MAP MY HOUSE

Geet Jethwani, Nachiketh Doraiswamy, Priyansh Mishra

Institute of Systems Science, National University of Singapore, Singapore 119615

ABSTRACT

We propose an effective approach for real-time finger and hand detection system using Yolov3 as well as hand gesture recognition for designing a layout of a house. Our proposed system helps you design the layout of your house through hand gestures and helps you place images of various furniture items into your room in real time. Hence, it helps you design your home by setting up the layout of your room and helping you visualize furniture items that you would like to add to your room. Thus, making it an all-in-one solution to design or map your house.

1 INTRODUCTION

It is often challenging to design your home as soon as you move in to a new house. The art or process of designing your new home can often be a daunting task, as a result of which, we hire an expensive interior designer to do the work for us. The designers charge a lot of money for their work. They can visualize the design and translate it accordingly but sometimes fail to communicate their ideas to the customers. Thus, it becomes essential to understand the needs of the customer. It can include getting input for a specific design, a particular layout, the color of the furniture, and various other preferences that can be communicated by the customer. It also includes allowing the customer to select different styles of layout, a position as well visualization of furniture articles, and finally deciding a layout to pick based on the customer's preference.

Keeping in mind the problems stated above we have come up with a computer vision-based solution to help the customers design and visualize by exploring different styles and layouts that suit their house. We provide a variety of functionalities and bundle them into an application. The customer can use it to design his or her home accordingly.

Some of the functionalities are:

A computer vision solution to help in setting up the layout of a home that can set up virtually. Based on an inventory of preset 2D images of furniture provided, customers can select their preferred designs, hence making it easy for visualization.

Availability of various combinations and color variations the customer can try and select a layout based on the customer's choice. A customer may want to include a new furniture article in their house. Generally, we see 2D images of multiple furniture items on various websites, however, we have added functionality to view the images continuously. This helps in achieving a 3D look and gives a full view of the surrounding.

By placing the 2D images of furniture on the frame while capturing the room of the customer in real-time we let the user place those images wherever he wishes to within that room and check for himself if it is aesthetically pleasing to look at the space by visualizing the combinations he had selected.

2 LITERATURE REVIEW

Hand gesture recognition The major step in hand gesture recognition is the detection of hands and the subsequent segmentation of the image regions. It is important to segment because it differentiates the task related data from the image background, before passing them to the subsequent tracking and recognition stages [1].

To abstract and model the human body parts motion several hand gesture representations and models have been proposed and implemented by the researchers. The two significant classifications of hand gesture recognition are 3D model-based methods and appearance-based techniques [2]. Few of the 3d model-based techniques include 3D Textured Volumetric, 3D geometric model and 3D skeleton model. Appearance based hand gesture representation consist of color-based model, silhouette geometry model, deformable model and motion-based model [3].

Appearance based methods are mainly classified as 2D static model and motion-based method. In appearance-based methods there is a color-based model that identifies the fingers and the motion of the fingers through multi scale color features and segregate those fingers by attaching a different color tag to each [4]. Another technique is the silhouette method that uses geometric properties of the silhouette such as bounding box, perimeter, area, centroid and orientation to recognize the hand gesture [5].

Deformable Gabarit model are generally based on the deformable active contours and Motion based models are used for recognition of an object or its motion based on the motion of object in an image sequence [6]. Ultimately, the fingers are segmented from other parts of the background in which the white pixels are the members of the hand region, while the black pixels belong to the background [7].

3 DATASET

Our dataset included 12 videos of 8 frames per second that was run for 1 minute and truncated to get 4800 files belonging to 48 different classes which resulted into a split of 3840 files used for training and 960 files used for validation. We preprocessed the given files using Binary threshold Inverse masks in order to subtract the background. We trained our model on both an un-augmented and an augmented data set in order to find out the difference in accuracy in both as shown in the conclusion.

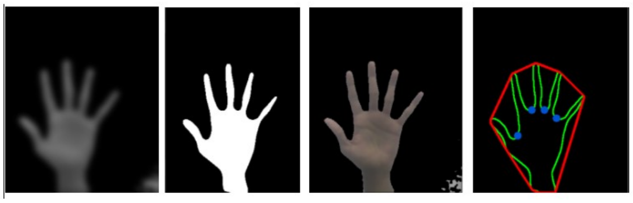


Figure 1: (a) Gaussian Mask (b) Binary threshold Inverted (c) Background Subtraction (d) Contouring

We also supplemented the dataset with our own hand images in order to increase its accuracy and to balance out the labels required. Our hand image dataset was of a uniform size of 180X180X3.

Inventory: We have also created an inventory dataset that includes images of the furniture articles used in an indoor environment (Bed, study table, lamp etc.). This dataset is taken from redwood.org, they have created ground-truth models of five complete indoor environments using a high-end laser scanner and captured RGB-D video sequences of those scenes. Apart from that we have created our own dataset which contains pre-defined scenes of a living room, which includes arrangement of sofa, table, TV, etc.

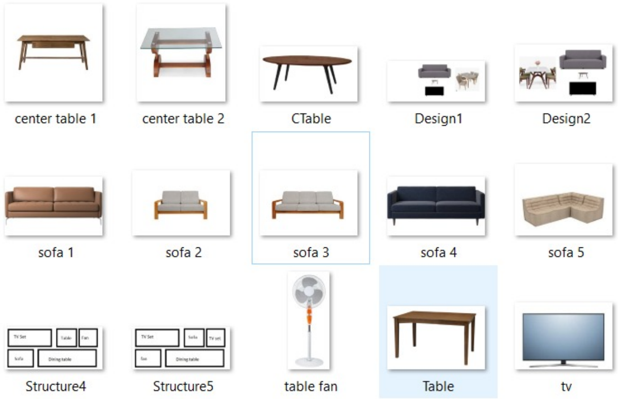


Figure 2: Inventory dataset

4 PROPOSED SYSTEM

Hand Gesture Recognition

Define the Region of Interest

Capture the video and then define the region of interest. Region of interest is where it performs the hand or finger detection and gesture recognition. So once a hand appears in the defined region of interest, only then will it be able to perform the gesture recognition. Thus, a well-defined green box is set up in the frame to identify the region of Interest.

BGR to HSV

Once the image is captured on the video stream, the image will be converted from BGR colors to HSV colors. Thus, we get a single channel value 'H' that holds the pixel color value and the other two channels 'S', 'V' holding the saturation and pixel level brightness.

HSV Range and Thresholding

After that we defined the skin color range in HSV. In mask, anything that is of skin color with be taken as 1 or white and anything that is not of skin color will be taken as black. Now the system has my hand in white and rest of the background in black.

Dilation

The dilation process will cause the white pixels to increase. Which will eventually make it easier to detect the hand properly. Thus, when the hand is placed in front of the web cam the intensified white pixels help in determining the strength or the probability that the hand is present in the ROI and what gesture is made.

Gaussian Blur

Then, the image is blurred using gaussian blur to decrease any amount of noise present in there. To eliminate the noise that remains in the image after the dilation process, we use gaussian blur to remove or get rid of the extra noise.

Contour Detection

After that contours are to be defined in the image. Contours are basically the outline of the object in the region of interest. Contours help us easily differentiate between the classes. In our case, we differentiate the black background with the intensified white pixels. Then I approximated contour for the maximum area (hand).

Convex Hull

Convex hull is a method used to detect contours or objects in an image. Once the convexhull is defined, we find the arearatio.

Arearatio = ((areahull- areacnt)/areacnt) * 100

The hand gestures are identified using the area ratio as each hand gesture has a different area ratio.

Defects

Then, we find the Defects in the convex hull. They are region in the hand not covered by the convex hull. All the angles between the fingers are usually between 30 and 60 degree. So, we know that the defects between the fingers would have an angle between 30 and 60 and any angle outside of that range should be ignored. The number of defects plus one will give us the number of fingers in the region of interest. Angle between defects is calculated using the cosine rule as shown:

angle = math.acos((b**2 + c**2 - a**2)/(2*b*c)) * 57

The overall procedure is as follows: 1. Capture the video and then define the region of interest. 2. Define skin color range in HSV. 3. Anything that is of skin color with be taken as 1 or white and anything that is not of skin color will be taken as black. 4. Blur the image using Gaussian blur. 5. Find the contours in the image. 6. Define a convex hull and calculate area ratio. 7. Find the Defects in the convex hull to identify fingers in the frame. 8. For each gesture propose an action: e.g., 1 will display the first layout of the interior design, etc.

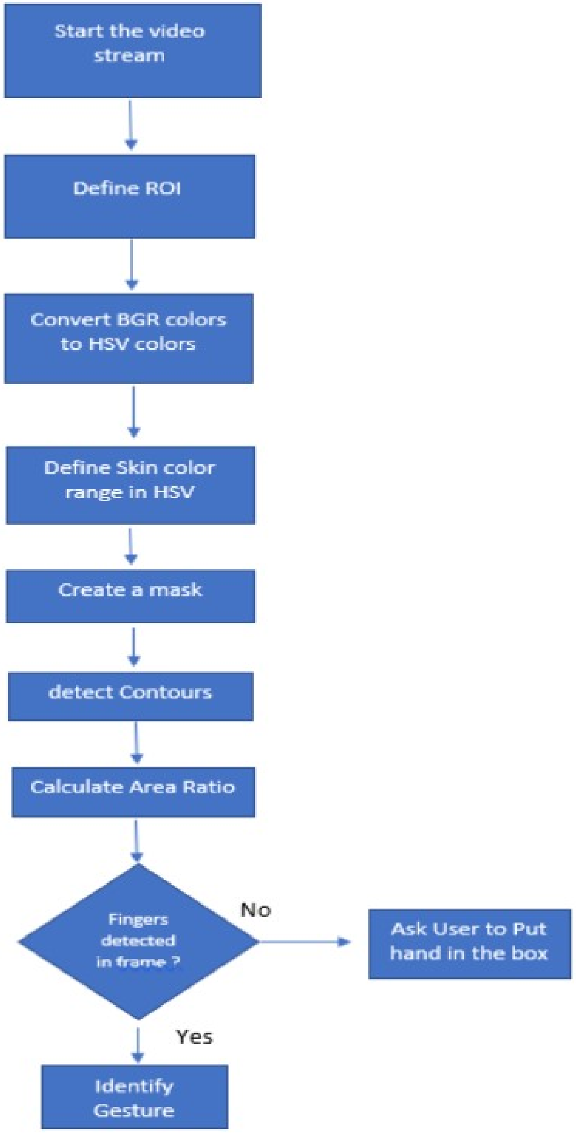

Start the video stream
Define ROI
Convert BGR colors to HSV colors
Define Skin color range in HSV
Create a mask
detect Contours
Calculate Area Ratio
Fingers detected in frame ?
No
Ask User to Put hand in the box
Yes
Identify Gesture

Figure 3: Flowchart for hand gesture recognition

Finger detection module For finger detection, we have used the Yolov3 finger detection dataset pre-trained weights. This dataset has a total of 500 images of index finger and thumb in different backgrounds. We can use the pre-trained weights to move the 2D images from one position to another. For finger detection, Yolov3 is the best fit model to get accurate detection.

The reason we used the Yolov3 model is due to its better accuracy as compared to SSD and RetinaNet. Yolov3 uses a variant of DarkNet. DarkNet has a 53-layer network that has been trained on ImageNet. As we are using Yolo for detection another 53-layer network has been stacked on the original layer to make it a 106 layer fully connected convolutional network for Yolov3. There is a trade-off in accuracy and speed of the model. Yolov3 is more accurate compared to the other models but has less speed during the training phase of the model.

We have used bounding boxes to highlight the detected fingers. As the model has been trained on only 500 images of different finger orientations it is more accurate on lighter backgrounds. It has been illustrated in the experimental results section how the images are being moved within the mainframe area from one position to another.

The images of furniture have been superimposed on the mainframe. As the index finger and the thumb come within a specified distance of the image, it triggers a movement action that causes the images within the frame to move from one-pixel location to another. By detecting the bounding boxes and calculating its Intersection over union with a specific furniture image we can determine the direction of movement of the image based on the movement of the detected fingers within the bounding boxes.

Hand Bounding Box Module The hand bounding box module is used to detect hands by the methodology of YOLO (You only look once). The YOLO network predicts bounding boxes and classes of objects at the same time based on activation maps output from the deep ConvNet.

When we first addressed the hand detection problems, the simple algorithm is to choose an approach like RCNN. A selective search is required to propose around 1000-2000 regions for an image and feed these regions into a binary classifier to see if an input region can be detected as a hand or not. This system also needs to deal with bounding box regression. In turn, it became clear that this system was extremely slow and time consuming. During the prediction time, the region proposal module itself took 8 seconds per image, including the time consumed by the binary classifier to predict 2000 regions for the image. So, after some deliberation of the type of environment as well as our input stream, we chose YOLO. YOLO simply looks at the final feature map output by ConvNets and predicts bounding boxes and classes scores simultaneously. The way YOLO works is as if it divides the input image to Sample × Sample cells and each cell is responsible for predicting its own bounding boxes and corresponding classes scores. As Figure 1 shows, the final output of the network is a Sample × Sample × N tensor, where N depends on number of classes and YOLO version. Each stage in the final output tensor looks at the cell in original image with the same spatial position and predicts B bounding boxes. For each box, a probability class is shown as well as its dimensions, such as height and width. The bounding box can encapsulate regions where a hand is detected, even if they are multiple inputs.

The images of the dataset are then overlaid on to the frame depending on the coordinates of the bounding box where the hands are detected. The images are non-uniform, so they go through a conditional preprocessing being scaled down if necessary. A timer counter is also present, which permanently overlays the furniture dataset image if the user chooses so by keeping the image at a particular area for 10 seconds.

5 EXPERIMENTAL RESULTS

Hand Gesture Recognition - Layout selection using hand gesture recognition

1. A region of Interest is defined as it can be seen in figure(4) below. A green bounding box is shown below is the region of interest. One must place their hand in that ROI for the system to identify the gesture.

2. Then the skin range is defined in HSV. Here it can be observed that since the skin range is well defined it can identify the skin color effectively (this can be observed in the mask)

3. As shown below anything that is of Skin Color is white and the rest is black

4. As the angle between the fingers is less than 60 degrees a small (circular) defect is seen between the fingers. This helps the system understand how many fingers are displayed in the image

5. Based on the number of defects, it can tell the number of fingers in the ROI. Suppose there are 4 defects, it would mean that five fingers are in the frame as:

The number of fingers = number of defects + 1

6. Finally, the different layouts are displayed based on the hand gesture.

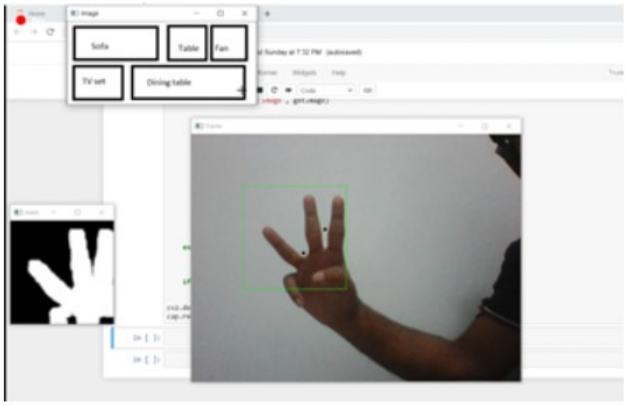


Figure 4: Layout Selection

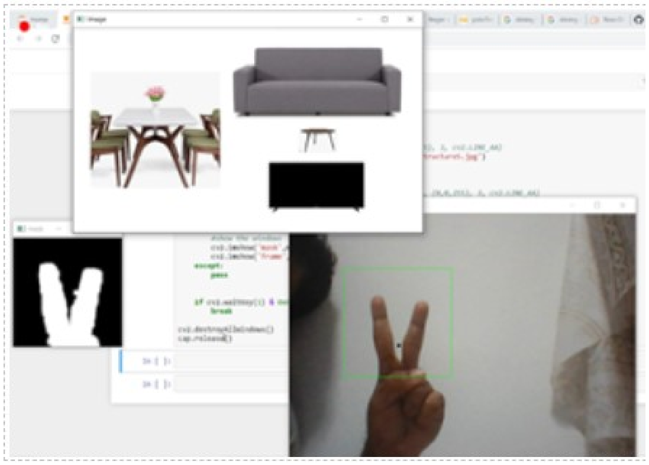


Figure 5: Furniture View

Moreover, there is also a component that allows the user to view a collection of 2d images as a 3d output. A set of images of indoor environment is placed in a directory and as soon as the corresponding gesture is displayed in the ROI, the images are retrieved from the dataset and loops through those images to provide a three-dimensional view of the indoor scene.

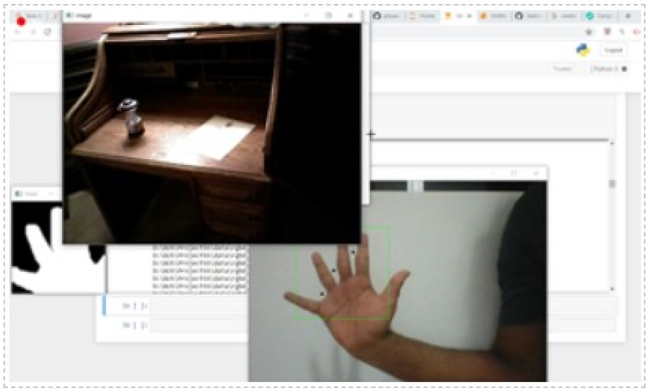


Figure 6: 3D view

Moving objects in your space (Finger detection)

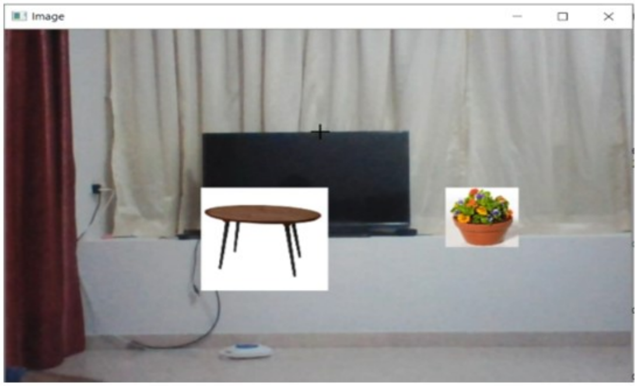


Figure 7: Images to displaced

This is the initial setup. There are a center table and a flower pot that has been taken from the inventory. The customer might want to see the layout with these two items in a specific position. As an example, the center table is required to be below the television facing towards the viewer, and the flower pot is required to be placed to the right of the television.

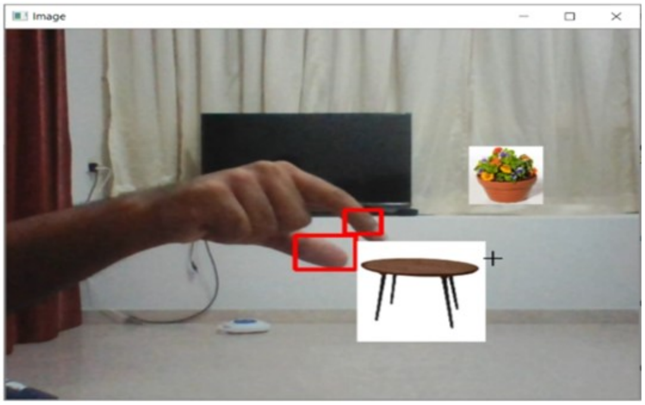


Figure 8: Detection

This is the finger detection step. The index finger and thumb are detected using the pre-trained weights of yolov3 model. In order to move the images to their specific positions the detected bounding boxes must be within a certain distance to the image. A function that calculates the distance between the bounding boxes as well as the image has been used, when the required conditions are satisfied the image can be moved around by virtually just using the finger movement.

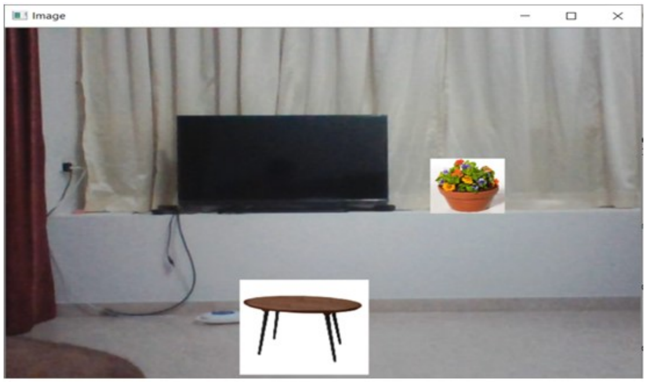


Figure 9: Displaced images

The final step shows that by just using the movement of the fingers within the frame the images can also be moved around to be in their required positions. This helps the customer visualize the layout and make it easier for him to set up his home as per his or her requirements.

Hand Recognition Results

For hand recognition using bounding boxes, we use YOLO. Yolo abbreviated as You only look once is a real time object detection method or technique where the object detection, in this scenario hands, are a regression problem to spatially separated bounding boxes and associated class probabilities.

In this approach, a neural network divides the image into certain regions, and predicts bounding boxes and probabilities for each region. The network then predicts the bounding boxes and class probabilities directly from the full images in one evaluation. Our model uses every frame detected by the webcam to recognize your hand above a certain threshold confidence by creating a bounding box around your hand. For aesthetic value, the bounding box which is usually drawn is erased, to provide an un-interfered point of view.

It then runs the image given in the path on a scaled down factor depending on the frame resolution of the chosen webcam. It takes the coordinates of the top of the bounding box which corresponds to your fingers and overlays the image on to the webcam to create the effect of augmented reality. Keeping your hand at a certain spot with some range leeway for 10 seconds pastes the image onto the webcam frame while you move on to the next item in your inventory.

Given the train vs test data that we used through our dataset, through the two training and validation accuracy graphs, we can easily identify that image augmentation is extremely effective as the number of datasets increase. This could be because of the augmented model recognizes hands even if they are at an orientation, which is not possible when the model which uses un-augmented data is used.

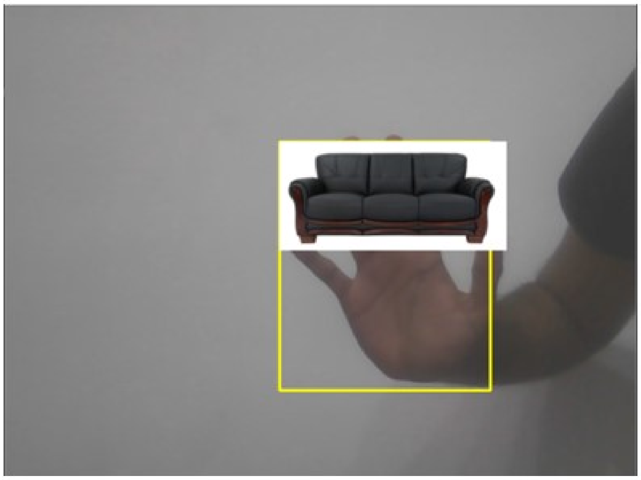


Figure 10: Image displacement with hand movement

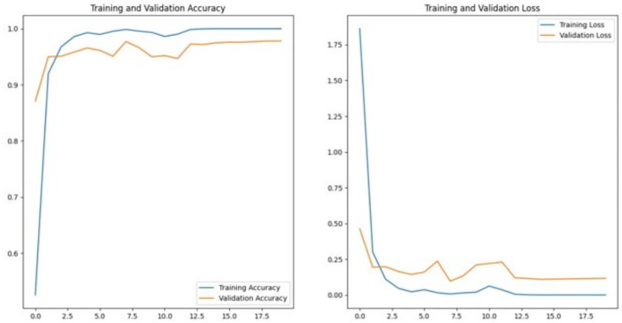


Figure 11: Accuracy and loss curve when data augmentation is not used

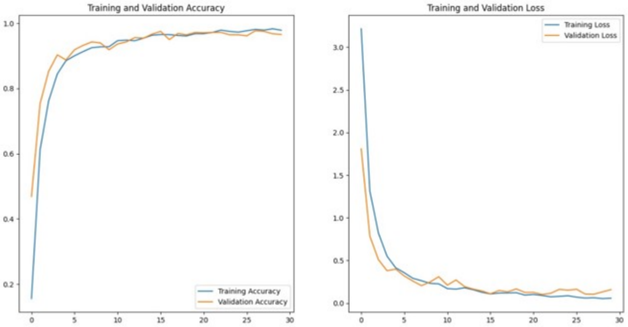
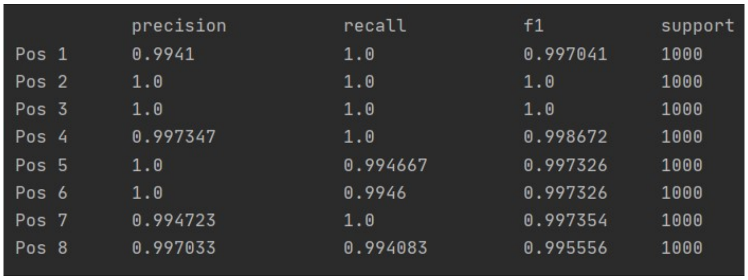


Figure 12: Accuracy and loss Curve when data augmentation is used

By the two graphs, it is clear to see that when data augmentation is used, the accuracy for the training and validation increase and the loss decrease exponentially when compared against the model that was not trained on the data. This is a product of the introduction of rotation into the dataset where augmentation has taken place, as our newer model is able to recognize oriented and flipped images of hands as well, which is the case when a webcam is used as it provides a flipped frame when captured. Augmentation also increases the overall accuracy by decreasing the percentage of the model relying on overfitting.



```
         precision      recall        f1          support
Pos 1    0.9941         1.0           0.997041    1000
Pos 2    1.0            1.0           1.0         1000
Pos 3    1.0            1.0           1.0         1000
Pos 4    0.997347       1.0           0.998672    1000
Pos 5    1.0            0.994667      0.997326    1000
Pos 6    1.0            0.9946        0.997326    1000
Pos 7    0.994723       1.0           0.997354    1000
Pos 8    0.997033       0.994083      0.995556    1000
```

Figure 13: Confusion matrix

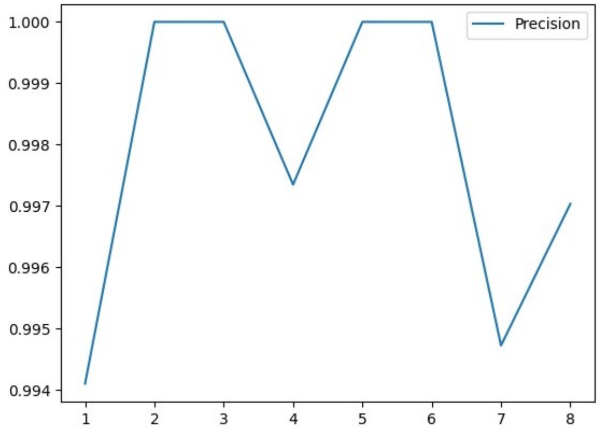


Figure 14: precision curve

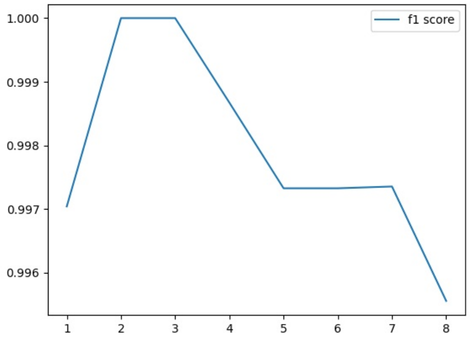


Figure 15: F1 score

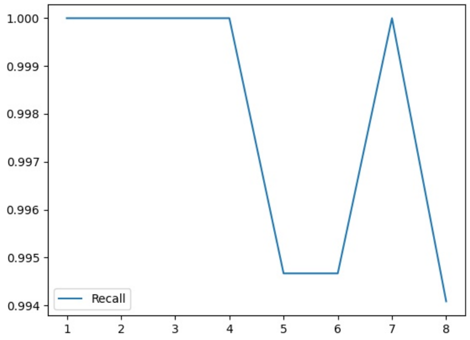


Figure 16: Recall

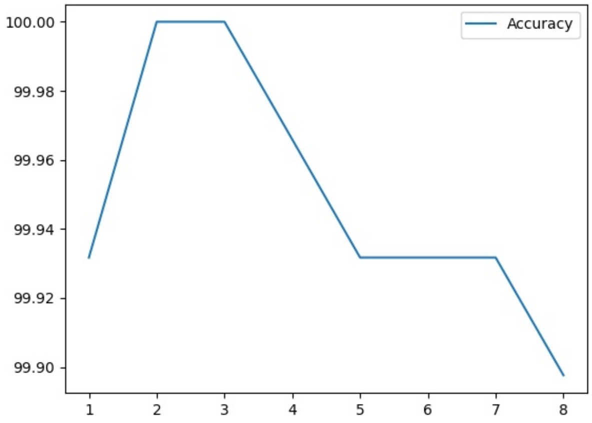


Figure 17: Accuracy

Image resizing feature

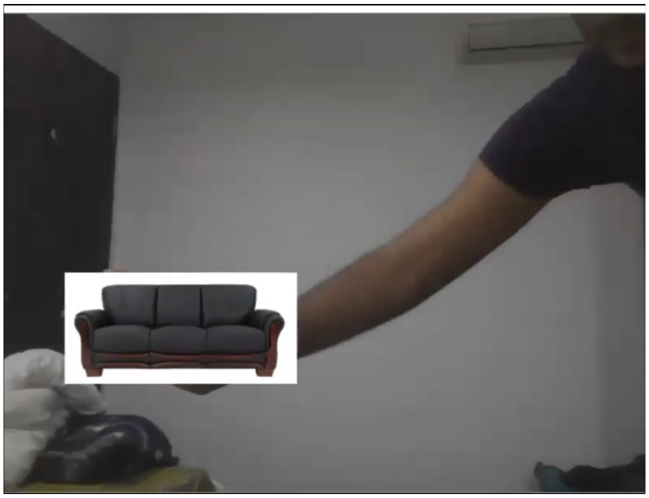


Figure 18: Regular image

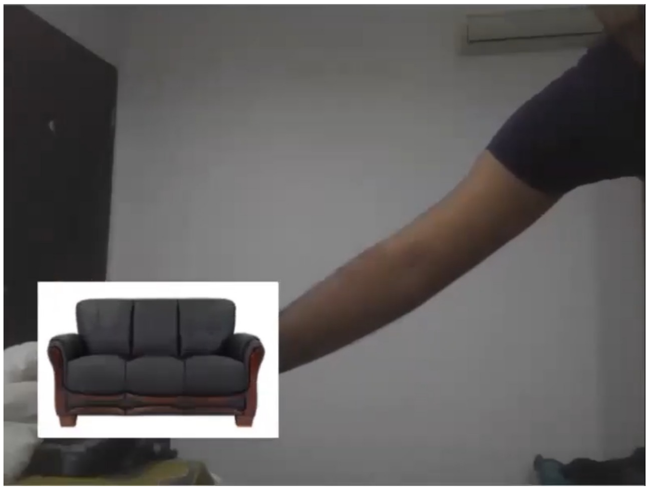


Figure 19: Resized image

Figure 18 shows the initial size of the image that was taken from the inventory. However , there are times when the image may be small and does not fit well in the room or is not big enough to visualize the item in the room. That's when we need to resize it such that the image size becomes suitable to that of one's space or room. Thus , in figure 19 it can be seen that the initial size of the image is resized to form an image that fits well in the room.

6 CONCLUSIONS AND FUTURE WORK

In this report we present a Yolov3 based hand and finger detection method as well hand gesture recognition for designing a layout of a house. We detect the fingers and hands using the bounding boxes and calculate the distance of the bounding boxes from the 2D images we would wand to displace. We also perform image augmentation on the training dataset to make the model robust to variations in hand sizes and environmental conditions. Experimental results indicate that our proposed approach achieves state of the art results as compared to other similar models.

For future work, we would like to enhance our system to include 3D CNN techniques to perform 3D image segmentation and provide virtual 3D layouts for the house as well as include various GAN techniques to generate new furniture image dataset which would be according to the preferences of the customers. Our future work also includes using YOLO or other techniques to identify objects and to be able to virtually remove or move them. This would require object detection as well as background addition to the space left behind by the object. A good example of this is an Augmented reality battlefield, where generals can use a holographic board in order to move troops as and how they see fit using only gestures or hand controls.

7 CONTRIBUTIONS

Geet Jethwani – Hand Gesture recognition. This module takes care of selecting the layout, furniture patterns and colors and visualizing the furniture items.

Nachiketh Doraiswamy - finger detection component that dealt with placing images in one's personal space and moving the furniture images as per the customer's choice.

Priyansh Mishra - Movement of images using hand detection. Placing images in one's personal space by hand movement and resizing those images as per the customer's choice and requirement.

8 REFERENCES

[1] Oudah, Munir, Ali Al-Naji, and Javaan Chahl. "Hand Gesture Recognition Based on Computer Vision: A Review of Techniques." Journal of Imaging 6.8 (2020): 73.

[2] Syahputra, Mohammad Fadly, Siti Fatimah, and Romi Fadillah Rahmat. "Interaction on Augmented Reality with Finger Detection and Hand Movement Recognition." International Conference on Augmented Reality, Virtual Reality and Computer Graphics. Springer, Cham, 2018.

[3] Rautaray, Siddharth S., and Anupam Agrawal. "Vision based hand gesture recognition for human computer interaction: a survey." Artificial intelligence review 43.1 (2015): 1-54.

[4] Binh, Nguyen Dang, Enokida Shuichi, and Toshiaki Ejima. "Real-time hand tracking and gesture recognition system." Proc. GVIP.

[5] Rautaray, Siddharth S., and Anupam Agrawal. "Real time hand gesture recognition system for dynamic applications." International Journal of UbiComp 3.1 (2012): 21.

[6] Hasan, Haitham, and Sameem Abdul-Kareem. "Retracted article: Human–computer interaction using vision-based hand gesture recognition systems: A survey." Neural Computing and Applications 25.2 (2014): 251-261.

[7] Chen, Zhi-hua, et al. "Real-time hand gesture recognition using finger segmentation." The Scientific World Journal 2014 (2014).

[8] Ryan J. Visée, Jirapat Likitlersuang and José Zariffa. “An Effective and Efficient Method for Detecting Hands in Egocentric Videos for Rehabilitation Applications”. IEEE sensors journal, vol. 15, pp. 1321-1330, 2015.