CHAPTER 1

INTRODUCTION

1.1 PURPOSE:

In our project “Target Detection and Tracking” the main aim is to detect any specific object in the video stream. Recognize and Identify the moving objects for (specific) interest, and to track those moving objects throughout the video frame. It can be widely used in computer vision task such as face detection, face recognition, video object co-segmentation. It is also used in Object Detection, Object Tracking, Object Classification, Video Surveillance, Background Modelling for example tracking a ball during a football match, tracking movement of a cricket bat, tracking a person in a video, detect the movement of car, bike truck in games and real life, to detect a specific target in surveillance video footage. This project also aims to solve some real time problem for example the issue that video feeds can’t be processes in real time and we cannot track the object in real time with accuracy. This model can be used as a standalone tracking system and can be used as a part of defense grade program to track and target a specific object. Detecting of target can also be used in medical fields like to detect skin cancer from a patient, we can make bots that can selectively extract cells from our selected area without making any harm to other areas.

1.2 OBJECTIVE:

Recognize and Identify the moving objects for (specific) interest, and to track those moving objects throughout their life spans. This project aims for the existence challenging issue in the area of surveillance and security. For this we need to create dataset that have wide range of variety so that it can detect those part of the object that are not in the previous dataset that will be done only by using a specific and correct algorithm and by choosing correct images for our first dataset. We need to create an algorithm that will detect our specific object in a fast and accurate manner (able to detect a specific object at a speed of nearly 30 frame per second) so that it will be used in our video stream.

1.3 MOTIVATION:

- The motive of object detection is to recognize, locate and track a known object in a scene.
- Difficult to track an object with incomplete datasets so, our motive is to design an algorithm than can update the datasets time to time.
- Preferably in 3D space, recovering pose of objects in 3D is very important for robotic control systems.
- The information from the object detector can be used for obstacle avoidance and other interactions with the environment.

1.4 Definition

The object detection is identifying the particular object in an image and target tracking is to detect object in the video frame stream. This project was tested on the tennis ball but can be trained on different datasets of about 1500 images and can be used widely used in different fields. This project only identifies tennis ball and rest of the images as unknown.

1.5 Model Overview

This project is implementation of Inception-ResnetV2. It is one of the fastest algorithms present now. This model was presented in the paper by Google.Inc on 23 Aug 2016. In 2012 when AlexNet was presented and it was found out that deeper neural networks are needed for classification. But processing it takes more time. So, it is tradeoff between accuracy and speed.
Google came up with new approach and divide the model in modules. This approach is also used in the Inception-ResnetV2. This model is divided into following modules:

Following figures shows schematic diagram of the model:

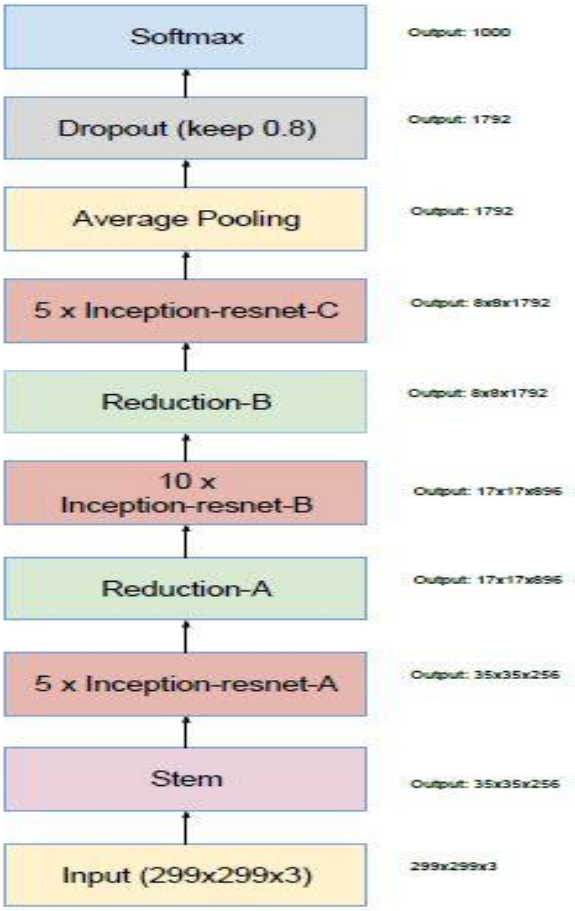

Fig 1.1 Schema Diagram [1]

It takes input image of size 299*299*3 in stem cell of the model. The following figure represents the Stem cell of the model.

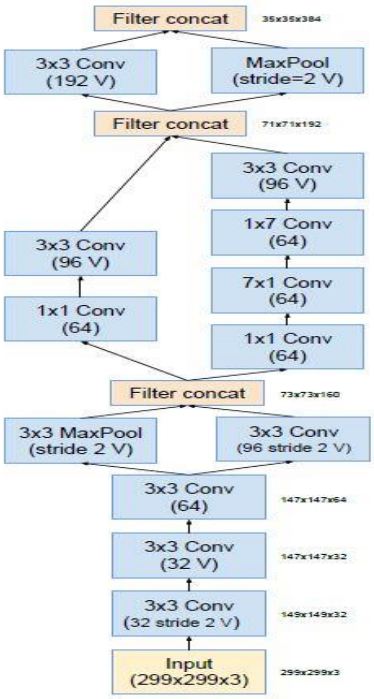

Fig 1.2 Stem cell [1]

It is divided into two branches and so that processing can be divided and then results of them are concatenated. Then it passes its output to the module A showed in following figures

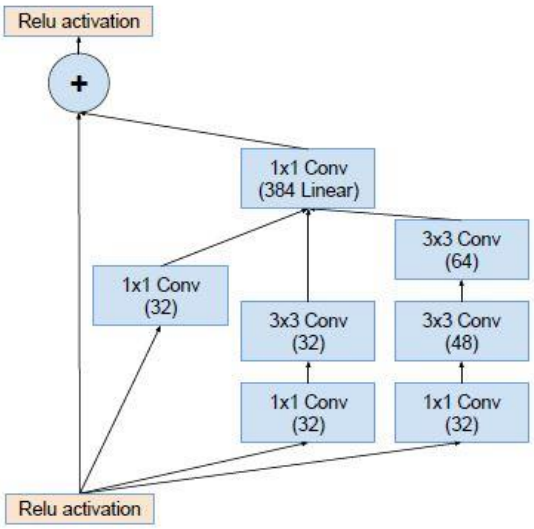

Fig 1.3 Module A [1]

After this output of this layer is passed to the Reduction A layer showed by figure

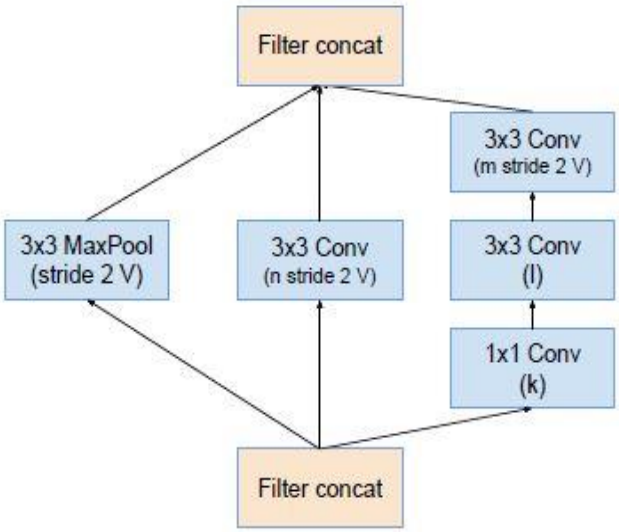

Fig 1.4 Reduction A layer [1]

After this output of this layer is passed to the Inception-Resnet B layer showed by figure

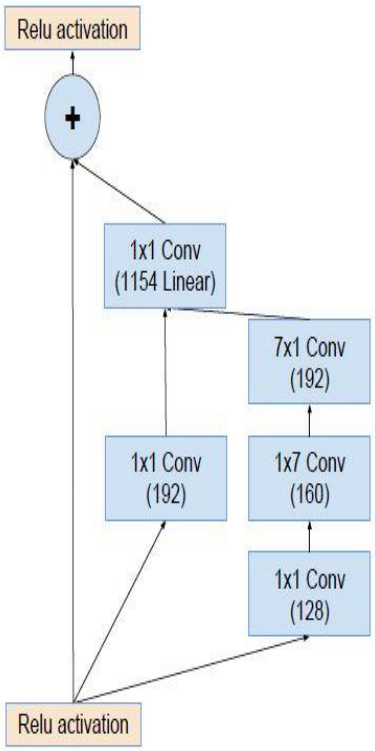

Fig 1.5 Module B [1]

The output of this layer is passed to the Reduction B layer shown in the following figure

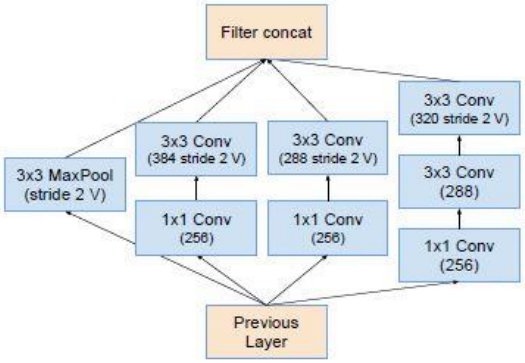

Fig 1.6 Reduction Layer [1]

This layer output is passed onto the Inception-Resnet-C layer shown in the following figure

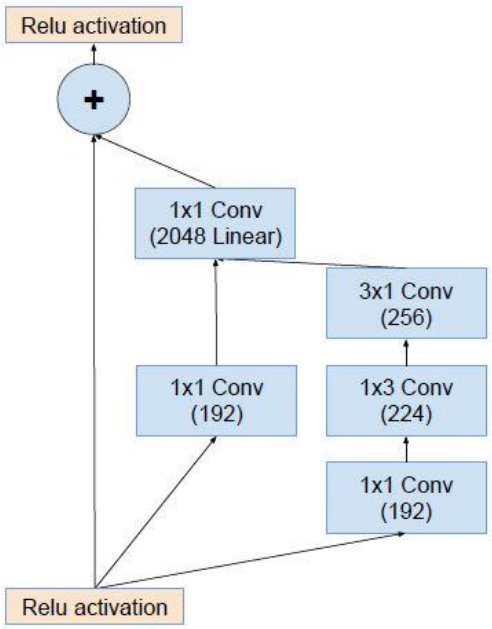

Fig 1.7 Module C [1]

We take output of this layer and do Average Pooling then do Dropout (keep 0.8) and the passed it to the SoftMax classifier and we get the output of the predicted classes.

CHAPTER 2

OVERALL DESCRIPTION

2.1 Project Perspective:

2.1.1 Developers View:

As per developers' points of view we design a system which can detect and track object in image and video frames, which help users for object classification and track those objects about their life spans. And it is easy to train model for different data sets and can be widely used.

2.1.2 Users View:

As per user point of view, users can use our system in multisector such as defense and security surveillance, gamming and virtualization, medical diagnosis and sports, to classify, detect and track images and video frames and made it fast and simpler for ease of use for the users.

2.2 Project Functions:

2.2.1 Image Classifications:

Our project basic functionality is to classify images on basis of trained datasets as per loaded into system during classifications. It can classify images with 78% of accuracy.

2.2.2 Video Tracking:

Our project secondary functionality is to track and record video frames and label it during executions, through out the object life span.

2.3 DFD Diagrams:

2.3.1 Image Classification:

Fig 2.1 Level 0

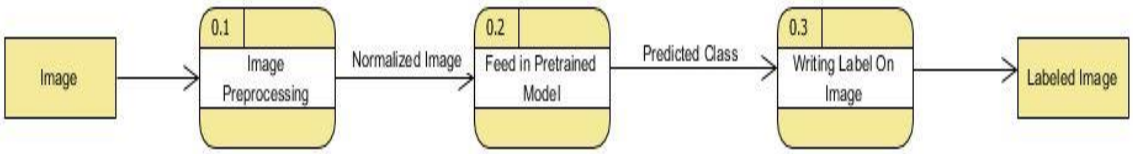

Fig 2.2 Level 1

2.3.2 Object Tracking:

Fig 2.3 Level 0

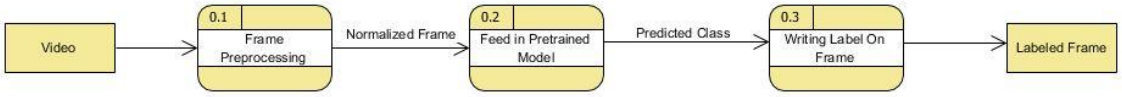

Fig 2.4 Level 1

2.4 Use Case Diagram:

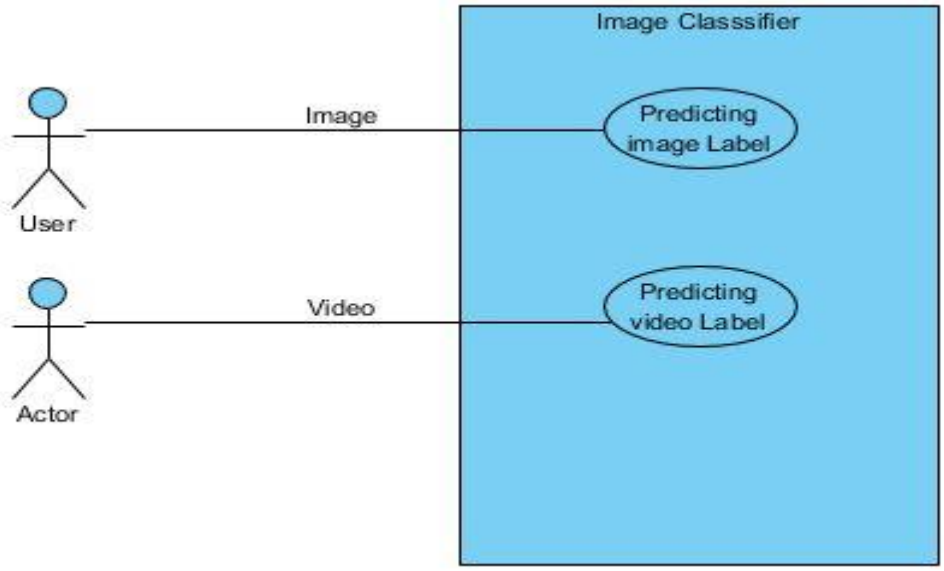

Fig 2.5 Use case

Text Description:

User 1: Will input image in the use case and will get labeled image.

User 2: Will input video in the use case and will get output labeled video.

CHAPTER 3

EXTERNAL INTERFACE REQIRMENTS:

3.1 HARDWARE INTERFACE:

Processor: Intel® Core™ i7-6500U CPU @ 2.50GHz 2.60GHz

Installed memory (RAM): 16.00 GB

GPU: Nvidia K8

System type: 64-bit operating system, x64-based processor

3.2 SOFTWARE INTERFACE:

Platform: OS - Windows 8 and above /Mac/Linux

Tools: Colab, Jupiter notebook, python kernel, python library

CHAPTER 4

SYSTEM FEATURES

4.1 System Basic Feature:

- **Platform Independent**

  Our program is platform independent, it can run on any platform having python libraries.

- **Stable**

  Our code is stable can be run on any python compiler or can be run online without any difficulties.

- **Adaptable**

  Our program can adapt itself with any dataset loaded or trained and can work according to it.

- **Simple design**

  Our code is simple in design can be understand or use by anyone, who have basic knowledge of python.

- **Faster**

  Our system is faster as compared to other models with a greater efficiency.

- **Accurate**

  Our program has 78.87% accuracy with great operating power.

CHAPTER 5

OTHER NON-FUNCTIONAL REQUIREMENTS

5.1 Performance Requirements:

Our system should have enough memory and speed to process near about 12 million parameters and operations between them and able to store data sets and should be able to process images and video frames.

5.2 Software Quality Attributes:

5.2.1 Interoperability:

Our program works in any operating system having python environments.

5.2.2 Maintainability:

Our model can be change according to different data sets and can be trained accordingly as per required.

5.2.3 Usability:

Our project main purpose is to use it in multiple sectors such as defense and security surveillance, gamming and virtualization, medical as well as in sports etc.

5.2.4 Efficiency:

Our algorithm gives 18.7% of one error & 4.5 % of top 5 errors compare to others, it is very less and process it faster.

5.2.5 Classifications:

Our model can classify object and can track them separately in image and video streams.

5.3 TESTING REQUIREMENTS:

5.3.1 DEBUGGING:

- **Using debuggers:**

  We used in built debuggers in python console to identify major errors like syntax and semantics errors. After all the syntax and semantics errors were removed, we setup check points in our program which is divided into following segments**.**

  - **Data Preprocessing:**
    In this segment we check whether our data is processed correctly or not, by checking image array shape and size.

  - **Model Defining:**
    In model defining, we define CNN architecture and checked whether it is compiled correctly or not by using model.summary(), which provide summary of our CNN architecture.

  - **Training Model:**
    In this segment our model was trained so we use different batch size in our model as 8, 16, 32, 64 so we can get highest batch size without running out of CPU and GPU memory. It was important to find the right size of the batch so that it won't over fit or won't crash our system.

  We have use print statement in every check point to rectify & correct our program segment wise.

- **Backtracking:**

  We start from the point where problem occurred and go back through the code in sequence to rectify where error has been occurred and then to add corrective measures in the segment, so that the part can run properly and the proceed the same process further, until the whole program is checked and corrected.

5.3.2 Performance Testing:

- The following graph shows over fitting of our model.

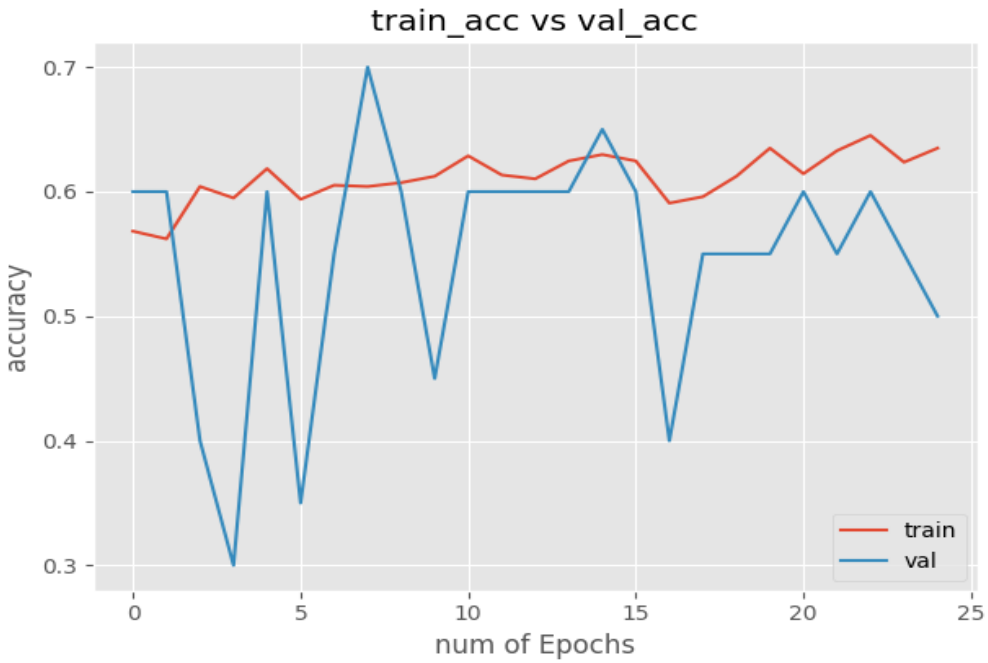

Fig 5.1 Train acc and val acc

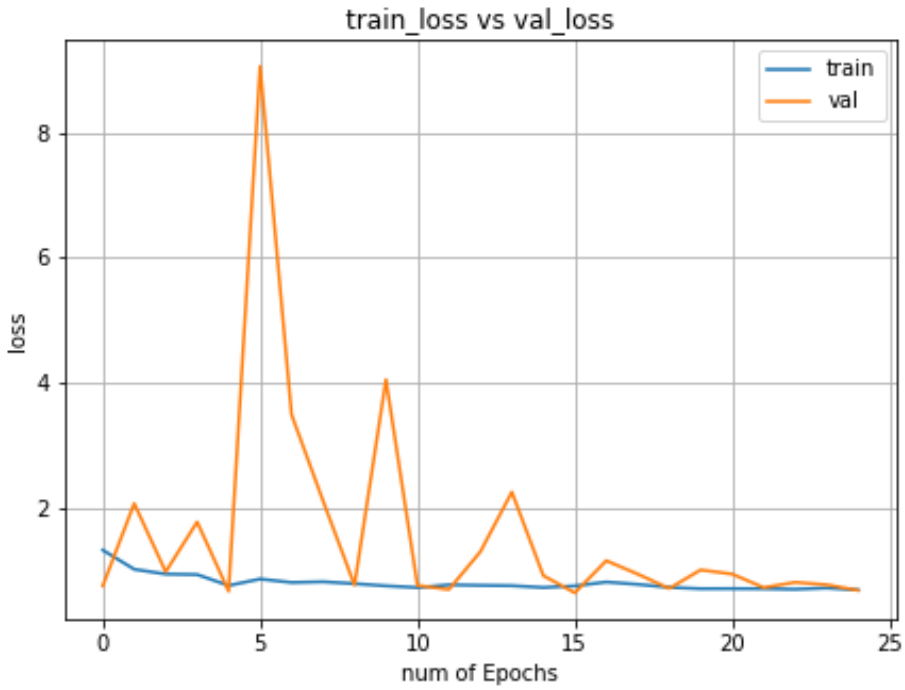

Fig 5.2 Train loss vs val loss

- We removed this overfitting by increasing number of samples in our datasets and also increase number of batches size and trained it again and got the following graph

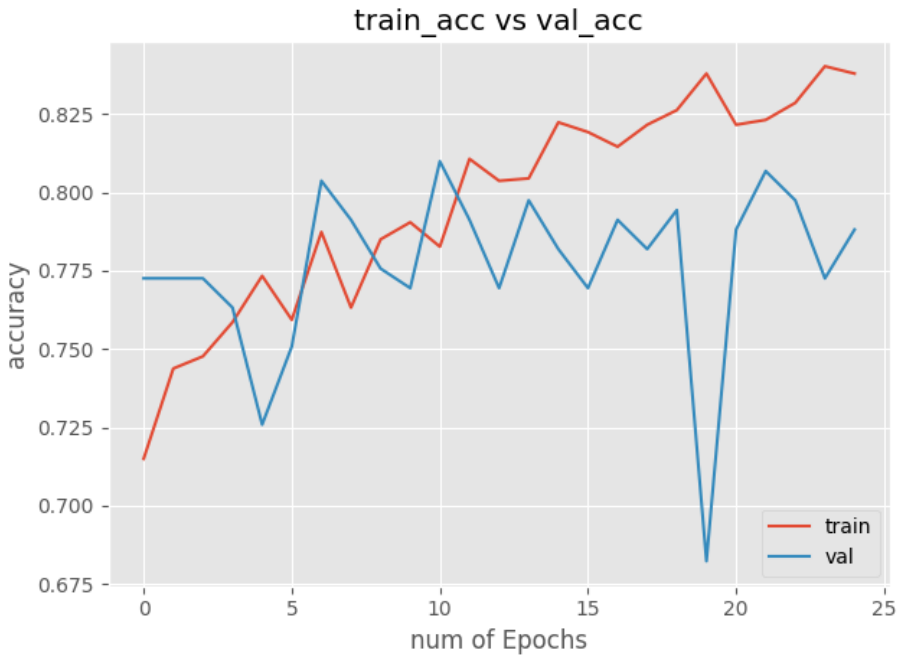

Fig 5.3 Train acc vs val acc

Fig 5.4 train acc vs val acc

- We also checked it accuracy against other models is more effective.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION:

In this project, a ball detection and tracking algorithm is presented for Object Detection, Object Tracking, Object Classification, Video Surveillance, Background Modelling sequence. We have presented our model based on research paper inception V4 and inception-resent that classify images and label videos based on the training it received it has many applications in field of medical diagnoses, defense, games and virtualization and have high accuracy and high speed compare to other model presented so far. For this project, we achieve with an object-tracking program that can automatically track multiple objects. It can also use to detect multiple object in wide range.

6.2 SCOPE FOR FUTURE WORK:

- **Processing time:** Full search in the frame takes large amount of time. We reduce searching time by searching only in some parts of the image. We use motion to trigger the search and search only in the moving region, However, this is still too slow for real time applications. Searching algorithm such as Inception V4 and Inception resent version 2 algorithm might be able to make this program faster because it reduces number of pixels to be searched.
- **Motion Analysis:** Another area that is very challenging. The segmented moving object from tracking can be further analyzed with the statistics of each motion to verify whether object is spreading or not.
- **Occlusion:** We do not include occlusion in this program. If occlusion occurs, tracking accuracy will be severely degraded. One way to fix it to use the template data of a couple of last frames, and keep track of the mean absolute difference or correlation in template matching. If the difference increases (or the correlation decreases) dramatically and abruptly, it suggests that occlusion already occurs. Then, the program could wait until the occlusion passes, or use an old template in a database instead of an occluded object.
- We can reduce computation by decreasing input size from 299 x 299 to 32x32.
- It can be used in research field such as for making bots that can selectively extract cells from our selected area without making any harm to other areas,

REFERENCES

1. https://arxiv.org/pdf/1602.07261.pdf