



**ESPE**  
**UNIVERSIDAD DE LAS FUERZAS ARMADAS**  
**INNOVACIÓN PARA LA EXCELENCIA**

**Reconocimiento de gestos corporales basado en SVM e imágenes monoculares para el control de micro vehículos aéreos multirrotor embebido en una aplicación móvil en tiempo real**

Morales Bustillos, Kevin Freddy

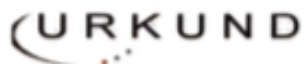
Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización y Control

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica,  
Automatización y Control

Ph.D. Aguilar Castillo, Wilbert Geovanny

24 de febrero del 2021



## Urkund Analysis Result

Analysed Document: Tesis Morales.pdf (D96577111)  
Submitted: 2/26/2021 1:40:00 AM  
Submitted By: kfmorales@espe.edu.ec  
Significance: 4 %

### Sources included in the report:

TESIS\_George\_Bryan\_Cobeña\_Zambrano.pdf (D35797794)  
TESIS\_CHAUCA\_BRYAN\_urkund.pdf (D78121896)  
Proyecto de Investigación - Alvarez Leandro.pdf (D44976218)  
<https://repositorio.upct.es/xmlui/bitstream/handle/10317/6446/tfg-ros-sis.pdf?sequence=1&isAllowed=y>

### Instances where selected sources appear:

18

Firma:



**Ph.D. Aguilar Castillo, Wilbert Geovanny**

C. C.: 0703844696



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA Y  
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y  
CONTROL**

**CERTIFICACIÓN**

Certifico que el trabajo de titulación, “**Reconocimiento de gestos corporales basado en SVM e imágenes monoculares para el control de micro vehículos aéreos multirotor embebido en una aplicación móvil en tiempo real**” fue realizado por el señor **Morales Bustillos Kevin Freddy** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 24 de febrero de 2021

Firma:



**Ph.D. Aguilar Castillo, Wilbert Geovanny**

C. C.: 0703844696



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA Y  
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y  
CONTROL**

**RESPONSABILIDAD DE AUTORÍA**

Yo, **Morales Bustillos Kevin Freddy**, con cédula de ciudadanía n° 1717893901, declaro que el contenido, ideas y criterios del trabajo de titulación: **Reconocimiento de gestos corporales basado en SVM e imágenes monoculares para el control de micro vehículos aéreos multirotor embebido en una aplicación móvil en tiempo real** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 24 de febrero de 2021

Firma

.....  
**Morales Bustillos Kevin Freddy**

C.C.: 1717893901



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA Y  
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y  
CONTROL**

**AUTORIZACIÓN DE PUBLICACIÓN**

Yo **Morales Bustillos Kevin Freddy**, con cédula de ciudadanía n° 1717893901, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Reconocimiento de gestos corporales basado en SVM e imágenes monoculares para el control de micro vehículos aéreos multirrotor embebido en una aplicación móvil en tiempo real** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 24 de febrero de 2021

Firma

**Morales Bustillos Kevin Freddy**

C.C.: 1717893901

## Dedicatoria

El presente trabajo investigativo se lo dedico principalmente a Dios, por ser el inspirador y darme fuerza para continuar en este proceso de obtener uno mis anhelos más deseados.

A mi hija Madison Morales Jácome por ser mi principal pilar en mi vida, darme la fuerza y motivación para seguir adelante para cumplir con todos mis sueños.

A mis padres, Freddy Morales Sotomayor y Verónica Bustillos Zambrano por su amor, trabajo y sacrificio en todos estos años, gracias a ustedes he logrado llegar hasta aquí convertirme en lo que soy. A mi hermana Kelly Morales Bustillos por ser una motivación diaria para superarme en mis estudios.

A mi esposa Jimena Jácome Delgado por su amor incondicional y estar siempre presente, acompañándome en buenos y malos momentos, brindándome una de las mejores etapas de mi vida.

A todas las personas que me han apoyado y han hecho que el trabajo se realice con éxito en especial a mis amigos que me abrieron las puertas y compartieron sus conocimientos.

## **Agradecimiento**

Agradezco a Dios por bendecirme en esta vida, por guiarme a lo largo de mi existencia, ser el apoyo y fortaleza en aquellos momentos de dificultad y de debilidad.

A mi hija Madison Morales Jácome y a mi esposa Jimena Jácome Delgado por el amor incondicional y ser una motivación a diario para superarme en mis metas y conseguir mis sueños.

A mis padres Freddy Morales Sotomayor y Verónica Bustillos Zambrano, a mi hermana Kelly Morales Bustillos, por ser los principales promotores de mis sueños, por confiar y creer en mis expectativas, por los consejos, valores y principios que me han inculcado.

Finalmente agradezco al Dr. Wilbert Aguilar de la Universidad de las Fuerzas Armadas – ESPE, por haber compartido sus conocimientos a lo largo del proyecto de investigación, guiándome con su paciencia, y su rectitud como docente.

## Índice de contenido

<i>Certificación</i> .....	3
<i>Responsabilidad de autoría</i> .....	4
<i>Autorización de publicación</i> .....	5
<i>Dedicatoria</i> .....	6
<i>Agradecimiento</i> .....	7
<i>Índice de contenido</i> .....	8
<i>Índice de tablas</i> .....	11
<i>Índice de figuras</i> .....	12
<i>Resumen</i> .....	17
<i>Abstract</i> .....	18
<i>Capítulo I</i> .....	19
<i>Introducción</i> .....	19
<i>Antecedentes</i> .....	19
<i>Justificación e importancia</i> .....	22
<i>Alcance del proyecto</i> .....	24
<i>Objetivos</i> .....	26
<i>Objetivo general</i> .....	26
<i>Objetivo específico</i> .....	26
<i>Capítulo II</i> .....	28



<b>Estado del arte .....</b>	<b>28</b>
Métodos de caracterización de gestos corporales.....	28
Tipos de extracción de características .....	29
Tipos de clasificadores.....	31
Métodos de clasificación para identificar los gestos corporales .....	31
Aplicaciones de los micro vehículos aéreos multirotor.....	33
<b>Capítulo III.....</b>	<b>36</b>
<b>Desarrollo del algoritmo .....</b>	<b>36</b>
Introducción .....	36
Análisis de esqueletización y extracción de características .....	37
ANÁLISIS MATEMÁTICO DE SVM.....	39
Análisis de imágenes monoculares y articulaciones .....	41
Desarrollo de algoritmo para clasificación de gestos mediante SVM .....	48
<b>Capítulo IV .....</b>	<b>53</b>
<b>Desarrollo de la aplicación móvil .....</b>	<b>53</b>
Introducción .....	53
DESARROLLO DE INTERFAZ MÓVIL.....	53
CONFIGURACIÓN DE LIBRERÍAS PARA EL DESARROLLO.....	57
Desarrollo de interfaz para captura de datos .....	62
Comunicación de la interfaz y control del micro vehículo aéreo .....	67
<b>Capítulo V .....</b>	<b>76</b>

<b><i>Pruebas y resultados</i></b> .....	<b>76</b>
<b><i>Pruebas</i></b> .....	<b>76</b>
<b><i>Análisis de resultados</i></b> .....	<b>112</b>
<b><i>Trabajos futuros</i></b> .....	<b>112</b>
<b><i>Capítulo VI</i></b> .....	<b>114</b>
<b><i>Conclusiones y recomendaciones</i></b> .....	<b>114</b>
<b><i>Conclusiones</i></b> .....	<b>114</b>
<b><i>Recomendaciones</i></b> .....	<b>115</b>
<b><i>Referencias</i></b> .....	<b>117</b>

**Índice de tablas**

Tabla 1 <i>Tabla de dificultad en el ambiente de la mañana en el territorio 1</i> .....	90
Tabla 2 <i>Tabla de dificultad en el ambiente de la mañana en el territorio 2</i> .....	93
Tabla 3 <i>Tabla de dificultad en el ambiente de la mañana en el territorio 3</i> .....	96
Tabla 4 <i>Tabla de dificultad en el ambiente de la tarde en el territorio 1</i> .....	99
Tabla 5 <i>Tabla de dificultad en el ambiente de la tarde en el territorio 2</i> .....	102
Tabla 6 <i>Tabla de dificultad en el ambiente de la tarde en el territorio 3</i> .....	105
Tabla 7 <i>Tabla de dificultad en el ambiente de la noche en el territorio 1</i> .....	109
Tabla 8 <i>Tabla de dificultad en el ambiente de la noche en el territorio 2</i> .....	110
Tabla 9 <i>Tabla de dificultad en el ambiente de la noche en el territorio 3</i> .....	111

## Índice de figuras

Figura 1 <i>Diagrama del proyecto de investigación.....</i>	26
Figura 2 <i>Ejemplo de cuadros para extracción de características de HAAR.....</i>	29
Figura 3 <i>Diagrama de extracción de características de una imagen.....</i>	30
Figura 4 <i>Diagrama de Core ML para la integración del archivo Core ML model en un aplicativo de ios realizado con Swift.....</i>	36
Figura 5 <i>Diagrama de flujo de Core ML.....</i>	37
Figura 6 <i>Imágenes para el movimiento de aterrizaje del drone en el dashboard de Microsoft Custom Vision, donde se agregarán todas las imágenes de acuerdo a los modelos creados....</i>	39
Figura 7 <i>Subida de imágenes a Custom Vision de cada combo a su respectivo modelo creado.</i>	41
Figura 8 <i>Gesto para función despegue del Bebop 2.....</i>	42
Figura 9 <i>Gesto para función aterrizaje del Bebop 2.....</i>	42
Figura 10 <i>Gesto para función hacia la derecha del Bebop 2.....</i>	43
Figura 11 <i>Gesto para función hacia la izquierda del Bebop 2.....</i>	43
Figura 12 <i>Gesto para función hacia adelante del Bebop 2.....</i>	44
Figura 13 <i>Gesto para función hacia atrás del Bebop 2.....</i>	44
Figura 14 <i>Gesto para función sobre el eje del Bebop 2.....</i>	45
Figura 15 <i>Datos de precisión de Custom Vision, con estos datos se observa la precisión del modelo construido para su uso.....</i>	45
Figura 16 <i>Exportación de modelo en Core ML. Desde este espacio se exporta el modelo para importarlo en el proyecto de Xcode.....</i>	47
Figura 17 <i>Modelo de Core ML, el cual servirá para importar en el proyecto de Xcode y poder utilizarlo en el aplicativo. Este archivo es el modelo el cual ayuda a reconocer los gestos con la cámara.....</i>	48
Figura 18 <i>Diagrama de bloques del algoritmo de validación cruzada K-Fold que describe el proceso de búsqueda de cuadrícula y validación cruzada de k-fold para el modelo SVM / SVR.</i>	49

Figura 19 Proyecto de titulación en Custom Vision, el proyecto creado para este aplicativo. ...	51
Figura 20 Bloque de funcionalidad de la arquitectura MVC.....	54
Figura 21 Inicio del IDE Xcode, este es el inicio de un proyecto de una aplicación para iOS. ....	55
Figura 22 Configuración inicial del proyecto en Xcode.....	56
Figura 23 Pantalla inicial del proyecto en Xcode. ....	57
Figura 24 Carpetas al descomprimir el archivo SDK de Parrot. ....	58
Figura 25 Vista de BUILD SETTINGS en Xcode.....	59
Figura 26 Vista de HEADER SEARCH PATHS en Xcode. ....	60
Figura 27 Vista de LIBRARY SEARCH PATHS en Xcode.....	60
Figura 28 Vista de OTHER LINKER FLAGS en Xcode.....	61
Figura 29 Vista inicial de la aplicación móvil, en este caso las credenciales son por conveniencia del autor.....	62
Figura 30 Botón de despegue, el cual al presionar el drone ascenderá. ....	63
Figura 31 Botón de aterrizaje, el cual al presionar el drone descenderá. ....	63
Figura 32 Botón de avance, el cual al presionar el botón avanzará. ....	64
Figura 33 Botón de izquierda, el cual al presionar el drone se dirigirá a la izquierda.....	64
Figura 34 Botón de derecha, el cual al presionar el drone se dirigirá a la derecha. ....	64
Figura 35 Botón de retroceder, el cual al presionar el drone retrocederá.....	65
Figura 36 Botón de eje izquierda, el cual al presionar el drone se moverá a su izquierda. ....	65
Figura 37 Botón de eje derecha, el cual al presionar el drone se moverá a su derecha.....	66
Figura 38 Botón de paro de emergencia, el cual al presionar el drone regresará a su posición de inicio.....	66
Figura 39 Pantalla para controlar con botones, esta constituida mediante varios botones embebido en un scroll view (pantalla deslizante).....	67
Figura 40 Red configurable del drone Bebop 2 para establecer conexión con el aplicativo.....	68

Figura 41 <i>Diagrama de unión de Objective C y Swift, esto es un proceso a seguir para unificar los lenguajes de programación de Apple si es requerido.</i> .....	69
Figura 42 <i>Mensaje cuando se declara un puente en Xcode.</i> .....	70
Figura 43 <i>Diagrama de flujo del inicio de la aplicación móvil.</i> .....	72
Figura 44 <i>Aparición de drone Bebop 2 al conectarse.</i> .....	72
Figura 45 <i>Diagrama de flujo de funcionamiento de gestos.</i> .....	74
Figura 46 <i>Parrot Bebop 2, drone utilizado para el proyecto de titulación.</i> .....	75
Figura 47 <i>Manos de usuario 1.</i> .....	78
Figura 48 <i>Manos de usuario 2.</i> .....	79
Figura 49 <i>Manos de usuario 3.</i> .....	80
Figura 50 <i>Manos de usuario 4.</i> .....	81
Figura 51 <i>Manos de usuario 5.</i> .....	82
Figura 52 <i>Ciudadela del Chofer, Santo Domingo de los Tsáchilas.</i> .....	83
Figura 53 <i>Territorio 1 visualizado desde Google Maps.</i> .....	84
Figura 54 <i>Territorio 2 visualizado desde Google Maps.</i> .....	84
Figura 55 <i>Territorio 3 visualizado desde Google Maps.</i> .....	85
Figura 56 <i>Despegue con el usuario 1.</i> .....	86
Figura 57 <i>Despegue con el usuario 2.</i> .....	87
Figura 58 <i>Aterrizaje con el usuario 3.</i> .....	88
Figura 59 <i>Despegue con el usuario 4.</i> .....	88
Figura 60 <i>Despegue con el usuario 5.</i> .....	89
Figura 61 <i>Hacia la derecha con el usuario 1.</i> .....	90
Figura 62 <i>Hacia la izquierda con el usuario 2.</i> .....	91
Figura 63 <i>Despegando con el usuario 3.</i> .....	91
Figura 64 <i>Despegando con el usuario 4.</i> .....	92
Figura 65 <i>Hacia la derecha con el usuario 5.</i> .....	92

Figura 66 <i>Despegue con el usuario 1</i> .....	93
Figura 67 <i>Hacia la izquierda con el usuario 2</i> . ....	94
Figura 68 <i>Aterrizaje con el usuario 3</i> .....	95
Figura 69 <i>Hacia la izquierda con el usuario 4</i> . ....	95
Figura 70 <i>Hacia la derecha con el usuario 5</i> . ....	96
Figura 71 <i>Hacia la derecha con el usuario 1</i> . ....	97
Figura 72 <i>Hacia la derecha con el usuario 2</i> . ....	97
Figura 73 <i>Despegue con el usuario 3</i> .....	98
Figura 74 <i>Despegue con el usuario 4</i> .....	98
Figura 75 <i>Hacia la derecha con el usuario 5</i> . ....	99
Figura 76 <i>Despegue con el usuario 1</i> .....	100
Figura 77 <i>Aterrizaje con el usuario 1</i> .....	100
Figura 78 <i>Despegue con el usuario 2</i> .....	101
Figura 79 <i>Despegue con el usuario 2</i> .....	101
Figura 80 <i>Sobre el eje con el usuario 5</i> . ....	102
Figura 81 <i>Despegue con el usuario 1</i> .....	103
Figura 82 <i>Hacia la derecha con el usuario 2</i> . ....	103
Figura 83 <i>Hacia la derecha con el usuario 3</i> . ....	104
Figura 84 <i>Despegue con el usuario 3</i> .....	104
Figura 85 <i>Despegue con el usuario 3</i> .....	105
Figura 86 <i>Despegue con el usuario 1</i> .....	106
Figura 87 <i>Despegue con el usuario 2</i> .....	107
Figura 88 <i>Despegue con el usuario 3</i> .....	107
Figura 89 <i>Despegue con el usuario 4</i> .....	108
Figura 90 <i>Despegue con el usuario 5</i> .....	108
Figura 91 <i>Intentando empezar el despegue</i> . ....	109

Figura 92 *Territorio #3 en la noche, donde no se puede apreciar nada del modelo.* ..... 111



## Resumen

El siguiente proyecto de investigación “Reconocimiento de gestos corporales basado en SVM e imágenes monoculares para el control de micro vehículos aéreos multirotor embebido en una aplicación móvil en tiempo real”, tiene como objetivo final la integración de los conocimientos adquiridos en la malla de la carrera de Ingeniería en Electrónica, Automatización y Control, cierta parte de programación orientada a objetos para la realización de una aplicación de móvil para smartphones la cual controle un micro vehículo aéreo multirotor (conociendolo hoy en día como drone) con gestos definidos, siendo ellos despegar, aterrizar, hacia delante, hacia atrás, derecha, izquierda y sobre el eje . El principal objetivo de este proyecto de investigación es permitir a cualquier persona, aun sin experiencia en el manejo de micro vehículos aéreos, controlar drones de manera intuitiva y natural. En la primera etapa del proyecto se realiza la extracción de características del cuerpo de una persona utilizando el sensor de la cámara de un smartphone. En su segunda etapa consiste en aplicar la tecnología de Custom Vision de Microsoft (aprendizaje de máquina supervisado con SVM) la cual se construye un modelo Core ML para el estudio inteligente de los gestos, se requiere de recolección de datos utilizando diferentes usuarios para generar el modelo de predicción. En su tercera etapa se centra en el desarrollo “front end”, la interfaz de usuario para la aplicación contempla la información necesaria, manejo adecuado y relevante para el control del usuario. En su cuarta y última etapa se realiza pruebas de funcionamiento para comparar el sistema de reconocimiento de gestos desarrollado con el algoritmo versus al método tradicional del drone de control de vuelo de una persona natural.

### Palabras claves:

- SVM
- CORE ML
- CUSTOM VISION

### **Abstract**

The following research project "Recognition of body gestures based on SVM and monocular images for the control of micro multirotor aerial vehicles embedded in a mobile application in real time", has as its final objective the integration of the knowledge acquired in the race mesh of Engineering in Electronics, Automation and Control, a certain part of object-oriented programming for the realization of a mobile application for smartphones which controls a micro multirotor aerial vehicle (known today as a drone) with defined gestures, being them take off, land, forward, backward, right, left and on the axis. The main objective of this research project is to allow anyone, even without experience in the handling of micro aerial vehicles, to control drones in an intuitive and natural way. In the first stage of the project, characteristics are extracted from a person's body using the camera sensor of a smartphone. In its second stage, it consists of applying Microsoft's Custom Vision technology (supervised machine learning with SVM), which builds a Core ML model for the intelligent study of gestures, requires data collection using different users to generate the prediction model. In its third stage, it focuses on "front end" development, the user interface for the application includes the necessary information, appropriate and relevant handling for user control. In its fourth and final stage, functional tests are carried out to compare the gesture recognition system developed with the algorithm versus the traditional method of the drone of flight control of a natural person.

#### **Keywords:**

- **SVM**
- **CORE ML**
- **CUSTOM VISION**

## Capítulo I

### Introducción

#### Antecedentes

Los sistemas de control de vuelo de vehículos aéreos no tripulados o sus siglas en inglés UAV han despertado un gran interés, debido al reto que supone conseguir una plataforma aérea totalmente autónoma (Aguilar, Casaliglla, & Pólit, Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles, 2017) (Aguilar, Álvarez, Grijalva, & Rojas, 2019) (Aguilar, Angulo, & Costa-Castello, Autonomous Navigation Control for Quadrotors in Trajectories Tracking, 2017), independientemente de los agentes externos, así permitiéndole ayudar a un pilotaje más sencillo y natural por parte del operario de drones. Algunas empresas han desarrollado vehículos aéreos con la capacidad de reconocimiento de gestos, pudiendo ser operado mediante movimiento de manos (Schlömer, 2008) (Aguilar, Cobeña, Rodríguez, Salcedo, & Collaguazo, 2018).

La teleoperación es la comunicación más fiable que puede consistir en una combinación de funciones autónomas (Grijalva & Aguilar, Landmark-Based Virtual Path Estimation for Assisted UAV FPV Tele-Operation with Augmented Reality, 2019). Simplemente un vehículo operado es la teleoperación, es utilizado para entornos difíciles de alcanzar, para reducir el costo de la misión y poder evitar la pérdida de vidas, abarcando un espectro más amplio de control manual al de supervisión (Schlömer, 2008). Gracias a esta tecnología la interacción hombre – robot o también nombrado con sus siglas en inglés HRI, ha evolucionado hoy en día.

La HRI ha desarrollado en las últimas décadas de acuerdo al avance de la tecnología, para ser mas intuitiva, utilizable y receptiva a las necesidades del usuario, fácil de usar y con interfaz de usuario óptima (Sanna, 2013) (Amaguaña, Collaguazo, Tituaña, & Aguilar, 2018).

Los dispositivos móviles pueden incorporar una cámara que permite grabación de video en alta definición, como HD o FullHD, combinado con los procesadores móviles de última generación, hace posible la ejecución de algoritmos basados en visión en tiempo real que mejoren la interacción del control con distintas aplicaciones móviles (Nieto, 2008) (Aguilar, Angulo, & Pardo, Motion intention optimization for multicopter robust video stabilization, 2017). Una de las aplicaciones es reconocimiento de gestos para distintas funcionalidades, en la literatura existen varias propuestas para el reconocimiento de gestos (Tovr, 2017).

El LEAP Motion es un controlador que reconoce gestos de las manos y los transmite a una estación ubicada en tierra, la cual se puede ejecutar en el Sistema Operativo de Robot (ROS) (Aguilar, Rodríguez, Álvarez, Sandoval, Quisaguano, & Limaico, Real-Time 3D Modeling with a RGB-D Camera and On-Board Processing, 2017) en Linux, esta plataforma se programa con el lenguaje de programación Python para poder interpretar los gestos y capturarlos con el LEAP (Sarkar, 2016). Mediante la cámara Kinect se puede extraer las características del cuerpo humano ya que esta cámara es de profundidad RGB – D está utilizando el algoritmo para la obtención de posiciones de varias articulaciones del cuerpo (Cobeña, 2018). Para la identificación de gestos del ser humano se implementa una máquina de vectores de soporte para el aprendizaje del control de gestos para estas aplicaciones que se desarrollan con drones (Cobeña, 2018). La empresa Nintendo desarrollo una consola denominada Wii la cual contiene una biblioteca desarrollada que explota datos de Wii-Sensor (sensor desarrollado por Nintendo para detectar movimiento) y emplea un modelo oculto de Markov para entrenar y reconocer gestos elegidos por el usuario (Schlömer, 2008). No obstante, hay varias aplicaciones de acuerdo a las anteriores propuestas que requieren una cantidad necesaria de sensores para su funcionalidad.

Una de las técnicas utilizadas para la esqueletización es el adelgazamiento homotópico ordenado por distancia o también llamado con sus siglas en inglés DOHT para esqueletizar

imágenes en 3D, DOHT produce esqueletos que son homotópicos y delgados. Esto se logra eliminando puntos secuencialmente en orden de distancia ascendente hasta que no se puedan eliminar de forma segura. Un punto se puede eliminar de forma segura solo si al hacerlo se conserva la topología (Wathen-Dunn, 1998).

El algoritmo de esqueletización y la red neuronal convolucional para crear el algoritmo de reconocimiento que reducen el impacto del ángulo de disparo y el entorno en el efecto de reconocimiento, y mejoran la precisión del reconocimiento de gestos en entornos complejos (Jiang, 2018). Los resultados esperados de este algoritmo muestran un 96.01% de factibilidad para la comparación con el método de máquina de vectores de soporte (SVM), el aprendizaje de diccionarios más la representación dispersa.

La necesidad de rendimiento y eficiencia energética en dispositivos móviles es evidente con el cambio de computación más intensiva a plataformas móviles (Future of mobile software for smartphones and drones: energy and performance, 2017). La eficiencia del rendimiento no está relacionada con la eficiencia energética ni la ineficiencia (Jiang, 2018). A medida que las plataformas móviles continúan evolucionando, los nuevos escenarios y casos de uso que involucran dispositivos móviles están en aumento (Future of mobile software for smartphones and drones: energy and performance, 2017).

Hay SpeakLiz Vision es otra aplicación móvil en la que se basa en reconocimiento de gestos para las personas con discapacidad visual. Para el diseño de la plataforma se uso algoritmos de inteligencia artificial que ejecutan los smartphone, con el fin de asistir a las personas con dicha discapacidad. Esta aplicación puede detectar objetos y mediante el altavoz del smartphone reproducirlo, reconoce los billetes avisándole al usuario que moneda se trata y el valor monetario, reconoce los colores y de la misma forma que los objetos

mediante el altavoz del smartphone lo reproduce, entre otras características de movimiento del ser humano (Retinosis, 2020).

Hay numerosos algoritmos que utilizan la cámara RGB que ahora es un lugar común en dispositivos móviles disponibles en el mercado. El algoritmo de In-Air Gestures Around Unmodified Mobile Devices reconoce de manera robusta una amplia gama de gestos en el aire, soportando la variación del usuario y las diferentes condiciones de iluminación (Song, 2014). Los teléfonos móviles "inteligentes" se han vuelto casi omnipresentes en la sociedad y están típicamente equipados con una cámara digital de alta resolución que se puede usar para producir una imagen de manera muy conveniente (Choodum, 2013).

### **Justificación e importancia**

La Administración Federal de Aviación o también nombrado por sus siglas en inglés FAA designa a los drones como cualquier aeronave volada por un piloto a través de un sistema de control en tierra o de manera autónoma mediante el uso de una computadora a bordo (Gregory, 2016). La mayoría de los drones no vienen con un conjunto de instrucciones sobre lo que son y no puedes hacer con ellos (DroneRush, 2013). En la Fuerza Aérea, necesitas ser un oficial para el pilotaje de drones y debe tener un título de cuatro años pasado por algún nivel de comisionamiento como academias o escuelas de capacitación de oficiales (Quora, 2012) (Jara-Olmedo A. , Medina-Pazmiño, Tozer, Aguilar, & Pardo, 2018).

La HRI por más de medio siglo se limitó al uso de teclado y mouse para su comunicación (Sanna, 2013). Lo que permite a este modelo muy restringido, como modelo de interacción para usuarios. Con la tecnología en constante crecimiento aparece varios sistemas de comunicación en HRI, entre ellos, la comunicación con gestos corporales mediante una

cámara o sensor. Los gestos son una habilidad que todos las personas pueden realizar sin ser necesario una gran capacidad de análisis y pensamiento por parte de dicha persona.

Las aplicaciones móviles hoy en día son un factor importante para la vida del ser humano ayudando en productividad, diversión, conocimiento, etc. La mayoría de seres humanos dispone de un dispositivo inteligente o también llamado smartphone. Los sistemas operativos más comerciales como iOS y Android en el mercado dan así una lucha constante de smartphones entre la compañía Apple y Google (Pascual, 2018).

El control de micro vehículos aéreos no tripulados o también llamado por sus siglas en inglés UAV, principalmente conocidos por drones debido a que han adquirido gran popularidad en la última década por la autonomía, y alto grado de aplicación en diversas actividades como: detección de objetos (Aguilar W. G., y otros, Pedestrian Detection for UAVs Using Cascade Classifiers and Saliency Maps, 2017), vigilancia (Aguilar W. G., y otros, 2017), recolección de datos (Basantes, y otros, 2018), detección de fuego (Pardo, Aguilar, & Toulkeridis, 2017), seguimiento (Aguilar W. G., Luna, Moya, Abad, Parra, & Ruiz, Pedestrian detection for UAVs using cascade classifiers with meanshift, 2017) y planificación de trayectorias (Aguilar & Morales, 3D Environment Mapping Using the Kinect V2 and Path Planning Based on RRT Algorithms, 2016) (Aguilar, Manosalvas, Guillén, & Collaguazo, 2018), fumigación, entre otras (DroneRush, 2013).

Los drones disponen de numerosas aplicaciones hoy en día, pero muchas personas por falta de experiencia o conocimiento no pueden ejecutar dichas funcionalidades al no estar familiarizadas con las interfaces de control que conlleva el drone y que muchas veces son complejas o difíciles de entender para el usuario (Zurita, Aguilar, & Enríquez, 2019). Las interfaces realizadas para el control no son tan óptimas y requieren de una investigación previa

para su funcionamiento (Liu, 2015). La FAA podría entrenar a los pilotos de una forma más rápida y este conocimiento sería utilizado para un manejo óptimo del micro vehículo aéreo en momentos de emergencia o necesidad rápida de pilotar.

El proyecto de investigación se desarrolla con el fin de ayudar a cualquiera persona a tener una aplicación móvil en su dispositivo inteligente o smartphone, sin experiencia ni conocimiento previo del manejo de micro vehículos aéreos multirrotor, controlar drones con una interfaz de usuario natural e intuitiva.

### **Alcance del proyecto**

El siguiente trabajo pertenece al proyecto de investigación “Sistemas inteligentes de monitoreo y aterrizaje para UAVs tácticos y mutirotores Fase 1 - SmartDrone1” y tiene como objetivo el reconocimiento de gestos corporales, para controlar los movimientos de un micro vehículo aéreo multirrotor los cuatro grados de libertad más comunes: arriba, abajo, adelante, atrás, derecha, izquierda, despegar y aterrizar. Se dividirá en tres etapas importantes siendo la última etapa donde se realizarán las pruebas necesarias para conseguir el objetivo.

En la primera etapa del proyecto, se realizará un estudio de las técnicas utilizadas para la detección de puntos de referencia del cuerpo humano. Posteriormente se desarrollará una interfaz para captura de datos mediante el sensor RGB de la cámara del smartphone a bordo que permita reconocer puntos de articulaciones y su respectiva posición en el espacio.

La segunda etapa del proyecto consiste en la clasificación de patrones e identificación de gestos corporales. Estos gestos identificados se usarán para el control del micro vehículo aéreo. Se desarrollará un algoritmo utilizando máquinas de vectores de soporte (SVM).

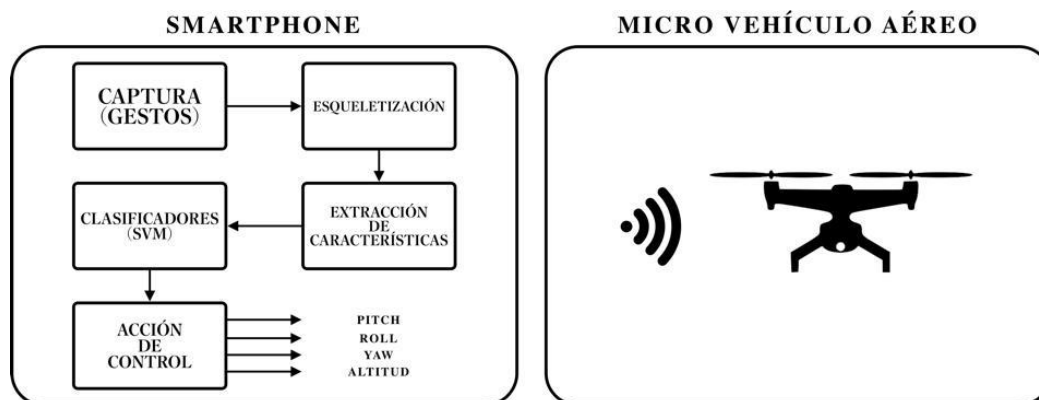


La última etapa del proyecto se enfoca en desarrollar aplicación móvil para iOS (Xcode) o Android (Android Studio) con una interfaz fácil e intuitiva de manipular para el usuario donde se conviertan las señales del SVM en movimientos pre programados, permitiendo el manejo adecuado en tiempo real.

Para el control del micro vehículo aéreo en la aplicación se implementará las librerías del drone Parrot Bebop 2 en el cuál se podrá reconocer gestos mediante un algoritmo de programación. La técnica para la detección de objetos se realizará con imágenes monoculares, proyectando así un algoritmo en el cuál se detecte los gestos que realizará el usuario. El procesamiento se llevará a cabo a bordo de un smartphone, la cuál recibirá toda información de las imágenes monoculares transmitidas desde el drone Bebop 2 con la cámara integrada del smartphone. Al finalizar las tres etapas anteriores, el proyecto estará orientado enfocará en la realización de pruebas de funcionamiento que busca analizar la precisión y velocidad de respuesta del sistema. En las pruebas se comparará el sistema de reconocimiento de gestos corporales versus los sistemas tradicionales para controlar un micro vehículo aéreo como por ejemplo el control remoto de fábrica.

Figura 1

Diagrama del proyecto de investigación.



## Objetivos

### Objetivo general

Desarrollar el reconocimiento de gestos corporales basado en SVM e imágenes monoculares para el control de micro vehículos aéreos multirotor embebido en una aplicación móvil en tiempo real.

### Objetivo específico

Implementar un estudio del arte sobre algoritmos de esqueletización, extracción de características y clasificadores para reconocimiento de gestos.

Desarrollar un sistema para detectar la posición de puntos de referencia del cuerpo humano basados en algoritmos de caracterización, descripción de puntos de interés que se extraerán de imágenes monoculares.

Determinar un algoritmo de extracción de características basado en reconocimiento de gestos para el control de un vehículo aéreo.

Clasificar gestos corporales definidos mediante el entrenamiento y aprendizaje de máquina de supervisado.

Construir y diseñar una interfaz de un smartpone para el usuario fácil de manipular e intuitiva para poder comunicar un micro vehículo aéreo en tiempo real mediante gestos.

Realizar prueba de funcionamiento del sistema de reconocimiento de gestos corporales embebido en la aplicación de un smartphone versus a un sistema de control remoto de vuelo, mediante pruebas basadas en el modelo construido.

## Capítulo II

### Estado del arte

#### Métodos de caracterización de gestos corporales

El término “gesto” ha sido definido en base a su función en la comunicación interpersonal, en este sentido es común encontrar definiciones sujetas a la función de expresar en comunicación como movimiento del rostro (Galindo, Aguilar, & Reyes Ch, 2019), de las manos o de otras partes del cuerpo con el que se expresan diversos afectos del ánimo (Grijalva, Cueva, Ramírez, & G, 2019). Sin embargo, el estudio constante del cuerpo en los procesos de significación de la experiencia humana, ha propiciado el surgimiento de múltiples definiciones del término “gesto” (Cadoz, 2010).

En un estudio previo que se comparó cuatro tipologías gestuales con el fin de estimar su pertinencia para la clasificación de los gestos observados en tareas de lectura cantada a primera vista, surgió la necesidad de redefinir al gesto y de elaborar una nueva categorización basada en las observaciones realizadas, debido a que las existentes no respondían completamente a las necesidades del estudio. (Cadoz, 2010).

Un método de caracterización del gesto corporal fue la patada de balón en movimiento en un partido de fútbol sala donde se estudió un tipo de caso. Se hicieron diez capturas del gesto deportivo, caracterizando la biomecánica de la técnica del deportista, evaluando desplazamiento articular, velocidad y centro de gravedad, los resultados se contrastaron con una evaluación fisioterapéutica en la que se incluyó postura, flexibilidad y fuerza muscular (Segarra, Caballeros, Aguilar, Samà, & Rodríguez-Martín, 2019). Al aplicar este método se pudo concluir que el deportista no realizaba de forma adecuada el gesto deportivo, porque no alcanza los arcos de movimiento esperados durante la fase de pateo con balón

quieto, lo que lleva a una disminución de la velocidad del gesto y de la potencia del pateo (Castro, 2013).

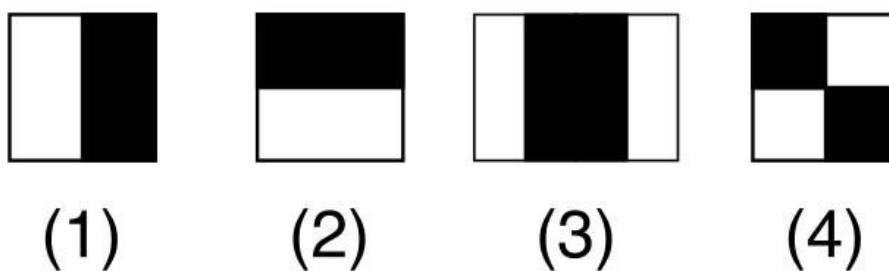
En el desarrollo un sistema con el método de caracterización del cuerpo humano, se utiliza un sensor Kinect, el cual se encarga de proveer las coordenadas en el espacio de las articulaciones del cuerpo humano gracias a la combinación de la cámara RGB y el sensor de profundidad (Cobeña, 2018) (Aguilar, Rodríguez, Álvarez, Sandoval, Quisaguano, & Limaico, Visual SLAM with a RGB-D Camera on a Quadrotor UAV Using on-Board Processing, 2017).

### Tipos de extracción de características

- **HAAR:** las características de HAAR utilizan escala de gris, apreciando los colores negros y blancos para la detección de los puntos de una imagen (Aguilar, Salcedo, Sandoval, & Cobeña, 2017) (Salcedo, Aguilar, Cobeña, Pardo, & Zahira, 2018).

### Figura 2

*Ejemplo de cuadros para extracción de características de HAAR.*



Nota: Tomado de (HAAR, 2019)

Usan el algoritmo de Viola-Jonas con la siguiente fórmula:

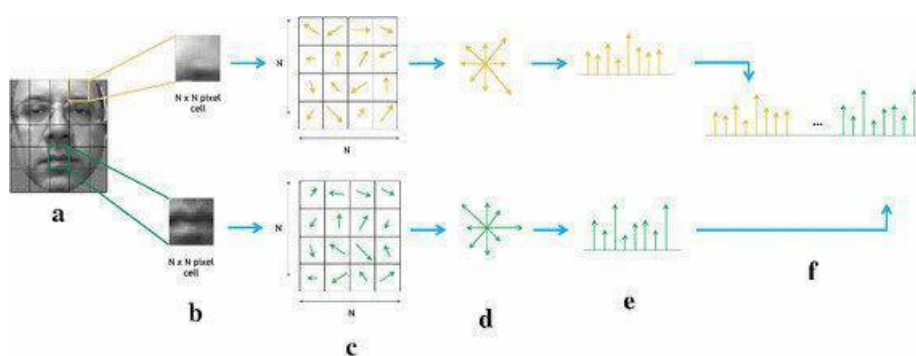
$$\frac{1}{\Sigma} \left( \frac{1}{\Sigma(\cdot)} \right)$$

La fórmula siguiente debería dar un resultado más cercano a 1. Cuanto más cerca esté el valor de 1, mayor será el cambio de detección de la característica HAAR en la imagen (Adakane, 2019).

- **HOC:** en el proceso de extracción de características la imagen se divide en celdas de tamaño  $N \times N$  píxeles. La orientación de todos los píxeles se calcula y se acumula en un histograma de orientaciones de  $M$ -bins. Finalmente, todos los histogramas de celda se concatenan para construir el vector de características finales. El ejemplo informa un tamaño de celda de 4 píxeles y 8 bandejas de orientación para los histogramas de celda (Gate, 2015).

**Figura 3**

*Diagrama de extracción de características de una imagen.*



Nota: Tomado de (ResearchGate, 2015)

- **LBP:** La función LBP (patrón binario local) tiene aplicaciones esenciales en el análisis facial y se ha aplicado en el reconocimiento de género en los últimos años. La función LBP normalmente adoptada encontrará una explosión de dimensión con el aumento de la densidad de muestreo del operador LBP, lo que no podría mejorar notablemente el rendimiento de la clasificación de género (Fang, 2008).

### **Tipos de clasificadores**

- **Redes Neuronales Artificiales:** Las redes neuronales artificiales (ANN por sus siglas en inglés) consideran la clasificación como una de las áreas de investigación y aplicación más dinámicas. ANN es la rama de la Inteligencia Artificial (IA). La red neuronal se entrenó mediante un algoritmo de retropropagación. Se estudian las diferentes combinaciones de funciones y su efecto al usar ANN como clasificador y se analiza la exactitud de estas funciones para varios tipos de conjuntos de datos (Saravanan, 2014) (Calderón, Aguilar, & Merizalde, 2020).
- **Redes Neuronales Convencionales:** Una de las formas más impresionantes de arquitectura ANN es la de la red neuronal convolucional (CNN por sus siglas en inglés). Las CNN se utilizan principalmente para resolver tareas difíciles de reconocimiento de patrones basadas en imágenes y, con su arquitectura precisa pero simple, ofrecen un método simplificado para comenzar con las ANN (O'Shea, 2015) (Aguilar, Quisaguano, Rodríguez, Alvarez, Limaico, & S, 2018) (Aguilar, Quisaguano, Alvarez, Pardo, & Zahira, 2018).

### **Métodos de clasificación para identificar los gestos corporales**

Para realizar un reconocimiento de gestos corporales es muy importante tomar en cuenta el método a utilizar, esto podría facilitar o complicar las funcionalidades con el hardware o con el modelo.

En base a (Rodrigo Ibañez - Álvaro Soria - Alfredo Teyseyre - Marcelo Campo, 2014) el Kinect identifica personas dentro del área de detección y calcula la posición en el espacio 3D

de 20 partes del cuerpo humano. Estas posiciones son recalculadas 30 veces por segundos y empaquetados en una estructura llamada "stick model". Cada stick model contiene la posición (X, Y, Z) de las 20 partes del cuerpo en un determinado momento. Al observar una secuencia de stick models sucesivos durante un intervalo de tiempo se obtienen los movimientos de las partes del cuerpo. Analizar estos movimientos permite a los desarrolladores reconocer gestos y crear un mecanismo de interacción natural entre humano-computadora, principalmente orientado en Machine Learning para el reconocimiento de gestos.

El método propuesto para la por (Franco Ronchetti, 2017) para clasificación de señas propone un esquema modular con subclasificadores parciales capaces de interpretar tres características principales en una seña: la posición, el movimiento y la configuración. Como sub-clasificador de configuración se utilizótambién una red tipo PromSOM para clasificar las 16 configuraciones del LSA. Este trabajo fue primero evaluado por separado para general descriptores apropiados que permitieron luego adicionar la información temporal de las diferentes configuraciones que puede tener una seña. Por último, como sub-clasificadores de las posiciones de las señas, se utilizaron distribuciones estadísticas con modelos gaussianos de las posiciones iniciales y finales que cada mano posee en una seña.

El sensor Kinect de Microsoft utiliza un procesamiento para poder detectar el esqueleto del cuerpo del ser humano realizando el método descrito por (Luis Villacres, 2018) donde el Kinect proyecta una serie de puntos, elabora un mapa de profundidad a partir de dichos puntos localizados. Tiene la facultad de lograr localizar el suelo (Aguilar, Rodríguez, Álvarez, Sandoval, Quisaguano, & Limaico, On-Board Visual SLAM on a UGV Using a RGB-D Camera, 2017) y separa los elementos del fondo para enfocarse en el entorno del cuerpo humano. Se encarga de localizar las partes para su posterior ordenamiento de partes del



cuerpo humano. El Kinect también reconoce articulación y permite simular el cuerpo de una persona utilizando el SDK de Microsoft.

Las SVM son un conjunto de algoritmos enfocados a problemas de aprendizaje, específicamente en clasificación y regresión. Estos métodos fueron desarrollados por Vladimir Vapnik y su equipo en la compañía AT&T, se presentaron en 1992 obteniendo unos resultados muy superiores al resto de opciones en aquella época, hecho que les hizo ganar una gran popularidad (JoséPérez, 2013).

Las SVM pertenecen a la rama de aprendizaje supervisado, es decir, requieren de datos de entrenamiento etiquetados para obtener un modelo capaz de predecir la etiqueta de una nueva muestra de datos no entrenada. Como en el caso de Adaboost, SVM es un clasificador binario y tal y como se explicaráposteriormente, se necesitarán herramientas adicionales para abordar el problema de la multi-clasificación (JoséPérez, 2013).

### **Aplicaciones de los micro vehículos aéreos multirotor**

En los últimos años, en el mercado de los UAV (Vehículos Aéreos Multirotor) o también llamado drone (Orbea, Moposita, Aguilar, Paredes, León, & Jara-Olmedo, Math Model of UAV Multi Rotor Prototype with Fixed Wing Aerodynamic Structure for a Flight Simulator, 2017), se ha incrementado exponencialmente en la fabricación (Orbea, Moposita, Aguilar, Paredes, Reyes, & Montoya, Vertical take off and landing with fixed rotor, 2017) y compra. Se espera que siga creciendo para implementar funcionalidades para el uso cotidiano. Los principales componentes de drone son su hardware y su software, este último controla la navegación o realiza diversas tareas según el campo de interés (Jara-Olmedo A. , Medina-Pazmiño, Mesías, Araujo-Villaroel, Aguilar, & Pardo, 2018) (Andrea, Byron, Jorge, Inti, & Aguilar, 2018) (Pardo, Aguilar, & Toulkeridis, 2017). La mayoría de fabricantes de UAV lanzan sus propias aplicaciones

móviles para el control de ellos y brindan funciones premium según la rama de funcionalidad del dron. Dado este factor se espera que en los próximos años la tienda de aplicaciones como el App Store o el Play Store estén llenas de este tipo de aplicaciones móviles para su uso (Narman Husnu, 2020).

Un estudio realizado para las aplicaciones muestra que 1,825 aplicaciones móviles UAV, en veinticinco categorías, con 162,250 comentarios. Hay 27 tipos de usuarios que se quejan de aplicaciones de drones. Las cuatro principales quejas observadas son error funcionalidad siendo así el 27.9%, compatibilidad del dispositivo el 16.8%, costo 16.2% y conexión/Sincronización 15.6% (Narman Husnu, 2020).

AnDrone combina un servicio en la nube con la primera arquitectura de virtualización de drones, permitiendo que el dron físico ejecute múltiples drones virtuales simultáneamente de manera aislada y segura a un bajo costo adicional. Algunos costos computacionales están dentro de un rango de precio bajo en comparación a costos operativos y de energía al poner el dron en el aire (Alexander Van't Hof, 2019).

AnDrone visualiza de una forma virtual una arquitectura de contenedor de Linux, siendo los contenedores virtuales de Android Things, proporcionando un entorno familiar para el usuario y el desarrollo que se ejecutará. La aplicación móvil estará desarrollada para el ambiente de Android y el hardware se implementó en base a una Raspberry Pi 3 (Alexander Van't Hof, 2019).

En otro estudio un trabajo se implementó un método de re direccionamiento de UAV (Vehículos Aéreos Multirrotor) o también llamado dron que puede cambiar la ruta preestablecida en una nueva ruta dibujando con el dedo en Google Maps. Esto se realiza en el

medio del vuelo del dron (Boyeon Kim - Munseong Jung - Yunseok Chang, 2018) (Merizalde, Aguilar, & Calderón, 2020). Para implementar dicho método de cambio de ruta, se realizó una aplicación de planificación de vuelo y se modifico el software integrado de control de vuelo de DJI para elaborar la aplicación personalizada para el trabajo (Boyeon Kim - Munseong Jung - Yunseok Chang, 2018) (Tenezaca, Canchignia, Aguilar, & Mendoza, 2019).

## Capítulo III

### Desarrollo del algoritmo

#### Introducción

Un modelo es el resultado de aplicar un algoritmo de aprendizaje automático a un conjunto de datos de entrenamiento. Utiliza un modelo para hacer predicciones basadas en nuevos datos de entrada. Los modelos pueden realizar una amplia variedad de tareas que serían difíciles o poco prácticas de escribir en código. Por ejemplo, puede entrenar un modelo para categorizar fotos o detectar objetos específicos dentro de una foto directamente desde sus píxeles (Apple Developer, 2020).

#### Figura 4

*Diagrama de Core ML para la integración del archivo Core ML model en un aplicativo de ios realizado con Swift.*



Nota: Tomada de (Apple Developer, 2020)

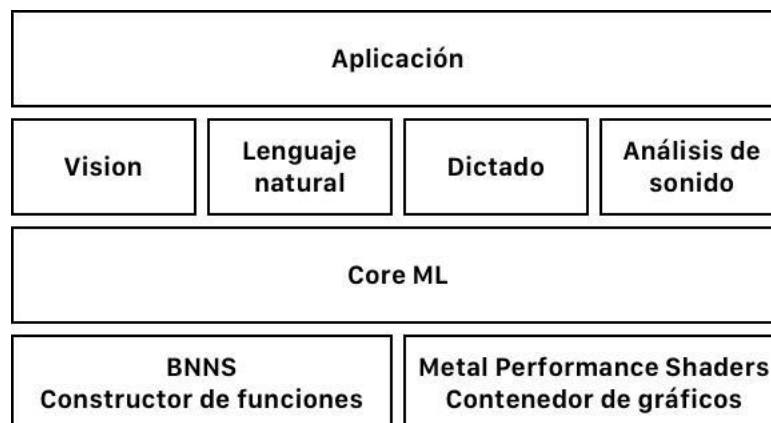
Se puede crear y entrenar un modelo con la aplicación “Create ML” incluida con Xcode. Los modelos entrenados con “Create ML” están en el formato de modelo Core ML y están listos para usar en su aplicación. Alternativamente, puede usar una amplia variedad de otras bibliotecas de aprendizaje automático y luego usar Core ML Tools para convertir el modelo al formato Core ML. Una vez que un modelo está en el dispositivo de un usuario, puede usar Core ML para volver a capacitarlo o ajustarlo en el dispositivo, con los datos de ese usuario (Apple Developer, 2020).

Core ML optimiza el rendimiento en el dispositivo al aprovechar la CPU, la GPU y el motor neuronal mientras minimiza su huella de memoria y el consumo de energía. Ejecutar un modelo estrictamente en el dispositivo del usuario elimina cualquier necesidad de una conexión de red, lo que ayuda a mantener la privacidad de los datos del usuario y su aplicación responde.

Core ML es la base de la funcionalidad y los marcos de trabajo específicos del dominio. Core ML admite Vision para analizar imágenes, Natural Language para procesar texto, Speech para convertir audio en texto y Sound Analysis para identificar sonidos en audio. El propio Core ML se basa en primitivas de bajo nivel como Accelerate y BNNS, así como en Metal Performance Shaders (Apple Developer, 2020).

### Figura 5

*Diagrama de flujo de Core ML.*



### Análisis de esqueletización y extracción de características

La API de Custom Vision permite entrenar fácilmente a un clasificador utilizando sus propios datos. En realidad, el modelo se ha construido sobre control de movimientos de un dron que pueden tener varias capas. Dado que el tema de este documento se refiere al

reconocimiento de señales o gestos de una mano, no había modelos previamente optimizados ya contruidos, a diferencia de otros temas de clasificación de imágenes. Por lo tanto, las tasas de éxito dependen en gran medida del conjunto de datos que se utilice. Un conjunto de datos debe tener una plétora de imágenes que varíen mucho en función del rayo, el brillo, el ángulo de la cámara, etc.

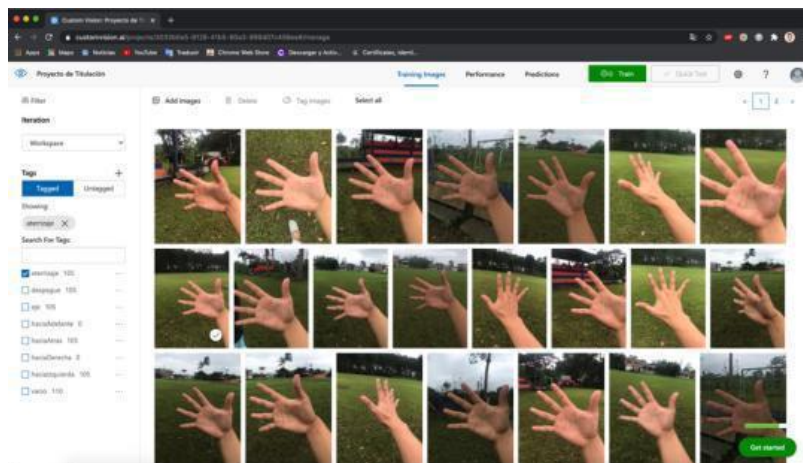
Está bastante claro que, para tal discrepancia entre las imágenes, las numerosas imágenes deben aplicarse en las fases de entrenamiento. Para tal fin, el conjunto de datos utilizado para el entrenamiento y las pruebas se diseñó fusionando ocho conjuntos de imágenes diferentes.

La razón principal para fusionar esos conjuntos de datos fue tener suficientes datos con características altamente distintivas que pudieran permitir que el modelo notara la discrepancia entre las señales de tráfico y la clasificación exitosa. Como se dijo anteriormente, las tasas de clasificación de éxito del modelo dependen de la variedad de imágenes en un conjunto de datos y su etiquetado.

Cuanto más divergentes sean las imágenes de los mismos gestos de movimiento, más fácil será encontrar características específicas que puedan separar esa señal de las demás. A continuación, se puede observar una captura de pantalla que representa una parte simple del conjunto de datos (imágenes para el movimiento de aterrizaje) con imágenes que representan el mismo gesto de para el aterrizaje en diferentes posiciones y algunas otras características:

**Figura 6**

*Imágenes para el movimiento de aterrizaje del dron en el dashboard de Microsoft Custom Vision, donde se agregarán todas las imágenes de acuerdo a los modelos creados.*



Nota: Tomado de (Microsoft Custom Vision, 2019)

## ANÁLISIS MATEMÁTICO DE SVM

Máquina de vectores de soporte o sus siglas en inglés SVM es un algoritmo de aprendizaje usado comúnmente para tareas de clasificación y regresión (SVM, 2019). Es una forma de aprendizaje supervisado que toma un conjunto de datos de entrenamiento con ejemplos que consisten elementos y clasificación para cada categoría asignada, construyendo un modelo clasificador (Alpaydin, 2009).

Se eligió SVM para compara el rendimiento con el modelo basado en los movimientos con el fin de averiguar qué tan bien uno de los algoritmos más comunes clasifica varias clases de señales de tráfico. También se utiliza a menudo para la clasificación de imágenes para las que SVM logra una alta precisión después de varias rondas de entrenamiento. Con el fin de comparar el rendimiento, se utilizaron imágenes de la primera iteración para construir un modelo y obtener las métricas.

El modelo se creó utilizando Python 3 y sus bibliotecas ampliamente utilizadas del conjunto de herramientas “Scikit-Learn”. Esas imágenes, 360 de ellas, se redimensionaron a la misma dimensión (3024px X 4032px) y luego se ajustó el modelo. Se aplicó el enfoque uno contra uno. Esto significa que se construyeron 105 imágenes para cada una de las 8 etiquetas y cada clasificador entrenó datos de los movimientos. El número de clasificadores se calcula mediante la siguiente fórmula:

$$\frac{n * (n - 1)}{2}$$

El ajuste se procesó en el conjunto de entrenamiento después de lo cual la fase de predicción estaba lista para comenzar. En esta fase, el clasificador deberá haber clasificado las imágenes del conjunto de datos de prueba.

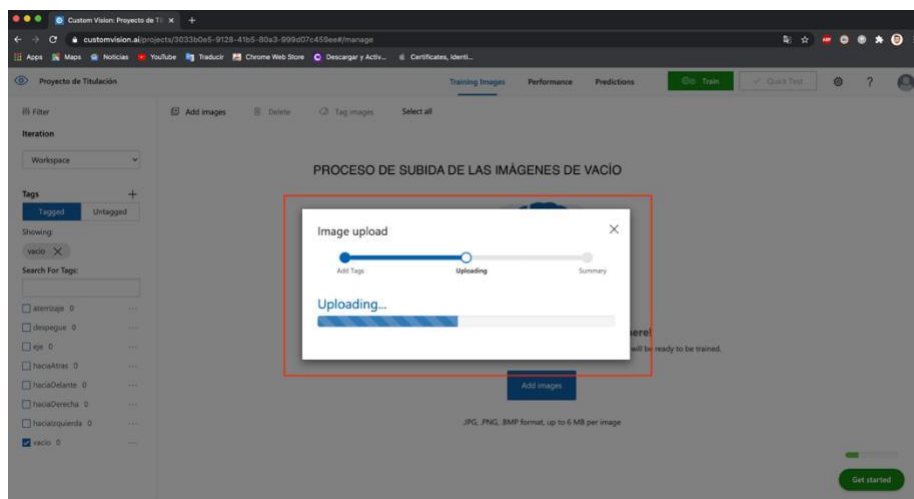
Se redimensionaron al mismo tamaño y se cambió el formato de imagen de HEIC a JPG ya que la herramienta de Microsoft no aceptaba este formato. Parecía que las imágenes muy pequeñas y las borrosas no se clasificaron correctamente porque en el conjunto de datos de entrenamiento de la primera iteración, había pocas imágenes borrosas o muy pequeñas.

Por lo tanto, en el conjunto de datos de entrenamiento para SVM deben agregarse imágenes adicionales con diferentes tipos de distinción entre imágenes. Sin embargo, para fines de comparación, el conjunto de datos aplicado había sido suficiente.



**Figura 7**

*Subida de imágenes a Custom Vision de cada combo a su respectivo modelo creado.*



Nota: Tomada de (Microsoft Custom Vision, 2019)

### **Análisis de imágenes monoculares y articulaciones**

Las imágenes se construyeron realizando 105 capturas por cada categoría, siendo las categorías las siguientes:

- Despegue
- Aterrizaje
- Hacia la derecha
- Hacia la izquierda
- Hacia adelante
- Hacia atrás
- Sobre el eje

Se siguió las directrices dadas por Microsoft al utilizar su herramienta de Azure llamada Custom Vision. Las imágenes fueron tomadas con un iPhone 7 y en una zona de un campo de fútbol.

Las articulaciones elegidas fueron las siguientes, siendo así el gesto de despegue es donde los dedos meñique, índice y pulgar estarán en posición recta (180 grados) respecto a la palma. En cuanto los dedos anular y medio esta recogidos hacia la palma de la mano.

### Figura 8

*Gesto para función despegue del Bebop 2.*



Nota: Tomada de (Vecteezy, 2019)

El gesto de aterrizaje es básicamente la palma de la mano abierta, los dedos meñiques, anular, medio, índice y pulgar en posición recta (180 grados) respecto a la palma.

### Figura 9

*Gesto para función aterrizaje del Bebop 2.*



Nota: Tomada de (Vecteezy, 2019)

El gesto el cual el drone Bebop 2 se desplazará hacia la derecha es donde los dedos meñiques, anular, medio e índice estarán recogidos hacia la palma y el dedo pulgar estará en posición recta vertical.

**Figura 10**

*Gesto para función hacia la derecha del Bebop 2.*



Nota: Tomada de (Vecteezy, 2019)

El gesto el cual el drone Bebop 2 se desplazará hacia la izquierda es donde los dedos meñiques, anular, medio y pulgar estarán recogidos hacia la palma y el dedo índice estará en posición recta horizontal.

**Figura 11**

*Gesto para función hacia la izquierda del Bebop 2.*



Nota: Tomada de (Vecteezy, 2019)

El gesto el cual el drone Bebop 2 se desplazará hacia adelante es donde todos los dedos estarán recogidos hacia la palma, básicamente formando un puño.

### Figura 12

*Gesto para función hacia adelante del Bebop 2.*



Nota: Tomada de (Vecteezy, 2019)

El gesto el cual el drone Bebop 2 se desplazará hacia atrás es donde los dedos meñiques, anular, medio e índice estarán en posición recta (180 grados) respecto a la palma y el dedo pulgar estará recogido hacia la palma.

### Figura 13

*Gesto para función hacia atrás del Bebop 2.*



Nota: Tomada de (Vecteezy, 2019)

El gesto el cual el drone Bebop 2 se moverá de acuerdo a su eje es donde los dedos medio e índice estarán en posición recta (180 grados) respecto a la palma y los dedos meñique, anular y pulgar estarán recogidos hacia la palma.

Figura 14

*Gesto para función sobre el eje del Bebop 2.*

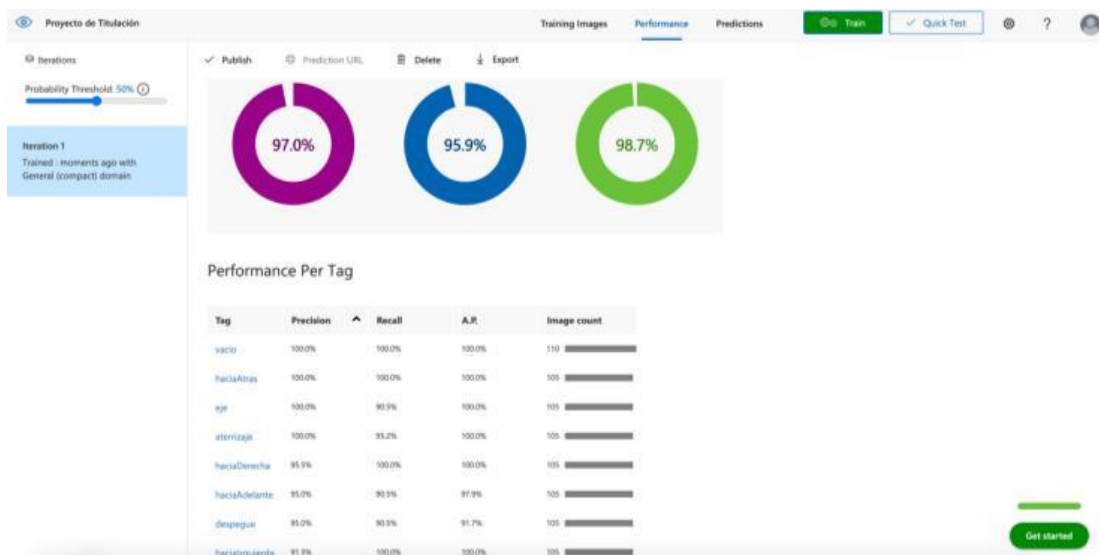


Nota: Tomada de (Vecteezy, 2019)

Una vez creado cargado las imágenes a la herramienta de Microsoft llamada Custom Vision, se podrá detectar la precisión con la que el modelo se construyó.

Figura 15

*Datos de precisión de Custom Vision, con estos datos se observa la precisión del modelo construido para su uso.*



Una vez finalizado el entrenamiento, se estima y se muestra el rendimiento del modelo. El servicio Custom Vision utiliza las imágenes que envió para la capacitación para calcular la precisión y la recuperación, mediante un proceso llamado validación cruzada de K-

Fold (algoritmo usado para la detección) (Microsoft Custom Vision, 2019). La precisión y la recuperación son dos medidas diferentes de la eficacia de un clasificador:

- La precisión indica la fracción de clasificaciones identificadas que eran correctas. Por ejemplo, si el modelo identifica 100 imágenes como perros, y 99 de ellas son en realidad de perros, entonces la precisión sería del 99%.
- Recordar indica la fracción de clasificaciones reales que se identificaron correctamente. Por ejemplo, si en realidad hubiera 100 imágenes de manzanas y el modelo identificara 80 como manzanas, el retiro del mercado sería del 80%.

El algoritmo se entrena con estos datos y calcula su propia precisión probándose a sí mismo en esas mismas imágenes. Una vez que el algoritmo está entrenado, se proseguirá a probarlo, volver a entrenarlo y eventualmente usarlo para clasificar nuevas imágenes de acuerdo con las necesidades de su aplicación.

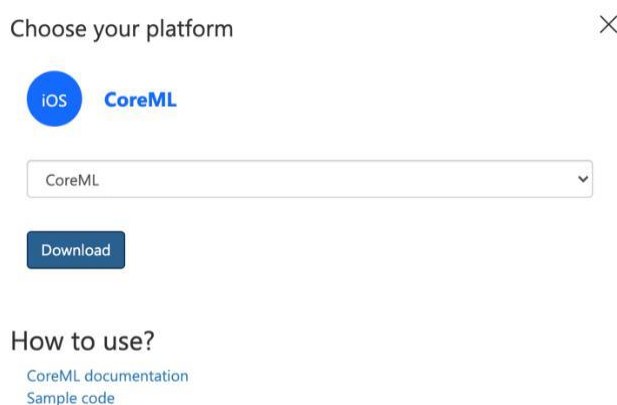
Se observa el control deslizante Umbral de probabilidad en el panel izquierdo de la pestaña Rendimiento. Este es el nivel de confianza que debe tener una predicción para que se considere correcta (a los efectos de calcular la precisión y la recuperación).

Cuando interpreta las llamadas de predicción con un umbral de alta probabilidad, tienden a devolver resultados con alta precisión a expensas de la recuperación, las clasificaciones detectadas son correctas, pero muchas permanecen sin ser detectadas. Un umbral de baja probabilidad hace lo contrario, detecta la mayoría de las clasificaciones reales, pero hay más falsos positivos dentro de ese conjunto (Microsoft Custom Vision, 2019).

Con esto en mente, debe establecer el umbral de probabilidad de acuerdo con las necesidades específicas de su proyecto. Más adelante, cuando reciba resultados de predicción en el lado del cliente, debe usar el mismo valor de umbral de probabilidad que usó aquí. El modelo será exportado en Core ML para su utilización con el código en Xcode 5 una vez analizado estos datos de la herramienta Custom Vision.

### Figura 16

*Exportación de modelo en Core ML. Desde este espacio se exporta el modelo para importarlo en el proyecto de Xcode.*

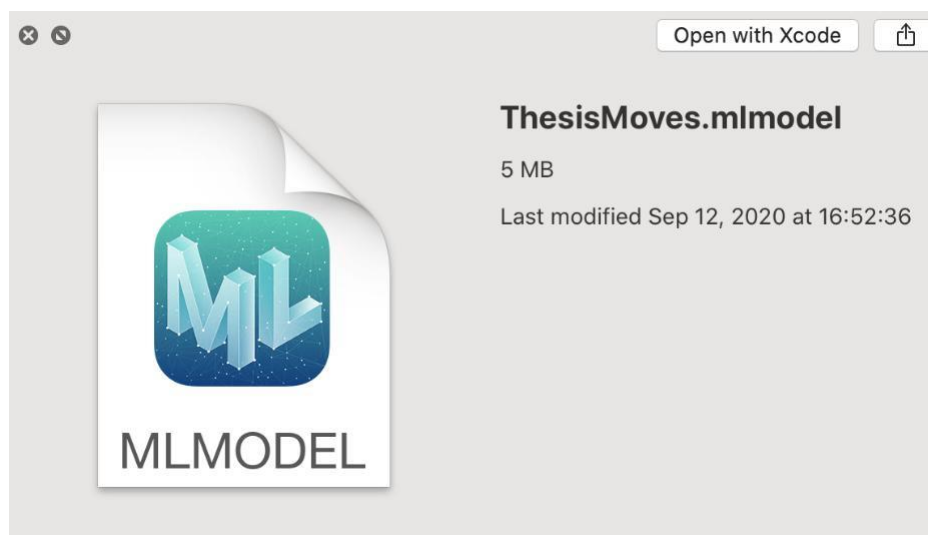


Nota: Tomada de (Microsoft Custom Vision, 2019)

Una vez descargado se tendrá un archivo del formato mlmodel el cual se usará en Xcode 5 para la construcción de la aplicación. Se cambio el nombre a *"ThesisMoves.mlmodel"*, ya que el nombre que descargaba por defecto era model mlmodel.

**Figura 17**

*Modelo de Core ML, el cual servirá para importar en el proyecto de Xcode y poder utilizarlo en el aplicativo. Este archivo es el modelo el cual ayuda a reconocer los gestos con la cámara.*



Nota: Tomada de (Microsoft Custom Vision, 2019)

### **Desarrollo de algoritmo para clasificación de gestos mediante SVM**

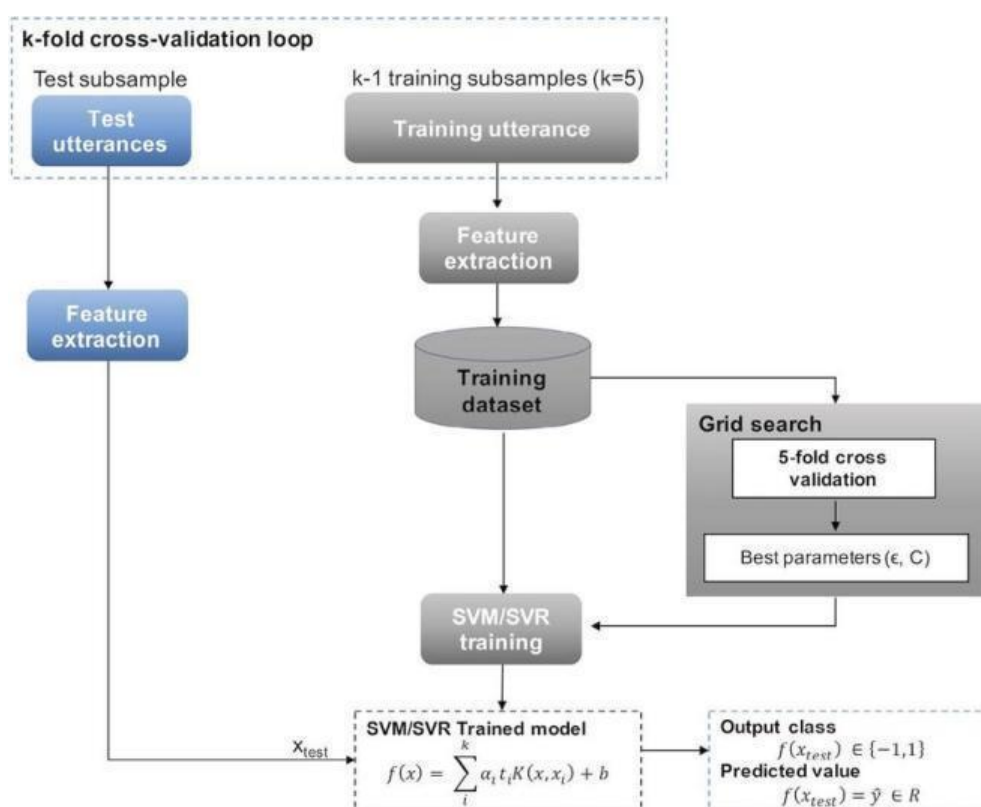
El algoritmo de validación cruzada K-Fold es una de las diversas técnicas de validación de modelos similares para evaluar cómo los resultados de una estadística el análisis se generalizará a un conjunto de datos independientes. Se utiliza principalmente en entornos en los que el objetivo es la predicción y se desea estimar la precisión con la que un modelo predictivo funcionará en la práctica. En un problema de predicción, un modelo generalmente recibe un conjunto de datos conocidos en los que se ejecuta el entrenamiento (conjunto de datos de entrenamiento) y un conjunto de datos de datos desconocidos (o datos vistos por primera vez) con los que se prueba el modelo (llamado conjunto de datos de validación o prueba conjunto) (Microsoft, 2019).



El objetivo de la validación cruzada es probar la capacidad del modelo para predecir nuevos datos que no se utilizaron en la estimación, con el fin de marcar problemas como sobreajuste o sesgo de selección y dar una idea de cómo el modelo se generalizará a un conjunto de datos independientes (es decir, un conjunto de datos desconocido, por ejemplo, de un problema real) (Microsoft, 2019).

**Figura 18**

*Diagrama de bloques del algoritmo de validación cruzada K-Fold que describe el proceso de búsqueda de cuadrícula y validación cruzada de k-fold para el modelo SVM / SVR.*



Nota: Tomada de (ResearchGate, Cross-Validation K-Fold, 2020)

Otro dato a tomar en cuenta del algoritmo es la validación cruzada típica, los conjuntos de entrenamiento y validación deben cruzarse en rondas sucesivas de modo que cada punto

de datos tenga la posibilidad de ser validado. La forma básica de validación cruzada es la validación cruzada de k veces por ello su nombre. Otras formas de validación cruzada son casos especiales de validación cruzada de k veces o implican rondas repetidas de validación cruzada de k veces (ResearchGate, Cross-Validation K-Fold, 2020).

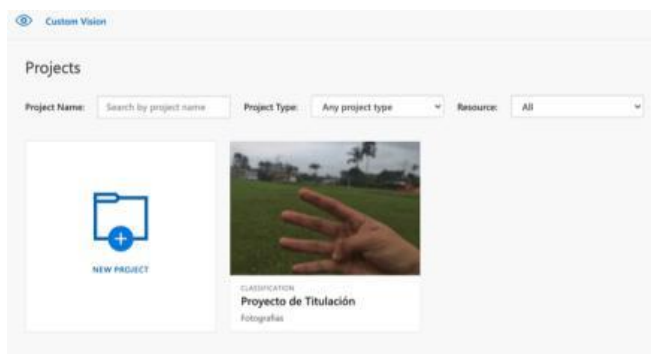
En los últimos años, el aprendizaje automático se ha convertido en una palabra de moda en la industria. Hay tantos campos que utilizan algún algoritmo o algún enfoque para facilitar el negocio y mejorar nuestra vida diaria. Uno de esos campos es la clasificación de imágenes que nos ayuda a detectar partes importantes de la imagen mediante redes neuronales convolucionales. Custom Vision ayuda a los ingenieros a construir un modelo que satisfaga mejor sus necesidades. Este modelo se puede utilizar para la clasificación de imágenes con cierta precisión (Matea Pejčinović,2019).

Al construir un modelo adecuado es muy importante la clasificación de imágenes, esto significa que el modelo garantizará una precisión de predicción satisfactoria al momento de cargarlo en la aplicación. Gracias a Azure Custom Vision de la empresa estadounidense Microsoft esta solución es básicamente un conjunto de varias API que ayudan a los desarrolladores e ingenieros a integrar funciones de inteligencia artificial dentro de su aplicación. Entre esas API también se encuentra una API de visión personalizada (Microsoft, 2019). Esta API esta basada en la nube de Azure que ayuda a los usuarios a crear un modelo que conducirá a la mejora de los clasificadores de imágenes personalizadas teniendo a favor un conjunto de datos de prueba que deben etiquetarse correctamente para representar clases o categorías.

El servicio Custom Vision está disponible como un conjunto de SDK nativos, así como a través de una interfaz basada en web (Azure Custom Vision, 2019). Antes de poder utilizar este servicio para la clasificación de imágenes, es necesario crear un modelo, en el proyecto se utilizó 8 categorías para incluir una serie de 105 imágenes por categoría de cada movimiento. Con esto se construirá y entrenará el modelo Core ML. Es posible construir un modelo para la clasificación de imágenes o para una tarea de detección de objetos. Las etiquetas se pueden aplicar en forma de una sola etiqueta o varias etiquetas por imagen. La tarea de clasificación podría referirse a varios dominios optimizados según el tipo específico de imágenes. Estos dominios son dominio genérico, alimentario, emblemático, minorista, adulto y compacto.

### Figura 19

*Proyecto de titulación en Custom Vision, el proyecto creado para este aplicativo.*



Nota: Tomada de (Microsoft Custom Vision, 2019)

Para la clasificación de los gestos se usó la herramienta de Microsoft Custom Vision en la plataforma Azure. Esta plataforma permite subir una serie de imágenes para construir el modelo que será exportado en Core ML para poder agregarle código y sea funcional en una aplicación móvil. Las imágenes están etiquetadas según el movimiento que se usará para el drone Bebop 2.

De acuerdo al número de imágenes que se obtenga por cada movimiento, la precisión subirá y será más óptimo al momento de controlar el drone Bebop 2 siendo lo ideal 50 imágenes por movimiento para una precisión funcional sin problemas. Para la creación de imágenes se replicó 105 veces cada imagen. Custom Vision le brinda un panel que muestra el rendimiento de su modelo entrenado, midiéndolo en predicciones y porcentajes de recuperación (Sayalee Pote, 2019).

## Capítulo IV

### Desarrollo de la aplicación móvil

#### Introducción

Para este capítulo se describirá como se construyó Gestdrone mediante el IDE de programación Xcode (programa que se usa para la realización de aplicaciones de móviles para la compañía Apple) y el lenguaje de programación Swift lanzado por Apple en 2014 (Apple, Swift Language Organization, 2014). La versión utilizada para el IDE fue Xcode 11.4 con la versión del lenguaje de programación Swift 5.0 (Apple, Swift Language Organization, 2014).

La aplicación es muy simple y cuenta cuatro pantallas, una principal en la cual el usuario introducirá su usuario y contraseña para ingresar a Gestdrone, la segunda pantalla aparecerá un título indicando si la conexión es exitosa mediante wifi o no con el micro vehículo aéreo multirrotor, ocho botones para poder manipularlo con las siguientes acciones despegue, aterrizaje, hacia delante (avance), hacia atrás (retroceder), derecha, izquierda, sobre el eje y un paro de emergencia para cualquier fallo pueda el micro vehículo aéreo multirrotor regresar a su lugar de partida. En la tercera pantalla aparecerá directamente una ventana con la que muestre la cámara, un botón de paro de emergencia. Mediante la cámara se realizará las acciones anteriormente nombradas con dichos gestos predeterminados para el trabajo de titulación. En la última pantalla se presenta información acerca de Gestdrone.

#### DESARROLLO DE INTERFAZ MÓVIL

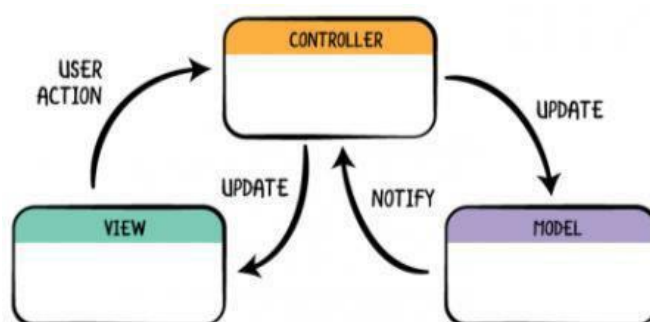
MVC (Model-View-Controller) es probablemente uno de los patrones de diseño de iOS más utilizados. Existen otros patrones de diseño en iOS como Delegation, Singleton y KVO entre otros.

MVC consiste en dividir el software de 3 capas distintas y separadas, donde cada capa tiene una única responsabilidad en la que se podrá programar cada vista con una clase para su funcionamiento.

El desarrollo de la interfaz móvil se construyó mediante la arquitectura MVC del lenguaje de programación Swift, el cual tiene el siguiente bloque de funcionalidad (Razeware, 2018).

**Figura 20**

*Bloque de funcionalidad de la arquitectura MVC.*



Nota: Tomada de (Razeware, 2018)

**Model:** Contienen datos de la aplicación y suelen ser estructuras o clases simples.

**View:** Muestran elementos visuales y controles en la pantalla.

**Controller:** Enlaza las clases codificadas con lo construido en una vista.

El funcionamiento interno de una aplicación iOS diseñada con MVC podría ser que actúa sobre la interfaz de la app. La View de la app comunicará al Controller que el usuario ha realizado la acción. El Controller ejecuta la lógica asociada a esa acción y toda la programación realizada (Efecto Apple, 2014).

Si esa lógica está asociada a modificaciones de los datos de la aplicación, el Controller actualiza el Model. Como el Controller ha modificado el modelo, es probable que haya que actualizar la View que muestra esos datos al usuario, por lo que el Controller actualiza también la View para mostrar los nuevos datos. También puede suceder el proceso contrario. Es decir, que los cambios comiencen por el Model (Efecto Apple, 2014).

A continuación, se describirán como se construyó el proyecto en Xcode 11.4 con el lenguaje de programación Swift 5.0 y las cuatro pantallas con su respectivo código:

Al abrir Xcode 11.4 se encontrará una pantalla parecida a esta con los proyectos recientes a su derecha, se creó un proyecto nuevo el cual se nombró Tesis y de nombre de aplicativo Gestdrone (dado por el autor a su conveniencia).

### Figura 21

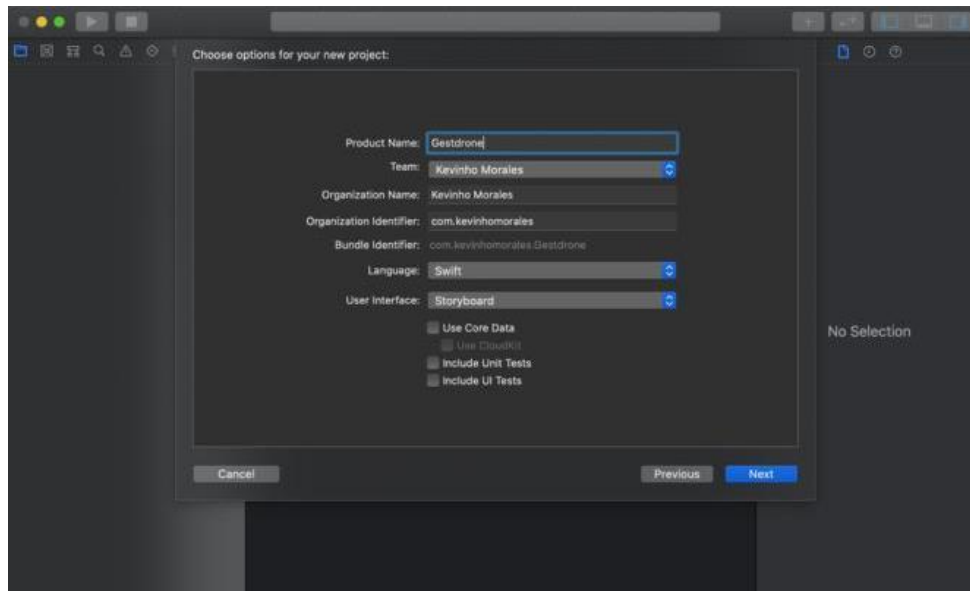
*Inicio del IDE Xcode, este es el inicio de un proyecto de una aplicación para iOS.*



Una vez creado el proyecto se seleccionó el lenguaje de programación Swift (el más actual para la empresa Apple Inc.) usando la interfaz de Storyboard donde construiremos el diseño estructural de la aplicación.

**Figura 22**

*Configuración inicial del proyecto en Xcode.*

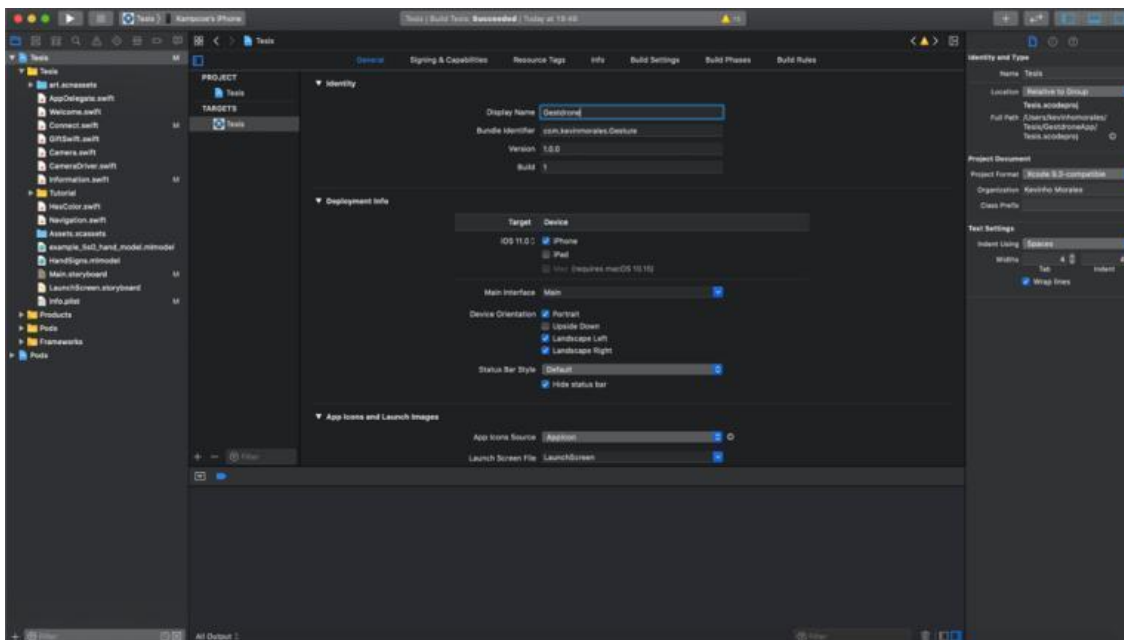


Cuando el proyecto de Xcode 11.4 esta abierto se mostró una pantalla similar a esta, donde a la izquierda podemos ver todas las clases que se construyó el algoritmo y diseño de la aplicación



Figura 23

*Pantalla inicial del proyecto en Xcode.*



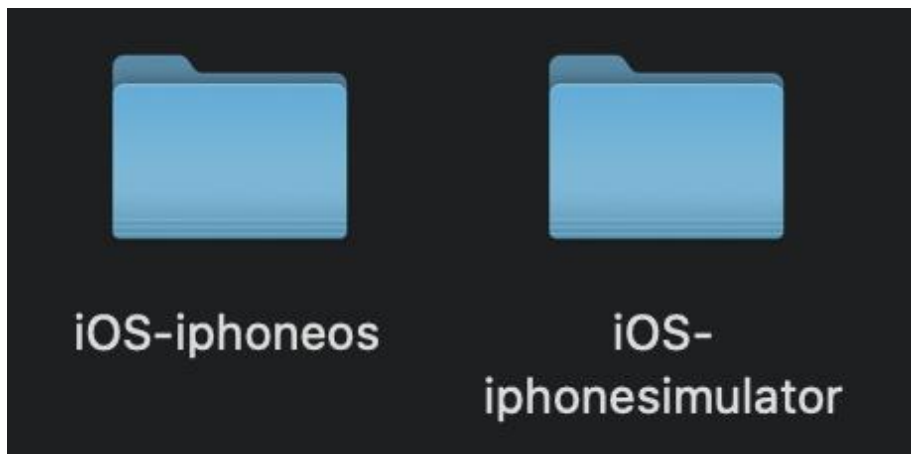
## CONFIGURACIÓN DE LIBRERÍAS PARA EL DESARROLLO

Para la configuración de las librerías del Parrot Bebop 2 se utilizó la documentación de la página web de desarrolladores de Parrot. Seguiremos los siguientes pasos para poder instalar las librerías correspondientes en el IDE de programación Xcode:

Primero se descargó la versión 3.9.2 del SDK (última versión al momento de este escrito) desarrollado por Parrot para acceder a diferentes funciones del Bebop 2. Se descarga un archivo en formato .ZIP el cual se descomprimió y para mostrar dos carpetas:

**Figura 24**

*Carpetas al descomprimir el archivo SDK de Parrot.*



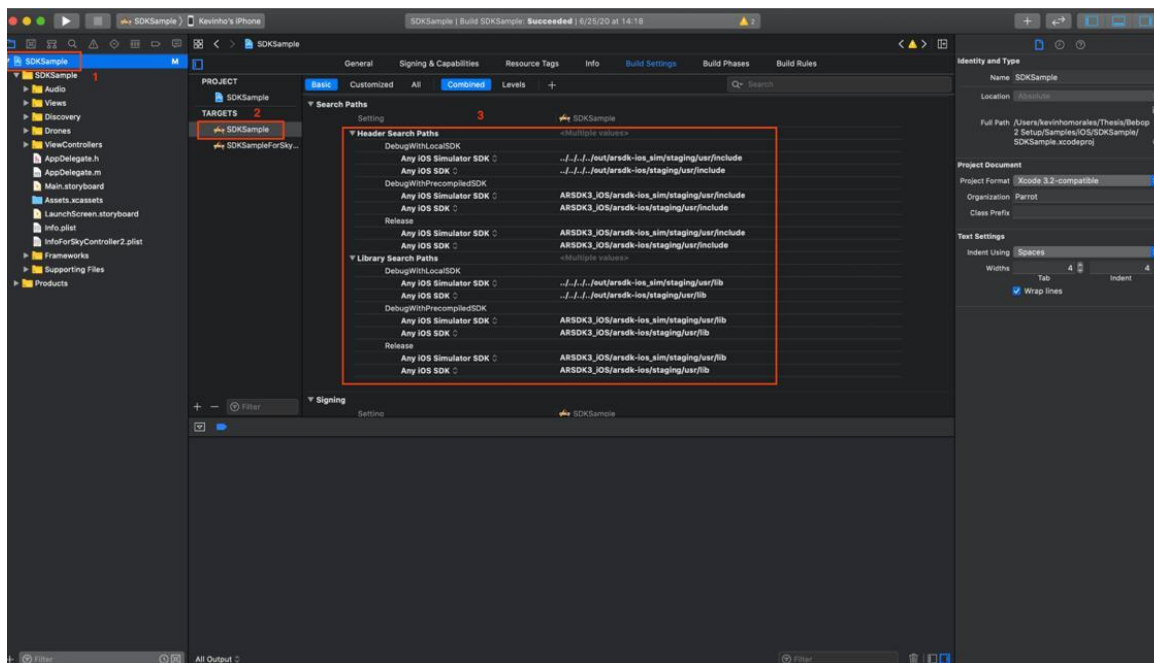
La primera carpeta **iOS-iphoneos** contiene librerías las cuales permitirán que funcione en un dispositivo físico. Con estas librerías se accederá a las acciones de cada función incluida para el desarrollo de una app para control del Bebop 2.

La segunda carpeta **iOS-iphonesimulator** contiene librerías las cuales permitirán que funcione para pruebas en un simulador de Xcode. Esto es muy importante ya que se podrá realizar pruebas al momento de dar acciones a las funciones de las librerías.

Para agregar las librerías en el IDE Xcode, se accederá a la pestaña “PROJECT” después en “TARGET” e ingresaremos a “BUILD SETTINGS” como se muestra en la siguiente imagen:

Figura 25

Vista de BUILD SETTINGS en Xcode.



En el punto 1, se visualiza el proyecto general. Después en el punto 2 ingresaremos al Targets y para finalizar en el punto 3 se puede visualizar donde se insertarán las librerías mediante una dirección de “PATH” al computador existente (Parrot, 2012).

Para el llamado de las librerías “INCLUDE” en el “HEADER SEARCH PATHS” o encabezado de búsqueda de caminos del archivo se llamó al path <Carpeta descomprimida de librerías>/iOS-iphonesimulator/staging/usr/include y para el iOS SDK se llamará al path <Carpeta descomprimida de librerías iOS-iphoneos/staging/usr/include.

Figura 26

Vista de *HEADER SEARCH PATHS* en Xcode.

▼ Header Search Paths		<Multiple values>
DebugWithLocalSDK		
Any iOS Simulator SDK	◇	../../../../../out/arsdk-ios_sim/staging/usr/include
Any iOS SDK	◇	../../../../../out/arsdk-ios/staging/usr/include
DebugWithPrecompiledSDK		
Any iOS Simulator SDK	◇	ARSDK3_iOS/arsdk-ios_sim/staging/usr/include
Any iOS SDK	◇	ARSDK3_iOS/arsdk-ios/staging/usr/include
Release		
Any iOS Simulator SDK	◇	ARSDK3_iOS/arsdk-ios_sim/staging/usr/include
Any iOS SDK	◇	ARSDK3_iOS/arsdk-ios/staging/usr/include

Para el llamado de las librerías “LIB” en el “LIBRARY SEARCH PATHS” o librerías de búsqueda de caminos del archivo se llamó al path <Carpeta descomprimida de librerías>/iOS-iphonesimulator/staging/usr/lib y para el iOS SDK se llamará al path <Carpeta descomprimida de librerías iOS-iphoneos/staging/usr/lib.

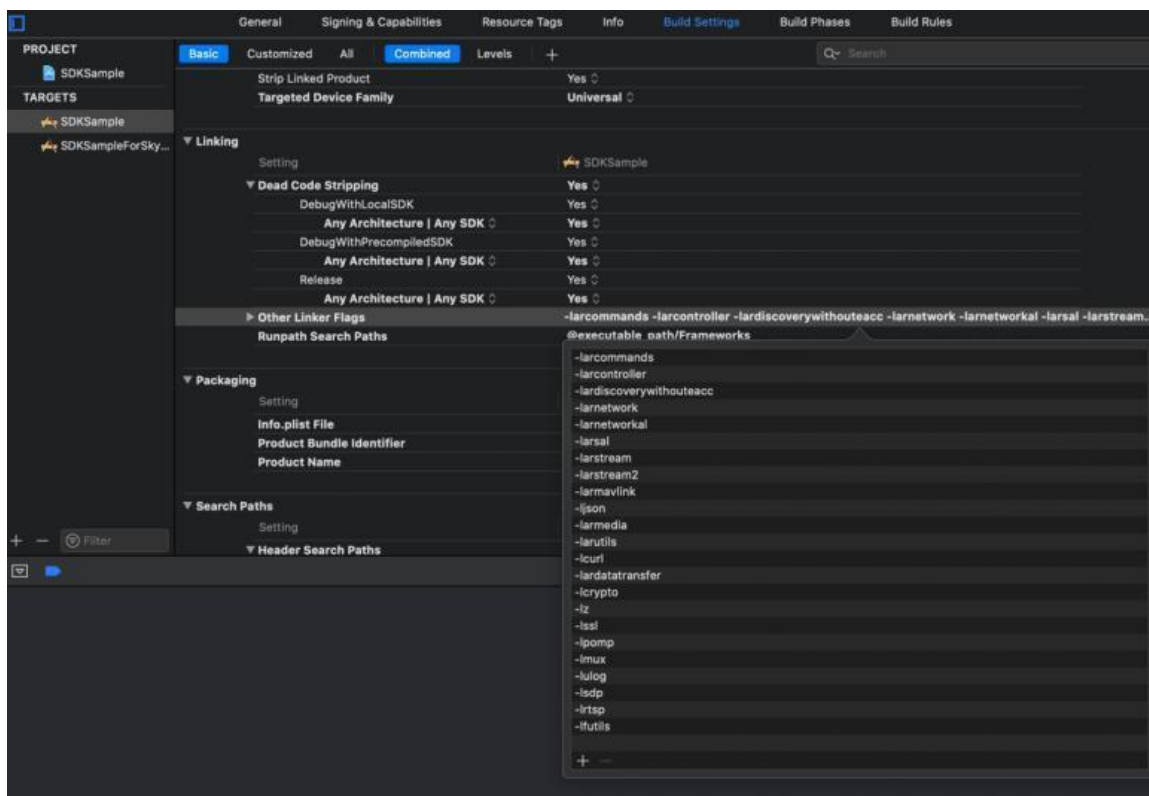
Figura 27

Vista de *LIBRARY SEARCH PATHS* en Xcode.

▼ Library Search Paths		<Multiple values>
DebugWithLocalSDK		
Any iOS Simulator SDK	◇	../../../../../out/arsdk-ios_sim/staging/usr/lib
Any iOS SDK	◇	../../../../../out/arsdk-ios/staging/usr/lib
DebugWithPrecompiledSDK		
Any iOS Simulator SDK	◇	ARSDK3_iOS/arsdk-ios_sim/staging/usr/lib
Any iOS SDK	◇	ARSDK3_iOS/arsdk-ios/staging/usr/lib
Release		
Any iOS Simulator SDK	◇	ARSDK3_iOS/arsdk-ios_sim/staging/usr/lib
Any iOS SDK	◇	ARSDK3_iOS/arsdk-ios/staging/usr/lib

Figura 28

Vista de *OTHER LINKER FLAGS* en Xcode.



En la sección de “OTHER LINKER FLAGS” se agrego toda esta lista para su funcionamiento: `-larcommands -larcontroller -lardiscovery -larnetwork -larnetworkal -larsal -larstream -larstream2 -larmavlink -ljson -larmedia -larutils -lcurl -lardatatransfer -lcrypto -lssl -lz`

Con las librerías configuradas ya estará listo para hacer una prueba y correr la aplicación en un dispositivo real o simulador el cual podremos tener acceso a todas las librerías (Parrot, 2012).

## Desarrollo de interfaz para captura de datos

El desarrollo de la interfaz de aplicación móvil se creó mediante la construcción de interfaz de usuario llamado STORYBOARD de Xcode.

La primera pantalla a mostrar es una pantalla de inicio de sesión o LOGIN el cual incluye un botón para continuar después inserte las credenciales, dos campos de texto en el cual se introducirá un correo electrónico y una contraseña por control del drone, idealmente se incorporará un LOGIN con redes sociales si es que la aplicación es aceptada en la App Store para ser más escalable. También incluye una imagen del logo con un título del nombre de la aplicación.

### Figura 29

*Vista inicial de la aplicación móvil, en este caso las credenciales son por conveniencia del autor.*



La segunda pantalla cuenta con una interfaz la cual muestra ocho botones, siete para control del drone Bebop 2 y un botón de tamaño más grande de color rojo para "PARO DE

EMERGENCIA". A continuación, se describirá los siete botones de control del dron. Se realizó este control para realizar pruebas con manipulación humana para después solo enviar las funciones a las acciones de cada gesto de la cámara.

El botón "AVANCE" realizará la acción de movimiento ascenso del dron Bebop 2, con este botón el dron se moverá hasta que el usuario deje de presionar el botón.

### Figura 30

*Botón de despegue, el cual al presionar el dron ascenderá.*



El botón "ATERRIZAJE" realizará la acción de movimiento descenso del dron Bebop 2, con este botón el dron se moverá hasta que el usuario deje de presionar el botón.

### Figura 31

*Botón de aterrizaje, el cual al presionar el dron descenderá.*



El botón "AVANCE" realizará la acción de movimiento hacia delante del dron Bebop 2, con este botón el dron se moverá hasta que el usuario deje de presionar el botón.

**Figura 32**

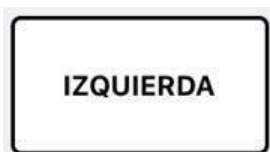
*Botón de avance, el cual al presionar el botón avanzará.*



El botón "IZQUIERDA" realizará la acción de movimiento hacia izquierda del drone Bebop 2, con este botón el drone se moverá hasta que el usuario deje de presionar el botón.

**Figura 33**

*Botón de izquierda, el cual al presionar el drone se dirigirá a la izquierda.*



El botón "DERECHA" realizará la acción de movimiento hacia la derecha del drone Bebop 2, con este botón el drone se moverá hasta que el usuario deje de presionar el botón.

**Figura 34**

*Botón de derecha, el cual al presionar el drone se dirigirá a la derecha.*

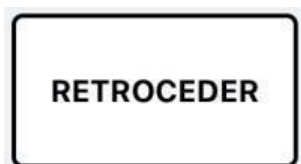




El botón “RETROCEDER” realizará la acción de movimiento hacia atrás del drone Bebop 2, con este botón el drone se moverá hasta que el usuario deje de presionar el botón.

**Figura 35**

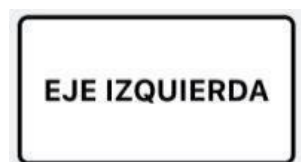
*Botón de retroceder, el cual al presionar el drone retrocederá.*



El botón “EJE IZQUIERDA” realizará la acción de movimiento sobre el eje (en este caso se utilizó para la izquierdo) del drone Bebop 2, con este botón el drone se moverá hasta que el usuario deje de presionar el botón.

**Figura 36**

*Botón de eje izquierda, el cual al presionar el drone se moverá a su izquierda.*



El botón “EJE DERECHA” realizará la acción de movimiento sobre el eje (en este caso se utilizó para la derecha) del drone Bebop 2, con este botón el drone se moverá hasta que el usuario deje de presionar el botón.

**Figura 37**

*Botón de eje derecha, el cual al presionar el drone se moverá a su derecha.*



Después de analizar los ocho botones de acciones del drone Bebop 2, se analizará el botón de "PARO DE EMERGENCIA" el cuál detectará que alguna acción errónea esta sucediendo con el drone y la acción principal que realizará es el de "BACK TO HOME" o también nombrado regreso a casa volviendo al lugar en donde partió su salida.

**Figura 38**

*Botón de paro de emergencia, el cual al presionar el drone regresará a su posición de inicio.*

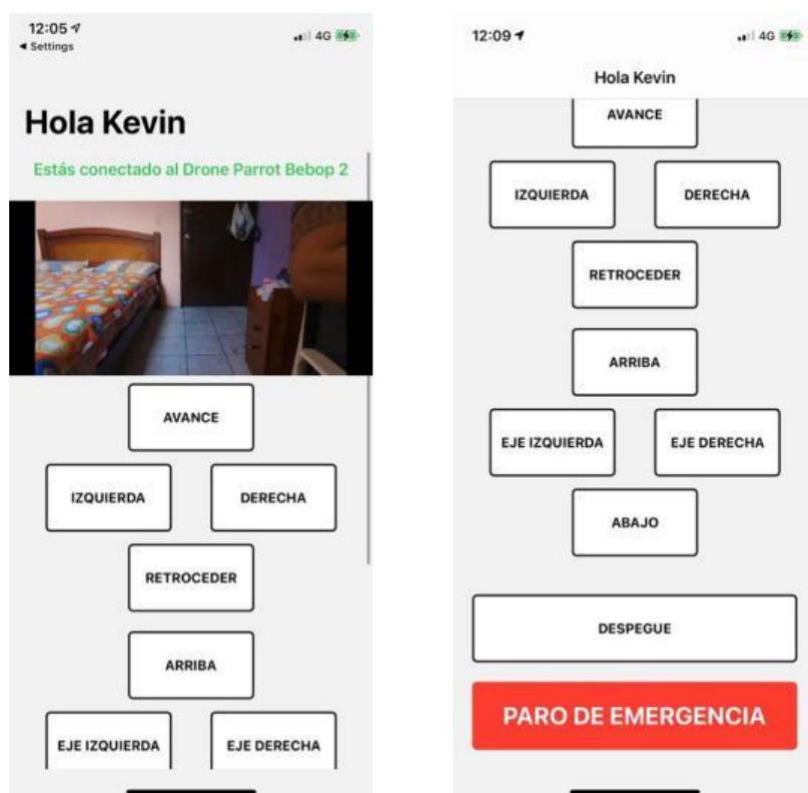


También consta un texto el cual indicará si el drone Bebop 2 tiene conexión con el celular o no, es un claro indicador antes de comenzar a manipular el drone. En la parte superior de la vista encontraremos accesos a información sobre la aplicación Gestdrone y la cámara para poder controlar el drone Bebop 2 con gestos y cumplir con el propósito de la tesis.

A continuación, vemos una captura de toda la interfaz de los botones y los textos que aparecen en la pantalla principal.

**Figura 39**

*Pantalla para controlar con botones, esta constituida mediante varios botones embebido en un scroll view (pantalla deslizable).*



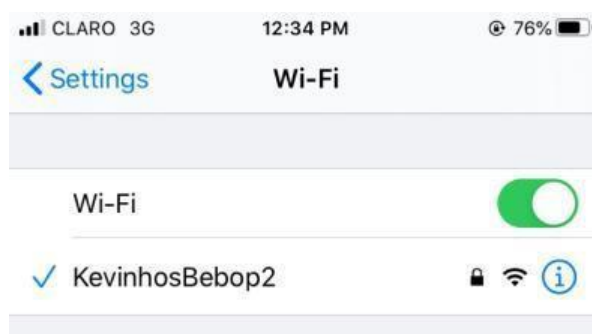
### Comunicación de la interfaz y control del micro vehículo aéreo

El desarrollo de la conexión con la interfaz móvil se centra en la conexión Wifi, ya que el drone dispone de una tarjeta Wifi la cuál mediante las librerías del Bebop 2 podemos acceder.

Se tuvo que resetear la el nombre de la red Wifi del drone realizando unos simples pasos. Una vez reseteado, se podrá configura el nombre de la red mediante la aplicación de Parrot dedicada a este drone en la cual dispone de una vista de ajustes teniendo como resultado el nombre nuevo de la red.

#### Figura 40

*Red configurable del drone Bebop 2 para establecer conexión con el aplicativo.*



El SDK lo ayudará a conectarse, pilotar, recibir transmisiones, guardar y descargar medios (fotos y videos), enviar y reproducir planes de vuelo de piloto automático y actualizar su dron. Es imprescindible para usar el drone Bebop 2.

Este SDK está escrito principalmente en Objective C (código oficial de Apple durante el 2010 hasta el 2014), proporciona bibliotecas para el sistema Unix, Android e iOS. Al momento que se creo la aplicación en iOS fue creada en el lenguaje Swift 5, entonces se tuvo que elaborar un llamado bridge (puente) para la conexión entre Objective C y Swift.

Se puede usar archivos Objective C y Swift juntos en un solo proyecto, sin importar qué idioma usó originalmente el proyecto. Esto hace que la creación de objetivos de marco y

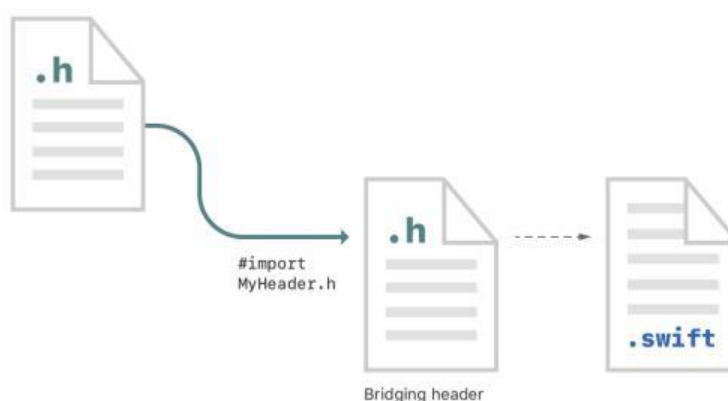
aplicaciones en varios idiomas sea tan sencilla como crear un objetivo de marco o aplicación escrito en un solo idioma (Apple Developer Program, 2019).

Para referirnos a este lenguaje en una breve introducción, una clase Objective C es una estructura C. Una instancia es una referencia a una estructura de este tipo que se ha asignado en la memoria. Una clase tiene un nombre y una instancia tiene un estado o valor (Drawnonward, 2019).

Lo que distingue a una clase Objective C de una estructura C es la capacidad de buscar direcciones de métodos por nombre. En términos simplificados, la estructura tiene una tabla hash de punteros de función codificados por nombre (Drawnonward, 2019).

#### Figura 41

*Diagrama de unión de Objective C y Swift, esto es un proceso a seguir para unificar los lenguajes de programación de Apple si es requerido.*



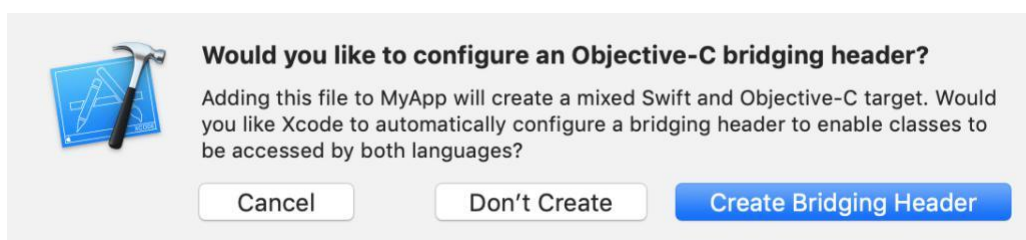
Nota: Tomada de (Apple Developer, 2020)

El proceso para usar sus declaraciones de Objective C de su código Swift en destinos de idiomas mixtos difiere levemente dependiendo de si está escribiendo una aplicación o un marco. Ambos procesos se describen a continuación.

Para importar un conjunto de archivos Objective C en código Swift dentro del mismo objetivo de la aplicación, confía en un archivo de encabezado de puente Objective C para exponer esos archivos a Swift. Xcode ofrece crear este encabezado cuando agrega un archivo Swift a una aplicación Objective C existente, o un archivo Objective C a una aplicación Swift existente.

#### Figura 42

*Mensaje cuando se declara un puente en Xcode.*



Xcode crea el archivo de encabezado de puente o llamado bridging header en inglés, junto con el archivo que estaba creando y lo nombra usando el nombre de su módulo de producto seguido de "-Bridging-Header.h". Alternativamente, puede crear un encabezado puente usted mismo eligiendo Archivo> Nuevo> Archivo> [sistema operativo]> Fuente> Archivo de encabezado. Al editar el encabezado puente para exponer su código Objective C a su código Swift. En su encabezado de puente de Objective C, importe cada encabezado de Objective C que desee exponer a Swift.

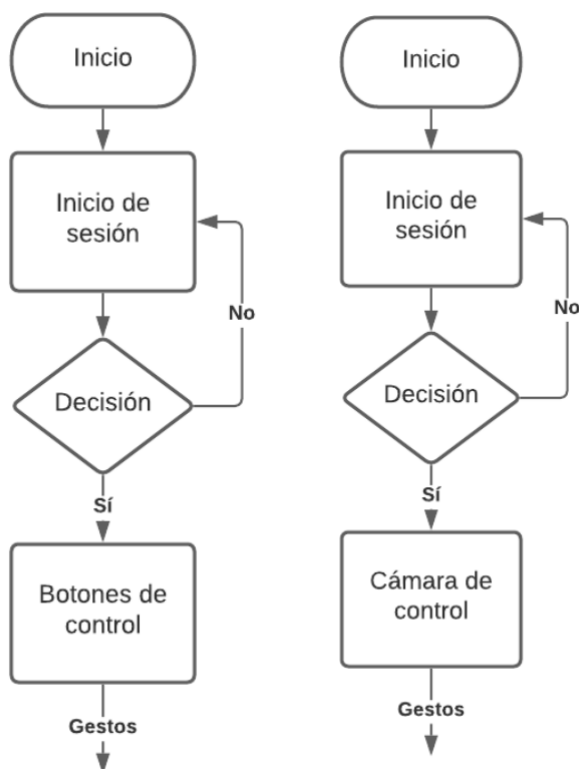
En Build Settings, en Swift Compiler - General, tomando en cuenta que la configuración de compilación del encabezado de puente de Objective C tenga una ruta al archivo de encabezado de puente. La ruta debe ser relativa al proyecto, similar a la forma en que se especifica su ruta Info.plist en Build Settings. En la mayoría de los casos, no necesitará modificar esta configuración (Apple Developer Program, 2019).

Cualquier encabezado público de Objective C enumerado en el encabezado puente son visibles para Swift. Las declaraciones de Objective C están disponibles automáticamente desde cualquier archivo Swift dentro de ese objetivo, sin declaraciones de importación. Use clases y otras declaraciones de su código Objective C personalizado con la misma sintaxis Swift que usa para las clases del sistema (Apple Developer Program, 2019).

Para el desarrollo de la aplicación se obtuvo los diagramas de flujo siguientes, donde el primero se enfoca en el desarrollo de la primera vista y en la cual se inicia sesión. En este caso el nombre de usuario es "Kevin" y la credencial es "12345". Se muestran dos diagramas, el primero se realizó con direccionamiento a botones para las pruebas internas antes de transformar esas acciones a gestos. El segundo flujo si es para la ejecución con los gestos.

Figura 43

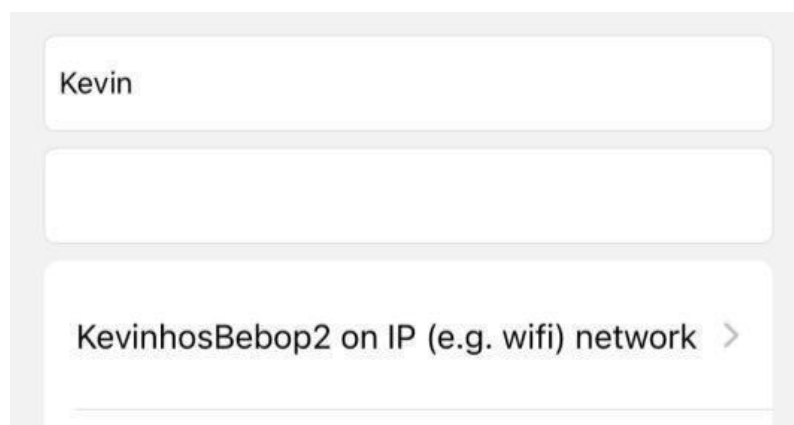
Diagrama de flujo del inicio de la aplicación móvil.



Automáticamente una vez se conecte el drone con el wifi del smartphone, aparecerá el nombre del drone Bebop 2 para ingresar.

Figura 44

Aparición de drone Bebop 2 al conectarse.



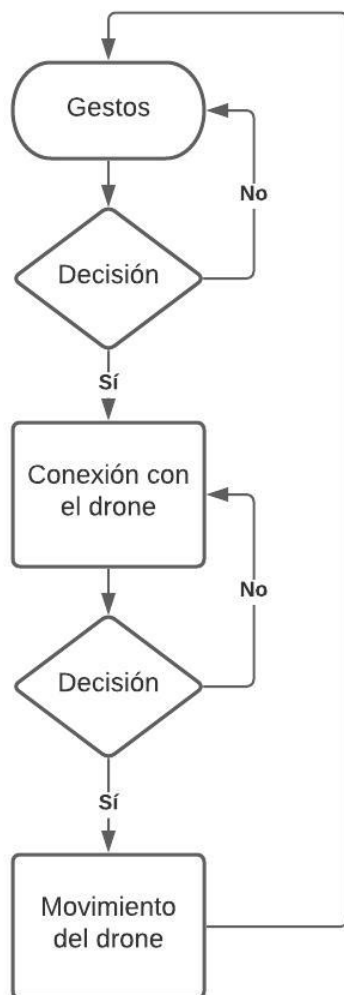


El diagrama de flujo de los gestos con los movimientos se presenta a continuación, en este se puede observar los gestos inician con una validación la cual reconoce o no, si esto es de caso exitoso se conecta con el drone Bebop 2 enviando información, de nuevo se instancia una validación en la que si la conexión es exitosa al reconocimiento enviará la acción para que el drone ejecute el movimiento.

Esto se repite constantemente en loop cuando se cambia de gesto por el usuario. La acción de los gestos se va enviando de acuerdo a las características que el modelo vaya captando, es decir, con el framework de Custom Vision detecta de una forma óptima el gesto y capta la acción para traducirla en funcionalidad.

Figura 45

Diagrama de flujo de funcionamiento de gestos.



El drone utilizado para este proyecto es el Bebop 2 de la empresa francesa Parrot S.A, siendo un drone controlado por teléfono inteligente con una cámara de 14 megapíxeles que ofrece grabación de video de 1080p. El sucesor del modelo Parrot Bebop original, cuenta con una lente ojo de pez que permite una grabación sorprendentemente libre de vibraciones e incluso cuenta con rotores reparables por el usuario (Techradar, 2017). Es ideal para el desarrollo y para implementar las librerías de la empresa y poder construir diferentes aplicaciones móviles con el.

**Figura 46**

*Parrot Bebop 2, drone utilizado para el proyecto de titulación.*



El Parrot Bebop 2 Drone ofrece un rendimiento de vuelo superior con un pilotaje fácil e intuitivo desde tu smartphone. Ligera y compacta, la tecnología de estabilización digital (Aguilar & Angulo, Real-Time Model-Based Video Stabilization for Microaerial Vehicles, 2016) (Aguilar & Angulo, Real-time video stabilization without phantom movements for micro aerial vehicles, 2014) de tres ejes permite imágenes aéreas estables y claras para que pueda volar con confianza en todas las condiciones (Apple, Accesorios, 2020).

Un potente LED trasero intermitente proporciona una visibilidad mejorada, mientras que una mayor duración de la batería significa que puede disfrutar de hasta 25 minutos de vuelo. La aplicación de control FreeFlight 3 desarrollada por Parrot SA es fácil de usar para acceder a todas las funciones esenciales, como pilotaje, fotos / videos, plan de vuelo, configuración de velocidad y altitud (Apple, Accesorios, 2020).

## Capítulo V

### Pruebas y resultados

#### Pruebas

Al momento de realizar pruebas el objetivo de este proyecto fue presentar una descripción general de la herramienta de Microsoft Custom Vision y como poder realizar un modelo para la aplicación. Custom Vision ayuda mucho a los usuarios en el desarrollo de clasificadores de imágenes cuando se trata de ciertas tareas para las que no tenemos un modelo de clasificación ya construido.

Una vez se realizaron las pruebas de entrenamiento, es necesario probar el modelo construido en un conjunto de datos de prueba. Esto significa que es necesario probar el modelo en imágenes que este modelo no había visto antes, pero que se espera que se clasifique según el conocimiento previo. Los gestos para los movimientos del drone Bebop 2 debe encontrar las características de las imágenes de prueba y compararlas con el conocimiento adquirido en el conjunto de datos de entrenamiento. El modelo basado en Custom Vision realiza una evaluación de una o varias imágenes.

Utilizando la primera iteración del clasificador se demostró que los gestos para los movimientos relacionados con la etiqueta de dirección no se clasificaron con gran precisión. Las imágenes que mostraban una prohibición fueron clasificadas con un alto nivel de precisión, esto se hizo con 50 imágenes mínimo, que son las que Custom Vision recomienda para una funcionalidad óptima.

Dado que el modelo estaba sesgado, en la segunda ronda de construcción del modelo, se agregaron más imágenes a las etiquetas que carecían de datos hasta llegar a 105 por cada etiqueta. Después de eso, se inició una tarea de predicción de prueba y se demostró que las imágenes, donde hay un mano al frente, generalmente se clasifican con una tasa de probabilidad de al menos 84%.

Se realizó pruebas con cinco personas de edades y contexturas diferentes, la primera persona es Kelly Morales con una edad de 12 años (usuario 1), la segunda persona es Verónica Bustillos con una edad de 46 años (usuario 2), la tercera persona es Fredy Morales con una edad de 48 años (usuario 3), la cuarta persona es Jimena Jácome con una edad de 24 años (usuario 4) y la quinta persona es Kevin Morales con una edad de 26 años (usuario 5).

A continuación, se muestra dos fotografías de las manos de cada persona que participo en las pruebas, una fotografía muestra la parte de enfrente y la otra la parte trasera de cada mano. Las manos utilizadas para estas pruebas son la mano derecha ya que el modelo de Custom Vision fue construido todo con mano derecha. Las manos tendrán una breve descripción para saber el tipo de persona con las que se esta desarrollando las pruebas.

**Figura 47**

*Manos de usuario 1.*



La mano del usuario 1 es fina y cuenta con dedos delgados la cual la hace larga, tiene poca visibilidad de las venas ya que es una mano de una niña de 12 años de edad.

**Figura 48**

*Manos de usuario 2.*



La mano del usuario 2 es una mano con apariencia normal en la que se detecta los huesos rápidamente sin ningún movimiento.

**Figura 49**

*Manos de usuario 3.*



La mano del usuario 3 es una mano que se brota varias venas y larga, esto quiere decir que los dedos son delgados. Sin ningún movimiento se puede observar las venas.



**Figura 50**

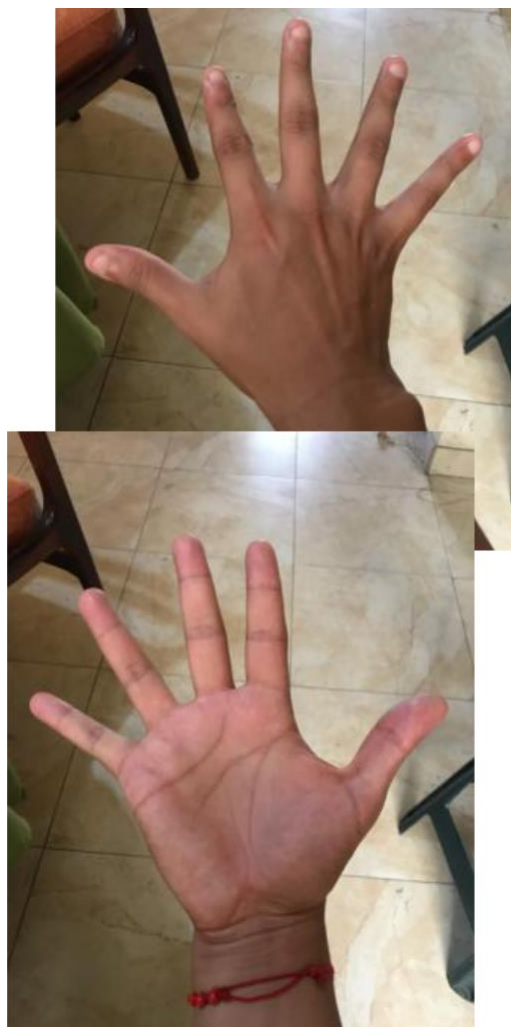
*Manos de usuario 4.*



La mano del usuario 4 continúa una anchura y una longitud de los dedos normal, los huesos y las venas no se le notan a primera vista.

**Figura 51**

*Manos de usuario 5.*



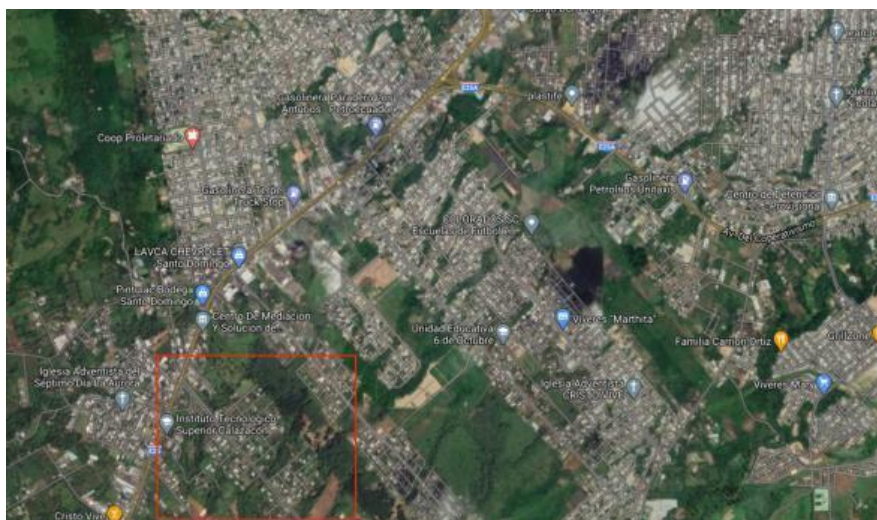
La mano del usuario 5 tiene una apariencia normal, la cuál cuando aprieta se le brotan las venas y los huesos.

Una vez mostrado las manos de los usuarios que realizarán dichas pruebas al modelo construido, se ejecutará una serie de pruebas para ir validando el modelo desarrollado por Custom Vision que funcione con normalidad.

Las pruebas consistieron en que cada usuario manipulará el modelo en diferentes ambientes (mañana, tarde y noche) y en diferentes territorios (territorio 1, territorio 2 y territorio 3) los cuales están ubicados dentro de la Ciudadela del Chofer en la ciudad de Santo Domingo de los Tsáchilas. A continuación, vemos una vista de Google Maps de esta ciudadela.

### Figura 52

*Ciudadela del Chofer, Santo Domingo de los Tsáchilas.*

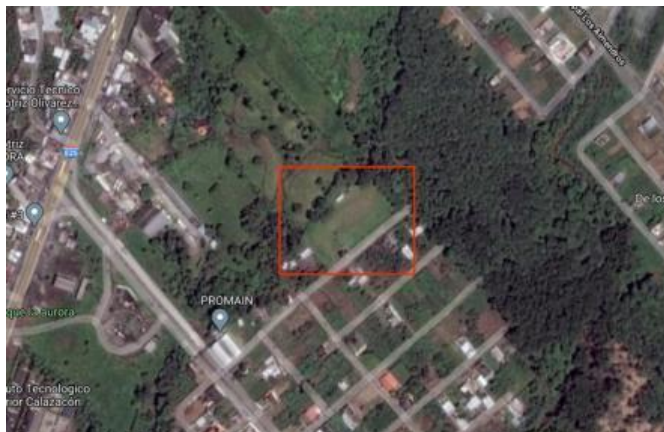


A continuación, se mostrará una imagen de Google Maps donde esta situado cada territorio.

El primer territorio está ubicado en la Ciudadela del Chofer a pocos metros de la entrada en la garita, es un campo de fútbol medianamente grande donde juegan personas habitualmente. Será denominado TERRITORIO #1 para dirigirse posteriormente en el texto. Al momento de las pruebas se irá mostrando fotografías de este territorio.

**Figura 53**

*Territorio 1 visualizado desde Google Maps.*



El segundo territorio esta ubicado en la Ciudadela del Chofer a dos kilómetros aproximadamente de la entrada en la garita, es un campo de fútbol grande donde juegan personas habitualmente. Será denominado TERRITORIO #2 para dirigirse posteriormente en el texto. Al momento de las pruebas se irá mostrando fotografías de este territorio.

**Figura 54**

*Territorio 2 visualizado desde Google Maps.*



El tercer territorio está ubicado en la Ciudadela del Chofer a dos kilómetros aproximadamente de la entrada en la garita y en frente del territorio #2, es una zona de aprendizaje para manejar carro donde una compañía de carros brinda sus servicios. Será denominado TERRITORIO #3 para dirigirse posteriormente en el texto. En la fotografía de Google Maps esta como un terreno vacío, pero actualmente cuenta con pequeñas vías para el aprendizaje. Al momento de las pruebas se irá mostrando fotografías de este territorio.

### Figura 55

*Territorio 3 visualizado desde Google Maps.*



La primera persona que manipuló con el dron fue el usuario 1, el cuál se le complicó un poco al momento de despegarlo ya que sus dedos son delgados y largos. En la fotografía se puede observar como despegó el dron Bebop 2 con el gesto correspondiente y lo mantiene en el aire hasta mostrar otro gesto en la cámara del celular. Se aprecia que para el ambiente de la mañana no tuvo problemas para manipularlo, únicamente fue hasta que el celular pudo reconocer la mano y obtener el gesto de ella.



**Figura 56**

*Despegue con el usuario 1.*



Para el usuario 2 fue distinto, la facilidad de elevarlo con el gesto de despegue fue rápido y conciso. No hubo ninguna novedad, en la fotografía se muestra el momento de despegue. En la fotografía se puede observar como despegue el dron Bebob 2.

**Figura 57**

*Despegue con el usuario 2.*



Para el usuario 3, no hubo novedades al momento de despegarlo, girarlo sobre su eje y consecuentemente aterrizarlo acotando que la piel de este usuario es más morena. En la fotografía se puede observar como aterriza el dron Bebob 2.

**Figura 58**

*Aterrizaje con el usuario 3.*



Con el usuario 4, hubo problemas al momento de despegar ya que los dedos de las manos no se estiraban bien, este problema se mantenía para gestos que abrían la mano completa para realizar el movimiento. El usuario mantuvo el vuelo durante algunos minutos y después tuvo que aterrizarlo por emergencia ya que el drone entendía algunos movimientos diferentes por complicaciones al abrir y cerrar la mano por completo. En la fotografía se puede observar como despegar el drone Bebop 2.

**Figura 59**

*Despegue con el usuario 4.*





Para el usuario 5 no hubo complicaciones, mantuvo el vuelo sin novedad y pudo aterrizarlo sin problema. En la fotografía se puede observar como despegó el drone Bebop 2.

**Figura 60**

*Despegue con el usuario 5.*



A continuación, se presenta una tabla con las dificultades MENOR, MEDIO y MAYOR propuestas por cada usuario al momento de la usabilidad:

**Tabla 1**

*Tabla de dificultad en el ambiente de la mañana en el territorio 1.*

Usuario	Gestos						Eje
	Despegue	Aterrizaje	Derecha	Izquierda	Delante	Atrás	
1	MAYOR	MEDIO	MAYOR	MAYOR	MAYOR	MAYOR	MEDIO
2	MENOR	MENOR	MAYOR	MAYOR	MAYOR	MAYOR	MENOR
3	MENOR	MENOR	MEDIO	MEDIO	MEDIO	MEDIO	MENOR
4	MENOR	MENOR	MEDIO	MEDIO	MEDIO	MEDIO	MEDIO
5	MENOR	MENOR	MENOR	MENOR	MENOR	MENOR	MENOR

La siguiente prueba se llevo a cabo en el territorio 2 donde el ambiente fue el mismo en la mañana. El primer usuario siendo el primero en probar tuvo las mismas dificultades.

**Figura 61**

*Hacia la derecha con el usuario 1.*



El usuario 2 no tuvo dificultad al momento de manipular el dron con los diferentes gestos, en la fotografía se puede ver como guía al dron hacia la izquierda.

**Figura 62**

*Hacia la izquierda con el usuario 2.*



El usuario 3 sin novedad al momento de controlar en este territorio, únicamente le costo al momento de reconocer el gesto por la apertura de la mano.

**Figura 63** *Despegando con el usuario 3.*



El usuario 4 tuvo el mismo problema al despegar que el usuario 3 ya que no podría abrir completamente la mano y situar los dedos rectos.

#### **Figura 64**

*Despegando con el usuario 4.*



El usuario 5 no tuvo problemas al manipular el dron en los diferentes gestos mencionados anteriormente. En la fotografía se puede observar aplicando el gesto hacia la derecha.

#### **Figura 65**

*Hacia la derecha con el usuario 5.*





A continuación, se presenta una tabla con las dificultades MENOR, MEDIO y MAYOR propuestas por cada usuario al momento de la usabilidad:

**Tabla 2** *Tabla de dificultad en el ambiente de la mañana en el territorio 2.*

Usuario	Gestos						
	Despegue	Aterrizaje	Derecha	Izquierda	Delante	Atrás	Eje
1	MEDIO	MEDIO	MEDIO	MEDIO	MEDIO	MEDIO	MAYOR
2	MAYOR	MAYOR	MENOR	MENOR	MAYOR	MAYOR	MEDIO
3	MENOR	MENOR	MENOR	MENOR	MENOR	MENOR	MENOR
4	MEDIO	MEDIO	MEDIO	MEDIO	MEDIO	MEDIO	MEDIO
5	MENOR	MENOR	MENOR	MENOR	MENOR	MENOR	MENOR

La siguiente prueba se llevo a cabo en el territorio 3 donde el ambiente fue el mismo en la mañana. El primer usuario siendo el primero en probar tuvo las mismas dificultades de los demás territorios mencionados al momento, además de que el territorio es diferente en fondo.

### Figura 66

*Despegue con el usuario 1.*



Con el usuario 2 hubo problemas al momento del despegue ya que en este territorio además de zonas verdes también hay zonas de asfalto donde en el modelo no se mostraba.

**Figura 67**

*Hacia la izquierda con el usuario 2.*



El usuario 3 no tuvo problemas al ejecutar los gestos, pero mismo problema que los anteriores usuarios el fondo cambia y el control cambia un poco al ejecutarlo cualquier movimiento.

**Figura 68**

*Aterrizaje con el usuario 3.*



El usuario 4 no tuvo problemas al momento de ejecutar los movimientos, pero si un poco de dificultad al iniciar con el despegue.

**Figura 69**

*Hacia la izquierda con el usuario 4.*



El usuario 5 no tuvo novedades, pero al momento de ejecutar los movimientos por primera vez tuvo un poco de dificultad por el fondo del territorio.





La siguiente prueba se llevo a cabo en el territorio 1 pero se cambio el ambiente a la tarde. El primer usuario siendo el primero en probar tuvo las mismas dificultades de los demás territorios mencionados al momento.

### Figura 71

*Hacia la derecha con el usuario 1.*



El usuario 2 no tuvo dificultad al momento de ejecutar todos los gestos y que el drone Bebop 2 responda correctamente.

### Figura 72

*Hacia la derecha con el usuario 2.*



El usuario 3 no tuvo dificultad al momento de ejecutar todos los gestos y que el dron  
Bebop 2 responda correctamente.

### **Figura 73**

*Despegue con el usuario 3.*



El usuario 4 no tuvo dificultad al momento de ejecutar todos los gestos y que el dron  
Bebop 2 responda correctamente.

### **Figura 74**

*Despegue con el usuario 4.*



El usuario 5 no tuvo dificultad al momento de ejecutar todos los gestos y que el dron  
Bebop 2 responda correctamente.



La siguiente prueba se llevo a cabo en el territorio 2 pero se cambio el ambiente a la tarde. El primer usuario funciono sin problema en este territorio.

### **Figura 76**

*Despegue con el usuario 1.*



El usuario 1 no tuvo dificultad al momento de ejecutar todos los gestos y que el dron e Bebop 2 responda correctamente.

### **Figura 77**

*Aterrizaje con el usuario 1.*



El usuario 2 no tuvo dificultad al momento de ejecutar todos los gestos y que el dron  
Bebop 2 responda correctamente.

### Figura 78

*Despegue con el usuario 2.*



El usuario 3 no tuvo dificultad al momento de ejecutar todos los gestos y que el dron  
Bebop 2 responda correctamente.

### Figura 79

*Despegue con el usuario 2.*







La siguiente prueba se llevo a cabo en el territorio 3 pero se cambio el ambiente a la tarde. El primer usuario funciono con leves problemas por el territorio ya que el modelo muestra zonas verdes y este territorio hay mezclado con asfalto.

### Figura 81

*Despegue con el usuario 1.*



El usuario 2 funciono con leves problemas por el territorio ya que el modelo muestra zonas verdes y este territorio hay mezclado con asfalto.

### Figura 82

*Hacia la derecha con el usuario 2.*



El usuario 2 funciono con leves problemas por el territorio ya que el modelo muestra zonas verdes y este territorio hay mezclado con asfalto.

**Figura 83**

*Hacia la derecha con el usuario 3.*



El usuario 2 funciono con leves problemas por el territorio ya que el modelo muestra zonas verdes y este territorio hay mezclado con asfalto.

**Figura 84**

*Despegue con el usuario 3.*







La siguiente prueba se llevo a cabo en el territorio 1 pero se cambio el ambiente a la noche. Con el primer usuario funciono levemente por el ambiente, ya que al no haber mucha luz y que el modelo estaba construido en unas zonas verde el drone Bebop 2 no pude ejecutar las acciones. Lo que hizo que despegue el drone Bebop 2 pero no tener un control total fue una lámpara de luz la cual enfocaba el sitio de pruebas.

### **Figura 86**

*Despegue con el usuario 1.*



Con el usuario 2 funciono levemente por el ambiente, ya que al no haber mucha luz y que el modelo estaba construido en unas zonas verde el drone Bebop 2 no pude ejecutar las acciones. Lo que hizo que despegue el drone Bebop 2 pero no tener un control total fue una lámpara de luz la cual enfocaba el sitio de pruebas.

**Figura 87**

*Despegue con el usuario 2.*



Con el usuario 3 funciono levemente por el ambiente, ya que al no haber mucha luz y que el modelo estaba construido en unas zonas verde el drone Bebop 2 no pude ejecutar las acciones. Lo que hizo que despegue el drone Bebop 2 pero no tener un control total fue una lámpara de luz la cual enfocaba el sitio de pruebas.

**Figura 88**

*Despegue con el usuario 3.*



Con el usuario 4 funciono levemente por el ambiente, ya que al no haber mucha luz y que el modelo estaba construido en unas zonas verde el drone Bebop 2 no pude ejecutar las acciones. Lo que hizo que despegue el drone Bebop 2 pero no tener un control total fue una lámpara de luz la cual enfocaba el sitio de pruebas.

### **Figura 89**

*Despegue con el usuario 4.*



Con el usuario 5 funciono levemente por el ambiente, ya que al no haber mucha luz y que el modelo estaba construido en unas zonas verde el drone Bebop 2 no pude ejecutar las acciones. Lo que hizo que despegue el drone Bebop 2 pero no tener un control total fue una lámpara de luz la cual enfocaba el sitio de pruebas.

### **Figura 90**

*Despegue con el usuario 5.*



A continuación, se presenta una tabla con las dificultades MENOR, MEDIO y MAYOR propuestas por cada usuario al momento de la usabilidad:

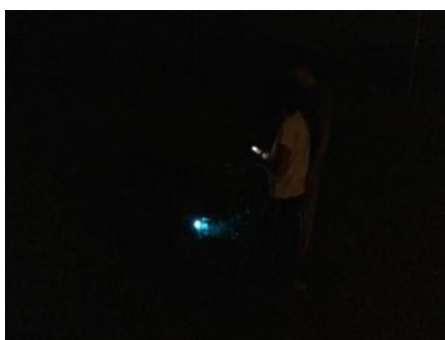
**Tabla 7** *Tabla de dificultad en el ambiente de la noche en el territorio 1.*

Usuario	Gestos						
	Despegue	Aterrizaje	Derecha	Izquierda	Delante	Atrás	Eje
1	MAYOR	MAYOR	MAYOR	MAYOR	MAYOR	MAYOR	MAYOR
2	MAYOR	MAYOR	MAYOR	MAYOR	MAYOR	MAYOR	MAYOR
3	MAYOR	MAYOR	MAYOR	MAYOR	MAYOR	MAYOR	MAYOR
4	MAYOR	MAYOR	MAYOR	MAYOR	MAYOR	MAYOR	MAYOR
5	MAYOR	MAYOR	MAYOR	MAYOR	MAYOR	MAYOR	MAYOR

La siguiente prueba se llevo a cabo en el territorio 2 pero se cambio el ambiente a la noche. No funciono con ningún usuario por el ambiente, ya que al no haber luz y que el modelo estaba construido en unas zonas verde el drone Bebop 2 no pude ejecutar las acciones.

**Figura 91**

*Intentando empezar el despegue.*



A continuación, se presenta una tabla con las dificultades MENOR, MEDIO y MAYOR propuestas por cada usuario al momento de la usabilidad:

**Tabla 8**

*Tabla de dificultad en el ambiente de la noche en el territorio 2*

Usuario	Gestos						
	Despegue	Aterrizaje	Derecha	Izquierda	Delante	Atrás	Eje
1	NO	NO	NO	NO	NO	NO	NO
	APLICA	APLICA	APLICA	APLICA	APLICA	APLICA	APLICA
2	NO	NO	NO	NO	NO	NO	NO
	APLICA	APLICA	APLICA	APLICA	APLICA	APLICA	APLICA
3	NO	NO	NO	NO	NO	NO	NO
	APLICA	APLICA	APLICA	APLICA	APLICA	APLICA	APLICA
4	NO	NO	NO	NO	NO	NO	NO
	APLICA	APLICA	APLICA	APLICA	APLICA	APLICA	APLICA
5	NO	NO	NO	NO	NO	NO	NO
	APLICA	APLICA	APLICA	APLICA	APLICA	APLICA	APLICA

La siguiente prueba se llevo a cabo en el territorio 3 pero se cambio el ambiente a la noche. No funciono con ningún usuario por el ambiente y porque estaba completamente oscuro.



### **Análisis de resultados**

Según las tablas adquiridas por los usuarios, en el mejor territorio para la manipulación de los usuarios fue en la tarde, ya que en la mañana no hubo tanta luz como en la tarde por el sol. El territorio preferido fue el segundo ya que la zona verde era mucho más grande y la ejecución era más efectiva.

Se dio algunas fallas al momento de realizar las pruebas en el territorio 2 con los usuarios ya que el territorio era diferente en su fondo cambiaba y se agregaba asfalto de color gris, este asfalto pudo tener algunas complicaciones y mezclas de movimientos al momento de su funcionamiento con los usuarios.

Los dos últimos casos en la noche y en los territorios 2 y 3 se produjo un error no aplicable ya que el drone no pudo despegar con ningún usuario, esto se debe a que no había nada de luz y se complicó ya que el modelo se construyó con luz y en zonas verdes.

Los movimientos también en ciertos momentos se confundieron, ya que el algoritmo realizó 105 imágenes y tenía una precisión cerca del 100%. Además, los gestos seleccionados para los movimientos eran parecidos y en ciertos gestos no cambiaba mucho la forma de la mano.

### **Trabajos futuros**

En base a los movimientos con los gestos mediante Custom Vision habría una posibilidad de realizar la opción "*customizable*" o también llamado personalizable, donde el usuario podría agregar sus propios gestos de acuerdo a su interés.



En base a la aplicación móvil a trabajos a futuro se podría intentar implementar un servicio de distribución en la plataforma de iOS, incluyendo así unas compras dentro de la aplicación en la cual se pueda distribuir a nivel mundial. Otro aspecto es mejor la pantalla inicial de la aplicación incluyendo un *“Social Login”* o también llamado Inicio de sesión mediante redes sociales. Esto ayudará a tener unas analíticas de los usuarios que descarguen la aplicación.

La aplicación puede ser escalable a largo plazo y poder distribuirla también los dispositivos con sistema operativo Android. Esto se tendría que realizar con el IDE Android Studio el cual se programará con la librería de Parrot S.A. ya desarrollada en lenguaje JAVA.

De acuerdo a la creación del modelo mediante Microsoft Custom Vision se planteará construir un modelo con diversas imágenes en diferente ambiente, se podrá detectar funcionamiento óptimo al momento de que el dron Bebob 2 este en el aire. Se podrá utilizar un número de imágenes más amplia, es decir, mas de 105 imágenes por gesto humano.

## Capítulo VI

### Conclusiones y recomendaciones

#### Conclusiones

En el primer capítulo hubo una sencilla introducción donde se elaboró la importancia de la clasificación de imágenes, especialmente en algunos campos de la industria. La clasificación de imágenes tiene un papel importante en las características de nuestros teléfonos móviles, en la clasificación de numerosas entidades, en la medicina, pero también en la industria del automóvil autónomo. Debido al último ejemplo, en este trabajo se introdujo un problema de clasificación de movimientos para controlar el drone Bebop 2.

Se muestra cómo construir un modelo usando un conjunto de datos que necesita estar completamente equilibrado. Las imágenes que difieren en tamaño, luz, ángulo de cámara, pero también borrosas, están teniendo un gran impacto en el modelo y su precisión y recuerdo.

El servicio Custom Vision orientado al control de movimientos del drone Bebop 2 es un algoritmo de última generación con gran capacidad de aprendizaje. Dado que los movimientos son una tarea tediosa que requiere bastante tiempo que los usuarios a menudo no tienen, un servicio de visión personalizado proporciona esta parte para construir un clasificador, el cual se utilizará para construir el modelo y adaptarlo a la aplicación móvil. El usuario necesita preparar los datos y etiquetarlos adecuadamente, debe proporcionar un número adecuado de imágenes en el conjunto de datos y dividirlo en una proporción adecuada entre los datos de entrenamiento y de prueba.

Se demostró que este enfoque arroja resultados interesantes y una tasa de precisión bastante satisfactoria al momento de leerlos en la herramienta de Custom Vision. Sin embargo, se mostraron algunas limitaciones, por ejemplo, el modelo debería entrenarse en escenarios más reales donde podría haber muchos elementos delante del gesto, pero el modelo aún lograría clasificar el gesto con una mejor tasa de probabilidad. Sin embargo, los resultados obtenidos hasta ahora también fueron muy buenos ya que sobrepasaron todos el 90% de precisión.

Deben mejorarse para que este modelo pueda convertirse en parte de algo más grande que pueda tener un impacto en la industria. La comparación con SVM mostró un buen rendimiento del modelo Custom Vision, lo que confirma que este enfoque en el aprendizaje automático podría ser bien aceptado y desarrollado en futuras aplicaciones móviles.

### **Recomendaciones**

Realizar una investigación profunda con los diferentes frameworks para realizar una esqueletización, extracción de características y clasificadores para el reconocimiento de gestos. Algunos de estos frameworks optimizarán el tiempo al implementarlos en las aplicaciones móviles desarrolladas para smartphones.

Al momento de desarrollar la aplicación móvil es ideal realizar una prueba interna con diferentes modelos Custom Vision para tener en cuenta cuentas imágenes aproximadamente necesitaremos para que nuestro modelo funcione con optimización y sea lo mas preciso posible.

Custom Vision es una herramienta desarrollada por Microsoft la cual es una gran opción para realizar este tipo de proyectos de aprendizaje de máquina de supervisado ya que ahorra tiempo al construir el modelo e ir implementando el desarrollo de la aplicación móvil

No olvidar las configuraciones para el desarrollo de Open Source con el drone Bebop 2 al momento de programar con Swift y Objective C ya que para que estos dos lenguajes funcionen debe realizarse un puente o también llamado Bridge, una vez configurado el puente se puede acceder a las clases de la librería del drone Bebop 2.

## Referencias

Adakane, D. (2019, Noviembre 12). *What are Haar Features used in Face Detection ?* Retrieved from Medium: <https://medium.com/analytics-vidhya/what-is-haar-features-used-in-face-detection-a7e531c8332b>

Aguilar, W. G., & Angulo, C. (2016). Real-Time Model-Based Video Stabilization for Microaerial Vehicles. *Neural Processing Letters*, 43 (2), 459-477.

Aguilar, W. G., & Angulo, C. (2014). Real-time video stabilization without phantom movements for micro aerial vehicles. *EURASIP Journal on Image and Video Processing*, 1, 1-13.

Aguilar, W. G., & Morales, S. (2016). 3D Environment Mapping Using the Kinect V2 and Path Planning Based on RRT Algorithms. *Electronics*, 5 (4), 70.

Aguilar, W. G., Álvarez, L., Grijalva, S., & Rojas, I. (2019). Monocular Vision-Based Dynamic Moving Obstacles Detection and Avoidance. In *Lecture Notes in Computer Science* (pp. 386-398).

Aguilar, W. G., Angulo, C., & Costa-Castello, R. (2017). Autonomous Navigation Control for Quadrotors in Trajectories Tracking. In *Lecture Notes in Computer Science* (pp. 287-297).

Aguilar, W. G., Angulo, C., & Pardo, J. A. (2017). Motion intention optimization for multirotor robust video stabilization. *Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucón, Chile.

Aguilar, W. G., Casaliglla, V. P., & Pólit, J. L. (2017). Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles. *Electronics*, 6 (1), 10.

Aguilar, W. G., Cobeña, B., Rodriguez, G., Salcedo, V. S., & Collaguazo, B. (2018). SVM and RGB-D Sensor Based Gesture Recognition for UAV Control. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (pp. 713-719). Springer.

Aguilar, W. G., Luna, M. A., Moya, J. F., Abad, V., Parra, H., & Ruiz, H. (2017). Pedestrian detection for UAVs using cascade classifiers with meanshift. *IEEE 11th international conference on semantic computing (ICSC)*.

Aguilar, W. G., Luna, M. A., Moya, J. F., Abad, V., Ruiz, H., Parra, H., et al. (2017). Pedestrian Detection for UAVs Using Cascade Classifiers and Saliency Maps. In *Lecture Notes in Computer Science* (pp. 563-574).

Aguilar, W. G., Luna, M. A., Ruiz, H., Moya, J. F., Luna, M. P., Abad, V., et al. (2017). Statistical Abnormal Crowd Behavior Detection and Simulation for Real-Time Applications. In *Lecture Notes in Computer Science* (pp. 671-682).

Aguilar, W. G., Manosalvas, J. F., Guillén, J. A., & Collaguazo, B. (2018). Robust Motion Estimation Based on Multiple Monocular Camera for Indoor Autonomous Navigation of Micro Aerial Vehicle. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (pp. 547-561). Springer.

Aguilar, W. G., Quisaguano, F. J., Alvarez, L. G., Pardo, J. A., & Zahira, P. (2018). Monocular depth perception on a micro-UAV using convolutional neuronal networks. In *Lecture Notes in Computer Science* (pp. 392-397).

Aguilar, W. G., Quisaguano, F. J., Rodríguez, G. A., Alvarez, L. G., Limaico, A., & S, S. D. (2018). Convolutional neuronal networks based monocular object detection and depth perception for micro UAVs. In *Lecture Notes in Computer Science* (pp. 401-410).

Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (2017). On-Board Visual SLAM on a UGV Using a RGB-D Camera. In *Lecture Notes in Computer Science* (pp. 298-308).

Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (2017). Real-Time 3D Modeling with a RGB-D Camera and On-Board Processing. In *Lecture Notes in Computer Science* (pp. 410-419).

Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (2017). Visual SLAM with a RGB-D Camera on a Quadrotor UAV Using on-Board Processing. In *Lecture Notes in Computer Science* (pp. 596-606).

Aguilar, W. G., Salcedo, V., Sandoval, D., & Cobeña, B. (2017). Developing of a Video-Based Model for UAV Autonomous Navigation. *Communications in Computer and Information Science*, 94-105.

Alexander Van't Hof. (2019). *AnDrone: Virtual Drone Computing in the Cloud*. Retrieved from ACM: Digital Library.

Alpaydin. (2009). *MIT*. Retrieved from Introduction to machine learning.

Amaguaña, F., Collaguazo, B., Tituaña, J., & Aguilar, W. G. (2018). Simulation System Based on Augmented Reality for Optimization of Training Tactics on Military Operations. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (pp. 394-403). Springer.

Andrea, C. C., Byron, J. Q., Jorge, P. I., Inti, T. C., & Aguilar, W. G. (2018). Geolocation and Counting of People with Aerial Thermal Imaging for Rescue Purposes. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (pp. 171-182). Springer.

Apple. (2014). Retrieved from Swift Language Organization.

- Apple. (2020, Octubre 18). *Accesorios*. Retrieved from <https://www.apple.com/sg/shop/product/HJW62PA/A/parrot-bebop-2-drone>
- Apple Developer. (2020). *Apple Developer Documentation*. Retrieved from Developer.apple.com: <https://developer.apple.com/documentation/coreml>
- Apple Developer Program. (2019). *Importing Objective-C into Swift*. Retrieved from [https://developer.apple.com/documentation/swift/imported\\_c\\_and\\_objective-c\\_apis/importing\\_objective-c\\_into\\_swift](https://developer.apple.com/documentation/swift/imported_c_and_objective-c_apis/importing_objective-c_into_swift)
- Azure Custom Vision. (2019). *Build a Custom Vision Service classifier*. Retrieved from Custom Vision: <https://docs.microsoft.com/en-us/azure/cognitive-services/Custom-Vision-Service/getting-started-build-a-classifier>
- Basantes, J., Godoy, L., Carvajal, T., Castro, R., Toulkeridis, T., Fuertes, W., et al. (2018). Capture and processing of geospatial data with laser scanner system for 3D modeling and virtual reality of Amazonian Caves. *IEEE Ecuador Technical Chapters Meeting (ETCM)*. Samborondón, Ecuador.
- Boyeon Kim - Munseong Jung - Yunseok Chang. (2018). *Inflight Drone Re-routing Method with Google Map on Smart Pad*. Retrieved from *Cyber-Physical Systems, & Applications*.
- Cabras, P., Rosell, J., Pérez, A., Aguilar, W. G., & Rosell, A. (2011). Haptic-based navigation for the virtual bronchoscopy. *18th IFAC World Congress*. Milano, Italy.
- Cadoz. (2010). *Pereira Ghiena Alejandro*. Retrieved from *El gesto corporal como acción epistémica en la lectura cantada a primera vista*.
- Caiza, O., Aguilar, W. G., Albán, P., & Fernández, Y. (2020). Kinect and Manipulator-Based Sample Collection System for Military Robot. In *Smart Innovation, Systems and Technologies* (pp. 75-87).



Calderón, M., Aguilar, W. G., & Merizalde, D. (2020). Visual-Based Real-Time Detection Using Neural Networks and Micro-UAVs for Military Operations. In *Smart Innovation, Systems and Technologies* (pp. 55-64).

Castro, L. E. (2013). *Caracterización del gesto deportivo patada de balón en movimiento en el fútbol sala*. Retrieved from Estudio de caso.

Choodum, A. (2013). *Using the iPhone as a device for a rapid quantitative analysis of trinitrotoluene in soil*. Science Direct.

ChrisPudney. (2002). *Distance-Ordered Homotopic Thinning: A Skeletonization Algorithm for 3D Digital Images*. Nedlands: ScienceDirect.

Cobeña, B. (2018). *RECONOCIMIENTO DE GESTOS CORPORALES BASADO EN SVM Y SENSOR RGB-D PARA EL CONTROL DE MICRO VEHÍCULOS AÉREOS MULTIROTOR*. Sangolquí: ESPE.

Developer, P. f. (2012). *How to use iOS Libraries*. Parrot Bebop Drone.

Drawnonward. (2019). *Stack Overflow*. Retrieved from Concise description of how .h and .m files interact in objective c? : <https://stackoverflow.com/questions/2619048/concise-description-of-how-h-and-m-files-interact-in-objective-c>

DroneRush. (2013). *First time flying a drone? Don't make these mistakes*. New York: DroneRush.

Efecto Apple. (2014). *Implementación de MVC*. Retrieved from MVC.

Fang, Y. (2008, Agosto 31). *IEEE Xplore*. Retrieved from Improving LBP features for gender classification: <https://ieeexplore.ieee.org/abstract/document/4635807>

Franco Ronchetti. (2017). *Reconocimiento de gestos dinámicos y su aplicación al lenguaje de señas*. Argentina: Facultad de Informática, Universidad Nacional de La Plata.

(2017). *Future of mobile software for smartphones and drones: energy and performance*.

Buenos Aires: ACM Digital Library.

Galarza, J., Pérez, E., Serrano, E., Tapia, A., & G, A. W. (2018). Pose estimation based on monocular visual odometry and lane detection for intelligent vehicles. In *Lecture Notes in Computer Science* (pp. 562-566).

Galindo, R., Aguilar, W. G., & Reyes Ch, R. P. (2019). Landmark based eye ratio estimation for driver fatigue detection. In *Lecture Notes in Computer Science* (pp. 565-576). Germany: Springer.

Gate, R. (2015, Noviembre 5). *ResearchGate*. Retrieved from HOG feature extraction: [https://www.researchgate.net/figure/HOG-features-extraction-process-image-is-divided-in-cells-of-size-N-N-pixels-The\\_fig10\\_283543555](https://www.researchgate.net/figure/HOG-features-extraction-process-image-is-divided-in-cells-of-size-N-N-pixels-The_fig10_283543555)

Gregory. (2016). *The future of UAVs in ecology: an insider perspective from the Silicon Valley drone industry*. Berkeley: Journal of Unmanned Vehicle Systems.

Grijalva, S., & Aguilar, W. G. (2019). Landmark-Based Virtual Path Estimation for Assisted UAV FPV Tele-Operation with Augmented Reality. In *Lecture Notes in Computer Science* (pp. 688-700). Germany: Springer.

Grijalva, S., & Aguilar, W. G. (2019). Landmark-Based Virtual Path Estimation for Assisted UAV FPV Tele-Operation with Augmented Reality. In *Lecture Notes in Computer Science* (pp. 688-700).

Grijalva, S., Cueva, G., Ramírez, D., & G, A. W. (2019). HSVM-Based Human Activity Recognition Using Smartphones. In *Lecture Notes in Computer Science* (pp. 217-228).

HAAR. (2019). *What are Haar Features used in Face Detection ?* Retrieved from Medium: [https://upload.wikimedia.org/wikipedia/commons/thumb/3/31/VJ\\_featureTypes.svg/640px-VJ\\_featureTypes.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/3/31/VJ_featureTypes.svg/640px-VJ_featureTypes.svg.png)

- Jara-Olmedo, A., Medina-Pazmiño, W., Mesías, R., Araujo-Villaroel, B., Aguilar, W. G., & Pardo, J. A. (2018). Interface of Optimal Electro-Optical/Infrared for Unmanned Aerial Vehicles. In *Smart Innovation, Systems and Technologies* (pp. 372-380).
- Jara-Olmedo, A., Medina-Pazmiño, W., Tozer, T., Aguilar, W. G., & Pardo, J. A. (2018). E-services from Emergency Communication Network: Aerial Platform Evaluation. *International Conference on eDemocracy & eGovernment (ICEDEG)* (pp. 251-256). IEEE.
- Jiang, D. (2018). *Gesture recognition based on skeletonization algorithm and CNN with ASL database*. Beijing: Spring Link.
- JoséPérez. (2013). *Multi-clasificación Discriminativa de Partes Corporales basada en Códigos Correctores de Errores*. Barcelona: Facultat de Matemàtiques Universitat de Barcelona.
- Liu, Z. (2015). *Rise of Mini-Drones: Applications and Issues*. Hangzhou: ACM Digital Library.
- Luis Villacres. (2018). *Diseño de un sistema interfaz para el reconocimiento y traducción de gestos corporales al lenguaje natural (escrito, hablado) mediante el sensor Kinect de Microsoft, para personas con capacidades diferentes*. Guayaquil: Universidad Católica de Santiago de Guayaquil.
- Matea Pejčinović.(2019). *A Review of Custom Vision Service for Facilitating an Image Classification*. Varaždin: Asseco SEE.
- Merizalde, D., Aguilar, W. G., & Calderón, M. (2020). Autonomous Navigation Based on Proportional Controller with GPS Setpoint for UAV in External Environments. In *Smart Innovation, Systems and Technologies* (pp. 89-99).
- Microsoft Custom Vision. (2019). *Evaluate the classifier*. Retrieved from <https://docs.microsoft.com/en-us/azure/cognitive-services/custom-vision-service/getting-started-build-a-classifier>

- Microsoft. (2019). *What is Azure Custom Vision?* Retrieved from Custom Vision:  
<https://docs.microsoft.com/en-us/azure/cognitive-services/Custom-Vision-Service/home>
- Narman Husnu. (2020). *What Do People Complain About Drone Apps? A Large-Scale Empirical Study of Google Play Store Reviews*. Retrieved from El Sevier.
- Nieto, O. (2008). *Hand Gesture Recognition in Mobile Devices*. Nueva York.
- Orbea, D., Moposita, J., Aguilar, W. G., Paredes, M., León, G., & Jara-Olmedo, A. (2017). Math Model of UAV Multi Rotor Prototype with Fixed Wing Aerodynamic Structure for a Flight Simulator. In *Lecture Notes in Computer Science* (pp. 199-211).
- Orbea, D., Moposita, J., Aguilar, W. G., Paredes, M., Reyes, R. P., & Montoya, L. (2017). Vertical take off and landing with fixed rotor. *Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucón, Chile.
- O'Shea, K. (2015). *An Introduction to Convolutional Neural Networks*. Cornell University.
- Pardo, J. A., Aguilar, W. G., & Toulkeridis, T. (2017). Wireless communication system for the transmission of thermal images from a UAV. *Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucón, Chile.
- Parrot. (2012). *How to use iOS libraries*. Retrieved from Parrot Bebop Drone for Developers.
- Pascual, J. A. (2018). *Google vs Apple: la guerra de los smartphones en cifras*. Madrid.
- Quora. (2012). *How difficult is it to become a military drone pilot?* Georgia: Quora.
- Razeware. (2018). *Design Patterns: MVVM*. RayWenderlich .
- ResearchGate. (2015). Retrieved from [https://www.researchgate.net/figure/HOG-features-extraction-process-image-is-divided-in-cells-of-size-N-N-pixels-The\\_fig10\\_283543555](https://www.researchgate.net/figure/HOG-features-extraction-process-image-is-divided-in-cells-of-size-N-N-pixels-The_fig10_283543555)

- ResearchGate. (2020, Enero 25). *Cross-Validation K-Fold*. Retrieved from [https://www.researchgate.net/figure/Block-diagram-describing-k-fold-cross-validation-and-grid-search-process-for-SVM-SVR\\_fig8\\_338630782](https://www.researchgate.net/figure/Block-diagram-describing-k-fold-cross-validation-and-grid-search-process-for-SVM-SVR_fig8_338630782)
- Retinosis. (2020, Febrero 02). *SpeakLiz*. Retrieved from <https://retinosis.org/speakliz-vision-la-app-que-permite-reconocer-objetos-a-las-personas-con-discapacidad-visual/>
- Rodrigo Ibañez - Álvaro Soria - Alfredo Teyseyre - Marcelo Campo. (2014). *Evaluación de técnicas de Machine Learning para el*. ISISTAN Research Institute.
- Salcedo, V. S., Aguilar, W. G., Cobeña, B., Pardo, J. A., & Zahira, P. (2018). On-board target virtualization using image features for UAV autonomous tracking. In *Lecture Notes in Computer Science* (pp. 384-391).
- Sanna, A. (2013). *A Kinect-based natural interface for quadrotor control*. Science Direct.
- Saravanan. (2014). *REVIEW ON CLASSIFICATION BASED ON ARTIFICIAL NEURAL NETWORKS*. India: Department of Computer Engineering, Erode Sengunthar Engineering College, Erode,.
- Sarkar, A. (2016). *Gesture control of drone using a motion controller*. Sharjah: IEEEExplore.
- Sayalee Pote. (2019). *Using Core ML and Custom Vision to Build a Real-Time Hand Sign Detector in iOS*. Retrieved from Medium: <https://heartbeat.fritz.ai/building-a-real-time-hand-sign-detector-in-ios-using-core-ml-and-custom-vision-e76aad7961ac>
- Schlömer, T. (2008). *Gesture recognition with a Wii controller*. Bonn: ACM Digital Library.
- Segarra, D., Caballeros, J., Aguilar, W. G., Samà, A., & Rodríguez-Martín, D. (2019). Orientation Estimation Using Filter-Based Inertial Data Fusion for Posture Recognition. In *Lecture Notes in Computer Science* (pp. 220-233).
- Song, J. (2014). *In-air gestures around unmodified mobile devices*. Hawaii: ACM Digital Library.

SVM. (2019). *Support Vector Machines*. Retrieved from

<https://scikit-learn.org/stable/modules/svm.html>

Tech. (2011). *Control de gestos, aprendizaje profundo y realidad virtual: los drones se sofistican*. Madrid: Tech.

Techradar. (2017). *Parrot Bebop 2*. Retrieved from

<https://www.techradar.com/reviews/parrot-bebop-2>

Tenezaca, D., Canchignia, C., Aguilar, W. G., & Mendoza, D. (2019). Implementation of Dubin Curves-Based RRT\* Using an Aerial Image for the Determination of Obstacles and Path Planning to Avoid Them During Displacement of the Mobile Robot. In *Smart Innovation, Systems and Technologies* (pp. 205-215).

Tovr, R. (2017). *ManoMotion Brings Hand Gesture Input to Apple's ARKit*. Chicago.

Vecteezy. (2019). Retrieved from <https://es.vecteezy.com/arte-vectorial/1437117-coleccion-de-iconos-de-lenguaje-de-senas-y-gestos-con-las-manos>

Wathen-Dunn. (1998). Distance-Ordered Homotopic Thinning: A Skeletonization Algorithm for 3D Digital Images. *ScienceDirect* .

Zurita, M., Aguilar, W. G., & Enríquez, V. (2019). Toward the Development of Surveillance and Reconnaissance Capacity in Ecuador: Geolocation System for Ground Targets Based on an Electro-Optical Sensor. In *Smart Innovation, Systems and Technologies* (pp. 123-133).