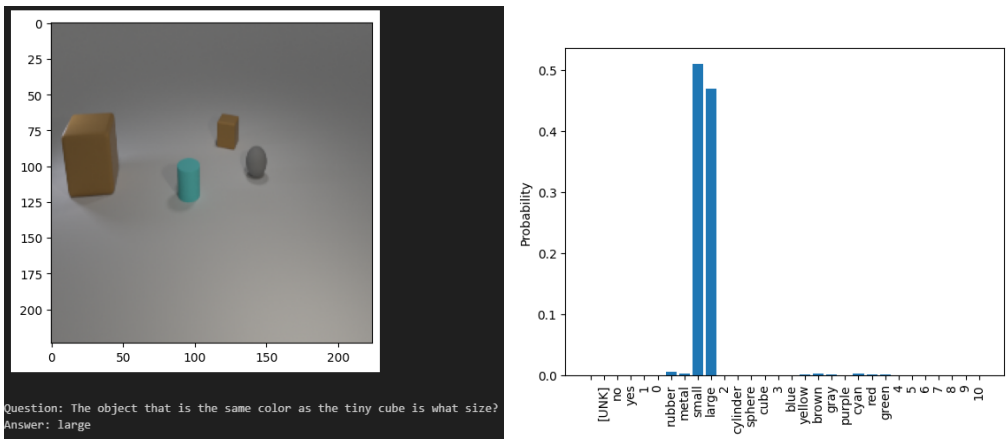
Report on the Performance of the Model - 02476922

After creating the model in the coursework, and training it, it resulted in an accuracy of around 45% on the validation set. This accuracy is not great, however, we were operating with a scaled-down version of the model specified in the paper, so should not expect incredible results. The training/validation curves in the coursework file show that in both accuracy and loss cases, the validation data set was a lot more volatile than the training set when it came to evaluation by the model. This is to be expected since the validation set is unseen data. Despite this, what is interesting is that, even when the model gets the response wrong, it usually has the correct sort of response. For example, if the correct response was a material, the model would predict a different material. If the correct response was "yes", the model might predict "no". Also, when the model does get the answer wrong, the correct answer usually only has a slightly lower predicted probability than the incorrect one. This shows that even with the restrictions imposed on the construction of the model for this coursework, leading to a reduced capacity, the model still learns the relative meanings within the answer vocabulary. This is shown below in figure 1.



(a) Image and Question

(b) Distribution of Responses

Figure 1: An Example of an Incorrect Prediction

Moving on to comparing with the model in the original paper, the most significant difference is that the model in this coursework used only one resblock, whereas the model in the paper used four resblocks. However, even when using only one resblock, the model in the paper attained a validation accuracy of 93.5% which is significantly greater than our 45%. This is due to some subtle differences in the architecture of the model created in this coursework, and slight differences in the training algorithm. First, the paper specifies that a GRU network is used with 4096 hidden units. This is significantly more than the 128 hidden units that are present in our model. Next, the feature extractor in the paper has 4 repeated sub-blocks whereas ours only has 2. Our model used 1 resblock whereas the paper chose to use 4. Finally, in the classifier block at the end of the model, the paper used 1024 hidden units in the first hidden layer, whereas the model we made only used 512. The model we used was scaled down due to the computational complexity of the full model, but this does show that with the more complex model and higher computational power, better results are achieved. Moreover, while most of the parameters used in the training algorithm were identical to those used in the paper, we only trained for 20 epochs, as opposed to 80 maximum with early stopping, a much longer training time. The paper also used a weight decay of 1e-5 which we did not. These differences in architecture and the training algorithm explain why the results obtained using the model in this coursework were worse than the paper.