DATE: November 20, 2020

TO: Ananth Dodabalapur
Hanan Hashem
Venkata Suresh Rayudu

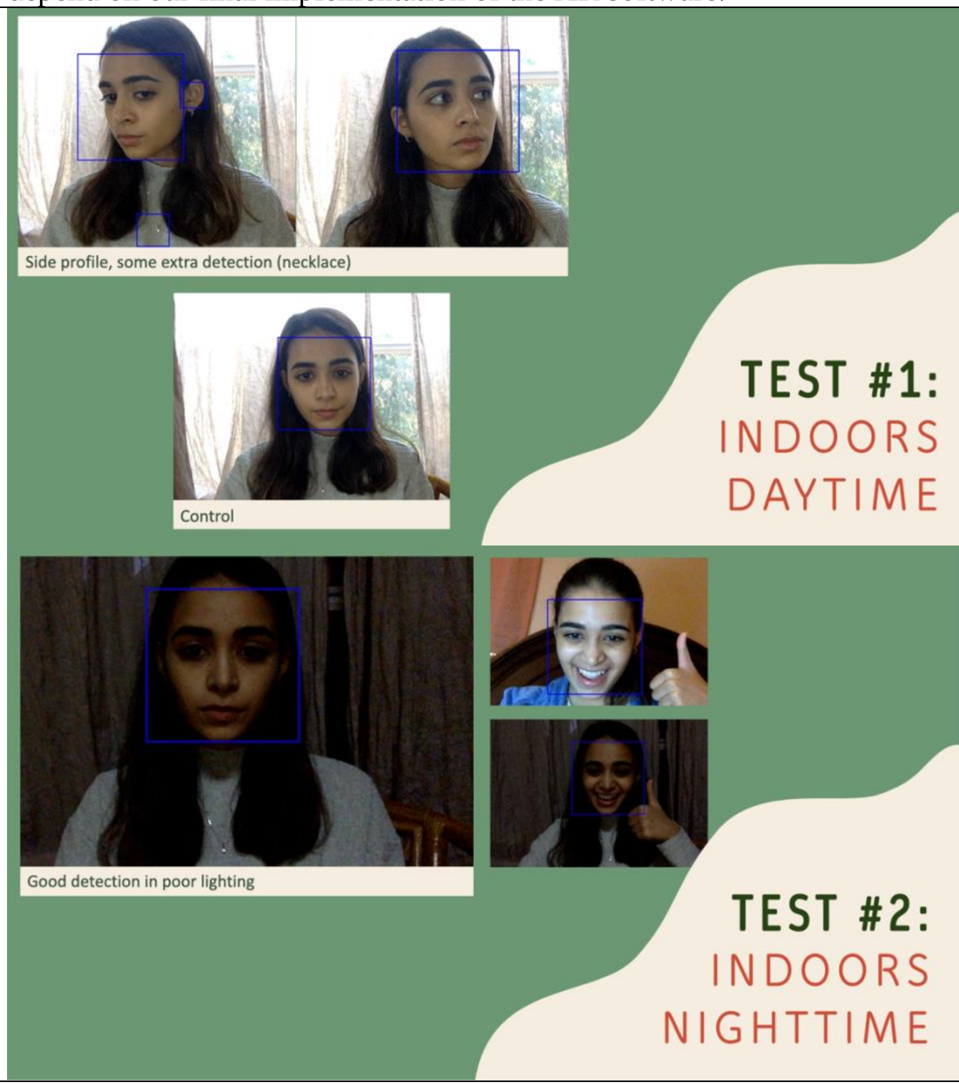
FROM: Evan Canter
Tanjeem Mazid
Andrea Nguyen
Deepanshi Sharma
Alexander Voth

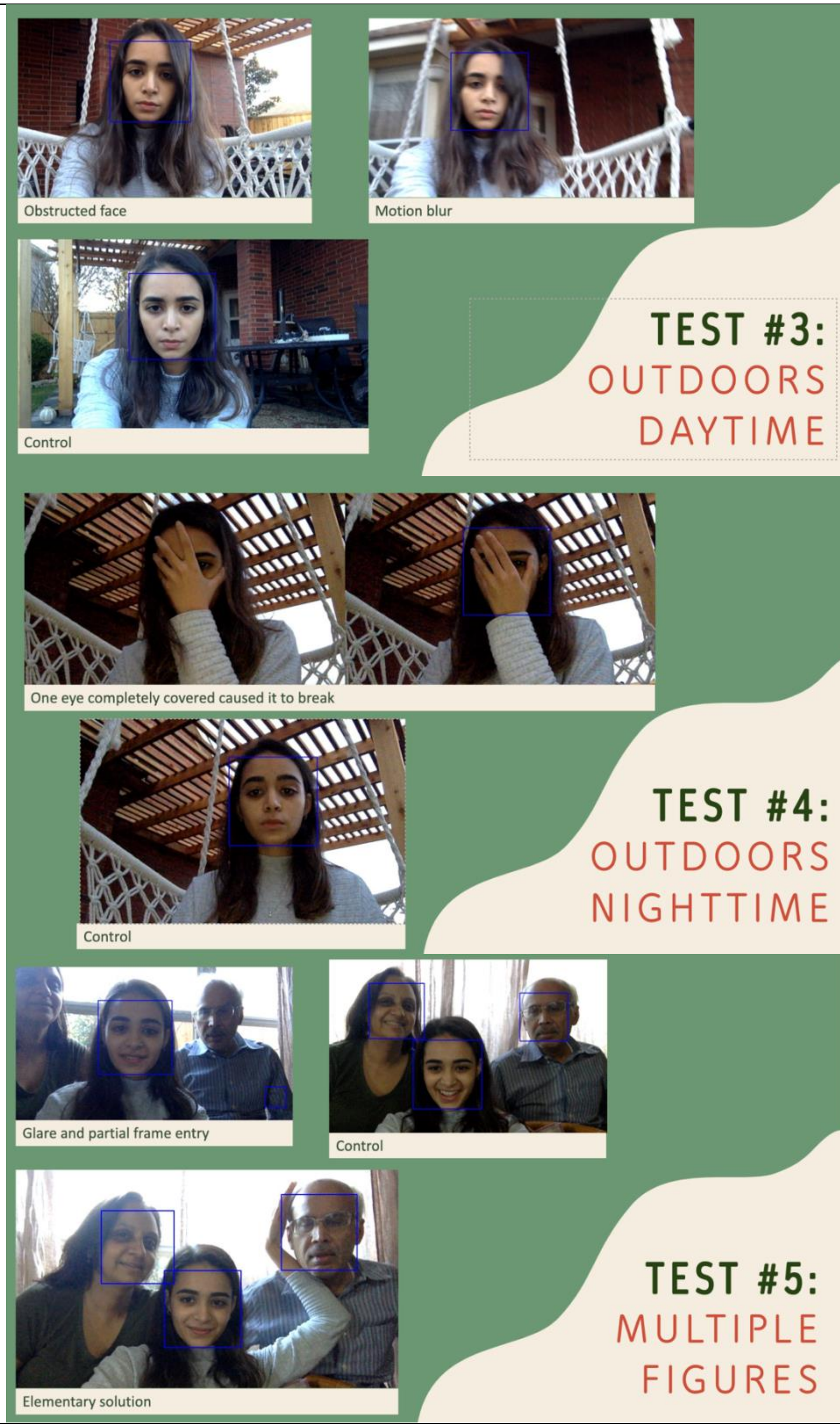
SUBJECT: Risk Reduction Experiments

TEAM RISK REDUCTION

Experiment #1: Markerless Tracking of Faces in a Frame	
Type	Software
Risk	After completing our prior art report, the team discovered several existing AR captioning technologies that implemented subtitles in a fixed position or required users to configure caption placement. However, users found the captions were obstructive or cumbersome to configure. This would only be a minor annoyance to a customer using the product in a specific recreational setting such as a movie theater. However, this potential obstruction of vision has much stronger consequences for our design because it is intended for use in a natural environment. In order to address this concern, our software will harness computer vision to track relevant figures in the frame and automatically reposition the text in an unobstructive manner. By identifying the coordinates of faces in the user's frame of vision, we can make sure to dynamically place all of our graphics in free space.
Experiment	In order to mitigate the risk of obstructing a user's view, it is imperative that our team is able to track the positions of important figures, such as faces. As a result, our experiment includes creating a sample module that will track faces in a frame without the use of markers [1, 3]. This sample module will be tested with single figure, multifigure, and differently light environments. Our intended use for the final product includes indoors and outdoors, so we tested a variety of lighting conditions in both environments.

Preliminary Results	Before conducting the experiment, we consulted Dr. Atlas Wang on viable facial tracking technologies. Dr. Wang advised us to select a platform that could interface well with future implementations of sentiment analysis and speaker recognition that could be useful in the later stages of our project. Dr. Wang also encouraged us to pursue marker-less tracking, which makes use of natural features instead of specific markers. However, we must keep in mind that current marker-less tracking technologies still require a trade-off between precision and efficiency [2]. We decided to build on an OpenCV algorithm for face detection using Haar Cascades [3]. The team created two implementations of the facial recognition module: one in Python and one in C++. We will make a decision on which implementation to utilize when we have finalized a screen since it will need to be compatible. The OpenCV algorithm accomplishes facial detection using Haar feature-based cascade classifiers [1, 3]. It is a machine learning based approach where a cascade function is trained from a large dataset of positive and negative images. This trained function is then used to detect faces in other images. In each of the tested environments (inside, outside, single figure, multiple figures, low light, and bright light), the module was able to detect the faces and generate a bounding box around the face. Captured images of the module's functionality in the various test cases is shown below. However, while the algorithm was very reliable in detecting faces in the frame, it did sometimes try to identify other objects in the frame as faces that were not human figures (i.e. a button on the test subject's clothing). This misidentification does not pose a great risk since it will only decrease the amount of free space rather than cause an obstructive placement. However, this issue can be resolved by training the algorithm against a larger and more diverse dataset of images. The results from this experiment allowed us to select a facial tracking implementation that performs with high accuracy. This will allow us to reposition our graphics in free space and reduce the risk of obstructing the user's vision.
Next Steps	While the module developed in this experiment is able to exhibit basic functionality, it can still be improved in three areas: efficiency, fidelity, and unit-reporting. Primarily, our choice of marker-less tracking requires a trade-off between precision and efficiency. We must address this by evaluating the performance of the algorithm with respect to processing power and computational efficiency [2]. This will also be affected by our choice of a Python vs C++ implementation of the algorithm. While C++ allows a greater low-level emphasis on optimizing performance, Python gives a more flexible implementation that can also interface with vast image processing libraries to accelerate training. The team will have to weigh the pros and cons of the two

	implementations to decide which method is better suited and feasible. Also, the module has a noticeable error margin when recognizing non-human objects as faces in the frame. While this extra detection does not pose a significant threat to the design goal, the next step to address this error would be to train the function on a larger and more exhaustive dataset to improve its functionality. Finally, the team will have to decide what units the algorithm should use to output the detected coordinates. Some options are: pixel matrix, rotation and translation matrix, Euler angles, or quaternion matrix. This will depend on our final implementation of the AR software.
Demo	


Obstructed face
Motion blur
Control
TEST #3:
OUTDOORS
DAYTIME
One eye completely covered caused it to break
Control
TEST #4:
OUTDOORS
NIGHTTIME
Glare and partial frame entry
Control
Elementary solution
TEST #5:
MULTIPLE
FIGURES

References	[1] "Cascade Classifier," OpenCV. [Online]. Available: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html. [Accessed: 24-Nov- 2020]. [2] Schmidt, Adam & Kasiński, Andrzej. (2007). The Performance of the Haar Cascade Classifiers Applied to the Face and Eyes Detection. 10.1007/978-3-540-75175-5_101. [3] "Face Detection using Haar Cascades," OpenCV. [Online]. Available: https://opencv-Python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/ py_face_detection/py_face_detection.html. [Accessed: 24-Nov-2020].

Experiment #2: Audio Transcription	
Type	Software
Risk	Our design problem includes bringing greater accessibility to hearing-impaired individuals. One risk our technology could pose is creating more difficulty by providing inaccurate data. Specifically, inaccurate captioning could lead to miscommunication, further hindering the user's sensory perception. Another goal of our project is to help users to clearly understand speaker emotions through corresponding graphics. This will be achieved through sentiment analysis which relies on natural language processing and text analysis. Therefore, if we are unable to transcribe speaker audio, sentiment analysis will not be possible. Additionally, if we were to incorrectly transcribe the speaker audio, we risk depicting the wrong emotion.
Experiment	The experiment to reduce the risk of miscommunication is to create the sample module that will take an audio stream and output the correct English transcription. This module will be tested in both quiet and noisy environments and evaluated on accuracy.
Preliminary Results	In conducting this experiment, we first researched the available transcription technology. Through our own research and listening to podcasts with experts such as Ramu Sunkara, CEO of Alan AI, we found that Google Cloud's Speech-to-Text API provides the most accurate transcriptions using powerful machine learning models. After determining that Google Cloud would be our best option, we implemented the API in python to take a continuous audio stream and output the corresponding text to the console. We tested the module in both a noisy and quiet environment. In the noisy environment, we played background noise throughout the transcription process. While in the quiet environment, no background noise was present. Please refer to the video links below for a demonstration. After we tested the

	module several times (in both environments), we consistently received accurate results. By successfully creating this module early on, the team found and tested the best captioning technology. We were also able to reduce the risk of miscommunication and prepare for audio processing and creating graphics that rely on accurate transcription.
Next Steps	Currently, the module waits for periods of silence to output text. The next step is to continuously output text as a person speaks, so that the captioning matches with lip movements. Additionally, the module often time out when running for long periods of time. To fix this issue, we need to close the connection during periods of silence and reestablish the connection when someone speaks. Another potential direction is to complete some pre-processing on the audio before transcription. For example, in multi-speaker scenarios we'd want to separate the audio into multiple channels and caption each speaker independently.
Video Demo	Testing in a Quiet Environment Testing in a Noisy Environment

Experiment #3: Micro display evaluation	
Type	Hardware to Software interfacing
Risk	Before buying all of the necessary components to realize our design, we have to be sure that the end result will meet certain requirements defined in our project definition. Our AR system needs to produce a visible image in a well-lit room. Our choice in display must support this requirement. Furthermore, the display must interface with our software system.
Experiment	To narrow down the vast market of micro displays, certain minimums and criteria must be set. Mainly, a goal for the display's brightness and a preferred interface protocol should be found; this requires extensive research of what is available online, and what lies within our project budget.
Preliminary Results	**Screen type** After searching multiple vendors, marketplaces and websites, OLED micro displays are simply too expensive or unavailable to consumers for use in this project. To meet both the resolution and brightness requirements, we would need top of the line displays that only a select few companies make. Furthermore, these displays would require extensive hardware to drive and interface with our software system.

LCOS reflectors can meet the brightness requirements with added LEDs, but also drastically increase the complexity of the optical system needed. Using LCOS, we would have to pivot our display design to one similar to google glass, requiring 2 polarizing beam splitters and LEDs.

LCDs, while not that bright, are both cheap and easily interfaceable. They will be the #1 preference for our AR system.

Brightness – Nits in relation to our design's efficiency

Nits (equivalently candelas per square meter) is a unit of luminance that describes the amount of light coming from a certain direction. This unit is useful for AR systems, and screens in general, as it can show the amount of light that will actually reach the eyeball. If we set a target for how much light should reach the eye, we can use our optic's predicted efficiency to find out how bright the screen should be. At the current stage of the project, it is likely the optics will be non-polarized and use a single 50-50 beam splitter. This means that the light from the screen will make two passes though the splitter, resulting in a light efficiency of ~25%.

Andrew Braun, in a tech blog post, summarized the brightness levels of multiple consumer products. Laptop screens can be usable indoors at a brightness level of around 200 Nits [1]. This metric sets a goal for our display choice: 200 Nits to the eye with 25% efficiency requires an 800 Nit screen minimum.

The seemingly poor efficiency of the optics is what limits our design's efficacy outdoors and in well-lit rooms. For use in presence of sunlight, phone screens can output 400-500 Nits [1]. Using the same efficiency of 25%, our display would have to output 1600-2000 Nits. Micro displays at this brightness are only made from high end companies (e.g., eMagin, Kopin), and do not sell individual parts to consumers.

Overall, our AR system could produce a visible image indoors with an 800 Nit LCD module. The image would resemble that of a laptop display at a medium brightness level.

Display Interfacing

The interface used by a given micro display affects the peripheral hardware necessary to drive the display. There are many protocols that display modules can support; most commonly, they use I2C, SPI, or MIPI DSI. The first two are serial interfaces would require a microcontroller to operate. However, our software platform operates on a laptop. Using a microcontroller would create a disconnect between the graphical data produced on the computer and the machine driving the display. MIPI DSI seems similar to SPI/I2C, as it is also a high-speed serial interface. The key difference is that DSI is widely adopted, and it is a protocol that is designed specifically for displays (DSI stands for display serial interface). I found adapters that

	can convert an HDMI signal into DSI, meaning the display data could be directly from the computer. Having the display mirror the laptops display would give the software team more freedom when generating graphics. Anything that can be displayed on a laptop screen can be mirrored to the AR display through this HDMI adapter. **Summary** With all of these factors in mind, our project would benefit greatly from a MIPI DSI compatible LCD with a brightness of around 800 nits. **References** [1] A. Braun, "What Is a Nit of Screen Brightness and How Many Do You Need?," *Make Tech Easier*, 03-Jan-2020. [Online]. Available: https://www.maketecheasier.com/what-is-nit-of- screen-brightness.
Component Links	**HDMI DSI bridge** display module ($100) || alibaba ($208 sample) || alibaba ($65) **LCD panels** Digikey, 500 Nits, MIPI, $11.52 || alibaba, 800 nits, MIPI, $7 alibaba, 600 nits, MIPI, $36, Circular

Experiment #4: Zemax OpticStudio Demo	
Type	Hardware (Digital Optical Model)
Risk	Before starting any tangible build of an optical system, we need to make sure our design functions in an optical modeling software. The angles of the light incident on our lenses must be correct, and the behavior of light passing through the lenses must be predictable and controllable.
Experiment	The main procedure in this risk reduction was building a feasible optical model that would demonstrate the behavior of ray tracing and light. At first, we attempted to see if we would be able to model light interacting with a beam splitter, as anticipated in our optical planning. However, due to our novel use of the software we decided to instead model this part with another surface. We were able to manipulate the surfaces in the 3D layout so that the ray field reflected off at the angles that we planned in our actual optical model. The model could be developed with a single ray field or multiple ray fields with each field approaching a different focal point and tilt. Additionally, we were able to add as many or as little rays as we wanted to the model – we chose to display the single

	ray field with three rays for our final presentation in an attempt to help viewers better see the travel path of incident light.
Preliminary Results	The main takeaway that we got from this specific experiment was a better understanding of the way light behaves in an optical model like the one we have planned. **Spot Diagrams** Zemax OpticStudio outputs a spot diagram that shows the resulting incident light coming out of our optical design. The diagrams show each field and wavelength combination individually, and they display how well our design has confined and controlled the spread of light passing through our system. With these results, we can see that a stable image should be able to enter the system and come out exactly as expected. **Lens Data** The Zemax software is mainly programmed with a spreadsheet of desired lens data. Each surface is able to be manipulated to suit our designs. Moreover, the spreadsheet outputs useful data such as focal points and lens thickness which are quintessential for optical engineering. **Three-Dimensional Optical Model** Our modeled optical system shows light passing through a paraxial lens. This paraxial lens allows incident light to pass through and become focused to a single image about 31.4mm away. This focused image is known as a focal point in optics. At this focal point, we placed an angled mirror. This mirror bounces the light at a direct 90 angle onto a specified object. This 3D model is not indicative of our initial design, but without the results of the micro display evaluation, we cannot generate the required parts in Zemax. Moreover, our limited knowledge of the Zemax software prevented us from designing the required surfaces from scratch via the Lens Data Editor. **Summary** The model is not an exact representation of our design due to lack of knowledge about the display, but we believe this is a good representation of the use of Zemax for preliminary modeling of our optical system. Demo video: https://drive.google.com/file/d/1KrhsdW1N1el6TSYcoBsc3YkhRsBNBBb2/view?usp=sharing

Next steps	Going forward, after we have concrete components picked out such as a display, we will be better equipped to model the optics for this project with more accuracy in regard to amount of light and wavelength.