Report Date: 11/04/2022
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

All of the development parts and functions were finished to apply to the iOS program. The last feature of the project is to train the deep learning model for face recognition and integrate all sub-partitions. Additional research was conducted to improve the draft paper.

## What “Gangsture” completed this week:

- **Converted the machine learning model to CoreML format**
  - The trained machine learning model has to be converted to CoreML format for integrating into the iOS platform [1], [2].
  - The model was converted to CoreML format successfully and conveyed to the iOS development part.
- **Collected face datasets with webcam**
  - Face recognition function would give authority to the specific person for controlling the drone. It needs to train the deep learning model and also specific face datasets.
  - The collecting process only considered face angle, direction, and size. Therefore, this is different from the hand dataset collection process. Face datasets were collected based on certificated project members only through the webcam.
- **Trained the deep learning model to recognize the specific face**
  - For facial recognition, there are several steps such as, detecting face, calculating the embedding vector, preprocessing the input vector, and classifying faces through the model [3].
  - MTCNN (Multi-task Cascaded Convolutional Network) was used as a facial detector and FaceNet was used for creating face embedding vectors. Finally, SVM (Support Vector Machine) was used for classification of faces based on the input vector [4].
  - This process was developed on Google Colab and Jupyter environment. Entire training time was about 1~2 hours at once.
  - The laptop could immediately check the battery condition of the drone.
  - The drone could record videos and store them on the laptop [5].
- **Developed for saving video which the drone watches on Google drive automatically**
  - Using OpenCV, it is possible to fly while using a drone camera. The video could be checked on the laptop's screen in real time, recorded, and saved by pressing a button on the laptop using libraries: cv2, thread, and djitellopy [5], [6].
  - Saving the video on Google Drive was developed using Python libraries, GoogleAuth and GoogleDrive. Originally, it needed to be authorized manually each time. However, it was modified to use a text file which contains the authorization key to upload instantaneously [7].

Fig. 1. The video taken by a flying drone

- **Studied how to use Mediapipe in the iOS application**
    - Landmark coordinate systems of gesture are inputs of our own model.
    - MediaPipe is needed to gain landmark coordinate systems of gestures from a mobile camera.
- **Studied and searched about NUI (Natural User Interface) and drone market report**
    - Additional research about NUI and drone market growth was conducted.
    - The introduction and literature review part of the paper needed more details such as, statistics of drone market growth and the latest research trends of NUI [8], [9].

## Things to do by next week

- Edit the draft of paper, introduction, and literature review including contents about NUI and the commercial drone market
- Optimize and apply the trained deep learning model on the web application
- Change the Python code of saving the video on Google Drive to Swift
- Extract MediaPipe into a framework for use on Swift
- Control the drone using Swift

## Problems or challenges:

- **Integration issues with the machine learning model to the iOS platform**
    - There were a lot of issues with integrating, such as library conflicts, unclear official API document, and packing failure.
    - All libraries should be same with the model training environment and each packing library supports different functions. It needed to consider what types of packing library will be used and which structure has to be packaged [10].
- **Optimization problem of the deep learning model for face recognition**
    - The process for face recognition needs three models, MTCNN, FaceNet and SVM, and it means that a lot of computation will occur.
    - Some models should be replaced with computer vision technology to reduce the amount of computation. For example, MTCNN would be replaced with the facial recognition function for OpenCV.
- **Necessity of the internet to save the video automatically**

  - One of the novelties is using no extra devices but, a connection to the internet is necessary to save the video automatically while having a connection to the drone.
  - The iOS code development was discussed to connect to the drone and record, save, and upload the video to Google Drive by disconnecting from the drone and connecting to a network with the internet then uploading the video.

- **Integrating Mediapipe Graph on the iOS**
  - MediaPipe Graph Solution is only provided in Objective-C (iOS language prior to Swift)
  - The Mediapipe solution is implemented in Objective-C and does not support Swift. It should be created as a framework to be used in Swift [11].
  - There is no official document to switch to the framework so, it has to be researched [12], [13].

## References

[1] A. Singh, “Converting pytorch & tensorflow models into Apple Core ML using CoreMLTools,” *Analytics India Magazine*, 01-May-2021. [Online]. Available: https://analyticsindiamag.com/converting-pytorch-and-tensorflow-models-into-apple-core-ml-using-coremltools/. [Accessed: 04-Nov-2022].

[2] “Beginning machine learning with scikit-learn,” *kodeco.com*. [Online]. Available: https://www.kodeco.com/174-beginning-machine-learning-with-scikit-learn. [Accessed: 04-Nov-2022].

[3] J. Brownlee, “How to develop a face recognition system using FaceNet in Keras,” *Machine Learning Mastery*, 23-Aug-2020. [Online]. Available: https://machinelearningmastery.com/how-to-develop-a-face-recognition-system-using-facenet-in-keras-and-an-svm-classifier/. [Accessed: 04-Nov-2022].

[4] S. Ghosh, “How to create a face recognition model using FaceNet Keras?,” *Medium*, 10-Jul-2020. [Online]. Available: https://medium.com/clique-org/how-to-create-a-face-recognition-model-using-facenet-keras-fd65c0b092f1. [Accessed: 04-Nov-2022].

[5] “CV::VideoWriter class reference,” *OpenCV*. [Online]. Available: https://docs.opencv.org/3.4/dd/d9e/classcv_1_1VideoWriter.html. [Accessed: 04-Nov-2022].

[6] “Saving operated video from a webcam using opencv,” *GeeksforGeeks*, 11-Jul-2022. [Online]. Available: https://www.geeksforgeeks.org/saving-operated-video-from-a-webcam-using-opencv/. [Accessed: 04-Nov-2022].

[7] “Introduction to google drive API | google developers,” *Google*. [Online]. Available: https://developers.google.com/drive/api/guides/about-sdk. [Accessed: 04-Nov-2022].

[8] H. Liu, Y. Xi, W. Song, K. Um and K. Cho, "Gesture-Based NUI Application for Real-Time Path Modification," 2013 IEEE 11th International Conference on Dependable, Autonomic and Secure Computing, 2013, pp. 446-449, doi: 10.1109/DASC.2013.104.

[9] M. S. Sidhu and J. Iqbal, "Exploring Dancer Experience and Performance with Visual Sensor Camera Using NUI Environment," 2018 8th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), 2018, pp. 160-165, doi: 10.1109/ICCSCE.2018.8684995.

[10] “XGBoost,” *coremltools*. [Online]. Available: https://coremltools.readme.io/docs/xgboost-conversion. [Accessed: 04-Nov-2022].

[11] Google, “Swift support for mediapipe · issue #141 · google/mediapipe,” *GitHub*. [Online]. Available: https://github.com/google/mediapipe/issues/141. [Accessed: 04-Nov-2022].

[12] Google, “What is the way to dependent mediapipe into my IOS app? · issue #364 · google/mediapipe,” *GitHub*. [Online]. Available: https://github.com/google/mediapipe/issues/364. [Accessed: 04-Nov-2022].

[13] P. gg, “MediaPipe tutorial: Find memes that match your facial expression” *Medium*, 09-Jun-2020. [Online]. Available: https://powderapp.medium.com/mediapipe-tutorial-find-memes-that-match-your-facial-expression-9bf598da98c0. [Accessed: 04-Nov-2022].