Py-Glasses: Applications of Augmented Reality when combined with Artificial Intelligence

International Artificial Intelligence Fair Submission

Written By Flynn Cowell and Joshua Leach

Abstract

In this paper we try and look at the possible applications of Augmented Reality and Artificial Intelligence. Through this we built a prototype headset that we trained using artificial intelligence to assist a user in a variety of tasks, as a proof that the technology could be applied to a variety of uses. We used Python and Tensorflow as our software. As well as a Raspberry Pi and a VR Headset as our Hardware. During the course of the paper we describe, explain and analyze three case applications for these two emerging concepts: Medicine, Transport and Education. We explain the advantages and disadvantages using AI and AR with them, as well as how other things can be impacted. We also look at the ethical and privacy aspects of this technology as well.

Chapter 1

Introduction

Artificial Intelligence and Augmented Reality are often mentioned buzzwords in technology, and are often used as exciting points to what could be available in the future, we hope to look into them. In this article we try to use AI and AR to look at applications, such as in surgery, transport and education. Indeed many people are looking into these concepts, we hope to build on them, to develop our own "spin". We want this to be an open-source project, so that people can easily create their own programs for specific jobs. We are using Python to this as it is an extremely popular programming language, and is very easy to understand.

Chapter 2

Related Work

Our prior project for a different competition (PA Consulting Raspberry Pi) used a similar, however different workflow. The project was known as TrainBot, essentially it was a model train with a camera and distance sensor. The camera was able to recognise QR codes for it to change to different speed and the distance sensor was able to sense obstructions and stop the train until they had cleared.

Chapter 3

Implementation Method

I had an idea to use a readily available and easy to use programming language to create smart glasses that could perform a wide variety of functions . This language was going to be Python due to its simplicity and already existing libraries for this method.

I had previously created facial recognition and tracking software in Python, however this was using a long and ineffective way to detect an object (haarcascades - these are great for just one or two things, such as a face, however they are not very good at looking at a wide range of things). So for custom things (such as face ID) this method will work better.

For the main object detection i looked around and found a Python module named “Tensorflow” (For more information on how this works, there is an article here - https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html) This means that we can efficiently create object detection scripts.

So we are using a module Called “Cvlib” which uses Tensorflow’s vast library of objects alongside “OpenCV” to create object detection software, that we can call (i.e. say; when object x seen do ***).

This system is based off of a VR headset (the type you put your mobile phone into). This means that it is accessible very easily, and easy to use - as a lot of people have phones. Due to this needed a bit of computing power we intend on using a Raspberry Pi 3 to run the code, an added benefit is that this is small enough to easily fit onto the headset.

VR Headset

Chapter 4

Ethical Implications

We would like to make a point on the ethical implications of such technology, especially when it comes to the privacy of persons data collected through the camera when in use and who has access to that data, especially in certain applications, such as the medical and law-enforcement applications. Especially with regulations such as GDPR, it is important to consider the storage of images recorded with the headset, because of this we do not intend to work in a way to store

the images. Another concern is if the tech is used in War or by a criminal, something which could damage the reputation and lead to a more harmful world.

Chapter 5

Discussion

One of the points of this research is the applications of such a system. We have thought of three primary applications of our system.

1. Medicine and Surgery - The AI and Machine Learning, through the Computer Vision can be taught to recognize conditions, this can aid doctors and other medical professionals. in the diagnosis and treatment or someone's condition. This also can be used to aid surgeons in the course of their surgery identifying different parts.
2. Transport - (see Related Work) We previously built TrainBot, when you apply our system (modified of course), the AI could be taught to recognize various obstructions and work out how to counter them. With this it could be a good way to prevent unneeded accidents.
3. Education - This concept could be used to teach people, especially in more vocational subjects. For example it can be used to show different parts of a car, when teaching car mechanics. Because of this people's knowledge can be expanded in an engaging and informative way.

Chapter 6

Conclusion

In conclusion, we have created an open-source Python program that means we can detect objects, this can hopefully be used beyond its current state, and this system can be condensed even more. Due to this only being written in Python, this can work on any computer, laptop, or phone with a webcam. This means that almost anyone can create programs to be used with this system.

In future, I would love for this system to be used to save lives, in hospitals, even used by police forces to detect wanted criminals, however this system does not have a limit to what it can do, as long as there is a program for it, it has endless possibilities.

References:

Cvlib - {the module we are using for object detection}:
(documentation - https://docs.cvlib.net/object_detection/)
(website - https://www.cvlib.net/)
(Github - https://github.com/arunponnusamy/cvlib)

Tensorflow - {a module needed for Cvlib}:
(other module to co-inside with tensorflow -
https://github.com/devicehive/devicehive-video-analysis)
(website - https://www.tensorflow.org/)
(Github - https://github.com/opencv/opencv/wiki/TensorFlow-Object-Detection-API)

Server - {to produce images on the webpage}:
(Github - https://github.com/robot-websockets/openCV-server)

Yolo - {another way of object detection that we used before}:
(Github - https://github.com/arunponnusamy/object-detection-opencv)

The main object detection project:
https://github.com/legoman101/Py-Glasses-object-detection

The main HUD project:
https://github.com/legoman101/Py-Glasses-HUD

Acknowledgements :
Phil Stenning (some of the code for the server side stuff)
Arun Ponnusamy (for some of the Cvlib code)

Services we used:
OpenWeatherMaps - (https://openweathermap.org/api)

OpenCV - (https://opencv.org/)
Cvlib - (https://www.cvlib.net/)
Tensorflow - (https://www.tensorflow.org/)