

# DIP Project

## Google Dino Game With Hand Gestures

Saad Bazaz                      i180621  
Abraar Raza Samar            i180544

*Submitted to: Mr Shoaib Mehboob*

Digital Image Processing, Spring 2021

Bachelors of Computer Science, National University of Computing and Emerging Sciences, Islamabad.

### Versions:

#### **Version 1: Dino Game with hand gestures using mediapipe**

##### Explanation

In this version of the game we are using the mediapipe import. This library provides many ML related solutions. From this import we used Holistics to identify a person's right hand and obtain landmarks on a right hand. This gave us the positions of the person's right hand relative to the camera recording the person. It gave us x, y, and z coordinates of the person's hands where the x and y coordinates had been normalized between 0 and 1 across the width and length of the camera screen. So according to mediapipe coordinates, at the top of the camera screen the y coordinate was 0 and at the bottom of the screen the y coordinate was 1.

So to play the game, we set two horizontal boundaries across the screen one at 0.3 and the other at 0.7. So the 0.3 one is near the top of the screen and the 0.7 one is closer to the bottom.

Now since every time the hand is moved on the camera, its coordinates change accordingly and we are getting these coordinates thanks to the holistics. Now whenever the y coordinate of the hand went below the 0.3 line i.e the hand was moved up, the dino will jump. As long as the hand is between the 0.3 line and the 0.7 line the dino keeps running normally. Now when the hands y coordinate goes above the 0.7 line i.e the hand is moved towards the bottom of the screen, the dino ducks.

## **Version 2: Dino Game with hand gestures using opencv and fingers count**

### **Explanation**

In this version of the game, we use opencv and background subtraction technique to isolate the hand image from the background through segmentation.

**Background Subtraction:** To do background subtraction, we first let the computer save the background on which we will demonstrate the game using hand gestures. We show the computer the background and let it save the background to memory by letting it take the running average of the background only for 30 frames. After 30 frames have passed, the background has now been saved by the code. Now we may introduce our hand into the camera. Now what happens is that once the camera sees the hand in front of the background, the frame is read and stored. This frame is subtracted from the stored background and the absolute of this subtraction matrix is taken. Now the pixels having very small values tell us that there wasn't any major change in the newly read frame and the stored background. However the pixels that have large values tell us that in this pixel there has been a big change in the background. So basically the pixels having large values represent the hand and all other places are the same old background. After getting the subtraction matrix we threshold it to convert it into a binary matrix with 0's in places where the background remained the same and 1's in places where a new foreground object was introduced.

**Continue:** Now after thresholding we take the background subtracted and thresholded image and find contours in the image. Contours are basically the curves/lines formed by joining those points that lay on a boundary having the same intensity values. Using opencv functions we are easily able to obtain the contours. We then find the contour having the maximum area, and we will be assuming that our hand region will have the maximum area in the camera region of interest. Now using the cv2 convex hull function we are able to get the extreme points of the hand on camera. Now by finding the mean of opposite laying coordinates we are able to find the center of the palm. We now construct a circle around the center of the palm and see where the circle and fingers intersect. The number of times the circle intersects with the fingers will be the number of fingers open. Now we just made the dino jump, duck and keep running with 1, 2 and no fingers.

### **Version 3: Dino Game with hand gestures using SVM**

**Explanation:** In this version we again do background subtraction and everything up to the part where we draw a circle around the hand. Now what we do is, we record some frames with our

hands open, and extract the features of the hand open frames, then we record some frames with our hands closed and extract the features of those frames. Our feature vector included

```
feature_vector = np.array([
    hull_mean,
    hull_std,
    maximum_distance,
    minimum_distance,
    distances_mean,
    distances_std,
    radius,
    circumference,
    circular_roi_mean,
    circular_roi_std,
    np.mean(cnts_mean),
    np.mean(cnts_std),
    count
], dtype=np.float32)
```

These values, which were calculated for every sample we took by recording. We then trained the model using these feature vectors and made predictions by the model.

#### References:

<https://gogul.dev/software/hand-gesture-recognition-p1>

<https://gogul.dev/software/hand-gesture-recognition-p2#contours>

[https://www.youtube.com/watch?v=QkO\\_3absfdw](https://www.youtube.com/watch?v=QkO_3absfdw)

<https://google.github.io/mediapipe/>