Hand Gesture Recognition

University of California San Diego

Arshia Zafari, Hayk Hovhannisyan, Erik Seetao, Silver De Guzman

Abstract

Given a collection of hand gesture training images, we aim to design a classification model that distinguishes different gestures from new images. This problem has widespread application in areas such as ASL translation and driver/pedestrian hand recognition for safe and smart driving. As of recently, deep learning has seen tremendous growth in different computer vision and machine learning applications, so we attempt the problem of hand gesture recognition by designing and training a three-layered Convolutional Neural Network. The network takes in a set of images from 10 different gestures and is able to classify them with around 90.2% accuracy.

Data

The Hand Gesture Recognition Database has 20,000 infrared images of hand gestures split into 10 different classes from 10 users, 5 male and 5 female; data was captured by Leap Motion sensor and converted to single-channel, normalized grayscale.

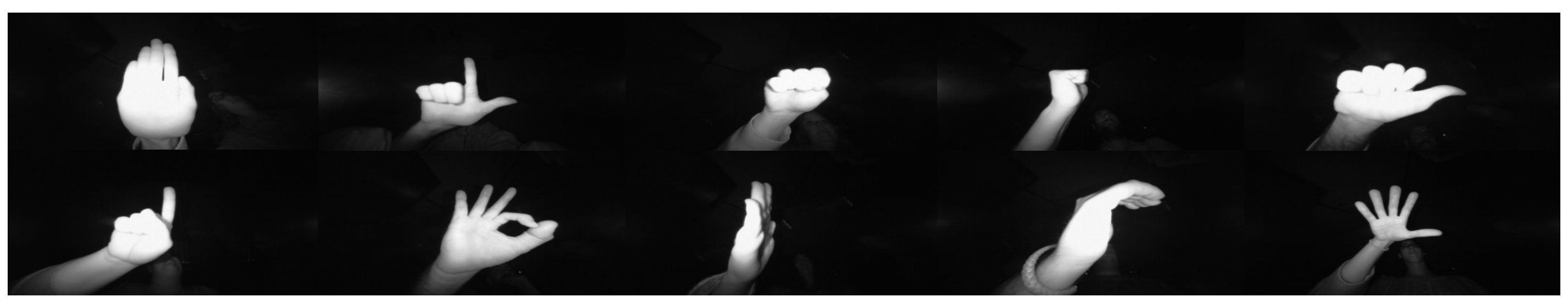

Figure 1. palm, L, fist, fist side, thumb (top row), index, OK, palm side, C, down (bottom row)

Each image is given as a 640x240 .png file and using findContours() by OpenCV to locate the hand center, each image was cropped to 120x120. Further data augmentation included creating a completely new set of the same images but mirrored to improve robustness of model and to ensure right/left hand representation, for a total of 40,000 images.

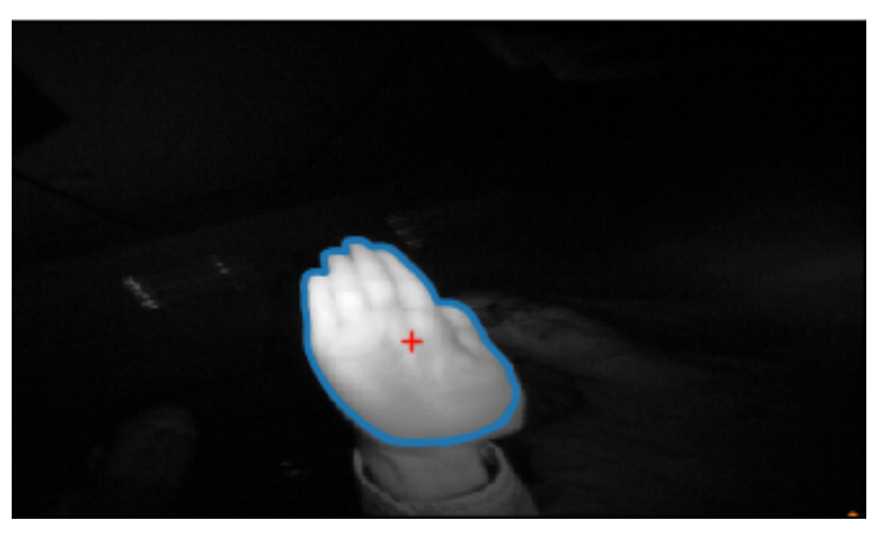

Figure 2. findContours()

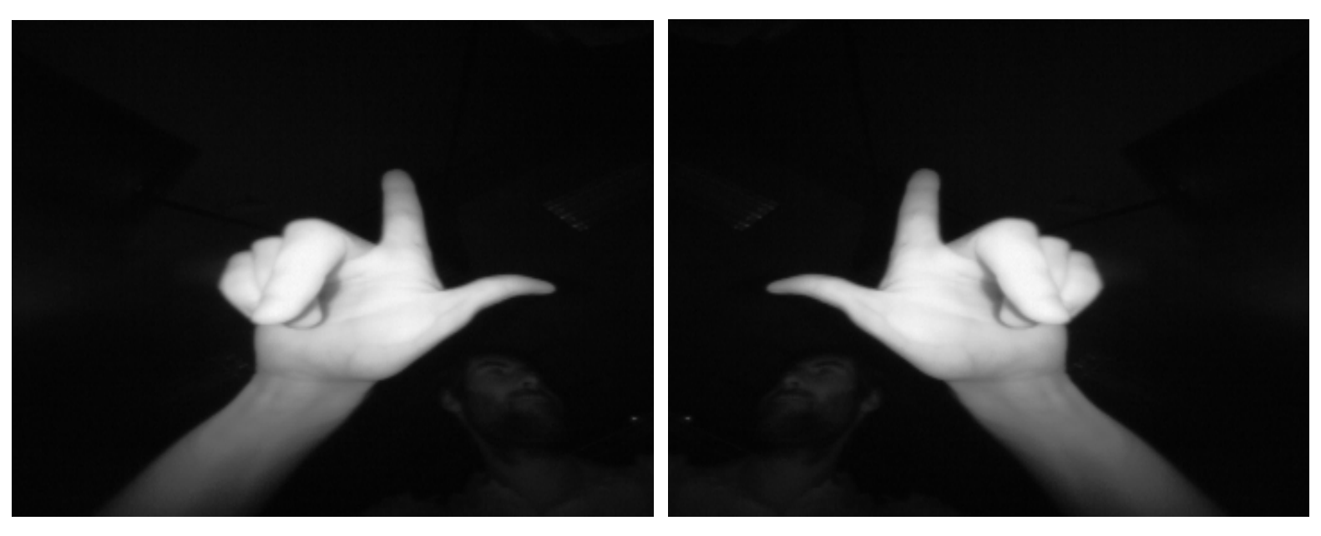

Figure 3. Mirrored images

Model

The Convolutional Neural Network consists of three convolution layers, each followed by Gaussian noise, batch normalization, and max pooling layers for downsampling with ReLU activation functions. Since we are using a neural network model, there are no handcrafted features, the model learns the relevant features itself. All 2D convolutional layers use a 3x3 kernel size with a dilation rate of 2. Increasing the dilation rate effectively increases the receptive field of the output. Max pooling also doubles the receptive field while reducing the number of parameters.

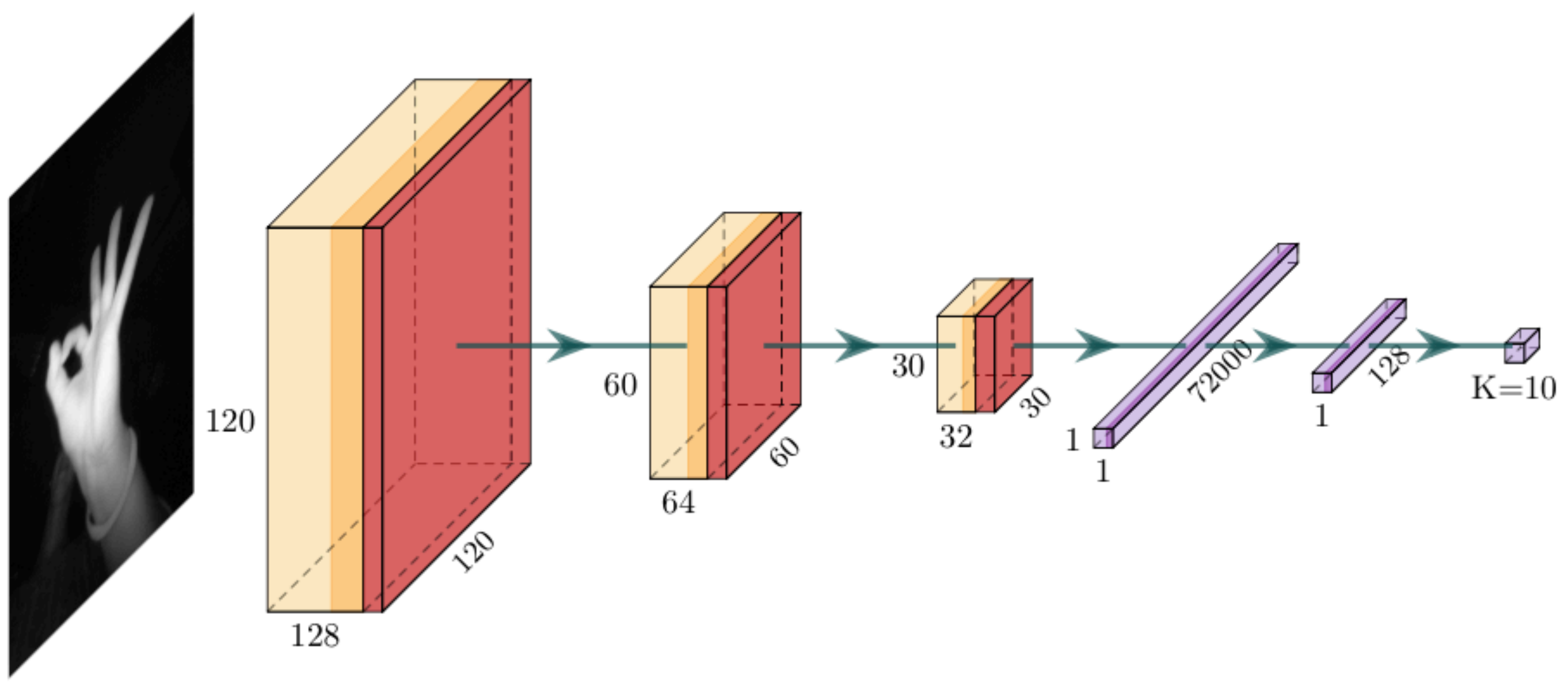

Figure 4. CNN model

The addition of Gaussian noise improves the robustness of the model to new images and prevents overfitting on the training data. The final layer of the model contains a softmax layer to appropriately classify one of the 10 hand gestures. The table below lists the specific parameters used to train the model.

Table 1. Model parameters

Learning Rate	0.01
Epochs	25
Batch Size	64
Loss Function	Cross entropy
Optimizer	RMSprop

Results

The training results show significant fluctuations in validation loss across the duration of the training. Since the validation set is struggling to stabilize while the training accuracy remains low, the network is most likely overfitting.

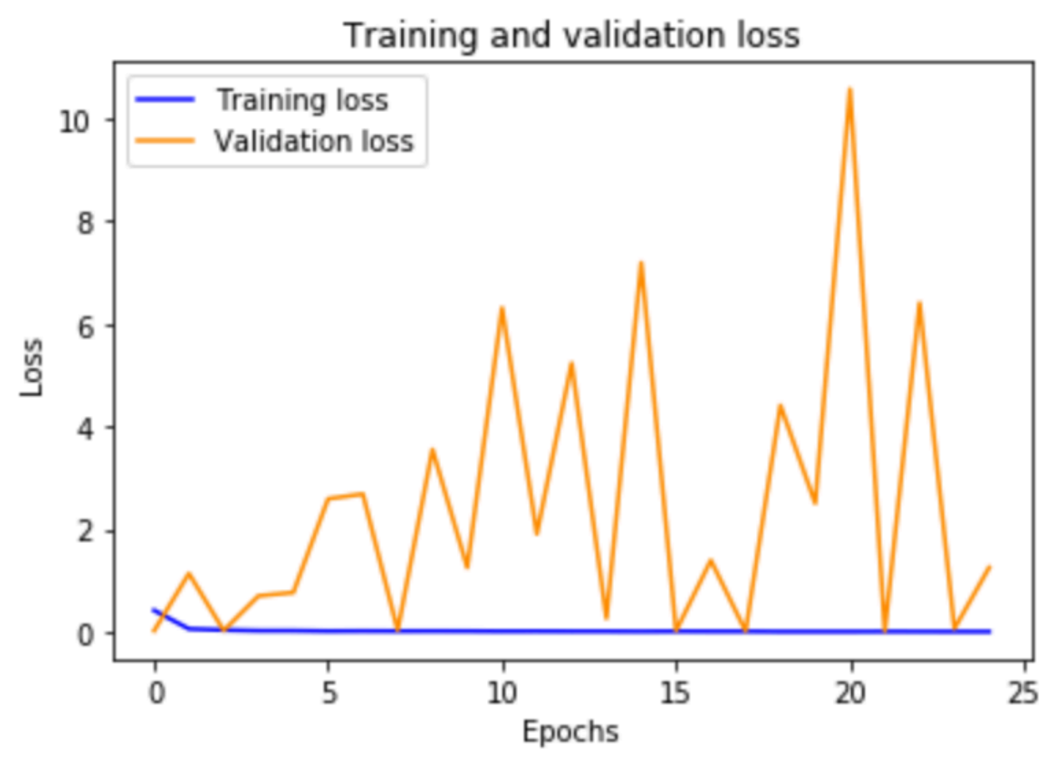

Figure 5. Training results

The testing results show about 90.2% accuracy, with a top-2 accuracy of 97.7% and a top-3 accuracy of 99.7%, so the predicted labels are generally within the top set of guesses.

Table 2. Test results

Test Loss	Test Accuracy	Top 2 Acc	Top 3 Acc
1.21636	0.902	0.977	0.997

The confusion matrix below shows significant misclassification for the ‘index’ class. Most of the testing images for the ‘index’ hand gesture was incorrectly classified as a ‘fist’.

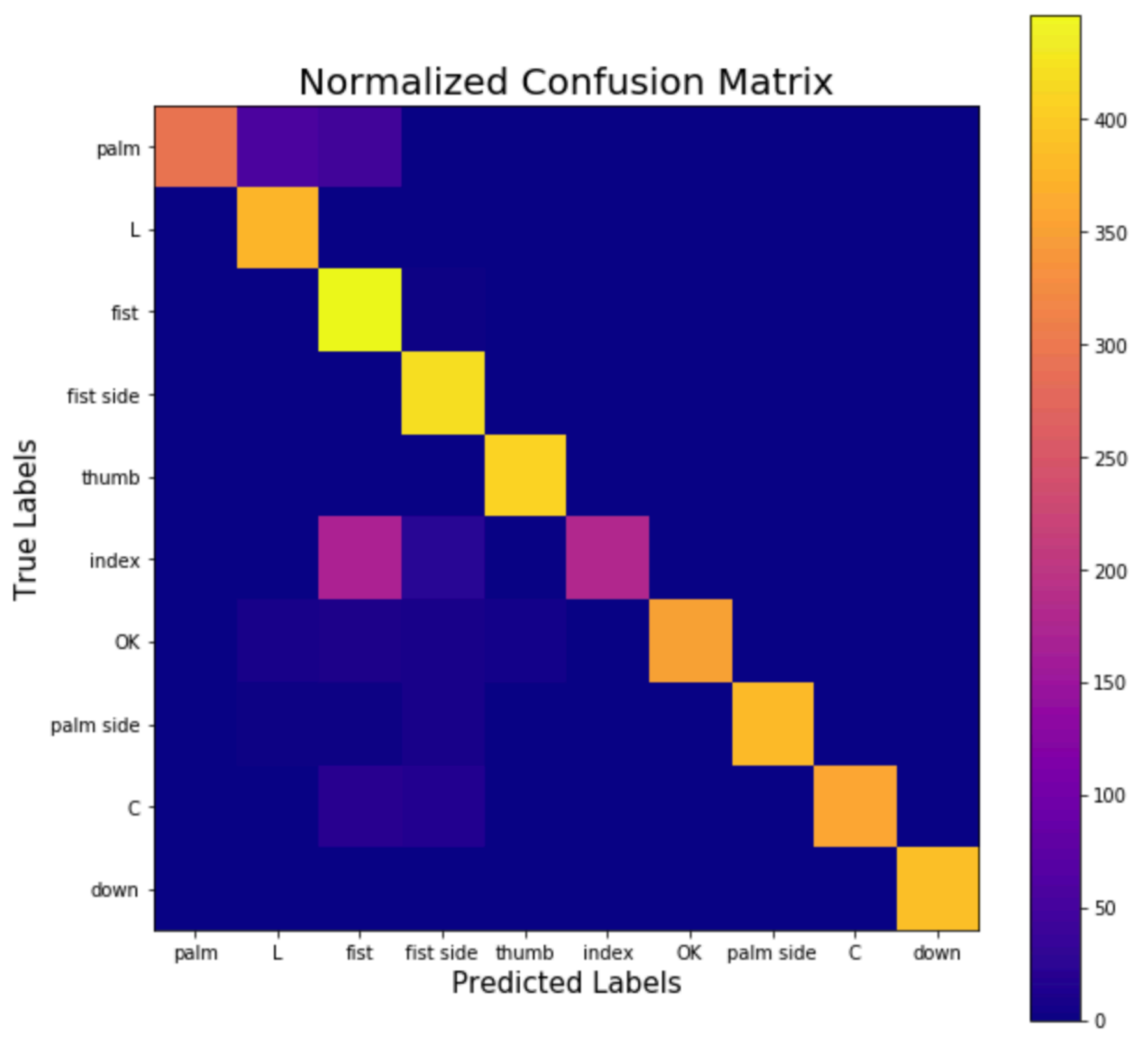

Figure 6. Confusion matrix

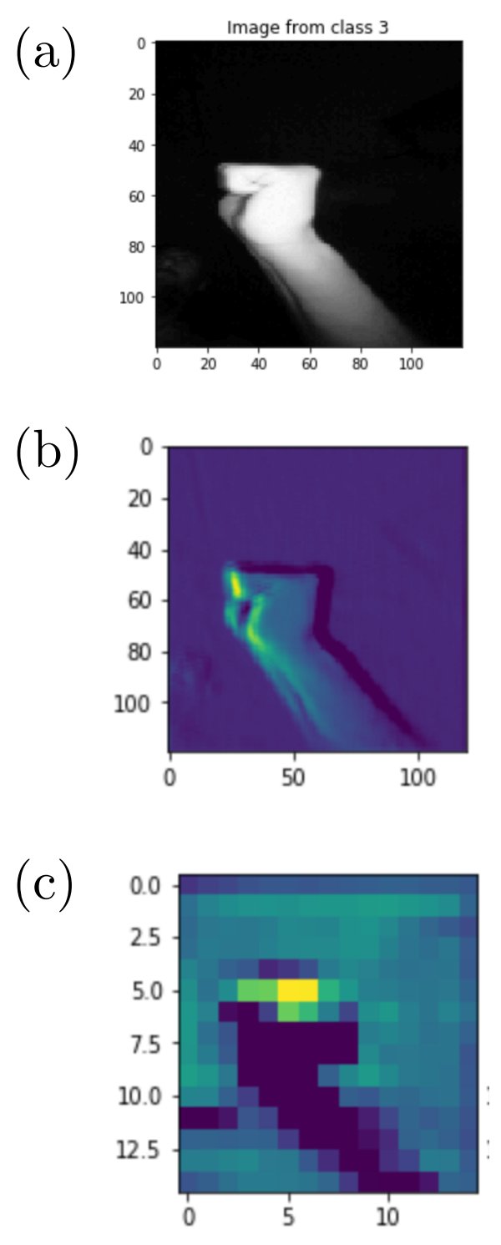

Figure 7. Layer visualization

The layer visualization above shows an original image (a), the output of the first convolutional layer (b), and the output of the last convolutional layer (c), illustrating the kinds of patterns the network is trying to learn from the images.

Discussion

Training the network on the original dataset gave almost perfect accuracy since there wasn’t drastic variability between the image samples, which is why the layers of noise and data augmentation of mirrored images was implemented. Although this decreased the accuracy, this made the model more robust to different gesture perspectives, however the model still experienced some overfitting during training. The variance in the validation loss is also likely because of the small batch size since there are less images to compute the error for each epoch. Misclassifications of the ‘index’ gesture as a ‘fist’ is likely due to the fact that the ‘index’ is essentially the ‘fist’ gesture with one finger raised, so the single finger can be hard to recognize. The layer visualizations are difficult to interpret but it give us insight on the “features” that the network is trying to learn, like gradients and shapes.

Conclusions

With 90.2% accuracy, the model implementation here still has room for improvement. For further advancement of this application, we can expand classification to several frames at once for real-time videos of hand gestures. With more complex 3D convolutional neural network architectures, this would allow us to translate live motion of changing gestures.

Contact

Arshia Zafari, azafari@ucsd.edu, A11167578
Erik Seetao, eseetao@ucsd.edu, A10705834
Hayk Hovhannisyan, hhovhann@ucsd.edu, A12466074
Silver De Guzman, j5deguzm@ucsd.edu, A53212113

References

[1] Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Jan Kautz, Hand Gesture Recognition with 3D Convolutional Neural Networks, IEEE 2015 CVPRW

[2] Abhishek Singh, asingh33/CNNGestureRecognizer: CNN Gesture Recognizer (Version 1.3.0), Zenodo. http://doi.org/10.5281/zenodo.1064825, Nov.2017

[3] T. Mantecn, C.R. del Blanco, F. Jaureguizar, N. Garca, Hand Gesture Recognition using Infrared Imagery Provided by Leap Motion Controller, Int. Conf. on Advanced Concepts for Intelligent Vision Systems, ACIVS 2016, Lecce, Italy, pp. 47-57, 24-27 Oct. 2016. (doi:10.1007/978-3-319-48680-25)

[4] Christian Zimmermann, Thomas Brox. Learning to Estimate 3D Hand Pose from Single RGB Images. In Proc. of the IEEE Conf. On International Conference on Computer Vision(ICCV), 2017

[5] H.Iqbal, P.Fernandez, W.Ji, L.Liebel, PlotNeuralNet, https://github.com/HarisIqbal88/PlotNeuralNet, 2019

[6] Leap Motion (2018), Hand Gesture Recognition Database. San Francisco, CA. https://www.leapmotion.com

[7] A. Kumar, Understanding CNN with Keras, https://www.kaggle.com/amarjeet007/visualize-cnn-with-keras

[8] A. Sharma, Autoencoder as a Classifier using Fashion-MNIST Dataset, 2018