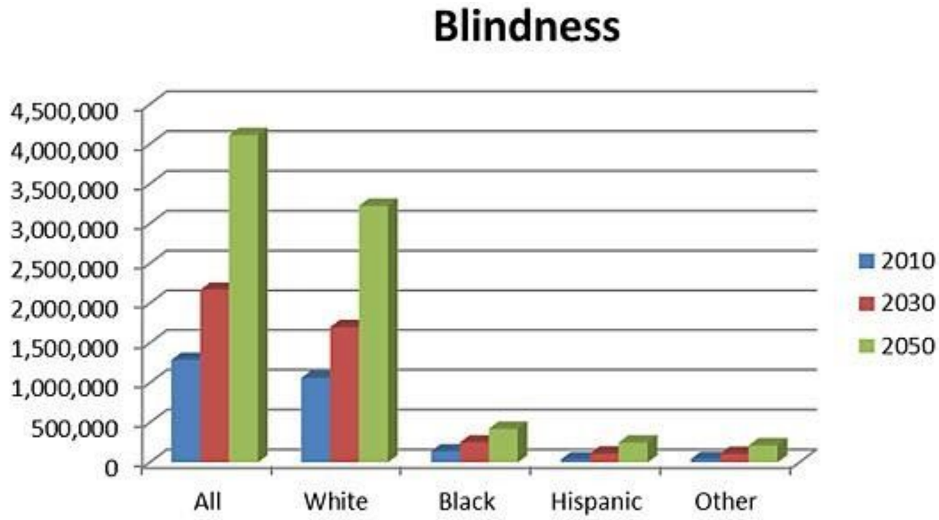
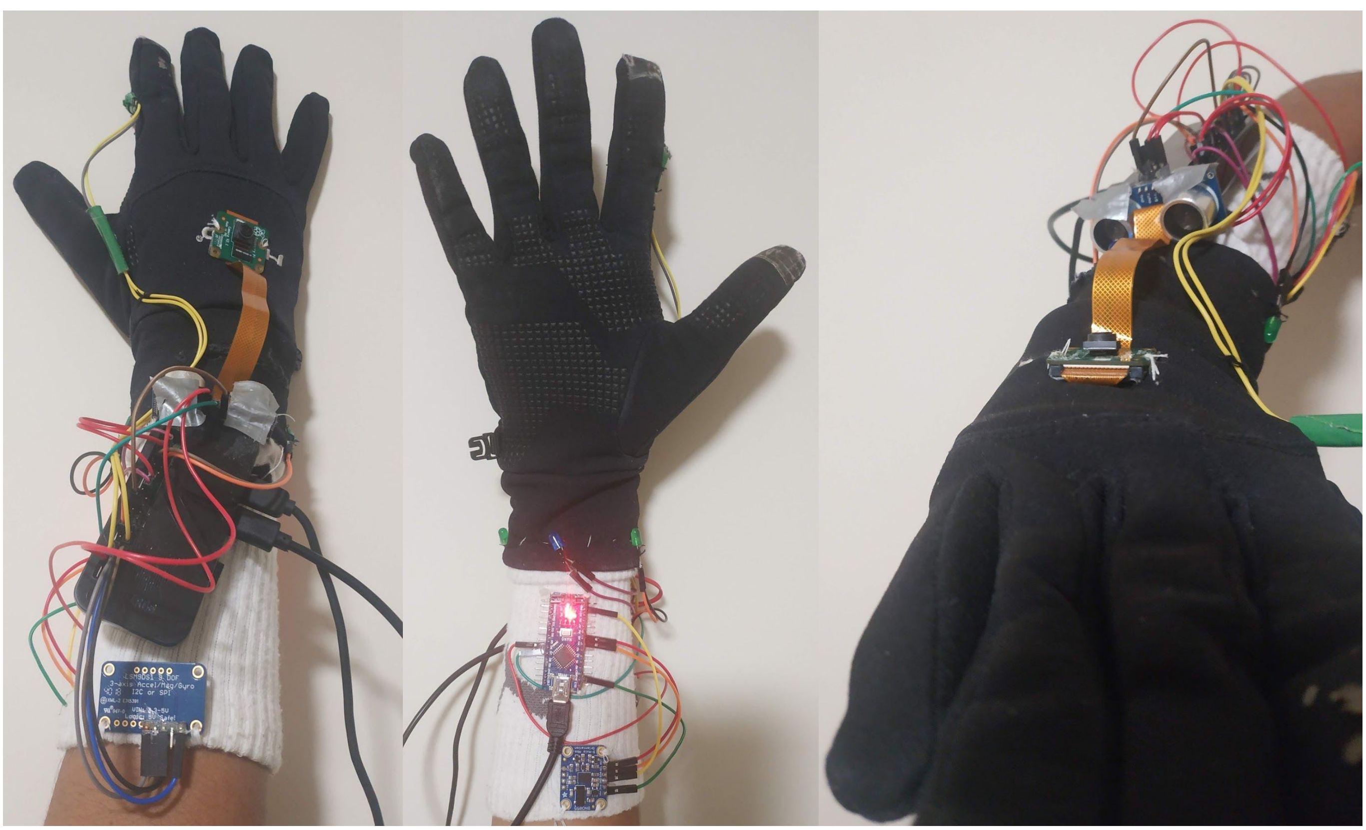
Project Name: Real-Time Object Search and Detection for the Visually Impaired

## INTRODUCTION




*From NIH 2019.*

**Figure 1.** This graph depicts the projection of the visually-impaired population in the United States alone for 2010, 2030, and 2050. The overall population is expected to triple by 2050.



A B C

**Figure 2.** The glove above is the  current prototype of the Wearevision. It is worn on the right hand in the images above. (A) View of the back of the hand where Raspberry Pi Zero W is housed along with the primary IMU and camera which is on the back of the hand. (B) Frontal view of the glove. The Arduino Nano and secondary IMU are placed under the hand. (C) Head-on view of the glove. The distance sensor is attached to the Raspberry Pi casing, facing in the direction of the hand.

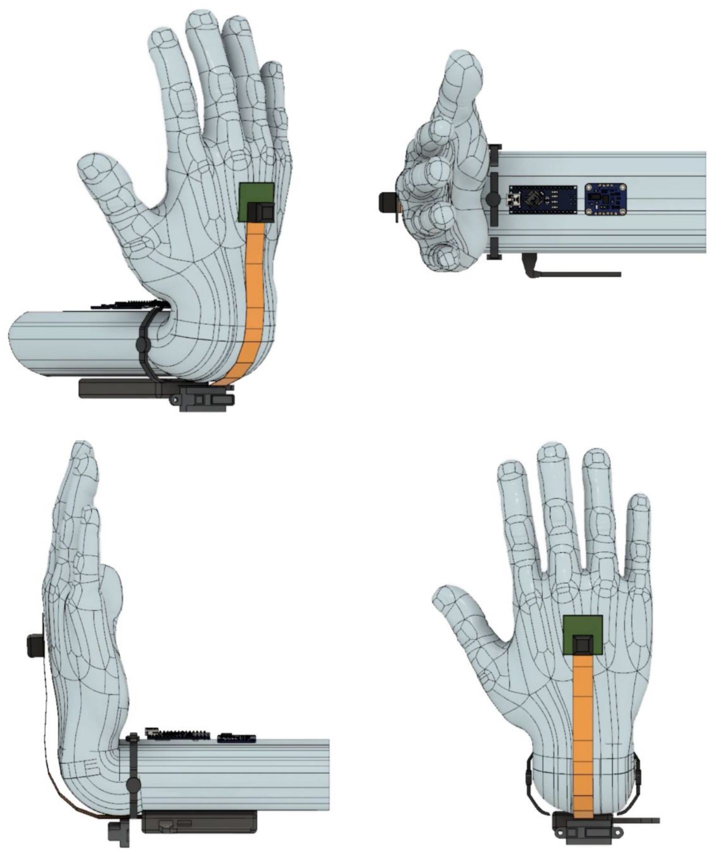
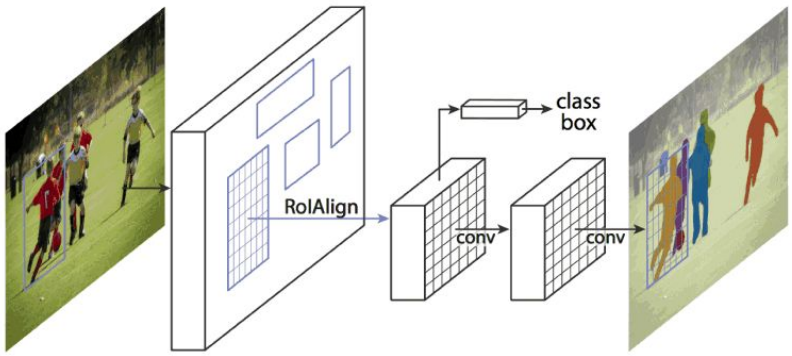


**Figure 3.** The models above are different views of the early CAD model of the Wearovision. Several changes were made from the CAD model to the current prototype as shown by Figure 2, such as casing for the raspberry pi and the addition of a button on the side of the index finger.

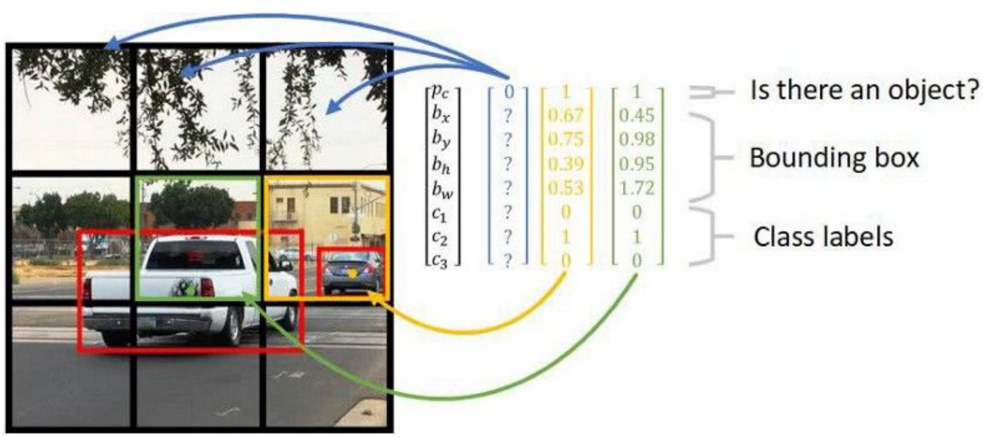
## METHODS




The Mask R-CNN framework for instance segmentation

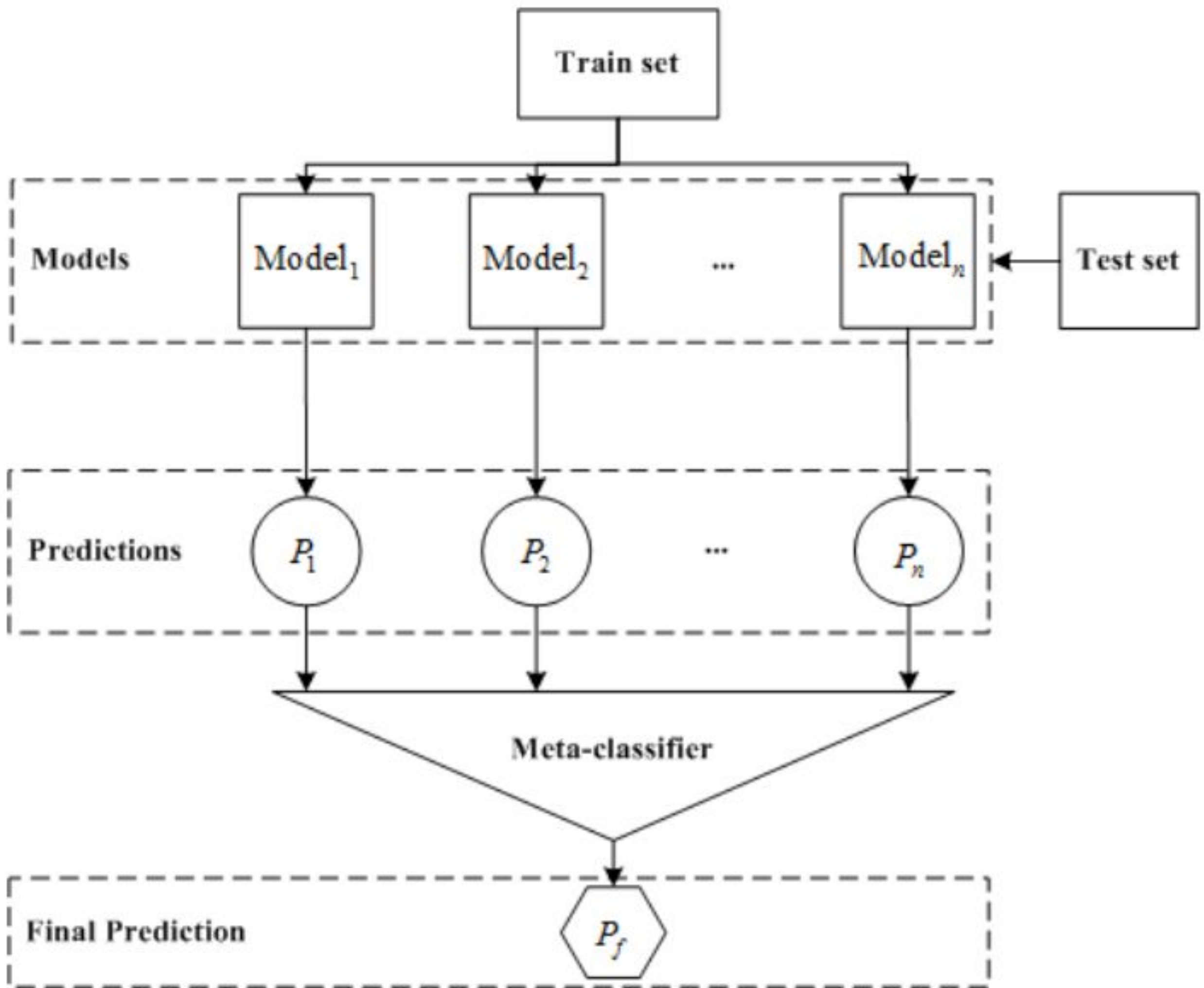
*From He et al. 2018.*

**Figure 4.** A diagram depicting how Mask RCNN processes a specific region - in this case a child. The image first passes through a CNN and feature map and divides the image into regional proposals. Then the region in question passes through another set of CNNS which classify the image, create refined boundary boxes, and generate a binary mask.



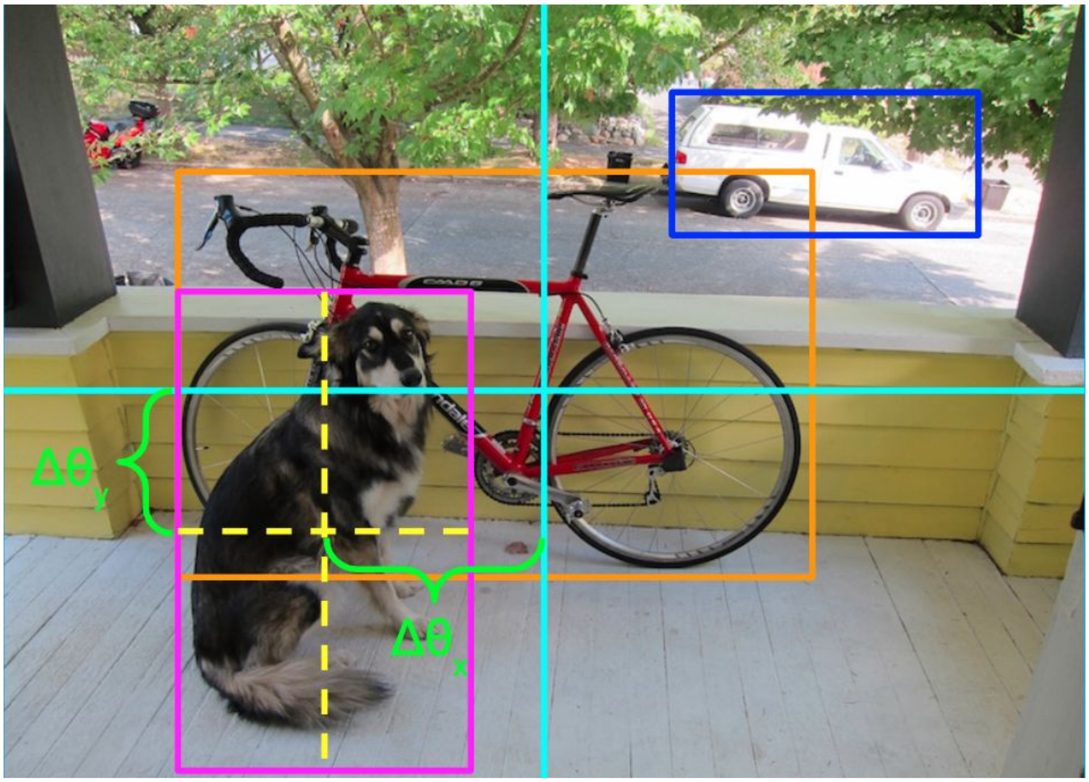

*From Redmon and Farhadi 2018.*

**Figure 5.** A diagram depicting YOLO’s approach to processing an image. YOLO first resizes the image then runs it through a CNN, outputting a feature map. It divides features into N X N cells and predicts 2 boundary boxes for each cell as well as the class probabilities for that cell.



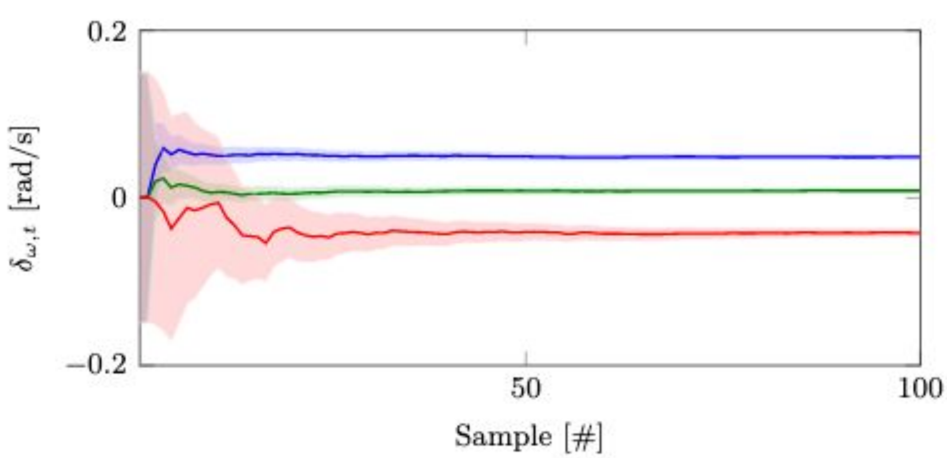

*From Jiang et al. 2019*

**Figure 6.** This diagram shows the process involved in stacking in ensemble learning. The different models each make individual predictions that get passed into the meta-classifier where we have a series of filters. This results in the final prediction.




*From Redmon and Farhadi 2018.*

**Figure 7.** Schematic of how the search algorithm operates. The blue lines indicate the center of the user's view field in the horizontal and vertical axis. If the user wishes to locate the dog, then the angular displacement from the center of the dog's boundary box and the center of the user's view field is calculated. These values update in real-time as the video live stream is taken at 5 frames per second.



*From Kok, Hol, and Schon 2018.*

**Figure 8.** An example of how the Kalman algorithm reduces and filters gyroscopic bias in the data. This allows for reliable long-term measurements.


*From Natarajan et al. 2018*

**Figure 9.** Examples of what Pythia is capable of. It uses vision and language processing to extract greater detail from the objects in the image.

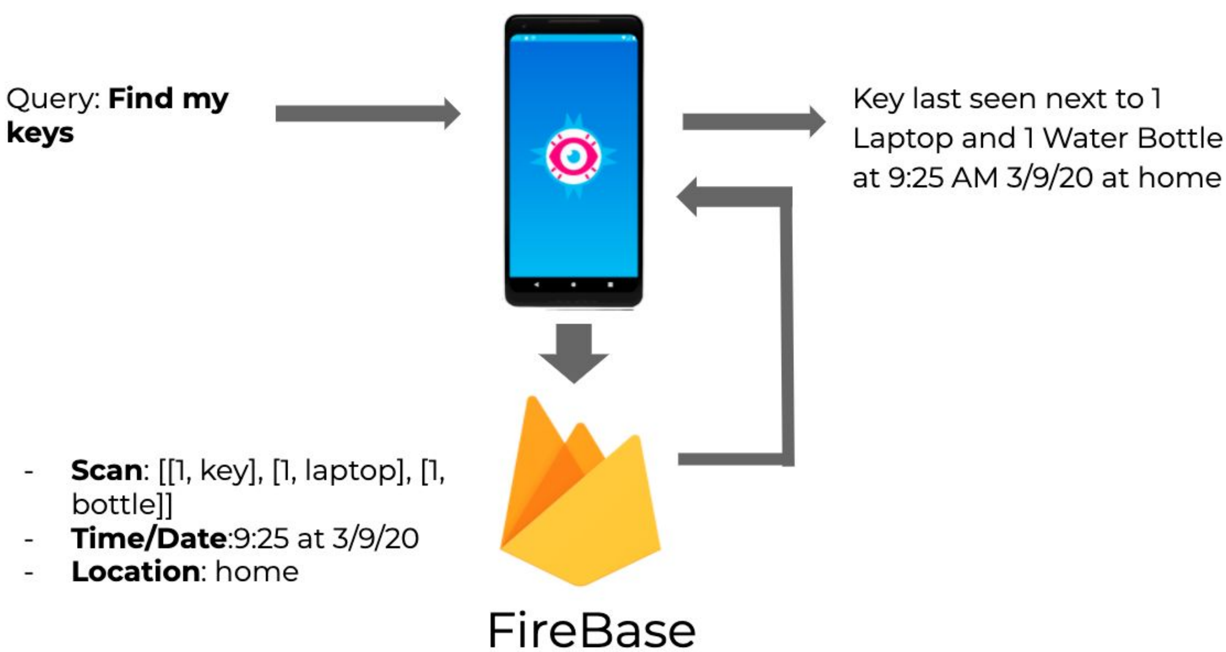



**Figure 10.** Process for the “Find” method. A query on the app searches the Firebase database and returns where and when the item in question was last detected.

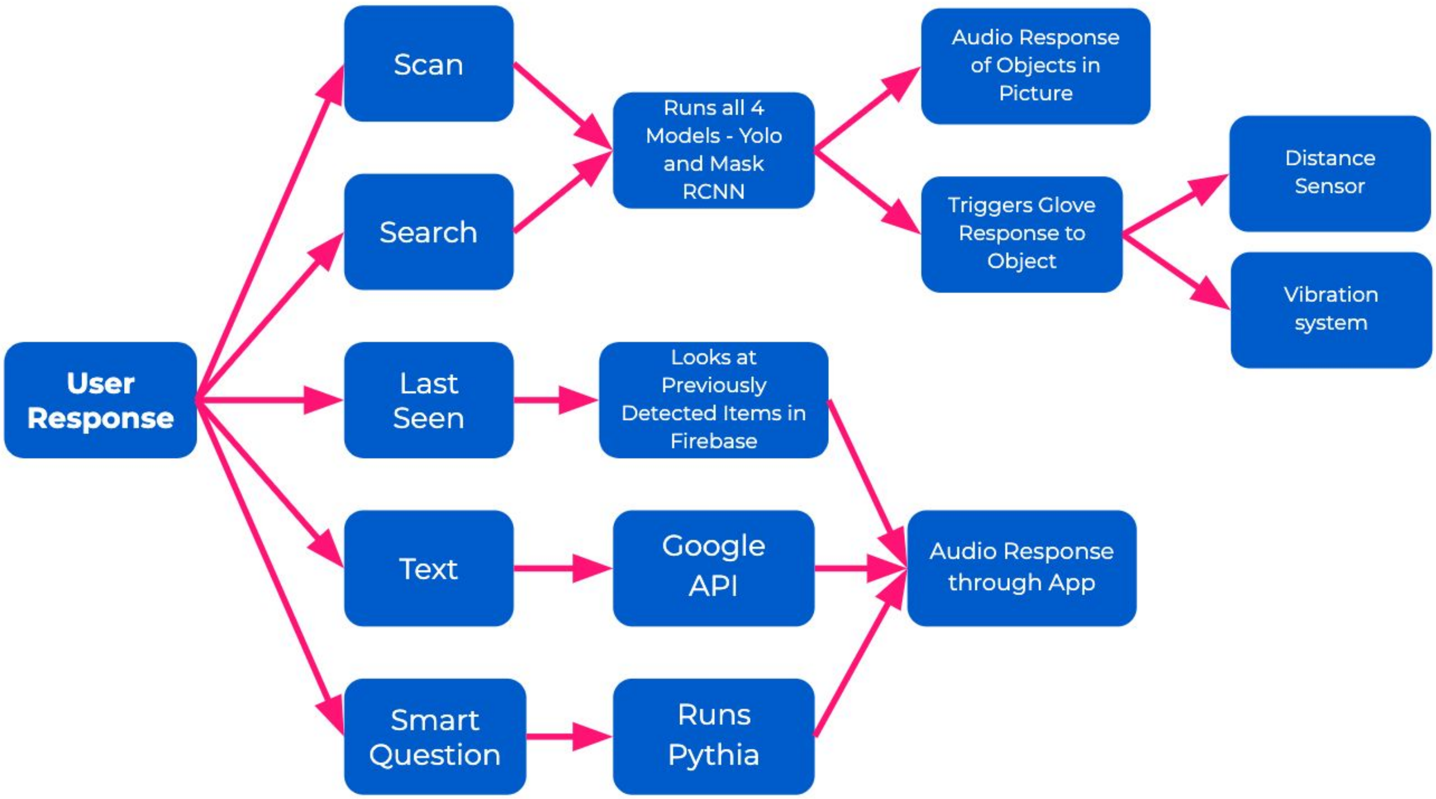



**Figure 11.** Complete flowchart for all the features after the user response is given. All processes involve an interaction with the app and Virtual Machine. These processes work to make the device more versatile in day-to-day use.

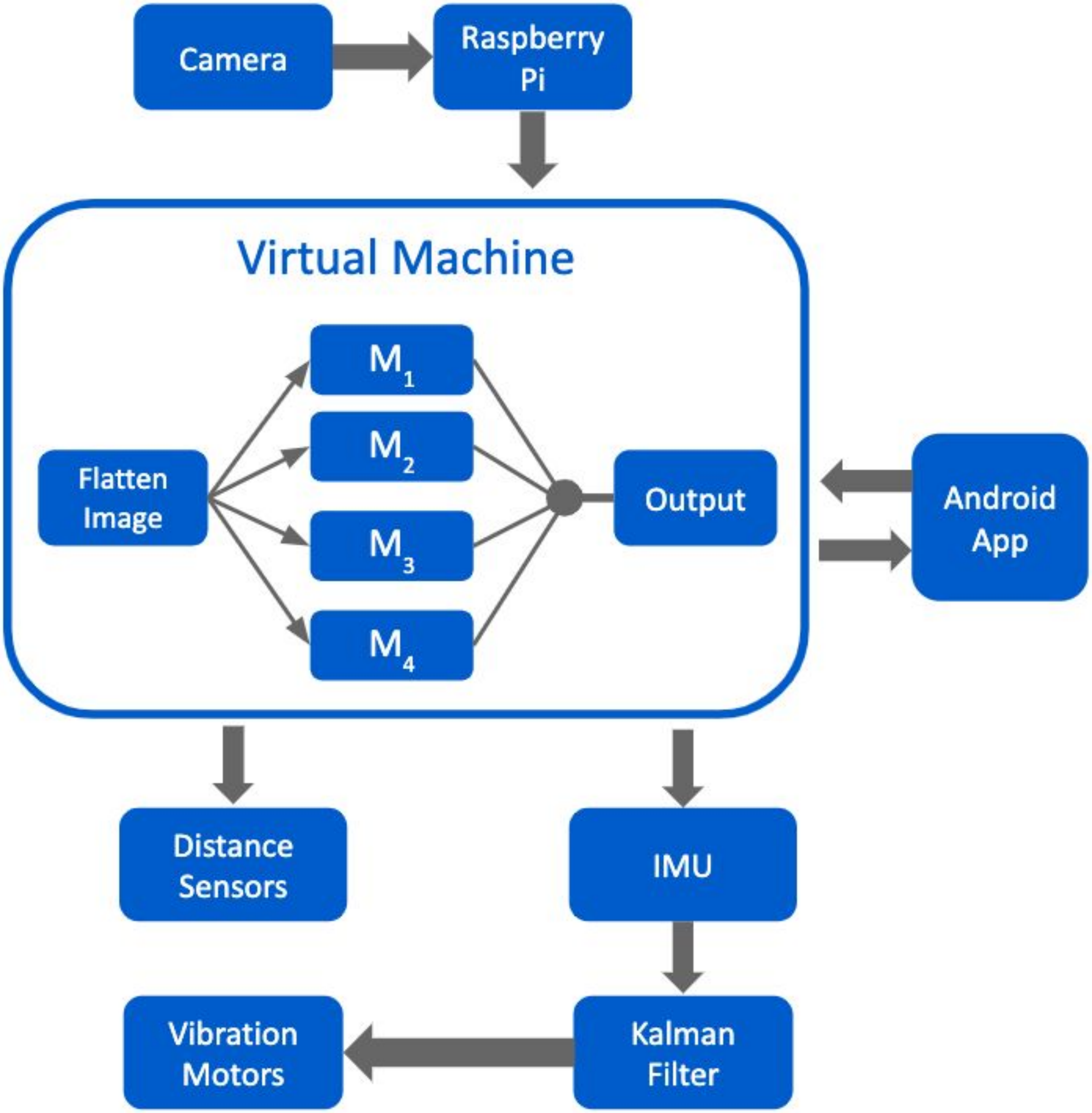



**Figure 12.** Flowchart depicting how the electronics, virtual machine, and app interact. The Raspberry Pi sends the image to the Virtual Machine or VM, where the VM then sends the classifications to the app. If the mode is “search”, the VM also activates the electronics on the Raspberry Pi.

**RESULTS**

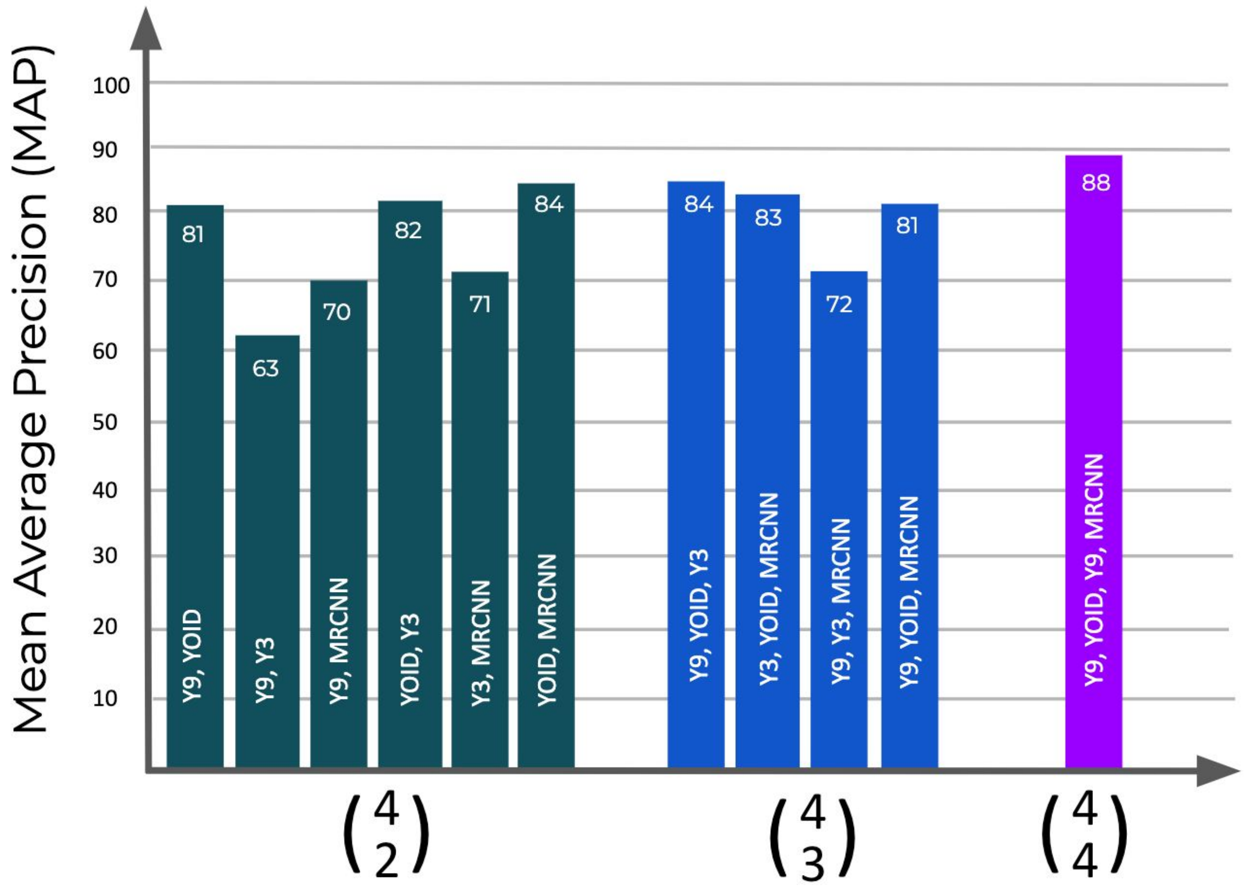



**Figure 13.** Results for the different combinations of models. The average accuracy for models in groups of 2 was 75% and 80% for the models in groups of 3. Running all 4 models yielded that highest accuracy, 88%, but is also the most computationally expensive. All of these combinations is much higher than the MAP of the highest model on its own, Mask RCNN, which is around 60%, showing how effective the meta-classifier is.

## WORKS CITED


1. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. Retrieved from https://pjreddie.com/media/files/papers/YOLOv3.pdf
2. He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2018, January). Mask R-CNN. Retrieved from Cornell University website: https://arxiv.org/pdf/1703.06870.pdf
3. Jiang, Y., Natarajan, V., Chen, X., Rohrbach, M., & Batra, D. (2018, July). Pythia v0.1: the Winning Entry to the VQA Challenge 2018. Retrieved from https://textvqa.org/
4. Manon Kok, Jeroen D. Hol and Thomas B. Schon (2017), ”Using Inertial Sensors for Position and Orientation Estimation”, Foundations and Trends in Signal Processing: Vol. 11: No. 1-2, pp 1-153.http://dx.doi.org/10.1561/2000000094
5. Blindness Data and Statistics. (2019, July 17). Retrieved April 3, 2020, from https://www.nei.nih.gov/learn-about-eye-health/resources-for-health-educators/eye-health-data-and-statistics/blindness-data-and-statistics
6. Jiang, Weili & Chen, Zhenhua & Xiang, Yan & Shao, Dangguo & Ma, Lei & Zhang, Junpeng. (2019). SSEM: a novel self-adaptive stacking ensemble model for classification. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2933262.