Playing a class of Game using CNN

Focus on Runner Games

Jimut Bahan Pal[1]

[1]Department of Computer Science
Ramakrishna Mission Vivekananda Educational and Research Institute

A presentation for Communicative English course

Table of Contents

Table of Contents

Introduction

This was a pet project during my 2nd year on St. Xavier's College!

- Motivation

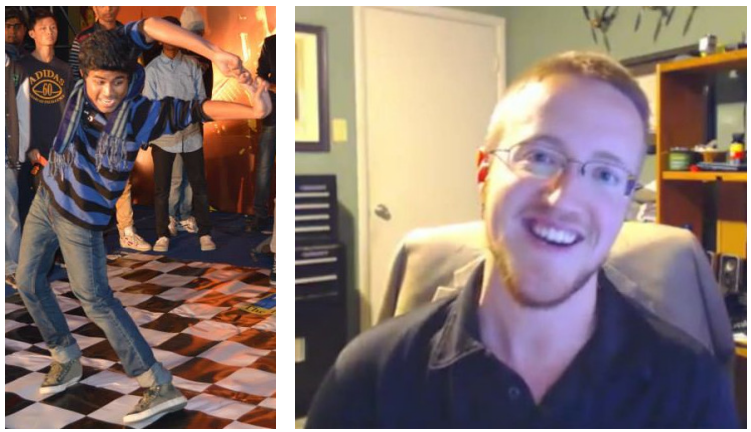


Figure: Indranil and Harrison

Introduction

This was a pet project during my 2nd year on St. Xavier's College!

- Motivation

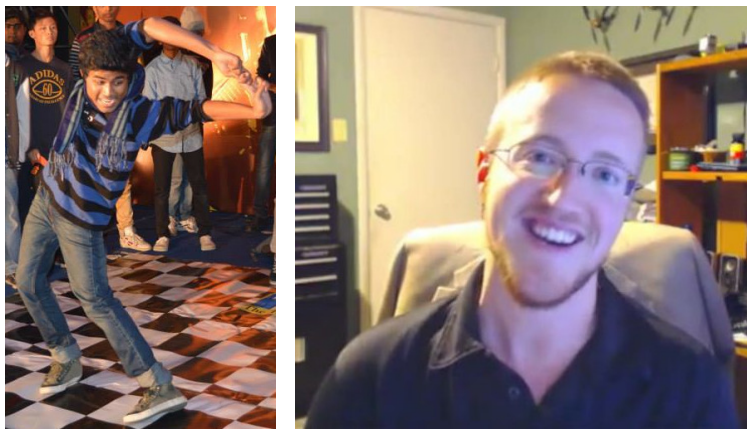


Figure: Indranil and Harrison

- CNN - to analyze visual imagery, face detection, earlier application include detection of numbers for post cards (Yann Lee Cunn)

Introduction

This was a pet project during my 2nd year on St. Xavier's College!

- Motivation

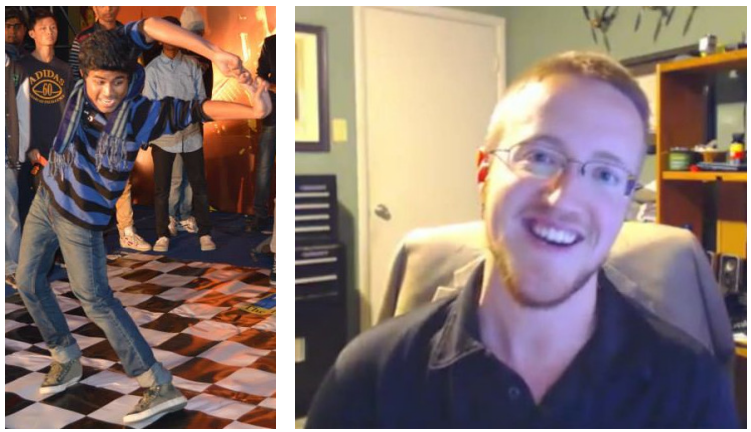


Figure: Indranil and Harrison

- CNN - to analyze visual imagery, face detection, earlier application include detection of numbers for post cards (Yann Lee Cunn)
- Cifar 10 model, 80% accuracy in 2010 on CIFAR-10 dataset

Introduction

This was a pet project during my 2nd year on St. Xavier's College!

- Motivation

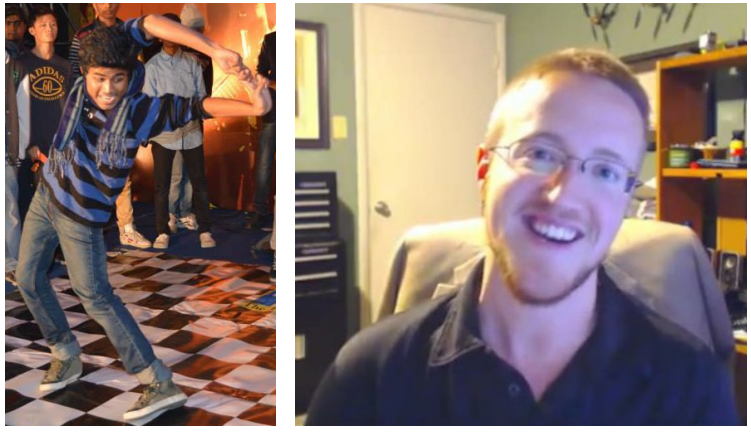


Figure: Indranil and Harrison

- CNN - to analyze visual imagery, face detection, earlier application include detection of numbers for post cards (Yann Lee Cunn)
- Cifar 10 model, 80% accuracy in 2010 on CIFAR-10 dataset
- Can we make something innovative using simple technology?!

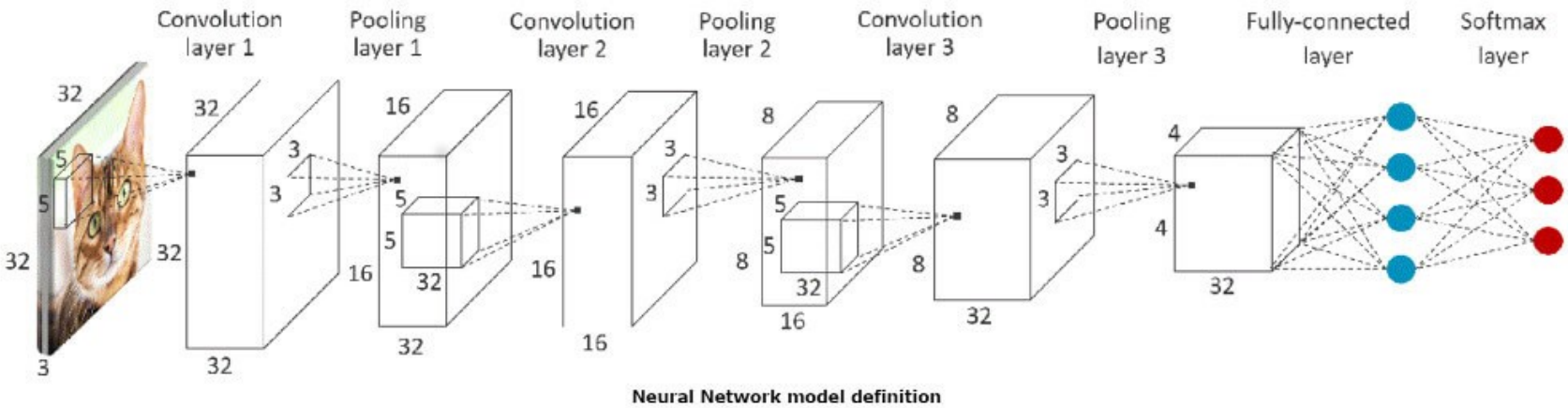


Figure: The famous CIFAR-10 model which we used for training

Table of Contents

Selection of a wrong game!

The game that came to our mind - PONG!
Pong - Simplest table tennis video game from 1972 atari console.

Figure: Uprighted Cabinet of Pong

Selection of a wrong game!

The game that came to our mind - PONG!
Pong - Simplest table tennis video game from 1972 atari console.

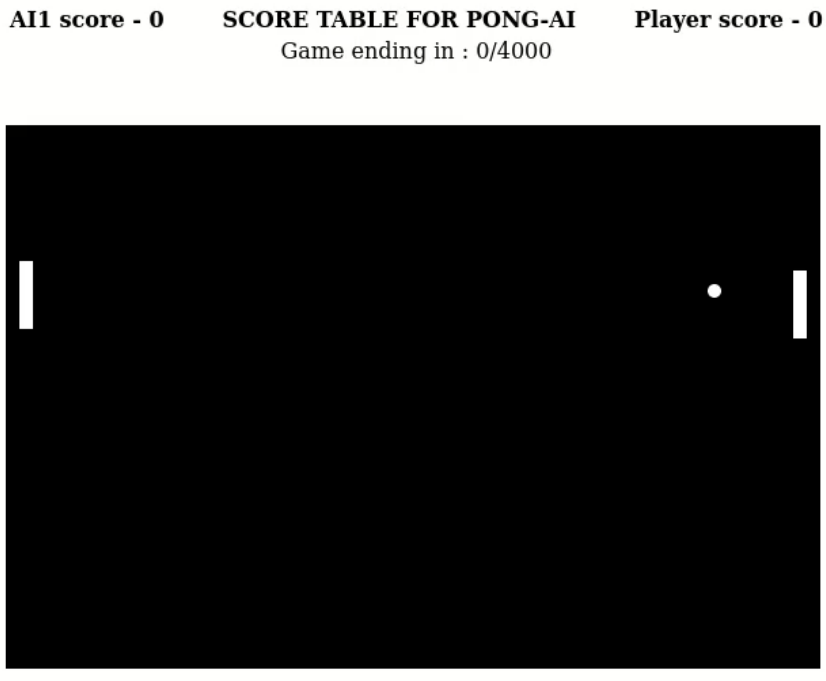


Figure: Our JS implementation of Pong

Figure: Data used in training of Pong Game: Frames and corresponding action values (0 - down and 1 - up)

https://colab.research.google.com/drive/1ECPOkDGplF6cBA5EuLKLfJHfM8f0ubz7#scrollTo=9DTGfNpyeKhP

Wiki | Most Visited | Forum | Wiki | Getting Started | Mozilla News | Mozilla News | Wiley journals on algo... | OSF Preprints | Wisp: ...

dino_cnn.ipynb

File Edit View Insert Runtime Tools Help

+ Text

```
Epoch 14/30
1960/1960 [==============================] - 36s 19ms/step - loss: 0.1160 - acc: 0.9582 - val_loss: 0.1777 - val_acc: 0.9000
Epoch 15/30
1960/1960 [==============================] - 37s 19ms/step - loss: 0.1146 - acc: 0.9561 - val_loss: 0.1978 - val_acc: 0.9750
Epoch 16/30
1960/1960 [==============================] - 36s 19ms/step - loss: 0.1139 - acc: 0.9541 - val_loss: 0.1693 - val_acc: 0.9000
Epoch 17/30
1960/1960 [==============================] - 36s 18ms/step - loss: 0.1061 - acc: 0.9622 - val_loss: 0.1772 - val_acc: 0.9000
Epoch 18/30
1960/1960 [==============================] - 36s 19ms/step - loss: 0.1060 - acc: 0.9592 - val_loss: 0.1705 - val_acc: 0.9250
Epoch 19/30
1960/1960 [==============================] - 36s 19ms/step - loss: 0.0986 - acc: 0.9633 - val_loss: 0.1705 - val_acc: 0.9000
Epoch 20/30
1960/1960 [==============================] - 36s 18ms/step - loss: 0.0937 - acc: 0.9653 - val_loss: 0.1711 - val_acc: 0.9000
Epoch 21/30
1960/1960 [==============================] - 36s 19ms/step - loss: 0.0938 - acc: 0.9628 - val_loss: 0.1603 - val_acc: 0.9250
Epoch 22/30
1960/1960 [==============================] - 36s 18ms/step - loss: 0.0960 - acc: 0.9612 - val_loss: 0.1641 - val_acc: 0.9500
Epoch 23/30
1960/1960 [==============================] - 36s 18ms/step - loss: 0.0933 - acc: 0.9658 - val_loss: 0.1882 - val_acc: 0.9000
Epoch 24/30
1960/1960 [==============================] - 36s 19ms/step - loss: 0.0919 - acc: 0.9673 - val_loss: 0.1920 - val_acc: 0.9250
Epoch 25/30
1960/1960 [==============================] - 36s 19ms/step - loss: 0.0893 - acc: 0.9684 - val_loss: 0.1826 - val_acc: 0.9250
Epoch 26/30
1960/1960 [==============================] - 36s 18ms/step - loss: 0.0880 - acc: 0.9643 - val_loss: 0.1934 - val_acc: 0.9250
Epoch 27/30
1960/1960 [==============================] - 37s 19ms/step - loss: 0.0807 - acc: 0.9694 - val_loss: 0.1812 - val_acc: 0.9250
Epoch 28/30
1960/1960 [==============================] - 36s 18ms/step - loss: 0.0843 - acc: 0.9673 - val_loss: 0.1821 - val_acc: 0.9250
Epoch 29/30
1960/1960 [==============================] - 36s 19ms/step - loss: 0.0787 - acc: 0.9714 - val_loss: 0.1979 - val_acc: 0.9000
Epoch 30/30
1960/1960 [==============================] - 36s 19ms/step - loss: 0.0847 - acc: 0.9699 - val_loss: 0.2228 - val_acc: 0.9000
```

```
import matplotlib.pyplot as plt
print(history.history.keys())
# summarize history for accuracy
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
```

Figure: When training in Google Collaboratory Platform

```
20000/20000 [==============================] - 210s
Epoch 22/50
20000/20000 [==============================] - 209s
Epoch 23/50
14144/20000 [====================>.........] - ETA:
```

Inspector | Console | Debugger | Network | Style

Filter Output

```
allow pasting
SyntaxError: unexpected token: identifier [Learn More]
function ClickConnect(){
console.log("Working");
document.querySelector("colab-toolbar-button#connect").click()
}
setInterval(ClickConnect,60000)
16465
Working
Working
```

Figure: A simple hack to never stop collaboratory running

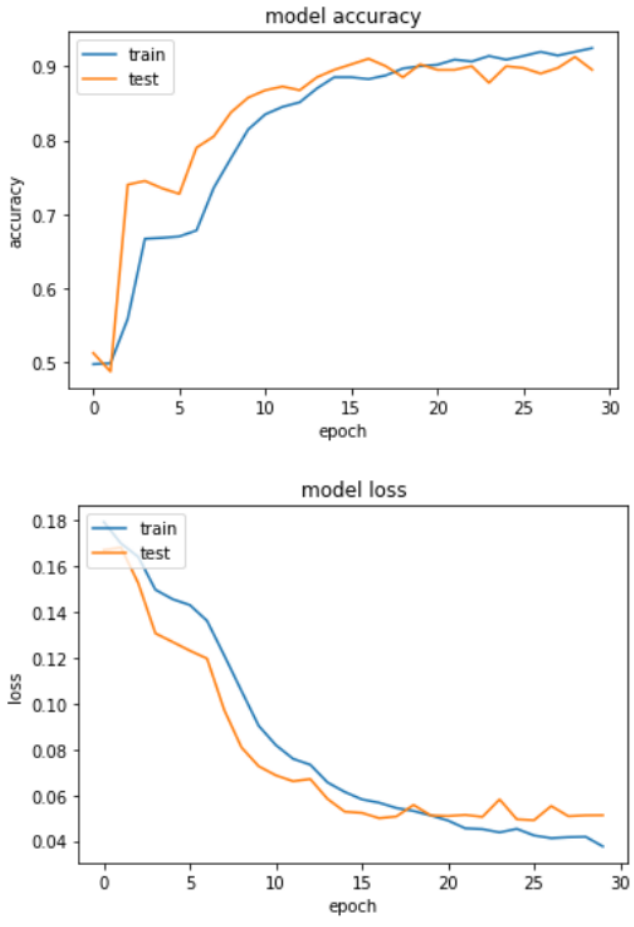


Figure: Loss and accuracy of the Pong game trained with 2K images

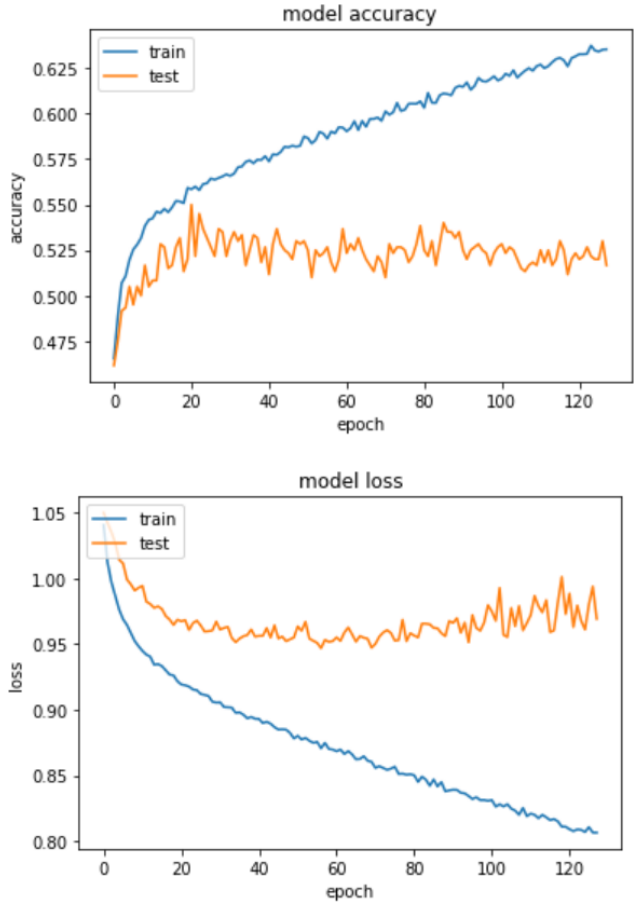


Figure: Loss and accuracy of the Pong game when trained with 30K images

Table of Contents

Why pong failed?

Pong's Computer counterpart is a robot not an AI, it just calculates according to coordinates of the ball. We needed human touch!

- More data more accuracy, what is wrong?

Why pong failed?

Pong's Computer counterpart is a robot not an AI, it just calculates according to coordinates of the ball. We needed human touch!

- More data more accuracy, what is wrong?
- We trained the model using Google's free collaboratory platform, which gives free GPU and is computationally effective.

Why pong failed?

Pong's Computer counterpart is a robot not an AI, it just calculates according to coordinates of the ball. We needed human touch!

- More data more accuracy, what is wrong?
- We trained the model using Google's free collaboratory platform, which gives free GPU and is computationally effective.
- Guess the move by looking at the Pong's picture

Table of Contents

Runner Games

Runner games are those kind of games which have a definite move for every instance of the environment

Figure: Famous Dino game!

Runner Games

Runner games are those kind of games which have a definite move for every instance of the environment

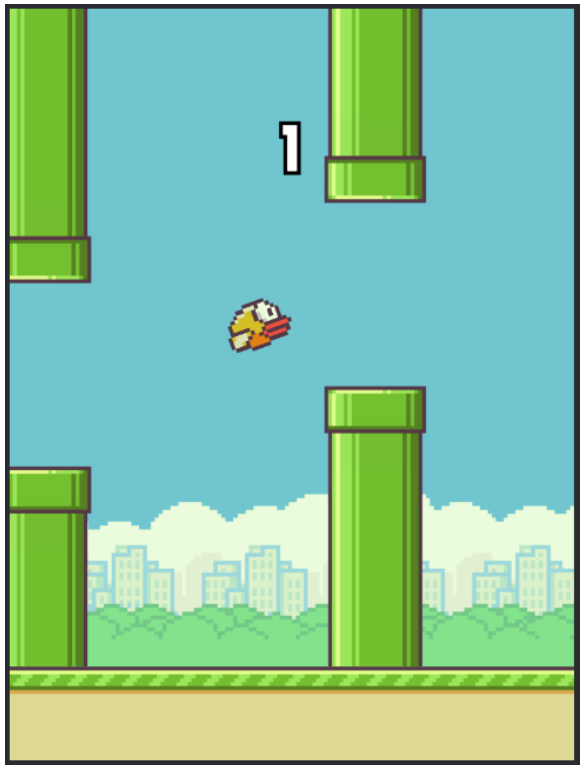


Figure: Famous flappy bird game!

Runner Games

Runner games are those kind of games which have a definite move for every instance of the environment

Figure: Asphalt overdrive game

Runner Games

Runner games are those kind of games which have a definite move for every instance of the environment

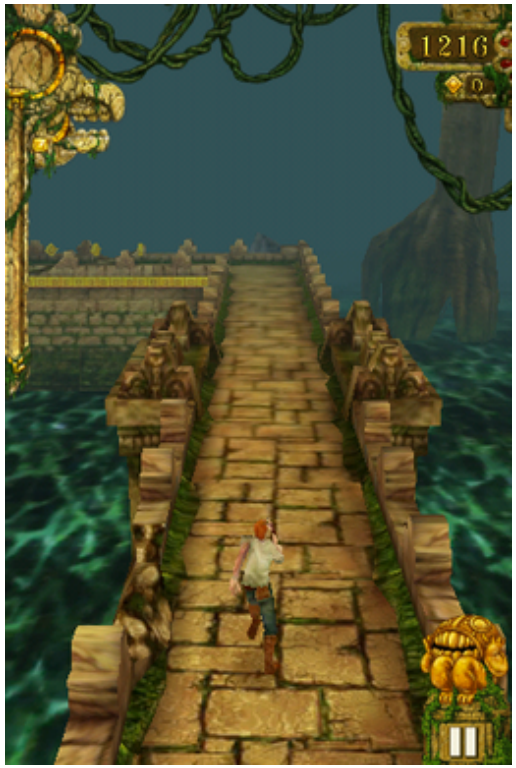


Figure: Famous Temple run game

Runner Games

Runner games are those kind of games which have a definite move for every instance of the environment

Figure: The Road Rash

Runner Games

Runner games are those kind of games which have a definite move for every instance of the environment

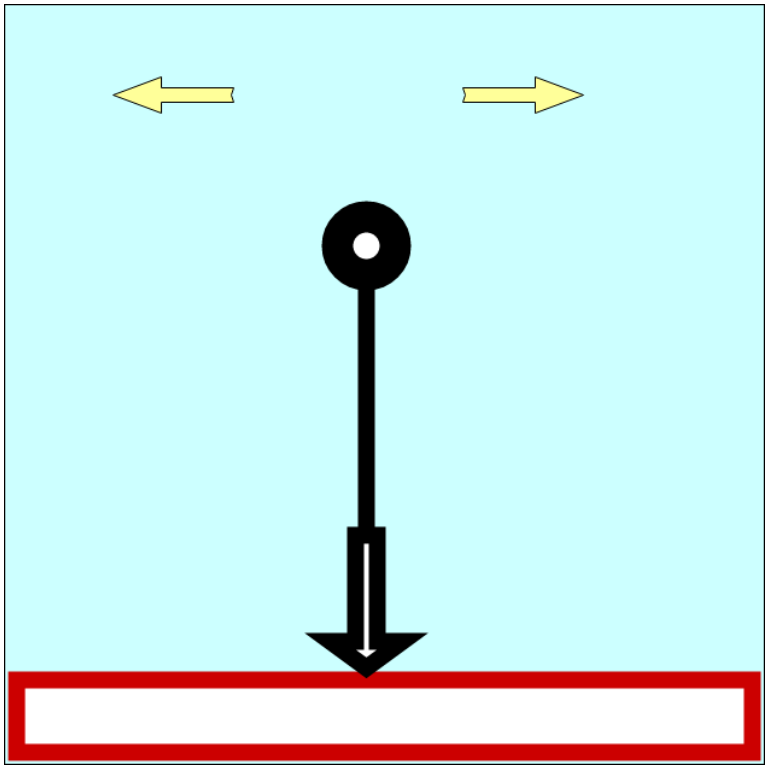


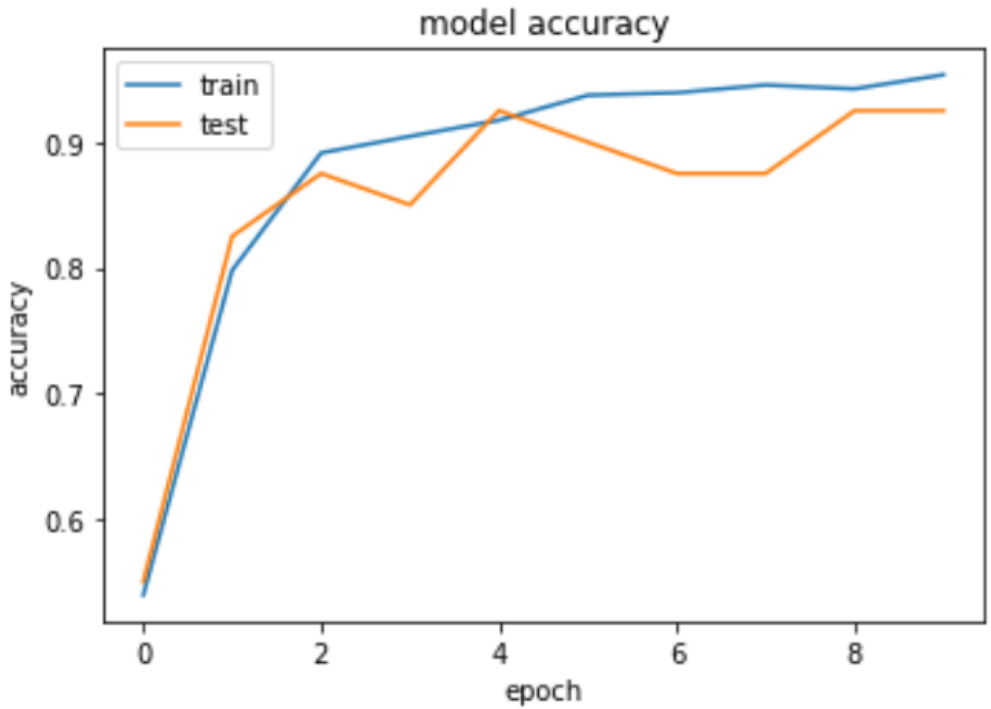
Figure: A custom made game named "Balancer"

Figure: Data used in training of Dino Game: Frames and corresponding action values (0 - nothing and 1 - up)

Table of Contents

Results for the Dino game

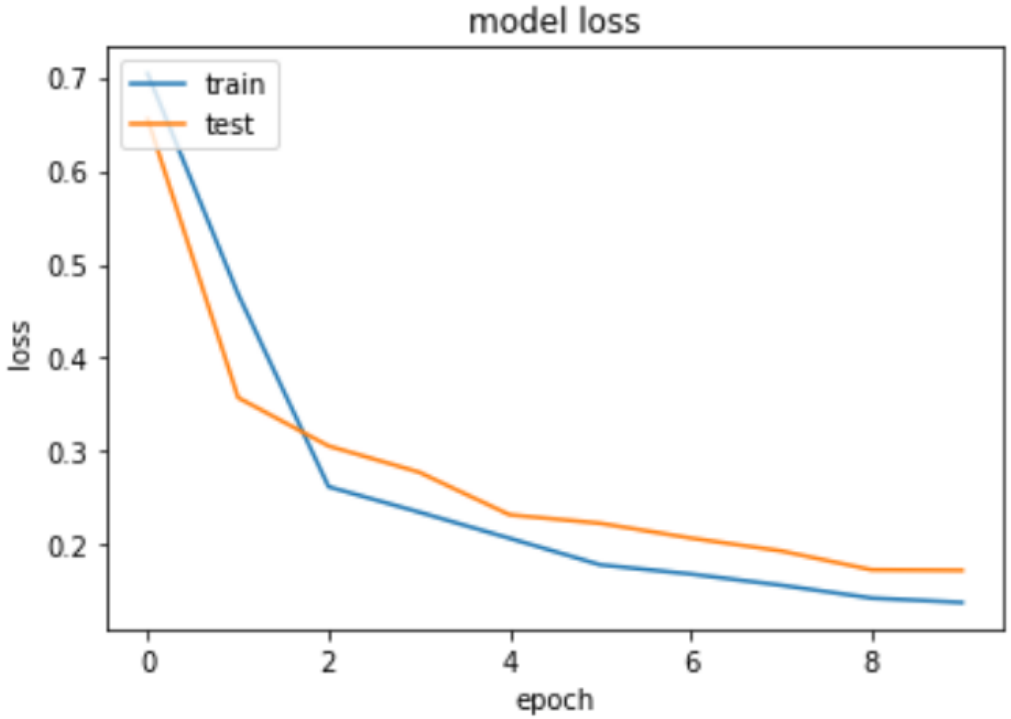
The results obtained are satisfactory. It performed well and runs genuinely with 90% accuracy. From these results we can conclude that certain class of games performs well with just a simple technology like CNN.



Accuracy obtained from Dino game training

Results for the Dino game

The results obtained are satisfactory. It performed well and runs genuinely with 90% accuracy. From these results we can conclude that certain class of games performs well with just a simple technology like CNN.



Loss from the dino game training

Table of Contents

Conclusion

Modern technologies and algorithms like Reccurrent Neural Network, Reinforcement Learning, Genetic algorithms are more powerful than the method that was implemented. We will implement these in the Dino game in the near future. The YOLO works on this exact same model, i.e., extract frames from the video and predict from those taken pictures.

Table of Contents

Acknowledgements

I acknowledge the help recieved from Tamal Maharaj, Prof. Janardan Ghosh, Indranil Das and Harrison Kinseley for their suggestions and discussions.

Table of Contents

References

- Pal, J.B., (2019), Playing a class of games using CNN, Blog in github pages, https://jimut123.github.io/blogs/cnn_games_ai.html available on the web, last accessed on 14-11-2019 .
- Kinseley, H., (2017), Python Plays GTA V, Tutorial in https://pythonprogramming.net/game-frames-open-cv-python-plays-gta-v/ available on the web, last accessed on 14-11-2019 .

Thank You

jimutbahanpal@{gmail,yahoo,outlook}.com