Learning Model: We implemented a 2D Convolutional Neural Network (2D-CNN) deep learning model, on which we trained our dataset (See Appendix A). Our architecture consists of 4

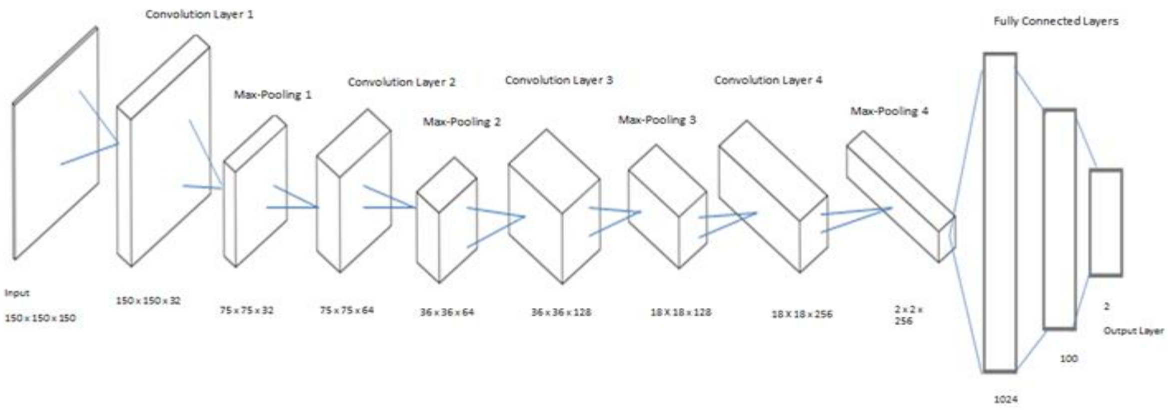

Fig.1. Convolutional Model

Convolution layers, each with a filter size of 3 x 3; coupled with 4 -2 x 2 Max-Pooling layers. The convolutional layers are then followed by fully connected layers, with an output layer consisting of 2 nodes corresponding to engaged and not engaged. We use the ReLu activation function at each layer, and compile the model using binary cross entropy loss as it is a binary classification model. For optimization, we use the Nadam optimizer and a learning rate of 0.0001.

We trained our dataset on this model and observed that the accuracy is about 81 percent. We also tried using pre-trained models for this purpose, but it did not give us good accuracy since our dataset was gathered by us, and is not compatible with the images in the datasets used in the pretrained models used for transfer learning. After training and compiling the model, we saved it in an hd5 format. Finally, we converted it to a tensorflow lite version to run it on the Raspberry Pi. A summary of the accuracies and F1 scores obtained and ROC curves can be found below.

TABLE I: LEARNING MODEL STATISTICS

Model	**Accuracy (%)**	**F1- Score**
Original Model	81.1	Engaged - 0.72 Unengaged - 0.72
Tensorflow Lite Model	70	Engaged - 0.72 Unengaged - 0.72
Transfer Learning		
VGG19	60.1	Engaged - 0.42 Unengaged - 0.33
ResNet50	56.2	Engaged - 0.22 Unengaged - 0.10
InceptionV3	61.5	Engaged - 0.13 Unengaged - 0.10

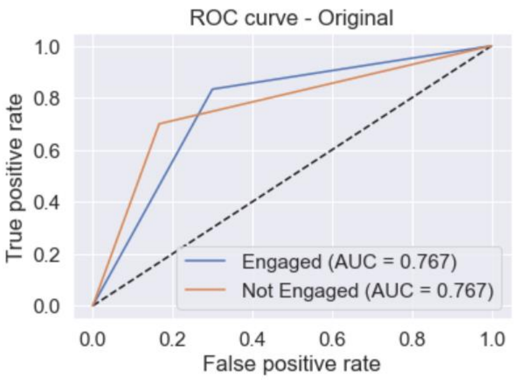

Fig. 2. ROC for Original Model

The original model gives an accuracy of 81.1% and an F1 score of 0.72 for both engaged and not engaged classes. From the confusion matrix (Fig. 7.), we can see that users who are really engaged are mostly classified as such, but the model does not do quite the job for unengaged users, i.e, false positives are high. This could be due to the variable nature of the dataset- the variances in pose, sofa setting, facial expressions, etc. If we examine the number of positive cases predicted correctly, we can say the model gives a precision of about 50% and a sensitivity of 60%, that is positive cases are predicted correctly about 60% of the time. The ROC curve (Fig. 3.) shows that the classifier does a fair job at predicting the classes as the curve lies on the top left of the graph. As a baseline, a random classifier is expected to give points lying along the diagonal (FPR = TPR). The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.

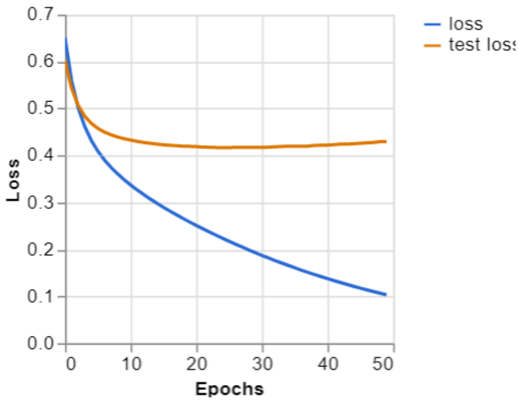

Fig. 3. Loss graph for original model

From the loss graph (Fig. 3.), one can see the test loss remains constant with each progressive epoch. Thus, we can say the level of overfitting is low; for the test loss does not increase along with the decrease in training accuracy with each successive epoch.

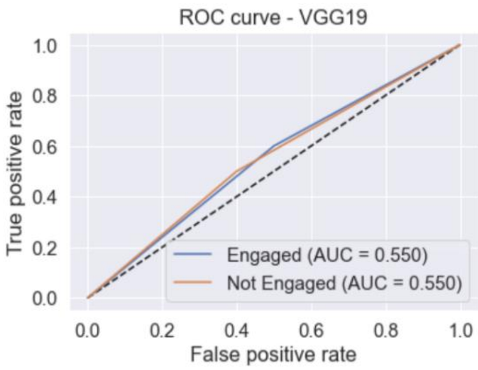

Fig. 4. ROC for VGG19

Comparing this to the other models obtained through transfer learning, we can see that we get progressively worse results. Transfer learning is the use of pre-trained models to solve newer and more complex but related problems. For this purpose, we used the ResNet50, VGG19, and InceptionV3 model.

Looking at the ROC curve for the VGG19 model (Fig. 4.), we can say it performs poorer than our original model because the curves lie closer to the baseline, resulting in higher false positives and negatives. The ResNet50 (Fig. 5.) acts as a random classifier, lying right along the FPR= TPR baseline. InceptionV3 (Fig. 6.) also gives the worst results, with the curve fully coinciding with the baseline.

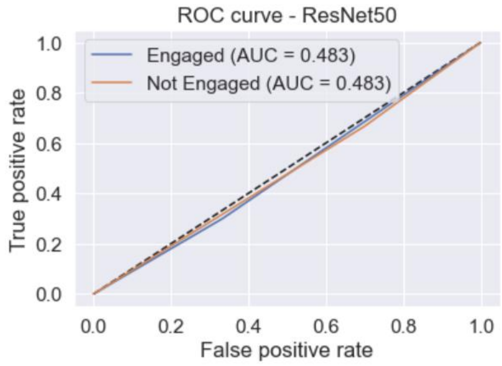

Fig. 5. ROC for ResNet50

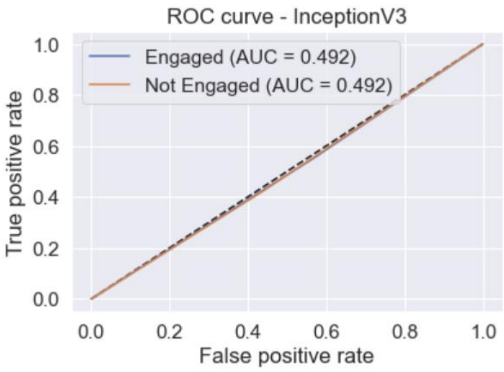

Fig. 6. ROC for InceptionV3

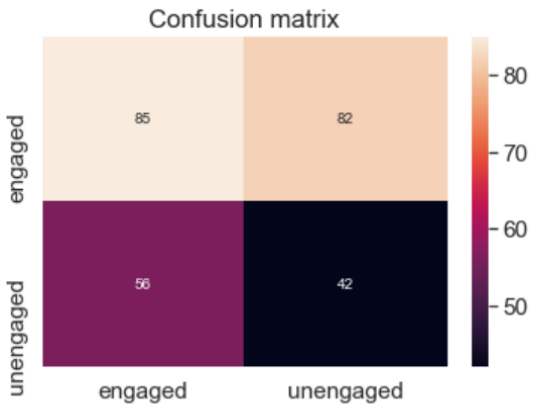

Fig. 7. Confusion Matrix for Original Model