InclusivityXR – an online tool for detecting inclusivity issues in AR and VR

Andy Woods[1]
andy.woods@rhul.ac.uk
Umar Farooq[2]
f.umar@surrey.ac.uk
James Bennett[1]
james.bennett@rhul.ac.uk
Marco Volino[2]
m.volino@surrey.ac.uk

[1] StoryFutures,
Royal Holloway, University of London
[2] Centre of Vision, Speech and Signal Processing (CVSSP),
University of Surrey

1 Introduction

There is a huge push to bring virtual and augmented reality into the mainstream by companies such as Meta ($36 billion, 2019-2022 [4]) and Apple (14k AR apps in the App store, 2022), with evidence suggesting that AR is at the point of mass adoption (e.g. 63% of people in the UK regularly use AR [1]). At the same time, one in five people in the UK have disabilities [5] and it is estimated the 'Purple pound' is valued in the UK at over £212bn per year [3]. Appropriately, there is a drive to make immersive content accessible to all.

Whilst there are tools in development for detecting accessibility issues for web-based immersive experiences that follow web-based WCAG guidelines (e.g. `https://aria-at.w3.org/`), we are not aware of tools specifically for immersive experiences developed for outside the web-browser (although Unity[1] and Unreal[2] do provide some general advice and tools). One reason for this is that there are a great variety of ways to define elements in immersive experiences, and it is hard to create tools flexible enough to deal with this variation – for example, a button can be created in a plethora of ways, from being a 'stock' button provided by a platform, to being crafted via a GUI, to being generated in code; or indeed being a hybrid of several of these techniques.

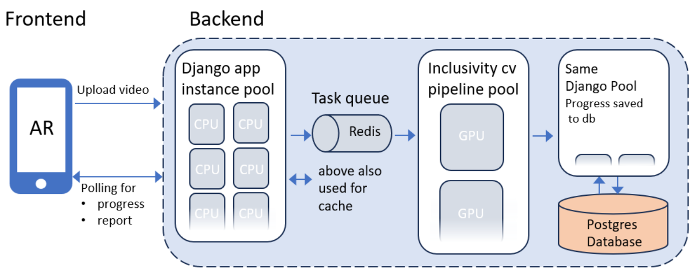

Figure 1: Server architecture. Note that multiple instances of the web-applications running at the same time is common policy, helping ensure reponsiveness during busy periods. During development, we hosted the above on single server, requiring about 10s to process / 1s realtime (8 x 2.1 GHz x86 intel Xeon Gold, 32GB RAM, no GPU)

2 InclusivityXR pipeline

InclusivityXR solves this problem by detecting issues of inclusivity at the level of pixels – whilst being agnostic to the code / schema that defines visual elements. It does this by means of a computer vision pipeline, through the following steps:

1. Developers upload a screen recording (video) of the app in action via a web application (Figure 1).

2. The video is processed (Figure 2) and an online report is generated, listing issues that have been detected.

Our approach to detecting inclusivity issues is as follows:

1. Identify key UI elements. We achieve this by building on the work by Chen, Jieshan, et al. [2] whose tool scans single still images and identifies UI elements using a range of traditional and deep-learning computer vision techniques. We extend their solution to work with video, in so doing augmenting a variety of time-based information to detected UI elements.

2. Apply a set of bespoke detectors each focusing on a given issue of inclusivity. So far we have implemented the following detectors: text too small; colour contrast inappropriate for regularly sighted individuals as well as individuals with 3 most common forms of colour-blindness; UI element too small; font overcrowding.

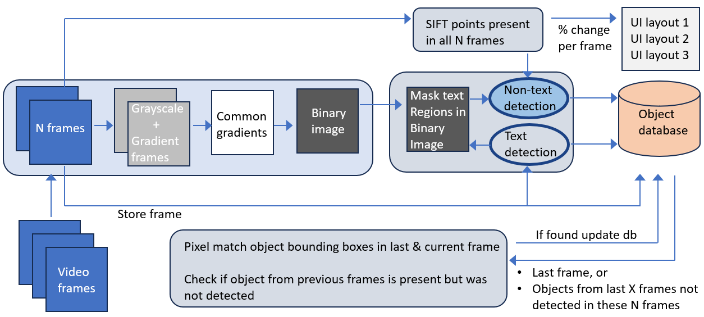

Figure 2: Computer vision pipeline

3 Future Work and Conclusion

We have in place the groundworks of a framework that can be readily extended with new bespoke detectors for other issues of inclusivity. A key next step is to identify which inclusivity issues the tool can have the most impact on for end-users and developers; to achieve this we plan to work with charities and individuals with different disabilities. A longer term goal is to enhance our pipeline via deep learning components, for which we will need more developers interested in AR/VR inclusivity to upload videos to the platform.

4 References

[1] J. Bennett, P. Dalton, O. Goriunova, C. Preece, L. Whittaker, I. Verhulst, and A. T. Woods. The story of immersive users. Technical report, StoryFutures, 2021. URL `www.storyfutures.com/resources/audience-insight-report`.

[2] J. Chen, M. Xie, Xing Z., C. Chen, X. Xu, Zhu L., and G. Li. Object detection for graphical user interface: Old fashioned or deep learning or a combination? In *28th ACM joint meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020.

[3] Department for Work and Pensions. High street could be boosted by £212 billion 'purple pound' by attracting disabled people and their families. Technical report, Gov.UK, 2014. URL `https://tinyurl.com/5n7khzrr`.

[4] ImmerseUK. Immersive economy report. Technical report, 2022. URL `www.immerseuk.org/wp-content/uploads/2022/10/Immersive-Economy-2022-final-13-Oct.pdf`.

[5] M. A. Vaughan. Family resources survey: Financial year 2019 to 2020. Technical report, Gov.UK, 2021. URL `https://tinyurl.com/ye2474s5`.

[1] `https://www.foundations.unity.com/fundamentals/accessibility`
[2] `https://docs.unrealengine.com/5.0/en-US/accessibility-in-unreal-engine/`