Report Date: 10/14/2022
To: ematson@purdue.edu, ahsmith@purdue.edu, lee3450@purdue.edu
From: The Team Gangsture (Gesture Drone Control)

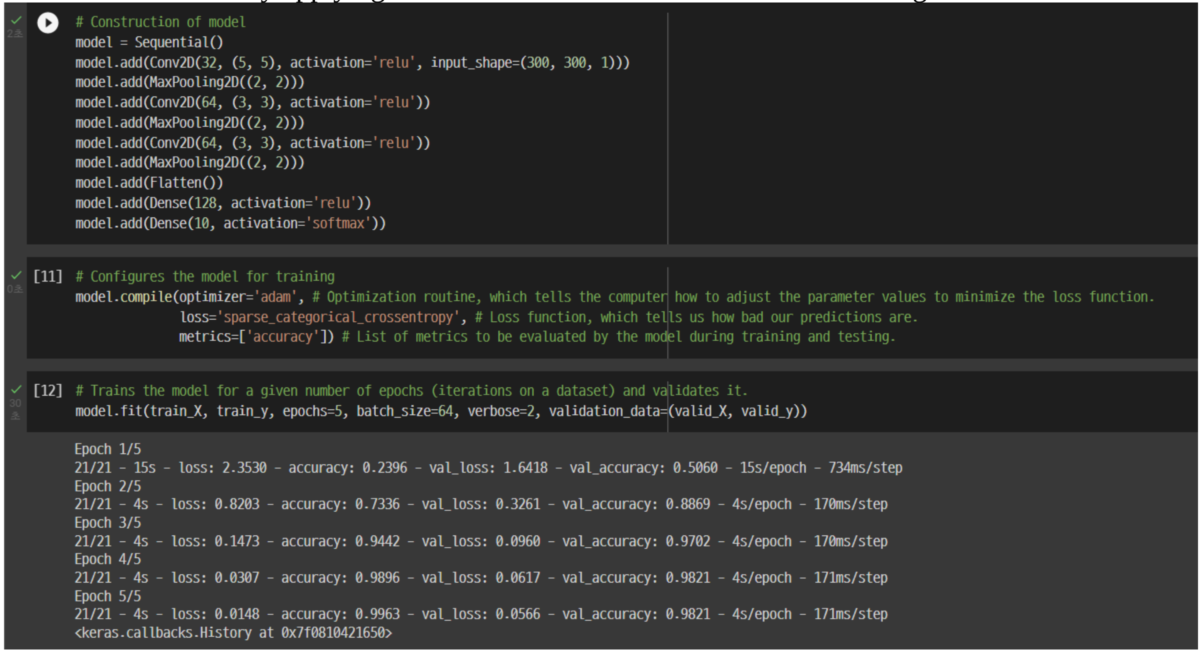
- Hyeongbin Park (The Leader / mtanger@kw.ac.kr)
- Seunghwan Kim (franz0602@stu.jejunu.ac.kr)
- Yujin Lee (yj5878@kw.ac.kr)
- Soeun Lee (thdms8477@jejunu.ac.kr)

Summary

The deep learning model was built and trained based on the various types of datasets and were tested. The hardware problem for training was fixed but, there are still practical challenges to use model. The server was developed. Studying and searching about CoreML is in progress.

What "Gangsture" completed this week:

- **Collect sample dataset**
  - The process to find out the optical dataset for the most effective model training.
  - It was decided to collect same quantity of dataset everyday but, there are some changes to environment such as, lightness, hand directions, data capacity, location, and time.
- **Build and train a deep learning model**
  - Regardless of library such as, Tensorflow and Pytorch. A few types of image deep learning model were built based on the collected dataset. They had different adjustment such as, epochs, the structure of layer, and hyper parameters.
  - Average accuracy of the simple CNN model was higher than 90% but, it needed to be tested by applying them to the real-world because of over-fitting.



```
# Construction of model
model = Sequential()
model.add(Conv2D(32, (5, 5), activation='relu', input_shape=(300, 300, 1)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(10, activation='softmax'))
```

```
[11] # Configures the model for training
     model.compile(optimizer='adam', # Optimization routine, which tells the computer how to adjust the parameter values to minimize the loss function.
                   loss='sparse_categorical_crossentropy', # Loss function, which tells us how bad our predictions are.
                   metrics=['accuracy']) # List of metrics to be evaluated by the model during training and testing.
```

```
[12] # Trains the model for a given number of epochs (iterations on a dataset) and validates it.
     model.fit(train_X, train_y, epochs=5, batch_size=64, verbose=2, validation_data=(valid_X, valid_y))

     Epoch 1/5
     21/21 - 15s - loss: 2.3530 - accuracy: 0.2396 - val_loss: 1.6418 - val_accuracy: 0.5060 - 15s/epoch - 734ms/step
     Epoch 2/5
     21/21 - 4s - loss: 0.8203 - accuracy: 0.7336 - val_loss: 0.3261 - val_accuracy: 0.8869 - 4s/epoch - 170ms/step
     Epoch 3/5
     21/21 - 4s - loss: 0.1473 - accuracy: 0.9442 - val_loss: 0.0960 - val_accuracy: 0.9702 - 4s/epoch - 170ms/step
     Epoch 4/5
     21/21 - 4s - loss: 0.0307 - accuracy: 0.9896 - val_loss: 0.0617 - val_accuracy: 0.9821 - 4s/epoch - 171ms/step
     Epoch 5/5
     21/21 - 4s - loss: 0.0148 - accuracy: 0.9963 - val_loss: 0.0566 - val_accuracy: 0.9821 - 4s/epoch - 171ms/step
     <keras.callbacks.History at 0x7f0810421650>
```

Fig. 1. The structure and accuracy of simple CNN model

- **Create sample code for applying the trained model**
  - After training, model was packaged and used in the sample web code. If the model got a input streaming video, the model would predict what is this gesture as label number. The model was trained on eight gestures so it would predict label from 1 to 8.

    - If there were some problems to recognize the gesture exactly, the model would be tuned and trained again, or the dataset used for training would be changed. This process is for finding the most robust and general AI.
- **Set a server**
    - The server was set by Google cloud platform.
- **Study and search about CoreML**
    - Studied integrating the example Keras model into CoreML.

Things to do by next week

- Collect and record results of dataset and model constantly
- Modify and look for a better deep learning model or algorithm of hand recognition
- Finish to write literature review of the draft paper
- Prepare the mid presentation and paper
- Conduct the test about UDP socket
- Try to integrate real model to CoreML

Problems or challenges:

- **Insufficient memory and GPU capacity for training AI**
    - The lack of memory and GPU is critical problem to train the model. This project has to train the large image dataset to model and it needs more capacity of hardware.
    - The easiest way to solve this problem is to use cloud training server such as Google Colab. But because of data capacity, it was not possible with a basic cloud server.
    - It was decided to buy Google Colab Pro+ and additional purchases would be mad after the trial period.
- **Using the trained model in practical situation**
    - Even though the accuracy of the model is higher than 90%, the real-world test was bad.
    - There are so many expected factors and to find out them is very complicated thing. This project has to keep track of and analyze everything related to this problem continuously.
    - For the mid presentation, the trained machine learning model would be used to show the result of recognition instead of the deep learning model.
- **Problem of setting server**
    - If the local server is used, there should be a laptop in this project.
    - This project's novelty is convenient for equipment.
    - As the result, Google cloud platform will be used.

References

[1] filipefborba, "filipefborba/HandRecognition : Machine Learning example project using images of hand gestures," *GitHub*. [Online]. Available: https://github.com/Kazuhito00/hand-gesture-recognition-using-mediapipe. ****[Accessed: 14-Oct-2022].

[2] Andrew Ng, "Classifying Images of Hand Signs: Defining a Convolutional Network and Loading Image Data," [Online]. Available: https://cs230.stanford.edu/blog/handsigns/. [Accessed: 14-Oct-2022].

[3] “Deep Learning for Hand Gesture Recognition with PyTorch,” [Online]. Available: https://people.minesparis.psl.eu/fabien.moutarde/ES_MachineLearning/mini-projets/devineau_2018_deep_learning_hand_gesture_pytorch_model_quickstart.html. [Accessed: 14-Oct-2022].

[4] Fabiopk, “Fabiopk/RT_GESTURERECOGNITION: Real time gesture recognition using pytorch,” *GitHub*. [Online]. Available: https://github.com/fabiopk/RT_GestureRecognition. [Accessed: 14-Oct-2022].

[5] S. Som, “Image classification using pytorch,” *Medium*, 13-Aug-2020. [Online]. Available: https://blog.jovian.ai/image-classification-american-sign-language-using-pytorch-7bef9d7e8a25. [Accessed: 14-Oct-2022].

[6] A. Rosebrock, “PyTorch: Training your first Convolutional Neural Network (CNN),” *PyImageSearch*, 01-Jun-2021. [Online]. Available: https://pyimagesearch.com/2021/07/19/pytorch-training-your-first-convolutional-neural-network-cnn/. [Accessed: 14-Oct-2022].

[7] “Hand gesture recognition,” **Papers With Code**. [Online]. Available: https://paperswithcode.com/task/hand-gesture-recognition. [Accessed: 14-Oct-2022].