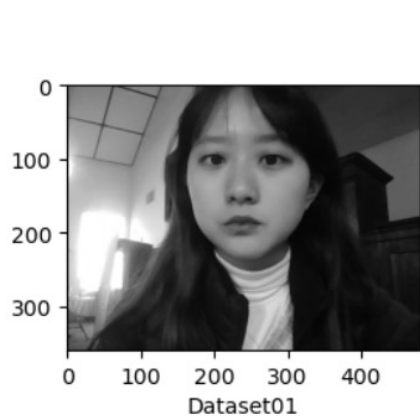
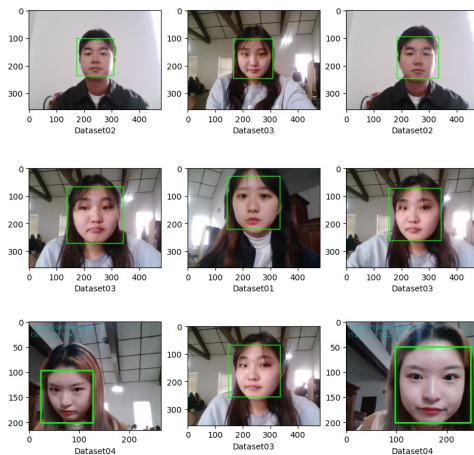


# 3. Methodology

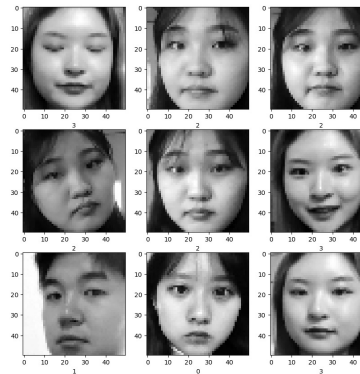
## Face Recognition with Computer Vision and ML: Preprocessing



1. Read the face image dataset as a gray scale.



2. Extract the face from the gray scale image using the **Haar Cascade Classifier**.



3. Crop and save the face based on the returned coordinates.

```
[[-0.0083408, 0.09042292, 0.00678223, 0.01164577, -0.07406723,
-0.0077776, -0.04922447, -0.03191379, 0.06613824, -0.03972566,
0.24079905, -0.00624357, -0.2239792, -0.03049533, -0.00640078,
0.09627665, -0.11431439, -0.05446113, -0.1398232, -0.12352334,
0.02760254, -0.01702521, 0.01452616, -0.02035574, -0.09400257,
-0.27898463, -0.00170043, -0.1107812, 0.00082851, -0.15381731,
0.04318428, 0.01764483, -0.12904049, -0.01124948, -0.02029468,
-0.0199079, 0.03557100, -0.05274077, 0.15050083, -0.01535819,
-0.12579067, -0.00784833, 0.01654950, 0.22943080, 0.22063054,
0.0203871, 0.02551018, -0.05526670, 0.04759990, 0.24063045,
0.00719433, 0.1153003, 0.0610044, 0.00087353, 0.0738312,
-0.14302395, 0.02553764, 0.05797878, -0.13036034, 0.0264425,
0.05215002, -0.1176679, 0.07930094, -0.00545089, 0.20140012,
0.00059311, -0.07800453, -0.0011657, 0.17147460, -0.10824564,
-0.02400123, 0.04396733, -0.10081847, -0.12064815, -0.24433994,
0.11080403, 0.3787249, 0.19050116, -0.10030423, 0.02395504,
-0.10567383, -0.00640407, 0.0141305, -0.00051829, -0.1055201,
-0.03354408, -0.00343011, 0.0790627, 0.16501495, -0.05444223,
0.05157606, 0.11057329, -0.0400572, 0.03054036, 0.01070801,
-0.02199657, -0.00801046, -0.0190808, -0.05059837, -0.00757048,
0.10258103, -0.15714751, 0.057192, 0.00739022, -0.10033116,
0.00737358, -0.03444197, -0.04479152, 0.02779513, 0.05365732,
-0.03450293, -0.0205511, 0.20151009, -0.17340081, 0.23824009,
0.21202017, 0.00729257, 0.10242070, 0.005013, 0.12340066,
-0.00919439, -0.01220037, -0.11564347, -0.04238573, -0.0232062,
-0.0100737, -0.010027, 0.00940612]],
```

4. Encode the saved face using the Python face recognition library

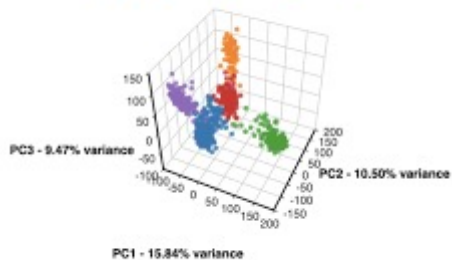
\* Haar Cascade Classifier:

A machine learning-based object detection algorithm.

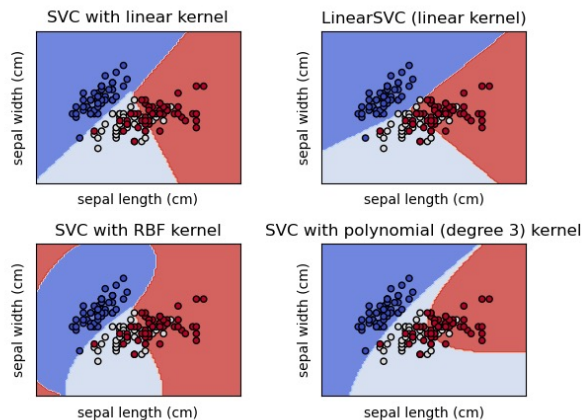
It is used to detect objects in a video or image based on a feature. It returns coordinate value of face position.

# 3. Methodology

## Face Recognition with Computer Vision and ML: Training and Evaluation



5. Compress the encoded high-dimensional values with **PCA**.



6. Process face classification with **SVM** trained on the face feature vectors.

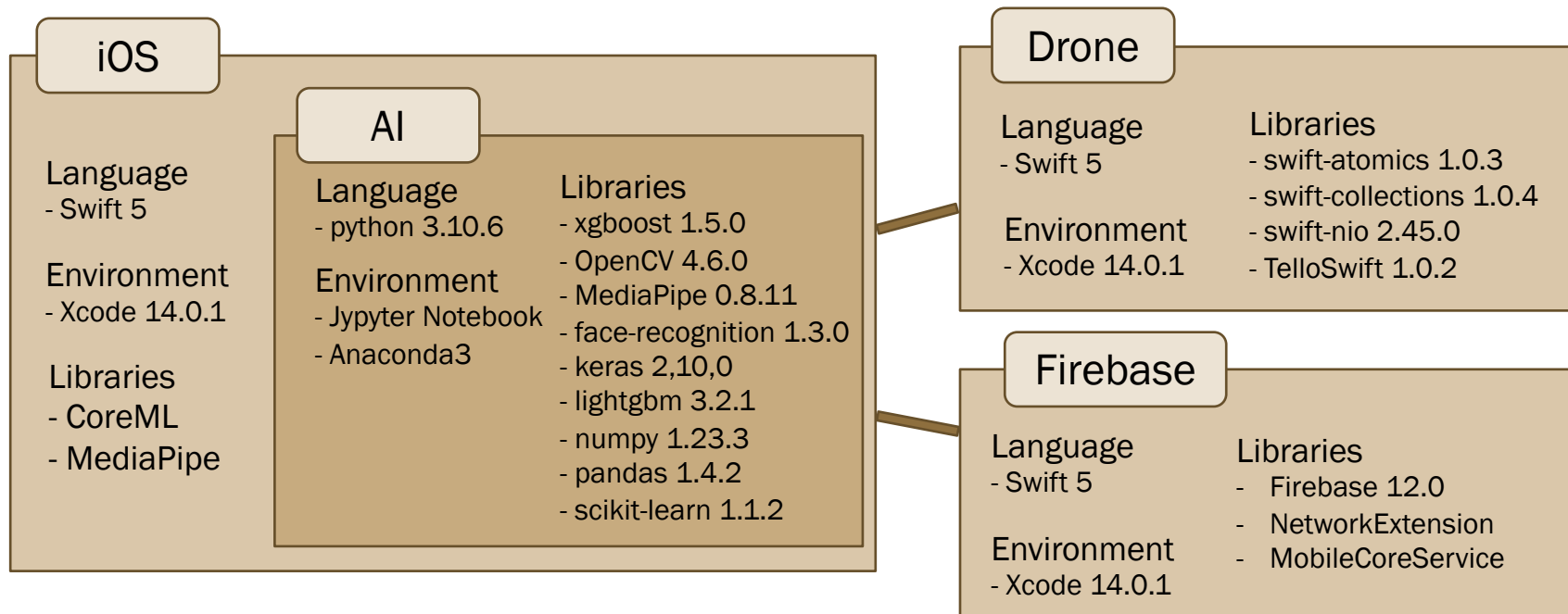
	precision	recall	f1-score	support
0	1.00	1.00	1.00	58
1	1.00	1.00	1.00	69
2	1.00	1.00	1.00	57
3	1.00	1.00	1.00	49
accuracy			1.00	233
macro avg	1.00	1.00	1.00	233
weighted avg	1.00	1.00	1.00	233

accuracy score: 1.0

- Multi-Class Classification (4 class)
- SVM with radial basis function kernel
- Evaluation Metrics  
Precision / Recall / F1 Score / Accuracy
- Train/Test and Train/Valid as 8:2

# 4. Implementation

## System Architecture with Dependencies



## 4. Implementation

### System flow 1. Face recognition in iOS application




Detect facial features through edge detector



The output is compressed to list and entered to the AI model.

## 4. Implementation

### System flow 2. If it is true, do hand detection in iOS app



Hand detection  
사진

- The front camera's screen is delivered to the MediaPipe every moment and the MediaPipe provides 21 normalized landmarks.
- The AI model using Python was packed in CoreML format , iOS's artificial intelligence framework, and added to the iOS.
- CoreML allows you to obtain the result value of artificial intelligence embedded in the application

## 4. Implementation

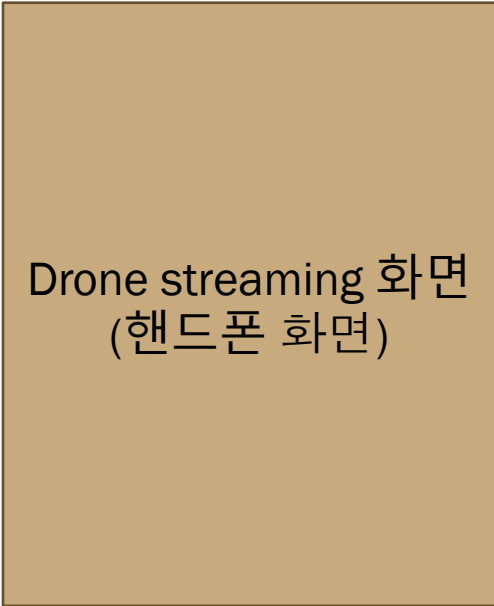
### System flow 3. Send command using UDP socket



- Set up a UDP client on the mobile device to send and receive messages from the Tello via the same port.
- Before sending any other commands, send “command” to the Tello via UDP PORT 8889 to initiate SDK mode.

## 4. Implementation

### System flow 4. Streaming and Recording while controlling

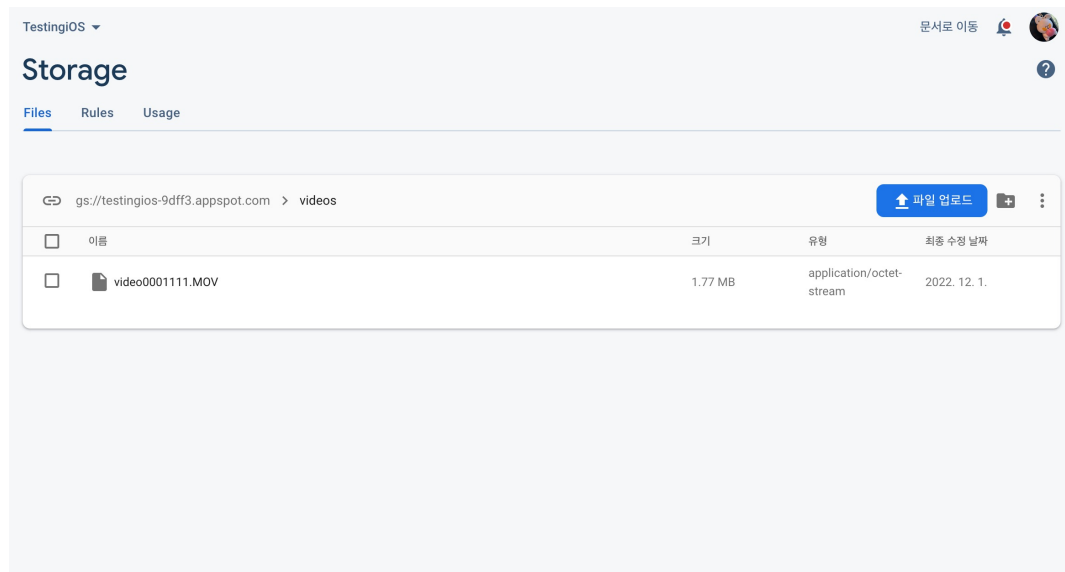


Drone streaming 화면  
(핸드폰 화면)

- Translated to h264 format through AVFoundation and transmitted to the iOS screen in the real-time
- Automatically stored when the flight is over
- Video recordings take place in separate threads.

## 4. Implementation

### System flow 5. Upload to Firebase

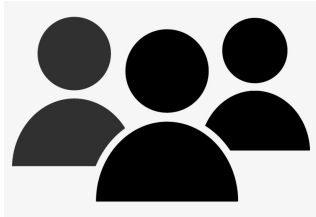


iOS UI 화면



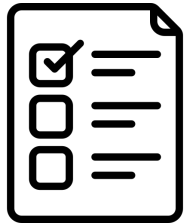
# 4. Implementation

## Experiment Design with Drone Gesture Controller



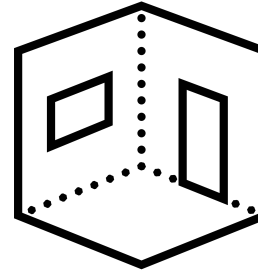
6 Subjects

- People who haven't used a drone before
- Three women and three men
- Over 22 ages



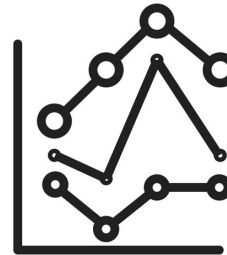
2 types of tasks

- One is to check up and down movement
- One is to check left and right movement
- Common task is reading over obstacles
- Each three people is assigned a task



Environment

- KSW Square building
- With 4 assistant for safe experiment
- One obstacle
- One sign over the obstacle



Evaluation

- Total experiment time
- Time to pass each task during the experiment
- Total number of trials
- Number of successes and failure
- Number of hits

# 5. Conclusion

## Experiment Results

Subject Number	Trial	Success	Fail	Hits	Total Time	T1	T2	T3	T4	Note
1	2	1	1	1	103	29.5	35	25	29	hit
2	3	1	2	1	132.67	52	44	18	50	hit and landing miss
3	1	1	0	0	136	35	43	15	53	
4	1	1	0	0	197	38	26	20	113	
5	1	1	0	0	116	29	22	9	56	
6	2	1	1	1	123.5	53.5	41	25	74	hit
Total	1.67	1.00	0.67	0.50	134.69	39.50	35.17	18.67	62.50	

## 5. Conclusion

### Demo and Future Plan

데모영상



It is impossible to control the drone with gestures at a moment when face is invisible.  
Even if the face detection is cut off for a while, control will not be affected

## Reference

[polytechnic.purdue.edu](https://polytechnic.purdue.edu)

