**AUTO-PLAYING GAME USING NEURL NETWORKS**

**Presented by-**

Rahul Agarwal

Tanishk Annapurna

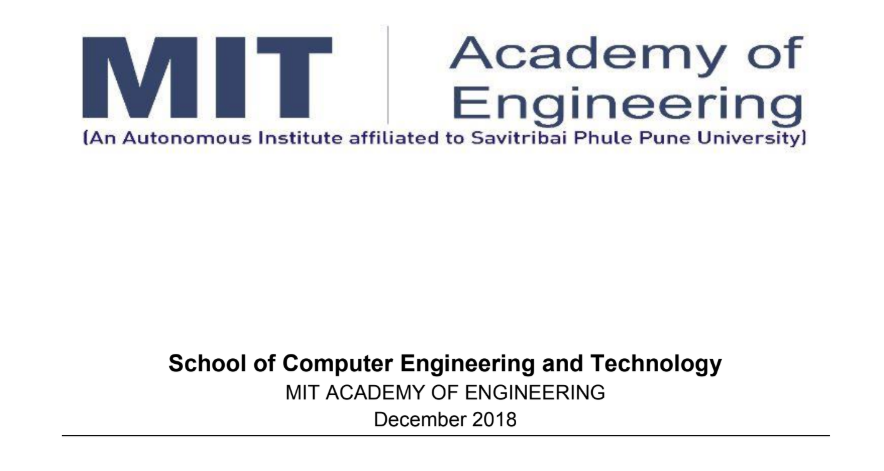
Shardul Dubey

Suraj Tannalli

**Under the supervision of-**

Mr. Tushar Kute

Ms. Pranali



**ACKNOWLEDGEMENT**

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations.

We would like to extend our sincere thanks to all of them. We would like to express our deep gratitude to Professor Ms. Pranali Lokhande and Mr. Tushar Kute for their patient guidance, enthusiastic encouragement and useful critiques of Project.

We would also like to extend our thanks to the technicians of the laboratory of the School of Computer Engineering and Technology for their help in offering us the resources in running the project.

Our thanks and appreciations also goes to our colleagues in developing the project and people who have willingly helped us out with their abilities.

21st December, 2018

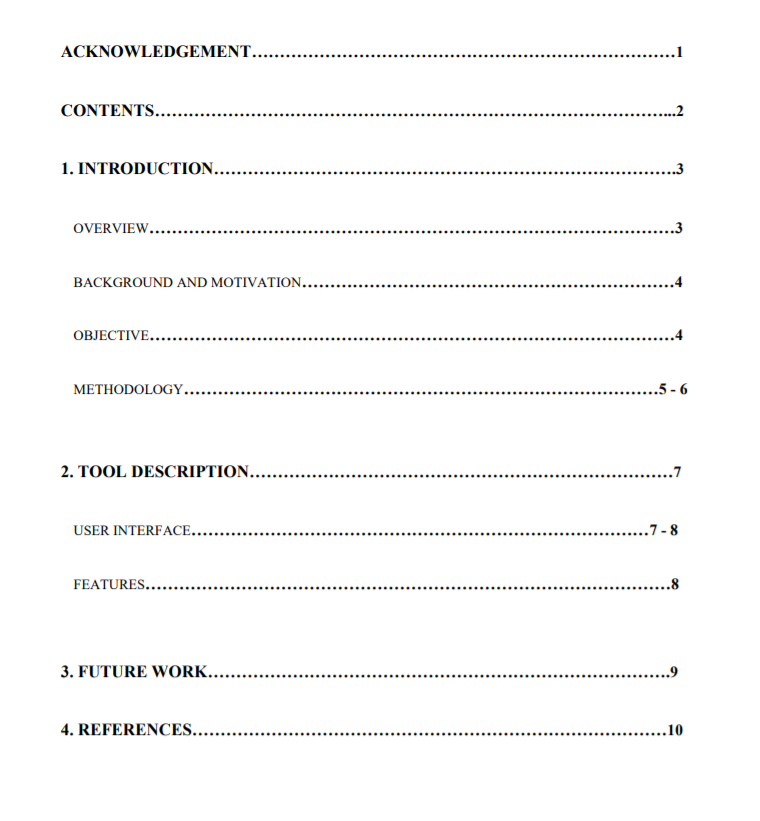
Rahul Agarwal

Tanishk Annapurna

Shardul Dubey

Suraj Tannalli

**CONTENT**



**1. INTRODUCTION**

OVERVIEW-

Most of us play games or have played a game in our life in a smartphone or a computer. These games have sometimes complex playing situations or some of them employ very basic system controls where only a tap on the smartphone screen or a single key stroke of the computer system is required.

Also these games have an obstacle type course or some of them have a very easy puzzle mode. Taking these games in mind, this project concentrates on the automation or self playing of such a game.

This project employs deep learning concepts to totally automate the playing of the popular dino-game from the chrome browser – teaching the dino slowly and steadily where to avoid obstacles and where to do nothing, the game slowly progresses which increasing the high score on the boards.

**BACKGROUND AND MOTIVATION –**

In [computer science](https://en.wikipedia.org/wiki/Computer_science), **Artificial intelligence** (**AI**), sometimes called **machine intelligence**, is [intelligence](https://en.wikipedia.org/wiki/Intelligence) demonstrated by [machines](https://en.wikipedia.org/wiki/Machine), in contrast to the **natural intelligence** displayed by humans and other animals. Computer science defines AI research as the study of "[intelligent agents](https://en.wikipedia.org/wiki/Intelligent_agent)": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. More in detail, [Kaplan](https://en.wikipedia.org/wiki/Andreas_Kaplan) and Haenlein define AI as “a system’s ability to correctly interpret external data, to learn from such data, and to use those learnings to achieve specific goals and tasks through flexible adaptation”. Colloquially, the term "artificial intelligence" is applied when a machine mimics "cognitive" functions that humans associate with other [human minds](https://en.wikipedia.org/wiki/Human_mind), such as "learning" and "problem solving".

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. Deep learning excels on problem domains where the inputs (and even output) are analog. Meaning, they are not a few quantities in a tabular format but instead are images of pixel data, documents of text data or files of audio data.

[Yann LeCun](https://en.wikipedia.org/wiki/Yann_LeCun) is the director of Facebook Research and is the father of the network architecture that excels at object recognition in image data called the [Convolutional Neural Network (CNN)](http://machinelearningmastery.com/crash-course-convolutional-neural-networks/). This technique is seeing great success because like multilayer perceptron feedforward neural networks, the technique scales with data and model size and can be trained with back propagation.

The main motivation behind this project can be said to the “OPEN AI” project pioneered by Elon Musk. OpenAI's mission is to build safe AGI, and ensure AGI's benefits are as widely and evenly distributed as possible. OpenAI was deployed to learn a complex strategy game called “DOTA 2” where it was made to ‘learn’ the game and after it ‘learned’ the game it was made to play against world champions for the respective game and it turned out that the AI could compete with the champions on equal grounds.

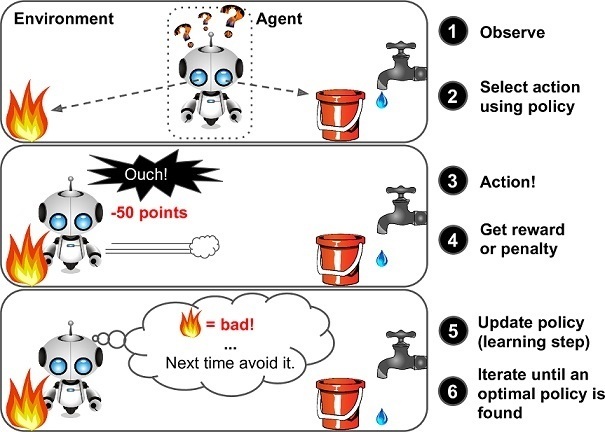


Fig 1. A vanilla Reinforcement Learning framework

**Q-learning –**

We use Q-learning, a technique of RL, where we try to approximate a special function which drives the action-selection policy for any sequence of environment states. Q-learning is a model-less implementation of Reinforcement Learning where a table of Q values is maintained against each state, action taken and the resulting reward. A sample Q-table should give us the idea how the data is structured. In our case, the states are game screenshots and actions, do nothing and jump[0,1]

| **q-table** |
| --- |
| *Fig 2. A sample Q-table* |

We take advantage of the Deep Neural Networks to solve this problem through regression and choose an action with highest predicted Q-value.

**OBJECTIVES –**

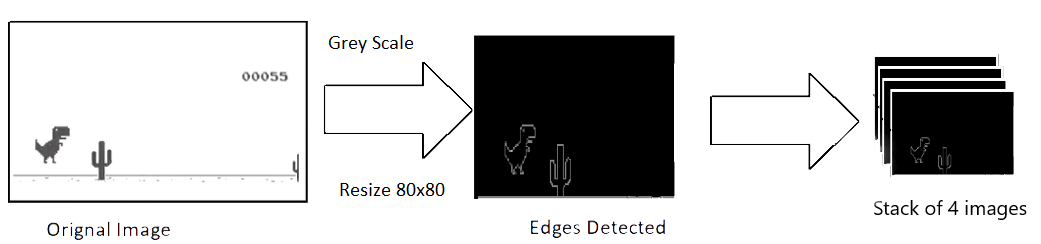
* To develop a program that will use deep learning concepts and implement them.
* To make a game run on itself without getting defeated and successfully run continuously making new high scores without human intervention.

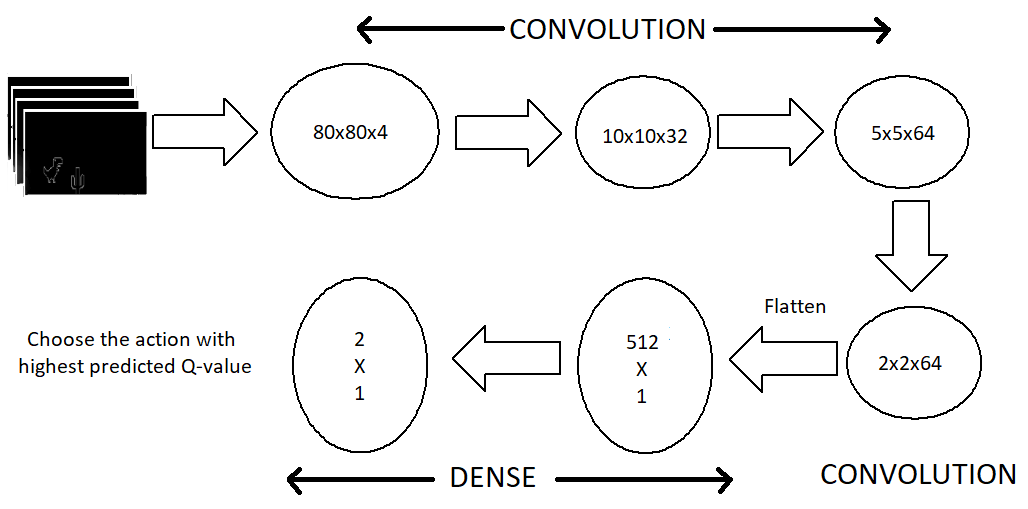
**METHODOLOGY –**

Our approach is quite simple and basic. By using various libraries and also using drivers, we import the game into the code from the chrome browser and then extract numerical data from the running game while simultaneously updating the Q-table and thus steadily increasing the alpha variable which further makes the game progress.

* Initially, importing the game from the browser using the chromium drivers.
* Taking random values into the Q-table initially.
* By making the game play itself, we collect the numerical data like score, time, failures and successful jumps and update these values into the Q-table respectively.
* To do this we use an image processing algorithm that take continuous screenshots of the game being played at a rate of 4fps and converts it into an 8-bit image which becomes easy and light for the processor to process and extract data from the completely processed images.
* This process continuously updates the alpha variable increasing the success rate of the game not ending.

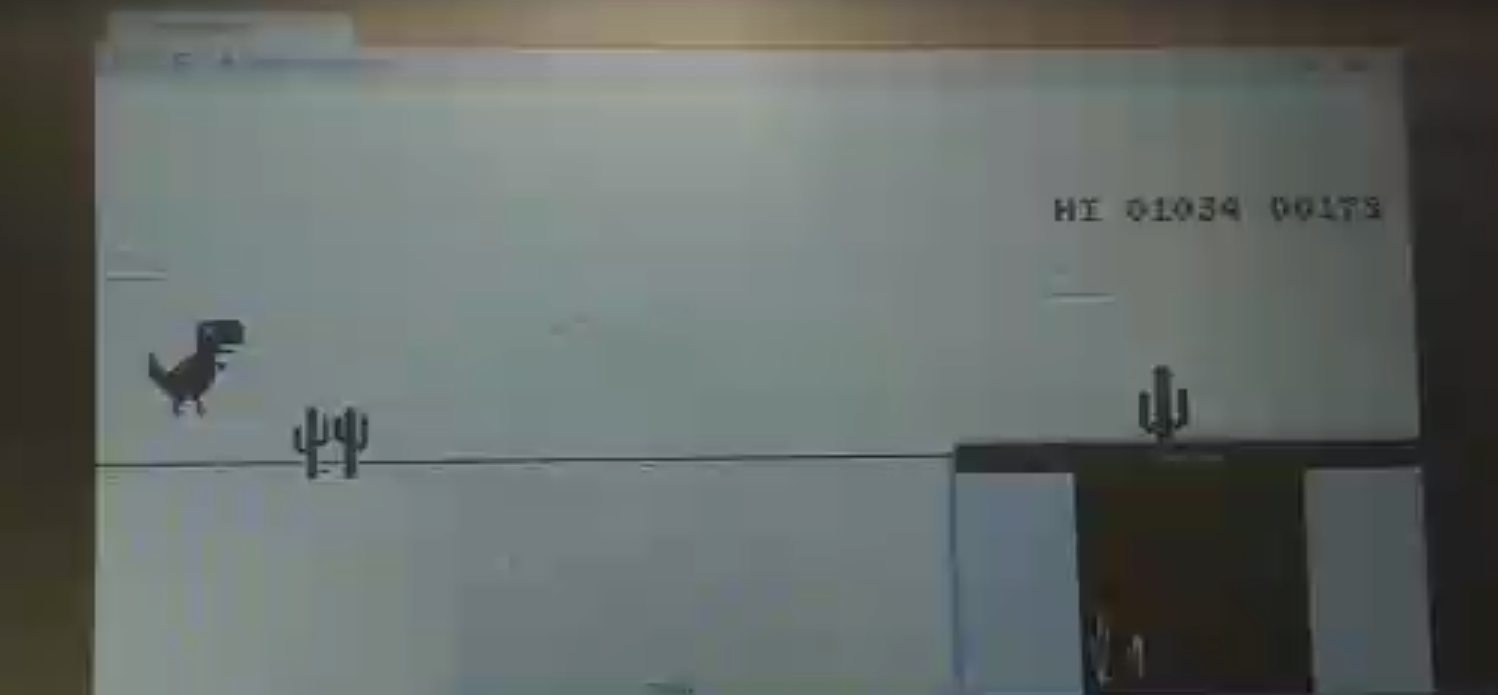
**2. TOOL DESCRIPTION**





*Fig 3. Image processing*

**USER INTERFACE**



*Fig 4. The game actually running and the Q-table being updated*

**FEATURES**

* Simple to Use - The simple and intuitive graphical user interface ensures that users can take full advantage with little or no ramp-up time.
* Accurate - The program works by analyzing entire game and giving a very high alpha value making the game unbeatable in the end.

**3. FUTURE WORK**

The program will be used to learn a particular game and play that specific game on its own. This program can be tweaked accordingly and be used to play any game. This program can also be used for playing a game but also can be used to do a lot of other things. This program after introducing new variables can also perform various complex tasks like simulations and also be deployed in the real world after sufficient ‘learning’.

**4. REFERENCES**

* <https://en.wikipedia.org/wiki/Artificial_intelligence>
* <https://machinelearningmastery.com/what-is-deep-learning>
* <https://openai.com/about/>
* <https://www.google.co.in/amp/s/blog.paperspace.com/dino-run/amp/>