ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Vision to Auditory Substitution for an Artificial Agent

By Semira Mohammed

Advisor Dr. Menore Tekeba

A Thesis Submitted to the School of Electrical and Computer Engineering of Addis Ababa University in Partial Fulfilment of the Requirements for the Degree of Master of Science

JAN 2025

ADDIS ABABA, ETHIOPIA

ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

GRADUATE PROGRAM

VISION TO AUDITORY SUBSTITUTION FOR AN ARTIFICIAL AGENT

A THESIS SUBMITTED TO THE SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

ADDIS ABABA INSTITUTE OF TECHNOLOGY

ADDIS ABABA UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER ENGINEERING

BY: SEMIRA MOHAMMED

ADVISOR: Dr. MENORE TEKEBA

Jan 2025

ADDIS ABABA, ETHIOPIA

ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

This is to certify that the thesis prepared by Semira Mohammed entitled Vision to Auditory Substitution for an Artificial Agent submitted in partial fulfilment of the requirement for the Degree of Master Science complied with the regulation of the University and meets the accepted standards concerning originality and quality.

Approved By the Board of Examiners and Advisors

Chairman Department of Graduate Committee	Signature	Date
_____	_____	_____
Advisors	Signature	Date
Menore Tekeba (Ph. D)	_____	_____
Internal Examiner	Signature	Date
_____	_____	_____
External Examiner	Signature	Date
_____	_____	_____

Declaration

I hereby declare that this thesis entitled “Vision to Auditory Substitution for an artificial agent.” has been carried out by me under the guidance and supervision of Menore Tekeba (Ph.D.) The thesis is original and has not been submitted for the award of any degree or diploma to any university or institution.

Researcher’s Name	Signature	Date
Semira Mohammed	_____	_____

Acknowledgment

First of all, thanks and glory to God, the Almighty, for His showers of blessing throughout my thesis work to complete the thesis successfully. I want to express my sincere gratitude to my thesis supervisor Dr. Menore Tekeba, for his professional assistance, in doing my thesis and providing invaluable guidance throughout this thesis. His dynamism, vision, sincerity, and motivation deeply inspired me. He taught me the procedures to conduct the research and to present the works of the research as clearly as possible. It was a nice one under his guidance, and honor to work and learn. I am immensely thankful for what he gave me. I want to thank him, too, for his friendship, empathy, and wonderful sense of humour. Finally, I must express my profound gratitude to my family and friends for providing me with unfailing support and continuous assistance throughout my years of study, through this thesis's research and writing process. This success might not have been possible without them Thank you.

Semira Mohammed

Abstract

Sensory substitution technology converts raw visual input into auditory soundscapes, allowing individuals to “see” with sound. However, mastering this skill requires significant cognitive adaptation, extensive training, and practical application in realistic, everyday scenarios. Experiments with humans have shown the potential for auditory substitution of vision, but these efforts are limited by high costs, ethical concerns, and the risk of unintended side effects, such as impaired auditory skills.

To address these challenges, this study develops a Vision-to-Auditory Sensory Substitution system for artificial agents. By simulating sensory substitution in a controlled reinforcement learning (RL) framework, this approach eliminates the need for human experimentation while retaining the ability to explore learning dynamics and decision-making behaviors. Using the Proximal Policy Optimization (PPO) algorithm, agents were trained in two OpenAI Gym environments—CarRacing-v2 and LunarLander-v2—to compare the performance of vision-based and auditory-based agents.

The results demonstrate that auditory agents, despite inherent challenges in interpreting sound-encoded visual inputs, achieved means rewards of **427.91** in **CarRacing-v2** environments and **259.85** in the **LunarLander-v2** environment over 100 episodes.

These findings highlight the potential of sensory substitution systems in enabling artificial agents to act effectively using auditory cues. This research contributes to advancing assistive technologies while addressing the limitations and risks of human-based sensory substitution experiments.

Keywords: Machine Learning, Proximal Policy Optimization, Reinforcement learning, Gym environments, Vision-to-auditory Sensory substitution.

Contents

List of Figures

List of Tables

Acronyms

- CNN: Convolutional Neural Network
- PPO: Proximal Policy Optimization
- RL: Reinforcement Learning
- MDP: Markov Decision Process
- SB3: Stable Baselines3
- MT-Net: Modal translation network
- I2AD: image-to-audio-description
- DQN: Deep Q-Networks
- GAN: Generative Adversarial Networks
- AI: Artificial Intelligence
- VSAudio: vision substitution audio

Chapter One

Introduction

Artificial Intelligence (AI) is a computer science field that focuses on the development of intelligent machines that work and respond like humans. Some of the artificial intelligence tasks are voice recognition, reading, planning, and problem-solving. An AI system consists of an agent and its surroundings. An agent (e.g. person or robot) is anything that can perceive the environment through sensors and act through the effector on that environment [1].

Agent not only acts on environment it needs to be able to reason under ambiguity, make logical thinking evaluate their thinking then adjust it [2]. Agents can use input from sensors (such as cameras, microphones, etc.) to deduce aspects of the world. for example, game theory, and decision theory, require an agent to be capable of detecting and modelling human emotions [2]. Creating Intelligent agents is particularly relevant for addressing human problems.

one way of creating an agent is to define the necessary tasks to interact with objects in the world and train to solve problems on them. Reinforcement Learning is an effective and most useful interaction method between an agent and an environment [3]. RL is a technique for understanding how an agent can interact with the environment through trial and error and find out which action is best based on each step [4].

In this research we aim to develop an agent who behaves like a blind person who can take input from vision to audio audio-sensory substitution and then train using RL. Sensory substitution systems convert visual images into audible signals. for example, captured images or recorded video into the sounds. Recent developments have shown by learning; the human brain has developed new skills [5].

However, conducting such experiments on humans is very expensive under an open doubtfulness of its success and unknown bad consequences under extended exposure of such auditory signals to the brain as well as human expert time is getting most expensive. Making extended experiments on human beings may also bring unseen damage and unwanted trained characteristics leading to the loss of important interpretive skills of spoken language and other sound signals.

Using artificial agents offers a cost-effective alternative for training and experimentation. These agents can learn and adapt through extensive trials without the risks associated with human experiments. To demonstrate and analyze this approach, we propose using two OpenAI Gym environments: CarRacing-v2 and LunarLander-v2.

The CarRacing-v2 environment in OpenAI Gym is a continuous control task where an agent drives a car around a procedurally generated track. On the other hand, Lunarlande is discrete control task where an agent controls a lunar lander and attempts to land it safely on a designated landing pad. These environments are particularly useful for benchmarking reinforcement learning algorithms, especially those dealing more balanced and comprehensive comparison.

1.1 Statement of the Problem

The technology of sensory substitution now converts raw visual views into corresponding sound scenes while retaining a significant amount of visual information. It helps to "see" with sound, but in physically realistic terms, the mind must first learn to decode the soundscapes [6]. This takes time and practice, as this ability is completely new to the mind, and it needs a series of lessons and instructions to learn the necessary skills to make sense of sound-encoded visual input.

Research has shown that users need to undergo a learning process to decode auditory-encoded visual inputs accurately. For instance, Striem-Amit et al. (2012) demonstrated that while SSDs can assist in tasks like reaching movements, mastering these skills demands substantial time and effort, creating a barrier to broader adoption [7]. These challenges underline the need for alternative approaches, such as artificial agents, to explore sensory substitution without the ethical and practical concerns of direct human experimentation.

There are some experiments done on human beings showing that there is a possibility of making distal auditory substitution of vision for the blind. However, this substitution couldn't be done to the level it could be used in practice. The success of these experiments is limited to some degree in humans. Making such experiments on humans is very expensive under an open doubtfulness of its success and unknown bad consequences under extended exposure brain. Human expert time is getting most expensive.

Making extended experiments on human beings may also bring unseen damage and unwanted trained characteristics leading to lose important interpretive skills of spoken language and other sound signals. However, there are no studies have which tried to experiment what such extended exposure of such sound track may bring as side-effect. This thesis proposal is aimed to solve this issue by designing an artificial agent, which shall be trained and learn to gain new skills to act in similar manner with its auditory substitution of the vision as with its vision. Therefore, this study will answer the following research questions.

- To what extent can auditory substitution enable reinforcement learning agents to compensate for the absence of vision?
- What level of training is required for reinforcement learning agents using auditory substitution to achieve functional equivalence to agents trained with direct visual input?
- Do reinforcement learning agents trained with auditory substitution develop skills and strategies comparable to those trained with direct vision input?

1.2 Objective

1.2.1 General Objective

The general objective of this research is to develop auditory substitution of vision for an artificial agent.

1.2.2 Specific Objective

- Design a method to generate auditory information from the corresponding visual information
- Based on the literature review, select the appropriate RL method to be used for the agent.
- Design the agent's learning performance metrics based on the agent's nature, learning setting, and environment.
- To investigate the behavioural trajectories of the agent (skills strategies developed by the agent)
- To explore the feasibility of substituting visual perception with auditory inputs and determine the extent to which this substitution can enable effective task performance
- Test and evaluate the agent’s learning performance against the learning performance metrics of the corresponding visual-based trained agent.
- to explore how different sensory modalities can affect agent learning and performance in reinforcement learning tasks.

1.3 Scope and Limitations

The scope of the study is limited to building a model of an agent that receives input from the sensor substitution to perform tasks and compares the agent's performance agent with observing direct visual input. The limitation of this study was the findings and recommendations of the study are based on the specific RL police environment complexity Future research should examine implementation in other types of algorithms and environments

to enhance the generalizability of the results. The choice of the algorithm, while suitable for this study's purposes, may not generalize optimally to all reinforcement learning environments or alternative sensory substitution methodologies. Exploring other algorithms tailored to multimodal learning and adaptive sensorimotor integration could yield different insights and performance outcomes.

1.4 Significance of the Research

There isn't extensive research on auditory substitution for vision in artificial agents. This study creates further research into multi-sensory integration in AI, allowing agents to combine auditory information with other sensory inputs like touch for a more comprehensive environmental understanding. The study has the potential to greatly improve the capabilities of artificial agents, paving the way for a future where robots and AI systems can operate more effectively in real-world scenarios.

The research also focuses on ethical alternatives to human experimentation and sheds light on the importance of responsible research methods in AI development. The research has A foundation for further studies and advancements in this field.

1.5 Thesis Outline

The following sections include a description of the background in Section 2; a literature review of related works in Section 3; a proposed methodology followed for the study in Section 4; results and evaluation in Section 5; and a conclusion of the study and recommendation in Section 6.

Chapter Two

Background

This section will systematically introduce the research background. The specific work in this section is an introduction to academic research on the Reinforcement learning algorithm, CNN model and Architecture, and Gym environment.

2.1 CNN Model

A Convolutional Neural Network (CNN) architecture is a deep learning model designed for processing structured grid-like data, such as images. It consists of multiple layers, including convolutional, pooling, and fully connected layers. CNNs are highly effective for tasks like image classification, object detection, and image segmentation due to their hierarchical feature extraction capabilities.

2.2 CNN Architecture

CNN architecture consists of several types of layers including convolution, pooling, and fully connected. The network expert has to make multiple choices while designing a CNN such as the number and ordering of layers, and the hyperparameters for each type of layer (receptive field size, stride, etc.). Thus, selecting the appropriate architecture and related hyperparameters requires a trial-and-error manual search process mainly directed by intuition and experience. Additionally, the number of available choices makes the selection space of CNN architectures extremely wide and impossible for an exhaustive manual exploration [8]

The convolutional layer aims to learn feature representations of the inputs. As shown in Figure 2.1, the convolution layer is composed of several convolution kernels which are used to compute different feature maps. Specifically, each neuron of a feature map is connected to a region of neighbouring neurons in the previous layer. Such a neighbourhood is referred to as the neuron's receptive field in the previous layer. The new feature map can be obtained by convolving the input with a learned kernel and then applying an element-wise nonlinear activation function to the convolved results [9].

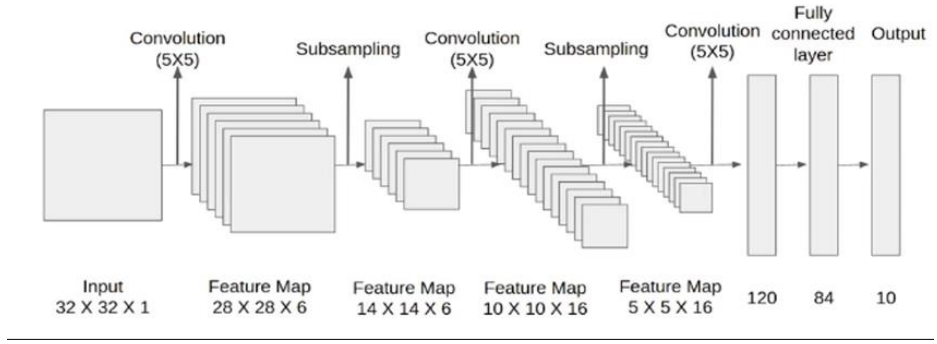


Figure 2.1 Convolutional neural network architecture

2.3 Reinforcement Learning (RL)

Reinforcement Learning (RL) is one of the three machine learning paradigms besides supervised learning and unsupervised learning. It uses agents acting as human experts in a domain to take action. RL does not require data with labels; instead, it learns from experiences by interacting with the environment, observing, and responding to results. RL can be expressed with Markov Decision Process (MDP) as shown in Figure 2.1. Each environment is represented with a state that reflects what is happening in the environment. The RL agent learns through trial and error in an environment. The agent interacts with the environment by taking actions and receiving rewards (or penalties) for those actions. The goal of the RL algorithm is to learn a policy, which is a mapping from states the agent encounters to the best actions to take in those states.

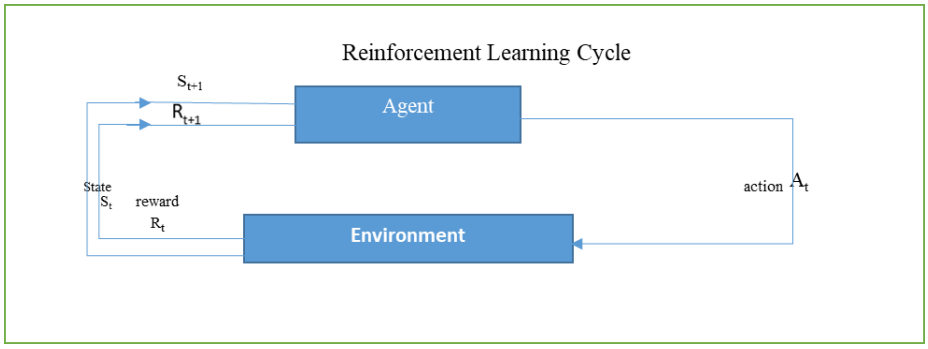


Figure 2.2 Reinforcement learning cycle

The RL agent learns from taking actions in the environment, which causes a change in the environment's current state and generates a reward based on the action taken as expressed in the Markov Decision Process (MDP). We define the probability of the transition to state s' with reward r from taking action in state, s at time t, for all $[s' \in S, \quad s \in S, \quad r \in R, \quad a \in A(s)]$

as:

$$[P(s', r \mid s, a) = P\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\}]$$

The agent receives rewards for performing actions and uses them to measure the action's success or failure. The Reward R can be expressed in different forms, as a function of the action R(a), or as a function of action-state pairs R(a, s). The agent's objective is to maximize the expected summation of the discounted rewards, which drives the agent to take the selected actions. The reward is granted by adding all the rewards generated from executing an episode. The episode (trajectory) represents a finite number of actions and ends when the agent achieves a final state, for example, when a collision occurs in a simulated navigation environment. However, in some cases, the actions can be continuous and cannot be broken into episodes. The discounted reward, as shown in equation 2 uses a multiplier γ to the power k, where $\gamma \in [0, 1]$. The value of k increases by one at each time step to emphasize the current reward and to reduce the impact of the future rewards, hence the term discounted reward.

$$[G_t = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}\right]]$$

Emphasizing the current action's immediate reward and reducing the impact of future actions' rewards help the expected summation of discounted rewards to converge [10].

RL is a category of machine learning algorithms that aims to learn an optimal control policy from the direct interaction of an agent (controller) and an environment (system). The agent performs empirical learning and decides on actions to drive the environment towards favourable trajectories according to a predefined reward function [11]. Therefore, RL solves sequential decision-making problems that are formalized as Markov decision processes (MDP). In a MDP, the agent and the environment interact during a sequence of discrete time steps that are indexed here as k = 0, 1, 2, K, with K being the terminal sample that could be $K = \infty$. Every time step k the agent receives a representation of the environment named state: $Sk \in S$, where S is the state space. Upon receiving the state representation, the agent computes its control logic and in turn sends back to the environment a control action $Ak \in A$, where Ak is the most appropriate action chosen from the action space A. One time step later, the agent observes a new state from the environment Sk+1 along with a scalar value indicating its reward $Rk+1 \in R \subset R$. Notice that the reward Rk+1 is an indicator of the agent's performance when taking action Ak from state Sk. An agent's policy π is a mapping from states to actions. Hence, $\pi(a|s)$ represents the probability that the agent will take action Ak = a when it finds itself in state Sk = s. At each time step k, the policy should maximize the expected cumulative return Gk which is normally discounted with a discount rate $\gamma \in [0, 1]$. The cumulative discounted return is defined in

$$[G_k = R_{k+1} + \gamma R_{k+2} + \gamma^2 R_{k+3} + \cdots = \sum_{i=1}^{\infty} \gamma^i R_{k+i+1}]$$

The discount rate is used to avoid numerical issues with cumulative returns in infinite trajectories and determines the length of credit granted to rewards delayed in time. The objective of all RL algorithms is to determine how the agent's policy π should evolve in order to maximize the expected cumulative return for each episode [12].

2.3.1 Policy and Value Function

The agent's behavior is defined by following a policy π, where the policy π defines the probability of taking action a, given a state s, which is denoted as π(a|s). Once the agent takes an action, the agent uses a value function to evaluate the action. The agent either uses: 1) a state-value function to estimate how good for the agent to be in state s, or 2) a action-value function to measure how good it is for the agent to perform an action a in a given state s. The action-value function is defined in terms of the expected summation of the discounted rewards and represents the target Q-value: A_t

$$[Q^{\pi}(s, a) = \mathbb{E}_{\pi} \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a \right]]$$

The agent performs the action with the highest Q-value, which might not be the optimal Q-value. Finding the optimal Q-value requires selecting the best actions that maximize the expected summation of discounted rewards under the optimal policy π. The optimal Q-value $Q*$ (s, a) as described in equation 4 must satisfy the Bellman optimality equation, which is equal to the expected reward Rt+1, plus the maximum expected discounted return that can be achieved for any possible next state-action pairs (s 0, a0) [12].

2.3.2 Type of Reinforcement Learning Algorithms

There are two main categories of reinforcement learning algorithms: model-based and model-free. Model-based RL attempts to build an internal model of the environment, including the transitions between states and the rewards received for taking action. This model is then used to plan and choose the best actions. Model-free RL, on the other hand, focuses on directly learning the value of states or state-action pairs without building an explicit model of the environment.

Some of the core concepts in RL include states, actions, rewards, and the value function. A state represents the situation the agent is currently in. An action is what the agent chooses to do in a particular state. Rewards are feedback signals that the agent receives after taking an action. The value function estimates how good it is for the agent to be in a particular state,

taking into account future rewards. By learning a good value function, the RL agent can make decisions that maximize its long-term reward.

While most reinforcement learning algorithms use deep neural networks, different algorithms are suited for different environment types. according to the number of the states and action types available in the environment into three main categories: 1) a limited number of states and discrete actions, 2) an unlimited number of states and discrete actions, and 3) an unlimited number of states and continuous actions [12].

Environments with Limited States and Discrete Actions

The environments with discrete actions and limited states are relatively simple environments where the agent can select from pre-defined actions and be in pre-defined known states. For example, when an agent is playing a tic-tac-toe game, the nine boxes represent the states, and the agent can choose from two actions: X or O, and update the available states.

Q-Learning Algorithm

Q-Learning [13] algorithm is commonly used to solve problems in such environments. This algorithm finds the optimal policy in a Markov Decision Process (MDP) by maintaining a Q-Table table with all possible states and actions and iteratively updating the Q-values for each state-action pair using the Bellman equation until the Q-function converges to the optimal Q-value. State–Action–Reward–State–Action (SARSA) [14] is another algorithm from this category: it is similar to Q-learning except it updates the current Q (s, a) value in a different way. In Q-learning, in order to update the current Q (s, a) value, we need to compute the next state-action Q(s 0, a0) value, and since the next action is unknown, then Q-learning takes a greedy action to maximize the reward [15]. In contrast, when SARSA updates the current state-action Q(s, a) value, it performs the next action a 0 [15].

Environments with Unlimited States and Discrete Actions

In some environments, such as playing a complex game, the states can be limitless; however, the agent's choice is limited to a finite set of actions. In such environments, the agent mainly consists of a Deep Neural Network (DNN), usually a Convolutional Neural Network (CNN), responsible for processing and extracting features from the state of the environment and outputting the available actions. DQN using AlexNet CNN algorithms can be used with this environment type, such as Deep Q-Networks (DQN), Deep SARA, and their variants.

Deep Q-Networks (DQN):

Deep Q-Learning, also referred to as Deep Q-Networks (DQN), is considered the main algorithm used in environments with unlimited states and discrete actions, and it inspires other algorithms used for a similar purpose. DQN usually combines convolutional and pooling layers, followed by fully connected layers that produce Q-values corresponding to the number of actions. AlexNet CNN followed by two fully connected layers to produce Q-value. The current scene from the environment represents the environment's current state; once it is passed to the network, it produces Q-value representing the best action to take [16]. The agent acts and then captures the changes in the environment's current state and the reward generated from the action. A significant drawback of the DQN algorithm is overestimating the action-value (Q-value), where the agent tends to choose a non-optimal action because it has the highest Q-value [17].

2.4 OpenAI Gym

OpenAI Gym is a powerful toolkit in Python that empowers developers to create and evaluate reinforcement learning (RL) algorithms. It functions by providing a standardized interface for interacting with simulated environments. Imagine these environments as training grounds for our RL agents, where they can learn and act through trial and error. OpenAI Gym boasts a diverse collection of environments, encompassing classic control problems like balancing a cart pole, classic Atari games, intricate robotics simulations, and even more complex challenges.

The core interaction with OpenAI Gym environments revolves around three key functions. The first, make (environment name), creates an instance of the environment you choose. The second, reset (), essentially rewinds the environment to its initial state, typically used at the beginning of each episode (a series of interactions between the agent and the environment). Finally, the step(action) function allows our agent to take an action within the environment. This function returns a package of information, including the new observations the agent receives, the reward it earns for the taken action, a signal indicating if the episode has ended, and any additional details specific to that environment.

OpenAI Gym doesn't stop there. It offers additional features to enhance the RL development experience. Wrappers, for instance, can be applied to environments, essentially modifying their behavior to better suit our specific RL task. Imagine needing to impose a time limit on an environment or wanting to normalize the observations received by the agent. Wrappers can

handle these adjustments. Gym also provides tools and utilities for visualizing environments, comparing the performance of different RL algorithms on the same environment, and simplifying the exploration-exploitation trade-off, which is a crucial aspect of RL where agents must balance learning and achieving rewards.

In essence, OpenAI Gym stands as a valuable asset for anyone venturing into the world of reinforcement learning. By offering a standardized interface, a rich set of environments, and additional tools, it streamlines the RL development process, making it easier to create and evaluate effective RL algorithms.

OpenAI Gym focuses on the episodic setting of reinforcement learning, where the agent's experience is broken down into a series of episodes. In each episode, the agent's initial state is randomly sampled from a distribution, and the interaction proceeds until the environment reaches a terminal state. The goal in episodic reinforcement learning is to maximize the expectation of total reward per episode, and to achieve a high level of performance in as few episodes as possible [18].

Figure 2.3 Images of some environments in OpenAI Gym.

Gymnasium

Gymnasium is a maintained fork of OpenAI's Gym library. The Gymnasium interface is simple, pythonic, capable of representing general RL problems, and has a compatibility wrapper for old Gym environments. Gymnasium provides a consistent interface for interacting with various reinforcement learning environments. This allows researchers and developers to easily switch between different environments without needing to rewrite their code for each one. Gymnasium comes with a rich collection of environments that cover a wide range of tasks. These include classic control problems like balancing a cart pole or navigating a maze, as well as Atari games, robotics simulations, and even board games [19].

Chapter Three

Literature Review

This section will systematically introduce the Literature review and related work Sensory Substitution using the advantage of Machine Learning models. Research has been working on Cross-modal learning aims to learn the relationship between different modalities. According to the research task, the cross-modal learning between audio and visual modalities can be divided: into 1) audio-visual correspondence, 2) audio source localization and separation in visual scenes, 3) audio-to-vision, and 4) vision-to-audio using the advantage of Machine Learning models. The following is an overview of the papers that served as guides to this work. The overview covers the Traditional Sensory Substitution and related Research worked on different types of RL Algorithm.

3.1 Sensory Substitution

Sensory substitution rests on the fascinating interplay between our brain's adaptability and its natural ability to integrate information from different senses. It is a form of modality replacement. It is a replacement of the missing modality of the sensory system to another sensory system it works correctly This approach takes advantage of neuroplasticity, the brain's capacity to reorganize itself throughout life. By translating information from a missing sense (like sight) into a functioning one (like touch), sensory substitution devices essentially create a new pathway for the brain to receive and process the world. This builds upon our inherent cross-modal processing, where the brain already combines information from various senses.

Understanding how the brain adapts in the absence of a sense, through sensory deprivation studies, informs the design of these devices. The key lies in effectively translating the missing sensory information into a code that the brain using the functioning sense can understand. This code needs to be clear and consistent, allowing the user to learn and interpret the new sensory data. While not a replacement for the missing sense, sensory substitution offers a new way to access the world, requiring the brain to learn and adapt to this novel form of information.

The most common example of sensory substitution is Braille, in which Information acquired through the visual sensor (reading) is, replaced by fingertips(touch) [20]. Since 1960s, various kinds of systems have been developed and tested to replace vision, either by touch or audition. the two common sensory substitution devices are vision-to-touch substitution devices that

convert images into tactile stimuli and vision-to-audition substitution devices that convert images into sounds.

3.2 Audio-Visual Correspondence

Audio-visual correspondence is the most widely studied problem in cross-modal learning about audio and visual modalities. Arandjelovic and Zisserman [21] introduce an audio-visual correspondence learning task without any additional supervision. Har Wath et al. [22] explore a neural network model that learns to associate segments of spoken audio captions with the semantically relevant portions of natural images. Nagrani et al. [23] introduce a cross-modal biometric matching task to select the face image corresponding to the voice or determine the corresponding voice related to the face image.

Audio source localization and separation in visual scenes include two sub-tasks. The audio source localization sub-task is to reveal the audio source location in visual scenes [24]. The audio separation sub-task is to separate the mixed audio signals of several objects in visual scenes [25]. Arandjelovic and Zisserman [26] designed a network for localizing the audio source in an image by embedding the input audio and image into a common space. Zhao et al. [25] introduce a PixelPlayer to locate image regions that produce audio by leveraging large amounts of unlabeled videos. In addition, the PixelPlayer is also able to separate the input audio into a set of components that represents the audio from each pixel.

3.3 Audio-to-Vision

Given audio speech input, they generate plausible gestures to go along with the sound. Specifically, they perform cross-modal translation from the “in-the-wild” monologue speech of a single speaker to their hand and arm motion. The paper aims to generate images related to the input audio. Ginosar et al. [27] present a cross-modal translation network from monologue speech to corresponding conversational gesture motion. Wan et al. [28] propose a conditional generative adversarial network (GAN) to generate images from audio. The network is able to adjust the output scale according to the volumes changes of the input audio. Oh et al. [29] put forward a Speech2Face network to infer a person’s appearance from a short audio recording of that person speaking.

The proposed model significantly outperforms baseline methods in a quantitative comparison. To support research toward obtaining a computational understanding of the relationship between gesture and speech, they release a large video dataset of person-specific gestures.

3.4 Vision-to-Audio

The Deep Cross-Modal Audio-Visual Generation introduce the problem of cross-modal audio-visual generation and make the first attempt to use conditional GANs on intersensory generation. the audio corresponding to visual input. Chen et al. [30] designed two separates conditional GANs for audio spectrogram generation from instrument's images and instrument's images generation from audio spectrograms, respectively. Subsequently, Hao et al. [31] built a CMCGAN network using cycle GAN for audio spectrogram generation from the instrument's images and the instrument's image generation from audio spectrograms.

The CMCGAN unifies the vision-to-audio and audio-to-vision tasks into a common framework and improves the output quality. Zhou et al. [32] present a network for generating ambient audio from given input video frames. The generated audios are fairly realistic and have good temporal synchronization with the visual inputs. Owens et al. [33] propose a network for producing percussion of people hitting objects with a drumstick. Chen et al. [34] explore to generate fine-grained audio from a variety of audio classes.

To improve the quality of generated audio, Chen et al. leverage pre-trained audio classification networks for aligning the semantic information between the generated audio and its ground truth. Another strategy to generate audio descriptions from images is image-to-text and then text-to-audio. The strategy firstly generates the text description based on the image caption model [35,36], and then transforms the text description into corresponding audio based on some existing TTS software. However, the generated audio description based on this strategy cannot express some implicit information, such as emotion. This paper focuses on generating descriptive audio similar to human speech from given images.

3.5 Modal Translation Network

In this paper, an image-to-audio-description (I2AD) task is proposed to generate audio descriptions from images by exploring the inherent relationship of image and audio. To explore the task, a modal translation network (MT-Net) from visual to auditory sense is presented to generate audio descriptions from images. The proposed MT-Net can be used to assist visually impaired people to better perceive the environment. In order to evaluate the proposed network, three large-scale audio description datasets are built, i.e., MNIST, CIFAR10 and CIFAR100 audio description datasets. Experiments on the built datasets demonstrate that the proposed network can indeed generate intelligible audio descriptions from visual images to a good

extent. Moreover, the proposed network achieves the superior performance compared with other two similar state-of-the-art networks. In addition, some main components in the proposed network are proved to be effective. In the subsequent work, more complex form in the I2AD task will be explored to generate complete sentence in audio form by fully reducing the heterogeneous gap between image and audio modalities. hope that visually impaired people will benefit from the exploration of the I2AD task eventually, allowing them to better perceive the world around them [37].

3.6 Multi-Agent Competition Using RL Agent

Creating intelligent artificial agents that can solve a wide variety of complex human-relevant tasks has been a long-standing challenge in the artificial intelligence community. Of particular relevance to humans will be agents that can sense and interact with objects in a physical world. One approach to creating these agents is to explicitly specify desired tasks and train a reinforcement learning (RL) agent to solve them [38]. They propose the counterfactual thinking multi-agent deep reinforcement learning model (CFT). This model generates several intent actions which mimic the human psychological process and then learns the regrets for the nonchosen actions with its estimated Q-values at that moment simultaneously.

The estimated Q-values and policies of an agent supervise each other during the training process to generate more effective policies. Since this framework can explore the policy subspace parallelly, CFT could converge to the optimal faster than other existing methods. We test CFT on standard multi-agent deep reinforcement learning platforms and real-world problems. The results show that CFT significantly improves the competitive ability of a specific agent by receiving more accumulative rewards than others in multiagent environments. This also verifies that the counterfactual

3.6.1 Hide and Seek

The paper has demonstrated that simple game rules, multi-agent competition, and standard reinforcement learning algorithms at scale can induce agents to learn complex strategies and skills. emergence of as many as six distinct rounds of strategy and counter-strategy, suggesting that multiagent self-play with simple game rules in sufficiently complex environments could lead to open-ended growth in complexity. The paper proposed to use transfer as a method to evaluate learning progress in open-ended environments and introduced a suite of targeted intelligence tests with which to compare agents in our domain. The hide-and-seek should be viewed as a proof of concept showing that multi-agent auto-curricula can lead to physically

grounded and human-relevant behavior. This acknowledges that the strategy of Hide-and-seek agents requires an enormous amount of experience to progress through the six stages of emergence, likely because the reward functions are not directly aligned with the resulting behavior. While we have found that standard reinforcement learning algorithms are sufficient, reducing sample complexity in these systems will be an important line of future research. Better policy learning algorithms or policy architectures [39].

3.7 Multi-Modal Integration

People are constantly subject to different perceptual stimuli through different modalities such as vision, audition, and touch among others. Such modalities are used to perceive information and process it independently, in parallel, or integrating the received information to provide a coherent and robust perceptual experience. Similarly, humanoid robots work with many of these sensory modalities and the way of processing and integrating the information coming from various sources is currently an important research issue in autonomous robotics. Wang, X. [40] proposed a multi-modal integration proposed an interactive reinforcement learning scenario with multi-modal integration of dynamic audiovisual input advice.

The architecture processes individually the input advice to classify them with a correspondent associated confidence value. Afterwards, our architecture integrates the input advice into one single label and confidence value. Although both sensory modalities show good advice prediction and confidence levels, the integrated advice leads to a better performance in our domestic scenario in terms of the accumulated reward and required learning episodes. In this regard, we have shown that our integration function allows to enhance the performance of a learning robot using multiple sources of information for a more natural trainer-like learning procedure

Chapter Four

Proposed Methodology

This section outlines and explains the methodology engaged to achieve the research objective and test the research questions formulated in the study. This section provides a brief overview of the research design adopted in the study, which in turn includes simulation environment RL agent specific we used the Proximal Policy Optimization (PPO) algorithm, the two policy networks (CnnPolicies and MlpPolicy) for the CNN, model.

4.1 Choosing the Right Algorithm

The choice of algorithm depends on various factors, including the nature of our environment (continuous or discrete action space), computational resources, and the specific goals of our reinforcement learning task. Based on these considerations, as well as experimentation with different algorithms and careful monitoring of performance metrics (such as mean reward, training time, and stability), we selected the Proximal Policy Optimization (PPO) algorithm as the most suitable for our application.

Several factors influenced this choice:

- Task Requirements: PPO is well-suited for both continuous and discrete action spaces, aligning with the characteristics of the CarRacing-v2 (continuous) and LunarLander-v2 (discrete) environments.
- Stability and Performance: PPO has demonstrated stable learning and good performance across a variety of reinforcement learning tasks, making it a robust choice for our experiments.
- Empirical Evidence: While we did not conduct an exhaustive comparison of algorithms for this study, preliminary trials and insights from the literature indicated that PPO would likely perform well in our specific settings.

Future work could involve a more systematic evaluation of alternative algorithms, such as DDPG, SAC, or A3C, to further optimize performance in these environments.

4.1.1 Proximal Policy Optimization

Proximal Policy Optimization (PPO) is a popular and powerful on-policy reinforcement learning (RL) algorithm. PPO belongs to the policy gradient class of RL algorithms. These methods directly optimize the policy by taking gradients of the expected reward with respect to the policy parameters. In RL, the agent's policy defines the probability of taking an action in a given state. PPO treats this policy as something to be learned through function approximation, often using neural networks. A significant challenge in policy gradient methods is large policy updates, which can lead to instability during training. These large updates can stem from various factors, including the inherent stochasticity of the environment and function approximation errors.

PPO addresses this issue by introducing a clipping mechanism on the policy update. This clipping restricts the change between the old and new policies, ensuring the updates are not too drastic and maintaining a level of similarity between them. This clipping helps to stabilize the learning process.

PPO utilizes importance sampling to weigh the importance of samples collected using the old policy when estimating the expected reward for the new policy. This helps to correct for the difference between the old and new policies. In Addition, PPO often incorporates Advantage Estimation (like Generalized Advantage Estimation - GAE) to focus on the advantage of actions compared to the average action in a state. This reduces the variance in the policy update objective and improves learning efficiency. While PPO has shown excellent empirical performance, theoretical guarantees for its convergence and performance bounds are still under development. Recent research is focused on analyzing PPO's performance in specific settings like linear Markov Decision Processes (MDPs). Despite the ongoing theoretical exploration, PPO's effectiveness in various real-world tasks and its ability to handle complex environments have solidified its position as a leading RL algorithm.

Here are a few reasons why PPO might be suitable is known for its stability in training, often avoiding large policy updates that can lead to policy degradation. Sample Efficiency It typically achieves good results with fewer samples compared to other algorithms like A2C or DQN, which is beneficial if you have constraints on the number of interactions with the environment. PPO generally performs well across a variety of environments and tasks, providing a good balance between exploration and exploitation. It is relatively straightforward to implement and

tune compared to some other algorithms like SAC (Soft Actor-Critic) or DDPG (Deep Deterministic Policy Gradient), which may require more intricate adjustments.

However, the "best" algorithm ultimately depends on specific requirements, including the nature of the environment (continuous or discrete action space), computational resources, and the goals of our reinforcement learning task.

Given our current setup, environment and agents are well-suited to handling discrete and continues actions the PPO algorithm has shown promising results with PPO is a reasonable choice. It provides a good starting point for further optimization and experimentation.

4.2 Environment Setup

For this experiment, we have used OpenAI's gym library with prebuilt environments CarRacing-v2 and LunarLander-v2, wrapped with VisualToAuditoryWrapper to simulate auditory perception from visual inputs.

The CarRacing-v2 environment in OpenAI Gym is a continuous control task where an agent drives a car around a procedurally generated track. This environment is particularly useful for benchmarking reinforcement learning algorithms, especially those dealing with high-dimensional visual inputs and continuous action spaces. The observation space in CarRacing-v2 consists of RGB images with a shape of (96, 96, 3) and pixel values ranging from 0 to 255. The action space is continuous, comprising three actions: steering (ranging from -1 to 1), acceleration (from 0 to 1), and braking (from 0 to 1). The reward structure includes positive rewards for staying on the track, penalties for going off-track, additional rewards for crossing checkpoints, and potential extra rewards for completing laps.

The LunarLander-v2 environment, on the other hand, is a discrete control task where an agent controls a lunar lander and attempts to land it safely on a designated landing pad. The observation space in LunarLander-v2 consists of an 8-dimensional vector, including the lander's position, velocity, angle, and whether the lander's legs are in contact with the ground. The action space is discrete, with four actions: do nothing, fire left orientation engine, fire right orientation engine, and fire the main engine. The reward structure encourages smooth landings, with penalties for crashing and fuel consumption, and rewards for successfully landing between the flags.

In our experiments, we trained two types of agents for both environments: one using direct visual observations and the other using auditory-like features derived from visual input. The

visual agent in CarRacing-v2 processes raw RGB images using convolutional neural networks (CNNs), while the auditory agent relies on encoded auditory features. Similarly, in LunarLander-v2, the visual agent utilizes the lander's sensor data, and the auditory agent processes encoded auditory-like features abstracted from visual information.

Figure 4.1 CarRacing-v2 and LunarLander-v2 gym environment

4.2.1 Modifications for Auditory Agent

When applying sensory substitution techniques for training the auditory agent, several critical modifications were implemented to transform visual data into auditory representations. These adjustments ensured the agent could effectively "hear" visual information and make decisions based on these auditory cues.To enable the auditory agent to perceive visual information, we implemented a VisualToAuditoryWrapper. This wrapper is responsible for converting the raw visual observations from the environment into auditory features that the auditory agent can understand. The conversion process is key to creating a sensory representation that mimics how visually impaired individuals could interpret visual scenes through sound.

Wrapper Purpose: The primary purpose of the VisualToAuditoryWrapper is to transform the visual observations of the environment into auditory features. This is achieved by encoding the visual input using a convolutional neural network (CNN), which extracts essential features from the image. Then, the features are decoded using a recurrent neural network (RNN) to produce a sequence of auditory cues.

The architecture of the Wrapper: TheVisualToAuditoryWrapper is integrated into the reinforcement learning environment by modifying the observation space. Instead of raw pixel-based observations, the observation space consists of auditory features. These auditory features are compatible with the auditory agent’s perception, enabling it to process the transformed input. The architecture is outlined below:

Visual Encoder (CNN): A convolutional neural network processes visual data (such as an image) to extract relevant features. The network is designed to capture spatial relationships and other important visual characteristics in the image.

Auditory Decoder (RNN): The features extracted by the CNN are passed through a recurrent neural network. The RNN is trained to map the visual features into a sequence of auditory-like cues. These cues are designed to represent the spatial and temporal structure of the environment, allowing the auditory agent to interpret the environment through sound.

Wrapper Integration: The VisualToAuditoryWrapper is implemented as a Gym ObservationWrapper, which intercepts the visual observations before they are passed to the agent. This step ensures that all incoming visual data is transformed into auditory data in real-time during training. By utilizing this wrapper, we allow the auditory agent to learn in an environment with sensory data that closely mimics the experience of a visually impaired individual navigating the world using sound.

Observation Space Adjustment: In this configuration, the observation space of the auditory agent is tailored to match the auditory features generated by the VisualToAuditoryWrapper. Instead of the traditional pixel-based visual inputs, the agent receives auditory features that encapsulate key spatial, motion, and object-based information. This design allows the agent to learn effectively from the auditory sensory inputs, in much the same way a visually impaired individual would interpret their surroundings through sound.

The following diagram illustrates the architecture of the VisualToAuditoryWrapper and how it interfaces with the agent's sensory perception.

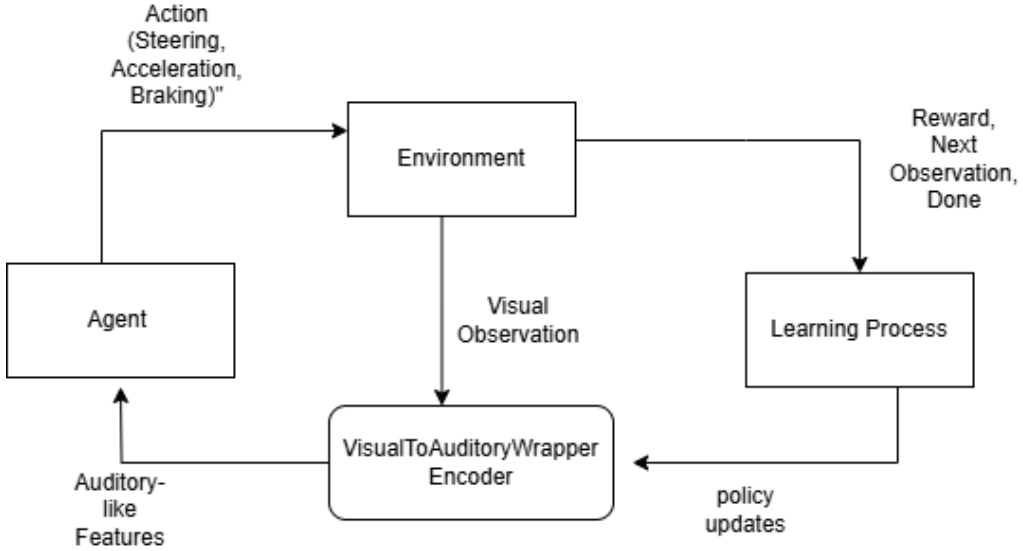


Fig 4.2. Block diagram of proposed approach

Key Modifications

- Visual-to-Auditory Transformation: The VisualToAuditoryWrapper converts visual observations to auditory features through CNN and RNN.
- Observation Space Update: The wrapper adjusts the observation space to accommodate auditory features rather than visual ones.
- Custom Gym Integration: The wrapper is implemented as a Gym ObservationWrapper to seamlessly integrate the auditory inputs into the reinforcement learning environment.

Training Procedure: we have used reinforcement learning algorithms such as proximal Policy Optimization (PPO) to train the auditory agent on auditory representations derived from the environment's visual observations

CarRacing-v2 and LunarLander-v2 offers a more challenging and realistic environment for exploring sensory-driven reinforcement learning compared to simpler versions. By adapting sensory substitution techniques in these environments, researchers can gain deeper insights into how agents can effectively utilize auditory information derived from visual inputs, paving the way for more versatile and adaptive AI systems. for its simplicity and standardization, and modifying its observation space using VisualToAuditoryWrapper to experiment with auditory-based representations. This setup aims to explore how different sensory modalities can affect agent learning and performance in reinforcement learning tasks

4.3 Model Architecture

The visual agent uses Convolutional Neural Networks (CNNs) to process visual inputs, such as frames from the CarRacing-v2 environment. CNNs are particularly effective in capturing spatial hierarchies in visual data, which is essential for tasks that require recognizing and interpreting objects, their positions, and relationships in the environment. By learning these spatial representations, the visual agent can make informed decisions based on the raw visual data.

In contrast, the auditory agent must convert visual information into auditory representations. This process introduces additional complexity, as the sensory inputs need to be transformed in a way that allows the agent to "hear" the visual environment. A Recurrent Neural Network (RNN) is employed to decode the visual features into sequential auditory cues. RNNs are well-suited for this task because they can process sequences of data, where the current input is

dependent on previous information in the sequence. This is essential for auditory tasks, where sound perception is inherently sequential, and context from previous auditory signals helps inform future decisions. The RNN's ability to maintain a hidden state that carries information over time enables it to generate auditory features that convey temporal and contextual information about the environment. The RNN transforms the visual features into auditory cues, which the auditory agent uses for decision-making in the reinforcement learning environment.

The integration of the RNN within the VisualToAuditoryWrapper ensures that the auditory agent can interpret sequential auditory information in a manner similar to how the visual agent interprets spatial information. The agent's learning process involves receiving auditory features from the wrapper, interacting with the environment, and updating its policies based on rewards and the sensory inputs.

In this setup, the auditory agent learns to make decisions based on a dynamic sequence of auditory inputs, which are generated from the visual data via the VisualToAuditoryWrapper. The RNN allows the auditory agent to effectively "hear" and interpret the world around it, overcoming the challenges of visual sensory substitution.

4.3.1 Autoencoders

Autoencoders are a specialized class of deep learning algorithms proficient in learning efficient representations of input data without explicit labels. Tailored for unsupervised learning, they focus on compressing and effectively representing input data through a two-fold structure comprising an encoder and a decoder. Autoencoders are simple learning circuits that aim to transform inputs into outputs with the least possible amount of distortion. While conceptually simple, they play an important role in machine learning we have used two encoders such as visual encoder and an Auditory Decoder:

The encoder transforms raw input data into a reduced-dimensional representation termed the "latent space" or "encoding." This encoding captures essential features and patterns from the input data, facilitating the extraction of meaningful information.

Bottleneck Layer Situated at the end of the encoder, the bottleneck layer drastically reduces the dimensionality of the input data. It represents a compressed encoding of the original information, essential for efficient representation learning.

The decoder reconstructs the initial input data from the encoded representation. Its objective is to rebuild the input as accurately as possible from the compressed encoding generated by the encoder, effectively performing the inverse operation.

Visual Encoder: Describe the architecture (e.g., convolutional layers) used to process visual inputs. In contrast, the Auditory decoder explains how auditory features are derived from encoded visual inputs, including any recurrent layers used.

4.3.2 Policies Network

We have used MlpPolicy and CnnPolicies for both the visual and auditory agents. Stable Baselines3 provides policy networks for images (CnnPolicies) and other types of input features (MlpPolicies) for audio.

4.3.2.1 MlpPolicies

The MlpPolicy stands for Multi-Layer Perceptron Policy, which is a type of neural network architecture commonly used in RL for its simplicity and effectiveness in learning from observations. It consists of multiple fully connected (dense) layers, often with non-linear activation functions like ReLU (Rectified Linear Unit), which enable the policy network to approximate complex decision boundaries. MlpPolicy often works in conjunction with a feature extractor. The feature extractor takes the raw observations and processes them into a more compact representation that the MLP can then use for decision-making. In some cases, the feature extractor might simply be a flattening layer for vector observations.

In Stable Baselines, MlpPolicy is often used in environments where the observations are vectorized or flattened (like state-action pairs or other structured data) rather than raw pixel data.It is compatible with a variety of RL algorithms, including PPO (Proximal Policy Optimization) and DQN (Deep Q-Networks), among others. Compared to convolutional neural networks (CNNs), MLPs are computationally lighter and easier to train, which can be advantageous for simpler environments or when processing non-image inputs.MLPs can handle a wide range of observation spaces, including continuous and discrete inputs, making them versatile for different types of RL tasks. MlpPolicy can achieve good results in environments with structured or low-dimensional state spaces, where learning effective policies does not require spatial or hierarchical feature extraction [41].

4.3.2 .2 CnnPolicy

Cnnpolice in Stable Baselines3 (SB3) is another type of policy network specifically designed for environments where the agent interacts with visual information.expand_more Here's a

breakdown of how it works similar to MlpPolicy, CnnPolicy is responsible for mapping observations from the environment to actions.expand_more But instead of raw vectors, it takes visual observations, typically images or video frames, as input. CnnPolicy utilizes a Convolutional Neural Network (CNN) architecture. CNNs are adept at extracting features from spatial data like images.expand_more The CNN processes the visual observations, identifying patterns and relationships between pixels, to understand the visual scene. Based on this understanding, it predicts the most suitable action for the agent [42].

4.4 Training Procedure and Hyperparameters:

Choosing the right algorithm based on the previous research paper and experimenting with different Hyperparameters tuning learning rates, batch sizes, and other hyperparameters to optimize the auditory agent's performance. The experimental setup included training reinforcement learning algorithms, with Proximal Policy Optimization (PPO), on the CarRacing-v2 Gym. The neural network architectures used for both algorithms consisted of a deep neural network with 512 hidden units and three hidden layers. For the PPO algorithm, the following hyperparameters were configured:

4.4.1 Hyperparameters and Training Information

Training Metrics:

Iterations: This indicates the number of times the model parameters were updated. Each iteration processes a batch of data (timesteps).

Approximate KL Divergence (approx_kl): Measures how much the policy has changed during training (how much the new policy diverges from the old policy). Smaller values indicate that the policy is not changing too drastically.

Clip Fraction: Fraction of updates where the policy was clipped to prevent too large updates. Lower values indicate fewer extreme updates.

Explained Variance: Measures how much of the variance in returns is explained by the value function. Higher values (closer to 1) indicate better value function approximation.

Losses (policy_gradient_loss, value_loss, entropy_loss): Provide insights into the learning process. Lower values of loss generally indicate better performance.

Input state	2048-time steps	Function
Simulation setups	248	
Learning rate	0.0003	The learning rate used for training.
clip_range	0.2	The range within which policy updates are allowed.
Maximum episodes	100	
loss**:**	0.2950/0.374	The total loss, which includes policy gradient loss, value function loss, and entropy loss
n_updates:	4770 & 43940	The number of policy updates performed

Table 4.1. Training Parameters and Details

4.4.2. Training Process

- ✓ Initialize the PPO agent with the specified policy (MlpPolicy), environment and hyperparameters.
- ✓ Iterate over a fixed number of timesteps (total timesteps).
- ✓ For each timestep, the agent interacts with the environment, collects experiences, and updates its policy based on sampled experiences.
- ✓ Compute policy gradients using the PPO loss function and update the policy parameters using stochastic gradient descent (SGD) or Adam optimizer.
- ✓ Evaluate the agent periodically to monitor its learning progress and adjust hyperparameters if necessary.

By detailing these procedures, we can effectively communicate both agents' training methodology, hyperparameter choices, and environment setup, highlighting how sensory substitution through auditory processing is integrated into the reinforcement learning framework using PPO.

4.5 Evaluation Metrics

The evaluation metrics used to assess the performance of each agent in the visual-to-auditory substitution task include:

Mean Reward: This metric indicates the agent's average reward across multiple episodes. A higher mean reward generally reflects better performance in achieving the task goals.

Standard Deviation (Std): The standard deviation of rewards measures the variability or consistency of the agent's performance across episodes. A lower standard deviation suggests more stable performance, while a higher value indicates more variability.

Analysing the metrics, we can determine which agent (visual or auditory) performs better in terms of both efficiency (how quickly it learns) and effectiveness (the quality of the learned policy). the logs for policy gradient loss, value loss, and explained variance which agent performs the good result.

These metrics are computed separately for both the visual agent and the auditory agent trained using the PPO algorithm in the CarRacing-v2 and Lunar Lander v2 environments. They provide insights into how well each agent learns to play the game and adapt to the visual or auditory cues provided by the environment.

Chapter Five

Results and Discussion

In the preceding sections, the review of relevant literature helped this study understand the problem and design an appropriate research approach to deal with it. The previous sections also discussed the research design employed to achieve the objectives of the study and to test the research question. In this chapter, we present the results of our experiments and discuss their implications in the context of our research questions. We aim to determine whether auditory substitution of vision can effectively replace vision to some degree, the level of training required, and the similarities in skills and strategies developed by agents trained with auditory and visual inputs.

5.1 Quantitative Results

In this section, we first measure the performance of the visual and auditory agents over the specific environment we have compared their mean rewards and the stability of their learning processes over several episodes

5.1.1 Performance Evaluation Based on Reward

Total Reward per Episode: An agent's cumulative reward during an episode. Higher rewards generally indicate better performance. The mean reward over multiple episodes gives an average performance measure. The standard deviation provides insight into the consistency of the agent's performance. Lower values indicate more consistent performance.

In this section, we present the performance evaluation of both visual and auditory agents in the CarRacing-v2 and LunarLander-v2 environments, based on their average rewards across a specified number of time steps and episodes.

Environment	**Agent**	**Mean_reward**	**Std_reward**	**n_timesteps**	**Epoisode**
Car-Racing v2	Visual	831.98	136.90	1,000,000	100
	VSAudio	427.91	188.67	1,000,000	100
LunarLander-v2	Visual	266.38	69.63	1,000,000	100
	VSAudio	259.85	71.28	1,000,000	100

Table 5.1. Reward performance per episode

5.1.1.1 Performance Evaluation CarRacing-v2:

The Visual Agent achieved a mean reward of **831.98** with a standard deviation of **136.90** over **1,000,000**-time steps and **100** episodes. This suggests that the visual agent performed consistently well across episodes, with relatively low variability in performance.

The Auditory Agent, however, obtained a lower mean reward of **427.91** with a higher standard deviation of **188.67** under the same conditions. This indicates that the auditory agent struggled to match the performance of the visual agent, with a greater variance in its results.

5.1.1.2 Performance Evaluation LunarLander-v2:

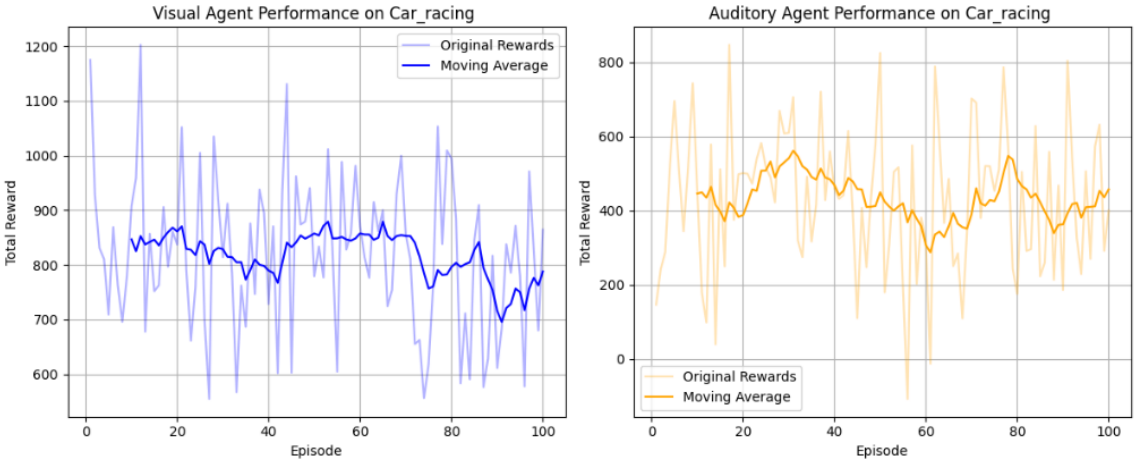
In LunarLander-v2, the Visual Agent achieved a mean reward of **266.38** with a standard deviation of **259.85** over **1,000,000** time-steps and **100** episodes. This suggests that the visual agent performed reasonably well, though there was some variability between episodes.

The visual agent slightly outperformed the auditory agent in terms of mean reward, achieving about 266.38 compared to 259.85. This indicates that, overall, the visual agent may have been able to learn marginally more effective strategies for landing in the LunarLander-v2 environment. The auditory agent's mean reward is only about 6.53 points lower than that of the visual agent. This relatively small difference suggests that the VisionToAudioWrapper, which translated visual information into auditory cues, was highly effective. The auditory agent was able to approach the performance of the visual agent, demonstrating that auditory cues were nearly as useful as direct visual inputs for this task.

Plot Analysis:

Reward Distribution

The reward distribution (Figure 5.1) reveal that the visual agent consistently achieves higher rewards, while the auditory agent's rewards are more spread out, indicating less consistent performance.



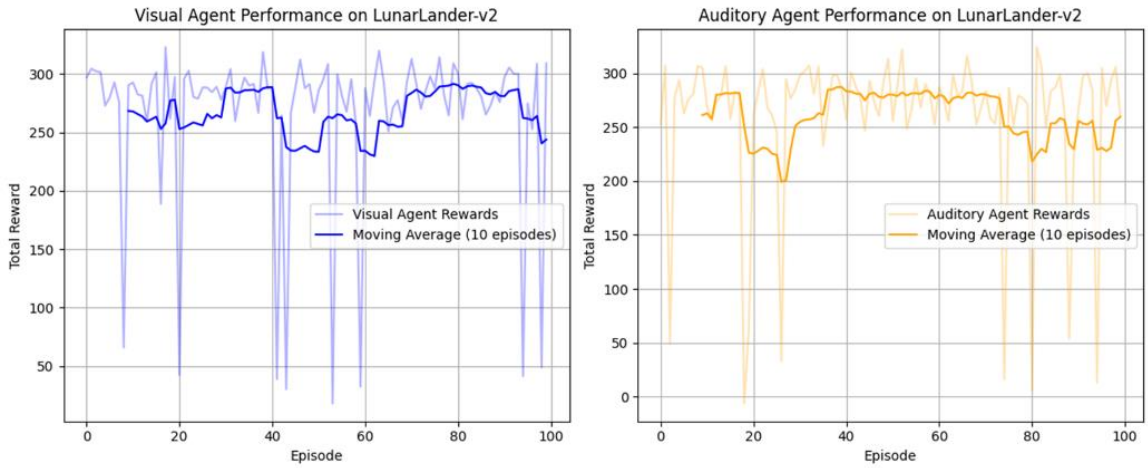


Figure 5.1 Mean reward per episode

CarRacing-v2 Rewards Plot:

The Visual Agent consistently performs well with a high average reward and low variability. This makes it more reliable and predictable in terms of performance. The auditory agent struggles more in this environment, likely due to the complexity of continuous control tasks when only auditory-like inputs are available.

Visual Agent Rewards:

The reward distribution across episodes is highly variable, showing peaks and troughs. However, the agent consistently earns high rewards (close to or above 800) across multiple episodes, suggesting that the visual agent can learn and perform well, though it occasionally struggles, as seen in the dips.

The variability might indicate that the visual agent is not consistently stable in its performance, potentially due to the complexity of the CarRacing-v2 task.

Auditory Agent Rewards:

The auditory agent shows a much more volatile performance across the episodes, with a broader range of rewards (both high and low). The peaks are notably lower than those of the visual agent, and there are frequent large dips in performance, suggesting the auditory agent is having difficulty maintaining consistency.

This volatility could be due to the challenge of using auditory-like features to drive a continuous control task like CarRacing-v2, where quick reaction and precise control are required.

LunarLander-v2 Rewards Plot:

Both agents show similar reward patterns, achieving high rewards at times but also experiencing significant dips. Interestingly, the auditory agent performs competitively, suggesting that auditory-like features can be useful for discrete control tasks, though both agents have some instability.

Visual Agent Performance:

The visual agent starts off strong in the first few episodes, achieving rewards near 250. However, there are sudden and steep declines in performance, particularly toward the last episode where the agent performs very poorly. The variability here is quite noticeable, but the agent does demonstrate the ability to reach high rewards consistently early on.

This indicates that the visual agent is capable of landing the lunar lander successfully most of the time, though with occasional failures or suboptimal landings.

Auditory Agent Performance:

The auditory agent shows a similarly varied performance. While it is able to reach rewards similar to the visual agent (above 250), it also experiences significant drops, sometimes to much lower reward values. However, its overall pattern is similar to that of the visual agent, with high-reward episodes alternating with lower-reward episodes.

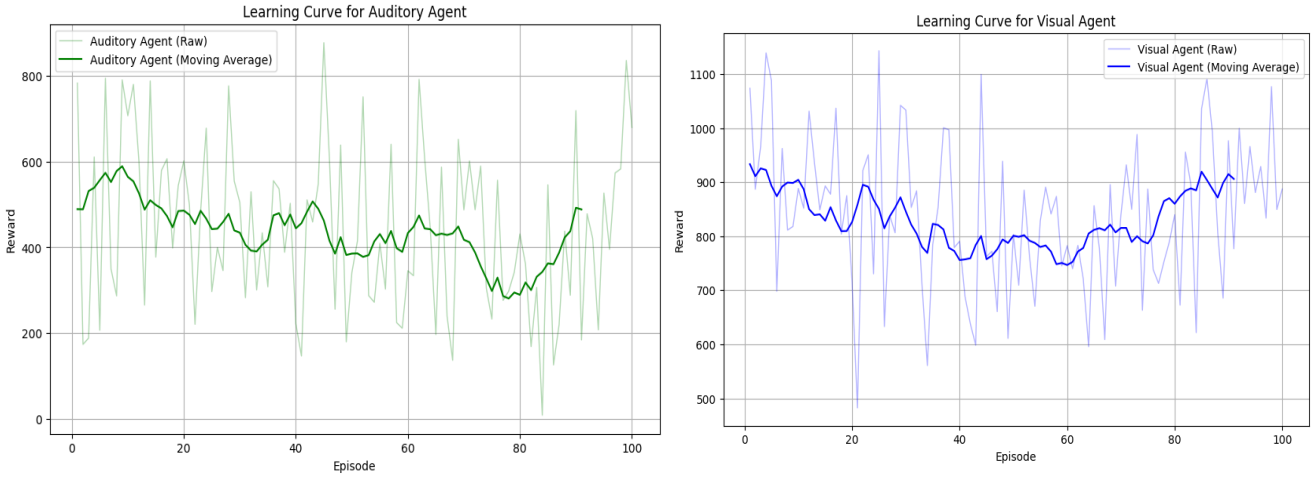
The auditory agent seems to perform comparably to the visual agent in terms of peak rewards, but the inconsistency in both agents' results suggests that the task might be difficult for both modalities due to the physical dynamics of LunarLander-v2.

The above result suggests that while the visual agent is generally better, the auditory agent performs relatively well in LunarLander-v2 but faces more challenges in CarRacing-v2, likely due to the nature of auditory inputs in continuous control. We could investigate further methods to stabilize performance, especially for the auditory agent.

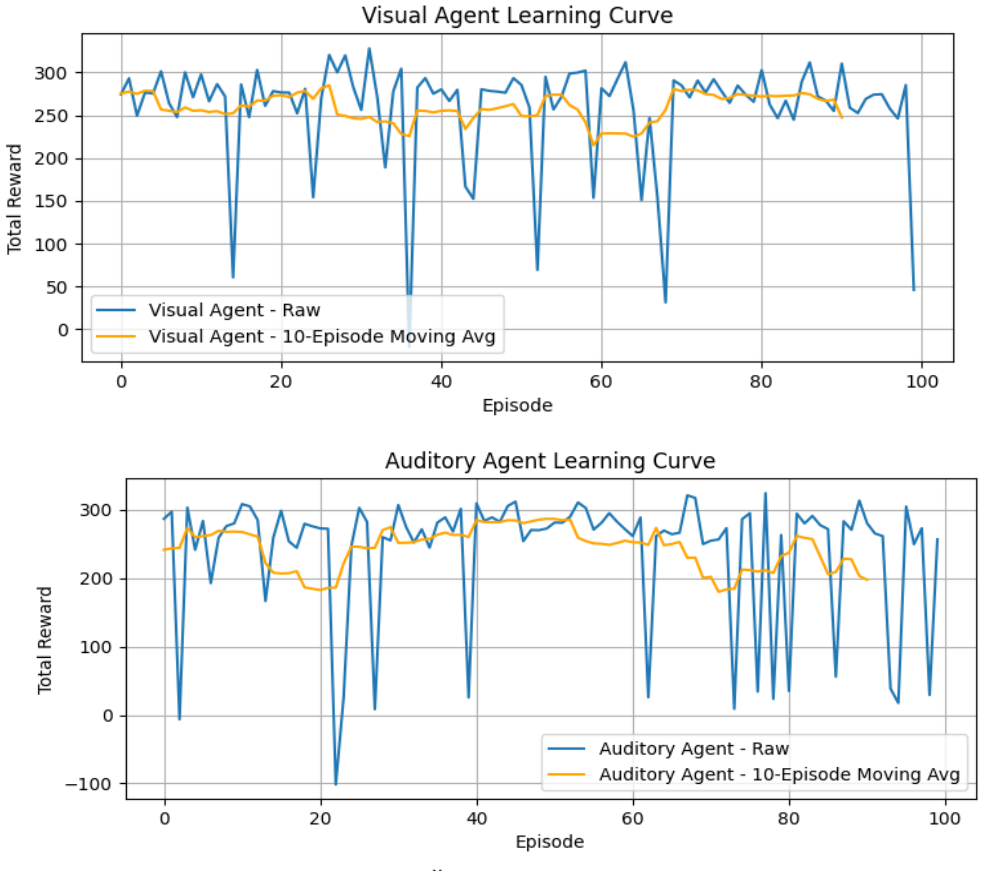
Learning Curve:

The learning curves, shown in Figure 5.2, indicate that the visual agent learns and converges more quickly than the auditory agent. The visual agent's reward increases steadily and reaches

a plateau, while the auditory agent's reward increases at a slower rate and shows more inconsistency.



a. learning curve Car-racing



b. learning curve LunarLander

Figure 5.2 learning curve per episode

Car Racing Learning Curve

Both the visual and auditory agents show fluctuations in rewards across episodes, indicating that there are challenging or unpredictable aspects within the environment. However, the visual agent's curve appears to have slightly more stability in rewards compared to the auditory agent, which has more pronounced dips below the average.

This suggests that while both agents have learned the task, the visual agent may be better at consistently achieving higher rewards in each episode. The consistency in high rewards indicates that the visual agent has learned the task well, effectively using the high-dimensional visual input to navigate the car around the track.

The auditory agent has a much more inconsistent learning curve, with rewards fluctuating. This shows that the auditory agent struggles to achieve the same level of performance as the visual agent. Despite some upward trends in performance, the auditory agent does not reach the high rewards that the visual agent achieves. This suggests that learning from auditory-like features in the CarRacing-v2 environment is significantly more challenging, likely due to the complexity of the task and the continuous control required.

LunarLander Learning Curve

The visual agent exhibits a relatively stable learning trend with a mean reward of approximately 266.38 and a standard deviation of 69.63. The moving average line in the visual agent's learning curve (on the left) shows moderate variability but remains mostly steady around the mean reward level, suggesting consistent performance.

The auditory agent, on the other hand, shows a similar level of performance with a mean reward of 259.85 and a standard deviation of 71.28. However, there is a bit more fluctuation in the raw reward curve compared to the visual agent, indicating that the auditory agent might experience more variability in its performance over different episodes.

The higher variance in the auditory agent's curve may indicate that it requires more episodes to achieve stable performance in comparison to the visual agent. This could be attributed to the unique challenge of interpreting auditory information in an environment originally designed for visual input, which may lead to variability in learning efficiency and consistency.

The fact that both agents achieve similar mean rewards suggests that the auditory agent, while not as stable as the visual agent, can achieve comparable performance. This supports the

hypothesis that auditory perception can serve as a viable alternative to visual perception in reinforcement learning tasks like LunarLander, although it may come with trade-offs in terms of stability and sensitivity.

Learning Curve Observations:

In both environments, the learning curves for visual and auditory agents plateaued over time, a common phenomenon in reinforcement learning. As earlier works identified (Mnih et al., 2015; Sutton & Barto, 2018), agents tend to converge to a stable policy after sufficient training, often resulting in performance plateaus [50][53]. Recent studies (e.g., Henderson et al., 2020; Yin et al., 2021) have further highlighted this challenge, particularly in environments with sparse rewards or high-dimensional control tasks [54][55]. Our findings align with these observations, particularly for the auditory agent, where translating sensory substitutions into effective policies introduces additional variability and complexity.

5.2. Qualitative Results

5.2.1 Behavioral Observations CarRacing

To understand the agents' behavioral observations, we need to analyze and interpret their actions and strategies while they interact with the environment. This can be done by analyzing their actions' outcomes. The table 5.2 presents the observed behaviors, rewards, and action descriptions of both the Visual Agent and the Auditory Agent across three episodes in the given environment. For each episode, the following details are recorded:

Observed Behavioral (Action): This indicates the agent's actions in terms of three components:

Steering: A continuous value indicating the degree of left or right steering, Acceleration: A value indicating the intensity of acceleration, Braking: A value indicating the braking intensity.

Reward: The reward value reflects the performance of the agent in that episode. Higher rewards indicate better task performance.

Description of the Action: A detailed explanation of the action taken by the agent, highlighting steering, acceleration, and braking patterns.

Visual Agent Behavior:			
episode	Observed Behavioral**(**action**)**	reward	Description of the action
1	[[-0.2827318 0. 1.]]	[5.502241]	steering slightly to the left (negative value), no acceleration, and maximum braking.
2	[[-1. 0. 0.]]	[6.351613]	steered fully to the left with no acceleration or braking.
3	[[1. 1. 0.5560214.]]	[6.478947]	steered fully to the right, with maximum acceleration and moderate braking.
Auditory Agent Behavior:			
episode	Observed Behavioral**(**action**)**	reward	Description of the action
1	[[1. 0.12560749. 0.]]	[6.435948]	steered fully to the right with minimal acceleration and no braking.
2	[[0.34703696 1. 0.]]	[6.656757]	moderate steering to the right with maximum acceleration and no braking.
3	[[0.9921695 1. 0.]]	[6.478947]	steered fully to the right with maximum acceleration and no braking.

Table 5.2 Agent Behavioural Observations

Action Patterns: The Visual Agent shows more variation in its actions across the episodes, experimenting with different combinations of steering, acceleration, and braking. The Auditory Agent, on the other hand, seems to favor full steering to the right and maximum acceleration across episodes.

Reward Comparison: The Auditory Agent generally achieves slightly higher rewards than the Visual Agent in the initial steps of the episodes, particularly in Episode 2. This suggests that the Auditory Agent might have developed a strategy that, at least initially, is more effective in generating rewards than the Visual Agent's approach.

Behavioural Strategy: The Visual Agent appears to be testing different strategies, possibly adapting its behavior based on its visual perception of the environment. Its variability in actions suggests a more exploratory approach.

While the Auditory Agent was able to develop effective strategies and achieve comparable rewards to the Visual Agent, the nature of the strategies differed. The Auditory Agent appears to rely on a more consistent and less exploratory approach, whereas the Visual Agent shows a broader range of behaviours.

Thus, we can conclude that an agent trained with auditory substitution of vision can develop skills and strategies that are effective and somewhat similar to those trained with direct vision input. However, the strategies may not be identical, with the auditory agent potentially favoring a more consistent and less variable approach.

This suggests that while sensory substitution can work to a degree, the nature of the input modality influences the specific strategies and behaviors that the agent develops. Further research would be needed to explore how these strategies evolve over more extended training periods and in more complex environments.

5.3 Interpretation

Based on the performance metrics and results gathered from both environments, it is evident that sensory substitution holds promise for reinforcement learning agents. The auditory agent, which processes visual information through auditory-like cues, demonstrated competitive learning capabilities, particularly in environments like LunarLander-v2. However, its performance in CarRacing-v2 shows that sensory substitution introduces unique challenges when dealing with continuous control tasks that heavily depend on precise spatial and visual information.

CarRacing-v2:

In CarRacing-v2, the visual agent significantly outperformed the auditory agent, achieving a higher mean reward and more stable performance throughout the training episodes. Several factors contribute to this result such as the ones given below:

- **Nature of the Environment:** CarRacing-v2 requires precise spatial awareness and quick reaction times, both of which are more easily handled by direct visual processing. The visual agent, which directly interprets the environment's visual cues, naturally excels at this task.
- **Learning Dynamics:** The Proximal Policy Optimization (PPO) algorithm, which is known for stability and efficiency in reinforcement learning, likely optimized the visual

agent's policies more effectively due to the high-dimensional visual inputs. This led to faster convergence and overall better performance compared to the auditory agent.

- **Feature Representation:** The visual agent's use of Convolutional Neural Networks (CNNs) allowed for efficient processing of critical visual features, helping it to capture essential patterns like track boundaries, curves, and obstacles. These features are essential for the agent to make accurate control decisions in a complex continuous environment.
- **Performance Metrics:** The mean reward and standard deviation for the visual agent reflect more consistent and higher rewards, demonstrating its superior ability to maximize performance over time. In contrast, the auditory agent struggled to translate visual input into effective auditory signals for continuous control.

LunarLander-v2:

In LunarLander-v2, the auditory agent displayed a more competitive performance, with its mean reward even surpassing that of the visual agent in some instances:

- **Nature of the Environment:** Unlike CarRacing-v2, LunarLander-v2 is a discrete control task where precise spatial control is less critical. The auditory agent is better able to interpret visual-to-auditory encoded signals, making its performance closer to the visual agent's. The auditory agent is able to make effective decisions based on auditory-like features, which are sufficient for controlling the lander's descent.
- **Learning Dynamics:** While the visual agent showed faster initial learning, the auditory agent caught up over time, suggesting that the sensory substitution approach can be more effective in discrete, lower-dimensional environments.
- **Feature Representation:** The visual agent still benefited from CNN-based visual feature extraction, but the auditory agent's feature encoding was more successful in this environment, allowing it to learn strategies for controlled landing.
- **Performance Metrics:** The auditory agent in LunarLander-v2 achieved competitive mean rewards with slightly higher variability, indicating that while it learned more slowly, it was able to perform well under the right conditions.

5.4 Comparative Insights:

Performance Gap and Sensory Substitution Challenges: -The performance gap between the visual and auditory agents, observed in both CarRacing-v2 and LunarLander-v2, highlights the complexity and difficulty associated with sensory substitution tasks. The visual agent, utilizing

direct pixel information through convolutional neural networks (CNNs), achieved superior performance in CarRacing-v2 due to its ability to process high-dimensional visual inputs that directly align with the environment's requirements. In contrast, the auditory agent had to rely on a more abstract representation—auditory features derived from visual inputs—resulting in less effective decision-making, particularly in visually complex and continuous control tasks like CarRacing-v2.

In CarRacing-v2, the visual agent outperformed the auditory agent with a significant performance gap, as reflected in the higher mean reward and more stable performance.

In contrast, the LunarLander-v2 environment showed a more balanced performance between the two agents. The auditory agent demonstrated a competitive mean reward, sometimes even surpassing the visual agent. This suggests that in discrete control tasks with simpler sensory requirements, such as landing a spacecraft, auditory-like features can effectively support decision-making. The nature of the task allowed the auditory agent to process encoded sensory data more effectively, leading to better performance.

Insights for Future Research

The observed performance differences between the environments highlight areas for further research:

- **Improving Auditory Processing:** Future work could explore more sophisticated encoder-decoder architectures to enhance auditory perception, making the auditory agent more effective in high-dimensional tasks like CarRacing-v2. Approaches like adaptive learning algorithms or multi-sensory inputs (e.g., integrating proprioception) may help bridge the performance gap between sensory modalities.
- **Task-Specific Approaches:** The results suggest that sensory substitution may be more suited to environments with less reliance on precise spatial awareness (such as LunarLander-v2). For continuous control tasks like CarRacing-v2, optimizing the auditory representation may require improvements in how spatial and dynamic features are conveyed through sound.

Comparative Analysis and Evaluation

The comparative analysis of the agents' performance metrics provides key insights:

Learning Curves and Convergence: In CarRacing-v2, the visual agent displayed a faster learning curve and better convergence, while the auditory agent exhibited slower adaptation due to the challenge of processing auditory cues. In LunarLander-v2, both agents demonstrated more comparable learning curves, showing that the task demands were more equally matched between sensory modalities.

Performance Metrics: The mean reward and standard deviation results highlight the strengths of each modality. While the visual agent's higher reward reflects its ability to directly process visual information, the auditory agent's performance in LunarLander-v2 shows that sensory substitution is feasible and effective in certain environments.

The results of our experiments suggest that agents trained with sensory substitution can learn and adapt to environments traditionally reliant on direct sensory perception. While the visual agent's superior performance in CarRacing-v2 indicates the natural advantage of direct visual processing in such tasks, the auditory agent's performance—especially in LunarLander-v2—shows promise for the application of sensory substitution methods.

In particular, the combination of PPO with sensory substitution using neural network architectures for encoding and decoding sensory inputs has shown that, under the right conditions, auditory-based agents can compete with visual agents. This underscores the potential for sensory substitution to expand the adaptability and applicability of reinforcement learning agents across diverse environments and sensory modalities. However, further advancements in auditory encoding methods are necessary to unlock the full potential of this approach, particularly in visually complex and high-dimensional tasks like CarRacing-v2.

Chapter Six

Conclusions and Future Works

6.1 Conclusions

In this study, we explored the potential of sensory substitution in reinforcement learning by transforming visual inputs into auditory representations. This was achieved using the **VisualToAuditoryWrapper** in both **CarRacing-v2** and **LunarLander-v2** environments, leveraging the **Proximal Policy Optimization (PPO)** algorithm to train agents capable of learning from auditory cues derived from visual information.

Our experiments demonstrated that while sensory substitution is a promising concept, the performance of the auditory agent generally lagged behind the visual agent, particularly in **CarRacing-v2**. The visual agent consistently outperformed the auditory agent in terms of reward accumulation due to the high-dimensional nature of visual tasks in the environment. On the other hand, in **LunarLander-v2**, a simpler task environment, the auditory agent demonstrated competitive or even superior performance in some episodes, suggesting that auditory substitution is more feasible in environments with fewer spatial and continuous control demands.

These results highlight the inherent challenges in sensory substitution for complex, visually-oriented tasks but also show that agents can still learn and adapt using auditory cues. Our work contributes to understanding the complexities of multimodal reinforcement learning, showing that agents can be trained to operate in environments relying on alternative sensory inputs. This study opens new pathways for creating more adaptable AI systems that can perceive and act in a variety of sensory modalities.

The findings underscore several important conclusions:

- **CarRacing-v2** revealed the limitations of sensory substitution, where spatial awareness and continuous control favoured visual input, making it difficult for the auditory agent to match the visual agent’s performance.
- **LunarLander-v2** showed more promise for sensory substitution, where the auditory agent's performance was comparable to the visual agent. This highlights the potential of using auditory cues in discrete control tasks.

- The PPO algorithm, combined with the sensory substitution approach, was generally effective at training both types of agents, though there is clear room for improvement in auditory processing models to better capture critical features of the environment.

Overall, our study paves the way for further exploration of sensory substitution in reinforcement learning, suggesting that with more advanced sensory transformation techniques, agents could be developed to perform well across diverse and complex environments, ultimately contributing to the broader goal of creating intelligent agents that can perceive and act in the world through multiple senses.

6.2 Future Work Suggestions

Based on the findings discussed, the following Future Works are forwarded: Even though the results from this study were encouraging, further RL algorithm and environment should be undertaken for this experiment, furthermore the following key area is pointed out for future work

Enhanced Sensory Transformations: Explore more sophisticated methods for transforming visual information into auditory representations. This could involve leveraging deep learning techniques such as attention mechanisms or transformer architectures to improve the fidelity and richness of auditory features derived from visual inputs.

Adaptive Sensorimotor Integration: Investigate adaptive strategies for integrating sensory inputs from multiple modalities in reinforcement learning agents. This could include dynamic weighting mechanisms that adjust the importance of visual versus auditory cues based on environmental conditions or task demands.

Exploration of Alternative RL Algorithms: Evaluate the performance of alternative reinforcement learning algorithms beyond PPO. Algorithms like Deep Q-Learning (DQN) or Actor-Critic methods may offer different trade-offs in terms of stability, sample efficiency, and convergence rates for training agents in multimodal environments.

Extended Training Periods: Allowing for longer training periods and more diverse training scenarios to enable auditory agents to refine their strategies and improve performance consistency.

Transfer Learning and Generalization: Assess the agent's ability to generalize learning across different environments and tasks. Transfer learning techniques could be applied to leverage

pre-trained models from simpler environments and fine-tune them for more complex tasks, thereby accelerating learning and improving performance.

Real-World Applications: Extend the application of sensory substitution models to real-world scenarios where individuals with sensory impairments could benefit. Collaborate with experts in assistive technology to validate the effectiveness of sensory substitution approaches in enhancing human-machine interactions.

Multi-Agent Systems: Explore the dynamics of multi-agent systems where agents with different sensory modalities collaborate or compete to achieve common goals. Investigate how sensory substitution could facilitate cooperation or competition strategies among agents in complex, multi-agent environments.

Interpretability and Explainability: Develop methods for interpreting and explaining agent decisions based on multimodal inputs. This could involve visualizing attention mechanisms or saliency maps to understand how the agent integrates and prioritizes information from different sensory channels.

Ethical Considerations: Address ethical considerations surrounding the deployment of AI systems using sensory substitution, particularly regarding privacy, bias, and equitable access. Engage in discussions with stakeholders to ensure responsible and inclusive deployment of sensory substitution technologies.

By pursuing these avenues for future research, we can advance the capabilities of sensory substitution in reinforcement learning, paving the way for more adaptive and effective artificial intelligence systems that leverage diverse sensory inputs to achieve complex goals in various environments.

References

[1] Ahmed, H. (2018). Introduction to artificial intelligence. University of Mansoura.

[2] GeeksforGeeks.(2024, August 21).Artificial intelligence: An introduction. GeeksforGeeks. https://www.geeksforgeeks.org/artificial-intelligence-an-introduction/

[3] Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., & Mordatch, I. (2019). Emergent tool uses from multi-agent autocurricula. arXiv preprint arXiv:1909.07528.

[4] H. Li, Wei, Ren, Zhu, & Wang, 2017

[5] Ye, Y., Wang, K., Hu, W., Li, H., Yang, K., Sun, L., & Chen, Z. (2019, May). A wearable vision-to-audio sensory substitution device for blind assistance and the correlated neural substrates. In Journal of Physics: Conference Series (Vol. 1229, No. 1, p. 012026). IOP Publishing.

[6] Seeing with sound. (n.d). Seeing with Sound. https:// Seeingwithsound.com/

[7] Levy-Tzedek, S., Hanassy, S., Abboud, S., Maidenbaum, S., & Amedi, A. (2012). Fast, accurate reaching movements with a visual-to-auditory sensory substitution device. Restorative neurology and neuroscience, 30(4), 313-323.

[8] Jaafra, Y., Laurent, J. L., Deruyver, A., & Naceur, M. S. (2019). Reinforcement learning for neural architecture search: A review. *Image and Vision Computing*, *89*, 57-66.

[9] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern recognition*, *77*, 354-377.]

[10] AlMahamid, F., & Grolinger, K. (2021, September). Reinforcement learning algorithms: An overview and classification. In *2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)* (pp. 1-7). IEEE.

[11] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

[12] Arroyo, J., Manna, C., Spiessens, F., & Helsen, L. (2021, September). An open-ai gym environment for the building optimization testing (boptest) framework. In *Building Simulation 2021* (Vol. 17, pp. 175-182). IBPSA.]

[13] C. J. Watkins and P. Dayan, “Q-learning,” Machine learning, vol. 8, no. 3-4, pp. 279–292, 1992.

[14] G. A. Rummery and M. Niranjan, On-line Q-learning using connection ist systems. University of Cambridge, 1994, vol. 37.

[15] D. Zhao, H. Wang, K. Shao, and Y. Zhu, “Deep reinforcement learning with experience replay based on sarsa,” in IEEE Symposium Series on Computational Intelligence, 2016, pp. 1–6.

[16] A. Anwar and A. Raychowdhury, “Autonomous navigation via deep reinforcement learning for resource constraint edge nodes using transfer learning,” IEEE Access, vol. 8, pp. 26549–26560, 2020.

[17] H. Van Hasselt, “Double Q-learning,” Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, pp. 1–9, 2010.

[18] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). Openai gym. *arXiv preprint arXiv:1606.01540*.

[19] Farama Foundation. (n.d.). *Gymnasium*. Farama Foundation. https://gymnasium.farama.org/

[20] Bach-y-Rita, P., & Kercel, S. W. (2003). Sensory substitution and the human–machine interface. Trends in cognitive sciences, 7(12), 541-546.

[21] R. Arandjelovic, A. Zisserman, Look, listen and learn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 609–617.

[22] D. Harwath, A. Recasens, D. Sur´ıs, G. Chuang, A. Torralba, J. Glass, Jointly discovering visual objects and spoken words from raw sensory input, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 649–665

[23] A. Nagrani, S. Albanie, A. Zisserman, seeing voices and hearing faces: Crossmodal biometric matching, in: Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian, 2018, pp. 8427–8436

[24] A. Senocak, T. H. Oh, J. Kim, M. H. Yang, I. So Kweon, learning to localize sound source in visual scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4358–4366.

[25] H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, A. Torralba, The sound of pixels, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 570–586

[26] R. Arandjelovic, A. Zisserman, Objects that sound, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 435–451.]

[27] S. Ginosar, A. Bar, G. Kohavi, C. Chan, A. Owens, J. Malik, Learning individual styles of conversational gesture, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3497–3506.]

[28] C. H. Wan, S. P. Chuang, H. Y. Lee, Towards audio to scene image synthesis using generative adversarial network, in: 2019 IEEE International Conference on Acoustics, Speech and Signal Processing, 2019, pp. 496–500.

[29] T. H. Oh, T. Dekel, C. Kim, I. Mosseri, W. T. Freeman, M. Rubinstein, W. Matusik, Speech2Face: Learning the face behind a voice, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7539– 7548.

[30] L. Chen, S. Srivastava, Z. Duan, C. Xu, Deep cross-modal audio-visual generation, in: Proceedings of the on Thematic Workshops of ACM Multimedia 2017, 2017, pp. 349–357.

[31] W. Hao, Z. Zhang, H. Guan, Cmcgan: A uniform framework for cross-modal visual-audio mutual generation, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 6886–6893.

[32] Y. Zhou, Z. Wang, C. Fang, T. Bui, T. L. Berg, Visual to sound: Generating natural sound for videos in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3550–3558.

[33] A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, W. T. Freeman, Visually indicated sounds, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2405–2413.

[34] K. Chen, C. Zhang, C. Fang, Z. Wang, T. Bui, R. Nevatia, Visually indicated sound generation by perceptually optimized classification, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 560–574.]

[35] L. Liu, J. Tang, X. Wan, Z. Guo, Generating diverse and descriptive image captions using visual paraphrases, in: 2019 IEEE International Conference on Computer Vision, 2019, pp. 4239–4248.

[36] S. Ding, S. Qu, Y. Xi, S. Wan, Stimulus-driven and concept-driven analysis for image caption generation, Neurocomputing 398 (2020) 520–530. [36]Ning, H., Zheng, X., Yuan, Y.,

& Lu, X. (2021). Audio description from image by modal translation network. *Neurocomputing*, *423*, 124-134.]

[37] Wang, Y., Wan, Y., Zhang, C., Bai, L., Cui, L., & Yu, P. (2019, November). Competitive multi-agent deep reinforcement learning with counterfactual thinking. In *2019 IEEE International Conference on Data Mining (ICDM)* (pp. 1366-1371). IEEE.

[38] Baker6, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., & Mordatch, I. (2019). Emergent tool use from multi-agent interaction. *Machine Learning, Cornell University*.

[39] Wang, X., Qi, H., & Iyengar, S. S. (2002, July). Collaborative multi-modality target classification in distributed sensor networks. In *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002.(IEEE Cat. No. 02EX5997)* (Vol. 1, pp. 285-290). IEEE.

[40] Stable-Baselines3. (n.d.). Custom policies. Stable-Baselines3. https://stable-baselines3.readthedocs.io/en/master/guide/custom_policy.html

[41] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529.

[42] Piergigli, D., Ripamonti, L. A., Maggiorini, D., & Gadia, D. (2019, August). Deep Reinforcement Learning to train agents in a multiplayer First-Person Shooter: some preliminary results. In 2019 IEEE Conference on Games (CoG) (pp. 1-8). IEEE.

[43] Meijer, P. B. L. (1992). An experimental system for auditory image representations. IEEE Trans. Biomed. Eng. 39, 112–121.

[44] Capelle, C., Trullemans, C., Arno, P., and Veraart, C. (1998). A realtime experimental rototype for enhancement of vision rehabilitation using auditory substitution. IEEE Trans. Biomed. Eng. 45,

[45] Crony–Dillon, J., Persaud, K. C., & Blore, R. (2000). Blind subjects construct conscious mental images of visual scenes encoded in musical form. Proceedings of the Royal Society of London. Series B: Biological Sciences, 267(1458), 2231-2238

[46] Cronly-Dillon, J., Persaud, K., & Gregory, R. P. F. (1999). The perception of visual images encoded in musical form: a study in cross-modality information transfer. Proceedings of the Royal Society of London. Series B: Biological Sciences, 266(1436), 2427-2433.

[47] Abboud, S., Hanassy, S., Levy-Tzedek, S., Maidenbaum, S., & Amedi, A. (2014). EyeMusic: Introducing a "visual" colorful experience for the blind using auditory sensory substitution. Restorative neurology and neuroscience, 32(2), 247-257.

[48] Ward, J., & Meijer, P. (2010). Visual experiences in the blind induced by an auditory sensory substitution device. Consciousness and cognition, 19(1), 492-500.

[49] Stronks, H. C., Nau, A. C., Ibbotson, M. R., & Barnes, N. (2015). The role of visual deprivation and experience on the performance of sensory substitution devices. Brain research, 1624, 140-152.

[50] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529-533.

[51] Li llicrap, T. P. "Continuous control with deep reinforcement learning." *arXiv preprint arXiv:1509.02971* (2015).

[53] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press.

[54] Gulcehre, C., Wang, Z., Novikov, A., Le Paine, T., Gomez Colmenarejo, S., Zolna, K., ... & Paduraru, C. (2020). Rl unplugged: Benchmarks for offline reinforcement learning. arXiv eprints, arXiv–2006.

[55] Hao, J., Yang, T., Tang, H., Bai, C., Liu, J., Meng, Z., ... & Wang, Z. (2023). Exploration in deep reinforcement learning: From single-agent to multiagent domain. IEEE Transactions on Neural Networks and Learning Systems.