**CHAPTER 5**

**PROBLEMS AND DRAWBACKS**

Since we have taken a region of reference so the hand needs to be in that region of reference for this to work otherwise it will not be able to predict the gesture. There should be no moving objects in the background as we have used the background elimination technique, so if there is a moving object in back it will consider it too. The hand gestures need to be done appropriately for the system to guess it more properly.

The framework may run slower on certain PCs with low computational abilities since a lot of complex calculations are done in a very short span of time. Although a standard personal computer has the required computational power for ideal execution of the framework. Another problem is that if the resolution of the camera is too high then the project may run at a slower speed. This issue can be resolved by decreasing the resolution of the captured picture by making changes in the framework.

**CHAPTER 6**

**CONCLUSION**

We have developed a hand gesture-controlled mouse pointer by using web camera. We have used OpenCV and Tensorflow software to implement our project. Along with OpenCV we coded in Python language, we used convolutional neural network to recognize different gestures and we have used PyAutoGUI, an inbuilt library in python to programmatically control the mouse and keyboard. This technology has great applications in the fields of computer graphics, gaming, prosthetics, and many more. This technology can be used to help patients who are physically challenged. In case of computer graphics and gaming this technology has been applied in modern gaming consoles to create interactive games where a person’s motions are tracked and interpreted as commands. Most of the earlier applications developed required additional hardware which is often very costly. Our aim was to create this technology in the cheapest all possible way and to create it on a standardized operating system. Various application can be developed using this technology with the minimum requirement of resources.

**6.1 Results**

Most of the gestures have a precision of above 95%. The system can easily distinguish between each of the different gestures. The system provides good results in both bright and low lighting conditions. The system needs a stable background to run as it uses background elimination technique to detect the hand. Also, it is preferred to have a solid background colour.

**CHAPTER 7**

**FUTURE ENHANCEMENTS**

We would improve the execution of our project particularly the gesture controlling. Moreover, we additionally need to reduce the reaction time of the project for cursor development so that it can totally be utilized to supplant our regular mouse. We are also intending to structure an equipment execution for the equivalent in order to improve exactness and increase the usefulness to different spaces, for example, a gaming controller or as a universally useful PC controller. We should be able to detect the hand an any part of the screen and it could be at any distance from the camera. Some transitions could be made smoother. Since this will work as a substitute to the actual mouse, we should be able to carry out every function which is done by the mouse like drag, pinch zoom etc. Also, there should be a better background elimination technique to recognize moving backgrounds and eliminate them. Sometimes the gesture is not accurately detected, and fluctuation occurs which can cause wrong gestures to be detected. This can be improved by training the system more intensively which would require a very high performance processor.