COPYCAT: REAL-TIME HAND GESTURE RECOGNITION USING A CNN

Julia Arnardottir (julia93@stanford.edu) Solveig Einarsdottir (einarsd@stanford.edu)
CS230 - Stanford University

Problem Description

An increasing number of manufacturers rely on robots to stay competitive and solutions where humans and robots work in the same space are entering the market. We wanted to look at how humans could communicate with robots via hand gestures, which eliminates the need for a separate communication interface (most often a control panel). We used two approaches to classify hand-gestures in real time and were able to get an accuracy of 0.655

Data

- The original EgoGesture dataset has 83 classes of hand gestures, we use 10 classes with about 3,000 samples
- We use 5 frames sampled from videos to evaluate a gesture
- Each frame is resized to 64 x 64, at this size the human eye can still classify the gesture as seen in Figure 1
- 80% train / 10% dev / 10% test split

Fig. 1: One set of resized images used as a sample for the class 'Applaud'.

Models

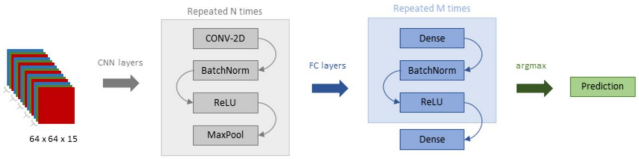

Fig. 2: CNN: Stacked arrays of images are passed through a 2D-convolution layers with 3x3 filters, followed by a batch normalization, a ReLU activation and a max pooling, this is repeated N=5 times and number of channels is doubled in each step. Next it is flattened before it enters 2 fully connected layers (M=1), and finally a softmax activation (argmax)

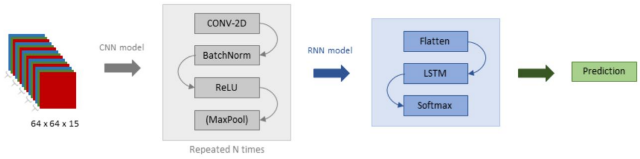

Fig. 3: CNN-RNN combination: The images are first passed through a 2D-CNN and the results from that are passed into a recurrent neural network (RNN). For the CNN model, the convolution layers are repeated $N = 8$ times and only some of those include max pooling. Adam optimizer and cross-entropy loss are used for training the model.

Results

Approach 1: CNN model

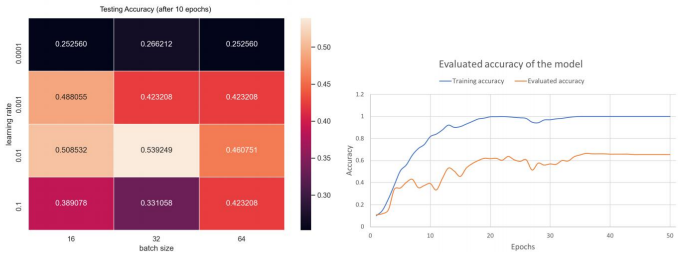

Fig. 4: Accuracy of model for various learning rates and batch sizes after 10 epochs (left) and training and evaluated error (right)

Approach 2: CNN-RNN model

Fig. 5: Perfomance of the CNN-RNN model when classifying 3 and 4 gestures

Remarks

For the 2D-CNN model we can see from figure 4 b) that the training error is close to 0% but our evaluated error is over 30% for 11 classes, and for the CNN-RNN model we can see that from figure 5 b) the training error is also about 0% whereas the evaluated error is over 70% for only six classes. We thus have a variance problem in both our models, and more so for the CNN-RNN model. This is most likely due to overfitting to our dataset since it is quite small. The error could possibly be reduced by using additional regularization techniques or more data with better resolution but that would require more computational power.