	0.003747556755 0.0	0.0002619940921 0.0	0.0001740546879 0.0	0.00676590819 0.0	0.05799739438 0.0	0.008760911056 0.0	0.05638458363 0.0	0.001038212402 0.0	0.0003731782659 0.000642134985
size config/	0.00020	0.00019	0.00012	0.00378	0.00213	0.00126	0.01588	0.00023	0.00033	0.01853	0.00576	0.00016	0.00208	0.01748	0.00013	0.0008	0.0075	0.01012	0.00758	0.01435	0.05458	0.00534	0.0031	0.0037	0.00026	0.00013	0.00067	0.05798	0.00876	0.05638	0.00108	0.00037
config/batch	256	256	128	256	512	64	512	256	256	64	64	512	128	128	512	128	512	512	64	256	256	256	128	128	256	64	128	128	256	128	64	128
sincerestore																																
ore iterations	2	-	-	_	-	36	60	-	-	-	-	-	2	4	-	-	-	-	-	œ	20	1	-	2	1	-	2	-	-	3	1	-
time-since restore iterations-since-restore config/batch-size config/1	81.78189349	43.38168335	41.41273308	42.96527529	48.9447186	1499.751575	144.3514209		46.17835569	44.51810813	45.08634496	45.82599568	84.91990399	174.0328171	47.20921993		53.4594996	46.60522342	44.97250843	371.7680519	870.9627922	44.34403586	41.50500727	85.49795771	46.12986994	44.46798253	252.429544	44.33067536	46.4411478	135.2399769	46.75437188	203.5815022
node.ip	201.159.221.20	1695056482 43.38168335 2021255 dgx-node-0-0.cedia.edu.ec 201.159.221.20	1695056610 41.41273308 2621255 dgx-node-0-0.ccdin.cclu.cc 201.159.221.20	dgx-node-0-0.cedia.edu.ec 201.159.221.20	2621255 dgx-node-0-0.cedia.edu.ec 201.159.221.20	c92af.00005 2022-09-18-17-47-23 1695059243 1499.751575 2621255 dgx-node-0-0-cedia-cedu.ce 201.159.221.30	c92af_00006 2023-09-18_18-09-13 1695060553 144.3514209 2621255 dgx-node-0-0.ocdia.och.cc 201.159.221.30	1695060745 46.15361738 2621255 dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.codin.edu.ec 201.159.221.20	1695061020 44.51810813 2621255 dgx-node-0-0.cedia.edu.ec 201.159.221.20	c92af_00010 2023-09-18_18_18_19-12 1695061132 45.08634496 2621255 dgx-node-0-0.codin.cdn.cc 201.159.221.30	1695061284 45.82599568 2621255 dgx-node-0-0.cedia.edu.ec 201.159.221.20	c92af_00012 2023-09-18_18_318_31-42 1695061902 84;91990399 2621255 dgx-node-0-0.cedia.ced 201.159.221.20 84;91990399	c92af.00013 2022-09-18_18-35-19 1695062119 174.0328171 2621255 dgx-node-0-0.cedin.cell.cell.cell.cell.cell.cell.cell.cel	c92af_00014 2021-09-18_18-37-30 1695062250 47,20921993 2021255 dgx-node-0-0.cedia.edu.ec 201.159.221.30	2023-09-18_19-29-43 1695065383 52.00335908 2621255 dgx-node-0-0.ceda.edu.ec 201.159.221.20	2621255 dgx-node-0-0.cedia.edu.ec 201.159.221.20	1695066099 46.60522342 2621255 dgx-node-0-0.cedia.edu.ec 201.159.221.20	c92af_00018 2025-09-18_19-43-55 1695066235 44.97250843 2621255 dgx-nock-0-0.ceda.edn.ec 201.159.221.20	e92af 00019 2022-09-18-19-51-32 1695066692 371.7680519 2621255 dgx-node-0-0-cedia-cedn.ce 201.159.221.30	2023-09-18_20-09-16 1695067756 870.9627922 2621255 dgx-node-0-0.oxdia.edu.ec 201.159.221.20	1695069024 44.34403586 2621255 dgx-node-0-0.cxlin.edu.ec 201.159.221.20	c92af.00022 2022-09-18-20-32-38 1695069158 41.50500727 2621255 dgx-node-0-0.cedia.edn.ec 201.159.221.20	2023-09-18_20-35-31 1695069331 85.49795771 2621255 dgx-node-0-0.cedia.edn.ec 201.159.221.20	1695069418 46.12986994 2621255 dgx-node-0-0.ccdia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec 201.159.221.20	2621255 dgx-node-0-0.cedia.edu.ec 201.159.221.20	c92af_00027 2021-09-18-20-50-49 1695070249 44.33067536 2621255 dgx-node-0-0.cedia.celn.ce 201.159.221.30	2621255 dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec 201.159.221.20	c92af 00030 2029-09-18-20-57-43 1695070663 46.75437188 2621255 dgx-node-0-0.cetia.cet 20.159.221.30	c92af_00031 202s-06-18_21-04-14 1695071054 203.5815022 2621255 dgx-node-0-0.cedn.celn.celn.ce 201.159.221.30 203.5815022
	dgx-node-0-0.cedia.edu.ec	0.ocdia.edu.ec	0.ccdin.edu.ec	0.codia.edu.ec	0.ocdia.edu.ec	0.codin.edu.ec	0.codia.edu.ec	0.oedia.edu.ec	0.codin.edu.ec	0.oxdia.edu.ec	0.codin.edu.ec	0.oxdia.edu.ec	0.ocdia.edu.ec	0.codin.edu.ec	0.oxdia.edu.ec	0.ocdia.edu.ec	0.codin.edu.ec	0.oxdia.edu.ec	0.ocdia.edu.ec	0.codin.edu.ec	0.oxdia.edu.ec	0.cedia.edu.ec	0.cedia.edu.ec	0.ocdia.edu.ec	0.cedia.edu.ec	0.oxdia.edu.ec	0.ocdia.edu.ec	0.codin.edu.ec	0.oxdia.edu.ec	0.ocdia.edu.ec	0.cedin.edu.ec	0.oxdia.edu.ec
hostname	-	-0-spou-xSp	-0-apou-xgp		-0-epou-xgp	-0-apou-x3p	-0-apou-xap	-0-epou-xgp	-0-apou-x3p	-0-apou-x3p	-0-apou-x3p	-0-apou-xsp	-0-spou-xSp	-0-apou-x3p	-0-apou-xsp	-0-spou-xSp	-0-apou-x3p	-0-apou-xsp	-0-apou-xSp	-0-apou-x3p	-0-apou-x3p	- dgx-node-0-	-0-apou-x8p	-0-spou-xSp	- dgx-node-0-	-0-apou-xsp	-0-spou-xSp	-0-apou-x3p	-0-apou-xsp	-0-spou-xgp	-0-apou-xsp	-0-apou-xsp
time_total_s pid	349 2621255	335 2621255	1308 2621255	7529 2621255		575 2621255	1209 2621255	.738 2621255	569 2621255	813 2621255	1496 2621255	1568 2621255	399 2621255	171 2621255	993 2621255	908 2621255		342 2621255	1843 2621255	519 2621255	922 2621255	1586 2621255	1727 2621255	5771 2621255	1994 2621255	253 2621255	44 2621255	536 2621255		769 2621255	188 2621255	022 2621255
\vdash	900 81.78189349	43.38168	610 41.41273	111 42,96527529	1695057648 48.9447186	243 1499.751	553 144.3514	745 46.15361	1695060887 46.17835569	020 44.51810	152 45.08634	284 45.82599	902 84:91990	119 174.0328	250 47.20921	383 52.0033	531 53.4594996	099 46.60522	235 44.97250	892 371.7680	756 870.9627	024 44.34408	158 41.50500	331 85.49790	418 46.12986	688 44.46798253	166 252.4290	249 44.33067	387 46.44114	614 135.2399769	563 46.75437	054 203.5812
timestamp	-20 1695053900	-22 1695056	-30 1695056	-51 1695057111		-23 1695059	-13 1695060		-	-00 1695061	-12 1695061		-42 1695061	-19 1695062	-30 1695062	-43 1695065	-11 1695065531	-39 1695066	-22 1695066	-32 1695066	-16 1695067	-24 1695069	-38 1695069	-31 1695069		-28 1695069688	-26 1695070	-49 1695070	-07 1695070	-54 1695070614	-43 1695070	-14 1695071
97	2023-09-18 16-18-20	23-09-18_17-01	2023-09-18_17-03-30	2023-09-18_17-11-51	2023-09-18_17-20-48	23-09-18_17-47	23-09-18_18-09	2023-09-18_18-12-25	2023-09-18 18-14-47	2023-09-18_18-17-00	23-09-18_18-19	2023-09-18 18-21-24	23-09-18_18-31	23-09-18_18-35	23-09-18_18-37	23-09-18_19-29	2023-09-18 19-32-11	2023-09-18_19-41-39	23-09-18_19-43	23-09-18 19-51	23-00-18_20-09	2023-09-18_20-30-24	23-09-18-20-32	23-09-18_20-35	2023-09-18_20-36-58	2023-09-18_20-41-28	23-09-18_20-49	23-09-18-20-50	2023-09-18_20-53-07 1695070387 46.4411478	2023-09-18_20-56-54	23-09-18-20-57	23-09-18_21-04
alid date	c92af_00000 203	c92af_00001 2023-09-18_17-01-22	c92af_00002 20;	c92af_00003 203	c92af_00004 203	2af_00005 20;	2af_00006 20;	c92af_00007 203	c92af_00008 20;	c92af_00009 203	2af_00010 20;	c92af_00011 205	2af_00012 20;	2af_00013 20;	2af_00014 200	c92af_00015 200	c92af_00016 20;	c92af_00017 203	2af_00018 205	2af_00019 20;	c92af_00020 203	c92af_00021 20;	2af 00022 200	c92af_00023 205	c92af_00024 20;	c92af_00025 203	c92af_00026 2023-09-18_20-49-26 1695070166 252.429544	2af_00027 20;	c92af_00028 205	c92af_00029 20;	2af_00030 20;	2af_00031 200
training iteration trialid	(G)	(60)	69	69	69	(6)	69	(G)	(6)	(G)	(6)	(G)	(60)	(6)	(G)	(60)	(6)	(G)	69	(6)	(G)	(c)	69	(60)	(4)	(G)	(60)	(6)	(G)	(29)	(G)	(G)
H	FALSE 2	FALSE 1	FALSE 1	FALSE 1	FALSE 1	FALSE 36	FALSE 3	FALSE 1	FALSE 1	FALSE 1	FALSE 1	FALSE 1	FALSE 2	FALSE 4	FALSE 1	FALSE 1	FALSE 1	FALSE 1	FALSE 1	FALSE 8	FALSE 20	FALSE 1	FALSE 1	FALSE 2	FALSE 1	FALSE 1	FALSE 5	FALSE 1	FALSE 1	TRUE 3	FALSE 1	FALSE 1
time this iters should checkpoint done	EA	[FA]	FA	EA	[FA]	EA	EA	[FA]	EA	EA	EA	EA	[FA]	EA	EA	[FA]	EA	EA	FA	EA	EA	EA	FA	[FA]	EA	EA	[FA]	EA	EA	TR	EA	EA
s should ch	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
time this iter	40.44378686	43.38168335	41.41273308	42.96527529	48.9447186	40.5281868	46.35516334	46.15361738	46.17835569	4451810813	45.08634496	45.82599568	43.52033758	43.93136978	47.20921993	52.00335908	53.4594996	46.60522342	44.97250843	47.70639682	42.27307773	44.34403586	41.50500727	44.5700407	46.12986994	44.46798253	37.96417809	44.33067536	46.4411478	49.55819941	46.75437188	203.5815022
vanpv	0.2037037037	0.2	0.2	0.8382352941 0.2268041237 42.965275	0.72727273 0.1888111888 48.944718	0.2090909091	0.2	0.2024539877 46.153617	0.2	0.1940298507	0.2155172414	0.2	0.2038216561 43.520337	0.1666666667	0.2	0.2038216561 52.003359	0.7941176471 0.1984732824 53.459499	0.1958041958	0.214953271	0.2115384615	0.3414634146	0.2026143791	0.244444444	0.2015503876	0.2	0.2038216561 44.467982	0.1985294118	0.2816901408	0.1972789116	0.222222222	0.2045454545	0.1948051948
Add-v				8382352941	72727273	81818182				7741935484	8367346939			7943262411			7941176471	TT2T2T2T2T	8275862069	8196721311	8467741935	833333333	853333333	8055555556			7931034483	8617021277	TTTTTTT8	833333333	81818182	T2T2T2T3T
tr.npv v	8023113263 1	7362190106 0	7430379859 0	8318036227 0	8017439067 0	8519781155 0	8561855258 0	7493235957 1	0.7550229171 0	8425160233 0	8438966517 0	7150604256 0	807708672 0	8536014469 0	7129495073 0	8155119217 0	8286243611 0	8205302212 0	8515673264 0	8569552324 0	8683179834 0	8339662242 0	8419717147 0	8626394095 0	7471017607 0	7759799679 0	8596245117 0	8424308455 0	8407856324 0	8412425458 0	8273504266 0	790127465 0
tr_ppv tr	.7527619618 0.	0.5241846072 0.7362190106 0	0.5621333606 0.7430379859 0	.7294671957 0.	6892189135 0.	.7667188048 0.	0.7628533052 0.8561855258 0.8	5809918283 0.	0.598405472 0.	6953520635 0.	.7336848358 0.	0.4098199713 0.7150604256 0	.7637042439 0.	.7327926333 0.	0.3897565248 0.7129495073 0	.7182991784 0.	.7059468895 0.	.6963997908 0.	.7199813718 0.	.7515450122 0.	.7221564461 0.	.7109961777 0.	.7435225317 0.	.758405511 0.	0.5535296573 0.7471017607	0.6515318551 0.7759799679 0.875	.7723303251 0.	67604501 0.	.7205239486 0.	.6811011113 0.	.7348819294 0.	.6870277047 0.
v_bpv tr	0.6018518519 0.7527619618 0.8023113263			0.4044851849 0.7670581341 0.7806354092 0.5325197089 0.7294671957 0.8318036227	0.4509404558 0.9277585149 0.7454814101 0.458041958 0.6892189135 0.8017439067	0.3670129182 [0.8190974593 [0.8096484601 0.5136393636] [0.7667188048 [0.8519781155 [0.81818182 [0.209099999] [40.5281868] [0.81818182 [0.81818182] [0.8181818] [0.8181818] [0.8181818] [0.8181818] [0.8181818] [0.8181818] [0.8181818] [0.8181818] [0.8181818] [0.8181818] [0.8181818] [0.8181818] [0.8181818] [0.8181818] [0.8181818] [0.8181818] [0.8181818] [0.8181818] [0.8181818] [0.81818] [0.8181818] [0.8181818] [0.81818]		0.6012269939 0.5809918283 0.7493235957		0.4809822473 1.055403233 0.7689340434 0.4841116996 0.6953520635 0.8425160233 0.7741935484 0.1940298507 44518108 0.7741935484 0.1940298507 44518108 0.7741935487	10 0.404574254 0.8374389415 0.7887907437 0.5261259676 0.7336848358 0.8438966517 0.8367346939 0.2155172114 45.0863448 0.8438966517 0.8467846939 0.846784699 0.84	0.1	12 0.3540434386 1.203875308 0.8122375555 0.539410828 0.7637042439 0.807708672 0.875	13 - 0.4089000434 + 0.564137859 + 0.7931970401 + 0.4804964539 + 0.7327926333 + 0.8536014469 + 0.7943262411 + 0.1666666667 + 43.9313697 + 0.201666666667 + 0.201666666667 + 0.2016666666667 + 0.2016666666667 + 0.2016666666667 + 0.20166666666666 + 0.20166666666666 + 0.20166666666666 + 0.2016666666666 + 0.2016666666666 + 0.2016666666666 + 0.2016666666666 + 0.2016666666666 + 0.2016666666666 + 0.201666666666 + 0.201666666666 + 0.201666666666 + 0.2016666666666 + 0.2016666666666 + 0.201666666666 + 0.2016666666666 + 0.2016666666666 + 0.2016666666666 + 0.2016666666666 + 0.201666666666 + 0.201666666666 + 0.201666666666 + 0.2016666666666 + 0.2016666666666 + 0.2016666666666 + 0.201666666666 + 0.201666666666 + 0.20166666666 + 0.201666666666 + 0.201666666666 + 0.201666666666 + 0.20166666666 + 0.20166666666 + 0.201666666666 + 0.201666666666 + 0.20166666666 + 0.201666666666 + 0.201666666666 + 0.201666666666 + 0.201666666666 + 0.201666666666 + 0.2016666666666 + 0.201666666666 + 0.201666666666 + 0.20166666666 + 0.2016666666666 + 0.2016666666666 + 0.20166666666666 + 0.20166666666666 + 0.2016666666666 + 0.2016666666666 + 0.2016666666666 + 0.201666666666666 + 0.20166666666666 + 0.20166666666666 + 0.20166666666666 + 0.20166666666666 + 0.20166666666666 + 0.201666666666666 + 0.20166666666666 + 0.20166666666666 + 0.201666666666666 + 0.20166666666666666 + 0.201666666666666666666666 + 0.20166666666666666666666666666666666666		15 0.4225117842 1.085533977 0.7669055501 0.539410828 0.7182991784 0.8155119217 0.875	16 0.4352161764 0.9129543304 0.7672856253 0.4962954648 0.7059468895 0.8286243611	0.4530499624 1.025083542 0.758465006 0.4842657343 0.6963997908 0.8205302212 0.7727272777 0.1958041958 46.605223 0.772727277 0.1958041958 46.605223 0.77272727 0.77272777 0.772727777 0.772727777 0.7727277 0.77272777 0.77272777 0.77272777 0.7727277 0.77272777 0.77272777 0.77272777 0.77272777 0.77272777 0.77272777 0.77272777 0.77272777 0.77272777 0.77272777 0.77272777 0.77272777 0.772727777 0.77272777 0.77272777 0.77272777 0.77272777 0.772727777 0.772727777 0.772727777 0.772727777 0.772727777 0.772727777 0.772727777 0.772727777 0.772727777 0.772727777 0.772727777 0.772727777 0.7727277777 0.772727777 0.7727277777 0.7727277777 0.7727277777 0.772727777 0.772727777 0.7727277777 0.7727277777 0.7727277777 0.7727277777 0.7727277777 0.77272777777 0.77272777777 0.77272777777 0.77272777777 0.77272777777 0.772727777777 0.7727277777777 0.77272777777 0.772727777777 0.7727277777777777 0.77272777777777 0.7727277777777777777777777777777777777	18 0.411200093 0.9889993548 0.7857743491 0.521260739 0.7199813718 0.8515673264 0.8273862069 0.214953271 44.972508 0.87180838 0.8818888998388 0.8818888998388 0.8818888998388 0.8818888989888 0.8818888989888 0.8818888888 0.8818888888 0.8818888888 0.8818888888 0.8818888888 0.88188888888 0.88188888888 0.8818888888888 0.8818888888888 0.88188888888888 0.88188888888888888888 0.8818888888888888888888 0.881888888888888888888888888888888 0.8818888888888888888888888888888888888	19 0.3771391257 0.8152868748 0.8042501223 0.5156052963 0.7515450122 0.8569552324 0.8196721311 0.2115384615 47.70639682	20 0.4804834753 0.645019412 0.7952372148 0.5941188041 0.7221564461 0.8683179834 0.8467741935 0.3414634146 42.273077	21 0.4155649232 1.125394344 0.772481201 0.5179738562 0.7109961777 0.8339662242 0.8333833333 0.2026143791 44.344035	22 0.3922818386 0.7861403728 0.7927471232 0.548888889 0.7435223317 0.8419717147 0.853333333 0.24444444 41.505007	23 0.3545721196 1.007864833 0.8105224602 0.5035329716 0.758405311 0.8026394095 0.805555556 0.2015503876 44.570040	0.1	0.539410828 0	26 0.348909186 0.9849087 0.8150774184 0.49581643 0.7723303251 0.896245117 0.7931034483 0.1985294118 37.964178	27 0.6476445533 0.726942122 0.7592379277 0.5716961343 0.67604501 0.8424308455 0.8617021277 0.2816901408 44.330677	0.7806547905 0.4875283447 0.7205239486 0.8407856324 0.777777778 0.1972789116 46.4411477	29 0.5725491409 1.082484007 0.7611718285 0.5277777778 0.6811011113 0.8412425458 0.83333333 0.222222222 49.568199	30 0.40650845 0.9187273383 0.781116178 0.5113686364 0.7348819294 0.8273504266 0.8181818182 0.2045454545 46,75437	31 0.4601310626 0.9692454994 0.7385775848 0.461038961 0.6870277747 0.790127465 0.7272727273 0.1948051948 203.581502
tr.bpv v	0.7775366441 (6302018089	6525856733	7806354092	7454814101	8093484601	8095194155 (665157712	6767141946	7689340434	7887907437	5624401984	8122375555	7931970401	551353016	7669055501	7672856253	758465006	7857743491	8042501223	7952372148	772481201	7927471232	8105224602	650315709	0.7137559115 0.539410828	8159774184	7592379277	7806547905	7611718285	781116178	7385775848
v_loss tr		0.5327957324 0.9982410669 0.6302018089 0.1	0.5217230046 0.9616182446 0.6525856733 0.1	7.7670581341 0.	0.9277585149 0.	3.8190974593 0.	0.367757046 0.9465242624 0.8095194155 0.5	0.5150528796 0.9168871045 0.665157712	0.5085870743 0.9944771528 0.6767141946 0.1	.065403233 0.	3.8744389415 0.	0.5703222262 0.9465276599 0.5624401984 0.1	.203875303 0.	0.5644137859 0.	14 0.5765861149 0.9432210326 0.551353016 0.1	.085533977 0.	0.9129543304 0.	.025083542 0.	0.9389993548 0.	0.8152868748 0.	0.645019412 0.	.125394344 0.	0.7961403728 0.	.007864833 0.	24 0.5212556239 0.9054829216 0.650315709	0.953974247 0.	0.9849087 0.	0.726942122 0.	.036997557 0.	.082484007 0.	0.9187273383 0.	1.9692454934 0.
tr_loss v	0.4407408592 0.999881804	5327957324 6	5217230046 6	4044851849 6	4509404558 6	3670129182 6	367757046 6	5150528796 6	5085870743 6	4809822473 1	404574254 6	5703222262 6	3540434386 1	4089060434 6	5765864149 6	4225117842 1	4352161764 6	4530493624 1	411200393 6	3771391257 6	4804834753 6	4155649232 1	3922818386 (3545724196 1	5212556239 6	0.479270631 0	348909186 (6476445533 6	28 0.4139017356 1.036997557	5725494409 1	40650845 C	4601310626 6
Ħ	0 0	1 0	2 0.	3	4 0.	5 0	9	7	8	9 0.	10 0.	11 0	12 0.	13 0.	14 0,	15 0.	16 0.	17 0.	18 0.	19 0,	20 0.	21 0.	22 0.	23 0.	24 0.	25 0.	26 0.	27 0.	28 0.	29 0.	30	31 0.

Table B.9: ResNet50 Hyperparameter Tuning on Original Dataset: best results per trial using validation loss as metric.

config/wd	0.001510451824	0.001140190987	0.001978035796	0.003386165854	0.009949709149	0.00850181588	0.0086421597	0.001056055423	0.007654271044	0.008537335751	0.007435148118	0.002074727611	0.001008242634	0.006012991941	0.0008896224369	0.00576383982	0.003248522892	0.004451056956	0.0007651831475	0.006405663003	0.0003600208533	0.003790434832	0.0002516977864	0.001439345557	0.007384154559	0.002874425557	0.00309630442	0.007304670286	0.004578114407	0.009887470506	0.005096968715	0.0006421349858
config/lr	0.0002047653346 0.001510451824	0.0001927539469	0.0001261700242	0.00378248955	0.002114775439	0.001262158737	0.01588512397	0.000232623984	0.0003361220842	0.01853113801	0.005768671246	0.0001678445014	0.002083973911	0.01748139827	0.00015132965 0.0008896224369	0.0008461506477	0.007547896257	0.01015502109	0.00758861567	0.01435163916	0.05458950306 0.0003600208533	0.005348451252	0.003154670321	0.003747556755 0.00143934555	0.0002619940921	0.0001740546879	0.000676590819	0.05799739438	0.008760911056 0.004578114407	0.05638458363	0.001038212402	0.0003731782659 0.0006421349858
config/batch_size	256	256	128	256	512	64	512	256	256	64	64	512	128	128	512	128	512	512	64	256	256	256	128	128	256	- 64	128	128	256	128	19	128
time_since_restore iterations_since_restore config/batch_size	2	e	2	2	3	37	2	2	2	3	1	4	8	1	4	-	12	2	2	8	28	2	-	8	2	2	9	8	e	2	2	2
time since restore	81.78189349	129.9947054	81.6971698	87.70751333	142.6454418	1542.266177	97.99625754	95.09731174	90.95594645	131.9341464	45.08634496	190.574362	128.0503812	43.80413604	196,5455949	52.00335908	575.6671512	92.1040144	87.13111115	137.5992906	1224385283	91.21010041	41.50500727	126.2706916	93.56017303	88.13361239	289.1905541	133.7745454	138.4707918	85.68177748	117.6422112	423.4903898
node.ip	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20
hostname	dgx-node-0-0.codia.edu.ec	2621255 dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.oxdin.edu.ec	dgx-node-0-0.codia.edu.ec	dgx-node-0-0.oedia.edu.ec	dgx-node-0-0.codin.edu.ec	97.99625754 2621255 dgx-node-0-0.ccdia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.codia.edu.ec	dgx-node-0-0.oxdia.edu.ec	dgx-node-0-0.codia.edu.ec	2621255 dgx-node-0-0.cedia.edu.ec 201.159.221.20	128.0503812 2621255 dgx-node-0-0-ochiochioc 201.159.221.20	dgx-node-0-0.codia.edu.ec	dgx-node-0-0.codia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.codin.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.codia.edu.ec	dgx-node-0-0.oxdia.edu.ec	91.21010041 2621255 dgx-node-0-0.csdin.edu.ec 201.159.221.20	dgx-node-0-0.oxdia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.codia.edu.ec	dgx-node-0-0.codia.edu.ec	289.1905541 2621255 dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.codia.edu.ec	dgx-node-0-0.codia.edu.ec	dgx-node-0-0.oedia.edu.ec	2621255 dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.codia.edu.ec
timestamp time totals pid	1695053900 81.78189349 2621255 dgx-node-0-0.cedia.edu.ec 201.159.221.20	129.9947054	81.6971693 2621255	2023-09-18_17-12-36 1695057156 87,70751333 2621255 dgx-node-0-0.cedia.cdn.ec 201.159.221.20	c92af_00004 2023-09-18_17-22-22 1695057742 142.6454418 2621255 dgx-node-0-0.cedin.edu.ec 201.159.221.30	c92af_00005 2021-09-18_17-48-06 1695059286 1542.266177 2621255 dgx-nocke-0-0.cedia.cdu.cc 201.159.221.30	507 97.99625754 2621255	1695060794 95.09731174 2621255	2028-09-18 18-15-32 1695060932 90.95594645 2621255 dgx-node-0-0.cedn.ce 201.159.221.30	c92af_00009 2021-09-18_18-18-27 1695061107 131.941464 2621255 dgx-node-0-0.cedia.cdu.ce 201.159.221.30	c92af_00010 2022-09-18_18_19-12 1695061152 45_08634496 2621255 dgx-nocke-0-0.cedia.edu.ec 201.159.221.30	190.574362	945 128.0503812 2621255	c92af_00013 2022-09-18_18-33-09 1695061989 43.80413604 2621255 dgx-nocke-0-0.cedin.edu.ec 201.159.221.30	c92af_00014 2022-09-18_18-39-59 1695062399 136.5455949 2621255 dgx-node-0-0.csdia.cdn.cc 201.159.221.30	1695065383 52.00835908 2621255 dgx-node-0-0.cedia.edu.ec 201.159.221.20	1695066053 575.6671512 2621255 dgx-node-0-0.cedia.edu.ec 201.159.221.20	92.1040144 2621255	c92af_00018 2025-09-18_19-44-37 1695066277 87.13111115 2621255 dgx-node-0-0.cedm.celn.ce 201.159.221.30	c92af.00019 2022-09-18-19-47-38 1695066458 137.3692906 2621255 dgx-nocke-0-0.cedia.ced.a.ce 201.159.221.30	2023-09-18_20-15-09 1695068109 1224.385283 2621255 dgx-node-0-0.oxdia.odu.oc 201.159.221.20	071 91.21010041 2621255	1695069158 41.50500727 2621255 dgx-node-0-0.cedia.edu.ec 201.159.221.20	c92af_00023 2025-09-18_20-36-12 1695069372 126.2706916 2621255 dgx-node_0-0.exdia.exdn.ex 201.159.221.30	c92af_00024 2022-09-18_20-37-46 1695069466 93.56017303 2621255 dgx-nock-0-0.cedia.edu.ec 201.159.221.30	1695069732 88.13361239 2621255 dgx-node-0-0.cedia.edu.ec 201.159.221.20	203 289.1905541 2621255	1695070338 133.7745454 2621255 dgx-node-0-0.cxdin.cdu.cc 201.159.221.20	c92af_00028 2022-09-18_20-54-39 1695070479 138_4707918 2621255 dgx-node-0-0.cedia.edn.ee 201.159.221.20	c92af_00029 2023-09-18_20-56-05 1695070565 85.68177748 2621255 dgx-nocke-0-0.cedin.eci 201.159.221.30	1695070733 117.6422112 2621255	1695071274 423.4903898 2621255 dgx-node-0-0.cedia.edu.ec 201.159.221.20
timestan		2-48 1695056568	4-10 1695056650	2-36 16950571	2-22 16950577	8-06 16950592	8-27 1695060507		5-32 16950608	8-27 16950611	9-12 16950611	3-48 16950614	2-25 1695061945	3-09 16950619	9-59 16950623	9-43 16950653		2-25 1695066145	4-37 16950662	7-38 1695066	5-09 16950681	1-11 1695069071	2-38 16950691	6-12 16950690	7-46 16950694		0-03 1695070203	2-18 16950700	4-39 1695070	6-05 1695070		7-54 16950713
date	2023-09-18_16-18-20	2023-09-18_17-02-48	2023-09-18_17-04-10		2023-09-18_17-5	2023-09-18_17-	2023-09-18_18-08-27	2023-09-18_18-13-14	2023-09-18_18-1	2023-09-18_18-1	2023-09-18_18-1	2023-09-18-18-23-48 1695061428	2023-09-18_18-32-25	2023-09-18_18-3	2023-09-18_18-3	2023-09-18_19-29-43	2023-09-18 19-40-53	2023-09-18_19-42-25	2023-09-18_19-4	2023-09-18_19-4	2023-09-18_20-1	2023-09-18_20-31-11	2023-09-18-20-32-38	2023-09-18_20-3	2023-09-18_20-3	2023-09-18_20-42-12	2023-09-18_20-50-03	2023-09-18-20-52-18	2023-09-18_20-7	2023-09-18_20-7	2023-09-18-20-58-53	2023-09-18_21-07-54
n trial.id	c92af_00000	c92af_00001	c92af_00002	c92af_00003	c92af_00004	c92af 00005	c92af_00006	c92af_00007	c92af_00008	c92af_00009	c92af_00010	c92af_00011	c92af_00012	c92af_00013	c92af_00014	c92af_00015	c92af_00016	c92af_00017	c92af_00018	c92af_00019	c92af_00020	c92af_00021	c92af_00022	c92af_00023	c92af_00024	c92af_00025	c92af_00026	c92af_00027	c92af_00028	c92af_00029	c92af_00030	c92af_00031
training iteration trial id	2	3	2	2	69	37	2	2	2	69	-	4	3	-	7	_	12	2	2	es	88	2	-	3	2	2	9	8	65	2	2	2
done	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE
time this iters should checkpoint	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TIME	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TIME	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
time this iters	40.44378686	42.85801196	40.28443623	44.74223804	44.98719239	42.51460218	48.36868906	48.91369435	44.77759075	43.23409653	45.08634496	47.80989361	43.13047719		53.81542087	52:00335908	46.13454008	45.49879098	42.15860	44.82791066		46.86600		40.77273393	47.4303031	43.66562986	36.76101017	41.59197092	0.2024539877 50.25948572	0.2012195122 41.77064204	70.88783932	219.9088876
vanpv	0.2037037037	0.20625	0.2037037037	0.2119205298	0.2052980132	0.2088607595	0.2	0.2049689441	0.2024539877	0.1987179487	0.2155172414	0.20625	0.2064516129	0.3043478261	0.20625	0.2038216561	0.2064516129	0.2071428571	0.2162162162	0.21875	0.2142857143	0.2025316456	0.244444444	0.2142857143	0.2049689441	0.2049689441	0.2088607595	0.2214765101	0.2024539877	0.2012195122	0.833333333 0.2074074074	0.7857142857 0.1986754967
vqq-v	1	1	1	9285714286	8571428571	-	8.0	1	-	777777778	8367346939	1	6.0	8403361345	1	0.875	6.0	0.84	9411764706	8648648649	1	8571428571	853333333	1	1	1	1	1	1	1	833333333	7857142857
tr_npv	0.8023113263	0.8202144056	0.8053372516	0.8594228206 0	0.8499519345 0	0.8520132942	0.853967827	0.8120561315	0.8170830301	0.8438128258 0	0.8438966517 0	0.8106778659	0.8654117683	0.8498158287 0	0.8048096196	0.8155119217	0.8674334266	0.857841594	0.8636015561 0	0.8578251822 0	0.8690660935	0.8586477818 0	0.8419717147 0	0.8683857727	0.8117631865	0.8267932055	0.8609953909	0.8431154148	0.8624794568	0.8416991106	0.851736533 0	0.83626353 0
tr-ppv	0.7527619618	0.7627404597 0.8202144056	0.7600154031	0.7674567516	0.7774978975	0.7676396663	0.7682862933 0.853967827	0.7621971885	0.7614663826	0.6935803856	0.7336848358	0.7591738594 0.8106778659	0.7656622868	0.7184203278	0.7630733829	0.7182991784	0.7714084744	0.7663397085	0.7337090812	0.7486808461	0.7207436248	0.7627208234	0.7435225317	0.7636981402	0.7588703566	0.761245442	0.7724322178	0.6686869561	0.759194868	0.6829457663	0.7605311138	0.7598290755
v_bpv	0.4407408592 0.999881804 0.7775366441 0.6018518519 0.7527619618 0.8023113263	0.603125	2 0.4373958743 0.9091505742 0.7826763273 0.6018518519 0.7600154031 0.8053372516	0.3575752306 1.143736482 0.8134397861 0.5702459792 0.7674567516 0.8164228206 0.9285714286 0.2119205298	4 0.3676904628 1.039849401 0.813724916 0.5312204352 0.7774978973 0.8499519345 0.8571428571 0.2052980132 44.9871	0.3671045568 1.156804919 0.8088264803 0.6044303797 0.7676396663 0.8520132942	0.5	1.008094192 0.78712666 0.602484472 0.7621971885 0.8120561315	8 0.4198865704 1.016808155 0.7892747063 0.6012269939 0.7614663826 0.8170830301	9 0.4977151072 1.411243916 0.768966057 0.4882478632 0.693580856 0.8438128288 0.777777778 0.1987179487 43.22409653	10 0.404574254 0.8744389415 0.7887907437 0.5261259676 0.7336848358 0.8488066517 0.8367546939 0.2155172414	0.603125	1.302020669 0.8155370275 0.5532258065 0.7656622868 0.8654117683	13 0.4353855658 0.5954728127 0.7841180783 0.5723419803 0.7184202278 0.8498158287 0.8403961345 0.3043478261	14 0.436262616 0.9489163756 0.7839415012 0.603125 0.7630733829 0.8048096196	1.085533977 0.7669055501 0.539410828 0.7182991784 0.8155119217	1.274319649 0.8194209505 0.5532258065 0.7714084744 0.8674334266	1.025324464 0.8120906512 0.5235714286 0.7663397085 0.857841594	18 0.3969767014 1.475003461 0.7986553186 0.5786963434 0.7337090812 0.8636015561 0.9411764706 0.2162162162	0.92972821 0.8032530142 0.5418074324 0.7486808461 0.8578251822 0.8648648649 0.21875	20 0.5274666653 2.516665936 0.7949048591 0.6071428571 0.7207436248 0.8690669935	21 0.3590640126 1.245127678 0.8106843026 0.5298372514 0.7627208234 0.8586477818 0.8571428571 0.2025316456	0.3922818386 0.7961403728 0.7927471232 0.548888889 0.7435223317 0.8419717147 0.853333333 0.244444444	23 0.3455470802 1.347769022 0.8160419565 0.6071428571 0.7636981402 0.8683857727	24 0.4292977138 1.004907846 0.7853167715 0.602484472 0.7588703566 0.8117631865	1.078464985 0.7940193237 0.602484472 0.761245442 0.8267932055	26 0.3469683109 1.231634378 0.8167138044 0.6044303797 0.7724322178 0.8609953909	0.6190669884 1.644363523 0.7559011855 0.610738255 0.6686869561 0.8431154148	28 0.3609203593 1.372754574 0.8108371624 0.6012269939 0.759194868 0.8624794568	29 0.5423451745 2.163249731 0.7623224385 0.6006097561 0.6829457663 0.8416991106	30 0.3693228402 0.9530773759 0.8061338234 0.5203703704 0.7605311138 0.851736533	31 0.3894930082 0.9995812774 0.7980463027 0.4921948912 0.7598290755
tr_bpv	0.7775366441	.035073996 0.7914774326	0.7826763273	0.8134397861	0.813724916	0.8098264803	1.23508656 0.8111270602	0.78712666	0.7892747063	0.7686966057	0.7887907437	0.7849258626	0.8155370275	0.7841180783	0.7839415012	0.7669055501	0.8194209505	0.8120906512	0.7986553186	0.8032530142	0.7949048591	0.8106843026	0.7927471232	0.8160419565	0.7853167715	0.7940193237	0.8167138044	0.7559011855	0.8108371624	0.7623224385	0.8061338234	0.7980463027
v_loss	0.999881804	_	0.9991505742	1.143736482	1.039849401	1.156804919			1.016898155	1.411243916	0.8744389415	0.9856032729	1.302020669	0.5954728127	0.9489163756				1.475603461	0.92972821	2.516665936	1.245127678	0.7961403728	1.347769022	1.004907846		1.231634378	1.644363523	1.372754574	2.163249731	0.9530773759	0.9995812774
tr_loss	0.4407408592	0.4136427058	0.4373958743	0.3575752306	0.3676901628	0.3671043568	0.3638113316	0.4283831962	0.4198865704	0.4977151072	0.404574254	11 0.4301776769 0.9856032729 0.7849258626	0.3465075004	0.4353855658	0.436262616	15 0.4225117842	16 0.3454713144	0.3599349584	0.3969767014	19 0.37932943	0.5274666653	0.3590640126		0.3455470802	0.4292977138	0.4044295437	0.3469683109	0.6190669884	0.3609203593	0.5423451745	0.3693228402	0.3894939082
Ĺ	0	-	5	60	7	10	9	1-	∞	6	10	=	1.2	13	7	12	16	14	18	10	20	21	22	23	24	22	56	27	28	50	8	31

Table B.10: ResNet50 Hyperparameter Tuning on Original Dataset: best results per trial using BPV as metric.

config/wd	0.009384354553	0.0009060375322	0.001494833556	0.00718497032	0.00510889679	0.003457431541	0.0009695053274	0.00909924585	0.002830014438	0.00191149391	0.002600000704	0.005557382235	0.00813777734	0.009643663561	0.004126489949	0.006339226341	0.001606588474	0.00342572095	0.002481549303	0.005002538198	0.002129117825	0.003113004261	0.008714659701	0.002760521473	0.002682478818	0.002352419654	0.001943863766	0.007327841193	0.006164784529	0.003525784484	0.00381535575	0.007621203956
rl/Siluco	0.003130050712	0.0001710673864 (0.002592438935	0.08252758289	0.003212016898	0.001057225127	0.0006086071445 (0.01024111548	0.0003985529465	0.01882583825	0.007150068007	0.0001793900563	0.003113427628	0.003824778902	0.04007224759	0.0007163141462	0.001572072356	0.09742549118	0.00112802594	0.0002377278672 0.005002538198	0.06822477006	0.005187498365	0.02960994494	0.008199559978	0.05848823932	0.02019677588	0.009309813073	0.07923634615	0.08928670772	-	H	0.01109896856
config/batch size	256	64	256	512	256			256	512	512		256	128		256	128	64	512	128	128	256		512	128	512	128	19	256	19	128		512
time since restore iterations since restore config/batch size	19	2	23	4	2	09	2	_		2	_	4	1	7	23	_	2	2	2	3	3	1	_	-	2	3	-	3	3	2	3	1
ime since restore	992.4959991	105,6664207	124.6404922	179.9535322	83.52409768	2515,110126	84.98949599	42.56730509	316.4088433	89.89830112	45.01878738	168.0542722	41.58632445	178.884155	126.982203	41.32743692	85.59259295	93.61065793	82.54106665	123.4215522	127.1233008	45.17783594	46.14945316	41.39049554	90.64143777	123.2103956	43.04032469	126.7951326	128.1994667	82.12927198	138.838876	45.53393316
di apou	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.30	201.159.221.20	201.159.221.20	201.159.221.30	201.159.221.20	201.159.221.20	201.159.221.30	201.159.221.20	201.159.221.30	201.159.221.20	201.159.221.20	201.159.221.30	201.159.221.30	201.159.221.20	201.159.221.20	201.159.221.30	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.30	201.159.221.20	201.159.221.20
hostname	2165215 dgx-node-0-0.cedia.edu.ec	2165215 dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	1695077573 179.9535322 2165215 dgx-node-0-0.cedia.edu.ec 201.159.221.20	e1dbd.00004 2023-09-18_22-55-47 1695077747 83.52409768 2165215 dgs-node-0-0.cedia.edu.ec 201.159.221.30	2165215 dgx-node-0-0.cedia.edu.ec 201.159.221.20	1695080557 84.98949599 2165215 dgx-node-0-0.cedia.edu.ec 201.159.221.20	2165215 dgx-node-0-0.cedia.edu.ec	1695081043 316.4088433 2165215 dgx-node-0-0.cedia.edu.ec 201.159.221.20	c1dbd,00009 2023-09-19.00-25-08 1695082988 89.89830112 2165215 dgx-node-0-0 cedia.edu.ee 201.159.221.20	1695083078 45.01878738 2165215 dgx-node-0-0.cedia.edu.ec	1695083475 168.0542722 2165215 dgx-node-0-0.cedia.edu.ec	41.58632445 2165215 dgx-node-0-0.cedia.edu.ec 201.159.221.20	c1dbd_00013 2023-09-19_01-18-30 1695086310 178.884155 2165215 dgscnode-0-0.cedia.edu.ee 201.159.221.30	c1dbd.00014 2023-09-19_01-22-09 1695086529 126.982203 2165215 dgs-node-0-0.cedia.edu.ec 201.159.221.30	1695086570 41.32743692 2165215 dgx-node-0-0.cedia.edu.ec	1695086738 85.59259295 2165215 dgx-node-0-0.cedia.edu.ec	2165215 dgx-node-0-0.cedia.edu.ec	1695087381 82.54106665 2165215 dgv-node-0-0.cedia.edu.ee 201.159.221.20	e.1dbd.00019 2023-09-19_01-39-06 1695087546 123.4215522 2165215 dgs-node-0-0.cedia.edu.ec 201.159.221.30	1695088042 127.1233008 2165215 dgx-node-0-0.cedia.edu.ec 201.159.221.20	1695088087 45.17783594 2165215 dgv-node-0-0.cedia.edu.ec 201.159.221.20	1695088224 46.14945316 2165215 dgx-node-0-0.cedia.edu.ec 201.159.221.20	c1dbd.00023 2023-09-19_01-52-35 1695083355 41.39049554 2165215 dgs-node-0-0.cedia.edu.ec 201.159.221.30	c.ldbd_00024 2023-09-19_01-55-28 1695088528 90.64143777 2165215 dgx-node-0-0.cedia.edu.ee 201.159.221.20	2165215 dgx-node-0-0.cedia.edu.ec	2165215 dgx-node-0-0.cedia.edu.ec	1695089085 126.7951326 2165215 dgx-node-0-0.cedia.edu.ec 201.159.221.20	c1dbd_00028 2023-09-19_02-06-54 1695089214 128.1994667 2165215 dgc-node-0-0-cedia.cel. 201.159.221.30	c1dbd_00029 2023-09-19_02-08-16 1695089296 82,12927198 2165215 dgx-node-0-0.cedia.edu.ce 201.159.221.30	c1dbd,00030 2023-09-19_02-13-18 1695089598 138.838876 2165215 dgc-node-0-0-cedia.ee 201.159.221.30	1695089782 45.53393316 2165215 dgx-node-0-0.cedia.edu.ec 201.159.221.20
bid	2165215 dg	2165215 ds	2165215 dg	2165215 dg	2165215 ds	2165215 dg	2165215 dg	2165215 ds	2165215 dg	2165215 dg	2165215 ds	2165215 dg	2165215 dg	2165215 dg	2165215 dg	2165215 ds	2165215 dg	2165215 dg	2165215 dg	2165215 ds	2165215 dg	2165215 dg	2165215 ds	2165215 dg	2165215 dg	2165215 ds	2165215 dg	2165215 dg	2165215 ds	2165215 dg	2165215 dg	2165215 dg
timestamp time total s pid	1695072517 992.4959991	105.6664207	124.6404922	179.9535322	83.52409768	1695080305 2515.110126	84.98949599	1695080642 42,56730509	316.4088433	89.89830112	45.01878738	168.0542722	41.58632445	178.884155	126.982203	41.32743692	85.59259295	1695087252 93,61065793	82.54106665	123.4215522	127.1233008	45.17783594	46.14945316	41.39049554	90.64143777	123,2103956	1695088874 43,04032469	126.7951326	128.1994667	82.12927198	138.838876	45.53393316
timestamp	1695072517	1695074997	1695077393	1695077573	1695077747	1695080305	1695080557	1695080642	1695081043	1695082988	82083028	1695083475	1695086048	1695086310	1695086529	025086570	1695086738	1695087252	188238291	1695087546	1695088042	1695088087	1695088224	1695088355	1695088528	169508831	169508874	1695089085	1695089214	1695080296	8626802691	1695089782
date	2023-09-18_21-28-37	2023-09-18_22-09-57	2023-09-18_22-49-53	cldbd,00003 2023-09-18, 22-52-53	023-09-18_22-55-47		cldbd_00006 2023-09-18_23-42-37	2023-09-18_23-44-02	c1dbd 00008 2023-09-18-23-50-43	023-09-19_00-23-08	cldbd 00010 2023-09-19 00-24-38	c1dbd 00011 2023-09-19 00-31-15	c1dbd_00012 2023-09-19_01-14-08	023-09-19_01-18-30	023-09-19_01-22-09	c1dbd_00015 2023-09-19_01-22-50	c1dbd 00016 2023-09-19 01-25-38	2023-09-19_01-34-12	cldbd_00018 2023-09-19_01-36-21	023-09-19_01-39-06	c1dbd 00020 2023-09-19-01-47-22	c1dbd_00021 2023-09-19_01-48-07	2023-09-19_01-50-24	023-09-19_01-52-35	023-09-19_01-55-28	2023-09-19_02-00-31	2023-09-19_02-01-14	2023-09-19_02-04-45	023-09-19 02-06-54	023-09-19_02-08-16	023-09-19_02-13-18	cldbd 00031 2023-09-19 02-16-22
trial id	cldbd_00000 2	c1dbd_00001 2	cldbd_00002 2	c1dbd_00003 2	c1dbd_00004 2	cldbd_00005 2	c1dbd_00006 2	c1dbd_00007 2	cldbd,00008 2	c1dbd_00009 2	c1dbd_00010 2	cldbd_00011 2	cldbd_00012 2	c1dbd_00013 2	cldbd_00014 2	cldbd_00015 2	c1dbd_00016 2	cldbd_00017 2	c1dbd_00018 2	c1dbd_00019 2	cldbd,00020 2	cldbd_00021 2	c1dbd_00022 2	cldbd_00023 2	c1dbd_00024 2	c1dbd_00025 2	cldbd_00026 2	c1dbd_00027 2	c1dbd_00028 2	cldbd,00029 2	c1dbd_00030 2	c1dbd 00031 2
done training iteration trial id	19	2	23	4	2	09	2	-	-1	2	-	4	-	4	23	_	2	2	2	8	65	-	-	_	2	8	_	89	8	2	3	1
	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE
time this iter s should checkpoint	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
time this iter s	53.02419806	52.13958406	41.69477057	46.28601098	41.52280593	41.81021619		42.56730509	44.11644244	44.13120604	0.7045454545 45.01878738	42,09741		44.6363	42,48877406	41.32748	42.77554	46.57350063	40.67666721	41,00081015		45.1778	46.14945316	41.39049554	46.07275414	40.58716393	43,04032469	42.40256	42.35511661	40.94844389	46.20971775	45.53393316
v-npv	0.7820512821	0.6842105263	0.6274509804	0.875	8.0	0.6989247312	0.6442307692	0.7671232877	0.6875	0.7052631579	0.7045454545	0.7021276596	0.776119403	0.7407407407	0.6470588235	0.6767676768	0.7530864198	0.5538461538	0.6947368421	0.6666666667	0.8571428571	0.688172043	0.6052631579	0.6388888889	0.6829268293	0.5982905983	0.6956521739	0.7826086957	0.6451612903	0.7605633803	0.7878787879	0.7096774194
vqq-v	0.7682926829	0.7692307692	0.724137931	0.6607142857	0.7333333333	0.776119403	0.7678571429	0.724137931	0.78125	8.0	0.75	0.78787879	0.6989247312	0.746835443	0.6666666667	0.7868852459	0.7594936709	0.7333333333	0.7846153846	0.7704918033	0.6576576577	0.7611940299	0.7608695652	0.7884615385	0.6923076923	0.7674418605	0.6483516484	0.5474452555	0.7014925373	0.7078651685	0.7021276596	0.7910447761
tr_npv	.7882108158	.7597581417	.7953422789	322383058	.7881612703	.7941798976	.7732671621	.7703318377		.7875639496	.744413059	3.762360696	0.767828516	.7911483861	.7660534396	.7497612853	.7865209058	.7758596679	.7868822188	.7736347608	.7659923615	.7573777029	724939064	.7622919886	.7830252645	.7722326286	.7630845354	.7570607091	.7269825123	.7631765604	.7899945386	.7542563679
tr.ppv	0.4649016859 0.557729125 0.776886608 0.7751719825 0.7645624001 0.7882108158 0.7682926829 0.7820512821 0.7820612821 0.7820	0.5699106263 0.7552458419 0.7267206478 0.7507335422 0.7597581417 0.7692307692 0.6842105263	0.6038373709 0.7837577104 0.6757944557 0.7721731418 0.7953422789	3 0.626567105 0.5528886914 0.761835558 0.7678571429 0.7496880536 0.773983058 0.6607142857 0.7496880536 0.773983058 0.6607142857 0.7496880536 0.773983058 0.6607142857 0.7496880536 0.773983058 0.6607142857 0.7496880536 0.773983058 0.6607142857 0.7496880536 0.773983058 0.6607142857 0.7496880536 0.773983058 0.7496880536 0.773983058 0.7496880536 0.7496880536 0.773983058 0.7496880536 0.773983058 0.7496880536 0.74968056 0.7496880536 0.7496880536 0.7496880536 0.7496880536 0.7496880536 0.7496880536 0.7496880536 0.7496880536 0.7496880536 0.7496880536 0.7496880536 0.7496880536 0.7496880536 0.7496880536 0.7496880536 0.7496880536 0.749688050 0.7496880	4 0.4632759493 0.5618635416 0.777291622 0.7666666667 0.7664219737 0.7881612703 0.7333333333	0.454806469 0.5565065145 0.7813173806 0.7375220671 0.7684548837 0.7941798976 0.776119403 0.6989247312	6 0.4821082796 0.5815806389 0.7657138065 0.706043956 0.7581604509 0.7732671621 0.7678571429 0.6442307692	0.4986674161 0.5755158663 0.7593172249 0.7456306094 0.7483026122 0.7703318377 0.724137931	0.4609190394 0.5738873482 0.7775841964 0.734375 0.7671200112 0.7880483816	9 0.4709929212 0.5556446314 0.7755735128 0.7526315789 0.7635830759 0.7875639496	0.5227284171 0.5631920695 0.7351531243 0.72727273 0.7258649428 0.744413059	11 0.4911968441 0.5763898492 0.7585710892 0.7450032237 0.7547814823 0.762360066 0.7878787879 0.7021276596	0.4981948383 0.5763033628 0.7587969442 0.7375220671 0.7497653724 0.767828516 0.6989247312 0.776119403	13 0.4621011701 0.5693436861 0.779049803 0.7437880919 0.7686615745 0.7911483861 0.746835443 0.7407407407	14 0.5494094354 0.604776144 0.67570770894 0.6568627451 0.7481007391 0.7660634396 0.666666667 0.647058235 42.48877 0.6447058236 0.6447058677 0.6447058677 0.6447058677 0.6447058677 0.6447058677 0.6447058677 0.6447058677 0.6447058677 0.6447058677 0.6447058677 0.6447058677 0.6447058677 0.6447058677 0.6447058677 0.6447058677 0.6447058677 0.6447058677 0.644705877 0.6447077	15 0.516256143 0.5781646967 0.7451485173 0.7318264613 0.7405357493 0.7497812853 0.7868852459 0.676767676	16 0.4618045824 0.5537469387 0.7751843572 0.7562900453 0.7638478085 0.7863209058 0.7594386709 0.7530864198	0.6171988387 0.7988038851 0.7634369729 0.6435897436 0.7510142779 0.7758596679 0.7333333333 0.5538461538	18 0.4615454339 0.5677378774 0.7761453975 0.7396761134 0.7654085763 0.786822188 0.7846153846 0.6947368421	19 0.4816080969 0.5812354684 0.7673185584 0.718579235 0.761002356 0.7736347608 0.7736343 0.6666666667 41,00081015 0.7736347608 0.7704918033 0.66666666667 41,00081015 0.7867666666666 0.786766666666 0.7867666666666 0.786766666666 0.7867666666666 0.786766666666 0.7867666666666 0.7867666666666 0.78676666666666 0.78676666666666 0.78676666666666 0.7867666666666666 0.78676666666666 0.786766666666666666666 0.78676666666666666666666666666666666666	20 0.6732380469 0.6319161654 0.7562216742 0.7574002574 0.7464509869 0.7659923615 0.6576576577 0.8571428571	21 0.5041546971 0.580237031 0.752165878 0.7246830364 0.746954053 0.7573777029 0.7611940299 0.688172043	22 0.6147186724 0.5923610926 0.7481411084 0.6830663616 0.7413431529 0.754939064 0.7608695652 0.6052631579	23 0.5082112106 0.5968908072 0.755155952 0.7136752137 0.7480198618 0.7622019886 0.7884615385 0.638888889 41.39049554	24 0.5593634462 0.5705282688 0.7695868131 0.6876172608 0.7561483618 0.7830252645 0.6923076923 0.6829208293 46.07275414	25 0.5548630579 0.6213840783 0.7623043025 0.6828662294 0.7523759764 0.7722326286 0.7674418605 0.5982905983	26 0.5271928752 0.5983988874 0.7538372637 0.6720019111 0.744589992 0.7630845354 0.6483516484 0.6956521739	0.6744740396 0.74789958 0.7465358545 0.6650269756 0.736011 0.7570607091 0.5474452555 0.7826086957 0.786011 0.74789679 0.74672555 0.7826086957 0.786011 0.747896979 0.74672575 0.7826086957 0.786011 0.74789679 0.74672575 0.7826086957 0.786011 0.74789679 0.74672575 0.7826086957 0.786011 0.74789679 0.74672575 0.7826086957 0.786011 0.74789679 0.786011 0.74789679 0.74672575 0.7826086957 0.786011 0.786011 0.74789679 0.74789679 0.786011 0.786	28 1.225679175 0.6911776642 0.7183093175 0.6733269138 0.7006361226 0.7269825123 0.7014925373 0.6451612903 42.35511661	29 0.5670856326 0.5618354082 0.7545355684 0.7342142744 0.7458945764 0.7631765604 0.7078651685 0.7605633803 40.94844	30 - 0.4644014494 = 0.5652540922 = 0.7790631925 = 0.7450632237 = 0.7681318465 = 0.789945386 = 0.7021276596 = 0.78787879 = 0.78787879	31 0.5233573439 0.565136075 0.7502161938 0.7503610677 0.7461760197 0.7542563679 0.7910447761 0.7096774194
v-bpv	0.7751719825	0.7267206478	0.6757944557	0.7678571429	0.7666666667	0.7375220671	0.706043956	0.7456306094	0.734375	0.7526315789	0.72727273	0.7450032237	0.7375220671	0.7437880919	0.6568627451	0.7318264613	0.7562900453	0.6435897436	0.7396761134	0.718579235	0.7574002574	0.7246830364	0.6830663616	0.7136752137	0.6876172608	0.6828662294	0.6720019111	0.6650269756	0.6733269138	0.7342142744	0.7450032237	0.7503610977
tr_bpv	0.776386608	0.7552458419	0.7837577104	0.7618355558	0.777291622	0.7813173906	0.7657138065	0.7593172249	0.7775841964	0.7755735128	0.7351531243	0.7585710892	0.7587969442	0.7799049803	0.7570770894	0.7451485173	0.7751843572	0.7634369729	0.7761453975	0.7673185584	0.7562216742	0.752165878	0.7481411084	0.7551559252	0.7695868131	0.7623043025	0.7538372637	0.7465358545	0.7183093175	0.7545355684	0.7790631925	0.7502161938
v_bess	0.557729125	0.5699105263	0.6038373709	0.5528886914	0.5618635416	0.5565065145	0.5815806389	0.5755158663	0.5738873482	0.5556446314	0.5631920695	0.5763898492	0.5763033628	0.5693436861	0.604776144	0.5781646967	0.5537469387	0.7988039851	0.5677378774	0.5812354684	0.6319161654	0.580237031	0.5923610926	0.5968908072	0.5705282688	0.6213849783	0.5983983874	0.744789958	0.6911779642	0.5618354082	0.5652540922	0.565136075
tr_loss	0.4649016859	0.4974797725	0.4481803309	0.626567105	0.4632759493	0.454806469	0.4821082796	0.4986674161	0.4609190394	0.4709929212	0.5227284171	0.4911968441	0.4981948383	0.4621011701	0.5494094354	0.516256143	0.4618045824	0.6171988387	0.4615454339	0.4816680969	0.6732930469	0.5041546971	0.6147186724	0.5082112106	0.5593634462	0.5548630579	0.5271928752	0.6744740396	1.225679175	0.5670856326	0.4644044494	0.5233573439
	0	-	2	3	4	ю	9	t	œ	6	10	=	12	13	77	12	16	14	18	10	20	21	22	23	24	22	56	27	58	53	30	31

Table B.11: ResNet50 Hyperparameter Tuning on Synthetic Dataset: best results per trial using validation loss as metric.

-V-	0.0	673864 0.0009060375322	438935 0.001494833556	758289 0.00718497032	0.003212016898 0.005108889679	0.001057225127 0.003457431541	0.0006086071445 0.0009695053274	.11548 0.00909924585	529465 0.002830014438	83825 0.00491149391	068007 0.002600000704	1900563 0.005557382235	427628 0.00813777734	778902 0.009643663561	24759 0.004126489949	1141462 0.006339226341	0.001572072356 0.001606588474	49118 0.00342572095	02594 0.002481549303	278672 0.005002538198	0.06822477006 0.002129117825	498365 0.003113004261	10702017010000	-	+	+		-	+			
L	0.003130050712	0.0001710673864	0.002592438935	0.08252758289	0.003212	0.001057	0.0006086	0.01024111548	0.0003985529465	0.01882583825	0.007150068007	0.0001793900563	0.003113427628	0.003824778902	0.04007224759	0.0007163141462	0.001572	0.09742549118	0.00112802594	0.0002377278672	0.06822/	0.005187498365	0.02960994494		0.008199559978	0.008199559978	0.008199	0.008199	0.008199559978 0.05848823932 0.02019677588 0.009309813073	0.008199559978 0.05848823932 0.02019677588 0.009309813073 0.07923634615	0.008199578978 0.05848823932 0.02019677588 0.009309813073 0.07923634615 0.08928670772	0.008199559978 0.00588823932 0.002019677588 0.0092804813073 0.00928670772 0.02738857786
4-4-U-gass	256	64	256	512	256	64	256	256	512	512	512	256	128	512	256	128	19	512	128	128	256	512	512		128	512	128 128	128 128	128 128 64	128 128 64 64	128 512 128 64 64 64	128 512 128 64 256 64 128
the company of the case of the case of the case of	19 256	6	3	4	2	42	1	2	_	2	1	4	2	4	2	1	2	-	2	3	3	1	2		1	- 7	6	- 77 62	→ ▽ €4 €5		- 4 63 65	
the same of the same of	992.4959991	464.4133132	124.6404922	179.9535322	83.52409768	1762.180256	42.78283668	84.83117604	316.4088433	89.89830112	45.01878738	168.0542722	82.97959399	178.884155	84.49342895	41.32743692	85.59259295	47.0371573	82.54106665	123.4215522	127.1233008	45.17783594	92.10551476	41.39049554		180.1937032	180.1937032	180.1937032	180.1937032 82.62323165 127.1327608	180.1937032 82.62323165 127.1327608 42.14145255	180.1937032 82.62323165 127.1327608 42.14145255 43.30345297	180.1937032 82.62323165 127.1327608 42.14145255 43.30345297 82.12927198
		201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20	201.159.221.20		201.159.221.20	201.159.221.30	201.159.221.30	201.159.221.30 201.159.221.30 201.159.221.30	201.159.221.20 201.159.221.20 201.159.221.20 201.159.221.20	201.159.221.30 201.159.221.30 201.159.221.30 201.159.221.30	201.159.221.30 201.159.221.30 201.159.221.30 201.159.221.30 201.159.221.30
The same where the same was	dgs-node-0-0-cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec	dgx-node-0-0.cedia.edu.ec 201.159.221.20	e1dbd.00004 2023-09-18_22-55-47 1695077747 83.52409768 2165215 dgs-node-0-0.cedia.edu.ec 201.159.221.30	c.id.bd.00005 2023-09-18_22-25-52 1695079552 1762.180256 2165215 dgc-node-0-0.cedia.ech.cc 201.159.221.30	dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec 201.159.221.20	c1dbd_00009 2023-09-19_00-22-08 1695082988 89.89830112 2165215 dgs-node-0-ocdia.edu.ee 201.159.221.30	c 1d bd 00010 2023-09-19_00-24-38 1695083078 45.01878738 2165215 dgs-node-0-0.cedia.ech.ec 201.159.221.30	1695083475 168.0542722 2165215 dgx-node-0-0.cedia.edu.ec 201.159.221.30	dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec 201.159.221.20	c1dbd.00014 2023-09-19_01-21-36 1695086486 84.49342895 2165215 dgs-node-0-0.cedia.edu.ec 201.159.221.30	e.ldbd,00015 2023-09-19_01-22-50 1695086570 41.32743692 2165215 dgs-node-0-0.cedia.ech.ec 201.159.221.30	dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec 201.159.221.20	1695087381 82.54106665 2165215 dgs-node-0-0.cedia.edu.ec 201.159.221.20	e.1dbd.00019 2023-09-19_01-39-06 1695087546 123.4215522 2165215 dgs-node-0-0.cedia.edu.ec 201.159.221.30	c.id.bd_00020 2023-09-19_01-47-22 1695088042 127.1233008 2165215 dgs-node-0-0.cedia.edu.ec 201.159.221.30	dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec 201.159.221.20	dgx-node-0-0.cedia.edu.ec	lox-norde-0-0 cedia edu ec	· Communication of the communi	1695088791 82.62323165 2165215 dgc-node-0-0.cedia.edu.cc 201.159.221.30	lgx-node-0-0.cedia.edu.ec	1695088791 82.62323165 2165215 dgs-node-0-0.cedia.edn.ec 201.159.221.30 1695088959 127.1327608 2165215 dgs-node-0-0.cedia.edn.ec 201.159.221.30	1695088791 82.62323165 2165215 dgs-nocle-Ot-ocedia-othree 201.159221.30 1695088059 127.1327008 2165215 dgs-nocle-Ot-ocedia-othree 201.159221.30 169508001 42.1414525 2165215 dgs-nocle-Ot-ocedia-othree 201.159221.30	dge-node-0-0 cedia.edu.ec dge-node-0-0 cedia.edu.ec dge-node-0-0 cedia.edu.ec dge-node-0-0 cedia.edu.ec	Edit-199052 (2012-0-19.0), 26.11 (1630-27) (1572) (1572) (1620-27) (1620-27) (1572) (1
	1695072517 992.4959991 2165215 .	464.4133132 2165215	2165215	1695077573 179.9535322 2165215	83.52409768 2165215	1762.180256 2165215	1695080515 42.78283668 2165215	2165215	2165215	89.89830112 2165215	45.01878738 2165215	168.0542722 2165215	1695086089 82.97959399 2165215	2165215	84.49342895 2165215	41.32743692 2165215	2165215	47.0371573 2165215	82.54106665 2165215	123.4215522 2165215	127.1233008 2165215	2165215	2165215	41.39049554 2165215	1695088617 180.1937032 2165215 dgx-node-0-0.cedia.edu.ec 201.159.221.20		82.62323165 2165215	82.62323165 2165215	82.62323165 2165215 1 127.1327608 2165215 1	82,62323165 2165215 1 127,1327608 2165215 4 42,14145255 2165215 4	82,62323165 2165215 - 127,1327608 2165215 - 42,14145255 2165215 - 43,30345297 2165215 - 1	82.62323165 2165215 127.1327608 2165215 42.14145255 2165215 43.30345297 2165215 82.12927198 2165215
	1695072517	1695075356	1695077393 124.6404922	1695077573	1695077747	1695079552		1695080684 84.83117604	1695081043 316.4088433	1695082988	1695083078			1695086310 178.884155	1695086486	0.29802691	1695086738 85.59259295	1695087205	1695087381	1695087546	1695088042	1695088087 45.17783594	1695088270 92.10551476	1695088355		1000000001	1030000131	109990001	1695080591 1695080591	1695088959	1695089059 127.1327608 1695089001 42.14145255 1695089129 43.30345297	1695089359 1695089001 1695080129 1695080296
	cldbd,00000 2023-09-18_21-28-37	2023-09-18_22-15-56	2023-09-18_22-49-53	2023-09-18_22-52-53	2023-09-18_22-55-47	2023-09-18_23-25-52	cldbd_00006 2023-09-18_23-41-55	2023-09-18_23-44-44	2023-09-18_23-50-43	2023-09-19_00-23-08	2023-09-19-00-24-38	cldbd 00011 2023-09-19 00-31-15	cldbd_00012 2023-09-19_01-14-49	2023-09-19_01-18-30	2023-09-19_01-21-36	2023-09-19_01-22-50	c1dbd 00016 2023-09-19-01-25-38	2023-09-19_01-33-25	2023-09-19_01-36-21	90-92-10-01-30-00	2023-09-19_01-47-22	2023-09-19_01-48-07	2023-09-19_01-51-10	2023-09-19_01-52-35	cldbd_00024 2023-09-19_01-56-57	c1dhd 00025 2023.00,19 01,50,51	TO BO TO TO OR OR	TO 00 TO-07 OO 0808	cldbd 00026 2023-09-19 02-02-39	2023-09-19_02-02-39 2023-09-19_02-03-21	2023-09-19-02-02-39 2023-09-19-02-03-21 2023-09-19-02-05-29	2023-09-19_02-02-39 2023-09-19_02-03-21 2023-09-19_02-05-29 2023-09-19_02-08-16
	c1dbd_00000	c1dbd_00001	c1dbd 00002	c1dpd700003	c1dbd_00004	c1dbd_00005	cldbd_00006	c1dbd_00007	c1dbd_00008	cldbd_0000_	c1dbd_00010	cldbd_00011	cldbd_00012	c1dbd_00013	cldbd_00014	c1dbd_00015	c1dbd_00016	cldbd_00017	cldbd,00018	c1dbd_00019	c1dbd_00020	c1dbd_00021	c1dbd_00022	c1dbd_00023	c1dbd_00024	c1dbd_00025			c1dbd_00026	c1dbd_00026	c1dbd,00026 c1dbd,00027 c1dbd,00028	c1dbd_00026 c1dbd_00027 c1dbd_00028 c1dbd_00029
the same of the case of the case of	19	6	3	4	2	42	-	2		2	1	+	2	7	2	1	2	-	2	3	3	-	2	-	7	2			3	3	1 1 3	3 1 2 3
	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE			TRUE	FALSE	FALSE	FALSE FALSE FALSE FALSE
	SOS TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	100000000000000000000000000000000000000		TRUE	TRUE	TRUE	TRUE
	53.02419806	52.45447254	41.69477057	46.28601098	41.52280593	41.80848765	42.78283668	42.26387095	44.11644244	44.13120604	45.01878738	42,09741926	41.39326954	44.6363976	42.2802043	41.32743692	42.77554039	47.0371573	40.67666721	41,00081015	42.41954923	45.17783594	45.9560616	41.39049554	44.96076751	41.20242095	40 00 ACO 900	10.000 at 00.00	42ARH0930B	42.14145255	42.14145255 43.30345297	42,00403303 42,14145255 43,30345297 40,94844389
		0.6956521739	0.6274509804	0.875	8.0	0.7529411765	0.6601941748	0.95	0.6875	0.7052631579	0.7045454545 45.018787	0.7021276596	0.7021276596	0.7407407407	0.5982905983	8929292929	0.7530864198	-	0.6947368421	0.6666666667	0.8571428571	0.688172043	0.9655172414	6888888890	0.95	0.9411764706	0.605030303.69	0.60503030369	WWW.	1	1 0.9655172414	0.9655172414 0.7605633803
	0.7682926829	0.7647058824	0.724137931	0.6607142857	0.7333333333	0.7866666667	0.7894736842	0.5642857143	0.78125		0.75	0.78787879	0.78787879	0.746835443	0.7674418605	0.7868852459	0.7594936709	0.5031446541	0.7846153846	0.7704918033	0.6576576577	0.7611940299	0.6030534351	0.7884615385	0.5642857143	0.5524475524	0.90/1079/109/0	0.0010700100	O'CAPIGIONISS	0.5031446541	0.5031446541	0.5031446541 0.6030534351 0.7078651685
****	0.557729125 0.776386608 0.7751719825 0.7645624001 0.7882108158 0.7682926829 0.7820512821	0.5772007704 0.7808368119 0.7301790281 0.769506972 0.7921666519 0.7647058824 0.6956521739	0.7953422789	0.5528886914 0.7618355558 0.7678571429 0.7496880536 0.773983058 0.6607142857	0.7881612703	0.7938557517	0.7373919831	0.7800278403	0.7880483816	0.7875639496	0.744413059	0.762360696	0.5806591511 0.7717404515 0.7450032237 0.7607570221 0.7827238809 0.7878787879 0.7021275596	0.5693436861 0.7799049803 0.7437880919 0.7686615745 0.7911483861 0.746835443 0.7407407407	0.7731078795	0.7497612853	0.5537469387 0.7751843572 0.7562900453 0.7638478085 0.7863209058 0.7594906709 0.7530864198	1.158835292 0.7280063334 0.751572327 0.707936872 0.7480757948 0.5031446541	0.7868822188	0.7736347608	0.7659923615	0.580237031 0.752165878 0.7246830384 0.746954053 0.7573777029 0.7611940299 0.688172043	0.61579144 0.7719659923 0.784285382 0.7602164828 0.7837155018 0.6030534351 0.9655172414	0.7622919886	0.7804334395	0.76708968	3331010220	333101F022V	0.1134131333	1.088027358 0.7323948069 0.751372327 0.7211557488 0.7436340451 0.5031446541	0.03027358 0.074905032 0.078906022 0.0730241023 0.0730340451 0.0680534651 0.0730346045 0.0730346045 0.0730346045 0.0730346045 0.0730346045 0.0730346045 0.0730346045 0.0730346045 0.0730346045 0.0730344634 0.073034876 0.0730347034703470347034703470347034703470	0.7436340451 0.7240041492 0.7631765604
***************************************	0.7645624001	0.769506972	0.7721731418	0.7496880536	0.7664219737	0.7694540837	0.7352671926	0.758005497	0.7671200112 0.7880483816	0.7635830759	0.7258649428	0.7547814823	0.7607570221	0.7686615745	0.7505989938	0.7405357493	0.7638478085	0.707936872	0.7654085763	0.761002356	0.7464509869	0.746954053	0.7602164828	0.7480198618	0.7600404176	0.7499730604	0.7555041000	A 7555041000	0.1303241023	0.7211557488	0.7211557488 0.7110764462	0.7211557488 0.7110764462 0.7458945764
	0.7751719825	0.7301790281	0.6757944557	0.7678571429	0.7866666667	0.7698039216	0.7248339295	0.7571428571	0.734375	0.7526315789	0.72727273	0.7450032237	0.7450032237	0.7437880919	0.6828662294	0.7318264613	0.7562900453	0.751572327	0.7396761134	0.718579235	0.7574002574	0.7246830364	0.7842853382	0.7136752137	0.7571428571	0.7468120115	0.73/40/60/0999	O TOTALOGOODS	U. MPERUNNAGO	0.751572327	0.751572327 0.7842853382	0.7542853382 0.7542853382 0.7542142744
-	0.776386608	0.7808368119	0.7837577104	0.7618355558	0.777291622	0.7816549177	0.7363295879	0.7690166686	0.5738873482 0.7775841964 0.734375	0.7755735128	0.7351531243	0.7585710892	0.7717404515	0.7799049803	0.7618534366	0.7451485173	0.7751843572	0.7280063334	0.7761453975	0.7673185584	0.7562216742	0.752165878	0.7719659923	0.7551559252	0.7702369285	0.7585313702	0.7644686509	0.7644686509	000000000000000000000000000000000000000	0.7323948969	0.7323948969	0.7323948969 0.7175402977 0.7545355684
	0.557729125		0.4481803309 0.6038373709 0.7837577104 0.6757944557 0.7721731418 0.7953122789	0.5528886914	0.4632759493 0.5618635416 0.777291622 0.766666667 0.7664219737 0.7881612703 0.7333333333	0.4525576086 0.5612998009 0.7816549177 0.7898039216 0.7694540837 0.7938557517 0.7968666667 0.7529411765 41.808487 0.7938677517 0.796666667 0.7529411765 41.808487 0.796766667 0.796666667 0.796966667 0.796766667 0.7967666667 0.79676667 0.796766667 0.796766667 0.79676667 0.79676667 0.79676667 0.79676667 0.7967667 0.7967667 0.7967667 0.7967677 0.796777 0.7967677 0.796777 0.796777 0.796777 0.796777 0.7967777 0.796	0.535516776 0.5866377354 0.7363295879 0.7248339295 0.7352671926 0.7373919831 0.7894736842 0.6601941748	0.4869905815 0.6383515596 0.7690166686 0.7571428571 0.758005497 0.7800278403 0.5642857143 0.780905815 0.7800278403 0.5642857143 0.780905815 0.7800278403 0.5642857143 0.780905815 0.7809078403 0.780907	0.5738873482	0.4709929212 0.5556446314 0.7755735128 0.7526315789 0.7635830759 0.7873639496	0.5631920695	0.4911968441 0.5763898492 0.7585710892 0.7450032237 0.7547814823 0.762360696 0.78787879 0.7021276596	0.5806591511	0.5693436861	0.5283026159 0.6095822453 0.7618534366 0.6828662294 0.7305989938 0.7731078795 0.7674418605 0.5982905983	0.516256143 0.5781646967 0.7451485173 0.7318264613 0.7405357493 0.7497612853 0.7868852459 0.6767676768 41.32743692	0.5537469387	1.158835292	0.4615454339 0.5677378774 0.7761453975 0.7396761134 0.7654085763 0.7868922188 0.7846153846 0.6947368421	0.5812354684	0.6319161654			0.5082112106 0.5968908072 0.7551559252 0.7136752137 0.7480198618 0.7622919886 0.7884615385 0.7884615385 0.7888618 0.7888618 0.788861818 0.788861818 0.7888618	0.7033413649	0.317533638 0.7335090041 0.7585313702 0.7468120115 0.7499730604 0.76708968 0.5524475524 0.9411764706 41.20242095 0.74733707 0.747379 0.7473797 0.7473797 0.7473797 0.7473797 0.7473797 0.7473797 0.7473797 0.7473797 0.7473797 0.7473797 0.7473797 0.7473797 0.747379 0.7473	0.5188250480 0.6218133473 0.7644686502 0.7349600038 0.775514699 0.7774131555 0.8048780488 0.6050430168	0.6318133473	OUTSTONE !	1.088027358	1.088027358 0.8777034879	1.01965560 1000027358 0.732398900 0.73157287 0.721157488 0.746630451 0.05014651 0.05012651724 0.2314163 0.77866026 0.687703857 0.731554584 0.741242748 0.741054054 0.751064040 0.705005485 0.705065890 0.933442
the lane	29	0.4514409789	0.4481803309	0.626567105	0.4632759493	0.4525576086	0.535516776	0.4869905815	0.4609190394	0.4709929212	10 0.5227284171 0.5631920695 0.7351531243 0.72727273 0.7258649428 0.7444413059	0.4911968441	12 0.4728653807	0.4621011701	0.5283026159	0.516256143	0.4618045824	1.822012047	18 0.4615454339	19 0.4816680969 0.5812354684 0.7673185584 0.718579235 0.761002356 0.7736347608 0.7704918033 0.6666666667 41.000810	20 0.6732930469 0.6319161654 0.7562216742 0.7574002574 0.7464509869 0.7659023615 0.6576576577 0.8571428571 42.41954923	0.5041546971	0.4955056686	0.5082112106	24 0.5521712357 0.7033413649 0.7702369285 0.7571428571 0.7600404176 0.780434395 0.5642857143	0.517533638	0.5100050400	0.51999594901	WOLCOSOTON .	1.019655616	1.019655616	1.019655616 1.206661026 0.5670856326

Table B.12: ResNet50 Hyperparameter Tuning on Synthetic Dataset: best results per trial using BPV as metric.

	AlexNet Mispronunciation Results: Original Dataset								
	Top-5 Best Trials with Validation Loss								
Trials	Epoch	Validation Loss	hn1	hn2	PPV	NPV	BPV		
21	33	0.386	4096	512	0.87	0.61	0.74		
11	22	0.391	2048	256	0.84	0.64	0.74		
9	11	0.393	512	2048	0.81	0.67	0.74		
0	43	0.398	512	1024	0.83	0.64	0.73		
6	47	0.403	1024	2048	0.84	0.67	0.75		
		Top-5 Best	Trials	with 1	BPV				
Trial	$egin{array}{c c c c c c c c c c c c c c c c c c c $								
16	1	0.907	4096	2048	0.80	1.00	0.90		
28	8	0.843	64	64	0.81	0.67	0.74		
6	10	0.831	1024	2048	0.83	0.78	0.81		
0	9	0.796	512	1024	0.82	0.55	0.69		
11	6	0.782	2048	256	0.81	0.67	0.74		

Table B.13: AlexNet Mispronunciation Results in Original Dataset

	AlexNet Mispronunciation Results:								
	Synthetic Dataset								
	Top-5 Best Trials with Validation Loss								
Trial	Epoch	$egin{array}{c} ext{Validation} \ ext{Loss} \end{array}$	hn1	hn2	PPV	NPV	BPV		
12	13	0.22	256	2048	0.83	1.00	0.92		
0	45	0.23	128	128	0.86	0.94	0.90		
13	13	0.23	2048	128	0.83	0.97	0.90		
8	3	0.23	4096	64	0.81	0.97	0.89		
7	4	0.23	1024	64	0.82	1.00	0.91		
		Top-5 Best	Trials	with	$\overline{\mathrm{BPV}}$				
Trial	$egin{array}{c c c c c c c c c c c c c c c c c c c $								
13	12	0.92	2048	128	0.83	1.00	0.92		
7	4	0.92	1024	64	0.82	1.00	0.91		
14	4	0.92	1024	64	0.82	1.00	0.91		
8	3	0.91	4096	64	0.81	0.97	0.89		
24	6	0.91	4096	128	0.87	0.97	0.92		

Table B.14: AlexNet Mispronunciation Results in Synthetic Dataset

	VGG19 Mispronunciation Results: Original Dataset								
	Top-5 Best Trials with Validation Loss								
Trial	Epoch	$egin{array}{c c c c c c c c c c c c c c c c c c c $							
29	3	0.42	1024	4096	0.83	0.64	0.73		
2	64	0.42	128	64	0.83	0.53	0.68		
20	4	0.43	1024	64	0.84	0.50	0.67		
19	6	0.43	256	2048	0.85	0.52	0.69		
10	6	0.43	8192	1024	0.84	0.50	0.67		
		Top-5 Best	Trials	with	$\overline{\mathrm{BPV}}$				
Trial	$egin{array}{ c c c c c c c c c c c c c c c c c c c$								
5	18	0.91	512	1024	0.81	0.40	0.60		
2	1	0.90	128	64	0.80	1.00	0.90		
8	7	0.90	32	128	0.81	0.50	0.65		
20	1	0.90	1024	64	0.80	1.00	0.90		
18	62	0.90	4096	1024	0.81	1.00	0.90		

Table B.15: VGG19 Mispronunciation Results in Original Dataset

	VGG19 Mispronunciation Results:								
	Synthetic Dataset								
	Top-5 Best Trials with Validation Loss								
Trial	Epoch	$egin{array}{c} ext{Validation} \ ext{Loss} \end{array}$	hn1	hn2	PPV	NPV	BPV		
25	59	0.21	16384	8192	0.83	0.94	0.88		
24	3	0.23	1024	128	0.82	0.89	0.85		
6	62	0.24	4096	1024	0.82	0.91	0.87		
18	12	0.24	1024	4096	0.87	0.83	0.85		
2	7	0.24	16384	256	0.83	0.87	0.85		
		Top-5 Bes	t Trials	with E	$\overline{\mathrm{BPV}}$				
Trial	Epoch	Validation	hn1	hn2	PPV	NPV	BPV		
IIIai		BPV	11111	111112	11 4	111	DI V		
2	3	0.92	16384	256	0.79	0.91	0.85		
25	2	0.92	16384	8192	0.80	1.00	0.90		
8	60	0.92	256	256	0.81	0.94	0.87		
11	1	0.92	512	8192	0.83	0.94	0.88		
17	16	0.92	4096	16384	0.80	0.89	0.84		

Table B.16: VGG19 Mispronunciation Results in Synthetic Dataset

F	ResNet50 Mispronunciation Results:								
	Original Dataset								
\mathbf{T}_{0}	Top-5 Best Trials with Validation Loss								
Trial	Epoch	Validation	PPV	NPV	$_{ m BPV}$				
11101	Бросп	Loss	11 1	111	DI V				
13	4	0.56	0.80	0.00	0.40				
20	20	0.65	0.80	0.00	0.40				
27	1	0.73	0.81	0.40	0.60				
3	1	0.77	0.80	0.00	0.40				
22	1	0.80	0.80	0.00	0.40				
	Top-	5 Best Trials	with I	3PV					
Trial	$\begin{array}{c cccc} Trial & Epoch & Validation \\ BPV & PPV & NPV & BPV \end{array}$								
27	3	0.61	0.80	0.00	0.40				
20	28	0.61	0.80	0.00	0.40				
23	3	0.61	0.80	0.00	0.40				
5	37	0.60	0.80	0.00	0.40				
26	6	0.60	0.80	0.00	0.40				

Table B.17: ResNet50 Mispronunciation Results in Original Dataset

ResNet50 Mispronunciation Results:								
Synthetic Dataset Top-5 Best Trials with Validation Loss								
Trial	Epoch	Validation Loss	PPV	NPV	BPV			
3	4	0.55	0.52	1.00	0.76			
16	2	0.55	0.61	1.00	0.80			
9	2	0.56	0.66	1.00	0.83			
5	60	0.56	0.68	1.00	0.84			
0	19	0.56	0.63	1.00	0.82			
	Top-	5 Best Trials	with I	BPV				
Trial	$\begin{array}{c cccc} \text{Trial} & \text{Epoch} & \begin{array}{c cccc} \text{Validation} & \text{PPV} & \text{NPV} & \text{BPV} \end{array}$							
22	2	0.78	0.65	1.00	0.82			
28	1	0.78	0.51	1.00	0.76			
0	19	0.78	0.63	1.00	0.82			
5	42	0.77	0.66	1.00	0.83			
3	4	0.77	0.52	1.00	0.76			

Table B.18: ResNet50 Mispronunciation Results in Synthetic Dataset

Appendix C

Exemplary Code

This appendix is devoted to present the code used in the graduation project. In the case of the CNN Features-based model, feature extraction for all models is shown. However, in the case of the CNN Transfer Learning-based model, part of the code shown for AlexNet is already identical to the other models, but with some modifications. Therefore, we only show the CNN Transfer Learning-based model using AlexNet architecture.

C.1 Dataset Splitting

C.2 Audio Dataset Augmentation, Resampling, and Conversion

```
1 from glob import glob
2 from audiomentations import Compose, AddGaussianNoise, TimeStretch,
     PitchShift, Shift, AdjustDuration, TimeMask
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import torchaudio
6 import os
7 import shutil
8 import torch
9 import librosa
11 default_sample_rate = 44100
12 augment_size = 100
14 #Define augmentation parameters
15 audio_augmentation = Compose([
      AdjustDuration(duration_seconds=3, p=1.0),
      Shift(min_shift=-2, max_shift=2, p=1.0, shift_unit="seconds",
     rollover=True),
      PitchShift(min_semitones=-4, max_semitones=4, p=0.5),
18
      TimeStretch(min_rate=0.5, max_rate=1.5, p=0.5),
      AddGaussianNoise(min_amplitude=0.001, max_amplitude=0.015, p=0.5),
      TimeMask(min_band_part=0.05, max_band_part=0.15, fade=True, p=0.5)
22 ])
24 #Given a training folder of audios, augment them and convert them into
      spectrograms
25 def spectrogramAugmentation(path, save):
      #Define spectrogram function
      toSpectrogram = torchaudio.transforms.Spectrogram(n_fft=512)
27
28
      #Create a list of .ogg and .wav audio files to convert
      audio_files = glob(os.path.join(path, "*.ogg")) + glob(os.path.
     join(path, "*.wav"))
      for audio in audio files:
32
          #Extract audio name for spectrogram image name
33
          spectogram_name = (audio.split("/")[-1]).split(".")[0]
35
```

```
#Read audio file (normalization takes place by default)
36
          waveform, sample_rate = torchaudio.load(audio)
37
38
          #Define resampling audio function and resample
          resamplingAudio = torchaudio.transforms.Resample(sample_rate,
40
     default_sample_rate, dtype=waveform.dtype)
          waveform = resamplingAudio(waveform)
41
42
          #Save augmentations in .png format (100 augmentations for each
43
      audio file)
          for i in range(1, augment_size + 1):
44
              #Augment audio file
45
              aug = audio_augmentation(samples = np.asarray(waveform),
46
     sample rate=default sample rate)
47
              #Transform raw audios into spectrograms
              audio spec = toSpectrogram(torch.from numpy(np.asarray(aug
49
     )))
50
              #Save spectrograms in save folder
              plt.imsave(os.path.join(save, spectogram_name + "_aug_"+
     str(i)) + ".png", librosa.power_to_db(audio_spec[0]))
54
55 #Given a validation and testing folder of audios, convert them into
     spectrograms
56 def spectrogram(path, save):
      #Define spectrogram function
      toSpectrogram = torchaudio.transforms.Spectrogram(n_fft=512)
      #Create a list of .ogg and .wav audio files to convert
      audio_files = glob(os.path.join(path, "*.ogg")) + glob(os.path.
     join(path, "*.wav"))
      for audio in audio_files:
          #Extract audio name for spectrogram image name
          spectogram_name = (audio.split("/")[-1]).split(".")[0]
          #Read audio file (normalization takes place by default)
67
          waveform, sample_rate = torchaudio.load(audio)
68
          #Define resampling audio function and resample
70
          resamplingAudio = torchaudio.transforms.Resample(sample_rate,
71
     default_sample_rate, dtype=waveform.dtype)
```

```
waveform = resamplingAudio(waveform)
72
73
          #Make all audios have a fixed duration of 3 seconds
74
           adjust_duration = AdjustDuration(duration_seconds=3, p=1.0)
           waveform = adjust_duration(np.asarray(waveform), sample_rate=
76
      default_sample_rate)
          #Transform raw audios into spectrograms
78
           audio_spec = toSpectrogram(torch.from_numpy(np.asarray(
     waveform)))
80
          #Save spectrogram in .png format
81
          plt.imsave(os.path.join(save, spectogram_name) + ".png",
      librosa.power to db(audio spec[0]))
83
84 #-----
85 # DATASET CONVERSION
88 #Convert training data with 0 and 1 label
89 spectrogramAugmentation("/home/ricardo.velasco/TT_kichwa/
      transfer_learning_model/split_ds/train/0",
                          "/home/ricardo.velasco/TT_kichwa/
     transfer_learning_model/spec/train/0")
91
93 spectrogramAugmentation("/home/ricardo.velasco/TT_kichwa/
     transfer_learning_model/split_ds/train/1",
                          "/home/ricardo.velasco/TT_kichwa/
     transfer_learning_model/spec/train/1")
96 #Convert validation data with 0 and 1 label
97 spectrogram("/home/ricardo.velasco/TT kichwa/transfer learning model/
     split_ds/val/0",
                          "/home/ricardo.velasco/TT_kichwa/
     transfer_learning_model/spec/val/0")
100 spectrogram("/home/ricardo.velasco/TT_kichwa/transfer_learning_model/
     split_ds/val/1",
                          "/home/ricardo.velasco/TT_kichwa/
     transfer_learning_model/spec/val/1")
103 #Convert testing data with 0 and 1 label
104 spectrogram ("/home/ricardo.velasco/TT_kichwa/transfer_learning_model/
```

```
split_ds/test/0",
                          "/home/ricardo.velasco/TT kichwa/
105
      transfer_learning_model/spec/test/0")
106
107 spectrogram ("/home/ricardo.velasco/TT_kichwa/transfer_learning_model/
      split_ds/test/1",
                          "/home/ricardo.velasco/TT_kichwa/
108
      transfer_learning_model/spec/test/1")
100
111 #-----
112 # SYNTETIC DATASET CONVERSION
113 #-----
#Convert training data with 0 and 1 label
116 spectrogramAugmentation("/home/ricardo.velasco/TT_kichwa/
      transfer learning model/split syn ds/train/0",
                          "/home/ricardo.velasco/TT kichwa/
117
     transfer_learning_model/syn_spec/train/0")
118
120 spectrogramAugmentation("/home/ricardo.velasco/TT_kichwa/
      transfer_learning_model/split_syn_ds/train/1",
                          "/home/ricardo.velasco/TT_kichwa/
      transfer_learning_model/syn_spec/train/1")
123 #Convert validation data with 0 and 1 label
124 spectrogram("/home/ricardo.velasco/TT_kichwa/transfer_learning_model/
     split_syn_ds/val/0",
                          "/home/ricardo.velasco/TT kichwa/
125
     transfer_learning_model/syn_spec/val/0")
127 spectrogram("/home/ricardo.velasco/TT kichwa/transfer learning model/
     split_syn_ds/val/1",
                          "/home/ricardo.velasco/TT_kichwa/
128
     transfer_learning_model/syn_spec/val/1")
130 #Convert testing data with 0 and 1 label
131 spectrogram ("/home/ricardo.velasco/TT_kichwa/transfer_learning_model/
     split_syn_ds/test/0",
                          "/home/ricardo.velasco/TT_kichwa/
      transfer_learning_model/syn_spec/test/0")
133
134 spectrogram ("/home/ricardo.velasco/TT_kichwa/transfer_learning_model/
```

```
split_syn_ds/test/1",

"/home/ricardo.velasco/TT_kichwa/
transfer_learning_model/syn_spec/test/1")
```

C.3 CNN Features-based Model

```
1 from collections import OrderedDict
2 from glob import glob
3 from PIL import Image
4 import os
5 import torch
6 import torchvision
7 import numpy as np
8 import torchextractor
9 import itertools
10 import Orange
11 import matplotlib
12 import matplotlib.pyplot as plt
14 from torchvision.datasets import ImageFolder
15 from torch.utils.data import DataLoader
16 from torchvision.transforms import transforms
17 from sklearn.metrics import precision_score, make_scorer, log_loss
18 from sklearn.neighbors import KNeighborsClassifier
20 #Feature selection methods
21 from skfeature.function.statistical_based.chi_square import chi_square
     , feature_ranking
24 matplotlib.rcParams["figure.dpi"] = 500
26 \text{ num\_cores} = 64
27 device = "cuda" if torch.cuda.is_available() else "cpu"
29 #-----
30 #Alexnet Feature Extraction
31 #-----
32
33 def alexnetFeatureExtraction(path):
    #Define resampling function
resample = transforms.Compose([
```

```
transforms.Resize(size=(224, 224)),
          transforms.ToTensor(),
37
          transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229,
38
     0.224, 0.225]),
      ])
39
40
      # Array containing datasets using corresponding layers
      d = np.array([{"train":np.empty([0, 4096]), "val":np.empty([0,
42
     4096]), "test":np.empty([0, 4096])},
                    {"train":np.empty([0, 4096]), "val":np.empty([0,
     4096]), "test":np.empty([0, 4096])}])
44
     1 = {"train":np.array([]), "val":np.array([]), "test":np.array([])
     }
46
      #Target layers in order to extract features, select inference mode
      and send it to GPU
      layers = ["classifier.1", "classifier.4"]
48
      model = torchvision.models.alexnet(weights='AlexNet_Weights.
49
     IMAGENET1K_V1')
      model = torchextractor.Extractor(model, layers)
50
      model.eval()
      model.cuda()
      #Perform feature extraction by batches
      for i in ["train", "val", "test"]:
          data = ImageFolder(os.path.join(path, i), transform=resample)
56
          dataloader = DataLoader(dataset=data, batch_size=1024,
     num_workers=num_cores)
          for j, (inputs, labels) in enumerate(dataloader):
58
              with torch.no_grad():
                  inputs = inputs.to(device)
                  outputs, features = model(inputs)
                  f0 = features[layers[0]].detach().cpu().numpy()
                  f1 = features[layers[1]].detach().cpu().numpy()
64
                  d[0][i] = np.concatenate((d[0][i], f0))
                  d[1][i] = np.concatenate((d[1][i], f1))
                  1[i] = np.concatenate((1[i], labels.numpy()))
68
      return d, 1
70
71
```

```
73 #VGG19 Feature Extraction
76 def vggFeatureExtraction(path):
       #Define resampling function
77
       resample = transforms.Compose([
78
           transforms.Resize(size=(224, 224)),
           transforms.ToTensor(),
80
           transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229,
81
      0.224, 0.225]),
      ])
82
83
      # Array containing datasets using corresponding layers
      d = np.array([{"train":np.empty([0, 4096]), "val":np.empty([0,
85
      4096]), "test":np.empty([0, 4096])},
                     {"train":np.empty([0, 4096]), "val":np.empty([0,
      4096]), "test":np.empty([0, 4096])}])
87
      1 = {"train":np.array([]), "val":np.array([]), "test":np.array([])
      }
89
       #Target layers in order to extract features, select inference mode
       and send it to GPU
       layers = ["classifier.0", "classifier.3"]
91
       model = torchvision.models.vgg19(weights='VGG19_Weights.
      IMAGENET1K_V1')
      model = torchextractor.Extractor(model, layers)
93
       model.eval()
94
       model.cuda()
95
96
       #Perform feature extraction by batches
97
       for i in ["train", "val", "test"]:
98
           data = ImageFolder(os.path.join(path, i), transform=resample)
99
           dataloader = DataLoader(dataset=data, batch_size=1024,
100
      num_workers=num_cores)
           for j, (inputs, labels) in enumerate(dataloader):
               with torch.no_grad():
                   inputs = inputs.to(device)
                   outputs, features = model(inputs)
104
                   f0 = features[layers[0]].detach().cpu().numpy()
106
                   f1 = features[layers[1]].detach().cpu().numpy()
107
                   d[0][i] = np.concatenate((d[0][i], f0))
```

```
d[1][i] = np.concatenate((d[1][i], f1))
110
                   1[i] = np.concatenate((1[i], labels.numpy()))
111
      return d, 1
114
117 #VGG19 Feature Extraction
118 #-----
119
120 def resnetFeatureExtraction(path):
      #Define resampling function
121
      resample = transforms.Compose([
           transforms.Resize(size=(224, 224)),
          transforms.ToTensor(),
124
          transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229,
125
      0.224, 0.225]),
      1)
126
127
128
      # Array containing datasets using corresponding layers
      d = {"train":np.empty([0, 2048]), "val":np.empty([0, 2048]), "test]}
      ":np.empty([0, 2048])}
130
      1 = {"train":np.array([]), "val":np.array([]), "test":np.array([])
     }
132
      #Target layers in order to extract features, select inference mode
133
       and send it to GPU
      layers = ["avgpool"]
      model = torchvision.models.resnet50(weights="ResNet50_Weights.
      IMAGENET1K V2")
      model = torchextractor.Extractor(model, layers)
      model.eval()
137
      model.cuda()
138
      #Perform feature extraction by batches
140
      for i in ["train", "val", "test"]:
141
           data = ImageFolder(os.path.join(path, i), transform=resample)
           dataloader = DataLoader(dataset=data, batch_size=1024,
143
      num_workers=num_cores)
           for j, (inputs, labels) in enumerate(dataloader):
144
               with torch.no_grad():
145
                   inputs = inputs.to(device)
146
                   outputs, features = model(inputs)
```

```
148
                   f0 = features[layers[0]].detach().cpu().numpy()
149
                   d[i] = np.concatenate((d[i], np.squeeze(f0)))
                  1[i] = np.concatenate((1[i], labels.numpy()))
      return d, 1
156 #-----
157 # Discretization function
158 #-----
160 #Discretize continuous data into categorical data by method of Fayyad
      and Irani [FI93]. Returns the discretized dataset.
161 def discretize(X, y):
      domain = Orange.data.Domain.from_numpy(X, y)
      data = Orange.data.Table.from_numpy(domain, X, y)
164
      disc = Orange.preprocess.Discretize() #Define discretize function
165
      disc.method = Orange.preprocess.discretize.EntropyMDL() #Select
     desired method
167
      discrete_data = disc(data) #Discretize dataset
168
169
      return discrete_data.X
172 #Discretize the entire dataset and split the training, validation and
     testing sets again
173 def discretizeDataset(X_train, X_val, X_test, y_train, y_val, y_test):
      #Concatenate all sets to form a unique dataset for discretization
174
      X = np.concatenate((X_train, X_val, X_test))
      y = np.concatenate((y_train, y_val, y_test))
176
177
      #Discretize the dataset
178
      X_disc = discretize(X, y)
179
180
      #Retrieve discretized datasets from training, validation and
181
     testing sets
      X_train_disc = X_disc[:X_train.shape[0], :]
182
      X_val_disc = X_disc[X_train.shape[0]:X_train.shape[0] + X_val.
183
      shape[0], :]
      X_test_disc = X_disc[X_train.shape[0] + X_val.shape[0]:, :]
184
185
      return X_train_disc, X_val_disc, X_test_disc
```

```
187
188
189 def metric(y_true, y_pred):
       ppv = precision_score(y_true, y_pred)
       npv = precision_score(y_true, y_pred, pos_label=0)
191
       bpv = (ppv + npv) / 2
192
       return ppv, npv, bpv
194
195
196
197 def knnWithoutFeatureSelection(X_train, X_val, X_test, y_train, y_val,
       y_test, title):
198
       val_results = np.array([])
       val best = 0
199
       best_k = 0
200
201
       for k in range(1, 11):
202
           #Declare KNN Neighbours Classifier
203
           knn_classifier = KNeighborsClassifier(n_neighbors=k, n_jobs =
204
      num_cores)
205
           #Train the model with training set.
206
           knn_classifier.fit(X_train, y_train)
207
208
           #Obtain prediction using validation set and testing set
209
           y_pred_val = knn_classifier.predict(X_val)
210
           y_pred_test = knn_classifier.predict(X_test)
211
212
           #Obtain validation bpv
213
           val_bpv = metric(y_val, y_pred_val)[2]
214
215
           #Save best k with best bpv
216
           if val best <= val bpv:
217
               test_ppv, test_npv, test_bpv = metric(y_test, y_pred_test)
218
               val_best = val_bpv
219
               best_k = k
220
221
           val_results = np.append(val_results, val_bpv)
223
224
       print("PPV: ",test_ppv, "NPV:", test_npv, "BPV: ", test_bpv, "k: "
      , best_k, "val_bpv", val_best)
       plt.plot([i for i in range(1,11)], val_results, label="Validation
226
      BPV")
```

```
plt.xlabel("Number of Neighbours (k)")
227
       plt.ylabel("Balanced Predictive Value")
228
       plt.title(title)
229
       plt.legend()
       plt.show()
231
232
234 def knnChi(X_train, X_val, X_test, y_train, y_val, y_test, k ,title,
      last):
       #Create array to save results
235
       val_results = np.array([])
236
       val_best = 0
237
238
       best_number_features = 0
239
       #Discretize datasets
240
       X_train, X_val, X_test = discretizeDataset(X_train, X_val, X_test,
241
       y_train, y_val, y_test)
242
       #Get selected features by ranking
243
244
       ranking = feature_ranking(chi_square(X_train, y_train))
245
       for i in range (10, 401):
246
           #Declare KNN Neighbours Classifier
           knn_classifier = KNeighborsClassifier(n_neighbors=k, n_jobs =
248
      num_cores)
249
           #Best ith features
250
           best_features = ranking[:i]
251
252
           #Train the model with training set.
253
           knn_classifier.fit(X_train[:,best_features], y_train)
254
255
           #Obtain prediction using validation set and testing set
256
           y_pred_val = knn_classifier.predict(X_val[:,best_features])
257
           y_pred_test = knn_classifier.predict(X_test[:,best_features])
258
259
           #Obtain metrics for validation and testing set
260
           val_bpv = metric(y_val, y_pred_val)[2]
261
262
           #Save best k with best bpv
263
           if val_best <= val_bpv:</pre>
264
               test_ppv, test_npv, test_bpv = metric(y_test, y_pred_test)
265
               val_best = val_bpv
266
               best_number_features = i
```

```
268
           val_results = np.append(val_results, val_bpv)
269
270
271
      print("PPV: ",test_ppv, "NPV:", test_npv, "BPV: ", test_bpv, "
272
      num_features: ", best_number_features, "val_bpv", val_best)
      plt.plot([i for i in range(10,401)], val_results, label="
273
      Validation BPV with Feature Selection")
      plt.axhline(y = last, color = 'r', linestyle = '-', label = "
      Validation BPV without Feature Selection")
      plt.xlabel("Number of Features")
275
      plt.ylabel("Balanced Predictive Value")
276
      plt.title(title)
277
      plt.legend()
278
      plt.show()
279
280
281 #Obtain dataset matrix and labels for dataset
282 alexData, alexLabels = alexnetFeatureExtraction("/home/ricardo.velasco
      /TT_kichwa/transfer_learning_model/spec")
283 vggData, vggLabels = vggFeatureExtraction("/home/ricardo.velasco/
      TT_kichwa/transfer_learning_model/spec")
284 resData, resLabels = resnetFeatureExtraction("/home/ricardo.velasco/
      TT_kichwa/transfer_learning_model/spec")
285
286 #Obtain transformed dataset matrix and labels for syntetic dataset
287 synalexData, synalexLabels = alexnetFeatureExtraction("/home/ricardo.
      velasco/TT_kichwa/transfer_learning_model/syn_spec")
288 synvggData, synvggLabels = vggFeatureExtraction("/home/ricardo.velasco
      /TT_kichwa/transfer_learning_model/syn_spec")
289 synresData, synresLabels = resnetFeatureExtraction("/home/ricardo.
      velasco/TT_kichwa/transfer_learning_model/syn_spec")
291
292 #KNN HyperParameter on Original Dataset, plot graphics and test
      parameter with testing set
293
294 knnWithoutFeatureSelection(alexData[0]["train"], alexData[0]["val"],
      alexData[0]["test"], alexLabels["train"], alexLabels["val"],
      alexLabels["test"], "AlexNet Hyperparameter Tuning without Feature
      Selection in Original Dataset: FC1")
295 knnWithoutFeatureSelection(alexData[1]["train"], alexData[1]["val"],
      alexData[1]["test"], alexLabels["train"], alexLabels["val"],
      alexLabels["test"], "AlexNet Hyperparameter Tuning without Feature
      Selection in Original Dataset: FC2")
```

```
296 knnWithoutFeatureSelection(vggData[0]["train"], vggData[0]["val"],
      vggData[0]["test"], vggLabels["train"], vggLabels["val"], vggLabels
      ["test"], "VGG19 Hyperparameter Tuning without Feature Selection in
       Original Dataset: FC1")
297 knnWithoutFeatureSelection(vggData[1]["train"], vggData[1]["val"],
     vggData[1]["test"], vggLabels["train"], vggLabels["val"], vggLabels
      ["test"], "VGG19 Hyperparameter Tuning without Feature Selection in
      Original Dataset: FC2")
298 knnWithoutFeatureSelection(resData["train"], resData["val"], resData["
      test"], resLabels["train"], resLabels["val"], resLabels["test"], "
     ResNet50 Hyperparameter Tuning without Feature Selection in
     Original Dataset: AveragePool")
299
300 #KNN HyperParameter on Synthetic Dataset, plot graphics and test
     parameter with testing set
301 knnWithoutFeatureSelection(synalexData[0]["train"], synalexData[0]["
     val"], synalexData[0]["test"], synalexLabels["train"],
     synalexLabels["val"], synalexLabels["test"], "AlexNet
     Hyperparameter Tuning without Feature Selection in Synthetic
     Dataset: FC1")
302 knnWithoutFeatureSelection(synalexData[1]["train"], synalexData[1]["
     val"], synalexData[1]["test"], synalexLabels["train"],
      synalexLabels["val"], synalexLabels["test"], "AlexNet
     Hyperparameter Tuning without Feature Selection in Synthetic
     Dataset: FC2")
303 knnWithoutFeatureSelection(synvggData[0]["train"], synvggData[0]["val"
     ], synvggData[0]["test"], synvggLabels["train"], synvggLabels["val"
     ], synvggLabels["test"], "VGG19 Hyperparameter Tuning without
     Feature Selection in Synthetic Dataset: FC1")
304 knnWithoutFeatureSelection(synvggData[1]["train"], synvggData[1]["val"
     ], synvggData[1]["test"], synvggLabels["train"], synvggLabels["val"
     ], synvggLabels["test"], "VGG19 Hyperparameter Tuning without
     Feature Selection in Synthetic Dataset: FC2")
305 knnWithoutFeatureSelection(synresData["train"], synresData["val"],
      synresData["test"], synresLabels["train"], synresLabels["val"],
      synresLabels["test"], "ResNet50 Hyperparameter Tuning without
     Feature Selection in Synthetic Dataset: AveragePool")
307
308 #KNN Feature Selection in Original Dataset with Chi Score
309 knnChi(alexData[0]["train"], alexData[0]["val"], alexData[0]["test"],
     alexLabels["train"], alexLabels["val"], alexLabels["test"], 9 ,"
     AlexNet Hyperparameter Tuning with Chi-Square Feature Selection in
     Original Dataset: FC1", 0.6239935587761675)
```

```
310 knnChi(alexData[1]["train"], alexData[1]["val"], alexData[1]["test"],
      alexLabels["train"], alexLabels["val"], alexLabels["test"], 4 ,"
     AlexNet Hyperparameter Tuning with Chi-Square Feature Selection in
     Original Dataset: FC2", 0.5754750175932442)
311
s12 knnChi(vggData[0]["train"], vggData[0]["val"], vggData[0]["test"],
     vggLabels["train"], vggLabels["val"], vggLabels["test"], 9 ,"VGG19
     Hyperparameter Tuning with Chi-Square Feature Selection in Original
      Dataset: FC1", 0.6018518518518519)
sii knnChi(vggData[1]["train"], vggData[1]["val"], vggData[1]["test"],
      vggLabels["train"], vggLabels["val"], vggLabels["test"], 5 ,"VGG19
     Hyperparameter Tuning with Chi-Square Feature Selection in Original
      Dataset: FC2", 0.6323902027027027)
315 knnChi(resData["train"], resData["val"], resData["test"], resLabels["
      train"], resLabels["val"], resLabels["test"], 5 , "ResNet50
     Hyperparameter Tuning with Chi-Square Feature Selection in Original
      Dataset: AveragePool", 0.6414285714285715)
316
317 #KNN Feature Selection in Synthetic Dataset with Chi Score
318 knnChi(synalexData[0]["train"], synalexData[0]["val"], synalexData[0][
      "test"], synalexLabels["train"], synalexLabels["val"],
      synalexLabels["test"], 9 , "AlexNet Hyperparameter Tuning with Chi-
      Square Feature Selection in Synthetic Dataset: FC1", 0.9)
319 knnChi(synalexData[1]["train"], synalexData[1]["val"], synalexData[1][
      "test"], synalexLabels["train"], synalexLabels["val"],
     synalexLabels["test"], 9 , "AlexNet Hyperparameter Tuning with Chi-
     Square Feature Selection in Synthetic Dataset: FC2",
      0.8827450980392156)
321 knnChi(synvggData[0]["train"], synvggData[0]["val"], synvggData[0]["
     test"], synvggLabels["train"], synvggLabels["val"], synvggLabels["
     test"], 10 , "VGG19 Hyperparameter Tuning with Chi-Square Feature
     Selection in Synthetic Dataset: FC1", 0.8896920175989944)
322 knnChi(synvggData[1]["train"], synvggData[1]["val"], synvggData[1]["
     test"], synvggLabels["train"], synvggLabels["val"], synvggLabels["
     test"], 1 , "VGG19 Hyperparameter Tuning with Chi-Square Feature
     Selection in Synthetic Dataset: FC2", 0.8952941176470588)
```

C.4 CNN Transfer Learning-based AlexNet Hyperparameter Tuning

```
1 from functools import partial
2 import os
3 import torch
4 import torch.nn as nn
5 import torch.nn.functional as F
6 import torch.optim as optim
7 import torchvision
8 import torchvision.transforms as transforms
9 from torchvision.datasets import ImageFolder
10 from torch.utils.data import DataLoader
11 import ray
13 from ray import tune
14 from ray.air import session, RunConfig, CheckpointConfig, Checkpoint
15 from ray.tune.schedulers import ASHAScheduler
17 device = "cuda" if torch.cuda.is_available() else "cpu"
19 \text{ num\_cores} = 64
20 gpus_per_trial = 1
21 \text{ max\_num\_epochs} = 64
22 num_samples = 32 #Number of items in hyperparameter search
24 trainingPath = "/home/ricardo.velasco/TT_kichwa/
     transfer_learning_model/spec/train"
validationPath = "/home/ricardo.velasco/TT_kichwa/
     transfer_learning_model/spec/val"
26 testingPath = "/home/ricardo.velasco/TT_kichwa/transfer_learning_model
     /spec/test"
28 synTrainingPath = "/home/ricardo.velasco/TT_kichwa/
     transfer_learning_model/syn_spec/train"
29 synValidationPath = "/home/ricardo.velasco/TT kichwa/
     transfer_learning_model/syn_spec/val"
30 synTestingPath = "/home/ricardo.velasco/TT_kichwa/
     transfer_learning_model/syn_spec/test"
31
32 #Load training, testing and validation function for hyperparameter
     searching
33 def load_data(train, val):
      #Input resmampling
35
      transform = transforms.Compose([
36
          transforms.Resize(size=(224, 224)),
```

```
transforms.ToTensor(),
38
          transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229,
39
     0.224, 0.225]),
      ])
40
41
      trainset = ImageFolder(train, transform=transform)
42
      valset = ImageFolder(val, transform=transform)
44
45
      return trainset, valset
47
48
49 def metrics(outputs, labels):
      TP = ((outputs == 1) & (labels == 1)).sum().item()
      FP = ((outputs == 1) & (labels == 0)).sum().item()
      TN = ((outputs == 0) & (labels == 0)).sum().item()
      FN = ((outputs == 0) & (labels == 1)).sum().item()
      ppv = (TP / (TP + FP)) if (TP + FP) > 0 else 0
      npv = (TN / (TN + FN)) if (TN + FN) > 0 else 0
56
      return ppv, npv
59
60 def training(config, train=None, val=None):
      #Import Alexnet pretrained model
      net = torchvision.models.alexnet(weights='AlexNet_Weights.
     IMAGENET1K_V1')
      #Freeze features' layer for transfer learning
      for param in net.features.parameters():
          param.requires_grad = False
      #Reset and modify classification layer suitable for binary
68
     classification
      net.classifier = torch.nn.Sequential(
          torch.nn.Dropout(p=0.5, inplace=False),
70
          torch.nn.Linear(in_features=9216, out_features=config["hn1"],
     bias=True),
          torch.nn.ReLU(inplace=True),
72
          torch.nn.Dropout(p=0.5, inplace=False),
73
          torch.nn.Linear(in_features=config["hn1"], out_features=config
     ["hn2"], bias=True),
          torch.nn.ReLU(inplace=True),
75
          torch.nn.Linear(in_features=config["hn2"], out_features=2,
```

```
bias=True))
77
       #Send the model to GPU and put it in training mode
78
       net.to(device)
       net.train()
80
81
       #Run always on GPU, if there are more GPUs, allow parallel data
      parallel training
       if torch.cuda.device_count() > 1:
83
               net = nn.DataParallel(net)
85
       #Define Loss Function
86
       criterion = nn.CrossEntropyLoss()
88
       #Define optimizer and establish a traininable parameter for
89
      learning rate
       optimizer = optim.SGD(net.classifier.parameters(), lr=config["lr"
      ], momentum=0.9)
91
       # To restore a checkpoint, use 'session.get_checkpoint()'.
92
       checkpoint = session.get_checkpoint()
93
94
       if checkpoint:
           checkpoint_state = checkpoint.to_dict()
96
           start_epoch = checkpoint_state["epoch"]
           net.classifier.load_state_dict(checkpoint_state["
      classifier_state_dict"])
           optimizer.load_state_dict(checkpoint_state["
99
      optimizer_state_dict"])
       else:
100
           start_epoch = 0
103
       #Import training, validation and testing datasets
104
       trainset, valset= load_data(train, val)
106
       #Load training dataset
       trainloader = DataLoader(
           dataset=trainset,
           batch_size=int(config["batch_size"]),
           num_workers=num_cores,
           shuffle = True
112
       )
113
114
```

```
#Load validation dataset
115
       valloader = DataLoader(
116
           dataset=valset,
117
           batch_size=len(valset), #Due to hardware, pick largest batch
118
      size possible to compute metrics
           num_workers=num_cores,
119
           shuffle = True
120
       )
       for epoch in range(start_epoch, max_num_epochs):
125
           # TRAINING STEP
126
           # -----
           train_loss, train_ppv, train_npv = 0.0, 0.0, 0.0
128
           for i, (inputs, labels) in enumerate(trainloader):
129
               # get the inputs; data is a list of [inputs, labels]
130
               inputs, labels = inputs.to(device), labels.to(device)
131
132
133
               # zero the parameter gradients
               optimizer.zero_grad()
134
135
136
               # forward + backward + optimize
137
               outputs = net(inputs)
               loss = criterion(outputs, labels)
138
               loss.backward()
139
               optimizer.step()
140
141
               #Accumulate statistics across a single batch
142
               _, y = torch.max(outputs.data, 1)
143
               train_loss += loss.item()
144
               train_ppv += metrics(y, labels)[0]
145
               train_npv += metrics(y, labels)[1]
146
147
           #Compute average of metrics across all batches
148
           train_loss /= len(trainloader)
149
           train_ppv /= len(trainloader)
           train_npv /= len(trainloader)
           # VALIDATION STEP
           #-----
155
           val_loss, val_ppv, val_npv = 0.0, 0.0, 0.0
156
```

```
for i, (inputs, labels) in enumerate(valloader):
158
               with torch.no_grad():
                    inputs, labels = inputs.to(device), labels.to(device)
                    #Compute outputs
                    outputs = net(inputs)
163
                    _, y = torch.max(outputs.data, 1)
                    #Compute loss
166
                    loss = criterion(outputs, labels)
167
168
                    #Accumulate statistics across a single batch
169
                    val_loss+= loss.item()
170
                    val ppv += metrics(y, labels)[0]
                    val_npv += metrics(y, labels)[1]
172
           #Compute average of metrics across all batches
174
           val_loss /= len(valloader)
           val_ppv /= len(valloader)
176
           val_npv /=len(valloader)
178
           # Here we save a checkpoint. It is automatically registered
179
      with
           # Ray Tune and can be accessed through 'session.get_checkpoint
180
      () (
181
           # API in future iterations.
182
           #Checkpoint data
183
           checkpoint_data = {
184
               "epoch": epoch,
185
               "classifier_state_dict": net.classifier.state_dict(),
186
               "optimizer_state_dict": optimizer.state_dict()
187
           }
188
189
           checkpoint = Checkpoint.from_dict(checkpoint_data)
           session.report({"tr_loss":train_loss, "v_loss": val_loss,
191
                            "tr_bpv": (train_ppv + train_npv) / 2,
                            "v_bpv": (val_ppv + val_npv)/2,
                           "tr_ppv":train_ppv, "tr_npv":train_npv,
194
                           "v_ppv": val_ppv, "v_npv": val_npv,},
      checkpoint = checkpoint)
196
198 #Hyperparameter Tuning and Testing for Dataset
```

```
199 #-
200
201 config = {
       "lr": tune.loguniform(1e-4, 1e-1),
202
       "batch_size": tune.choice([64, 128, 256, 512]),
203
       "hn1": tune.choice([32, 64, 128, 256, 512, 1024, 2048, 4096]),
204
       "hn2": tune.choice([32, 64, 128, 256, 512, 1024, 2048, 4096])
205
206 }
207
208 scheduler = ASHAScheduler(
       max_t=max_num_epochs,
209
       grace_period=3,
210
211
       reduction_factor=2,)
212
213
214 tuner = tune.Tuner(
215
       tune.with resources(
           tune.with_parameters(training, train=trainingPath, val=
216
      validationPath),
           resources={"cpu": num_cores, "gpu": gpus_per_trial}
217
       ),
218
       tune_config=tune.TuneConfig(
219
           metric="v_loss",
220
221
           mode="min",
           scheduler=scheduler,
222
223
           num_samples=num_samples,
       ),
224
       run_config=RunConfig(
225
           checkpoint_config=CheckpointConfig(
226
                num_to_keep = None, #SAVE ALL CHECKPOINTS
227
           )
228
229
       ),
       param_space=config,
230
231
233 results = tuner.fit()
234
236 #Hyperparameter Tuning and Testing for Syntetic Dataset
238 scheduler2 = ASHAScheduler(
       max_t=max_num_epochs,
239
       grace_period=3,
240
       reduction_factor=2,)
```

```
242
243 tuner2 = tune.Tuner(
       tune.with resources(
244
           tune.with_parameters(training, train=synTrainingPath, val=
      synValidationPath),
           resources={"cpu": num_cores, "gpu": gpus_per_trial}
246
       ),
247
       tune_config=tune.TuneConfig(
248
           metric="v_loss",
249
           mode="min",
           scheduler=scheduler2,
251
           num_samples=num_samples,
252
253
       ),
       run_config=RunConfig(
254
           checkpoint_config=CheckpointConfig(
                num_to_keep = None #SAVE ALL CHECKPOINTS
257
       ),
258
       param_space=config,
260
261
262 results2 = tuner2.fit()
```

C.5 CNN Transfer Learning-based AlexNet Testing

```
1 from functools import partial
2 import os
3 import torch
4 import torch.nn as nn
5 import torch.nn.functional as F
6 import torch.optim as optim
7 import torchvision
8 import torchvision.transforms as transforms
9 from torchvision.datasets import ImageFolder
10 from torch.utils.data import DataLoader
11 import ray
12
13 from ray import tune
14 from ray.air import session, RunConfig, CheckpointConfig, Checkpoint
15 from ray.tune.schedulers import ASHAScheduler
16
17 device = "cuda" if torch.cuda.is_available() else "cpu"
```

```
19 num_cores = 64
20 gpus_per_trial = 2
21 \text{ max\_num\_epochs} = 64
22 num_samples = 64 #Number of items in hyperparameter search
24 trainingPath = "/home/ricardo.velasco/TT_kichwa/
     transfer_learning_model/spec/train"
25 validationPath = "/home/ricardo.velasco/TT_kichwa/
     transfer_learning_model/spec/val"
26 testingPath = "/home/ricardo.velasco/TT_kichwa/transfer_learning_model
     /spec/test"
28 synTrainingPath = "/home/ricardo.velasco/TT kichwa/
     transfer_learning_model/syn_spec/train"
29 synValidationPath = "/home/ricardo.velasco/TT_kichwa/
     transfer_learning_model/syn_spec/val"
30 synTestingPath = "/home/ricardo.velasco/TT_kichwa/
     transfer_learning_model/syn_spec/test"
32 #Load training, testing and validation function for hyperparameter
     searching
33 def load_data(train, val):
34
      #Input resmampling
35
      transform = transforms.Compose([
36
          transforms.Resize(size=(224, 224)),
37
          transforms.ToTensor(),
38
          transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229,
     0.224, 0.225]),
      1)
40
41
      trainset = ImageFolder(train, transform=transform)
42
43
      valset = ImageFolder(val, transform=transform)
44
45
      return trainset, valset
46
47
48
49 def metrics(outputs, labels):
      TP = ((outputs == 1) & (labels == 1)).sum().item()
      FP = ((outputs == 1) & (labels == 0)).sum().item()
      TN = ((outputs == 0) & (labels == 0)).sum().item()
      FN = ((outputs == 0) & (labels == 1)).sum().item()
```

```
54
      ppv = (TP / (TP + FP)) if (TP + FP) > 0 else 0
      npv = (TN / (TN + FN)) if (TN + FN) > 0 else 0
      return ppv, npv
58
60 def training(config, train=None, val=None):
      #Import Alexnet pretrained model
61
      net = torchvision.models.alexnet(weights='AlexNet_Weights.
     IMAGENET1K_V1')
      #Freeze features' layer for transfer learning
      for param in net.features.parameters():
          param.requires_grad = False
      #Reset and modify classification layer suitable for binary
     classification
      net.classifier = torch.nn.Sequential(
          torch.nn.Dropout(p=0.5, inplace=False),
          torch.nn.Linear(in_features=9216, out_features=config["hn1"],
71
     bias=True),
          torch.nn.ReLU(inplace=True),
          torch.nn.Dropout(p=0.5, inplace=False),
73
          torch.nn.Linear(in_features=config["hn1"], out_features=config
74
     ["hn2"], bias=True),
          torch.nn.ReLU(inplace=True),
75
          torch.nn.Linear(in_features=config["hn2"], out_features=2,
76
     bias=True))
      #Send the model to GPU and put it in training mode
78
      net.to(device)
79
      net.train()
81
      #Run always on GPU, if there are more GPUs, allow parallel data
82
     parallel training
      if torch.cuda.device_count() > 1:
83
              net = nn.DataParallel(net)
84
      #Define Loss Function
86
      criterion = nn.CrossEntropyLoss()
87
      #Define optimizer and establish a traininable parameter for
89
     learning rate
      optimizer = optim.SGD(net.classifier.parameters(), lr=config["lr"
```

```
], momentum=0.9)
91
       # To restore a checkpoint, use 'session.get_checkpoint()'.
92
       checkpoint = session.get_checkpoint()
94
       if checkpoint:
95
           checkpoint_state = checkpoint.to_dict()
           start_epoch = checkpoint_state["epoch"]
97
           net.classifier.load_state_dict(checkpoint_state["
98
      classifier_state_dict"])
           optimizer.load_state_dict(checkpoint_state["
99
      optimizer_state_dict"])
100
       else:
           start epoch = 0
103
       #Import training, validation and testing datasets
104
       trainset, valset= load_data(train, val)
106
       #Load training dataset
       trainloader = DataLoader(
108
           dataset=trainset,
109
           batch_size=int(config["batch_size"]),
111
           num_workers=num_cores,
           shuffle = True
      )
113
114
       #Load validation dataset
      valloader = DataLoader(
           dataset=valset,
117
           batch_size=len(valset), #Due to hardware, pick largest batch
118
      size possible to compute metrics
           num_workers=num_cores,
119
           shuffle = True
120
      )
       for epoch in range(start_epoch, max_num_epochs):
           # -----
           # TRAINING STEP
126
           # -----
           train_loss, train_ppv, train_npv = 0.0, 0.0, 0.0
128
           for i, (inputs, labels) in enumerate(trainloader):
               # get the inputs; data is a list of [inputs, labels]
130
```

```
inputs, labels = inputs.to(device), labels.to(device)
131
132
               # zero the parameter gradients
133
               optimizer.zero_grad()
               # forward + backward + optimize
136
               outputs = net(inputs)
               loss = criterion(outputs, labels)
138
               loss.backward()
139
               optimizer.step()
140
141
               #Accumulate statistics across a single batch
142
143
               _, y = torch.max(outputs.data, 1)
               train loss += loss.item()
144
               train_ppv += metrics(y, labels)[0]
145
               train_npv += metrics(y, labels)[1]
146
147
           #Compute average of metrics across all batches
148
           train_loss /= len(trainloader)
149
           train_ppv /= len(trainloader)
150
           train_npv /= len(trainloader)
154
           # VALIDATION STEP
           #-----
           val_loss, val_ppv, val_npv = 0.0, 0.0, 0.0
156
157
           for i, (inputs, labels) in enumerate(valloader):
158
               with torch.no_grad():
                    inputs, labels = inputs.to(device), labels.to(device)
160
161
                   #Compute outputs
                    outputs = net(inputs)
163
                    _, y = torch.max(outputs.data, 1)
164
                    #Compute loss
166
                    loss = criterion(outputs, labels)
167
168
                    #Accumulate statistics across a single batch
169
                    val_loss+= loss.item()
170
                    val_ppv += metrics(y, labels)[0]
                    val_npv += metrics(y, labels)[1]
172
173
           #Compute average of metrics across all batches
```

```
val_loss /= len(valloader)
175
           val_ppv /= len(valloader)
176
           val_npv /=len(valloader)
177
178
           # Here we save a checkpoint. It is automatically registered
179
      with
           # Ray Tune and can be accessed through 'session.get_checkpoint
      () (
181
           # API in future iterations.
           #Checkpoint data
183
           checkpoint_data = {
184
                "epoch": epoch,
185
                "classifier_state_dict": net.classifier.state_dict(),
186
                "optimizer_state_dict": optimizer.state_dict()
187
           }
188
189
           checkpoint = Checkpoint.from_dict(checkpoint_data)
190
           session.report({"v_loss": val_loss,"v_ppv": val_ppv,
191
                             "v_npv": val_npv, "tr_loss":train_loss,
192
                             "tr_ppv":train_ppv, "tr_npv":train_npv},
193
      checkpoint = checkpoint)
195
196 def testing(path, hn1, hn2, c_path):
197
       #Resample function
       transform = transforms.Compose([
198
           transforms.Resize(size=(224, 224)),
199
           transforms.ToTensor(),
200
           transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229,
201
      0.224, 0.225]),
      ])
202
203
       #Define testing dataset from folder
204
       test = ImageFolder(path, transform=transform)
206
       #Load testing dataset
207
       testloader = torch.utils.data.DataLoader(
208
           test, batch_size=len(test), shuffle=True, num_workers=
209
      num_cores
       )
210
211
       #Load pretrained AlexNet model
212
       net = torchvision.models.alexnet(weights='AlexNet_Weights.
```

```
IMAGENET1K V1')
214
       #Modify classification layer with best parameters
215
       net.classifier = torch.nn.Sequential(
216
           torch.nn.Dropout(p=0.5, inplace=False),
217
           torch.nn.Linear(in_features=9216, out_features=hn1, bias=True)
218
           torch.nn.ReLU(inplace=True),
219
           torch.nn.Dropout(p=0.5, inplace=False),
220
           torch.nn.Linear(in_features=hn1, out_features=hn2, bias=True),
           torch.nn.ReLU(inplace=True),
222
           torch.nn.Linear(in_features=hn2, out_features=2, bias=True))
223
224
       net.to(device)
225
       net.eval()
226
227
       #Load checkpoint and weights
228
       checkpoint = Checkpoint.from_directory(c_path)
229
       checkpoint_state = checkpoint.to_dict()
230
       net.classifier.load_state_dict(checkpoint_state["
      classifier_state_dict"])
232
233
       test_ppv, test_npv = 0, 0
234
235
       with torch.no_grad():
           for data in testloader:
               images, labels = data
237
               images, labels = images.to(device), labels.to(device)
238
239
               outputs = net(images)
240
               _, y = torch.max(outputs.data, 1)
241
242
               test_ppv += metrics(y, labels)[0]
243
               test_npv += metrics(y, labels)[1]
244
           test_ppv /= len(testloader)
246
           test_npv /= len(testloader)
247
       return test_ppv, test_npv, (test_ppv + test_npv)/2
249
250
251 #Save table with best results from each trial given a metric and
      filter mode
252 def saveFile(results, metric, filter_mode, save):
       df = results.get_dataframe(filter_metric = metric, filter_mode =
```

```
filter mode)
      df.to_csv(save)
254
255
257 #Restore tuners given training folders corresponding to original
      dataset and synthetic dataset
258 tuner = tune.Tuner.restore("/home/ricardo.velasco/ray_results/
      training_2023-09-18_16-14-45", trainable=training)
259 syn_tuner = tune.Tuner.restore("/home/ricardo.velasco/ray_results/
      training_2023-09-18_19-57-26", trainable=training)
260
261 #Get results from each tuner
262 results = tuner.get_results()
263 syn_results = syn_tuner.get_results()
264
265 #Save results given metrics, filter_mode and save path for dataset
266 saveFile(results, "v loss", "min", "/home/ricardo.velasco/TT kichwa/
      transfer_learning_model/tuning/tables/alexnet_ds_loss.csv")
267 saveFile(results, "v_bpv", "max", "/home/ricardo.velasco/TT_kichwa/
      transfer_learning_model/tuning/tables/alexnet_ds_bpv.csv")
268
269 #Save results given metrics, filter_mode and save path for synthetic
      dataset
270 saveFile(syn_results, "v_loss", "min", "/home/ricardo.velasco/
     TT_kichwa/transfer_learning_model/tuning/tables/alexnet_syn_ds_loss
271 saveFile(syn_results, "v_bpv", "max", "/home/ricardo.velasco/TT_kichwa
      /transfer_learning_model/tuning/tables/alexnet_syn_ds_bpv.csv")
273 #Test data given checkpoints
274 #Original dataset best TOP 5 MODELS LOSS
275 a, b, c = testing(testingPath, 4096, 512, "/home/ricardo.velasco/
     ray results/training 2023-09-18 16-14-45/
     training_80559_00021_21_batch_size=64,hn1=4096,hn2=512,lr=0.0021
      _2023-09-18_16-14-53/checkpoint_000032/")
276 print(a, b, c)
277
278 a, b, c = testing(testingPath, 2048, 256, "/home/ricardo.velasco/
     ray_results/training_2023-09-18_16-14-45/
     training_80559_00011_11_batch_size=64,hn1=2048,hn2=256,lr=0.0004
      _2023-09-18_16-14-53/checkpoint_000021")
279 print(a, b, c)
280
281 a, b, c = testing(testingPath, 512, 2048, "/home/ricardo.velasco/
```

```
ray_results/training_2023-09-18_16-14-45/
      training_80559_00009_9_batch_size=64,hn1=512,hn2=2048,lr=0.0006
      2023-09-18 16-14-53/checkpoint 000010")
282 print(a, b, c)
283
284 a, b, c = testing(testingPath, 512, 1024, "/home/ricardo.velasco/
      ray_results/training_2023-09-18_16-14-45/
      training_80559_00000_0_batch_size=64,hn1=512,hn2=1024,lr=0.0052
      _2023-09-18_16-14-53/checkpoint_000042")
285 print(a, b, c)
286
287 a, b, c = testing(testingPath, 1024, 2048, "/home/ricardo.velasco/
      ray_results/training_2023-09-18_16-14-45/
      training 80559 00006 6 batch size=512, hn1=1024, hn2=2048, lr=0.0635
      _2023-09-18_16-14-53/checkpoint_000046")
288 print(a, b, c)
289
290
291 #Original dataset best TOP 5 MODELS BPV
292 a, b, c = testing(testingPath, 4096, 2048, "/home/ricardo.velasco/
     ray_results/training_2023-09-18_16-14-45/
      training_80559_00016_16_batch_size=128,hn1=4096,hn2=2048,lr=0.0438
      _2023-09-18_16-14-53/checkpoint_000000")
293 print(a, b, c)
294
295 a, b, c = testing(testingPath, 64, 64, "/home/ricardo.velasco/
     ray_results/training_2023-09-18_16-14-45/
      training_80559_00028_28_batch_size=128, hn1=64, hn2=64, lr=0.0015_2023
      -09-18_16-14-53/checkpoint_000007")
296 print(a, b, c)
297
298 a, b, c = testing(testingPath, 1024, 2048, "/home/ricardo.velasco/
     ray results/training 2023-09-18 16-14-45/
      {\tt training\_80559\_00006\_6\_batch\_size=512,hn1=1024,hn2=2048,lr=0.0635}
      _2023-09-18_16-14-53/checkpoint_000009")
299 print(a, b, c)
300
301 a, b, c = testing(testingPath, 512, 1024, "/home/ricardo.velasco/
     ray_results/training_2023-09-18_16-14-45/
      training_80559_00000_0_batch_size=64,hn1=512,hn2=1024,lr=0.0052
      _2023-09-18_16-14-53/checkpoint_000008")
302 print(a, b, c)
303
304 a, b, c = testing(testingPath, 2048, 256, "/home/ricardo.velasco/
```

```
ray_results/training_2023-09-18_16-14-45/
     training_80559_00011_11_batch_size=64,hn1=2048,hn2=256,lr=0.0004
      _2023-09-18_16-14-53/checkpoint_000005")
305 print(a, b, c)
306
307
308 #Synthetic dataset best TOP 5 LOSS TESTING
309 a, b, c = testing(synTestingPath, 256, 2048, "/home/ricardo.velasco/
     ray_results/training_2023-09-18_19-57-26/
     training_9792e_00012_12_batch_size=128, hn1=256, hn2=2048, lr=0.0034
      _2023-09-18_19-57-27/checkpoint_000012")
310 print(a, b, c)
311
312 a, b, c = testing(synTestingPath, 128, 128, "/home/ricardo.velasco/
     ray_results/training_2023-09-18_19-57-26/
     training_9792e_00000_0_batch_size=64,hn1=128,hn2=128,lr=0.0016_2023
      -09-18 19-57-27/checkpoint 000044")
313 print(a, b, c)
314
315 a, b, c = testing(synTestingPath, 2048, 128, "/home/ricardo.velasco/
     ray_results/training_2023-09-18_19-57-26/
     training_9792e_00013_13_batch_size=64,hn1=2048,hn2=128,lr=0.0102
      _2023-09-18_19-57-27/checkpoint_000012")
316 print(a, b, c)
317
a, b, c = testing(synTestingPath, 4096, 64, "/home/ricardo.velasco/
     ray_results/training_2023-09-18_19-57-26/
     training_9792e_00008_8_batch_size=512, hn1=4096, hn2=64, lr=0.0114
      _2023-09-18_19-57-27/checkpoint_000002")
319 print(a, b, c)
320
321 a, b, c = testing(synTestingPath, 1024, 64, "/home/ricardo.velasco/
     ray results/training 2023-09-18 19-57-26/
     training_9792e_00007_7_batch_size=128,hn1=1024,hn2=64,lr=0.0354
      _2023-09-18_19-57-27/checkpoint_000003")
322 print(a, b, c)
323
324 #Synthetic dataset TOP 5 BEST BPV TESTING
325 a, b, c = testing(synTestingPath, 2048, 128, "/home/ricardo.velasco/
     ray_results/training_2023-09-18_19-57-26/
     training_9792e_00013_13_batch_size=64,hn1=2048,hn2=128,lr=0.0102
      _2023-09-18_19-57-27/checkpoint_000011")
326 print(a, b, c)
```

```
328 a, b, c = testing(synTestingPath, 1024, 64, "/home/ricardo.velasco/
      ray_results/training_2023-09-18_19-57-26/
     training_9792e_00007_7_batch_size=128, hn1=1024, hn2=64, lr=0.0354
      _2023-09-18_19-57-27/checkpoint_000003")
329 print(a, b, c)
330
a, b, c = testing(synTestingPath, 1024, 64, "/home/ricardo.velasco/
      ray_results/training_2023-09-18_19-57-26/
     training_9792e_00014_14_batch_size=256,hn1=1024,hn2=64,lr=0.0265
      _2023-09-18_19-57-27/checkpoint_000003")
332 print(a, b, c)
333
334 a, b, c = testing(synTestingPath, 4096, 64, "/home/ricardo.velasco/
      ray_results/training_2023-09-18_19-57-26/
     training_9792e_00008_8_batch_size=512, hn1=4096, hn2=64, lr=0.0114
      _2023-09-18_19-57-27/checkpoint_000002")
335 print(a, b, c)
336
337 a, b, c = testing(synTestingPath, 4096, 128, "/home/ricardo.velasco/
      ray_results/training_2023-09-18_19-57-26/
      {\tt training\_9792e\_00024\_24\_batch\_size=64,hn1=4096,hn2=128,lr=0.0008}
      _2023-09-18_19-57-27/checkpoint_000005")
338 print(a, b, c)
```