



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**CARRERA DE INGENIERIA ELECTRÓNICA Y AUTOMATIZACIÓN**

**“DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL  
BASADO EN REDES CONVOLUCIONALES PARA EL CORRECTO  
POSICIONAMIENTO DE UN VEHÍCULO AUTÓNOMO EN UN  
CARRIL”**

**Trabajo de titulación**  
Tipo: Dispositivo Tecnológico

Presentado para optar al grado académico de:  
**INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN**

**AUTOR: YANDRY ALEXANDER JÁCOME MONTERO**

**DIRECTOR: ING. JORGE PAUCAR SAMANIEGO.**

Riobamba – Ecuador  
2021

**©2021, Yandry Alexander Jácome Montero**

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, Yandry Alexander Jácome Montero, declaro que el presente trabajo de titulación es de nuestra autoría y los resultados de este son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autores asumimos la responsabilidad legal y académica de los contenidos de este trabajo de titulación; El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

16-08-2021

---

Yandry Alexander Jácome Montero  
020247550-5

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**CARRERA DE INGENIERIA ELECTRÓNICA Y AUTOMATIZACION**

El Tribunal del trabajo de titulación certifica que: El trabajo de titulación: Tipo: Dispositivo Tecnológico, **DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL BASADO EN REDES CONVOLUCIONALES PARA EL CORRECTO POSICIONAMIENTO DE UN VEHÍCULO AUTÓNOMO EN UN CARRIL**, realizado por el señor YANDRY ALEXANDER JÁCOME MONTERO, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal autoriza su presentación.

<b>NOMBRE</b>	<b>FIRMA</b>	<b>FECHA</b>
ING. MAYRA ALEJANDRA PACHECO CUNDURI PRESIDENTA DEL TRIBUNAL	 Firmado digitalmente por: MAYRA ALEJANDRA PACHECO CUNDURI	16-08-2021
ING. JORGE LUIS PAUCAR SAMANIEGO DIRECTOR DEL TRABAJO DE TITULACIÓN	 Firmado digitalmente por: JORGE LUIS PAUCAR SAMANIEGO	16-08-2021
ING. JOSE LUIS TINAJERO LEON MIEMBRO DEL TRIBUNAL	 Firmado digitalmente por: JOSE LUIS TINAJERO	16-08-2021

## **DEDICATORIA**

El siguiente trabajo de titulación se lo quiero dedicar a todas esas personas que me ayudaron a hacer una mejor persona y enseñarme que puedo dar lo mejor de mí, en especial a mi familia: mis padres, mis hermanos y a mi novia que han sido un pilar fundamental en el desarrollo y en el apoyo para conseguir esto, después de ellos, mis amigos, mis maestros y todas las personas que ya no se encuentran aquí pero que me enseñaron cosas muy valiosas.

Yandry

## **AGRADECIMIENTO**

En primer lugar, agradezco a Dios por darme la fuerza la sabiduría y el temple para poder alcanzar todas las metas que me estoy proponiendo. agradezco a mis padres que han sido lo más importante en mi desarrollo con su apoyo incondicional y su amor que han hecho de esta persona un gran ser humano. agradezco a mis hermanos que han sido una pieza clave para obtener el impulso para cumplir mis metas. A todos mis amigos y maestros que me han enseñado mucho en toda esta etapa que la extrañaré con toda el alma.

Yandry

## TABLA DE CONTENIDO

INDICE DE TABLAS .....	xi
INDICE DE FIGURAS.....	xii
ÍNDICE DE ABREVIATURAS .....	xvi
ÍNDICE DE ANEXOS.....	xvii
RESUMEN .....	xviii
ABSTRACT.....	xix
INTRODUCCIÓN .....	1

## CAPÍTULO I

<b>1. MARCO TEÓRICO REFERENCIAL .....</b>	<b>3</b>
<b>1.1. Visión artificial .....</b>	<b>3</b>
<i>1.1.1. Ventajas .....</i>	<i>2</i>
<i>1.1.2. Aplicaciones.....</i>	<i>2</i>
<i>1.1.2.1. Filtros de Snapchat .....</i>	<i>2</i>
<i>1.1.2.2. Amazon Go.....</i>	<i>3</i>
<i>1.1.2.3. Lente de Pinterest.....</i>	<i>4</i>
<i>1.1.2.4. Amazon Echo Look.....</i>	<i>4</i>
<i>1.1.2.5. Control y seguridad vehicular .....</i>	<i>5</i>
<i>1.1.2.6. Robótica .....</i>	<i>5</i>
<b>1.3. Definiciones.....</b>	<b>6</b>
<i>1.3.1. Píxel .....</i>	<i>6</i>
<i>1.3.2. Imagen digital.....</i>	<i>7</i>
<i>1.3.3. Resolución .....</i>	<i>9</i>
<i>1.3.4. Brillo .....</i>	<i>9</i>
<i>1.3.5. Contraste.....</i>	<i>10</i>

1.3.6.	<b>Tono</b> .....	12
1.3.7.	<b>Saturación</b> .....	12
1.3.8.	<b>Color</b> .....	14
1.4.	<b>Procesamiento digital de imágenes</b> .....	14
1.4.1.	<b>Problemas al analizar Imágenes</b> .....	15
1.4.2.	<b>Elementos de un sistema de visión artificial</b> .....	16
1.4.2.1.	<i>Cámara digital</i> .....	16
1.4.2.2.	<i>Fuente de luz</i> .....	17
1.4.2.3.	<i>Óptica</i> .....	17
1.4.2.4.	<i>Software</i> .....	17
1.4.2.5.	<i>Actuadores</i> .....	17
1.5.	<b>Ordenador</b> .....	18
1.5.1.	<b>Raspberry Pi 4</b> .....	18
1.5.2.	<b>LeMaker HiKey 960</b> .....	19
1.5.3.	<b>Asus Tinker Board</b> .....	20
1.5.4	<b>Orange Pi Plus 2</b> .....	21
1.5.5.	<b>La mejor opción</b> .....	22
1.6.	<b>Módulos de Cámaras Raspberry</b> .....	23
1.6.1	<b>Raspberry Pi Módulo de cámara V2 – 8 megapíxeles, 1080p.</b> .....	23
1.6.2.	<b>Raspberry Pi Camera Board - Night Vision "IR-CUT" (5MP)</b> .....	24
1.7.	<b>Software usado en el desarrollo del proyecto</b> .....	25
1.7.1	<b>Raspbian</b> .....	25
1.7.2	<b>OpenCV</b> .....	26
1.7.3.	<b>Tensorflow</b> .....	27
1.7.4.	<b>Anaconda</b> .....	27
1.7.5.	<b>Keras</b> .....	28
1.7.6.	<b>Google Colab</b> .....	28

1.8.	<b>Redes neuronales</b> .....	29
1.8.1.	<i>Aprendizaje supervisado</i> .....	30
1.8.2.	<i>Aprendizaje reforzado</i> .....	31
1.8.3.	<i>Aprendizaje sin supervisión</i> .....	31
1.8.4.	<i>Desarrollo de una red neuronal</i> .....	32
1.9.	<b>Arquitectura de una red neuronal</b> .....	33
1.9.1.	<i>Red de alimentación directa de una sola capa</i> .....	33
1.9.2.	<i>Red de alimentación de varias capas</i> .....	34
1.9.3.	<i>Red recurrente</i> .....	35
1.10.	<b>Redes Neuronales Convolucionales (CNN)</b> .....	36
1.10.1.	<i>Capa convolucional</i> .....	38
1.10.2.	<i>Capa pooling</i> .....	39
1.10.3.	<i>Capa de densamente Conectada</i> .....	40

## CAPÍTULO II

2.	<b>MARCO METODOLÓGICO</b> .....	41
2.1.	<b>Requerimientos del prototipo</b> .....	41
2.2.	<b>Consolidación general del prototipo</b> .....	41
2.3.	<b>Arquitectura de hardware del prototipo</b> .....	42
2.4.	<b>Diseño de las etapas de <i>hardware</i> del prototipo</b> .....	43
2.4.1.	<i>Etapas de alimentación</i> .....	44
2.4.2.	<i>Etapas de obtención de datos</i> .....	44
2.4.3.	<i>Etapas de procesamiento</i> .....	45
2.5.	<b>Selección de componentes de <i>hardware</i> para la construcción del prototipo</b> .....	46
2.5.1.	<i>Chasis lamborghini RC</i> .....	46
2.6.1.	<i>Dispositivos de obtención de datos</i> .....	47
2.6.1.1.	<i>Raspberry Pi Módulo de cámara V2 – 8 megapíxeles, 1080p</i> .....	47

<b>2.6.2.</b>	<b><i>Dispositivos de procesamiento</i></b> .....	49
2.6.2.1.	<i>Arduino uno</i> .....	49
2.6.2.2.	<i>Raspberry pi 4 4GB RAM</i> .....	50
<b>2.6.3.</b>	<b><i>Dispositivos de alimentación</i></b> .....	51
2.6.3.1.	<i>Regulador de voltaje LM2596</i> .....	51
2.6.3.2.	<i>Batería Li-Po</i> .....	52
<b>2.7.</b>	<b>Diagrama de conexión para el procesador central</b> .....	54
<b>2.8.</b>	<b>Distribución de componentes internos</b> .....	56
<b>2.9.</b>	<b>Distribución de componentes externos</b> .....	58
<b>2.9.1.</b>	<b>Distribución de peso</b> .....	58
<b>2.9.2.</b>	<b><i>Posición de periféricos</i></b> .....	59
<b>2.10.</b>	<b>Funcionamiento mecánico</b> .....	60
<b>2.11.</b>	<b>Requerimientos de diseño para el software</b> .....	61
<b>2.12.</b>	<b>Arquitectura del <i>software</i> del prototipo</b> .....	61
<b>2.12.1.</b>	<b><i>Herramientas de software</i></b> .....	61
2.12.1.1.	<i>Arduino IDE 1.8.13</i> .....	61
2.12.1.2.	<i>Raspbian OS</i> .....	62
2.12.1.3.	<i>Python</i> .....	62
<b>2.12.2.</b>	<b><i>Desarrollo del código para el sistema de detección de carril</i></b> .....	62
<b>2.12.3.</b>	<b><i>Desarrollo del código para el procesador esclavo</i></b> .....	64
<b>2.12.4.</b>	<b><i>Protocolo de comunicación maestro-esclavo</i></b> .....	65
<b>2.12.5.</b>	<b><i>Corrección Mecánica</i></b> .....	65
<b>2.13.</b>	<b>Instalación de Raspbian</b> .....	66
<b>2.14.</b>	<b>Red neuronal convolucional creada</b> .....	70
<b>2.15.</b>	<b>Red neuronal convolucional entrenada</b> .....	72
<b>2.15.1.</b>	<b><i>Preprocesamiento de imágenes</i></b> .....	72
<b>2.15.2.</b>	<b><i>Parámetros de entrenamiento de la red neuronal convolucional</i></b> .....	73

### **CAPÍTULO III**

<b>3.</b>	<b>PRUEBAS Y RESULTADOS DEL PROTOTIPO</b> .....	75
<b>3.1.</b>	<b>Análisis de la precisión de la red neuronal convolucional por épocas</b> .....	75
<b>3.2.</b>	<b>Evaluación del desempeño del modelo</b> .....	81
<b>3.3.</b>	<b>Predicciones realizadas</b> .....	84
<b>3.4.</b>	<b>Prueba del sistema anti salida de carril.</b> .....	84
<b>3.4.1.</b>	<i>Prueba en pista línea recta</i> .....	85
<b>3.4.2.</b>	<i>Prueba en pista línea curva</i> .....	86
<b>3.5.</b>	<b>Prueba del sistema de visión</b> .....	91
<b>3.6.</b>	<b>Prueba de peso y autonomía del Prototipo.</b> .....	93
<b>3.7.</b>	<b>Análisis de tiempo de carga de la batería del Prototipo.</b> .....	95

### **CAPITULO IV**

<b>4.</b>	<b>EVALUACIÓN ECONÓMICA DEL PROTOTIPO</b> .....	97
<b>4.1.</b>	<b>Presupuesto del prototipo de sistema</b> .....	97
<b>4.2.</b>	<b>Estudio comparativo con otros sistemas y prototipos similares</b> .....	98

<b>CONCLUSIONES</b> .....	99
---------------------------	----

<b>RECOMENDACIONES</b> .....	101
------------------------------	-----

### **GLOSARIO**

### **BIBLIOGRAFIA**

### **ANEXOS**

## INDICE DE TABLAS

<b>Tabla 1-2:</b> Características del Chasis .....	46
<b>Tabla 2-2:</b> Características del Raspberry Pi Módulo de cámara V2 .....	47
<b>Tabla 3-2:</b> Características Arduino UNO .....	49
<b>Tabla 4-2:</b> Características del Raspberry Pi 4.....	50
<b>Tabla 5-2:</b> Consumo de corriente de los dispositivos del prototipo.....	52
<b>Tabla 6-2:</b> Tabla de conexión de terminales y módulos. ....	54
<b>Tabla 1-3:</b> Análisis de la precisión de la red neuronal .....	74
<b>Tabla 2-3:</b> Comparación de precisión alcanzada .....	81
<b>Tabla 3-3:</b> Peso de componentes de prototipo .....	92
<b>Tabla 4-3:</b> Análisis de autonomía de prototipo con peso adicional .....	93
<b>Tabla 5-3:</b> Registro de tiempo de carga en relación con los miliamperios aplicados .....	94
<b>Tabla 1-4:</b> Estudio de costos de componentes requeridos .....	94

## INDICE DE FIGURAS

Figura 1-1: Alto contraste vs bajo contraste.....	10
Figura 2-1: Histograma de bajo contraste vs alto contraste .....	11
Figura 3-1: Desaturación: El tono se vuelve menos dominante, se mueve al centro del círculo .....	13
Figura 4-1: Pasos para el procesamiento digital de imágenes .....	15
Figura 5-1: Diagrama para el procesamiento digital de imágenes .....	16
Figura 6-1: Raspberry pi 4.....	19
Figura 7-1: LeMaker HiKey 960 .....	20
Figura 8-1: Asus Tinker Board.....	21
Figura 9-1: Orange Pi Plus 2.....	22
Figura 10-1: Módulo de cámara V2 .....	23
Figura 11-1: Raspberry Pi Camera Board .....	24
Figura 12-1: Interfaz de Raspbian .....	25
Figura 13-1: Estructura de una red neuronal .....	29
Figura 14-1: Estructura de aprendizaje supervisado .....	30
Figura 15-1: Estructura de aprendizaje reforzado .....	31
Figura 16-1: Estructura de un aprendizaje sin supervisión .....	31
Figura 17-1: Estructura de una red neuronal .....	32
Figura 18-1: Red de alimentación directa de una sola capa .....	33
Figura 19-1: Red de alimentación de varias capas .....	34
Figura 20-1: Red recurrente .....	35
Figura 21-1: Red neuronal convolucional .....	36
Figura 22-1: Capa convolucional .....	38
Figura 1-2: Consolidación general del prototipo .....	41
Figura 2-2: Arquitectura de hardware del prototipo .....	42

<b>Figura 3-2: Diagrama de Bloques de la etapa de obtención de datos .....</b>	<b>43</b>
<b>Figura 4-2: Diagrama de Bloques de la etapa de Procesamiento .....</b>	<b>44</b>
<b>Figura 5-2: Chasis Lamborghini escala de 1 a 10 .....</b>	<b>45</b>
<b>Figura 6-2: Raspberry Pi Camera V2 .....</b>	<b>47</b>
<b>Figura 7-2: Arduino UNO .....</b>	<b>48</b>
<b>Figura 8-2: Raspberry Pi 4.....</b>	<b>49</b>
<b>Figura 9-2: Modulo conversor LM2596.....</b>	<b>51</b>
<b>Figura 10-2: Batería Li-Po Tattu 450mAh .....</b>	<b>51</b>
<b>Figura 11-2: Batería Li-Po R-Line 1300mAh .....</b>	<b>53</b>
<b>Figura 12-2: Diagrama de conexión de los componentes .....</b>	<b>54</b>
<b>Figura 13-2: Chasis del prototipo en su caja .....</b>	<b>55</b>
<b>Figura 14-2: Componentes internos del prototipo .....</b>	<b>56</b>
<b>Figura 15-2: Voltímetro del Regulador de voltaje LM2596.....</b>	<b>56</b>
<b>Figura 16-2: Posicionamiento de elementos electrónicos externos .....</b>	<b>57</b>
<b>Figura 17-2: Posicionamiento Raspberry pi 4.....</b>	<b>58</b>
<b>Figura 18-2: Posicionamiento Arduino UNO .....</b>	<b>59</b>
<b>Figura 19-2: Distribución de los motores .....</b>	<b>59</b>
<b>Figura 20-2: Diagrama de flujo funcionamiento del código .....</b>	<b>62</b>
<b>Figura 21-2: Diagrama de flujo Arduino UNO .....</b>	<b>63</b>
<b>Figura 22-2: Diagrama de flujo Funcionamiento del prototipo .....</b>	<b>65</b>
<b>Figura 23-2: Slot SD card .....</b>	<b>65</b>
<b>Figura 24-2: Descargar raspberry pi imager.....</b>	<b>66</b>
<b>Figura 25-2: Instalación de raspberry pi imager. ....</b>	<b>67</b>
<b>Figura 26-2: Interfaz Raspberry Pi imager. ....</b>	<b>67</b>
<b>Figura 27-2: Interfaz Raspberry pi imager. ....</b>	<b>68</b>
<b>Figura 28-2: Interfaz Raspbian.....</b>	<b>68</b>

<b>Figura 29-2: CNN arquitectura. La red tiene aproximadamente 27 millones de conexiones y 250 mil parámetros. ....</b>	<b>69</b>
<b>Figura 30-2: Código del modelo NVIDIA .....</b>	<b>71</b>
<b>Figura 31-2: Código aplicado para el procesamiento de las imágenes .....</b>	<b>72</b>
<b>Figura 32-2: Ejecución del código de la figura 2-31.....</b>	<b>72</b>
<b>Figura 33-2: Esquema de las partes de una red convolucional.....</b>	<b>73</b>
<b>Figura 34-2: Código de parámetros de la red neuronal convolucional .....</b>	<b>73</b>
<b>Figura 35-2: Épocas y precisión obtenida .....</b>	<b>73</b>
<b>Figura 1-3: Detección de líneas mediante visión artificial .....</b>	<b>77</b>
<b>Figura 2-3: Posición del vehículo toma aérea .....</b>	<b>77</b>
<b>Figura 3-3: Perspectiva Cámara a bordo .....</b>	<b>78</b>
<b>Figura 4-3: Resultado entrenamiento de 300 épocas con 16000 imágenes .....</b>	<b>79</b>
<b>Figura 5-3: Posición del Vehículo toma lateral .....</b>	<b>79</b>
<b>Figura 6-3: Posición vehículo toma frontal .....</b>	<b>80</b>
<b>Figura 7-3: Fiabilidad del sistema, primer analisis .....</b>	<b>81</b>
<b>Figura 8-3: Fiabilidad del sistema, segundo analisis .....</b>	<b>81</b>
<b>Figura 9-3: Fiabilidad del sistema, tercer analisis .....</b>	<b>81</b>
<b>Figura 10-3: Predicciones de CNN .....</b>	<b>81</b>
<b>Figura 11-3: Cámara a bordo prueba en línea recta .....</b>	<b>84</b>
<b>Figura 12-3: interpretación de la CNN de la prueba en línea recta .....</b>	<b>84</b>
<b>Figura 13-3: a. trayectoria del vehículo en curva step 1, b. Comportamiento de la CNN .....</b>	<b>85</b>
<b>Figura 14-3: a. trayectoria del vehículo en curva step 2, b. Comportamiento de la CNN .....</b>	<b>85</b>
<b>Figura 15-3: a. trayectoria del vehículo en curva step 3, b. Comportamiento de la CNN .....</b>	<b>85</b>
<b>Figura 16-3: a. trayectoria del vehículo en curva step 4, b. Comportamiento de la CNN .....</b>	<b>86</b>
<b>Figura 17-3: a. trayectoria del vehículo en curva step 5, b. Comportamiento de la CNN .....</b>	<b>86</b>
<b>Figura 18-3: a. trayectoria del vehículo en curva step 6, b. Comportamiento de la CNN .....</b>	<b>86</b>
<b>Figura 19-3: a. trayectoria del vehículo en curva step 7, b. Comportamiento de la CNN .....</b>	<b>87</b>

<b>Figura 20-3: a. trayectoria del vehículo en curva step 8, b. Comportamiento de la CNN .....</b>	<b>87</b>
<b>Figura 21-3: a. trayectoria del vehículo en curva step 9, b. Comportamiento de la CNN .....</b>	<b>87</b>
<b>Figura 22-3: a. trayectoria del vehículo en curva step 10, b. Comportamiento de la CNN .....</b>	<b>87</b>
<b>Figura 23-3: a. trayectoria del vehículo en curva step 11, b. Comportamiento de la CNN .....</b>	<b>88</b>
<b>Figura 24-3: a. trayectoria del vehículo en curva step 12, b. Comportamiento de la CNN .....</b>	<b>88</b>
<b>Figura 25-3: a. trayectoria del vehículo en curva step 13, b. Comportamiento de la CNN .....</b>	<b>88</b>
<b>Figura 26-3: a. trayectoria del vehículo en curva step 14, b. Comportamiento de la CNN .....</b>	<b>89</b>
<b>Figura 27-3: a. trayectoria del vehículo en curva step 15, b. Comportamiento de la CNN .....</b>	<b>89</b>
<b>Figura 28-3: a. trayectoria del vehículo en curva step 16, b. Comportamiento de la CNN .....</b>	<b>90</b>
<b>Figura 29-3: Comportamiento del vehículo con poca luz .....</b>	<b>90</b>
<b>Figura 30-3: Comportamiento de la CNN con poca luz .....</b>	<b>91</b>
<b>Figura 31-3: Peso del prototipo .....</b>	<b>92</b>
<b>Figura 32-3: Autonomía del Prototipo .....</b>	<b>93</b>

## ÍNDICE DE ABREVIATURAS

<b>3D</b>	Tres Dimensiones
<b>PPP</b>	Pixeles por pulgada
<b>CNN</b>	Convolutional Neural Network
<b>IA</b>	Inteligencia artificial
<b>C++</b>	Lenguaje de programación`
<b>ACC</b>	Adaptative Cruise Control

## **ÍNDICE DE ANEXOS**

**ANEXO A:** DATASHEET MICROCONTROLADORES EMPLEADOS

**ANEXO B:** CODIGO DE CONTROL DE VEHICULO

**ANEXO C:** CODIGO DE PROCESAMIENTO DE IMAGENES

**ANEXO D:** CODIGO RED NEURONAL NVIDIA

**ANEXO E:** CODIGO DE MAESTRO RASPBERRY PI 4

## RESUMEN

El presente trabajo de titulación tiene como objetivo implementar un sistema de visión artificial basado en redes convolucionales para el correcto posicionamiento de un vehículo en un carril compuesto por dos módulos. La primera, de procesamiento de imágenes que son capturadas por un sensor óptico por medio de inteligencia artificial aplicando OpenCV, Tensorflow y Keras. El segundo modulo se encarga del control de los motores y la interpretación de los datos obtenidos por el primer módulo de procesamiento. El sistema fue sometido a pruebas de funcionamiento, determinando que el sistema de detección de carril está dentro de los rangos precisos de interpretación para evitar salirse de los limites trazados en un carril a escala, la precisión de entendimiento de la red neuronal llega a un 98.54%. Se concluye que el prototipo de sistema permite una conducción estable de un vehículo y de interpretación de imágenes en óptimas condiciones de iluminación. Para futuros proyectos e investigaciones se recomienda el uso de lotes de imágenes más grandes y una capacidad de procesamiento más robusta para un aprendizaje más optimo y el aumento de las capacidades autónomas.

**PALABRAS CLAVE:** <APRENDIZAJE AUTOMÁTICO>, < REDES NEURONALES>, <ALGORITMOS GENÉTICOS>, <ALGORITMOS DE INDUCCIÓN>, <APRENDIZAJE PROFUNDO>, <SISTEMAS EXPERTOS>, <REPRESENTACIÓN DEL CONOCIMIENTO>, <COMPUTACIÓN COGNITIVA>, <INTELIGENCIA ARTIFICIAL>.



1444-DBRA-UTP-2021

## **ABSTRACT**

The present degree work aims to implement an artificial vision system based on convolutional networks for the correct positioning of a vehicle in a lane composed of two modules. The first, processing images that are captured by an optical sensor through artificial intelligence applying OpenCV, Tensorflow and Keras. The second module is in charge of the control of the motors and the interpretation of the data obtained by the first processing module. The system was subjected to functional tests, determining that the lane detection system is within the precise ranges of interpretation to avoid leaving the limits drawn on a scale lane, the understanding precision of the neural network reaches a 98.54 %. It is concluded that the prototype system allows a stable driving of a vehicle and image interpretation in optimal lighting conditions. For future projects and research, the use of larger image batches and a more robust processing capacity are recommended for more optimal learning and increased autonomous capabilities.

**KEYWORDS:** <SELF-LEARNING>, <NEURONAL NETWORKS>, <GENETIC ALGORITHMS>, <INDUCTION ALGORITHMS>, <DEEP LEARNING>, <EXPERT SYSTEMS>, <REPRESENTATION OF KNOWLEDGE>, <INTELLIGENCE COMPUTATION>, <COMPUTATION OF KNOWLEDGE>

## INTRODUCCIÓN

La inteligencia artificial (IA) está diseñada para permitir que las computadoras realicen las mismas funciones que la mente humana.

Ciertas habilidades (como el pensamiento) se denominan con frecuencia "inteligentes". Otros (como la visión) no lo son. Sin embargo, todos se basan en habilidades psicológicas (como percepción, asociación, predicción, planificación y control motor) que permiten a los humanos y otros animales lograr sus objetivos. La inteligencia no es una dimensión única, sino un área densamente organizada que rebosa de diferentes habilidades de procesamiento de información. De manera similar, la IA emplea una variedad de formas de realizar una amplia variedad de trabajos. (Boden 2016)

Similar al intelecto humano, la capacidad de abstracción. Para lograr esta hazaña con una máquina, hay que considerar que el cerebro humano es el sistema de reconocimiento más sofisticado y eficiente conocido por el hombre. Los seres humanos ejecutan actos que son inquietantemente similares. Cómo distinguir de oído una cara familiar entre la multitud o el solista en una interpretación de guitarra. Las actividades cognitivas en el cerebro humano se logran mediante la activación coordinada de aproximadamente 90.000.000.000 de células nerviosas unidas a través de conexiones sinápticas. La activación neuronal es un proceso biofísico complicado que se basa en un funcionamiento resiliente y adaptable para ejecutar actividades como el procesamiento de información sensorial, el control fisiológico de los órganos, el lenguaje y la abstracción matemática. (Benítez, R., Escudero, G., Kanaan, S. 2014)

Si bien muchos dirían que se desconoce la historia de los vehículos autónomos, realmente comenzó en 1939, con un automóvil eléctrico controlado por circuitos integrados en el pavimento de la carretera. Sin embargo, uno de los hitos en el desarrollo y evolución de estas unidades se remonta a 1994, cuando dos automóviles circulaban de forma autónoma durante más de 2 kilómetros por una autopista de París. Se demostró que los automóviles son capaces de realizar cambios de carril de forma independiente con una mínima participación humana.

Los vehículos autónomos prometen numerosos beneficios para el tráfico vehicular, incluida una mayor capacidad vial y flujo de tráfico como resultado de tiempos de respuesta más rápidos, menor consumo de combustible y contaminación como resultado de una conducción más progresista y menos accidentes como resultado de los sistemas de prevención de colisiones. Los conductores pueden utilizar el tiempo ahorrado para otras actividades. Para operar de forma segura en el tráfico diario o en situaciones peligrosas, se deben resolver una serie de desafíos de percepción, navegación y control, así como la infraestructura y la legislación pertinente. (Li et al. 2018)

El objetivo de este Trabajo de Fin de Grado es realizar una red neuronal profundas para gestionar la conducción un coche de forma autónoma. Dado que los datos de entrada son imágenes, la red será de naturaleza convolucional, ya que las capas que la componen, su diseño es óptimo para su manejo.

La técnica de trabajo seleccionada es el entrenamiento supervisado, ya que las redes aprenderán qué valores de salida generar cuando se les den ciertos valores de entrada a partir de una colección de datos recopilados manualmente en el mismo simulador (verdad en el terreno), proceso denominado aprendizaje End-to-End. Para comenzar, se construirán modelos de redes neuronales convolucionales (CNN) que regulen el ángulo de dirección del volante solo, mientras se mantiene siempre una velocidad constante del vehículo. Estos modelos se basarán en la CNN producida por NVIDIA en 2016.

Se realizarán una serie de ajustes a la arquitectura subyacente para probar su capacidad a la hora de evitar la salida del carril del prototipo

El material a continuación se divide en tres capítulos, cada uno de los cuales se centra en un aspecto diferente del trabajo creado y cómo contribuye al logro de los objetivos establecidos. El primer capítulo analiza la base teórica de la inteligencia artificial, el procesamiento de imágenes, el software de desarrollo, el hardware de desarrollo, las cámaras y los módulos del sistema de control, así como los lenguajes de programación Python y Arduino. El segundo capítulo establece el marco metodológico y prosigue con la descripción de las necesidades del dispositivo propuesto, así como la selección de herramientas software y componentes hardware esenciales para la ejecución del prototipo. Finalmente, el tercer capítulo discute y analiza los hallazgos de un conjunto exhaustivo de pruebas realizadas en el prototipo final. Las conclusiones se han elaborado de acuerdo con los resultados de las pruebas y las recomendaciones están orientadas a futuros estudios que ayudarán a mejorar el sistema.

# CAPÍTULO I

## 1. MARCO TEÓRICO REFERENCIAL

### 1.1. Visión artificial

Conocida también como visión por computadora, es considerada por muchos como una rama de la inteligencia artificial. La visión artificial es una de las tecnologías que actualmente está teniendo un increíble desarrollo y acogida, tal es así, que sus aplicaciones están abarcando todos los campos que se pueda imaginar: seguridad, control de calidad, robótica, militar, etcétera. (Pérez Grassi, Puente León 2009)

Se puede definir a la visión artificial como el análisis de imágenes mediante computadores para obtener cierta información de los objetos del mundo real que son captados por una cámara la principal finalidad de la visión artificial es dotar de “ojos” las máquinas para que puedan percibir y entender (hasta cierto punto) el mundo que los rodea, mediante una imagen o una secuencia de imágenes, de modo que puedan actuar según convenga en una determinada situación. (Pérez Grassi, Puente León 2009)

En los últimos años, la visión artificial ha experimentado un gran desarrollo y se debe a determinados avances tecnológicos como por ejemplo el desarrollo del hardware. con el paso del tiempo ha sido posible crear procesadores más potentes que requieren menor tiempo para procesar mayor cantidad de información, Asimismo ha sido posible reducir el tamaño de estos, de modo que sea posible incorporarlos en muchos dispositivos electrónicos que existen actualmente. (Pérez Grassi, Puente León 2009)

otro de los avances tecnológicos que ha propiciado el crecimiento de la visión por computador es el desarrollo de software. gracias a que cada vez existen más herramientas informáticas al alcance de todos, conocidas como herramientas de software libre, es posible que los programadores puedan desarrollar sus proyectos y la necesidad de empezar desde cero, sí no qué pueden utilizar herramientas de ayuda para el desarrollo que ya están desarrolladas, llamadas librerías de funciones, mismas que son capaces de realizar determinadas funciones ya definidas, hechos por una comunidad dedicada a un trabajo en específico.(Pérez Grassi, Puente León 2009)

### ***1.1.1. Ventajas***

Los sistemas de visión artificial tienen el enorme potencial de transformar la producción para aumentar el rendimiento, la calidad y la productividad. La inspección automatizada es una de las aplicaciones más comunes para los sistemas de visión artificial, ya que es una aplicación perfectamente adecuada para la automatización de imágenes.

La industria de fabricación de productos electrónicos ha experimentado un auge en las últimas décadas, coincidiendo con un aumento de la tecnología de visión artificial. Las cámaras de visión artificial han revolucionado el proceso de inspección de componentes electrónicos, incluidos los semiconductores.

Las ventajas de la visión por computadora que vienen con el territorio caen bajo una fascinante cantidad de encabezados. Casi todos los sectores, tanto privados como públicos, pueden beneficiarse del uso de computadoras para rastrear, analizar e interpretar el mundo que los rodea. A medida que las organizaciones más poderosas se den cuenta de lo que la visión por computadora y el aprendizaje automático pueden aportar, más veremos cómo esta tecnología de inteligencia artificial afecta nuestras vidas. (Guerra et al 2016)

### ***1.1.2. Aplicaciones***

La tecnología finalmente ha progresado hasta el punto en que el uso masivo de la visión por computadora se ha vuelto viable. Las aplicaciones móviles se han vuelto lo suficientemente poderosas y complejas como para facilitar las aplicaciones de visión por computadora de formas que ya están afectando su vida. La reacción y el aprendizaje en tiempo real han proporcionado un medio para una variedad de nuevas experiencias de usuario, tanto por diversión como por practicidad. Por ejemplo:

#### ***1.1.2.1. Filtros de Snapchat***

Snapchat es uno de los aplicadores más notables de la visión por computadora para entretener a su base de usuarios. Desde orejas de perro hasta coronas de flores y cascadas de arcoíris, Snapchat ofrece una variedad de formas de alterar tu rostro. Por supuesto, esto es posible gracias a la aparición

relativamente reciente de una aplicación de visión por computadora que puede manipular imágenes en tiempo real.

Los filtros funcionan mediante Snapchat analizando tu rostro y, en cuestión de segundos, reconociendo y cuantificando tus características y estructuras. El rostro humano tiene algunos puntos de referencia que proporcionan excelentes puntos de partida para este proceso, incluida la nariz, la boca, los ojos y las cejas. Una vez que se ha trazado su rostro, Snapchat se basa en su aprendizaje profundo para equiparar sus características a un "rostro promedio".

La "cara promedio" es la parte más importante para los filtros en tiempo real, ya que la visión por computadora crea una malla que se superpone con su estructura facial. A partir de ahí, el algoritmo puede reaccionar y manipular su selección de filtros para que se corresponda con cómo cambia su rostro(Xia, Xu, Nan 2017)

#### *1.1.2.2. Amazon Go*

Imagínese una tienda de conveniencia donde el proceso de compra se ha simplificado perfectamente. Entrar, agarrar lo que necesitas y salir sin molestarte nunca con un cajero. Si bien en algún momento pudo haber sido más ciencia ficción que realidad, Amazon ha entregado, a través del poder de la visión por computadora y el aprendizaje automático, exactamente eso.

Amazon Go es una colaboración de la aplicación y la tienda, necesitará una para ingresar a la otra. Utiliza visión artificial para realizar un seguimiento de las existencias, el mantenimiento y todos los clientes de la tienda para garantizar la seguridad y la eficacia. Sus cámaras y sensores, ubicados alrededor de la tienda, detectan y conectan a todos en la tienda a su cuenta de Amazon, al mismo tiempo que mantienen existencias de cada artículo que cada cliente lleva actualmente.

En pocas palabras, es impresionante y solo se puede lograr a través de esta tecnología de IA específica. Tan pronto como haya terminado de comprar, puede salir directamente por la puerta y Amazon cargará automáticamente en su cuenta todo lo que se haya llevado.(Cabrera Marino 2017)

### **1.1.2.3. *Lente de Pinterest***

En lugar de centrarse en el movimiento en tiempo real como lo han hecho Amazon y Snapchat, Pinterest se centra en lo que hace mejor: conectarte con tus intereses. Todo lo que necesitas es tomar una foto de algo que te gusta en el mundo, como un automóvil, una planta o una obra de arte, y Pinterest Lens te dirige de inmediato hacia cualquier cosa inspirada por ese interés.

Como siempre, la tecnología de inteligencia artificial es el componente crítico para que esto funcione, y lo hace a través de una acumulación de visión por computadora de aprendizaje profundo integral. Pinterest no es más que imágenes, un enorme catálogo de información que alimenta e informa su algoritmo. Dicho algoritmo de construye, analiza y luego compara la imagen que tomó con miles de otras en Pinterest y en la web.(Cabrera Marino 2017)

### **1.1.2.4. *Amazon Echo Look***

En lugar de centrarse en funciones de música y audio como otros en la línea de productos Echo de Amazon, Amazon Echo Look está dedicado a la moda. Esto incluye trabajo de cámara activado por voz, consejos de estilo solicitados para sus atuendos y cinematografía detallada para capturar la mejor imagen.

Lo más probable es que pueda ver dónde entra en juego la visión por computadora. Echo Look no solo analiza tus atuendos mientras afecta tu entorno para crear una imagen fotogénica, sino que sus componentes de inteligencia artificial incluso te ayudan a acentuar tu apariencia. También realiza un seguimiento de lo que hay en tu guardarropa, clasifica tu ropa y sugiere lo que puedes comprar en Amazon para completar tu estilo.

El algoritmo Echo Look deriva su conocimiento de aprendizaje profundo para aprovechar las experiencias y los comentarios recopilados de sus consumidores para construir una red más sólida dedicada al diseño y la estilización de moda. Es necesario tener en cuenta numerosos factores para hacerlo bien: tamaño, tono de piel, color, lo que está disponible, etc. La visión por computadora y el aprendizaje automático son los que lo hacen posible.(Cabrera Marino 2017)

#### **1.1.2.5. Control y seguridad vehicular**

El número de vehículos de carretera aumentó significativamente en décadas recientes. Esta tendencia acompañó la construcción de la infraestructura vial y el desarrollo de varios sistemas de control para aumentar la seguridad vial, la capacidad vial y la comodidad del viaje. En la seguridad del tráfico se ha logrado un desarrollo significativo y los sistemas actuales incluyen cada vez más cámaras y métodos de visión por computadora. Las cámaras se utilizan como parte de la infraestructura vial o en vehículos. En este trabajo se hace una revisión de los sistemas de visión por ordenador en vehículos desde el punto de vista de la ingeniería de tráfico. Se presentan los problemas de seguridad de los vehículos de carretera, se describen los sistemas de visión de vehículos de última generación y se discuten los problemas abiertos con las futuras direcciones de investigación.(Cabrera Marino 2017)

#### **1.1.2.6. Robótica**

Los robots se utilizan hoy en día en muchas tareas que adoptan una amplia gama de formas, desde humanos hasta robots que se asemejan a máquinas. Su diseño mecánico (como el número de brazos o la capacidad de navegar) está determinado por la aplicación a la que están destinados. Una cosa es casi común: usan cámaras para manejar tareas complicadas y asegurarse de que la actividad deseada se realice según lo previsto. Se utilizan varios tipos de cámaras, desde cámaras RGB normales, escáneres LIDAR, cámaras de infrarrojos, cámaras de profundidad y más.

Cuando las cámaras están a bordo (o en el brazo del robot), los algoritmos de visión artificial se encargan del análisis de la imagen. Ayudan al robot a "ver" objetos y comprender el entorno en el que se encuentra, de modo que pueda navegar sin tropezar con obstáculos. Algunos de los algoritmos de visión artificial son específicos de la aplicación, mientras que otros son comunes a casi todas las aplicaciones de robótica y robots que utilizan visión artificial.(Bertozzi et al. 2002)

Los robots industriales realizan tareas de montaje y "pick and place". Utilizan algoritmos de visión artificial para localizar y organizar piezas y verificar operaciones como la soldadura. Están contruidos como una máquina de múltiples brazos, donde cada brazo es una estructura de múltiples articulaciones que permite operaciones complicadas.

Los robots que sirven en la agricultura se pueden encontrar en los campos, ayudando a los cultivos en crecimiento y cerca de las líneas de clasificación y clasificación. Los robots que realizan tareas

relacionadas con el campo tienen la forma de una plataforma móvil y utilizan algoritmos de navegación además de algoritmos específicos de cultivo. La clasificación y clasificación generalmente requieren robots estacionarios, que utilizan brazos con una cámara montada en cada brazo. Los algoritmos de visión artificial manejan las tareas de localización y clasificación. Por lo general, aquí se pueden encontrar robots de almacén que transportan paquetes, apoyados por algoritmos de navegación.(Colombiana 2017)

Los robots en el comercio minorista se pueden encontrar en los pasillos de los grandes almacenes y en el almacén. Los primeros, los robots de los pasillos, tienen una forma amigable de humano o muy similar. Realizan tanto la orientación al cliente como la gestión de estanterías. Se utilizan algoritmos de visión artificial como el reconocimiento facial y la clasificación de objetos. Los robots de almacén utilizan algoritmos de detección y navegación de objetos.

Los robots se emplean en la industria de los semiconductores para realizar tareas de recogida y colocación de alta precisión, así como algunas tareas de detección y clasificación de defectos.

Los robots médicos sirven para el seguimiento y la asistencia de los pacientes, mientras que otros realizan cirugías de precisión. Los robots de monitoreo de pacientes tienen una forma humana amigable, como los robots minoristas que atienden a los clientes en los pasillos. Utilizan el mismo conjunto de algoritmos de visión artificial. Los robots quirúrgicos ayudan a los cirujanos en cirugías complejas y utilizan algoritmos de visión artificial para navegar dentro de la TC del paciente o simular la cirugía como en los sistemas de realidad virtual.(Guerra et al 2016)

### **1.3. Definiciones**

#### **1.3.1 Píxel**

Un píxel está representado por un punto o un cuadrado en la pantalla de un monitor de computadora. Los píxeles son los componentes básicos de una imagen o pantalla digital y se crean utilizando coordenadas geométricas.

Dependiendo de la tarjeta gráfica y el monitor de visualización, la cantidad, el tamaño y la combinación de colores de los píxeles varía y se mide en términos de la resolución de la pantalla. Por ejemplo, una computadora con una resolución de pantalla de 1280 x 768 producirá un máximo de 98,3040 píxeles en una pantalla.(Duarte et al. 2008)

La extensión de la resolución de píxeles también determina la calidad de visualización; más píxeles por pulgada de la pantalla del monitor produce mejores resultados de imagen. Por ejemplo, una imagen de 2,1 megapíxeles contiene 2.073.600 píxeles, ya que tiene una resolución de 1920 x 1080.

El tamaño físico de un píxel varía, dependiendo de la resolución de la pantalla. Será igual al tamaño de la distancia entre puntos si la pantalla está configurada en su resolución máxima y será más grande si la resolución es menor, ya que cada píxel utilizará más puntos. Debido a eso, los píxeles individuales pueden volverse visibles, dando lugar a una imagen en bloque y gruesa definida como "pixelada".

Los píxeles están dispuestos uniformemente en una cuadrícula bidimensional, aunque se encuentran disponibles algunos patrones de muestreo diferentes. Por ejemplo, en las pantallas LCD, los tres colores principales se muestrean en diferentes ubicaciones de una cuadrícula escalonada, mientras que las cámaras digitales en color utilizan una cuadrícula más regular.

En los monitores de computadora, los píxeles tienen forma cuadrada, lo que significa que sus tonos de muestreo verticales y horizontales son iguales. En otros sistemas, como el formato de pantalla ancha anamórfica del estándar de vídeo digital 601, la forma de un píxel es rectangular.

Cada píxel tiene una dirección lógica única, un tamaño de ocho bits o más y, en la mayoría de los dispositivos de visualización de alta gama, la capacidad de proyectar millones de colores diferentes. El color de cada píxel está determinado por la combinación específica de los tres componentes principales del espectro de colores RGB. (Duarte et al. 2008)

### **1.3.2. Imagen digital**

Las imágenes digitales están formadas por elementos de imagen llamados píxeles. Normalmente, los píxeles se organizan en una matriz rectangular ordenada. El tamaño de una imagen está determinado por las dimensiones de esta matriz de píxeles. El ancho de la imagen es el número de columnas y la altura de la imagen es el número de filas de la matriz. Por tanto, la matriz de píxeles es una matriz de M columnas x N filas. Para hacer referencia a un píxel específico dentro de la matriz de la imagen, definimos su coordenada en  $xy$ . El sistema de coordenadas de matrices de imágenes define  $x$  como creciente de izquierda a derecha e  $y$  como creciente de arriba a abajo. En comparación con la convención matemática normal, el origen está en la esquina superior izquierda y la coordenada  $y$  está invertida. ¿Por qué el sistema de coordenadas se invierte verticalmente? Originalmente, las imágenes digitales se definían en términos del patrón de barrido del haz de electrones de los televisores. El rayo

escaneó de izquierda a derecha y de arriba a abajo. Aparte de esta razón histórica, esta inversión de la coordenada y no sirve para nada. (Superiores, Producto 2016)

El tamaño de la imagen no debe confundirse con el tamaño de la representación del mundo real de una imagen. El tamaño de la imagen describe específicamente la cantidad de píxeles dentro de una imagen digital. La representación del mundo real de una imagen digital requiere un factor adicional llamado resolución. La resolución es la escala espacial de los píxeles de la imagen. Por ejemplo, una imagen de 3300x2550 píxeles con una resolución de 300 píxeles por pulgada (ppi) sería un tamaño de imagen del mundo real de 11 "x 8.5". Para aclarar los términos de resolución, ppp son píxeles por pulgada y ppp son puntos por pulgada. Ppi se refiere a matrices de píxeles, mientras que ppp se refiere a la resolución de la impresora. En realidad, estos dos términos de resolución se utilizan indistintamente. Otro término de resolución que puede encontrar es lpi, para líneas por pulgada, que describe la resolución de medios tonos y se utiliza en la impresión de revistas y periódicos. Muchas aplicaciones de edición de imágenes tienen una resolución predeterminada de 72 PPP. Esto también es válido para guardar imágenes JPG en Imagen. (Superiores, Producto 2016)

Habiendo definido el número de píxeles, MxN, solo proporciona una forma rectangular para nuestra imagen. Se necesita un parámetro más, la intensidad, para definir verdaderamente una imagen. Cada píxel tiene su propio valor de intensidad o brillo. Si todos los píxeles tienen el mismo valor, la imagen tendrá un tono uniforme; todo negro, blanco, gris o algún otro tono. Es en el tipo de intensidad utilizada para cada píxel donde varían los tipos de imágenes. Las imágenes en blanco y negro solo tienen una intensidad desde el gris más oscuro (negro) hasta el gris más claro (blanco). Las imágenes en color, por otro lado, tienen la intensidad del más oscuro y más claro de tres colores diferentes, rojo, verde y azul. Las diversas mezclas de estas intensidades de color producen una imagen en color. Por lo tanto, los dos tipos más básicos de imágenes digitales, B / N y Color, se conocen como imágenes en escala de grises e imágenes RGB. Además del tipo de intensidad de cada píxel, también varía el rango de valores de intensidad.

Los valores de intensidad en imágenes digitales se definen mediante bits. Un bit es binario y solo tiene dos valores posibles, 0 o 1. Un rango de intensidad de 8 bits tiene 256 valores posibles, de 0 a 255. Esto se puede ver matemáticamente por  $2^8$  (# de bits). Para una imagen de 1 bit o binaria,  $2^1 = 2$  valores posibles y para una imagen de 8 bits,  $2^8 = 256$  valores posibles. La fotografía digital estándar utiliza un rango de valores de 8 bits; Las imágenes RGB utilizan rangos de intensidad de 8 bits para cada color y las imágenes en blanco y negro tienen un solo rango de intensidad de 8 bits. Dado que las imágenes RGB contienen intensidades de  $3 \times 8$  bits, también se denominan imágenes en color de 24 bits. Hasta ahora solo hemos discutido el rango de valores posibles y no el intervalo entre valores.

Teóricamente, un rango de 8 bits podría ocupar valores de 0 a 1 usando incrementos de 1/256, pero en realidad, las imágenes de 8 bits están definidas para usar solo valores enteros (números enteros) de 0 a 255. (Superiores, Producto 2016)

### **1.3.3. Resolución**

La resolución es el número de píxeles (puntos de color individuales) contenidos en un monitor de visualización, expresado en términos del número de píxeles en el eje horizontal y el número en el eje vertical. La nitidez de la imagen en una pantalla depende de la resolución y el tamaño del monitor. La misma resolución de píxeles será más nítida en un monitor más pequeño y perderá gradualmente nitidez en monitores más grandes porque el mismo número de píxeles se distribuye en una mayor cantidad de pulgadas. (García Elorriaga 2015)

Un sistema de visualización de computadora determinado tendrá una resolución máxima que depende de su capacidad física para enfocar la luz (en cuyo caso el tamaño físico del punto, el tamaño del punto coincide con el tamaño del píxel) y, por lo general, varias resoluciones menores. Por ejemplo, un sistema de visualización que admita una resolución máxima de 1280 por 1023 píxeles también puede admitir resoluciones de 1024 por 768, 800 por 600 y 640 por 480. Tenga en cuenta que, en un monitor de tamaño determinado, la resolución máxima puede ofrecer una imagen más nítida, pero se puede distribuir en un espacio demasiado pequeño para leer bien.

La resolución de la pantalla no se mide en puntos por pulgada como suele ocurrir con las impresoras. Sin embargo, la resolución y el tamaño del monitor físico juntos le permiten determinar los píxeles por pulgada. Normalmente, los monitores de PC tienen entre 50 y 100 píxeles por pulgada. Por ejemplo, un monitor VGA de 15 pulgadas (ver modos de visualización) tiene una resolución de 640 píxeles a lo largo de una línea horizontal de 12 pulgadas o aproximadamente 53 píxeles por pulgada. Una pantalla VGA más pequeña tendría más píxeles por pulgada. (García Elorriaga 2015)

### **1.3.4. Brillo**

El brillo es la percepción de cuán intensa es la luz que proviene de una pantalla. Aunque es subjetivo, se puede medir en términos de la amplitud de la luz que emite una pantalla.

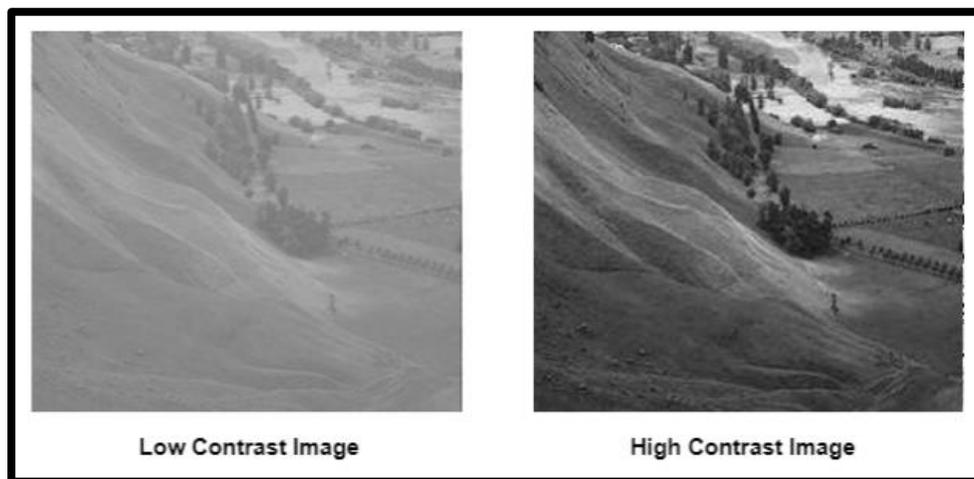
Las pantallas de color usan tres colores bajo el esquema RGB: rojo, verde y azul. El brillo de una pantalla se puede representar mediante la suma de la amplitud de los píxeles rojos, verdes y azules dividida por tres. En otras palabras, el promedio de los tres colores.

La percepción del brillo puede verse alterada por ciertas ilusiones ópticas para que parezcan más brillantes u oscuras. A medida que disminuye el brillo, los colores aparecen descoloridos, pero "más puros" a medida que aumenta el brillo.

En los dispositivos móviles, las configuraciones de brillo más altas pueden hacer que la batería del dispositivo se agote más rápido en comparación con una configuración más baja. Muchos dispositivos móviles tienen un sensor de luz ambiental para ajustar automáticamente el brillo a un nivel cómodo según la luz de una habitación. (Moles 2015)

#### 1.3.5. *Contraste*

El contraste es la diferencia de luminancia o color que hace que un objeto se distinga de otros objetos dentro del mismo campo de visión, como se observa en la figura 1-1.



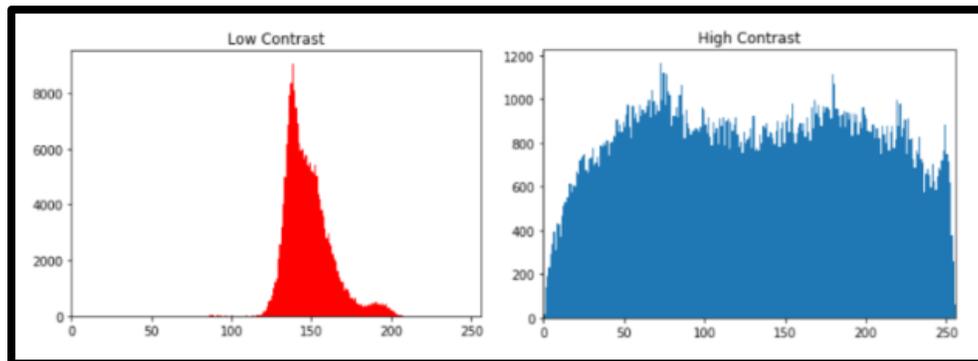
**Figura 1-1.** Alto contraste vs Bajo contraste

Fuente: Moles. 2015

Claramente, la imagen de la izquierda tiene un contraste bajo porque es difícil identificar los detalles presentes en la imagen en comparación con la imagen de la derecha.

Un ejemplo de la vida real puede ser un día soleado y con niebla. En un día soleado, todo nos parece claro, por lo tanto, tiene un alto contraste, en comparación con un día con niebla, donde todo parece casi de la misma intensidad (aspecto gris apagado y descolorido).

Una forma más válida de comprobar si una imagen tiene un contraste alto o bajo es trazar el histograma de la imagen. Tracemos el histograma de las imágenes anteriores, como se observa en la figura 2-1.



**Figura 2-1.** Histograma de Alto contraste vs Bajo contraste

Fuente: Moles. 2015

Claramente, en el histograma de la imagen de la izquierda, podemos ver que los valores de intensidad de la imagen se encuentran en un rango estrecho. Debido a que es difícil distinguir casi los mismos valores de intensidad (vea la figura siguiente, 150 y 148 son difíciles de distinguir en comparación con 50 y 200), por lo tanto, la imagen de la izquierda tiene un contraste bajo.

El histograma de la derecha aumenta esta brecha entre los valores de intensidad, los detalles de la imagen ahora son mucho más perceptibles para nosotros y, por lo tanto, produce una imagen de alto contraste.

Entonces, para el alto contraste, el histograma de la imagen debe abarcar todo el rango dinámico como se muestra arriba en el histograma de la derecha. En los próximos blogs, aprenderemos diferentes métodos para hacer esto.

Hay otro enfoque ingenuo en el que restamos los valores de intensidad máxima y mínima y, en función de esta diferencia, juzgamos el contraste de la imagen. No recomendaré seguir esto ya que

esto puede verse afectado por los valores atípicos (lo discutiremos en los próximos blogs). Por lo tanto, siempre trace el histograma para comprobarlo.

Las imágenes de bajo contraste pueden deberse a una iluminación deficiente, falta de rango dinámico en el sensor de imagen o incluso un ajuste incorrecto de la apertura de la lente durante la adquisición de imágenes, etc.(Moles 2015)

### **1.3.6. Tono**

En el arte, el término "tono" describe la calidad del color. Tiene que ver con si un color se percibe como cálido o frío, brillante o apagado, claro u oscuro, puro o "sucio". El tono de una obra de arte puede tener una variedad de efectos, desde establecer el ambiente hasta agregar énfasis.

Lo más probable es que hayas escuchado la frase "bajar el tono". En el arte, esto significa hacer que un color (o una combinación de colores general) sea menos vibrante. Por el contrario, "tonificarlo" puede significar hacer que los colores salgan de una pieza, a veces en una medida bastante sorprendente. Sin embargo, el tono en el arte va mucho más allá de esta simple analogía.

"Tono" es otra palabra para "valor", que es uno de los elementos centrales del arte. A veces usamos la frase valor tonal, aunque también se puede usar el tono. No importa cómo lo llames, todos significan lo mismo: la claridad u oscuridad de un color.(Nancy 2006)

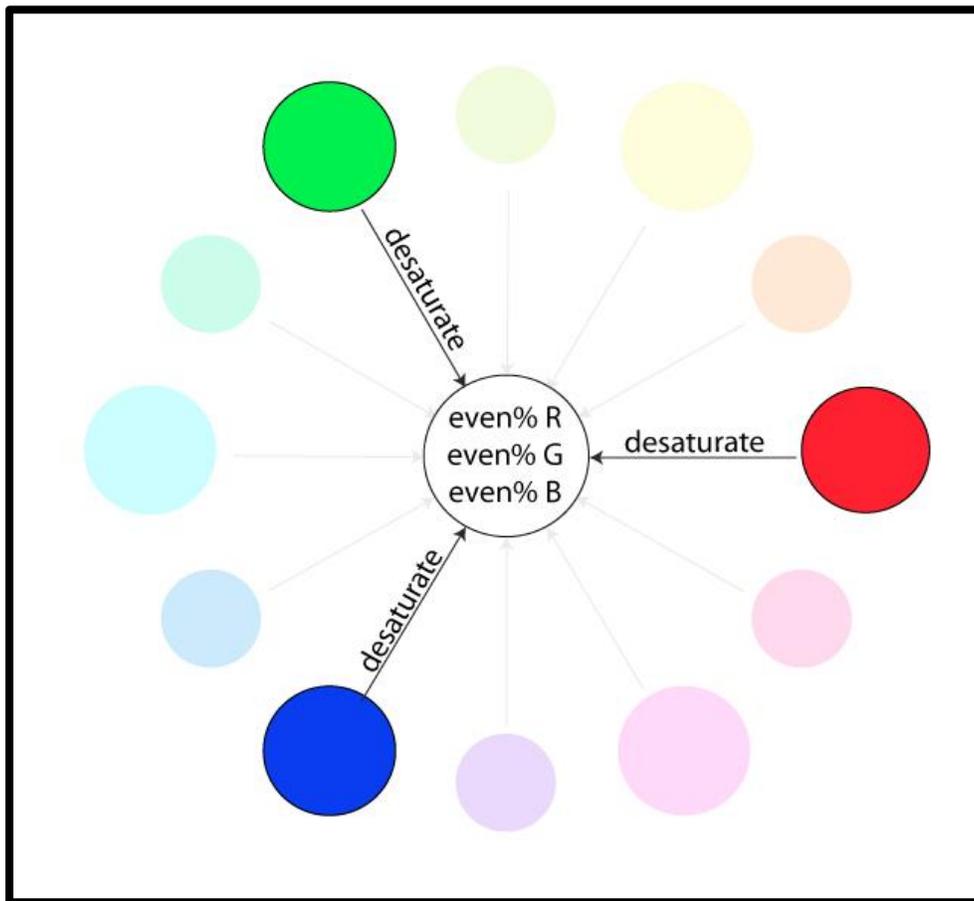
Una variedad de tonos se encuentra en todo lo que nos rodea. El cielo, por ejemplo, no es de un tono azul sólido. En cambio, es una variedad de tonos azules que forman un degradado de claro a oscuro.

Incluso un objeto que sea de un color sólido, como un sofá de cuero marrón, tendrá tonos cuando lo pintemos o fotografiemos. En este caso, los tonos se crean por la forma en que la luz incide sobre el objeto. Las sombras y las luces le dan dimensión, incluso si en realidad es un color uniforme.(Moles 2015)

### **1.3.7. Saturación**

La saturación también se conoce como "intensidad" y "croma". Se refiere al predominio del tono en el color. En el borde exterior de la rueda de tonos están los tonos "puros". A medida que se mueve hacia el centro de la rueda, el tono que estamos usando para describir el color domina cada vez menos.

Cuando llegas al centro de la rueda, no domina ningún tono. Estos colores directamente en el eje central se consideran desaturados, como se observa en la figura 3-1.



**Figura 3-1.** Desaturación: El tono se vuelve menos dominante, se mueve al centro del círculo  
**Fuente:** Moles. 2015

Naturalmente, lo opuesto a la imagen de arriba es saturar el color. El primer ejemplo a continuación describe la dirección general que el color debe moverse en el círculo de color para saturarse más (hacia el exterior). El segundo ejemplo muestra cómo un solo color se ve completamente saturado, sin otros matices presentes en el color. (Moles 2015)

### 1.3.8. *Color*

El color es el aspecto de las cosas causado por las diferentes cualidades de la luz que reflejan o emiten.

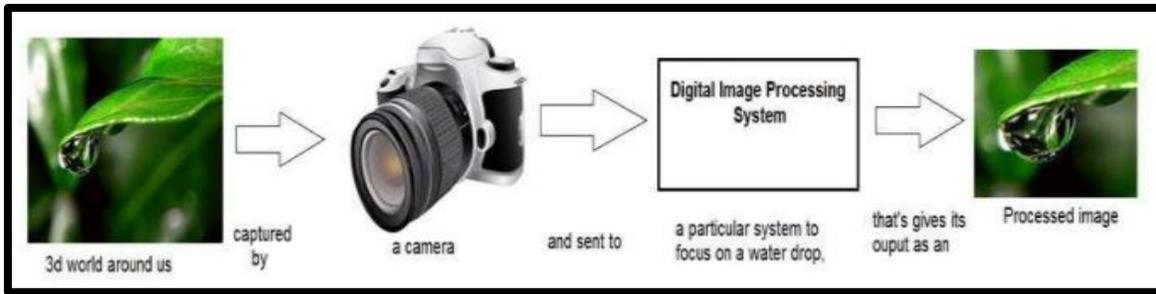
Para ver el color hay que tener luz. Cuando la luz incide sobre un objeto, algunos colores rebotan en el objeto y otros son absorbidos por él. Nuestros ojos solo ven los colores que rebotan o se reflejan.

Los rayos del sol contienen todos los colores del arco iris mezclados. Esta mezcla se conoce como luz blanca. Cuando la luz blanca incide en un crayón blanco o en un cilindro de marcador, nos parece blanco porque no absorbe ningún color y refleja todos los colores por igual. Una tapa de crayón o marcador negro absorbe todos los colores por igual y no refleja ninguno, por lo que nos parece negro. Mientras que los artistas consideran el negro como un color, los científicos no lo hacen porque el negro es la ausencia de todo color.

Todos los rayos de luz contienen color. La luz está hecha de ondas electromagnéticas. Estas ondas se propagan desde cualquier fuente de luz, como el sol. Las ondas de luz viajan a una velocidad tremenda (186.000 millas o 300.000 kilómetros por segundo). Los diferentes colores tienen diferentes longitudes de onda, que es la distancia entre las partes correspondientes de dos de las ondas. La longitud de onda de luz más larga que los humanos pueden ver es la roja. El más corto es el violeta. El ultravioleta tiene una longitud de onda aún más corta, pero los humanos no pueden verlo. Algunas aves y abejas pueden ver la luz ultravioleta. El infrarrojo tiene una longitud de onda más larga que la luz roja, y los humanos no pueden ver esta luz, pero pueden sentir el calor que genera el infrarrojo. (Nancy 2006)

### 1.4. **Procesamiento digital de imágenes**

El procesamiento de imágenes digitales se ocupa de la manipulación de imágenes digitales a través de una computadora digital. Es un subcampo de señales y sistemas, pero se centra especialmente en las imágenes. DIP se centra en el desarrollo de un sistema informático capaz de procesar una imagen. La entrada de ese sistema es una imagen digital y el sistema procesa esa imagen utilizando algoritmos eficientes y da una imagen como salida. El ejemplo más común es Adobe Photoshop. Es una de las aplicaciones más utilizadas para procesar imágenes digitales, como se observa en la figura 4-1.



**Figura 4-1.** Pasos para el procesamiento digital de imágenes

**Fuente:** Guerra et al, 2016

En la figura anterior, una imagen ha sido capturada por una cámara y ha sido enviada a un sistema digital para eliminar todos los demás detalles, y simplemente enfocar la gota de agua haciendo zoom de tal manera que la calidad de la imagen sigue siendo la misma. (Guerra et al 2016)

#### 1.4.1. *Problemas al analizar Imágenes*

Si está utilizando un programa de sistemas de visión artificial, es probable que surjan problemas a medida que se realizan las imágenes ópticas.

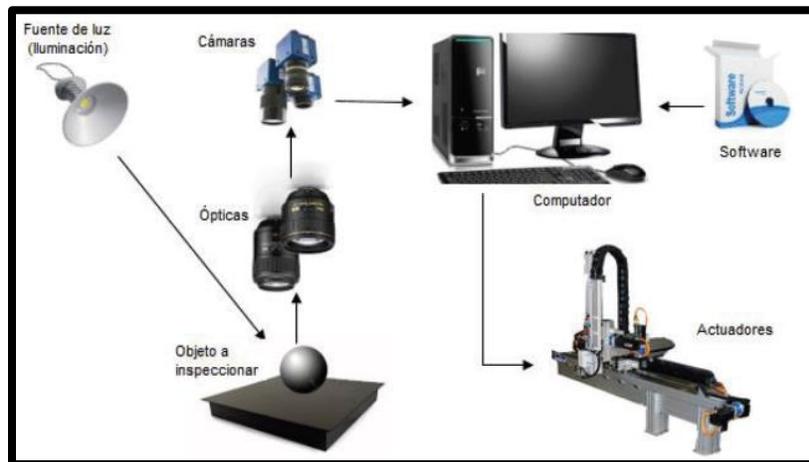
Las limitaciones o desafíos que deben resolverse para las aplicaciones industriales son mínimos cuando se ejecutan en un entorno regulado. Por otro lado, existen importantes desventajas con respecto a las aplicaciones que tienen que funcionar en un entorno intenso donde ocurren muchas variables.

Cuando se identifica un solo objeto en una imagen, hay muchas características de la imagen que pueden complicar esta tarea. Tal es la condición de la posición del objeto dentro de la escena según la ubicación de la cámara en el momento en que captura la toma, en el objeto de interés se puede ver en una ubicación determinada o con la posición girada, o incluso se puede ver desde mayor a menor escala. (Pérez Grassi, Puente León 2009)

Una imagen del objeto que se quiere detectar tampoco lo es siempre, pero está rodeada de un mundo de otros objetos en los que a veces no se tiene la menor confianza y que pueden quedar eclipsados, debido a estos escollos la tarea de analizar procesando una imagen, para detectar un determinado objeto en ella, puede conducir a complicaciones.

Otra gran desventaja es que el objeto a detectar no tiene la misma geometría, como es el caso de los seres vivos que, por su movimiento, pueden aparecer en diferentes lugares, dificultando su identificación.(Nancy 2006)

#### 1.4.2. Elementos de un sistema de visión artificial



**Figura 5-1.** Diagrama para el procesamiento digital de imágenes

Realizado por: Yandry Jácome,2021

Un sistema de visión Artificial se compone básicamente de los siguientes elementos:

##### 1.4.2.1. Cámara digital

Una cámara digital es un dispositivo de hardware que toma fotografías y almacena la imagen como datos en una tarjeta de memoria. A diferencia de una cámara analógica, que expone los productos químicos de la película a la luz, una cámara digital utiliza componentes ópticos digitales para registrar la intensidad y el color de la luz y la convierte en datos de píxeles. Muchas cámaras digitales son capaces de grabar videos además de tomar fotografías.(Calderón et al. 2008)

#### *1.4.2.2. Fuente de luz*

Una fuente de luz emite radiación electromagnética. Algunas fuentes con las que te encuentras todos los días son el sol, las bombillas y las microondas. Las fuentes de luz que ofrece StellarNet se dividen en tres categorías: láseres, lámparas y LED.(Calderón et al. 2008)

#### *1.4.2.3. Óptica*

La óptica tiene que ver con la luz: cómo se genera, propaga y detecta. Es un esfuerzo multidisciplinario con sus raíces en la física, la ingeniería eléctrica, la química y la ciencia de los materiales.

Las aplicaciones ópticas se pueden encontrar en todos los aspectos de nuestras vidas, desde lentes de contacto hasta comunicación por fibra óptica. El estudio de la óptica ha llevado a los científicos a producir inventos innovadores como el láser y el holograma.

#### *1.4.2.4. Software*

Para que o el sistema de visión artificial funcione, se requiere de un algoritmo de programación en este caso es desarrollado en Python en donde se ejecuta el sistema. generalmente el software está construido por una parte que es la interfaz hombre máquina y otra que es la lógica de análisis y control. el día de hoy existen librerías de software libre dedicada específicamente al procesamiento de imágenes que nos facilitan la aplicación de éstas, se han reducido la complejidad en el desarrollo de estos dos morimos ya que el desarrollador tiene una plantilla en donde hace el uso de sus funciones que ya están establecidas en estas librerías(Xing et al. 2018)

#### *1.4.2.5. Actuadores*

Un actuador es un componente de una máquina que se encarga de mover y controlar un mecanismo o sistema, por ejemplo, abriendo una válvula. En términos simples, es un "motor".

Un actuador requiere una señal de control y una fuente de energía. La señal de control es de energía relativamente baja y puede ser voltaje o corriente eléctrica, presión neumática o hidráulica, o incluso

energía humana. Su principal fuente de energía puede ser una corriente eléctrica, presión de fluido hidráulico o presión neumática. Cuando recibe una señal de control, un actuador responde convirtiendo la energía de la fuente en movimiento mecánico.

Un actuador es un mecanismo por el cual un sistema de control actúa sobre un entorno. El sistema de control puede ser simple (un sistema mecánico o electrónico fijo), basado en software (por ejemplo, un controlador de impresora, sistema de control de robot), humano o cualquier otra entrada. (Xing et al. 2018)

## 1.5. Ordenador

Es una máquina programable que responde a un conjunto específico de instrucciones de una manera definida, sí puede ejecutar un sinnúmero de instrucciones programadas.

### 1.5.1. *Raspberry Pi 4*

Raspberry Pi 4 es una actualización mayor de lo que podemos ver a primera vista, el cambio de procesador a un ARM Cortex-172 con cuatro núcleos a 1,5 GHz también implicaba pasar de los 40 nm a los 28 nm. En consecuencia, todos los componentes y la potencia del dispositivo han cambiado. (Chandana, Jilani, Hussain 2015)



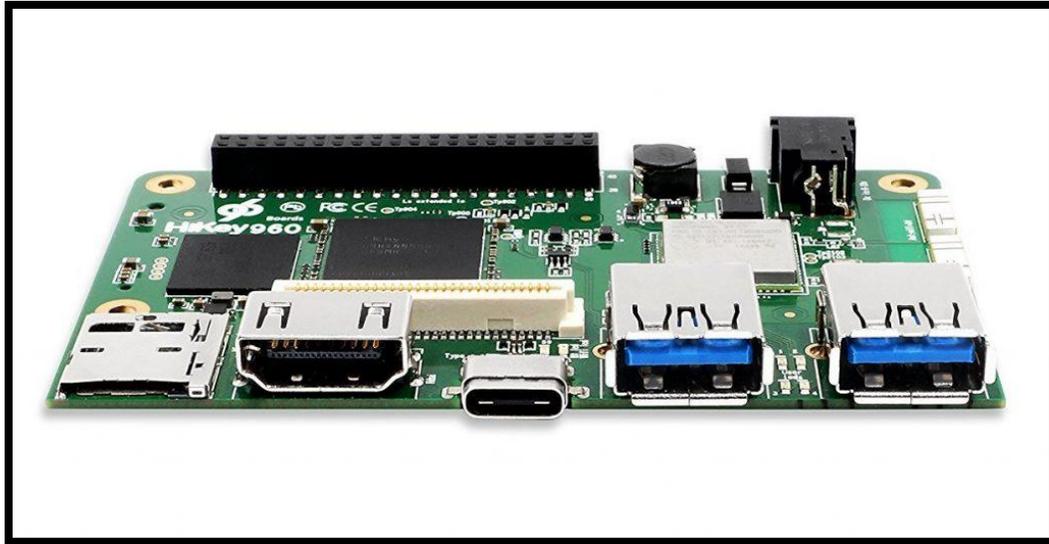
**Figura 6-1.** Raspberry pi 4

**Fuente:** Raspberry. 2020

Además de mejorar su potencia, un cambio interesante y a tener en cuenta en esta nueva Raspberry Pi 4 es la puesta al día de sus conexiones. Raspberry Pi 4 viene con Bluetooth 5.0 y Wi-Fi 802.11ac para las conexiones inalámbricas. También se ha cambiado el conector micro USB de alimentación por un USB-C que suma 500 mA extra de energía para alcanzar un total de 1.2 A, como se observa en la figura 6-1.

#### ***1.5.2. LeMaker HiKey 960***

Su capacidad de almacenamiento se ubica en los 32 GB pudiendo ampliar esta cantidad vía microSD. Entre las conexiones de las que dispone se encuentran 2 puertos USB 3.0, 1 USB OTG tipo C, 1 puerto HDMI y 1 Slot microSD.



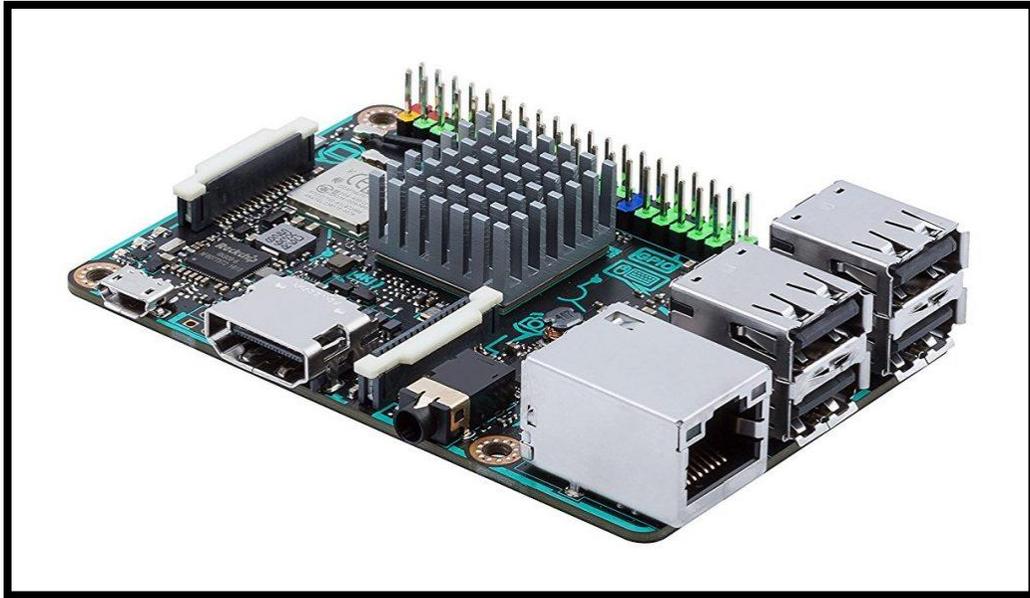
**Figura 7-1.** LeMaker HiKey 960

**Fuente:** LeMaker HiKey. 2020

Esta placa cuenta con un procesador de 8 núcleos Kirin 960 el cual vendrá acompañado de una memoria RAM de 3 GB. Correrá a una velocidad máxima de 2.4 GHz, con lo que la potencia no será un problema. También incorpora un módulo WiFi y Bluetooth 4.1, como se observa en la figura 7-1.(Kemenyash, Omelchenko, Hotsyanivskyy 2018)

### ***1.5.3. Asus Tinker Board***

Incorpora 1 puerto micro USB, 1 puerto HDMI, 1 puerto Ethernet y 4 puertos USB 2.0 entre otras opciones. También dispone de Wifi y Bluetooth 4.0 para completar las conexiones inalámbricas, como se observa en la figura 8-1.(Gómez Rodríguez et al. 2018)



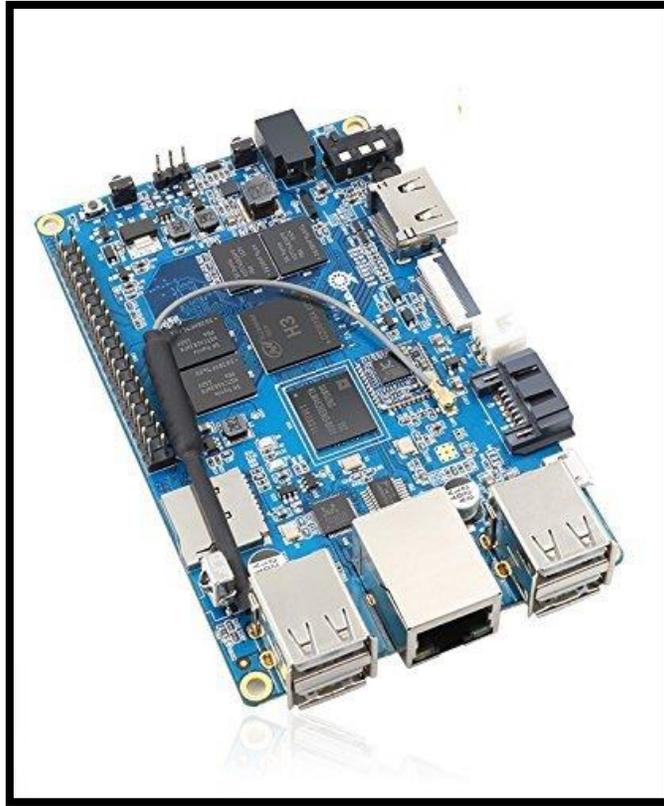
**Figura 8-1.** Asus Tinker Board.

**Fuente:** Asus Tinker Board. 2019

#### *1.5.4 Orange Pi Plus 2*

Podríamos decir que esta es la alternativa que más renombre ha tenido si hablamos de la competencia directa de la Raspberry Pi.

Sus conexiones pasan por 4 puertos USB 2.0, 1 puerto micro USB OTG, 1 puerto HDMI, 1 puerto Ethernet, 1 receptor IR, 1 micrófono y 1 entrada de audio y vídeo, como se observa en la figura 9-1.(PARA, ACERO 2011)



**Figura 9-1.** Orange Pi Plus 2

Fuente: Orange Pi Plus. 2020

### ***1.5.5. La mejor opción***

La elegida es la raspberry pi 4 de 8gb de RAM por su amplia aplicación, cabe recalcar que el coste está en el promedio de su competencia así que el factor más importante para su elección es la disponibilidad de esta en el país.

En términos de especificaciones todas tienen componentes muy similares.

## 1.6. Módulos de Cámaras Raspberry

### 1.6.1 Raspberry Pi Módulo de cámara V2 – 8 megapíxeles, 1080p.

La Raspberry Pi Camera v2 es un sensor de imagen de 8 megapíxeles Sony IMX219 de alta calidad diseñado a medida para Raspberry Pi, con un objetivo de enfoque fijo. Es capaz de imágenes estáticas de 3280 x 2464 píxeles, y también soporta vídeo 1080p30, 720p60 y 640x480p60/90. Se conecta a Pi por medio de uno de los pequeños zócalos en la superficie superior de la placa y utiliza la interfaz CSI dedicada, diseñada especialmente para interactuar con cámaras. La tabla en sí es pequeña, alrededor de 0.984 in x 0.906 in x 0.354 in. También pesa poco más de 0.11 oz, por lo que es perfecto para móviles u otras aplicaciones donde el tamaño y el peso son importantes. Se conecta a Raspberry Pi a través de un cable corto de cinta. El sensor de imagen IMX219 de alta calidad de Sony tiene una resolución nativa de 8 megapíxeles y tiene un objetivo de enfoque fijo a bordo. En términos de imágenes fijas, la cámara es capaz de imágenes estáticas de 3280 x 2464 píxeles, y también soporta vídeo 1080p30, 720p60 y 640x480p90, como se observa en la figura 10-1.(Pi 2015)



**Figura 10-1.** Módulo de cámara V2

**Fuente:** Raspberry. 2020

### 1.6.2. *Raspberry Pi Camera Board - Night Vision "IR-CUT" (5MP)*

La cámara de visión nocturna Raspberry Pi utiliza la lente Fisheye (Wide Angle) de 175 grados para tomar una imagen o vídeo de gran angular. Está equipado con resistencia fotosensible y IR-CUT incorporado, lo que permite a la cámara detectar y reconocer automáticamente la luz y cambiar automáticamente entre la visión nocturna y el modo de disparo diurno. La ventaja es que las fotografías tomadas no serán rojizas durante el día, lo que también elimina la necesidad de conmutación manual. Además, cuenta con dos proyectores LED infrarrojos de alta intensidad para la grabación nocturna, Así que básicamente, lo tenemos cubierto para sus necesidades de 24 x 7.

Esta cámara utiliza el sensor OV5647 Raspberry Pi visión nocturna de serie, y por lo tanto es capaz de proporcionar una resolución de imagen cristalina de 5MP o grabación HD 1080p a 30fps. El vídeo es una imagen de muy alta calidad, La distancia focal de la cámara es de 2,0 apertura (F) y 1,7 mm, por lo que es adecuada para el seguimiento, como se observa en la figura 11-1.(Pi 2015)



**Figura 11-1.** Raspberry Pi Camera Board

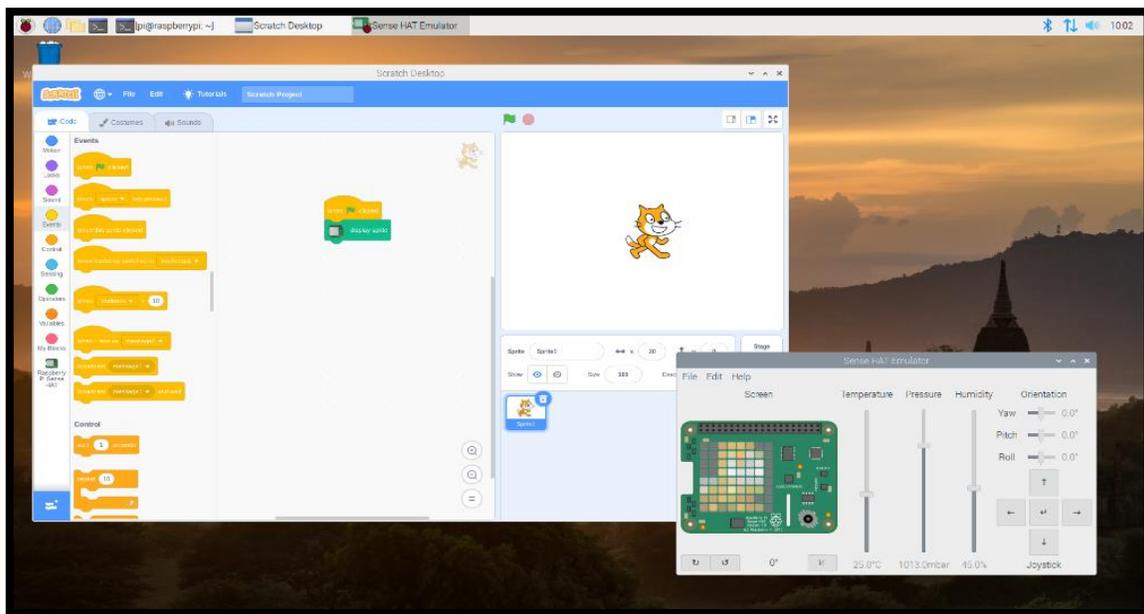
**Fuente:** Raspberry, 2020

## 1.7. Software usado en el desarrollo del proyecto

### 1.7.1 Raspbian

Raspbian es un sistema operativo gratuito basado en Debian optimizado para el hardware Raspberry Pi. Un sistema operativo es el conjunto de programas y utilidades básicos que hacen que su Raspberry Pi funcione. Sin embargo, Raspbian ofrece más que un sistema operativo puro: viene con más de 35,000 paquetes, software pre compilado incluido en un formato agradable para una fácil instalación en su Raspberry Pi.

La compilación inicial de más de 35.000 paquetes Raspbian, optimizados para el mejor rendimiento en Raspberry Pi, se completó en junio de 2012. Sin embargo, Raspbian todavía está en desarrollo activo con énfasis en mejorar la estabilidad y el rendimiento de tantos paquetes Debian como sea posible, como se observa en la figura 12-1.(Singh et al. 2017)



**Figura 12-1.** Interfaz de Raspbian

Fuente: Raspberry. 2020

### 1.7.2 *OpenCV*

OpenCV (Open Source Computer Vision Library) es una biblioteca de software de visión artificial y aprendizaje automático de código abierto. OpenCV se creó para proporcionar una infraestructura común para aplicaciones de visión por computadora y para acelerar el uso de la percepción de la máquina en los productos comerciales. Al ser un producto con licencia BSD, OpenCV facilita que las empresas utilicen y modifiquen el código.(Bradski, Kaehler 2008)

La biblioteca tiene más de 2500 algoritmos optimizados, que incluye un conjunto completo de algoritmos de aprendizaje automático y visión por computadora clásicos y de última generación. Estos algoritmos pueden usarse para detectar y reconocer rostros, identificar objetos, clasificar acciones humanas en videos, rastrear movimientos de cámara, rastrear objetos en movimiento, extraer modelos 3D de objetos, producir nubes de puntos 3D a partir de cámaras estéreo, unir imágenes para producir una alta resolución imagen de una escena completa, buscar imágenes similares de una base de datos de imágenes, eliminar ojos rojos de imágenes tomadas con flash, seguir los movimientos oculares, reconocer paisajes y establecer marcadores para superponerlos con realidad aumentada, etc. OpenCV tiene más de 47 mil personas de usuarios comunidad y un número estimado de descargas superior a 18 millones. La biblioteca se utiliza ampliamente en empresas, grupos de investigación y organismos gubernamentales.(Bradski, G., & Kaehler 2008)

Junto con empresas bien establecidas como Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota que emplean la biblioteca, hay muchas nuevas empresas como Applied Minds, VideoSurf y Zeitera, que hacen un uso extensivo de OpenCV. Los usos desplegados de OpenCV abarcan desde unir imágenes de streetview, detectar intrusiones en videos de vigilancia en Israel, monitorear equipos de minas en China, ayudar a los robots a navegar y recoger objetos en Willow Garage, detectar accidentes por ahogamiento en piscinas en Europa, ejecutar arte interactivo en España y Nueva York, revisando las pistas de aterrizaje en busca de escombros en Turquía, inspeccionando las etiquetas de los productos en fábricas de todo el mundo hasta la detección rápida de rostros en Japón.(Xu, Baojie, Guoxin 2017)

Tiene interfaces C ++, Python, Java y MATLAB y es compatible con Windows, Linux, Android y Mac OS. OpenCV se inclina principalmente hacia aplicaciones de visión en tiempo real y aprovecha las instrucciones MMX y SSE cuando están disponibles. Actualmente se están desarrollando activamente interfaces CUDA y OpenCL con todas las funciones. Hay más de 500 algoritmos y aproximadamente 10 veces más funciones que componen o admiten esos algoritmos. OpenCV está

escrito de forma nativa en C++ y tiene una interfaz con plantilla que funciona a la perfección con los contenedores STL.(Othman et al. 2019)

### 1.7.3. *Tensorflow*

TensorFlow es una biblioteca de código abierto desarrollada por Google y se ha vuelto muy popular con Machine Learning. TensorFlow ofrece API que facilitan el aprendizaje automático. TensorFlow también tiene un tiempo de compilación más rápido que otras bibliotecas de aprendizaje profundo como Keras y Touch. TensorFlow admite dispositivos informáticos tanto con CPU como con GPU.

Originalmente fue desarrollado para ejecutar grandes cálculos numéricos, lo que lo convierte en una gran herramienta para Deep Learning. Acepta datos en forma de matrices multidimensionales llamadas tensores. Esta aplicación se presta para manejar grandes cantidades de datos. El mecanismo de ejecución es en forma de gráficos para que sea más fácil ejecutar el código.(Dillon et al. 2017)

Una vez que haya accedido a los datos en TensorFlow, hay un cálculo que debe realizarse. Cada cálculo en TensorFlow se representa como un gráfico de flujo de datos. No es como la programación tradicional. Preparamos gráficos con nodos y luego se ejecutan en forma de sesión con los datos en los Tensores. Cada nodo del gráfico representa una operación matemática (sumar, restar, multiplicar, etc.) y cada borde representa matrices multidimensionales que son los tensores. A continuación, se ejecuta el gráfico y se procesan los datos.(Xia, Xu, Nan 2017)

Se necesita mucho tiempo para entrenar los modelos en aprendizaje profundo debido a la gran cantidad de datos y el uso de TensorFlow hace que sea más fácil escribir el código y luego ejecutarlo de manera distribuida.

### 1.7.4. *Anaconda*

Anaconda es una distribución de los lenguajes Python y R. de computación científica, esta distribución contiene paquetes de ciencia de datos de Windows, Linux y MacOS diseñados para facilitar la administración y la implementación de aplicaciones.

Los modelos de paquetes Anaconda son manejados por conda framework package control.

Este administrador de paquetes se ha distribuido como un paquete de código abierto separado, ya que era útil solo y para cualquier otra cosa que no fuera Python. También hay un pequeño bootstrap de

Anaconda llamado Miniconda, con solo conda, Python, los paquetes de los que dependen y una pequeña cantidad de otros paquetes.(Ditrich 1996)

#### 1.7.5. *Keras*

Keras es una biblioteca de alto nivel construida sobre Theano o TensorFlow. Proporciona una API de tipo scikit-learn (escrita en Python) para construir redes neuronales. Los desarrolladores pueden usar Keras para construir redes neuronales rápidamente sin preocuparse por los aspectos matemáticos del álgebra tensorial, las técnicas numéricas y los métodos de optimización.(Gulli, Pal 2017)

La idea clave detrás del desarrollo de Keras es facilitar la experimentación mediante la creación rápida de prototipos. La capacidad de pasar de una idea a un resultado con el menor retraso posible es clave para una buena investigación. Esto ofrece una gran ventaja tanto para los científicos como para los desarrolladores principiantes porque pueden sumergirse directamente en el aprendizaje profundo sin ensuciarse las manos con cálculos de bajo nivel. El aumento de la demanda de Deep Learning ha dado lugar a un aumento de la demanda de personas capacitadas en Deep Learning.(Cortés 2017)

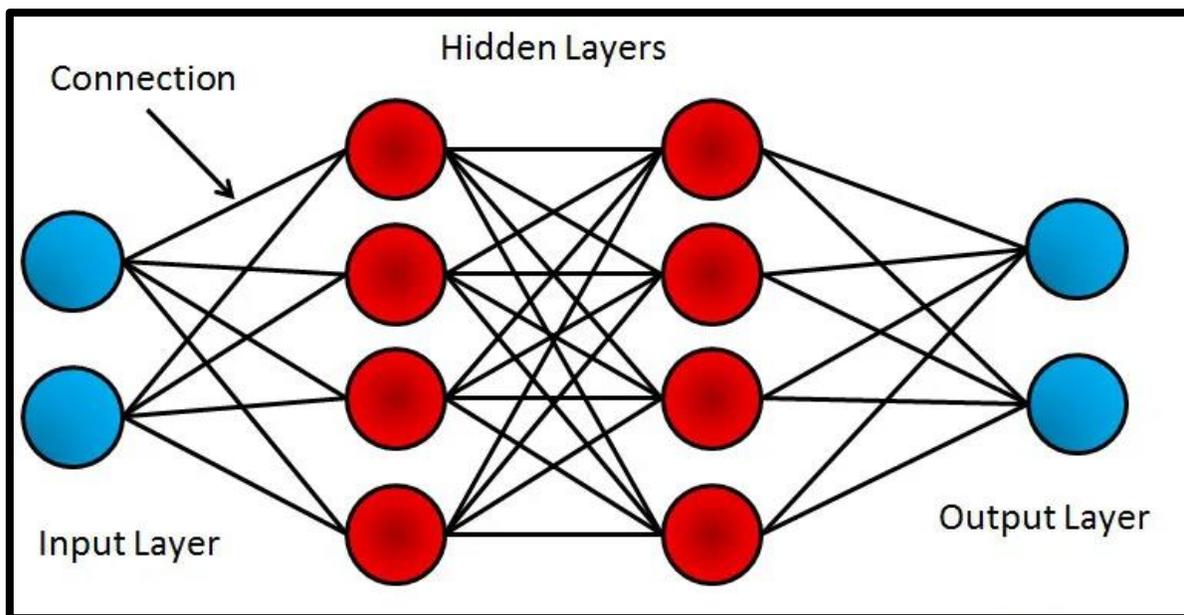
Todas las organizaciones intentan incorporar Deep Learning de una forma u otra, y Keras ofrece una API muy fácil de usar y lo suficientemente intuitiva para comprender, que esencialmente le ayuda a probar y crear aplicaciones de Deep Learning con un esfuerzo mínimo. Esto es bueno porque la investigación de Deep Learning es un tema tan candente en este momento y los científicos necesitan una herramienta para probar sus ideas sin perder tiempo en armar un modelo de red neuronal.(Jin, Song, Hu 2019)

#### 1.7.6. *Google colab*

Colab es un entorno de portátil Jupyter basado en la nube. Lo más significativo es que no requiere configuración y los miembros de su equipo pueden actualizar los cuadernos que crea al mismo tiempo, tal como lo hacen con los documentos de Google Docs. Colab incluye una serie de bibliotecas de aprendizaje automático conocidas que pueden instalarse simplemente en su computadora portátil. (Alves, Machado Vieira 2019)

## 1.8. Redes neuronales

Los sistemas informáticos inspirados en las redes neuronales biológicas para realizar diferentes tareas con una gran cantidad de datos involucrados se denominan redes neuronales artificiales o ANN. Se utilizan diferentes algoritmos para comprender las relaciones en un conjunto de datos dado para producir los mejores resultados a partir de las entradas cambiantes. La red está entrenada para producir los resultados deseados y se utilizan diferentes modelos para predecir los resultados futuros con los datos. Los nodos están interconectados para que funcione como un cerebro humano. Se utilizan diferentes correlaciones y patrones ocultos en los datos sin procesar para agrupar y clasificar los datos, como se observa en la figura 13-1. (Izaurieta, Saavedra 1999)



**Figura 13-1.** Estructura de una red neuronal

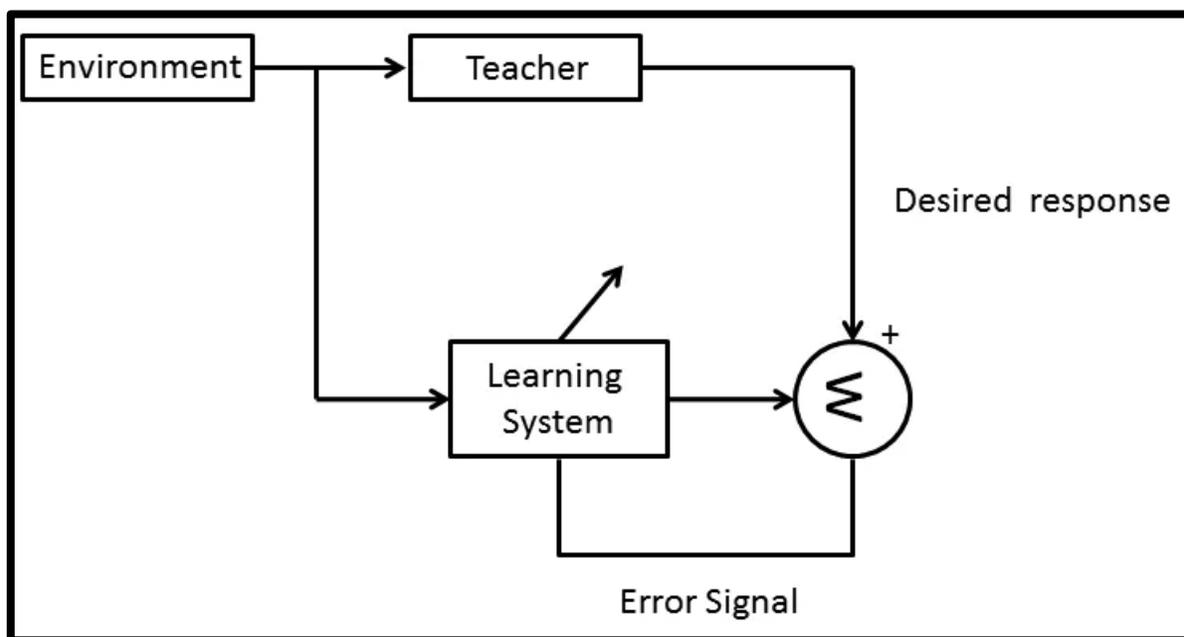
Realizado por: Yandry Jácome. 2021

Las redes neuronales se entrenan y se enseñan al igual que se entrena el cerebro en desarrollo de un niño. No se pueden programar directamente para una tarea en particular. Están entrenados de tal manera que puedan adaptarse de acuerdo con los cambios de entrada. Hay tres métodos o paradigmas de aprendizaje para enseñar una red neuronal.

- Aprendizaje supervisado
- Aprendizaje reforzado
- Aprendizaje sin supervisión

### 1.8.1. *Aprendizaje supervisado*

Como sugiere el nombre, el aprendizaje supervisado significa en presencia de un supervisor o un maestro. Significa que un conjunto de un conjunto de datos etiquetado ya está presente con la salida deseada, es decir, la acción óptima que debe realizar la red neuronal, que ya está presente para algunos conjuntos de datos. Luego, la máquina recibe nuevos conjuntos de datos para analizar los conjuntos de datos de entrenamiento y producir la salida correcta, como se observa en la figura 14-1. (Matich 2001)

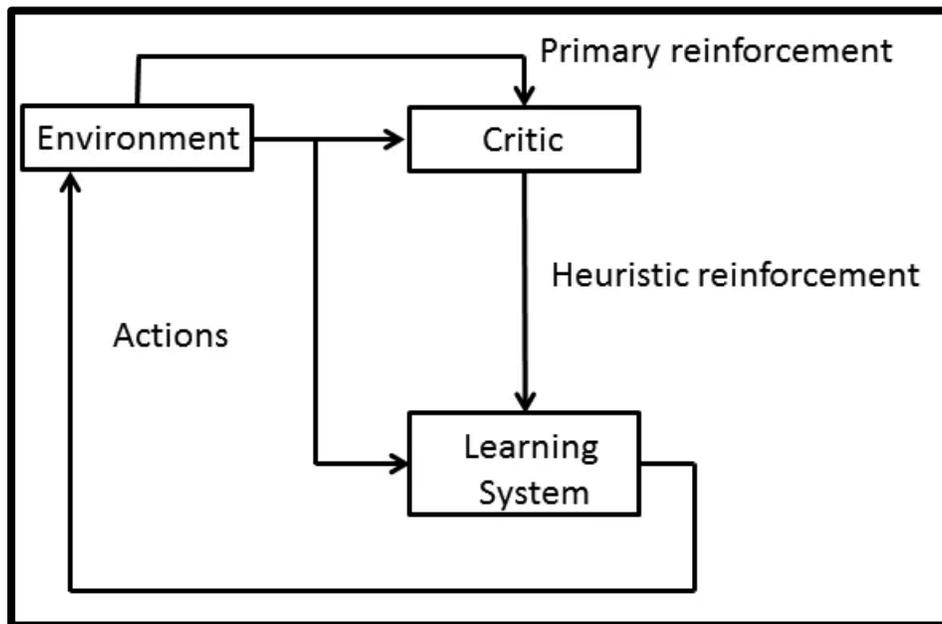


**Figura 14-1.** Estructura de aprendizaje supervisado

Realizado por: Yandry Jácome. 2021

### 1.8.2. Aprendizaje reforzado

En esto, el aprendizaje del mapeo de entrada-salida se realiza mediante la interacción continua con el entorno para minimizar el índice escalar de rendimiento. En esto, en lugar de un profesor, un crítico convierte la señal de refuerzo primaria, es decir, la entrada escalar recibida del entorno, en una señal de refuerzo heurística (señal de refuerzo de mayor calidad) entrada escalar, como se observa en la figura 15-1.(Neuronales, Neuronales, Larra 2015)

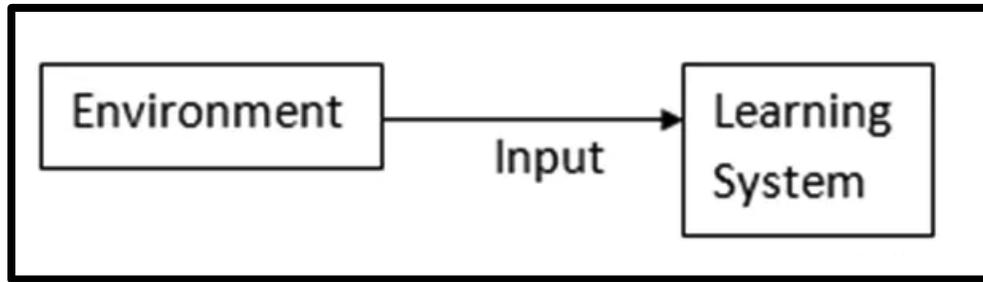


**Figura 15-1.** Estructura de aprendizaje reforzado

Realizado por: Yandry Jácome. 2021

### 1.8.3. Aprendizaje sin supervisión

Como sugiere el nombre, no hay ningún maestro o supervisor disponible. En esto, los datos no están etiquetados ni clasificados, y la red neuronal no dispone de una guía previa. En esto, la máquina tiene que agrupar los conjuntos de datos proporcionados de acuerdo con las similitudes, diferencias y patrones sin ningún entrenamiento proporcionado de antemano, como se observa en la figura 16-1.



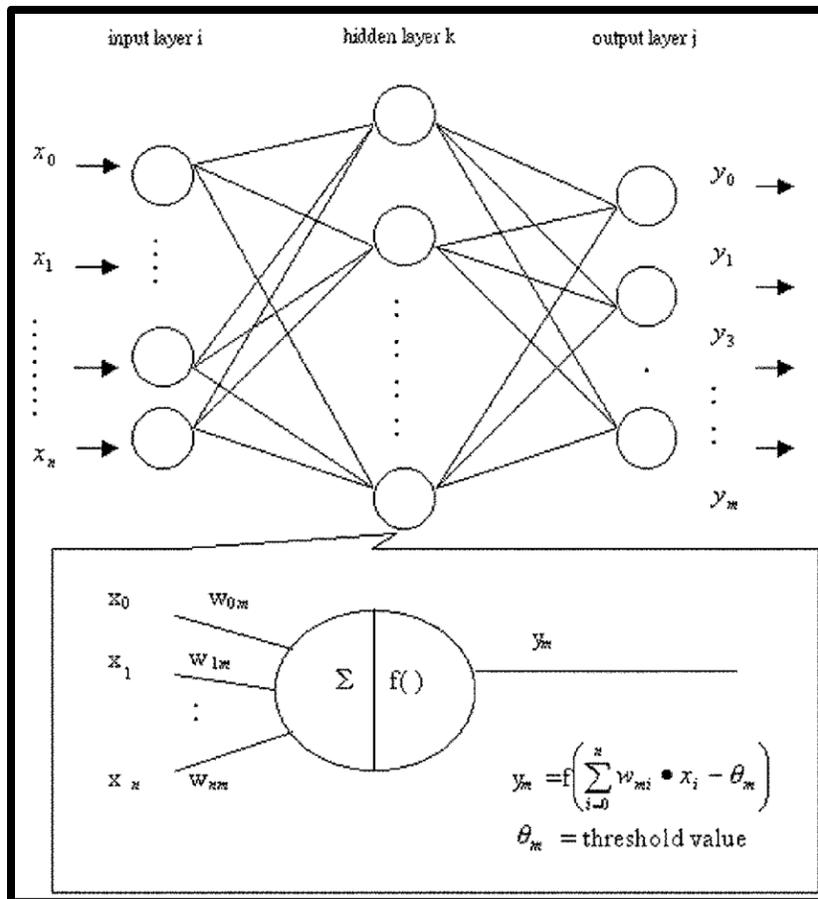
**Figura 16-1.** Estructura de un aprendizaje sin supervisión

Realizado por: Yandry Jácome. 2021

#### *1.8.4. Desarrollo de una red neuronal*

La red neuronal es un gráfico ponderado donde los nodos son las neuronas y los bordes con pesos representan las conexiones. Toma información del mundo exterior y se denota por  $x(n)$ .

Cada entrada se multiplica por sus respectivos pesos y luego se suman. Se agrega un sesgo si la suma ponderada es igual a cero, donde el sesgo tiene una entrada como 1 con ponderación  $b$ . Luego, esta suma ponderada se pasa a la función de activación. La función de activación limita la amplitud de la salida de la neurona. Hay varias funciones de activación como la función de umbral, la función lineal por partes o la función sigmoidea, como se observa en la figura 17-1.



**Figura 17-1.** Estructura de una red neuronal

Realizado por: Yandry Jácome. 2021

## 1.9. Arquitectura de una red neuronal

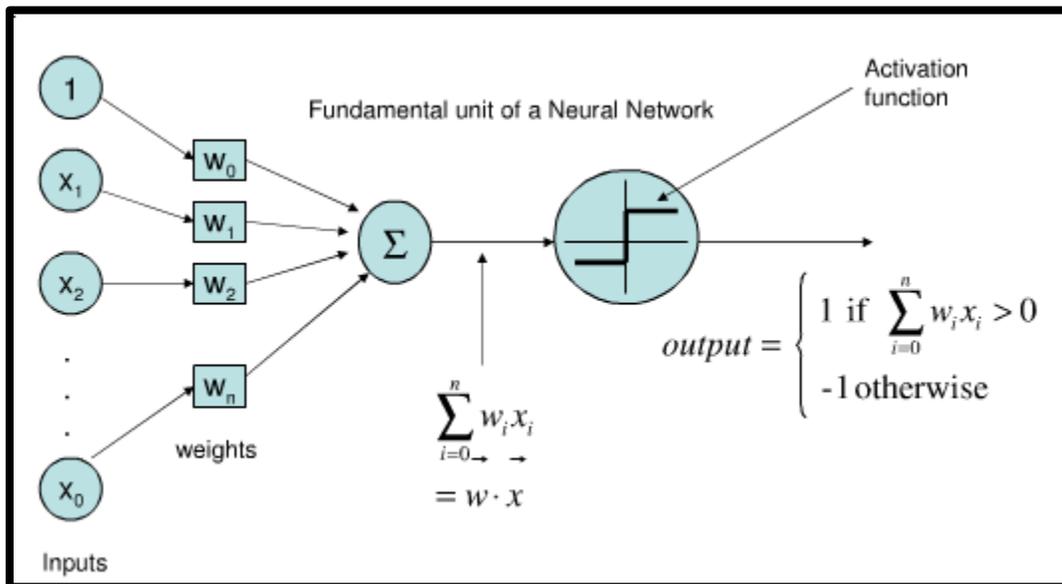
Básicamente, existen tres tipos de arquitectura de la red neuronal.

- Red de alimentación directa de una sola capa
- Red de alimentación de varias capas
- Red recurrente

### 1.9.1. Red de alimentación directa de una sola capa

En esto, tenemos una capa de entrada de nodos fuente proyectada en una capa de salida de neuronas. Esta red es una red acíclica o feedforward. Se denomina capa única porque solo se refiere a las

neuronas de cálculo de la capa de salida. No se realiza ningún cálculo en la capa de entrada; por tanto, no se cuenta, como se observa en la figura 18-1.(Dzib Goodin 2013)

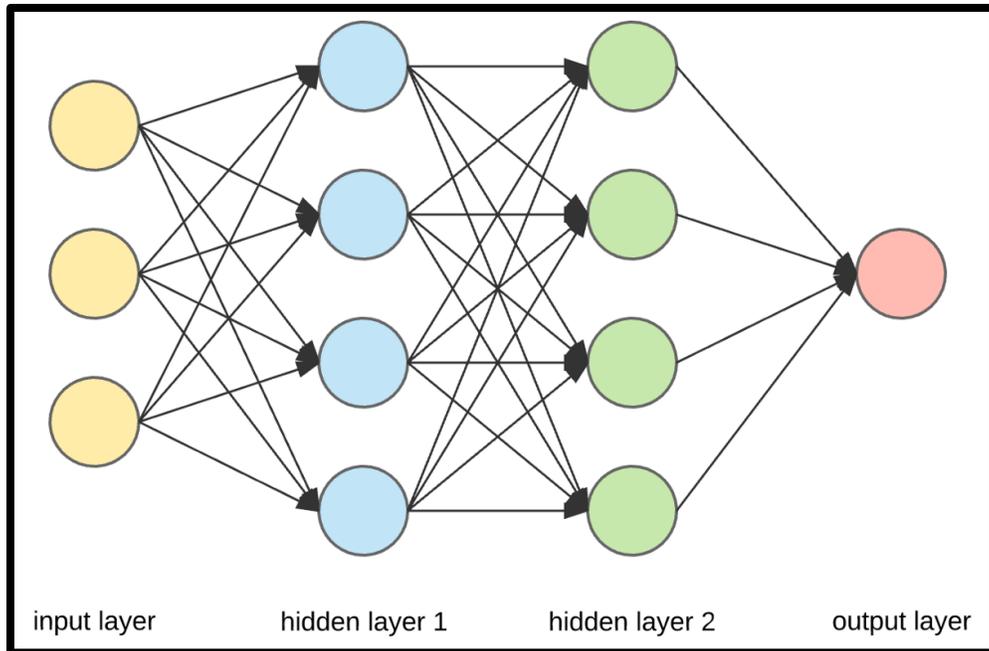


**Figura 18-1.** Red de alimentación directa de una sola capa

Realizado por: Yandry Jácome. 2021

### 1.9.2. Red de alimentación de varias capas

En esto, hay una o más capas ocultas, excepto las capas de entrada y salida. Los nodos de esta capa se denominan neuronas o unidades ocultas. El papel de la capa oculta es intervenir entre la salida y la entrada externa. Los nodos de la capa de entrada suministran la señal de entrada a los nodos de la segunda capa, es decir, la capa oculta, y la salida de la capa oculta actúa como entrada para la siguiente capa, y esto continúa para el resto de la red, como se observa en la figura 19-1. (López 2017)

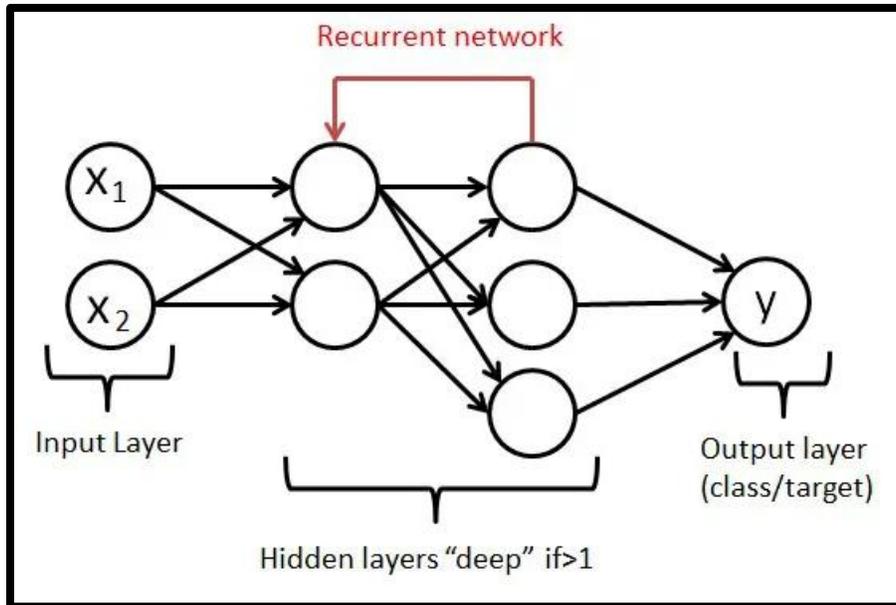


**Figura 19-1.** Red de alimentación de varias capas

Realizado por: Yandry Jácome. 2021

### 1.9.3. *Red recurrente*

Un recurrente es casi similar a una red feedforward. La principal diferencia es que al menos tiene un circuito de retroalimentación. Puede haber cero o más capas ocultas, pero al menos habrá un ciclo de retroalimentación, como se observa en la figura 20-1. (López 2017)

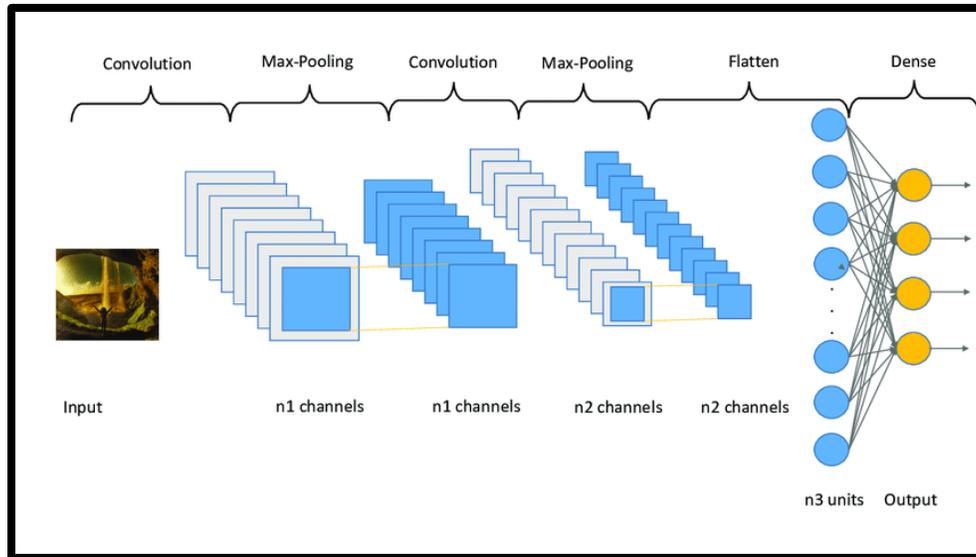


**Figura 20-1.** Red recurrente

Realizado por: Yandry Jácome. 2021

### 1.10. Redes neuronales convolucionales (CNN)

Una red neuronal convolucional es un tipo específico de red neuronal con múltiples capas. Procesa datos que tienen una disposición similar a una cuadrícula y luego extrae características importantes. Una gran ventaja de usar CNN es que no necesita hacer mucho preprocesamiento de imágenes, como se observa en la figura 21-1.



**Figura 21-1.** Red neuronal convolucional

Realizado por: Yandry Jácome. 2021

Con la mayoría de los algoritmos que manejan el procesamiento de imágenes, los filtros generalmente los crea un ingeniero basado en heurística. Las CNN pueden aprender qué características de los filtros son las más importantes. Eso ahorra mucho tiempo y trabajo de prueba y error, ya que no necesitamos tantos parámetros. (López 2017)

No parece un gran ahorro hasta que no trabaje con imágenes de alta resolución que tengan miles de píxeles. El objetivo principal del algoritmo de red neuronal convolucional es obtener datos en formularios que sean más fáciles de procesar sin perder las características que son importantes para descubrir qué representan los datos. Esto también los convierte en excelentes candidatos para manejar grandes conjuntos de datos. (Durán Suárez, Del Real Torres 2017)

Una gran diferencia entre una CNN y una red neuronal normal es que las CNN usan convoluciones para manejar las matemáticas detrás de escena. Se utiliza una convolución en lugar de la multiplicación de matrices en al menos una capa de la CNN. Las convoluciones toman dos funciones y devuelven una función.

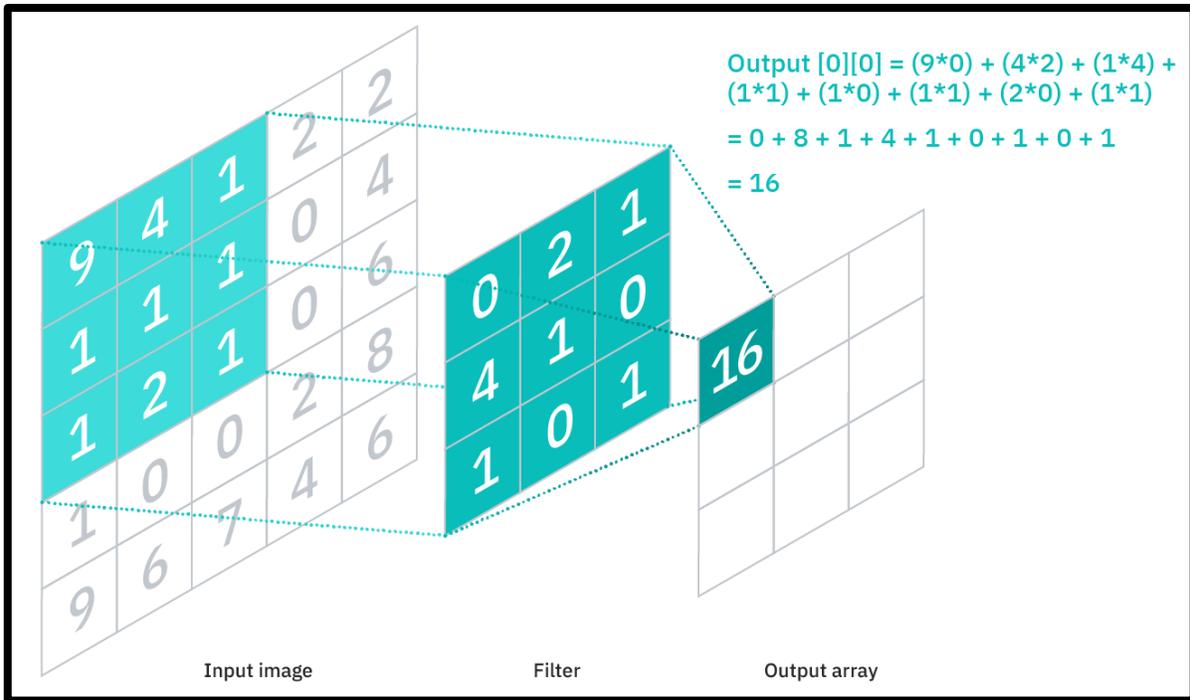
Las CNN funcionan aplicando filtros a sus datos de entrada. Lo que los hace tan especiales es que las CNN pueden ajustar los filtros a medida que ocurre el entrenamiento. De esa manera, los resultados se ajustan en tiempo real, incluso cuando tiene grandes conjuntos de datos, como con imágenes.

Dado que los filtros se pueden actualizar para entrenar mejor a la CNN, esto elimina la necesidad de filtros creados a mano. Eso nos da más flexibilidad en la cantidad de filtros que podemos aplicar a un conjunto de datos y la relevancia de esos filtros. Con este algoritmo, podemos trabajar en problemas más sofisticados como el reconocimiento facial.(Picazo Montoya 2018)

### *1.10.1. Capa convolucional*

La capa convolucional es el bloque de construcción central de una CNN, y es donde ocurre la mayor parte de los cálculos. Requiere algunos componentes, que son datos de entrada, un filtro y un mapa de características. Supongamos que la entrada será una imagen en color, que se compone de una matriz de píxeles en 3D. Esto significa que la entrada tendrá tres dimensiones (altura, ancho y profundidad) que corresponden a RGB en una imagen. También tenemos un detector de características, también conocido como kernel o filtro, que se moverá a través de los campos receptivos de la imagen, verificando si la característica está presente. Este proceso se conoce como convolución, como se observa en la figura 22-1.

El detector de características es una matriz bidimensional (2-D) de pesos, que representa parte de la imagen. Si bien pueden variar en tamaño, el tamaño del filtro suele ser una matriz de 3x3; esto también determina el tamaño del campo receptivo. A continuación, el filtro se aplica a un área de la imagen y se calcula un producto escalar entre los píxeles de entrada y el filtro. Este producto escalar se alimenta luego a una matriz de salida. Posteriormente, el filtro se desplaza a pasos agigantados, repitiendo el proceso hasta que el núcleo ha barrido toda la imagen. La salida final de la serie de productos escalares de la entrada y el filtro se conoce como mapa de características, mapa de activación o característica convolucionada.(Loncomilla 2016)



**Figura 22-1.** Capa convolucional

Fuente: Loncomilla. 2016

### 1.10.2. Capa pooling

La agrupación de capas (pooling), también conocida como submuestreo, lleva a cabo una reducción de dimensionalidad, lo que reduce el número de parámetros en la entrada. Similar a la capa convolucional, la operación de agrupación barre un filtro a través de toda la entrada, pero la diferencia es que este filtro no tiene ningún peso. En cambio, el kernel aplica una función de agregación a los valores dentro del campo receptivo, llenando la matriz de salida. Hay dos tipos principales de agrupación:

- Agrupación máxima: a medida que el filtro se mueve a través de la entrada, selecciona el píxel con el valor máximo para enviar a la matriz de salida. Por otro lado, este enfoque tiende a usarse con más frecuencia en comparación con la agrupación promedio.
- Agrupación promedio: a medida que el filtro se mueve a través de la entrada, calcula el valor promedio dentro del campo receptivo para enviarlo a la matriz de salida.

Si bien se pierde mucha información en la capa de agrupación, también tiene una serie de beneficios para la CNN. Ayudan a reducir la complejidad, mejorar la eficiencia y limitar el riesgo de sobreajuste. (De, De 2020)

### *1.10.3. Capa de densamente conectada*

El nombre de la capa totalmente conectada se describe a sí misma adecuadamente. Como se mencionó anteriormente, los valores de píxeles de la imagen de entrada no están conectados directamente a la capa de salida en capas parcialmente conectadas. Sin embargo, en la capa completamente conectada, cada nodo de la capa de salida se conecta directamente a un nodo de la capa anterior.

Esta capa realiza la tarea de clasificación en base a las características extraídas a través de las capas anteriores y sus diferentes filtros. Mientras que las capas convolucionales y agrupadas tienden a usar funciones ReLu, las capas FC generalmente aprovechan una función de activación softmax para clasificar las entradas de manera apropiada, produciendo una probabilidad de 0 a 1. (Fitria 2013)

## CAPÍTULO II

### 2. MARCO METODOLÓGICO

Este capítulo contiene información sobre la arquitectura general del sistema de detección de carriles, las especificaciones de diseño y la colección de dispositivos de componentes del sistema y elementos de software.

#### 2.1. Requerimientos del prototipo

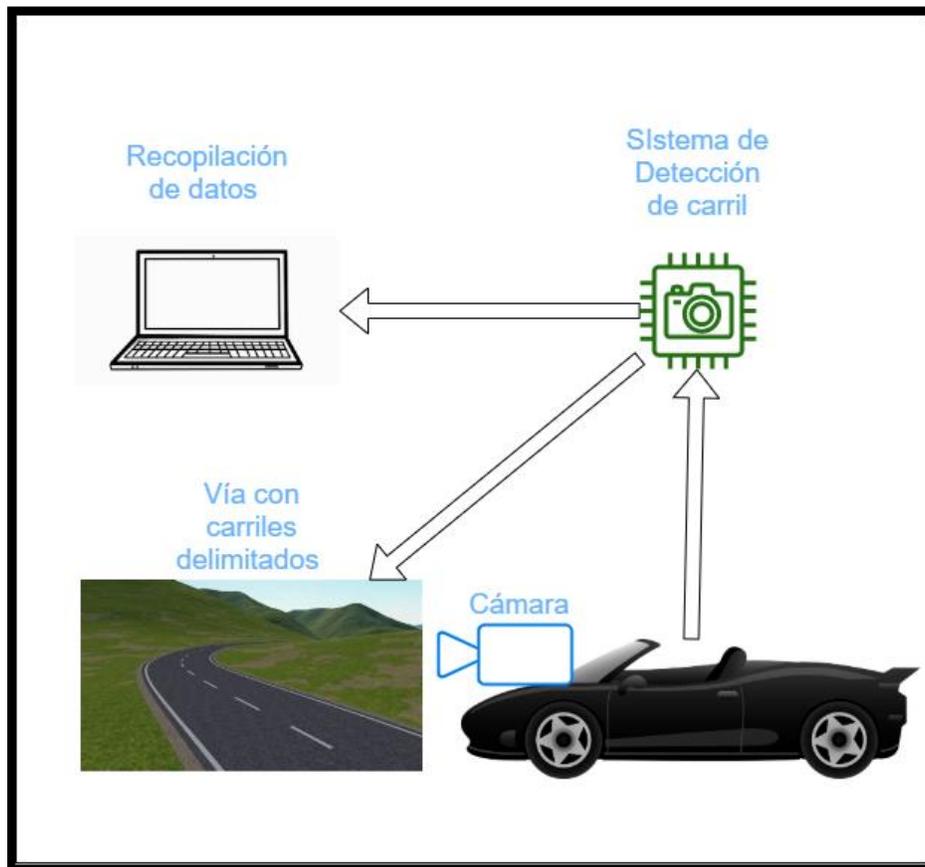
Las especificaciones para la implementación de este Sistema de Detección de Carriles se definieron en base al análisis realizado en el capítulo anterior, estos son:

- Capacidad para procesar imágenes a gran velocidad.
- Capacidad de detectar carriles a una distancia mínima de 30 cm de forma segura.
- Batería que permita una autonomía mínima de 15 minutos.
- Estructura portable y adaptativa.
- Capacidad de autoaprendizaje.
- Contar con protección estructural.
- Costo accesible.
- Capacidad de alerta al momento que no detecte el carril.
- Fácil instalación.

#### 2.2. Consolidación general del prototipo.

El sistema de detección de carril consta de una cámara de alta definición con una apertura f/1.8 con visión nocturna que puede grabar a alta definición, transmitir en vivo y una tarjeta de desarrollo raspberry pi 4 de 8gb que maneja un sistema operativo de 64 bits. Se utiliza un desarrollo de inteligencia artificial para que el vehículo corrija sus trayectorias manteniendo la posición correcta entre los carriles basado en redes convolucionales, que es una clase de red neuronal profunda que es más utilizada en el análisis de imágenes basado en su arquitectura de pesos compartidos y

características de invariancia de traducción, lo cual en la aplicación de este proyecto es perfecto, como se observa en la figura 1-2.



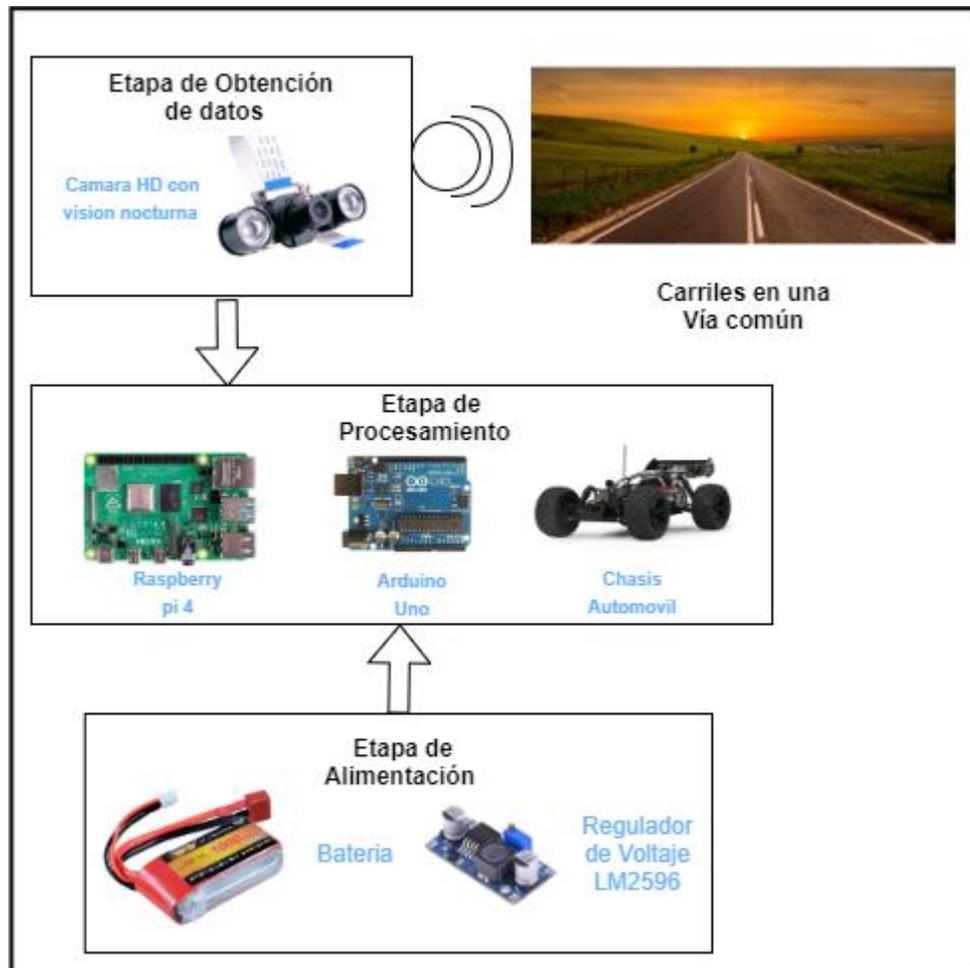
**Figura 1-2.** Consolidación general del prototipo

Realizado por: Yandry Jácome, 2021

### 2.3. Arquitectura de hardware del prototipo.

Se puede apreciar en la figura 2-2, este sistema está conformado por 3 etapas las cuales son: *la etapa de obtención de datos*, *la etapa de procesamiento* y *la etapa de alimentación*. *la etapa de obtención de datos* se encuentra una cámara la cual está encargada de enviar información del entorno hacia la *etapa de procesamiento* en la etapa mencionada como principal protagonista se tiene la tarjeta de desarrollo una raspberry pi 4 de 8 GB en la cual se encuentra programado el algoritmo de análisis de imágenes, Un microcontrolador a bordo se encarga del resultado del análisis de imágenes para proporcionar las órdenes dirección al vehículo y así evitar un descarrilamiento. *La etapa de*

*alimentación* suministra la energía necesaria para el funcionamiento de todos los componentes del sistema.



**Figura 2-2.** Arquitectura de hardware del prototipo

Realizado por: Yandry Jácome, 2021

#### 2.4. Diseño de las etapas de *hardware* del prototipo.

Cada una de sus fases se detalla hasta que se ha evaluado la arquitectura de hardware del prototipo y la operación se entiende mejor a continuación.

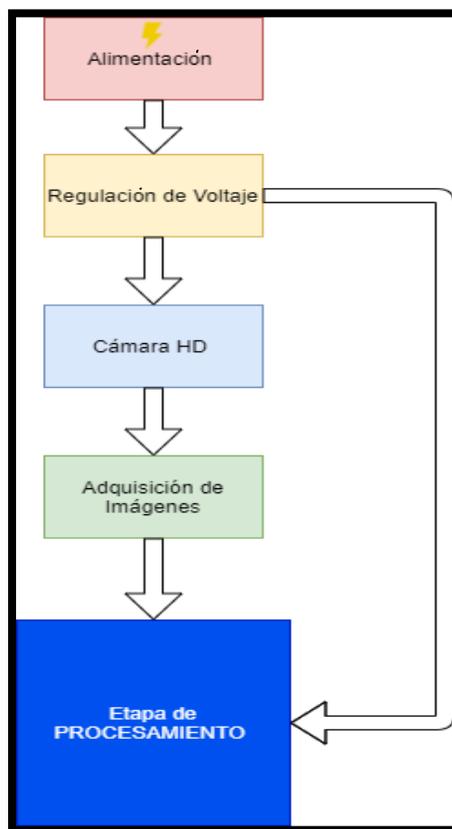
### 2.4.1. Etapa de alimentación

Es la etapa que proporciona a todos los componentes electrónicos la energía necesaria.

Se compone de reguladores de voltaje para reducir el voltaje de la batería a un voltaje que se adapte a la tarjeta de producción y sensores.

### 2.4.2. Etapa de obtención de datos

En la figura 3-2, se muestra el diagrama de bloques de esta etapa, cuyo propósito es capturar señales de la cámara y, a continuación, analizarlas en el bloque de procesamiento a las imágenes capturadas junto con métodos de inteligencia artificial. Permiten al usuario conocer la distancia a la que se encuentran los carriles delante del vehículo en tiempo real.

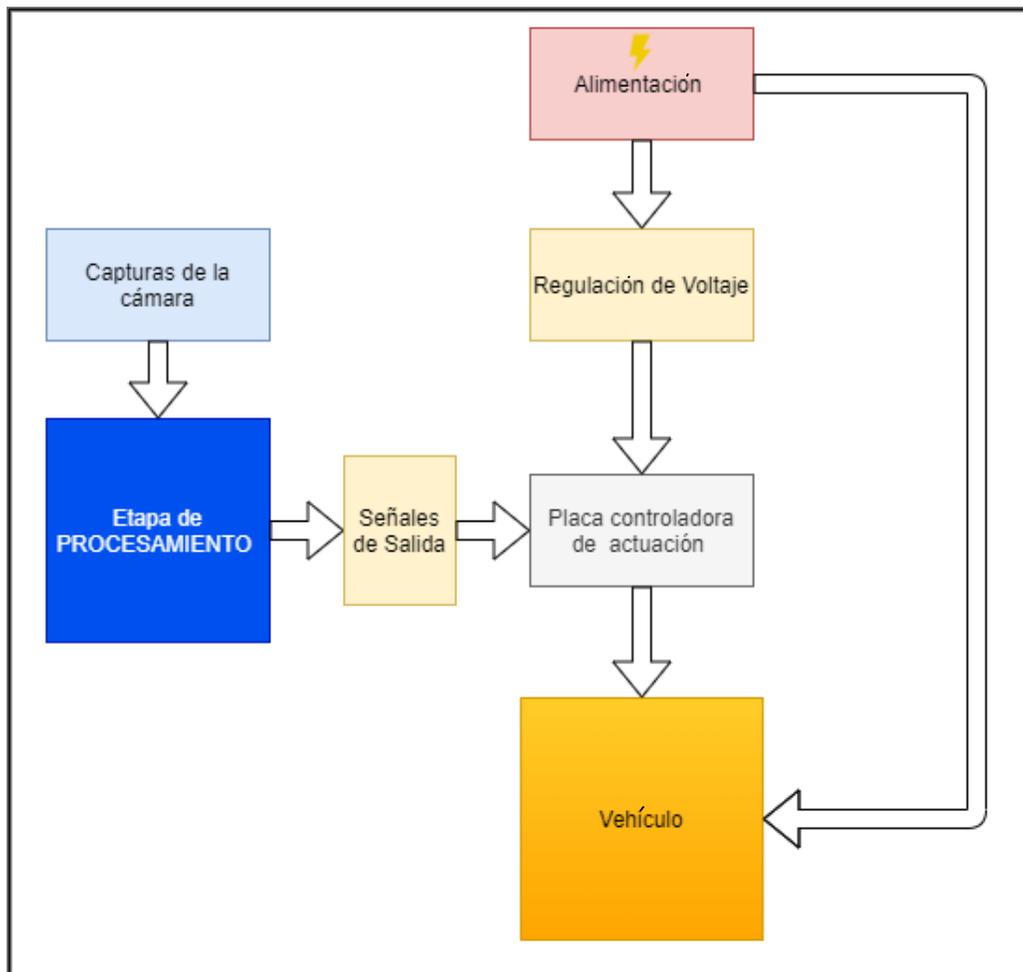


**Figura 3-2.** Diagrama de Bloques de la etapa de obtención de datos

Realizado por: Yandry Jácome, 2021

### 2.4.3. Etapa de procesamiento

Las señales receptadas por la cámara entraron en una etapa de procesamiento. Una vez procesadas por el microcontrolador se generan señales de salida, encargado de proporcionar los movimientos de estabilización del vehículo entre los carriles. En esta etapa se programa un controlador el cual modifica el sentido del vehículo para evitar un descarrilamiento, estas señales actúan en paralelo con otra señal de aceleración y freno que actúa en otro motor, como se observa en la figura 4-2.



**Figura 4-2.** Diagrama de Bloques de la etapa de Procesamiento

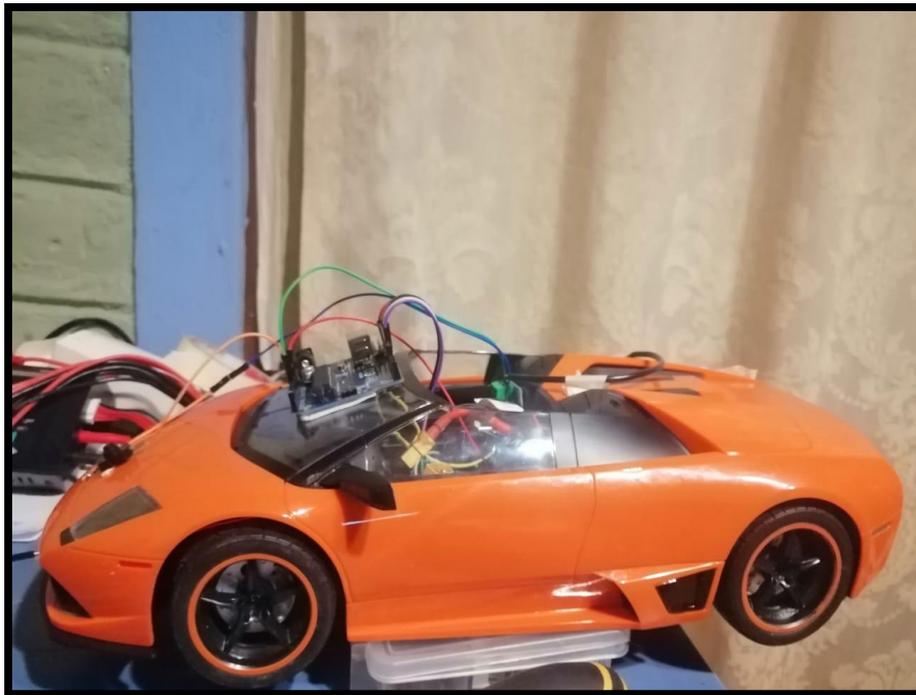
Realizado por: Yandry Jácome, 2021

## 2.5. Selección de componentes del *hardware* para la construcción del prototipo

Después de evaluar las especificaciones de hardware y cada uno de los bloques del dispositivo, continuaremos la selección de los componentes necesarios para el funcionamiento del prototipo.

### 2.5.1. *Chasis lamborghini RC*

Es un chasis a escala de 1 a 20 de un auto Ford Mustang, una plataforma confiable y compacta como se observa en la figura 5-2, con unas dimensiones de: 23.92 cm de longitud, 9.58 cm de anchura y 6.9 cm de altura, tiene una autonomía de 15 minutos, dependiendo la agresividad del terreno, la capacidad de la batería y la carga, como se observa en la figura 5-2.



**Figura 5-2.** Chasis Lamborghini escala de 1 a 10

Realizado por: Yandry Jácome, 2021

En la Tabla 1-2 se detallan las principales características del chasis RC

**Tabla 1-2:** Características del chasis

<b>CARACTERISTICAS</b>	
Peso	500 g
Largo	40 cm
Ancho	13 cm
Altura	9 cm
Distancia entre ejes	30.5cm

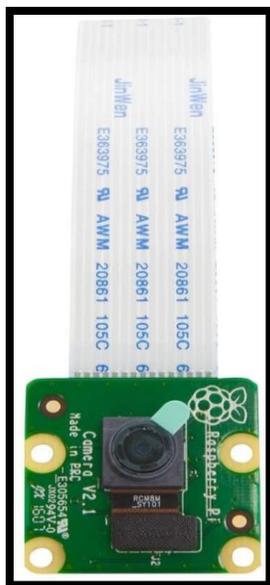
*Realizado por:* Yandry Jácome, 2021.

### **2.6.1. Dispositivos de obtención de datos**

Estos instrumentos permiten la adquisición de datos del ambiente con el fin de cuantificar fenómenos físicos como: voltaje, corriente, temperatura, distancia en forma de señales analógicas y convertirlos en variables digitales que pueden ser interpretadas por un microcontrolador.

#### **2.6.1.1. Raspberry Pi Módulo de cámara V2 – 8 megapíxeles, 1080p.**

La Raspberry Pi Camera v2 es un sensor de imagen de 8 megapíxeles Sony IMX219 de alta calidad diseñado a medida para Raspberry Pi, con un objetivo de enfoque fijo. Es capaz de imágenes estáticas de 3280 x 2464 píxeles, y también soporta vídeo 1080p30, 720p60 y 640x480p60/90. Se conecta a Pi por medio de uno de los pequeños zócalos en la superficie superior de la placa y utiliza la interfaz CSI dedicada, diseñada especialmente para interactuar con cámaras. La tabla en sí es pequeña, alrededor de 0.984 in x 0.906 in x 0.354 in. También pesa poco más de 0.11 oz, por lo que es perfecto para móviles u otras aplicaciones donde el tamaño y el peso son importantes. Se conecta a Raspberry Pi a través de un cable corto de cinta. El sensor de imagen IMX219 de alta calidad de Sony tiene una resolución nativa de 8 megapíxeles y tiene un objetivo de enfoque fijo a bordo, como se observa en la figura 6-2. En términos de imágenes fijas, la cámara es capaz de imágenes estáticas de 3280 x 2464 píxeles, y también soporta vídeo 1080p30, 720p60 y 640x480p90.(Pi 2015)



**Figura 6-2.** Raspberry Pi  
Camera V2

Fuente: Raspberry, 2020

En la Tabla 2-2 se detallan las principales características de Raspberry Pi Módulo de cámara V2

**Tabla 2-2:** Características del Raspberry Pi Módulo de cámara V2

<b>CARACTERISTICAS</b>	
Wide Angle	175 grados
Distancia focal	0.2
Resolución	5Mp
Altura	6.9 cm
Apertura	1.7mm
Visión Nocturna	No

Realizado por: Yandry Jácome, 2021.

## 2.6.2. Dispositivos de procesamiento

Estos dispositivos permiten la adquisición de datos ambientales para medir y traducir fenómenos físicos como voltaje, corriente, temperatura, distancia en forma de señales analógicas en variables digitales que pueden ser interpretadas por un microcontrolador.

### 2.6.2.1. Arduino uno

Es una placa de microcontrolador basada en el ATmega328P Tiene 14 pines de entrada / salida digital (de los cuales 6 se pueden usar como salidas PWM), 6 entradas analógicas, un resonador cerámico de 16 MHz (CSTCE16M0V53-R0), una conexión USB, un conector de alimentación, un encabezado ICSP y un botón de reinicio, como se observa en la figura 7-2. Contiene todo lo necesario para soportar el microcontrolador.



**Figura 7-2.** Arduino UNO

Fuente: Arduino, 2019

En la Tabla 3-2 se detallan las principales características del Arduino UNO:

**Tabla 3-2:** Características Arduino UNO

CARACTERISTICAS	
Microcontrolador	ATmega328P
Voltaje de entrada	5-12 v
Terminales Digitales	14
Terminales Analógicos	6
Memoria Flash	32Kb
Peso	12g

*Realizado por:* Yandry Jácome, 2021.

#### 2.6.2.2. Raspberry pi 4 4GB RAM

El Raspberry Pi 4 es un excelente ordenador de una sola placa que ofrece un alto nivel de potencia y puede ser un sustituto real de las computadoras de escritorio como se observa en la figura 8-2. Sin embargo, este modelo Pi no es la mejor opción si estás deseando usarlo para varios proyectos. Puedes usar este para aprender codificación y otras cosas eléctricas.



**Figura 8-2.** Raspberry Pi 4

**Fuente:** Raspberry, 2020

En la Tabla 4-2 se detallan las principales características de Raspberry pi 4:

**Tabla 4-2:** Características del Raspberry Pi 4

<b>CARACTERISTICAS</b>	
Procesador	1.5Ghz Broadcom BDM2711 (quad-core Cortex-A72) SoC
Memoria	4 Gb LPDDR4
gráficos	ARM VideoCore VI (500MHz)
Supported Codecs	OpenGL ES 3.0 graphics H.265 (4kp60 decode) H.264 (1080p60 decode, 1080p30 encode)
Salidas de Video	2 micro HDMI
GPIO	40 pin header
USB puertos	2 USB 3.0, 2 USB 2.0
Wired Networking	Gigabit Etheret
Wireless Networking	802.11ac Wi-Fi, Bluetooth 5.0

*Realizado por:* Yandry Jácome, 2021

### **2.6.3. Dispositivos de alimentación**

#### **2.6.3.1. Regulador de voltaje LM2596**

El módulo conversor DC-DC a diferencia de los tradicionales reguladores lineales como el LM317, puede proveer hasta 3 amperios a la salida de manera muy eficiente y por lo tanto emitiendo una menor cantidad de calor. Puede usarse para adaptar el voltaje de alimentación en distintas etapas de un circuito. En la figura 9-2, Se observa el módulo regulador de voltaje.



**Figura 9-2.** Modulo conversor LM2596

Fuente: Amazon, 2020

### 2.6.3.2. Batería Li-Po

Para el correcto funcionamiento del prototipo en conjunto con los controladores se han seleccionado baterías Li-Po compuestas por un Polímero de Litio, debido a que son usadas generalmente en coches RC principalmente por su relación entre capacidad y peso, como se observa en la figura 10-2.



**Figura 10-2.** Batería Li-Po Tattu 450mAh

Fuente: Amazon, 2020

Los parámetros principales de selección para baterías LiPo son:

- **El número de celdas.** - Este tipo de baterías tiene celdas de 3,7 voltios y 4.2 voltios cuando están totalmente cargadas conectadas en serie. Por ejemplo, una batería 4S tiene 4 celdas con un voltaje total de 14.8v descargada y 16.8 totalmente cargada. (Erle Robotics, 2018).
- **La capacidad de carga.** – Este parámetro indica cuánta energía puede mantener la batería y se indica en miliamperios (mAh). Por ejemplo, una batería de 1000 mAh sería completamente descargada en una hora con una carga de 1000 miliamperios. (Mobus, 2017).
- **La tasa de descarga.** - Es simplemente la rapidez con que una batería puede ser descargada de forma segura. Por ejemplo, una batería con una tasa de descarga de 10C se descarga a un ritmo de 10 más que la capacidad de la batería. (Bricogeek, 2018).

Para seleccionar la batería Li-Po a usar para este proyecto es necesario conocer el consumo de corriente de cada dispositivo electrónico que conforma el prototipo. En la tabla 5-2 se enumera cada uno de los componentes y la corriente que consume.

**Tabla 5-2:** Consumo de corriente de los dispositivos del prototipo

<b>DISPOSITIVO</b>	<b>CORRIENTE(MA)</b>
Arduino MEGA	40 mA
Raspberry Pi 4 junto con su módulo de Cámara	En reposo: 230 mA -> 1.2 W A pleno rendimiento: 350 mA -> 1.8 W
Motores	200 mA
<b>Total</b>	470mA-590mA

*Realizado por:* Yandry Jácome, 2021

Con el consumo de corriente total establecido, se procede a determinar la capacidad de la batería y establecer el tiempo de funcionamiento del sistema con todos los dispositivos electrónicos. Se ha seleccionado una batería Li-Po de 6 celdas con una capacidad de 1300mAh y una tasa de descarga de 95C de la marca R-Line con una protección estructural de plástico como se observa en la figura 11-2.



**Figura 11-2.** Batería Li-Po R-Line 1300mAh

Fuente: Amazon, 2020

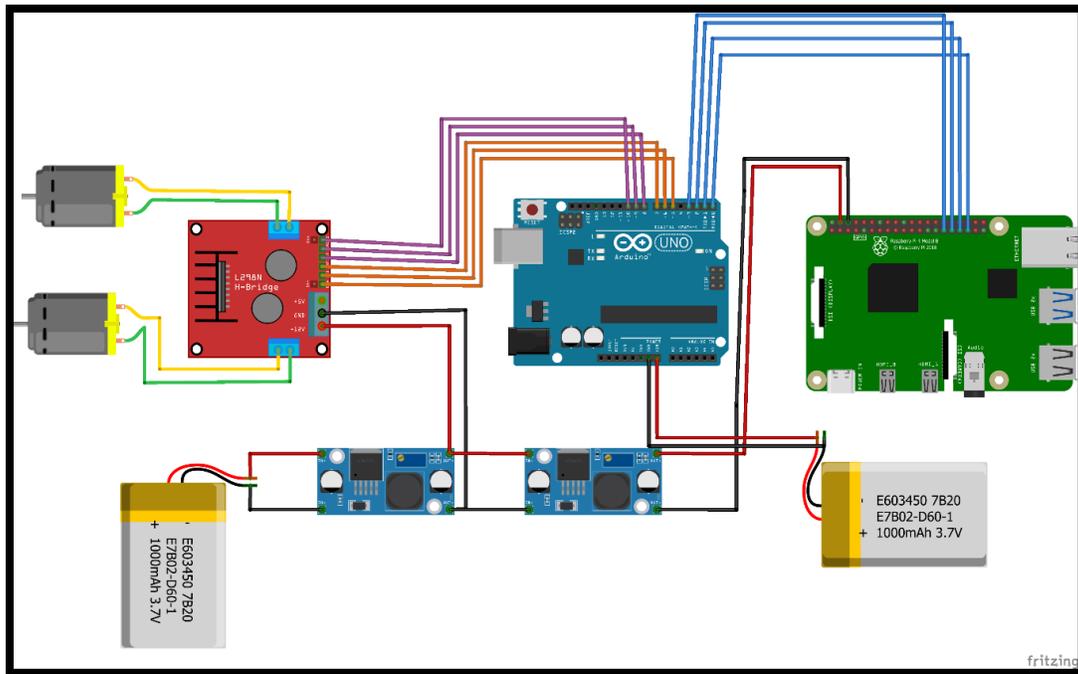
Para realizar el cálculo del tiempo de duración de la batería se conoce que el consumo máximo de corriente de todos los dispositivos electrónicos es de 590 mA, el voltaje de la batería es 22.2v y la potencia de la batería es 28.86wh dan una autonomía total de 2.20 horas.

## **2.7. Diagrama de conexión para el procesador central.**

En la figura 12-2. Se puede observar el esquema de comunicación electrónica del prototipo con todos los componentes que lo integran y los terminales de cada parte. El funcionamiento del procesador central se detalla a continuación.

- Para la etapa de alimentación el prototipo usa una batería li-po de 6 celdas que proporciona 25.2v. Para alimentar algunos elementos electrónicos como el puente H L298 y los motores se reduce el voltaje a 10V y una corriente de 3A, para ello se usa un módulo regulador LM2596 seguido de otro regulador LM2596 para reducir el voltaje a 5.05V y 3ª para la alimentación de la raspberry pi 4, y de manera independiente se alimenta el Arduino UNO con una batería de 3S de 11.1V, para evitar activaciones accidentales y controlar mediante un switch al módulo de los actuadores.
- El Arduino UNO es el encargado de procesar las señales provenientes de la raspberry hacia el puente H L298 y posteriormente a los actuadores.

- La Raspberry es la encargada de capturar las imágenes mediante una cámara y procesar dichas imágenes mediante inteligencia artificial para ubicar la línea de la carretera y así enviar las señales para un correcto direccionamiento del vehículo al Arduino UNO.



**Figura 12-2.** Diagrama de conexión de los componentes

Realizado por: Yandry Jácome, 2021.

En la tabla 6-2 se detallan las conexiones de todos los módulos que integran el *hardware* del procesador central

**Tabla 6-2:** Tabla de conexión de terminales y módulos.

TERMINALES ARDUINO UNO	MÓDULOS
Pines Analógicos	Raspberry
3	20
2	21
1 TX	22
0 RX	23
	Puente H
5	ENA

6	In1
7	In2
8	In3
9	In4
10	ENB

*Realizado por:* Yandry Jácome, 2021

## 2.8. Distribución de componentes internos.

La estructura del prototipo es un chasis de un auto RC hecho de polipropileno. Es un polímero termoplástico reciclable de gran resistencia y bajo peso ideal para simular un auto a escala. En la figura 13-2, se observa el auto RC en su paquete que sirve como el chasis del prototipo.



**Figura 13-2.** Chasis del prototipo en su caja

*Realizado por:* Yandry Jácome, 2021.

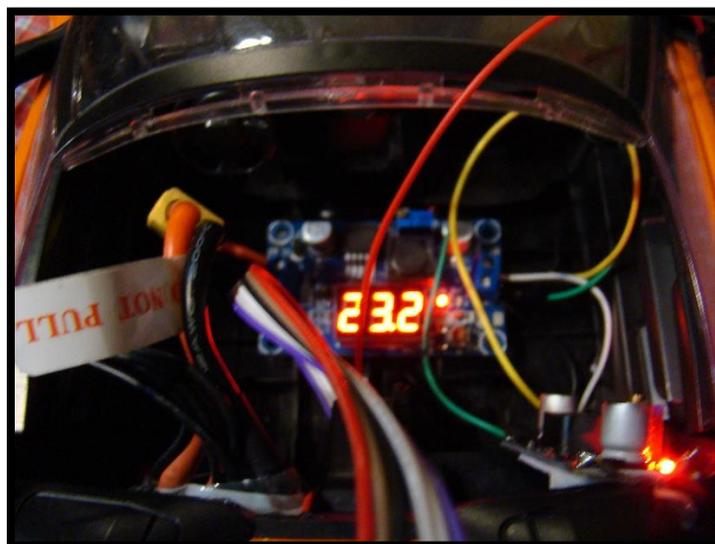
Luego de desarmar el auto RC por completo se desconecta el controlador y la antena que tiene el auto y empieza su transformación a un auto autónomo. Empezamos a medir los componentes y determinar la posición de los elementos de control del prototipo para optimizar el espacio tanto interno como externo. Colocamos la Batería y el Driver Puente H L298 en la parte interna del prototipo como se muestra en la figura 14-2.



**Figura 14-2.** Componentes internos del prototipo

**Realizado por:** Yandry Jácome, 2021.

Para facilitar la conexión de los motores y la alimentación del Driver Puente H L298. La Batería va conectada a un Regulador de voltaje LM2596 con Voltímetro que nos mostrara el estado de la batería como se observa en la figura 15-2, para evitar un mal funcionamiento de la raspberry.



**Figura 15-2.** Voltímetro del Regulador de voltaje LM2596

**Realizado por:** Yandry Jácome, 2021.

## 2.9. Distribución de componentes externos.

Para el posicionamiento de los componentes se tomó en cuenta varios parámetros entre ellos la distribución de peso y la posición los periféricos que se encargan de capturar datos para el correcto funcionamiento del prototipo.

### 2.9.1. Distribución de peso.

El prototipo cuenta con un motor DC de 12V-24V de 6000 a 12000 RPM de un gran par motor, en la parte trasera un motor DC de 12V de 3000 a 6000 RPM común de autos RC de juguete, esto se estableció por el peso que tiene la carrocería y los componentes electrónicos, distribuyendo los elementos más pesados a la parte trasera del vehículo para evitar un malfuncionamiento mecánico a la hora de efectuar un giro como se muestra en la figura 2-16. El posicionamiento correcto de los elementos nos da también acceso a correcciones de programación en las tarjetas de desarrollo.

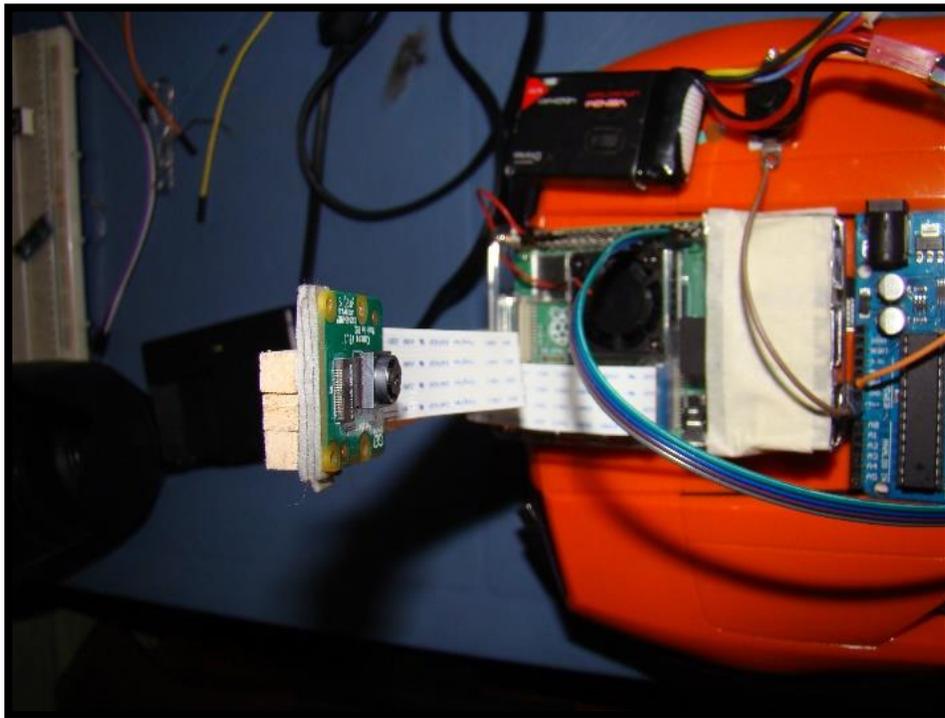


**Figura 16-2.** Posicionamiento de elementos electrónicos externos

Realizado por: Yandry Jácome, 2021.

### 2.9.2. Posición de periféricos.

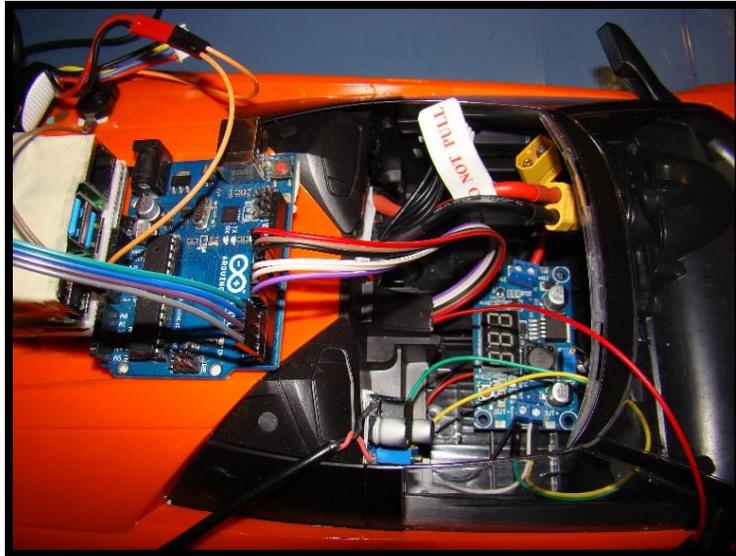
Como la posición de la cámara es determinante para el funcionamiento del prototipo y la corta longitud del cable de conexión de la Cámara (aproximadamente 15 cm), el cual se conecta a la Raspberry, condiciono mucho la posición de la tarjeta de desarrollo Raspberry pi 4 como se muestra en la figura 17-2. Pero logrando cumplir los parámetros que ponían en riesgo el funcionamiento del prototipo.



**Figura 17-2.** Posicionamiento Raspberry pi 4

**Realizado por:** Yandry Jácome, 2021.

El único componente que quedo con la libertad de posicionamiento es el Arduino UNO, y lo mejor fue ponerlo en el centro del vehículo para que este cerca de los componentes internos y externos como se muestra en la figura 18-2.

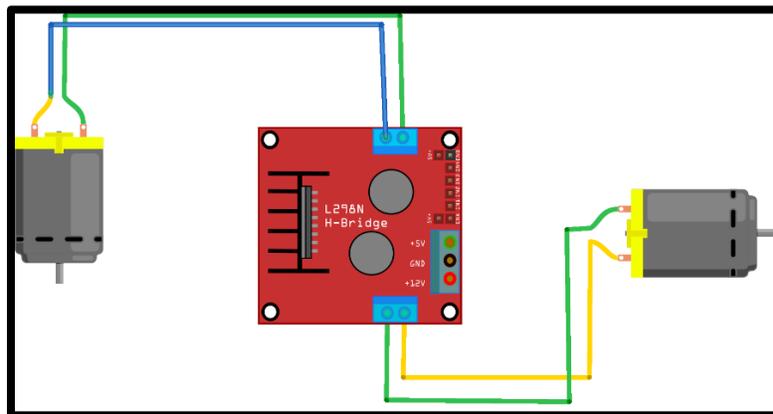


**Figura 18-2.** Posicionamiento Arduino UNO

Realizado por: Yandry Jácome, 2021.

## 2.10. Funcionamiento mecánico.

Para el funcionamiento se utilizan 2 motores eléctricos, uno encargado de la dirección ubicado en la parte frontal y otro encargado del impulso en la parte posterior del coche como se muestra en la figura 19-2. Los cuales son controlados por un Arduino UNO en modalidad de esclavo.



**Figura 19-2.** Distribución de los motores

Realizado por: Yandry Jácome, 2021.

## **2.11. Requerimientos de diseño para el software.**

Para procesar los datos generados por el sistema de visión artificial de detección de carril, se emplearon diferentes programas. Después de analizar los requerimientos de *hardware* se procedió a estudiar los requerimientos del *software*.

### **Los requerimientos del *software* del prototipo son:**

- Autocorrección de la trayectoria del vehículo mediante inteligencia artificial
- Autodetección de carril mediante análisis de imágenes e inteligencia artificial

## **2.12. Arquitectura del *software* del prototipo.**

Para la creación del software se utilizaron programas libres, para la programación de los procesadores se utilizó la tecnología Arduino ide basada en C ++, un código fuente abierto multipropósito con gran soporte. Se utilizó para la programación de Raspberry pi, el sistema operativo raspbian y el framework Python de código abierto.

### **2.12.1. Herramientas de software**

Esta sección proporciona un resumen de cada programa necesario para la creación del prototipo. Ambas herramientas de desarrollo utilizadas son tecnologías de robótica e ingeniería de acceso libre y código abierto.

#### **2.12.1.1. *Arduino IDE 1.8.13***

El software Arduino (IDE) de código abierto facilita la escritura de código y su carga en la placa. Este software se puede utilizar con cualquier placa Arduino. Tiene algunas instancias propias en cada una de las bibliotecas, lo que permite una mayor comprensión de la función de la maquinaria que se está tratando.

### 2.12.1.2. *Raspbian OS*

Raspbian es el sistema operativo preferido de Raspberry Pi construido sobre una distribución llamada Debian de GNU / Linux. La implementación del escritorio LXDE y Midori como navegador web por default.

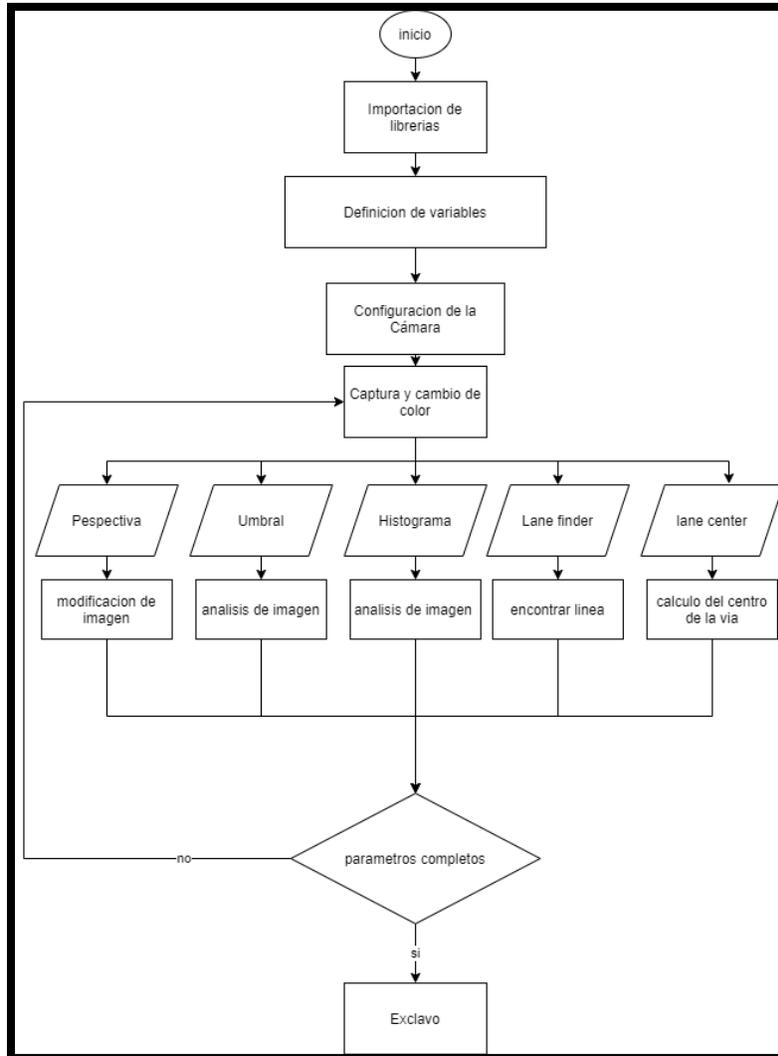
### 2.12.1.3. *Python*

Python es un lenguaje de programación interpretado, de alto nivel y de propósito general. La filosofía de diseño de Python enfatiza la legibilidad del código con su notable uso de sangría significativa. Sus construcciones de lenguaje y su enfoque orientado a objetos tienen como objetivo ayudar a los programadores a escribir código claro y lógico para proyectos de pequeña y gran escala.

## 2.12.2. *Desarrollo del código para el sistema de detección de carril.*

En la figura 20-2, podemos observar el diagrama de flujo del sistema de detección de carril programados en la raspberry en C++ y python. Las librerías aplicadas se muestran a continuación.

- Uno de los más importantes para el uso de inteligencia artificial para el análisis de imágenes es la aplicación de la librería de opencv mediante el código: `<opencv2/opencv.hpp>`
- Para el uso de la raspicam V2 y su obtención de datos se usó el código: `<raspicam_cv.h>`.
- Es una biblioteca estándar que nos permite utilizar operaciones de entrada/salida: `<iostream>`
- En esta biblioteca, proporciona un concepto de precisión neutral, al separar las duraciones y el momento: `<chrono>`.
- Interpreta el valor señalado por el temporizador como una hora del calendario y lo convierte en una cadena C que contiene una versión legible por humanos de la hora y fecha correspondientes, en términos de la hora local: `<ctime>`.
- WiringPi es una biblioteca de acceso GPIO basada en PIN escrita en C para los dispositivos SoC BCM2835, BCM2836 y BCM2837 utilizados en todas las Raspberry Pi. versiones. El código fuente no está disponible públicamente, pero puede estar disponible para aquellos que deseen soporte comercial: `<wiringPi.h>`.



**Figura 20-2.** Diagrama de flujo funcionamiento del código

Realizado por: Yandry Jácome, 2021.

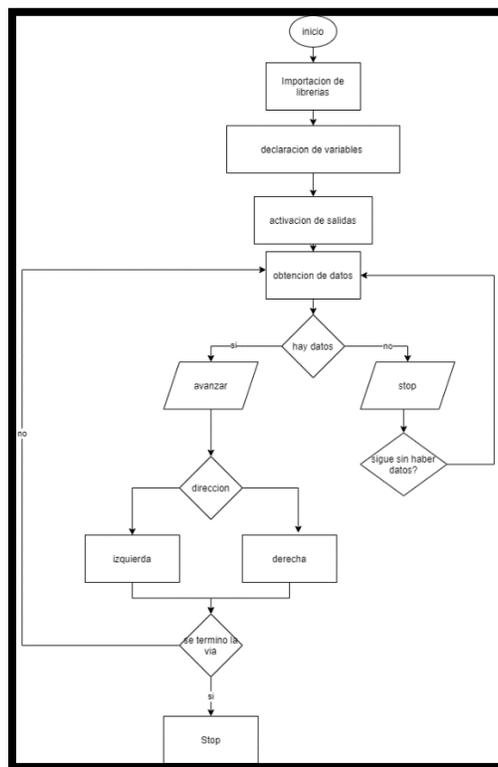
El algoritmo de para el sistema de detección de carril y evitar el descarrilamiento del vehículo se describe en el siguiente apartado:

- Se empieza con la importación de librerías y las declaraciones de las variables globales y valores iniciales de las mismas en el caso que se necesite.
- Empieza la activación y lectura de datos por parte de la cámara
- Los datos obtenidos por la cámara so procesados para obtener la información sin ningún tipo de ruido e interferencia externa

- Mediante inteligencia artificial se analizan y comparan los datos para determinar la posición del vehículo mientras se mueve, ante el posible descarrilamiento dependiendo del pronunciamiento de la curva el vehículo tomara la decisión de a qué velocidad girar las ruedas de dirección y la velocidad para tomar esa curva, tomando la mejor decisión entorno avanza.

### 2.12.3. Desarrollo del código para el procesador esclavo

Para evitar malfuncionamiento del sistema de análisis, los procesos fueron tomados por 2 procesadores diferentes. El Arduino UNO como se observa en la figura 21-2, se encarga de la lectura de datos enviados por la raspberry, mediante comunicación serial, para la posterior activación de los actuadores con el fin de evitar el descarrilamiento del vehículo.



**Figura 21-2.** Diagrama de flujo Arduino UNO

Realizado por: Yandry Jácome, 2021.

- Para la lectura de datos se establece una conexión a los pines 0, 1, 2, 3 del Arduino UNO los cuales serán activados dependiendo de la situación en que se encuentre el vehículo

- Al momento de activarse estos pines se desencadena un loop donde seguirá hasta que se pierda la detección de la línea e inmediatamente el auto frenará.

#### ***2.12.4. Protocolo de comunicación maestro-esclavo***

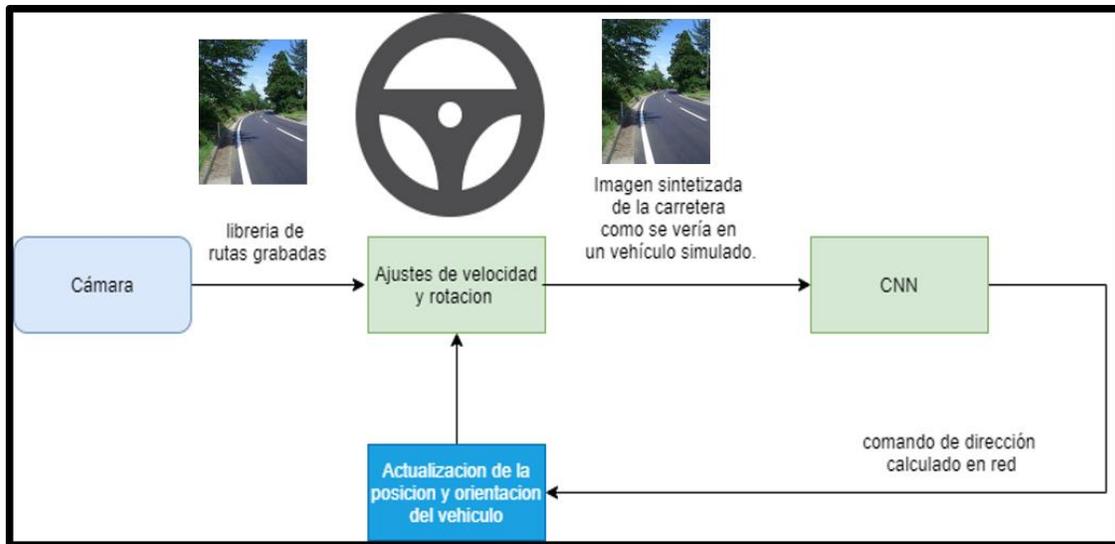
Ya que manejamos un prototipo donde necesitamos mayor velocidad de respuestas el protocolo a usar es: Serial Peripheral Interface (SPI), es un bus de interfaz que se usa comúnmente para enviar datos entre microcontroladores y pequeños periféricos como registros de desplazamiento, sensores y tarjetas SD. Utiliza líneas de datos y reloj separadas, junto con una línea de selección para elegir el dispositivo con el que desea hablar.

La comunicación SPI se implementa con 4 señales digitales:

- SCLK (Clock): Es el pulso que marca la sincronización. Con cada pulso de este reloj, se lee o se envía un bit.
- MOSI (Master Output Slave Input): Salida de datos del Maestro y entrada de datos al Esclavo.
- MISO (Master Input Slave Output): Salida de datos del Esclavo y entrada al Master.
- SS/Select: Para seleccionar un Esclavo, o para que el Maestro le diga al Esclavo que se active. Se requiere uno por cada Esclavo.

#### ***2.12.5. Corrección Mecánica***

Las correcciones mecánicas vienen dictadas por la red neuronal convolucional que al notar cualquier discrepancia en el camino envía inmediatamente una señal de corrección de los parámetros de orientación para que los actuadores ajusten su velocidad y orientación. Como podemos ver en la figura 22-2.



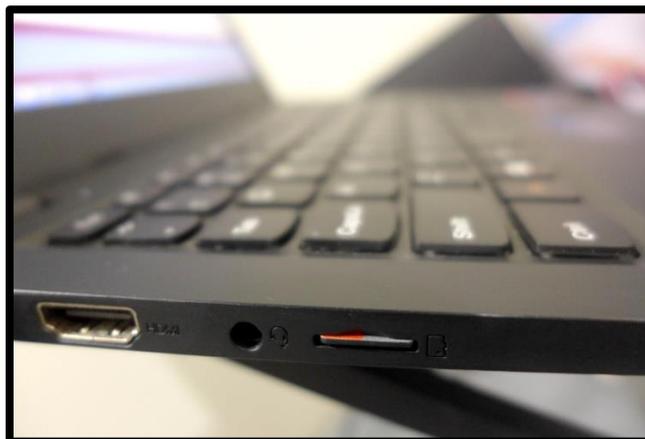
**Figura 22-2.** Diagrama de flujo Funcionamiento del prototipo

Realizado por: Yandry Jácome, 2021.

### 2.13. Instalación de raspbian

La instalación el sistema operativo Raspbian se realizó mediante Raspberry Pi Imager, la cual es una aplicación que nos permite seleccionar y descargar las imágenes de Raspbian disponibles e instalar de una manera fácil en la tarjeta SD, los pasos son los siguientes:

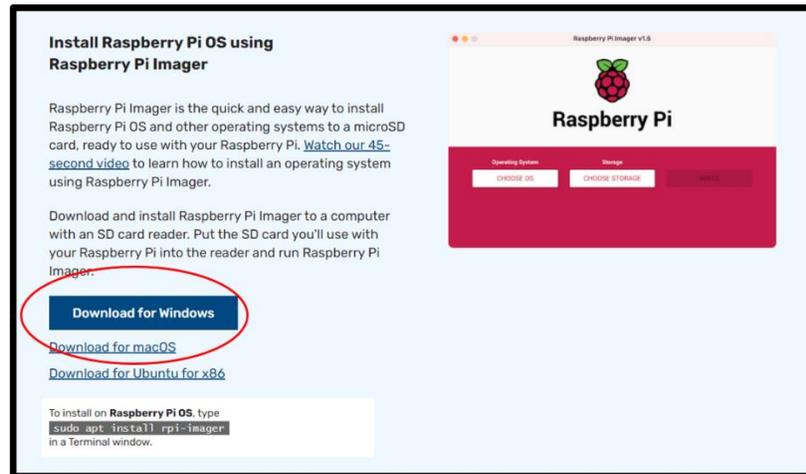
- Insertamos la tarjeta SD en el slot de la PC en el caso de no tenerlo podemos comprar un lector externo de tarjetas SD, como se observa en la figura 23-2.



**Figura 23-2.** Slot SD card

Realizado por: Yandry Jácome, 2021.

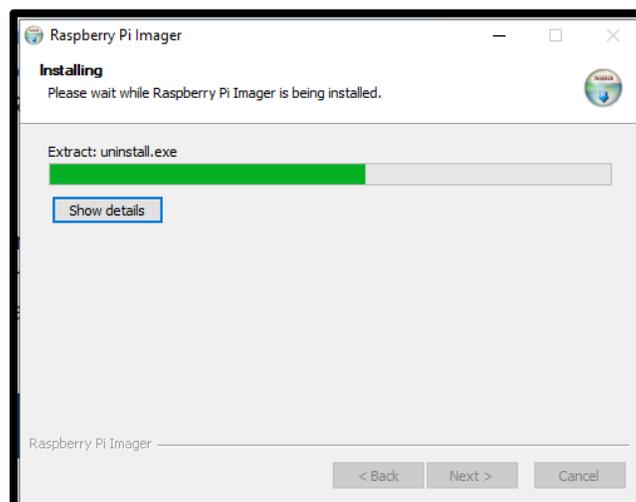
- Para descargarlo solo nos dirigimos a <https://www.raspberrypi.org/software/>
- Pulsamos en Download for windows (descargar para Windows), como se observa en la figura 24-2.



**Figura 24-2.** Descargar raspberry pi imager

Realizado por: Yandry Jácome, 2021.

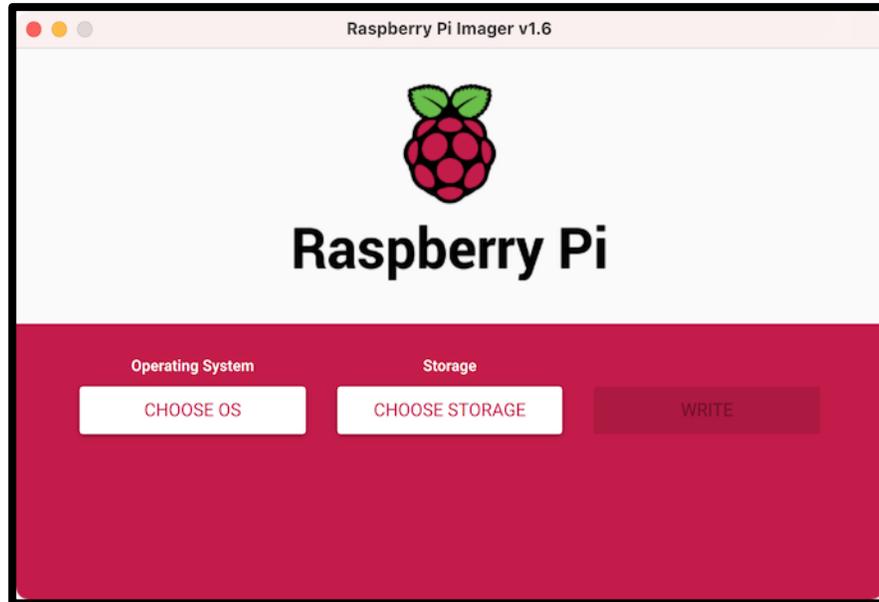
- Dependiendo de la velocidad de descarga del internet vamos a tener un archivo .exe el cual nos permite instalar raspberry pi imager, como se observa en la figura 25-2.



**Figura 25-2.** Instalación de raspberry pi imager

Realizado por: Yandry Jácome, 2021.

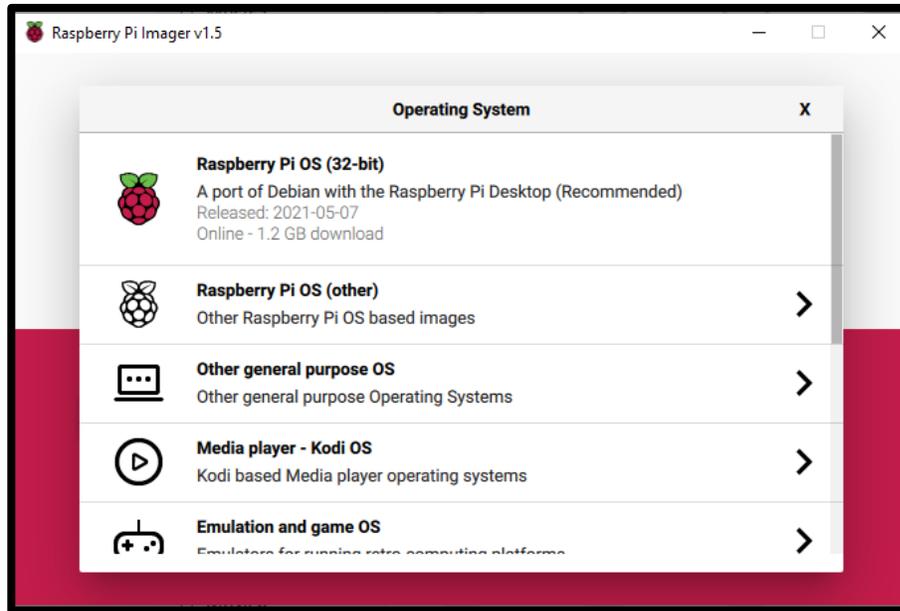
- Terminada la instalación se ejecuta el programa donde tendremos esta interfaz, como se observa en la figura 26-2.



**Figura 26-2.** Interfaz Raspberry Pi imager.

**Realizado por:** Yandry Jácome, 2021.

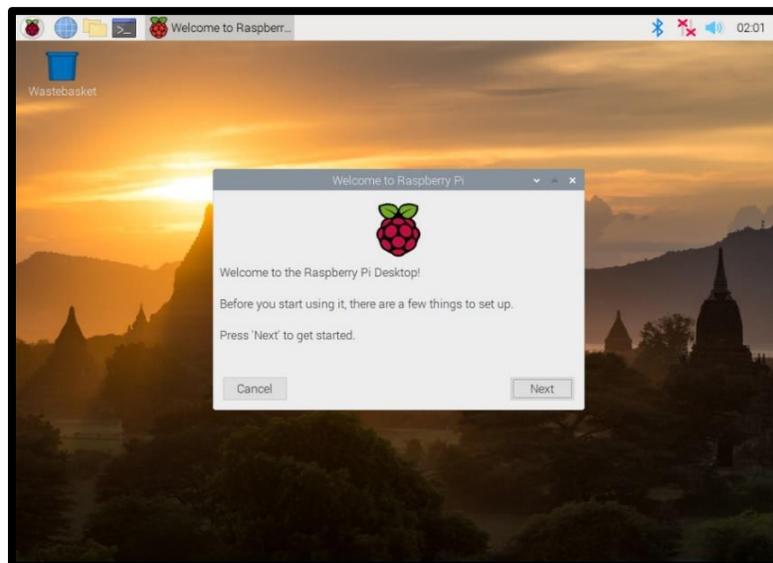
- Donde podremos elegir el sistema operativo que deseamos instalar en nuestro caso Raspbian que viene designado como Raspberry PI OS (32-bit), como se observa en la figura 27-2.



**Figura 27-2.** Interfaz Raspberry pi imager.

Realizado por: Yandry Jácome, 2021.

- Al seleccionar la imagen de Raspbian requerida se descarga e instala de forma automática donde al finalizar esta lista para su ejecución en la raspberry, como se observa en la figura 28-2.

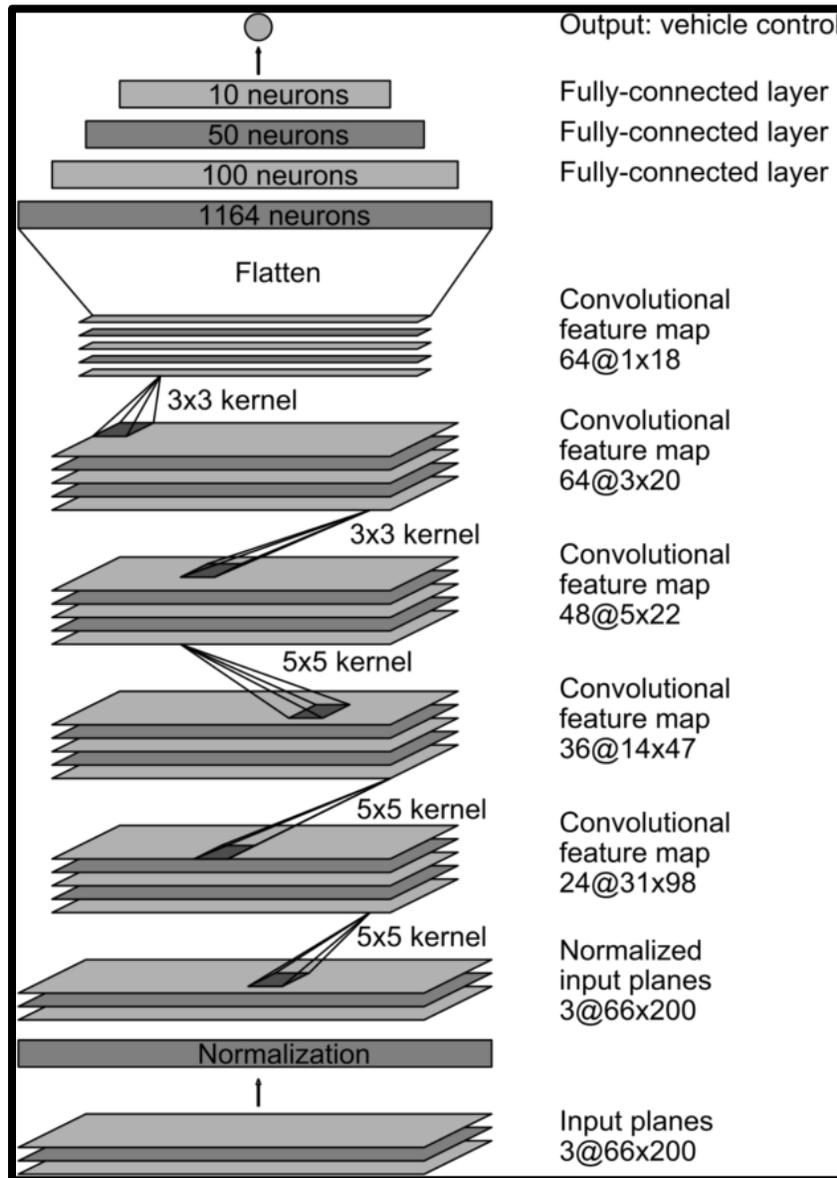


**Figura 28-2.** Interfaz Raspbian.

Realizado por: Yandry Jácome, 2021.

## 2.14. Red neuronal convolucional creada

Se creó una red neuronal convolucional basada en la arquitectura NVIDIA End-to-End Deep Learning for Self-Driving Cars como podemos ver en la figura 29-2.



**Figura 29-2.** CNN arquitectura. La red tiene aproximadamente 27 millones de conexiones y 250 mil parámetros.

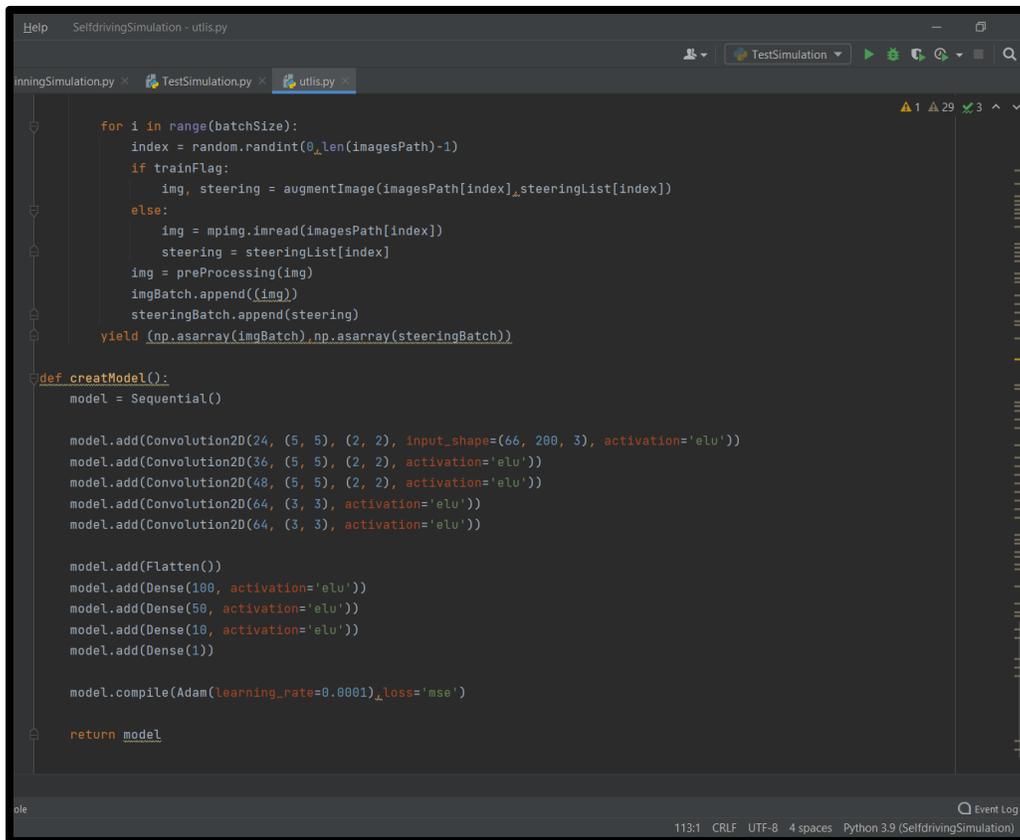
**Realizado por:** Yandry Jácome, 2021.

La primera capa de la red realiza la normalización de imágenes. El normalizador está codificado de forma rígida y no se ajusta en el proceso de aprendizaje. Realizar la normalización en la red permite modificar el esquema de normalización con la arquitectura de la red y acelerarlo mediante el procesamiento de la GPU.

Las capas convolucionales están diseñadas para realizar la extracción de características y se eligen empíricamente a través de una serie de experimentos que varían las configuraciones de las capas. Luego usamos convoluciones en zancadas en las primeras tres capas convolucionales con una zancada de  $2 \times 2$  y un núcleo de  $5 \times 5$ , y una convolución sin zancadas con un tamaño de grano de  $3 \times 3$  en las dos últimas capas convolucionales.

Seguimos las cinco capas convolucionales con tres capas completamente conectadas, lo que lleva a un valor de control de salida final que es el radio de giro inverso. Las capas completamente conectadas están diseñadas para funcionar como un controlador para la dirección, pero notamos que, al entrenar el sistema de un extremo a otro, no es posible hacer una separación clara entre qué partes de la red funcionan principalmente como extractor de características, y que sirven como controlador.

En la figura 30-2 podemos observar el modelo NVIDIA implementado en código para el entrenamiento.



```
for i in range(batchSize):
    index = random.randint(0, len(imagesPath)-1)
    if trainFlag:
        img, steering = augmentImage(imagesPath[index], steeringList[index])
    else:
        img = mpimg.imread(imagesPath[index])
        steering = steeringList[index]
        img = preProcessing(img)
        imgBatch.append((img))
        steeringBatch.append(steering)
yield (np.asarray(imgBatch), np.asarray(steeringBatch))

def creatModel():
    model = Sequential()

    model.add(Convolution2D(24, (5, 5), (2, 2), input_shape=(66, 200, 3), activation='elu'))
    model.add(Convolution2D(36, (5, 5), (2, 2), activation='elu'))
    model.add(Convolution2D(48, (5, 5), (2, 2), activation='elu'))
    model.add(Convolution2D(64, (3, 3), activation='elu'))
    model.add(Convolution2D(64, (3, 3), activation='elu'))

    model.add(Flatten())
    model.add(Dense(100, activation='elu'))
    model.add(Dense(50, activation='elu'))
    model.add(Dense(10, activation='elu'))
    model.add(Dense(1))

    model.compile(Adam(learning_rate=0.0001), loss='mse')

    return model
```

**Figura 30-2.** Código del modelo NVIDIA

Realizado por: Yandry Jácome, 2021.

## 2.15. Red neuronal convolucional entrenada

Para el desarrollo de la red neuronal convolucional se usó el software libre, Udacity self driving car nano degree program, ya que librerías de imágenes de carreteras no existen, el software nos brinda una variedad de entornos virtuales en los cuales se puede sacar capturas de diferentes perspectivas de las autopistas conformadas por curvas, rectas, cambios de terreno, cambios de elevación, etc.

Como resultado obtuvimos un total aproximado de dieciséis mil imágenes donde tenemos las perspectivas para el análisis y entrenamiento de nuestra red neuronal con el modelo NVIDIA.

### 2.15.1. Preprocesamiento de imágenes

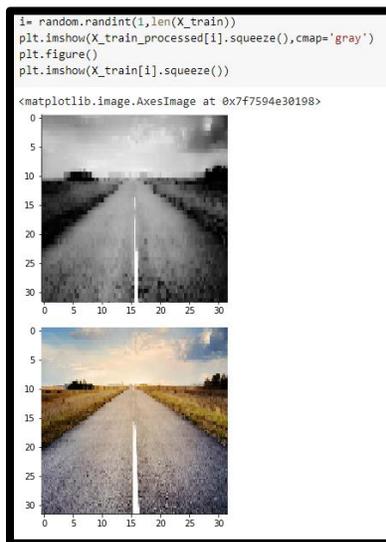
Se logra un correcto procesamiento de las imágenes al convertirlas a escala de grises para poder extraer su información con mayor facilidad con el código citado en la figura 2-21, seguido de un

proceso de ecualización del histograma donde se ajusta el contraste utilizando y por último un proceso de normalización los resultados se pueden observar en las figuras 31-2 y 32-2.

```
def preprocessing(img):  
    #convert to grayscale  
    img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)  
    #Applying histogram equalization  
    img = cv2.equalizeHist(img)  
    #Normalization  
    img = img/255  
    return img
```

**Figura 31-2.** Código aplicado para el procesamiento de las imágenes

**Realizado por:** Yandry Jácome, 2021.

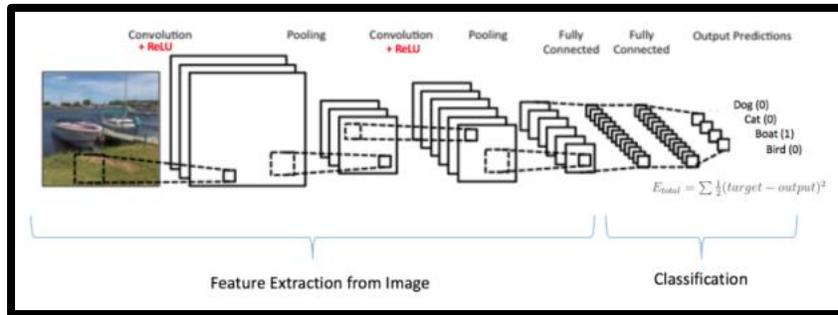


**Figura 32-2.** Ejecución del código de la figura 2-31

**Realizado por:** Yandry Jácome, 2021.

### 2.15.2. *Parámetros de entrenamiento de la red neuronal convolucional*

Las capas de convolución y de pooling actúan como extractores de características de la imagen de entrada mientras que la capa fully connected actúa como un clasificador, como se observa en la figura 33-2.



**Figura 33-2.** Esquema de las partes de una red convolucional.

Realizado por: Yandry Jácome, 2021.

El número de capas de convolución aumenta la precisión del modelo, así como también el número de épocas que se simula el modelo véase en la figura 2-34 y figura 2-35.

```

history = model.fit(X_train_processed,
                    y_train,
                    batch_size=500,
                    epochs=50,
                    verbose=1,
                    validation_data=(X_validation_processed,y_validation))

```

**Figura 34-2.** Código de parámetros de la red neuronal convolucional

Realizado por: Yandry Jácome, 2021.

```

Epoch 34/50 - 1s 14ms/step - loss: 0.3341 - accuracy: 0.9010 - val_loss: 0.3253 - val_accuracy: 0.9068
Epoch 35/50 - 1s 14ms/step - loss: 0.3220 - accuracy: 0.9056 - val_loss: 0.3156 - val_accuracy: 0.9079
Epoch 36/50 - 1s 14ms/step - loss: 0.3139 - accuracy: 0.9059 - val_loss: 0.3124 - val_accuracy: 0.9063
Epoch 37/50 - 1s 14ms/step - loss: 0.2990 - accuracy: 0.9084 - val_loss: 0.3052 - val_accuracy: 0.9127
Epoch 38/50 - 1s 14ms/step - loss: 0.2846 - accuracy: 0.9150 - val_loss: 0.2949 - val_accuracy: 0.9143
Epoch 39/50 - 1s 14ms/step - loss: 0.2841 - accuracy: 0.9177 - val_loss: 0.2883 - val_accuracy: 0.9136
Epoch 40/50 - 1s 14ms/step - loss: 0.2768 - accuracy: 0.9164 - val_loss: 0.2874 - val_accuracy: 0.9175
Epoch 41/50 - 1s 14ms/step - loss: 0.2739 - accuracy: 0.9170 - val_loss: 0.2796 - val_accuracy: 0.9166
Epoch 42/50 - 1s 14ms/step - loss: 0.2664 - accuracy: 0.9233 - val_loss: 0.2725 - val_accuracy: 0.9247
Epoch 43/50 - 1s 14ms/step - loss: 0.2498 - accuracy: 0.9261 - val_loss: 0.2717 - val_accuracy: 0.9209
Epoch 44/50 - 1s 14ms/step - loss: 0.2451 - accuracy: 0.9284 - val_loss: 0.2653 - val_accuracy: 0.9243
Epoch 45/50 - 1s 14ms/step - loss: 0.2412 - accuracy: 0.9267 - val_loss: 0.2657 - val_accuracy: 0.9231
Epoch 46/50 - 1s 14ms/step - loss: 0.2341 - accuracy: 0.9303 - val_loss: 0.2578 - val_accuracy: 0.9254
Epoch 47/50 - 1s 14ms/step - loss: 0.2201 - accuracy: 0.9362 - val_loss: 0.2560 - val_accuracy: 0.9240
Epoch 48/50 - 1s 14ms/step - loss: 0.2169 - accuracy: 0.9364 - val_loss: 0.2610 - val_accuracy: 0.9220
Epoch 49/50 - 1s 14ms/step - loss: 0.2104 - accuracy: 0.9389 - val_loss: 0.2477 - val_accuracy: 0.9265
Epoch 50/50 - 1s 14ms/step - loss: 0.2100 - accuracy: 0.9381 - val_loss: 0.2377 - val_accuracy: 0.9290

```

**Figura 35-2.** Épocas y precisión obtenida

Realizado por: Yandry Jácome, 2021.

## CAPÍTULO III

### 3. PRUEBAS Y RESULTADOS DEL PROTOTIPO

Los hallazgos obtenidos por el prototipo se revisan en esta sección para garantizar que se hayan cumplido las especificaciones, se hayan realizado pruebas de rango de detección de carril y se haya realizado una respuesta de prevención de salida de carril. También se han realizado pruebas de autonomía y de la precisión de la red neuronal convolucional.

#### 3.1. Análisis de la precisión de la red neuronal convolucional por épocas

Al ejecutar el código y tras 100 épocas con un lote de 500 imágenes se reflejan los siguientes datos mostrados en la siguiente tabla 1-3.

**Tabla 1-3:** Análisis de la precisión de la red neuronal

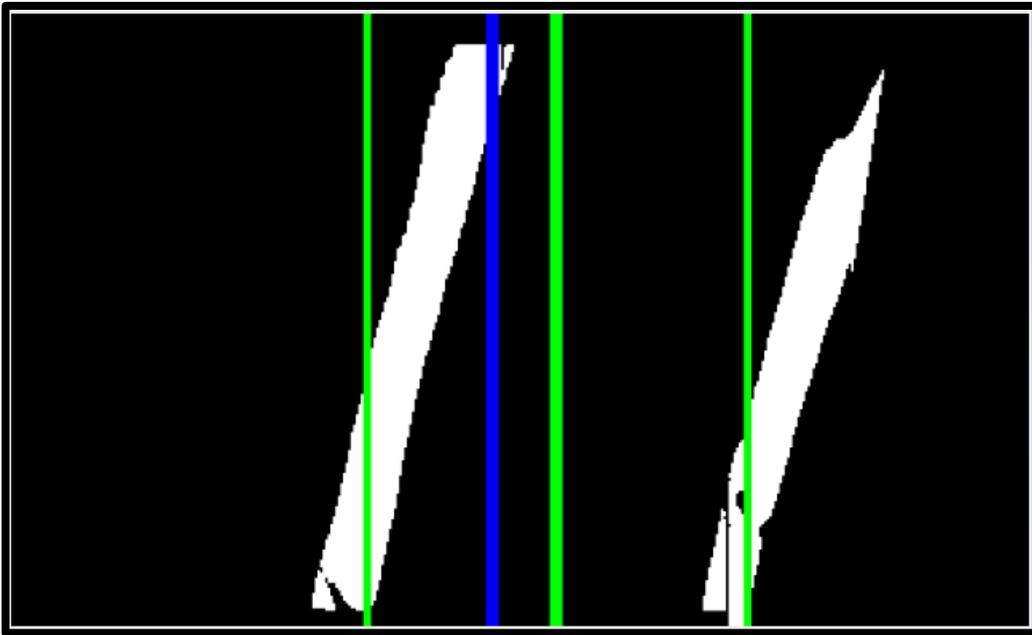
ÉPOCAS	PERDIDAS	PRECISIÓN	TIEMPO DE EJECUCIÓN
1	0.0447	0.9871	1s 52ms/step
2	0.0439	0.9867	1s 50ms/step
3	0.04551	0.9869	1s 50ms/step
4	0.0441	0.9871	1s 50ms/step
5	0.041	0.9883	1s 50ms/step
6	0.0431	0.988	1s 50ms/step
7	0.0419	0.9878	1s 50ms/step
8	0.0447	0.9871	1s 50ms/step
9	0.0411	0.9872	1s 50ms/step
10	0.042	0.988	1s 50ms/step
11	0.0419	0.9882	1s 50ms/step
12	0.0413	0.9886	1s 50ms/step
13	0.0399	0.989	1s 50ms/step
14	0.0404	0.989	1s 50ms/step
15	0.0417	0.9886	1s 50ms/step
16	0.041	0.9883	1s 50ms/step
17	0.411	0.988	1s 50ms/step
18	0.395	0.9886	1s 50ms/step
19	0.395	0.9891	1s 50ms/step
20	0.393	0.9888	1s 50ms/step
21	0.041	0.9877	1s 50ms/step
22	0.0389	0.9889	1s 50ms/step
23	0.0409	0.9879	1s 50ms/step
24	0.0393	0.9891	1s 50ms/step

25	0.0369	0.9896	1s 50ms/step
26	0.0384	0.9882	1s 50ms/step
27	0.0386	0.9891	1s 50ms/step
28	0.0362	0.9903	1s 50ms/step
29	0.0375	0.9894	1s 50ms/step
30	0.0357	0.9897	1s 50ms/step
31	0.0381	0.9889	1s 50ms/step
32	0.0379	0.989	1s 50ms/step
33	0.036	0.9895	1s 50ms/step
34	0.0353	0.9905	1s 50ms/step
35	0.0372	0.9889	1s 50ms/step
36	0.0338	0.9912	1s 50ms/step
37	0.0361	0.9894	1s 50ms/step
38	0.0364	0.9899	1s 50ms/step
39	0.035	0.9903	1s 50ms/step
40	0.0339	0.99	1s 50ms/step
41	0.0349	0.9902	1s 50ms/step
42	0.0323	0.9907	1s 50ms/step
43	0.033	0.9901	1s 50ms/step
44	0.036	0.9899	1s 50ms/step
45	0.0337	0.9906	1s 50ms/step
46	0.0341	0.9903	1s 50ms/step
47	0.0335	0.9901	1s 50ms/step
48	0.0325	0.9906	1s 50ms/step
49	0.0326	0.9907	1s 50ms/step
50	0.0324	0.9906	1s 50ms/step
51	0.0343	0.9904	1s 50ms/step
52	0.0338	0.9901	1s 50ms/step
53	0.0329	0.9907	1s 50ms/step
54	0.0319	0.9909	1s 49ms/step
55	0.0326	0.9908	1s 50ms/step
56	0.3	0.9916	1s 50ms/step
57	0.0317	0.9908	1s 50ms/step
58	0.0302	0.9915	1s 50ms/step
59	0.0312	0.9911	1s 50ms/step
60	0.0326	0.9906	1s 50ms/step
61	0.0304	0.9921	1s 49ms/step
62	0.0319	0.9911	1s 50ms/step
63	0.0309	0.9912	1s 50ms/step
64	0.0297	0.9918	1s 50ms/step
65	0.0294	0.9919	1s 50ms/step
66	0.0307	0.9911	1s 50ms/step
67	0.0289	0.9923	1s 50ms/step
68	0.0291	0.9916	1s 50ms/step
69	0.0299	0.9914	1s 50ms/step
70	0.029	0.9922	1s 50ms/step
71	0.0268	0.9926	1s 50ms/step
72	0.0279	0.9918	1s 50ms/step
73	0.0292	0.9916	1s 50ms/step

74	0.029	0.9923	1s 49ms/step
75	0.0292	0.9916	1s 50ms/step
76	0.0288	0.9918	1s 50ms/step
77	0.0271	0.992	1s 49ms/step
78	0.0297	0.9906	1s 50ms/step
79	0.0271	0.9928	1s 50ms/step
80	0.0264	0.9926	1s 50ms/step
81	0.0262	0.9923	1s 50ms/step
82	0.0286	0.9921	1s 50ms/step
83	0.0273	0.9926	1s 50ms/step
84	0.0258	0.9929	1s 50ms/step
85	0.0267	0.9926	1s 50ms/step
86	0.0271	0.992	1s 50ms/step
87	0.0259	0.993	1s 50ms/step
88	0.0259	0.9932	1s 50ms/step
89	0.0263	0.9928	1s 49ms/step
90	0.0268	0.992	1s 50ms/step
91	0.0263	0.9924	1s 50ms/step
92	0.0254	0.9927	1s 50ms/step
93	0.0258	0.9928	1s 50ms/step
94	0.0248	9930	1s 50ms/step
95	0.0248	0.9926	1s 49ms/step
96	0.0251	0.993	1s 50ms/step
97	0.0243	0.99932	1s 48ms/step
98	0.0243	0.9932	1s 50ms/step
99	0.0251	0.9927	1s 50ms/step
100	0.0225	0.9937	1s 50ms/step
promedio	0.0503761	0.9905	

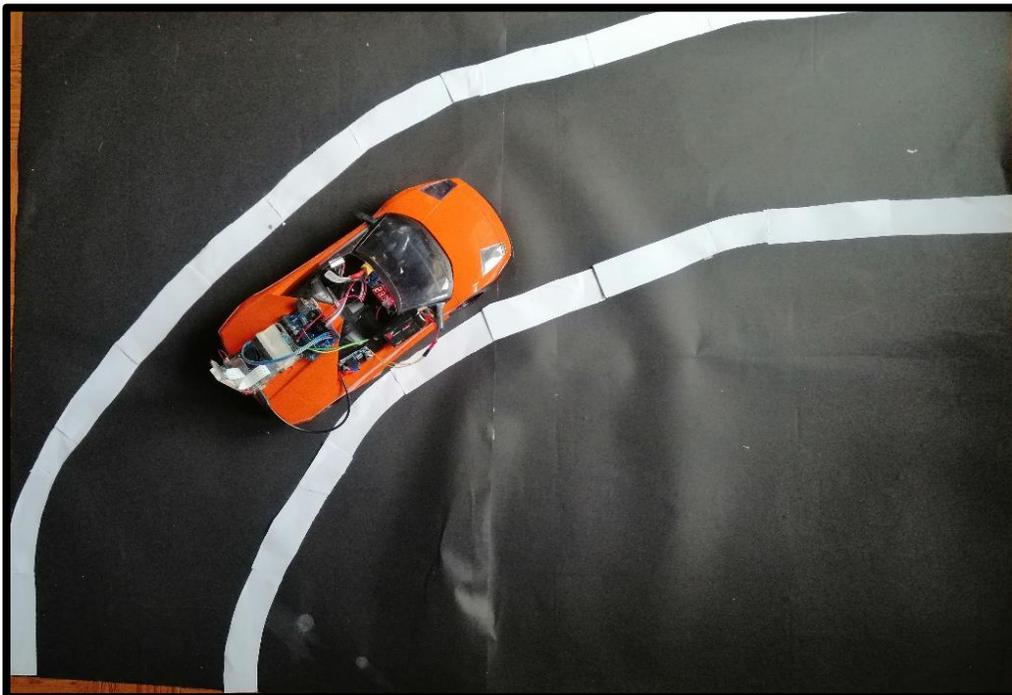
*Realizado por:* Yandry Jácome, 2021

Resultados del procesamiento de la imagen se observan en la figura 1-3 y figura 2-3.



**Figura 1-3.** Detección de líneas mediante visión artificial

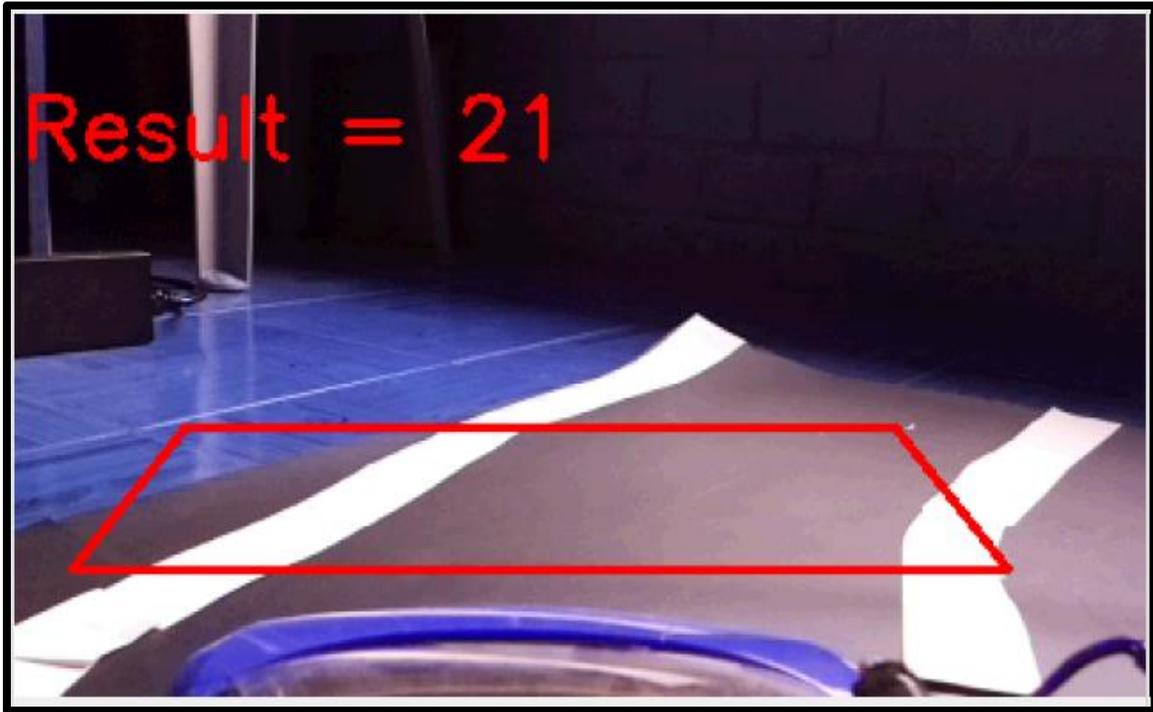
Realizado por: Yandry Jácome, 2021.



**Figura 2-3.** Posición del vehículo toma aérea

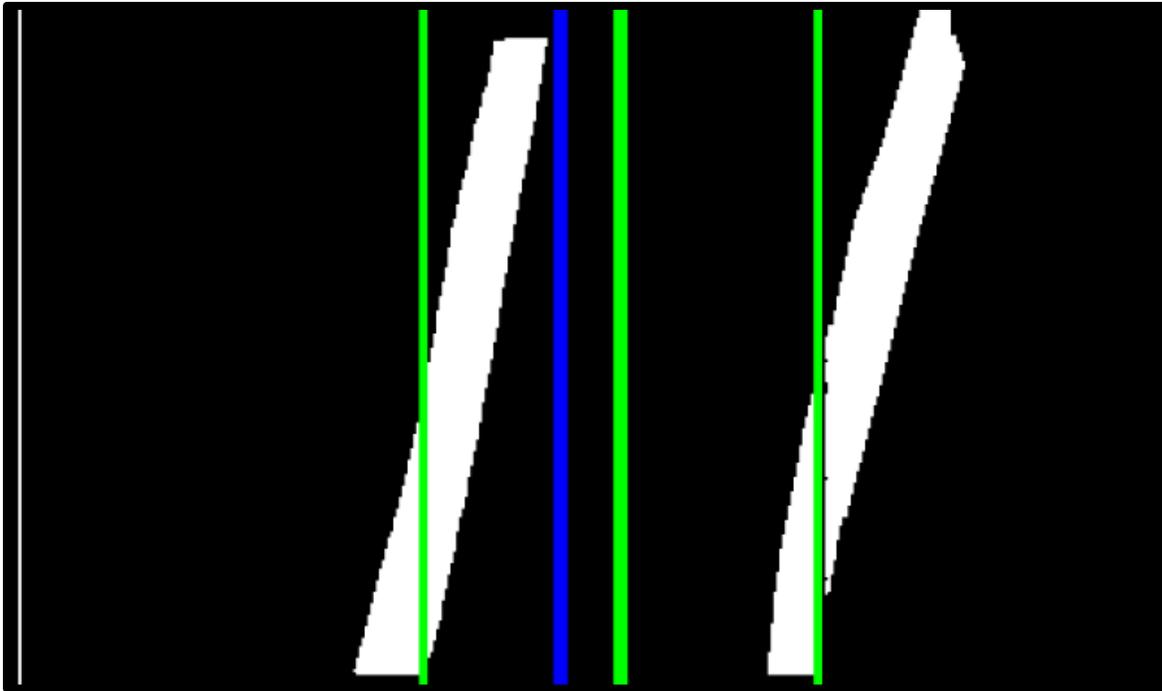
Realizado por: Yandry Jácome, 2021.

Según las pruebas realizadas la precisión aumenta mientras pasan las épocas, presentaremos el resultado en 300 épocas y con un lote de 16000 imágenes para observar la diferencia que causa un entrenamiento más extenuante, como se observa en la figura 3-3.



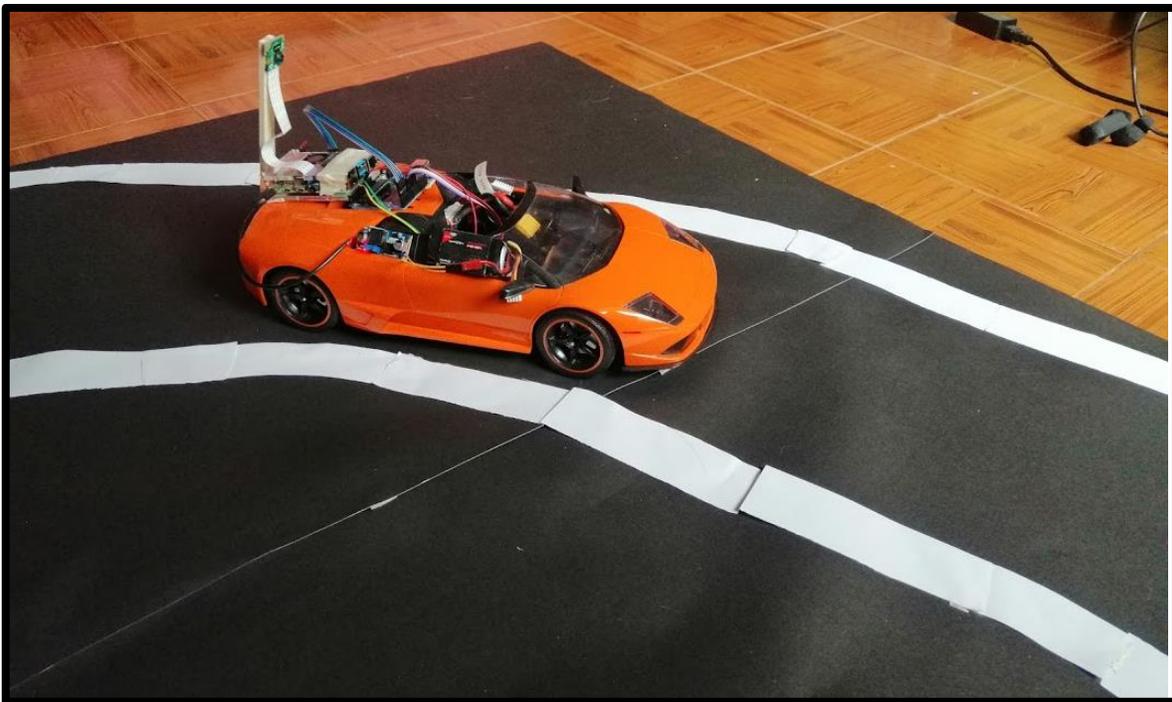
**Figura 3-3.** Perspectiva cámara a bordo

**Realizado por:** Yandry Jácome, 2021.



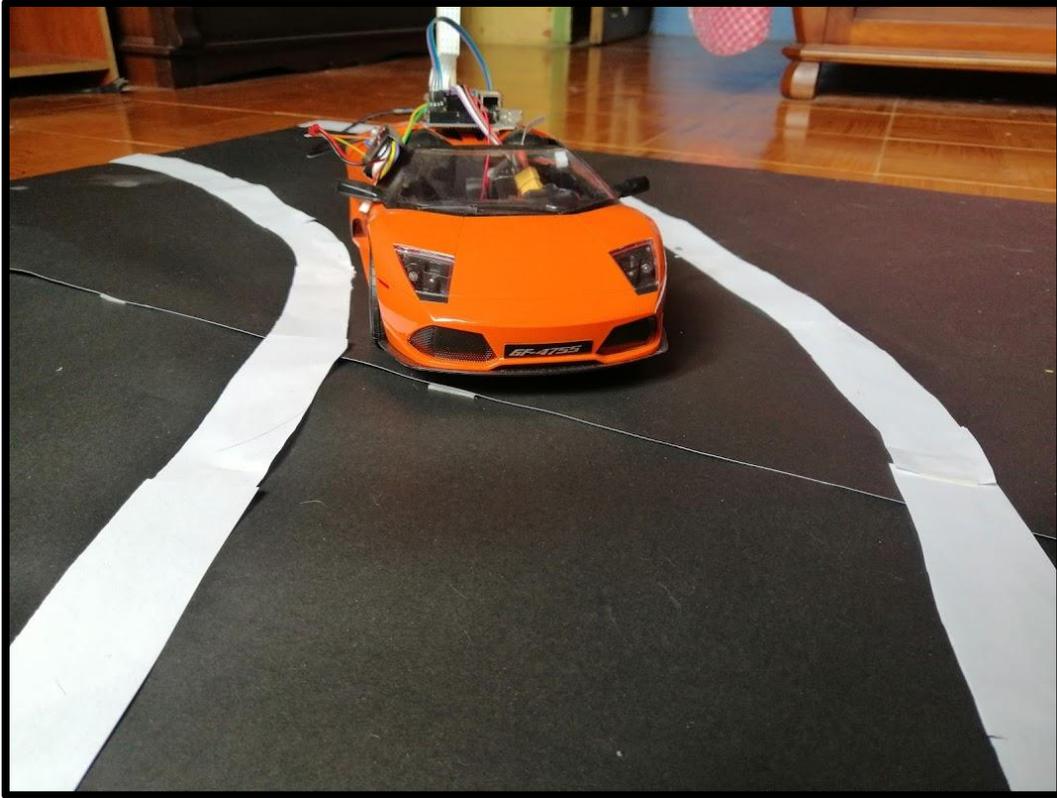
**Figura 4-3.** Resultado entrenamiento de 300 épocas con 16000 imágenes

Realizado por: Yandry Jácome, 2021.



**Figura 5-3.** Posición del Vehículo toma lateral

Realizado por: Yandry Jácome, 2021.



**Figura 6-3.** Posición vehículo toma frontal

Realizado por: Yandry Jácome, 2021.

### **3.2. Evaluación del desempeño del modelo**

En esta etapa de prueba se observa un significativo decrecimiento en la pérdida de información en cada época y la velocidad de procesamiento es mayor al anterior proceso de aprendizaje de la red neuronal convolucional, el acercamiento al valor de uno en la precisión es notorio en cada época.

Se realizó una prueba donde se suben los parámetros para llegar a una alta precisión de aprendizaje de la red neuronal convolucional, el resultado de este análisis con 20 épocas y un lote de 16000 imágenes se obtuvo un valor de 0.9898, muy cercanos a 1, las pruebas posteriores que se realizaron con 5 épocas y un lote de imágenes de 100 y 10 con un lote de imágenes de 2000 épocas obtuvieron una precisión de 0.8381 y 0.9069, esto nos dice que un número de épocas e imágenes es mejor ya que al momento de terminar el análisis la precisión llegó a 0.9898.

El punto negativo sería el tiempo de entrenamiento de la red neuronal convolucional, mientras que en la primera prueba de 5 épocas fue de apenas 54 segundos, la segunda prueba con 10 épocas es de

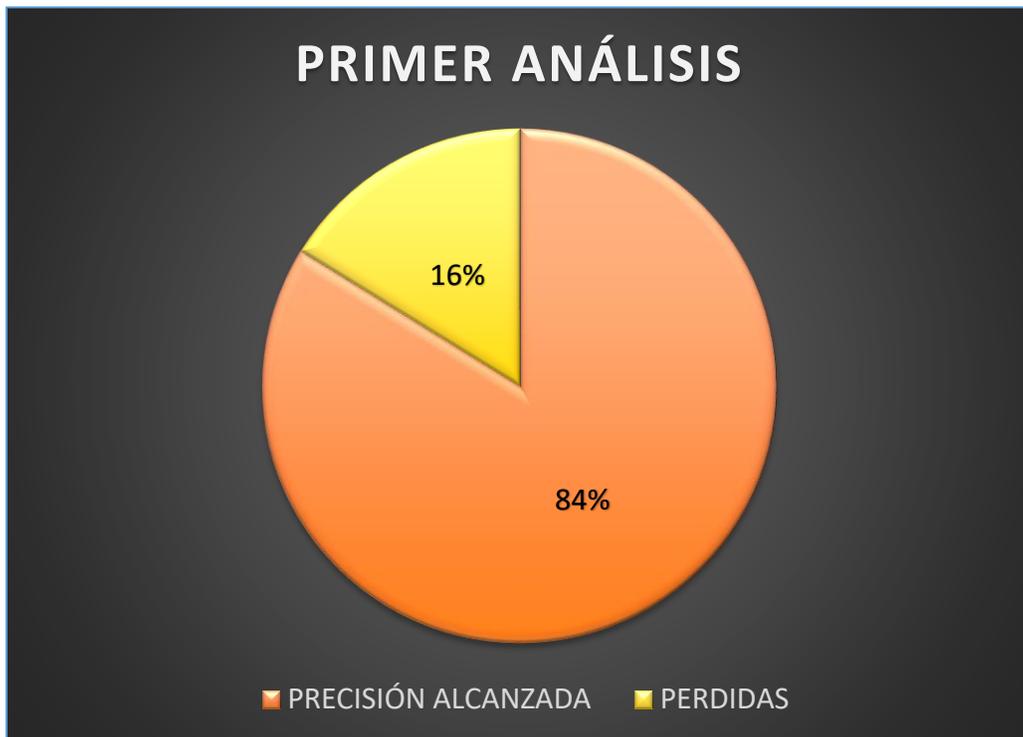
146 segundos y en la última prueba el tiempo de análisis son 2.6 horas, se muestra la relación en la siguiente tabla 2-3.

**Tabla 2-3:** Comparación de precisión alcanzada

ÉPOCAS ANALIZADAS	PASOS POR ÉPOCA	PRECISIÓN ALCANZADA	PERDIDAS	TIEMPO ESTIMADO DE ANÁLISIS	LOTE
5	300	0.8381	0.1619	54 s	100
10	300	0.9069	0.0931	146 s	2000
20	300	0.9898	0.0102	2,6 hora	16000

*Realizado por:* Yandry Jácome, 2021

En la figura 7-3, figura 8-3 y figura 9-3, se analiza la precisión de cada prueba realizada.



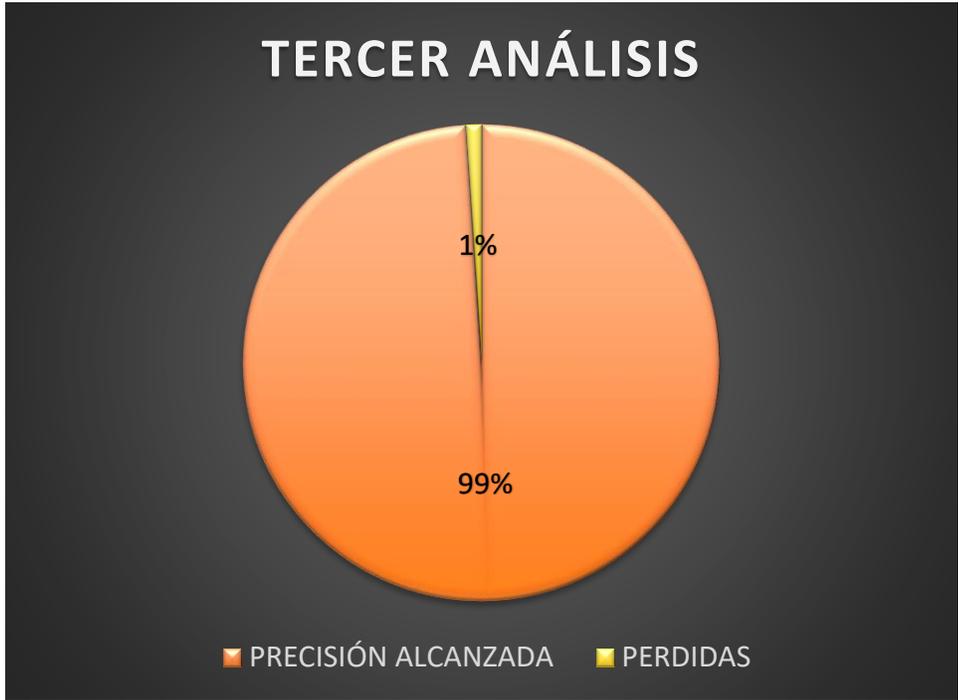
**Figura 7-3.** Fiabilidad del sistema, primer análisis

*Realizado por:* Yandry Jácome, 2021.



**Figura 8-3.** Fiabilidad del sistema, Segundo análisis

Realizado por: Yandry Jácome, 2021.

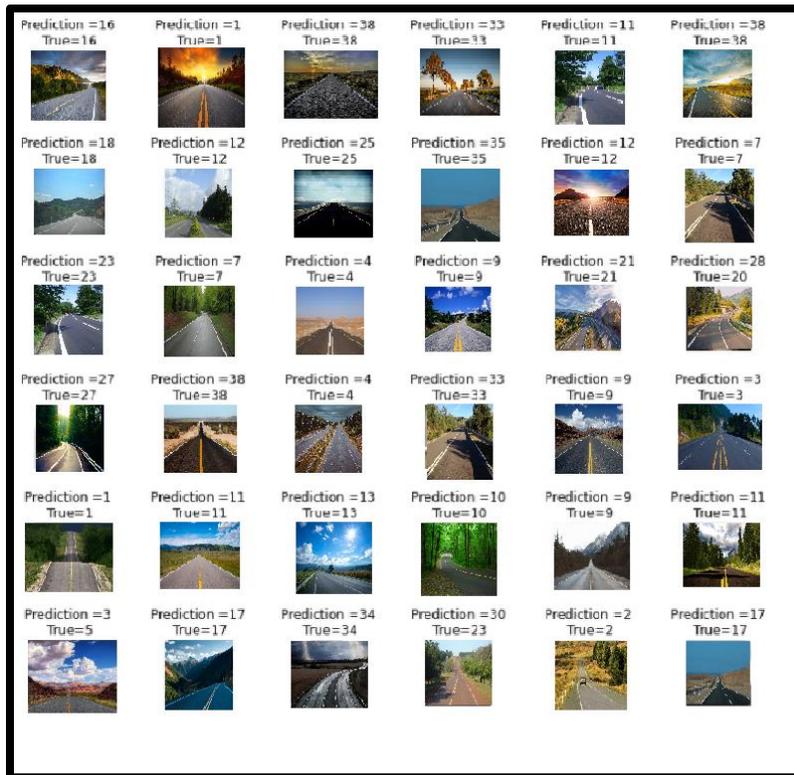


**Figura 9-3.** Fiabilidad del sistema, Tercer análisis

Realizado por: Yandry Jácome, 2021.

### 3.3. Predicciones realizadas.

Se saca una muestra, figura 7-3. para ver la precisión de la red neuronal convolucional con cada imagen.



**Figura 10-3.** Predicciones de CNN

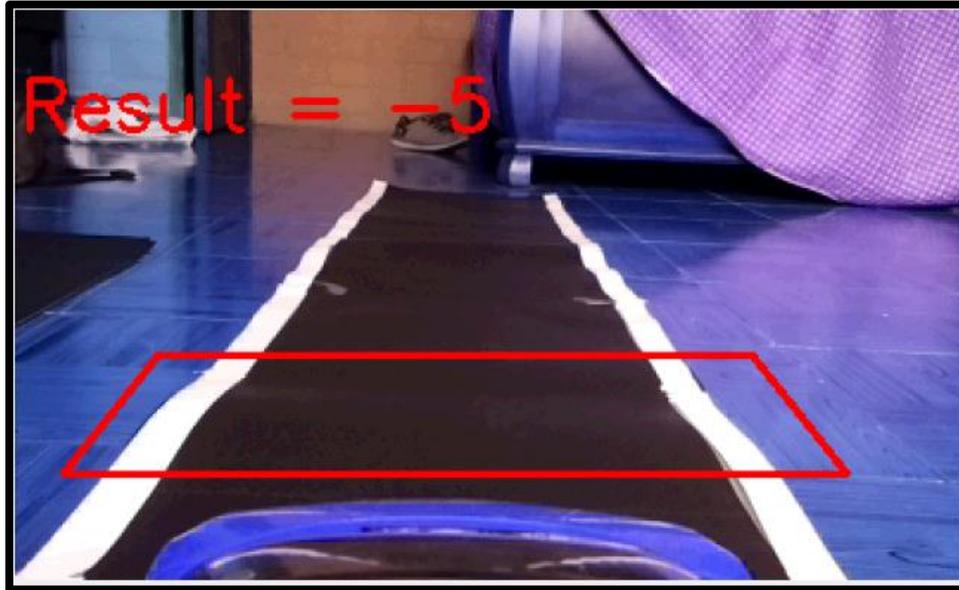
Realizado por: Yandry Jácome, 2021.

### 3.4. Prueba del sistema anti salida de carril.

El desarrollo del sistema anti salida de carril tuvo como objeto evitar la salida del vehículo en pruebas de conducción tanto en línea recta y la línea curva, como también verificar que el análisis de los datos obtenidos por la cámara sean los correctos para establecer el nivel de confianza del sistema.

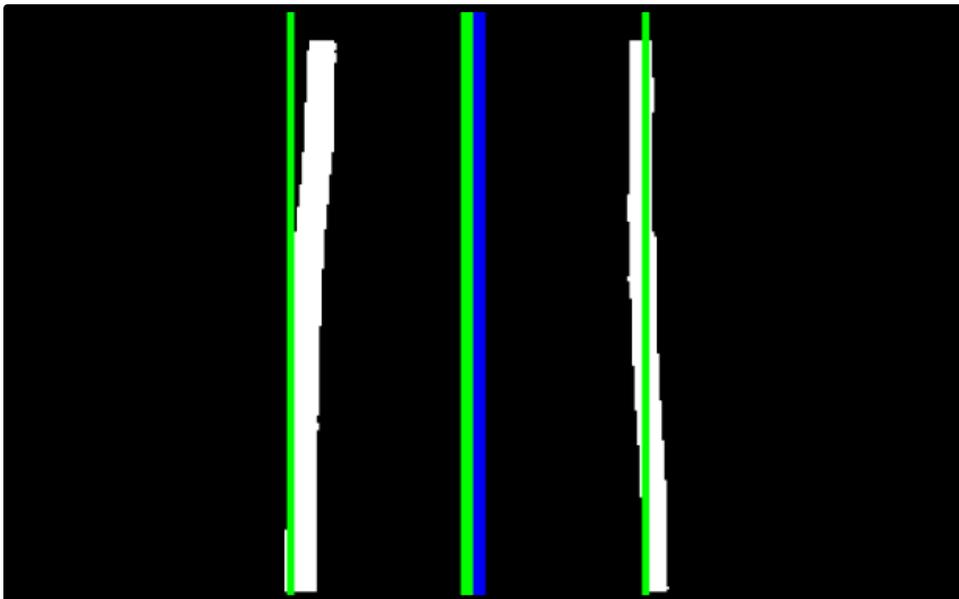
### 3.4.1. Prueba en pista línea recta

Para esta prueba se realizó una pista de aproximadamente de 2 metros de largo, delimitándola con líneas blancas para que se simule una carretera podemos ver la interpretación del sistema y del vehículo en las figuras 8-3 y 9-3 respectivamente



**Figura 11-3.** Cámara a bordo prueba en línea recta

Realizado por: Yandry Jácome, 2021.



**Figura 12-3.** Interpretación de la CNN de la prueba en línea recta

Realizado por: Yandry Jácome, 2021.

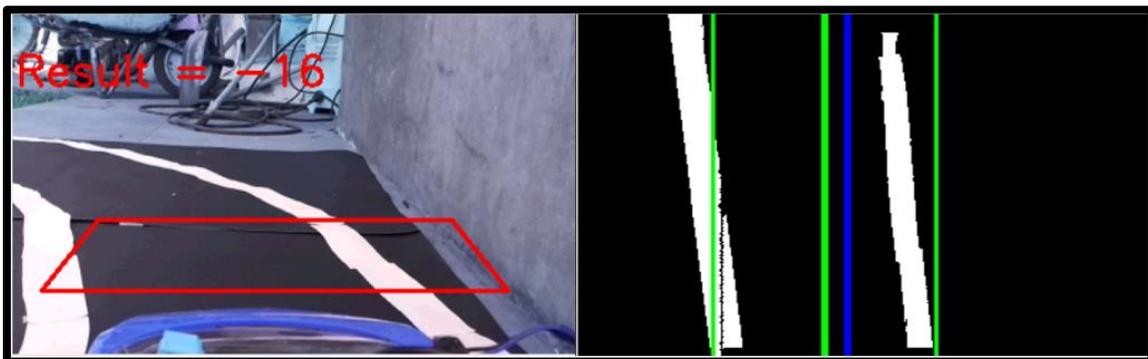
### 3.4.2. Prueba en pista línea curva

Para esta prueba se realizó una pista curva de aproximadamente 1.45 metros de largo, delimitándola con líneas blancas para que se simule una carretera podemos ver la interpretación del sistema y del vehículo en las figuras 10-3 a la 25-3 respectivamente.



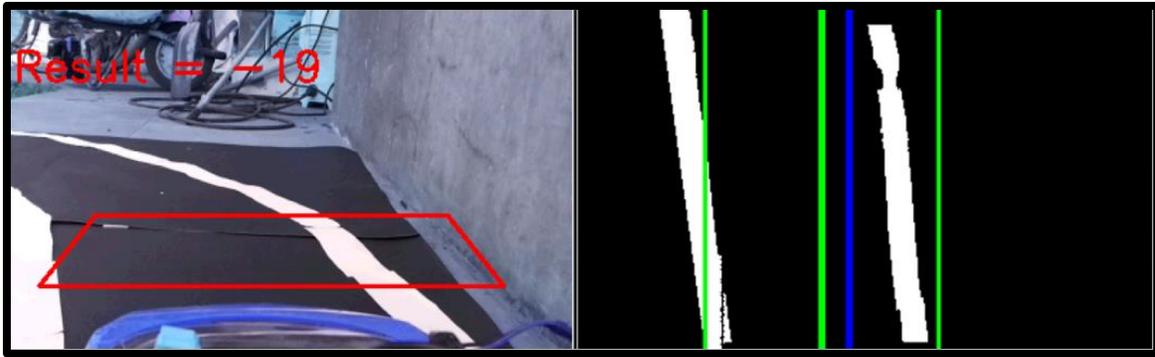
**Figura 13-3.** a. Trayectoria del vehículo en curva step 1, b. Comportamiento de la CNN

Realizado por: Yandry Jácome, 2021.



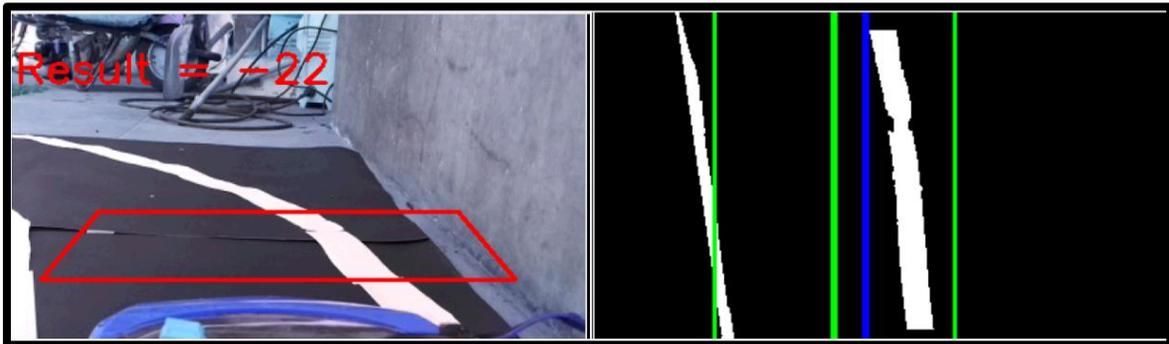
**Figura 14-3.** a. Trayectoria del vehículo en curva step 2, b. Comportamiento de la CNN

Realizado por: Yandry Jácome, 2021.



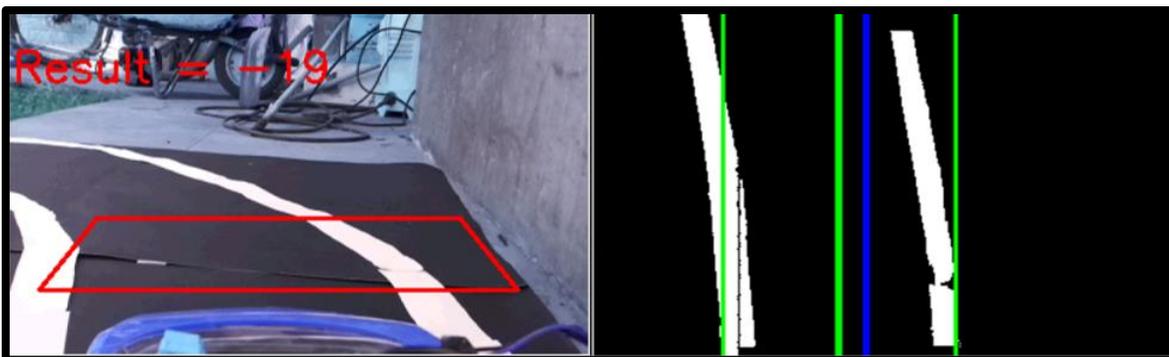
**Figura 15-3.** a. Trayectoria del vehículo en curva step 3, b. Comportamiento de la CNN

Realizado por: Yandry Jácome, 2021.



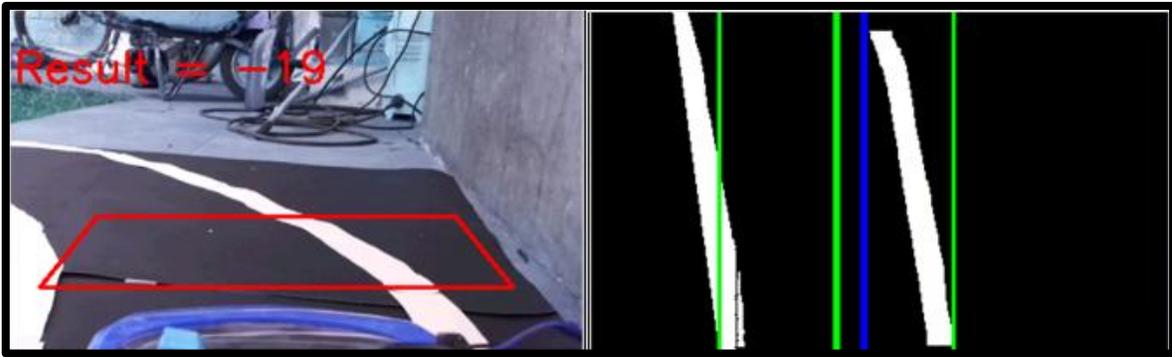
**Figura 16-3.** a. Trayectoria del vehículo en curva step 4, b. Comportamiento de la CNN

Realizado por: Yandry Jácome, 2021.



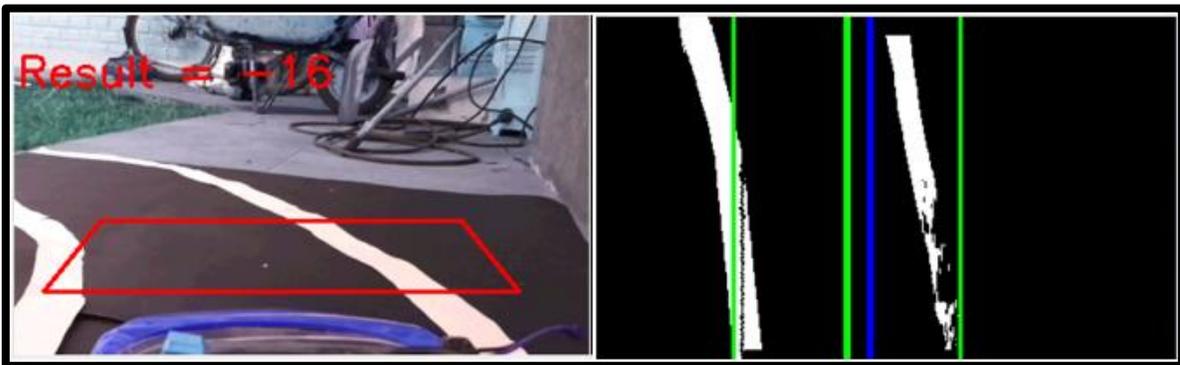
**Figura 17-3.** a. Trayectoria del vehículo en curva step 5, b. Comportamiento de la CNN

Realizado por: Yandry Jácome, 2021.



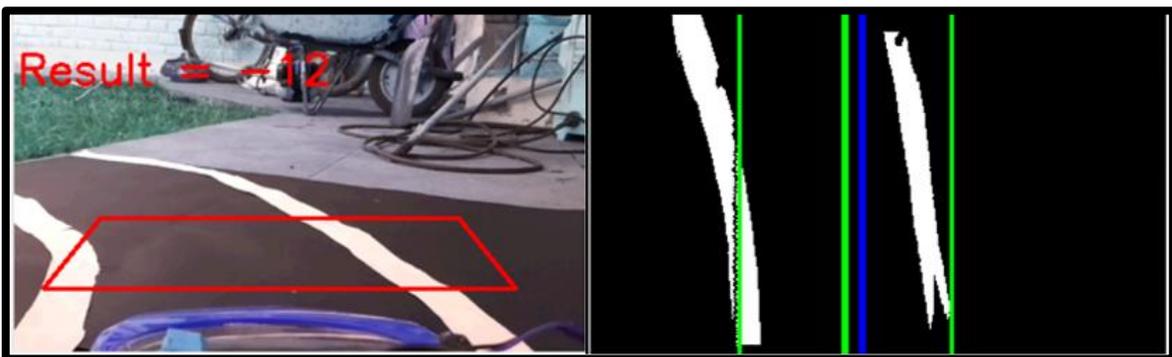
**Figura 18-3.** a. Trayectoria del vehículo en curva step 6, b. Comportamiento de la CNN

Realizado por: Yandry Jácome, 2021.



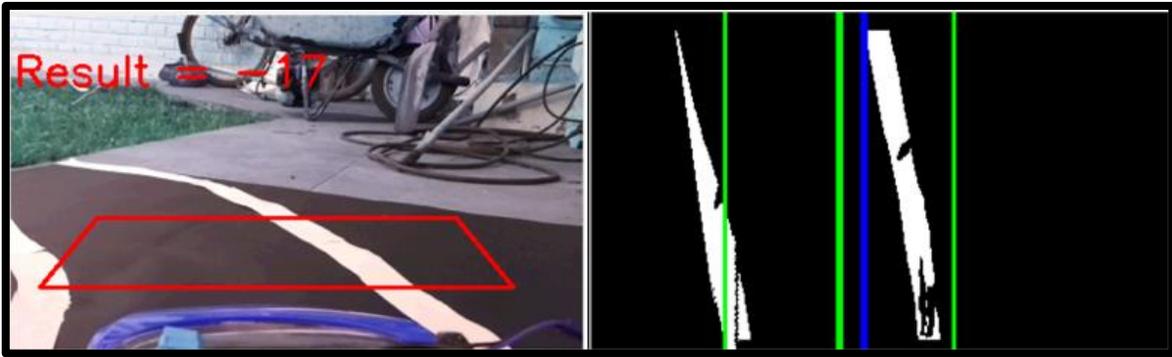
**Figura 19-3.** a. Trayectoria del vehículo en curva step 7, b. Comportamiento de la CNN

Realizado por: Yandry Jácome, 2021.

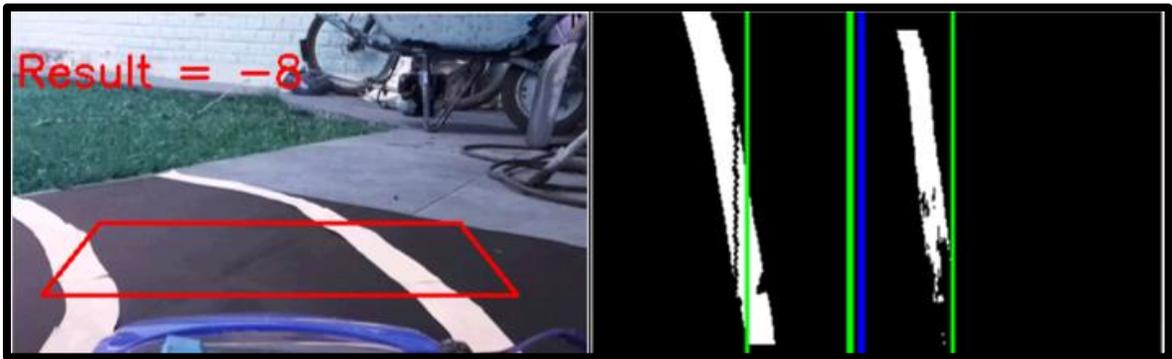


**Figura 20-3.** a. Trayectoria del vehículo en curva step 8, b. Comportamiento de la CNN

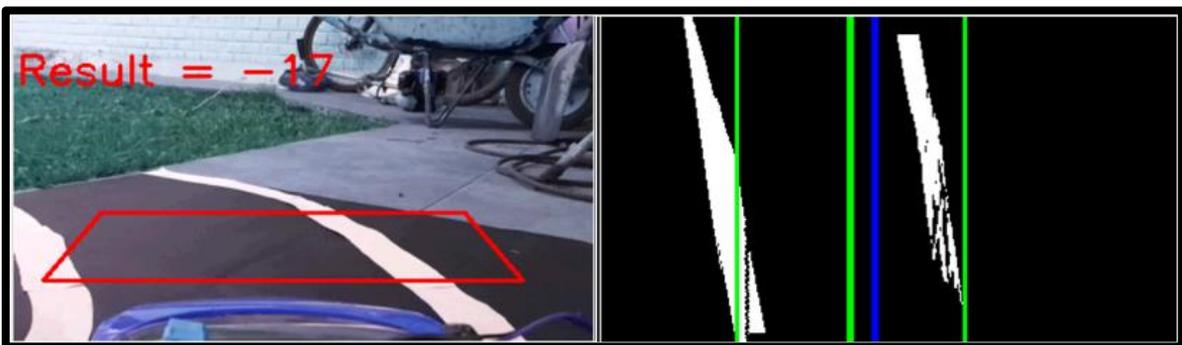
Realizado por: Yandry Jácome, 2021.



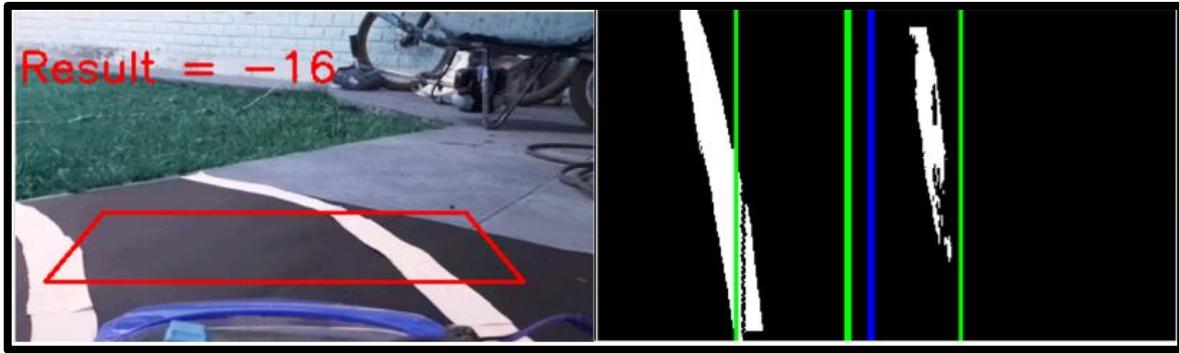
**Figura 21-3.** a. Trayectoria del vehículo en curva step 9, b. Comportamiento de la CNN  
 Realizado por: Yandry Jácome, 2021.



**Figura 22-3.** a. Trayectoria del vehículo en curva step 10, b. Comportamiento de la CNN  
 Realizado por: Yandry Jácome, 2021.

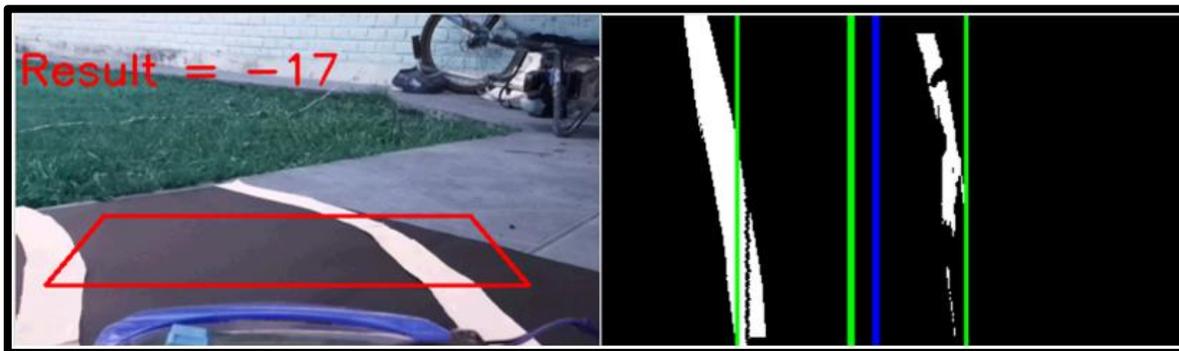


**Figura 23-3.** a. Trayectoria del vehículo en curva step 11, b. Comportamiento de la CNN  
 Realizado por: Yandry Jácome, 2021.



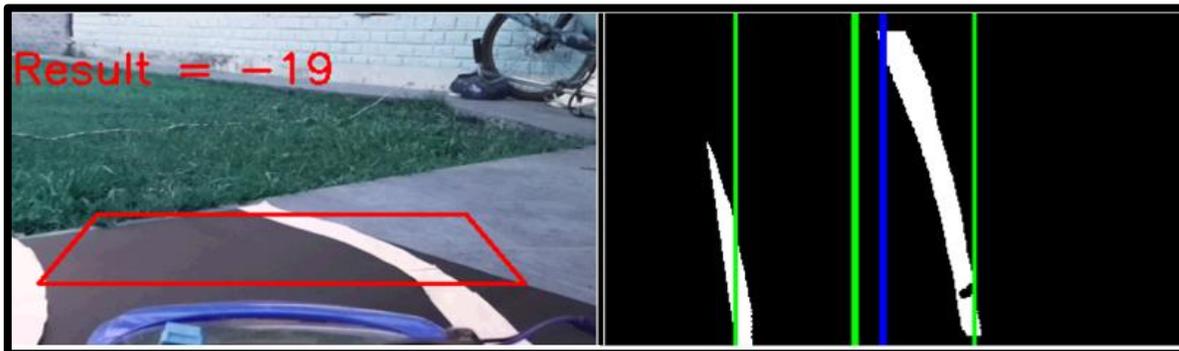
**Figura 24-3.** a. Trayectoria del vehículo en curva step 12, b. Comportamiento de la CNN

Realizado por: Yandry Jácome, 2021.



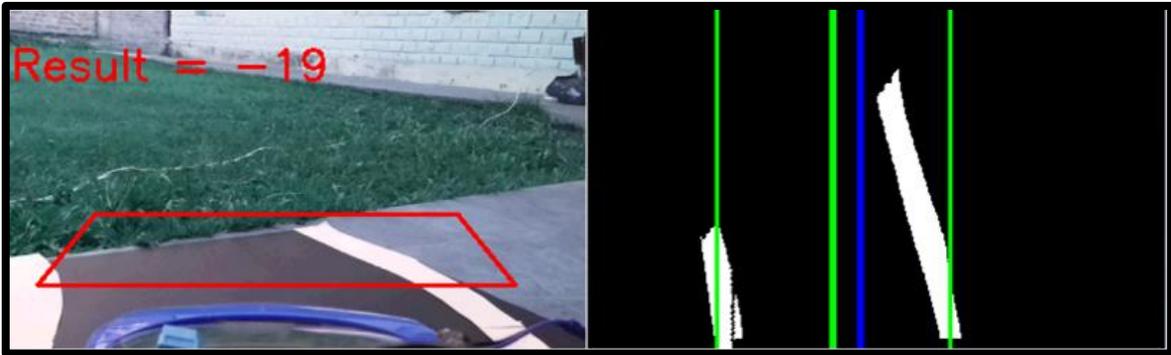
**Figura 25-3.** a. Trayectoria del vehículo en curva step 13, b. Comportamiento de la CNN

Realizado por: Yandry Jácome, 2021.



**Figura 26-3.** a. Trayectoria del vehículo en curva step 14, b. Comportamiento de la CNN

Realizado por: Yandry Jácome, 2021.



**Figura 27-3.** a. Trayectoria del vehículo en curva step 15, b. Comportamiento de la CNN

Realizado por: Yandry Jácome, 2021.

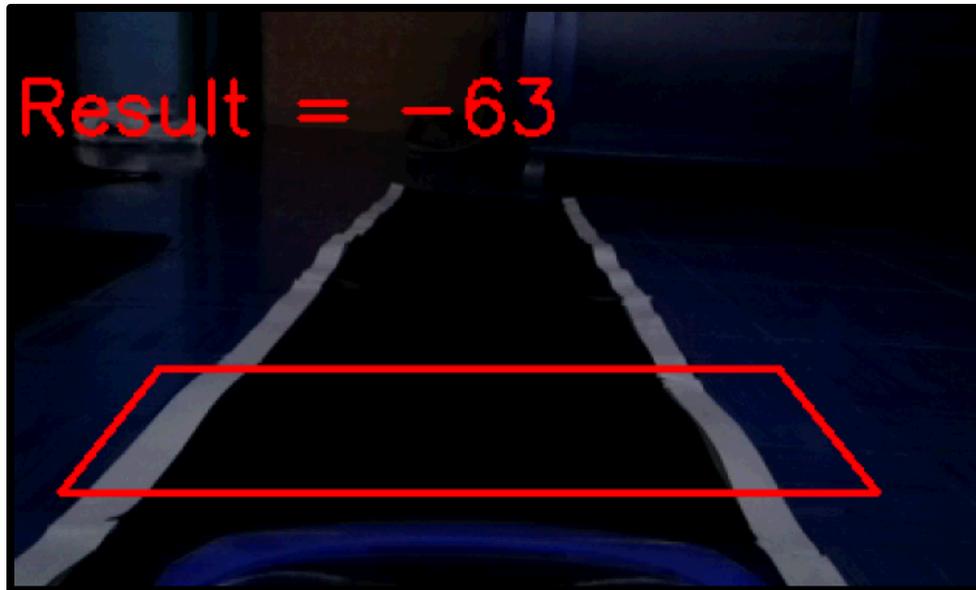


**Figura 28-3.** a. Trayectoria del vehículo en curva step 16, b. Comportamiento de la CNN

Realizado por: Yandry Jácome, 2021.

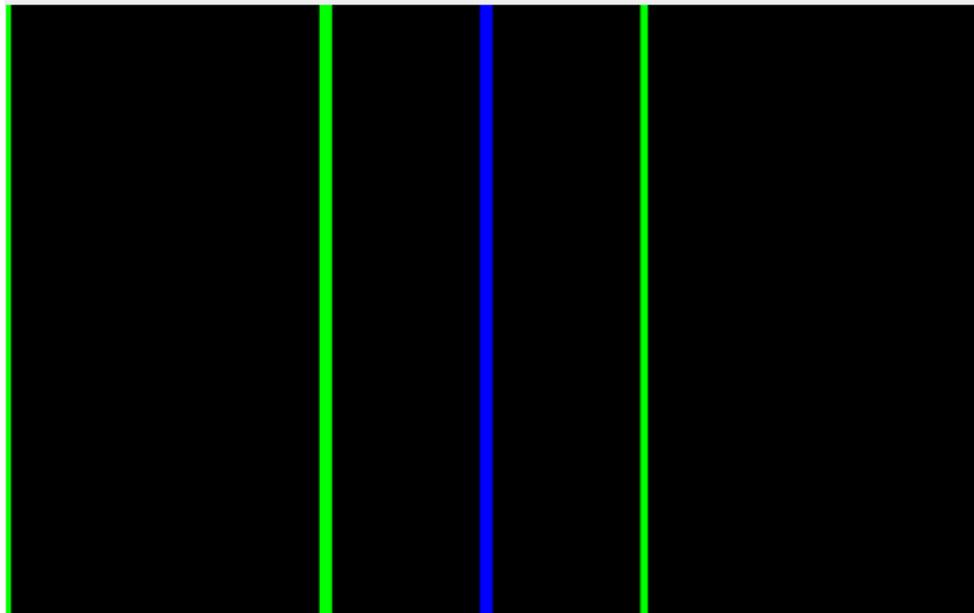
### 3.5. Prueba del sistema de visión.

El sistema de visión depende en gran parte de la iluminación tanto ambiental como artificial, ya que si no es la adecuada se tiende a perder detección de las líneas que delimitan el carril de la carretera, pero la red neuronal convolucional hace una interpretación del trayecto de la autopista como podemos apreciar en la figura 26-3 y en la figura 27-3.



**Figura 29-3.** Comportamiento del vehículo con poca luz

Realizado por: Yandry Jácome, 2021.

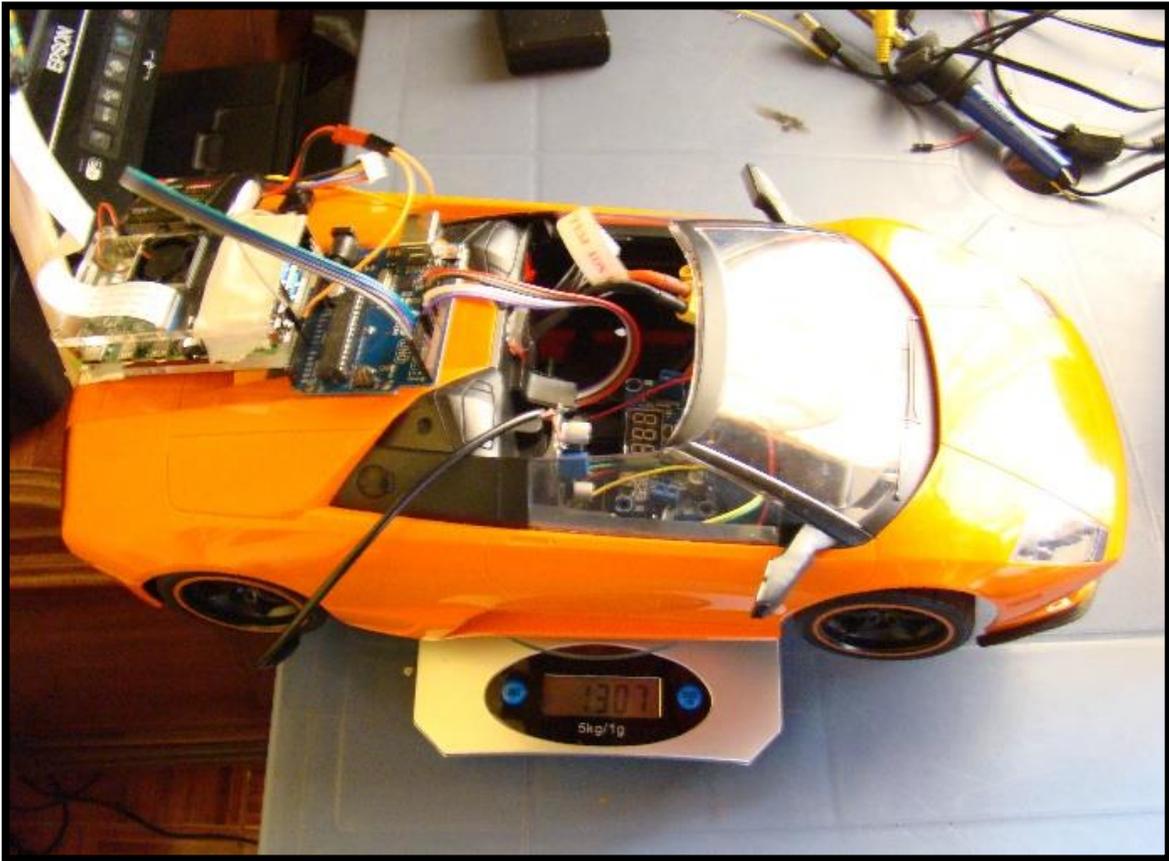


**Figura 30-3.** Comportamiento de la CNN con poca luz

Realizado por: Yandry Jácome, 2021.

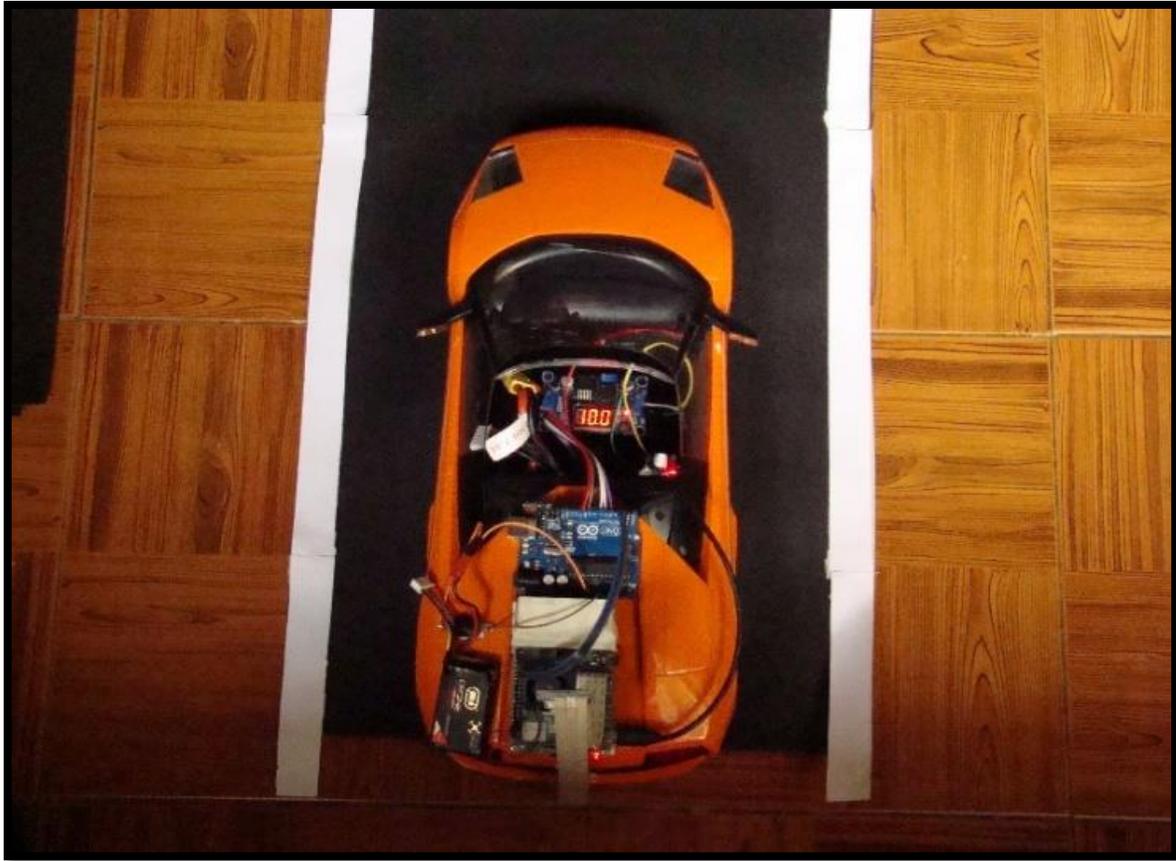
### 3.6. Prueba de peso y autonomía del Prototipo.

Se han recopiló 8 mediciones comparando la autosuficiencia del vehículo y el peso adicional para tener el tiempo de autonomía del prototipo, el tiempo de autonomía es inversamente proporcional al peso de este, En la figura 28-3 y en la figura 29-3, se determinó que el peso máximo es de 2kg para no comprometer la integridad estructural del prototipo con una autonomía de 20 minutos, cumpliendo con los requerimientos planteados



**Figura 31-3.** Peso del prototipo

Realizado por: Yandry Jácome, 2021.



**Figura 32-3.** Autonomía del prototipo

Realizado por: Yandry Jácome, 2021.

Antes de su composición se analizó el peso del prototipo que se muestra en la tabla 3-3.

**Tabla 3-3:** Peso de componentes del prototipo

COMPONENTE	PESO
Chasis	845g
Baterías	125g
Raspberry	40g
Arduino	16g
Puente H	8g
Reguladores	5g

Cableado	7g
Motores	261g
Peso total	1307g

*Realizado por:* Yandry Jácome, 2021

El comportamiento del prototipo al tener un peso extra se muestran en la tabla 4-3.

**Tabla 4-3:** Análisis de autonomía del prototipo con peso adicional

PRUEBA	PESO TOTAL	NOVEDAD	AUTONOMÍA
1	1307	Sin novedad	20 min
2	1400	Sin novedad	19 min
3	1450	Sin novedad	19 min
4	1500	Sin novedad	18 min
5	1550	Sin novedad	17 min
6	1600	Dificultad para arrancar	17 min
7	1800	Dificultad para arrancar	15 min
8	2000	Deformación estructural reversible	8 min

*Realizado por:* Yandry Jácome, 2021.

### 3.7. Análisis de tiempo de carga de la batería del Prototipo.

Para la recarga del prototipo se usó un cargado de baterías de 100W y depende mucho de la cantidad de amperios configurados al momento de hacer la recarga teniendo esto en cuenta tenemos la siguiente tabla 5-3, donde detallamos el tiempo de recarga en relación con los amperios administrados en 5 muestras.

**Tabla 5-3:** Registro de tiempo de carga en relación con los miliamperios aplicados

<b>MUESTRA</b>	<b>AMPERIOS</b>	<b>TIEMPO</b>
1	0.1 mA	2horas
2	1mA	24min
3	2 mA	12min
4	4 mA	8min
5	5 mA	6min

*Realizado por:* Yandry Jácome, 2021.

## CAPITULO IV

### 4. EVALUACIÓN ECONÓMICA DEL PROTOTIPO.

Esta sección se detalla el análisis del presupuesto referencial anticipado del prototipo del sistema completo, con el objetivo de compararlo con otros sistemas similares en el mercado ahora para evaluar su viabilidad y recursos de habilidades en comparación con estos sistemas.

#### 4.1. Presupuesto del Prototipo de Sistema

En la siguiente tabla 1-4, podemos observar el estudio de costos de cada componente aplicado al desarrollo del prototipo y gracias a los cuales se asegura el correcto funcionamiento de este.

**Tabla 1-4:** Estudio de costos de componentes requeridos

COMPONENETE	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
Chasis	1	49.99	49.99
Motor 14v 6000 rpm a 12000rpm	2	24.00	48.00
Batería lipo R-line 1500 mA 6s	1	90.00	90.00
Batería lipo venom 450 mA 3s	1	45.00	45.00
Cargador de Bateria Keenstone 100w	1	95.00	95.00
Puente H L298	1	5.00	5.00
Raspberry Pi 4	1	121.00	121.00
Arduino UNO	1	10.00	10.00
Cables	20	0.20	4.00
Raspicam V2	1	52.00	52.00
Regulador de Tension LM2596	2	4.00	8.00
TOTAL			527.99

*Realizado por:* Yandry Jácome, 2021

Como se puede observar en la tabla 1-4, el costo total del Prototipo de Sistema es de \$527.99. Este precio se encuentra dentro de un presupuesto general y accesible para su uso y empleo.

#### **4.2. Estudio comparativo con otros sistemas y prototipos similares**

Compararlo con otros sistemas es bastante desafiante, ya que no existe un sistema comparable en el mercado que posea todas las funciones y deberes que tiene el prototipo del sistema implementado, sin dejar de tener una autonomía completa o alguna función que el prototipo desarrollado en este trabajo incluya.

## CONCLUSIONES

- Se implementó un prototipo de sistema de visión artificial basado en redes convolucionales para el correcto posicionamiento de un vehículo autónomo en un carril.
- Al realizar las pruebas de ubicación del módulo RASPI CAM V2, se concluye que tanto una visión a la altura del parachoques frontal es desfavorable al momento de la captura de imágenes en relación con una imagen posterior.
- Al realizar el estudio de ubicación de los módulos del prototipo del sistema se concluyó que tanto el módulo de control y el módulo de procesamiento de las imágenes deben ir juntos el uno al otro, por su funcionalidad y en segundo lugar para tener un balance de peso del prototipo y evitar sobrecargas en los motores que eviten el funcionamiento correcto del prototipo.
- La precisión de la red neuronal convolucional presenta que mientras más números de épocas y muestras son obtenidas aumenta su precisión. En el último resultado se presenta una precisión del 99.87% con aproximadamente 2,7 horas de entrenamiento, para obtener este resultado se hizo la prueba con 300 de épocas y una muestra de 16000 imágenes.
- El uso de una red neuronal convolucional simplificó el proceso de aprendizaje del prototipo en relación con otros métodos de Machine learning como el método de cascada.
- La temperatura del módulo de procesamiento debe ser controlada por un cooler dada la gran exigencia de procesamiento a la que se somete ya que sin el cooler el módulo de procesamiento puede apagarse para evitar daños en su placa.

- La comunicación entre el maestro y el esclavo SPI fue seleccionada por la gran velocidad de transmisión de datos que tiene esta arquitectura y su facilidad de conexión.
- El módulo LM 2596 está integrado junto con un potenciómetro principalmente para monitorear el estado de la batería principal que alimentan los motores y tener un monitoreo del voltaje de salida hacia los motores.
- Al comparar el sistema implementado con otros similares en el mercado resulta un 232% más económico, además que el sistema construido puede ser adaptado a cualquier vehículo.

## RECOMENDACIONES

- Incorporar un sistema de iluminación en el prototipo que alumbré el camino que tiene por delante para evitar inconvenientes en el funcionamiento al momento de ausencia de luz.
- Se determina el uso de más cámaras en el prototipo que no sólo enfoquen hacia adelante sino también hacia los laterales para tener un mayor control y precisión del sistema anti salida de carril.
- Se determina emplear sensores de proximidad para evitar colisiones y que pueda desenvolverse en situaciones más complejas como, por ejemplo, en carriles que estén rodeados por paredes, vallas y cañones.
- Adaptar el prototipo contra diferentes condiciones climáticas
- Se determina reemplazar las tarjetas de desarrollo por su versión ligera para optimizar la autonomía y la estética en general del prototipo.
- Se determina el uso de motores brushless para ganar potencia y optimización de energía, así como, para mejorar la precisión de giro el uso de servomotores.
- Se determina la implementación de un PWM para el control de velocidad de giro de cada una de las ruedas delanteras para que el desplazamiento sea suave.

## GLOSARIO

**Aprendizaje automático:** La rama de la inteligencia artificial que implica la enseñanza de algoritmos para realizar tareas utilizando datos y ejemplos previos en lugar de órdenes explícitas programadas por humanos. Tres de los algoritmos más utilizados en el aprendizaje automático incluyen redes neuronales, algoritmos de inducción y algoritmos genéticos. Numerosas aplicaciones de inteligencia artificial se basan principalmente en el aprendizaje automático. Cuando las empresas analizan las capacidades de IA integradas en sus bienes y servicios, con frecuencia se refieren al aprendizaje automático.

**Redes neuronales:** Son los algoritmos de aprendizaje y los de los modelos informáticos que deben funcionar de manera similar a las neuronas del cerebro. Las redes neuronales se entrenan en ciertos conjuntos de datos que luego se utilizan para responder una consulta. La suposición de la red se compara con la respuesta precisa de la base de datos. Cuando ocurren errores, las "neuronas" se ajustan y el proceso se repite hasta que se reduce el nivel de error. Esta técnica algorítmica, denominada retro propagación, es análoga a la regresión estadística.

**Aprendizaje profundo:** Un subcampo del aprendizaje automático que se ocupa del diseño y entrenamiento de redes neuronales de múltiples capas. Cada capa de una red tiene la capacidad de detectar patrones en la salida de la capa superior. Las redes profundas se destacan en la clasificación de datos y la detección de anomalías.

**Visión de máquina:** El área de la inteligencia artificial se ocupa de cómo las computadoras imitan el sistema visual humano y su capacidad para ver e interpretar imágenes digitales del mundo real. Además, implica procesamiento de imágenes, reconocimiento de patrones y comprensión visual (convertir imágenes en descripciones que se pueden usar en otras aplicaciones).

**Sistemas expertos:** Además, se les conoce como sistemas de representación del conocimiento o sistemas de apoyo a la toma de decisiones. Los sistemas expertos son un tipo temprano de tecnología de inteligencia artificial que inicialmente estaba destinado a resolver problemas complicados

mediante el uso de una base de conocimientos y reglas para aplicar esa información. Los nuevos modelos de aprendizaje automático producen juicios más efectivos que los sistemas expertos debido a sus técnicas estadísticas más complejas y basadas en datos.

**Algoritmos de inducción:** Algoritmos que aprenden de ejemplos y buscan patrones en los datos para generar reglas que expliquen lo que está sucediendo. A diferencia del proceso deductivo, que utiliza un conjunto predefinido de reglas, estos algoritmos generan reglas para explicar los eventos a medida que ocurren.

## **BIBLIOGRAFIA**

**ALVES; et al.** The Newton Fractal's Leonardo Sequence Study with the Google Colab. *International Electronic Journal of Mathematics Education*. 2019. Vol. 15, no. 2. DOI 10.29333/iejme/6440.

**BENÍTEZ; el al.** *Inteligencia artificial avanzada*. 2014. Editorial.

**BERTOZZI; et al.** Artificial vision in road vehicles. *Proceedings of the IEEE*. 2002. Vol. 90, no. 7, p. 1258–1270. DOI 10.1109/JPROC.2002.801444.

**BODEN, Margaret A.**, 2016. *Inteligencia artificial*. Turner Pub.

**BRADSKI, G., & KAEHLER, A.**, 2008. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly M.

**BRADSKI; el al.** Histograms and Matching. *Learning OpenCV*. 2008. P. 193--.

**CABRERA MARINO & Karen Milagros.** Инновационные подходы к обеспечению качества в здравоохранении No Title. *Вестник Росздравнадзора*. 2017. Vol. 6, p. 5–9.

**CALDERÓN; et al.** La cámara digital como instrumento de laboratorio : estudio del tiro oblicuo. . 2008.

**CHANDANA; et al.** Smart Surveillance System using Thing Speak and Raspberry Pi. *International Journal of Advanced Research in Computer and Communication Engineering*. 2015. Vol. 4, no. 7, p. 214–218. DOI 10.17148/IJARCCE.2015.4749.

**COLOMBIANA, Revista**, 2017. Diseño De Un Sistema De Reconocimiento De Rostros Aplicando Inteligencia Y Visión Artificial. *Revista Colombiana De Tecnologías De Avanzada (Rcta)*. 2017. Vol. 2, no. 24. DOI 10.24054/01204211.v24.n24.2014.2342.

**CORTÉS, C.** Herramientas Modernas En Redes Neuronales: La Librería Keras. *Universidad Autónoma de Madrid* [online]. 2017. P. 60. Retrieved from:  
[https://repositorio.uam.es/bitstream/handle/10486/677854/antona\\_cortes\\_carlos\\_tfg.pdf?sequence=1&isAllowed=y](https://repositorio.uam.es/bitstream/handle/10486/677854/antona_cortes_carlos_tfg.pdf?sequence=1&isAllowed=y)

**DE, Rabajo and DE, I N.** Efecto de la sustitución de la función de Pooling en Redes Neuronales Convolucionales. . 2020.

**DILLON; et al.** Tensorflow distributions. *arXiv*. 2017.

**DITRICH, H.** A comparison of the renal structures of the anaconda and the Ball python. *Scanning Microscopy*. 1996. Vol. 10, no. 4, p. 1163–1172.

**DUARTE; et al.** Single-pixel imaging via compressive sampling: Building simpler, smaller, and less-expensive digital cameras. *IEEE Signal Processing Magazine*. 2008. Vol. 25, no. 2, p. 83–91. DOI 10.1109/MSP.2007.914730.

**DURÁN SUÁREZ; et al.** Redes Neuronales Convolucionales en R Reconocimiento de caracteres escritos a mano Redes Neuronales Convolucionales en R Reconocimiento de caracteres escritos a mano Redes Neuronales Convolucionales en R. [online]. 2017. P. 78. Retrieved from:  
<http://bibing.us.es/proyectos/abreproy/91338/fichero/TFG+Jaime+Durán+Suárez.pdf>

**DZIB GOODIN, Alma.** La evolución del aprendizaje: más allá de las redes neuronales. *Revista Chilena de Neuropsicología*. 2013. Vol. 8, no. 1, p. 20–25. DOI 10.5839/rcnp.2013.0801.04.

**FITRIA. 濟無**No Title No Title. *Journal of Chemical Information and Modeling*. 2013. Vol. 53, no. 9, p. 1689–1699.

**GARCÍA ELORRIAGA,Ricardo.** Localización por imagen de alta resolución del sistema cannabinoide endógeno en la mitocondria del músculo estriado de roedor TESIS DOCTORAL RICARDO ELORRIAGA GARCIA. . 2015. Vol. 2015, no. c.

**GÓMEZ RODRÍGUEZ; et al.** Computadoras de placa reducida Raspberry Pi 3 y Asus Tinker Board / Reduced plate computers Raspberry Pi 3 and Asus Tinker Board. *RECI Revista Iberoamericana de las Ciencias Computacionales e Informática*. 2018. Vol. 7, no. 14, p. 46–56. DOI 10.23913/reci.v7i14.86.

**GUERRA; et al.** Prototipo De Un Sistema Web Para El Conteo De Vehículos En Tiempo Real Utilizando La Librería De Visión Artificial Opencv. . 2016. P. 1–126.

**GULLI; et al.** Deep Learning with Keras. [online]. 2017. P. 318. Retrieved from: <https://www.packtpub.com/big-data-and-business-intelligence/deep-learning-keras>

**IZAURIETA; et al.** Redes Neuronales Artificiales. *Charlas de fisica*. 1999. P. 1–15. DOI 10.1016/S0210-5691(05)74198-X.

**JIN; et al.** Auto-keras: An efficient neural architecture search system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019. P. 1946–1956. DOI 10.1145/3292500.3330648.

**KEMENYASH; et al.** Multifunctional Robotic Complex Based on a Single-Board Computer. *Electronics and Control Systems*. 2018. Vol. 2, no. 56, p. 74–79. DOI 10.18372/1990-5548.56.12939.

**LI, Yohana; et al.** Vehículos autónomos: Innovación en la logística urbana. *Revista de Iniciación Científica*. 2018. Vol. 4, no. 1, p. 34–39. DOI 10.33412/rev-ric.v4.1.1864.

**LONCOMILLA, Patricio.** Deep learning: Redes convolucionales. *Recuperado de <https://ccc.inaoep.mx/~pgomez/deep/presentations>*. 2016.

**LÓPEZ, Mario.** Identificación de sistemas no lineales con redes neuronales convolucionales. . 2017. P. 103.

**MATICH, Damián Jorge.** Redes Neuronales: Conceptos Básicos y Aplicaciones. *Historia* [online]. 2001. P. 55. Retrieved from: <ftp://decsai.ugr.es/pub/usuarios/castro/Material-Redes-Neuronales/Libros/matich-redesneuronales.pdf>

**MOLES, A.** La IMAGEN Comunicación funcioanl. *Trillas* [online]. 2015. P. 19. Retrieved from: [https://s3.amazonaws.com/academia.edu.documents/54601969/13\\_IMAGEN.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1520894966&Signature=NZ%2Fxa003hns2LBp4w8rPVNOn%2FQw%3D&response-content-disposition=inline%3Bfilename%3DLA\\_IMAGEN\\_Comunicacion\\_funcional.p](https://s3.amazonaws.com/academia.edu.documents/54601969/13_IMAGEN.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1520894966&Signature=NZ%2Fxa003hns2LBp4w8rPVNOn%2FQw%3D&response-content-disposition=inline%3Bfilename%3DLA_IMAGEN_Comunicacion_funcional.p)

**NANCY, Jean-Luc.** La imagen: Mímesis & Méthexis. *La imagen: Mímesis & Méthexis*. 2006. Vol. 2, no. 2, p. 7–22. DOI 10.5209/rev\_ESIM.2006.v2.30268.

**LARRA, Pedro.** Tema 8. Redes Neuronales. . 2015. No. July.

**OTHMAN; et al.** An Embedded Real-Time Object Detection and Measurement of its Size. *2018 International Conference on Artificial Intelligence and Data Processing, IDAP 2018*. 2019. P. 1–4. DOI 10.1109/IDAP.2018.8620812.

**PARA, UNC and ACERO, GDEC DE,** 2011. Universidad De San Carlos De Guatemala Facultad De Ingeniería. *Emecanica.Ingenieria.Usac.Edu.Gt* [online]. 2011. P. 1–75. Retrieved from: <http://emecanica.ingenieria.usac.edu.gt/sitio/wp-content/subidas/6ARTÍCULO-III-INDESA-SIE.pdf>

**PÉREZ GRASSI, Ana PUENTE LEÓN, Fernanfo.** Vision artificial. *Universidad Técnica de Munich* [online]. 2009. Vol. 1, p. 2. Retrieved from: [http://www.aidima.es/vamad/news\\_Files/70\\_1.pdf](http://www.aidima.es/vamad/news_Files/70_1.pdf)

**PI, Raspberry,** 2015. Raspberry pi 3 model b. *online, https://www.raspberrypi.org*. 2015.

**PICAZO MONTOYA, Óscar,** 2018. Redes Neuronales Convolucionales Profundas para el reconocimiento de emociones en imágenes. [online]. 2018. P. 45. Retrieved from: [http://oa.upm.es/51441/%0Ahttp://oa.upm.es/51441/1/TFM\\_OSCAR\\_PICAZO\\_MONTOYA.pdf](http://oa.upm.es/51441/%0Ahttp://oa.upm.es/51441/1/TFM_OSCAR_PICAZO_MONTOYA.pdf)

**SINGH,; et al.** Design and Implementation of a Raspberry PI Surveillance Robot with Pan Tilt Raspbian Camera. . 2017. Vol. 11, no. 1, p. 69–73.

**SUPERIORES, Enseñanzas Artísticas and PRODUCTO, Especialidad De Diseño, 2016.**

Fotografía y Audiovisuales. . 2016. P. 1–9.

**XIA, Xiao-Ling; et al.** Facial Expression Recognition Based on TensorFlow Platform. *ITM Web of Conferences*. 2017. Vol. 12, p. 01005. DOI 10.1051/itmconf/20171201005.

**XING, Yang; et al.** Advances in Vision-Based Lane Detection: Algorithms, Integration, Assessment, and Perspectives on ACP-Based Parallel Vision. *IEEE/CAA Journal of Automatica Sinica*. 2018. Vol. 5, no. 3, p. 645–661. DOI 10.1109/JAS.2018.7511063.

**XU, Zhao;et al.** Canny edge detection based on Open CV. *ICEMI 2017 - Proceedings of IEEE 13th International Conference on Electronic Measurement and Instruments*. 2017. Vol. 2018-Janua, no. 1, p. 53–56. DOI 10.1109/ICEMI.2017.8265710.



Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
$V_{IL}$	Input low voltage <sup>a</sup>	VDD_IO = 3.3V	-	-	TBD	V
$V_{IH}$	Input high voltage <sup>a</sup>	VDD_IO = 3.3V	TBD	-	-	V
$I_{IL}$	Input leakage current	TA = +85°C	-	-	TBD	μA
$C_{IN}$	Input capacitance	-	-	TBD	-	pF
$V_{OL}$	Output low voltage <sup>b</sup>	VDD_IO = 3.3V, IOL = -2mA	-	-	TBD	V
$V_{OH}$	Output high voltage <sup>b</sup>	VDD_IO = 3.3V, IOH = 2mA	TBD	-	-	V
$I_{OL}$	Output low current <sup>c</sup>	VDD_IO = 3.3V, VO = 0.4V	TBD	-	-	mA
$I_{OH}$	Output high current <sup>c</sup>	VDD_IO = 3.3V, VO = 2.3V	TBD	-	-	mA
$R_{PU}$	Pullup resistor	-	TBD	-	TBD	kΩ
$R_{PD}$	Pulldown resistor	-	TBD	-	TBD	kΩ

<sup>a</sup> Hysteresis enabled

<sup>b</sup> Default drive strength (8mA)

<sup>c</sup> Maximum drive strength (16mA)

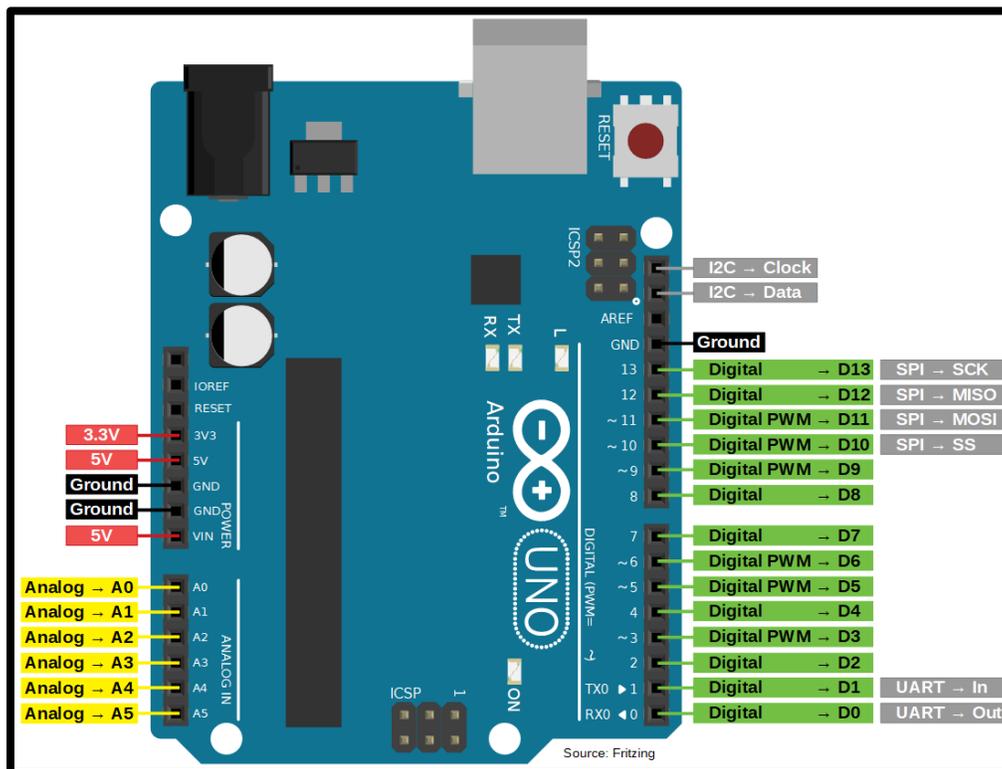
Table 3: DC Characteristics

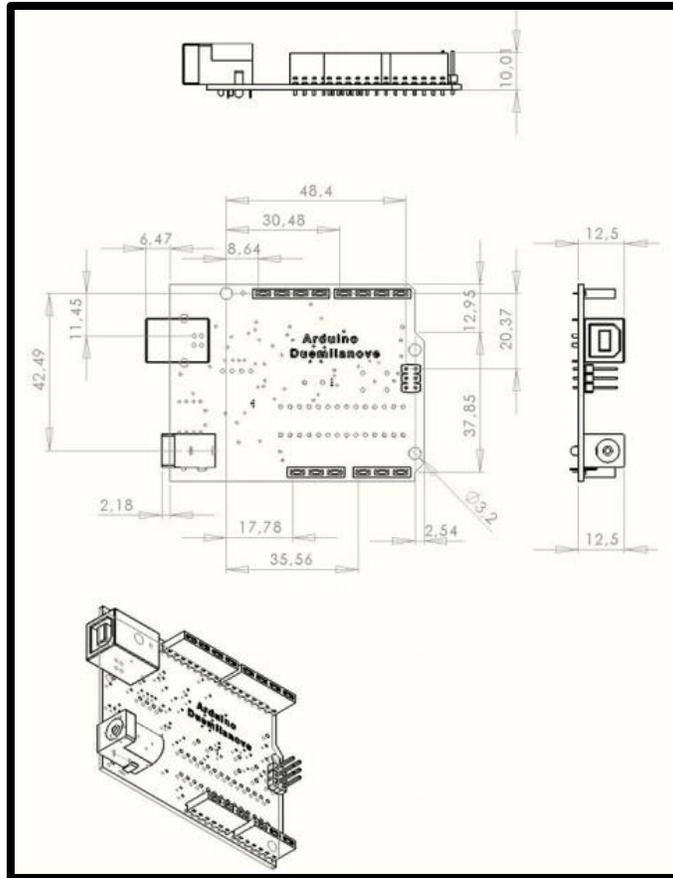
Pin Name	Symbol	Parameter	Minimum	Typical	Maximum	Unit
Digital outputs	$t_{rise}$	10-90% rise time <sup>a</sup>	-	TBD	-	ns
Digital outputs	$t_{fall}$	90-10% fall time <sup>a</sup>	-	TBD	-	ns

<sup>a</sup> Default drive strength, CL = 5pF, VDD\_IO = 3.3V

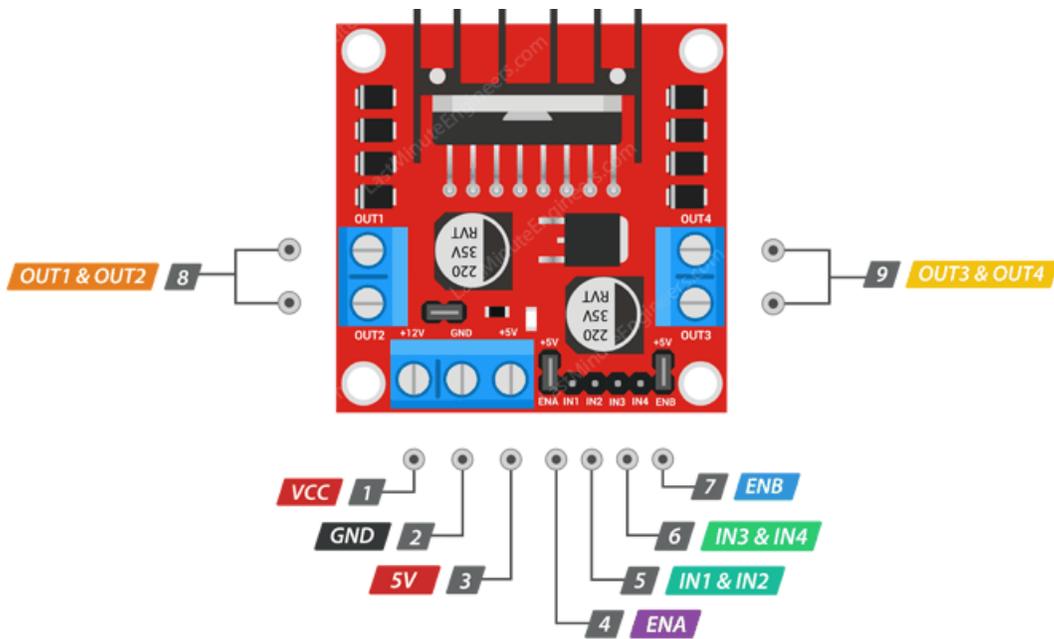
Table 4: Digital I/O Pin AC Characteristics

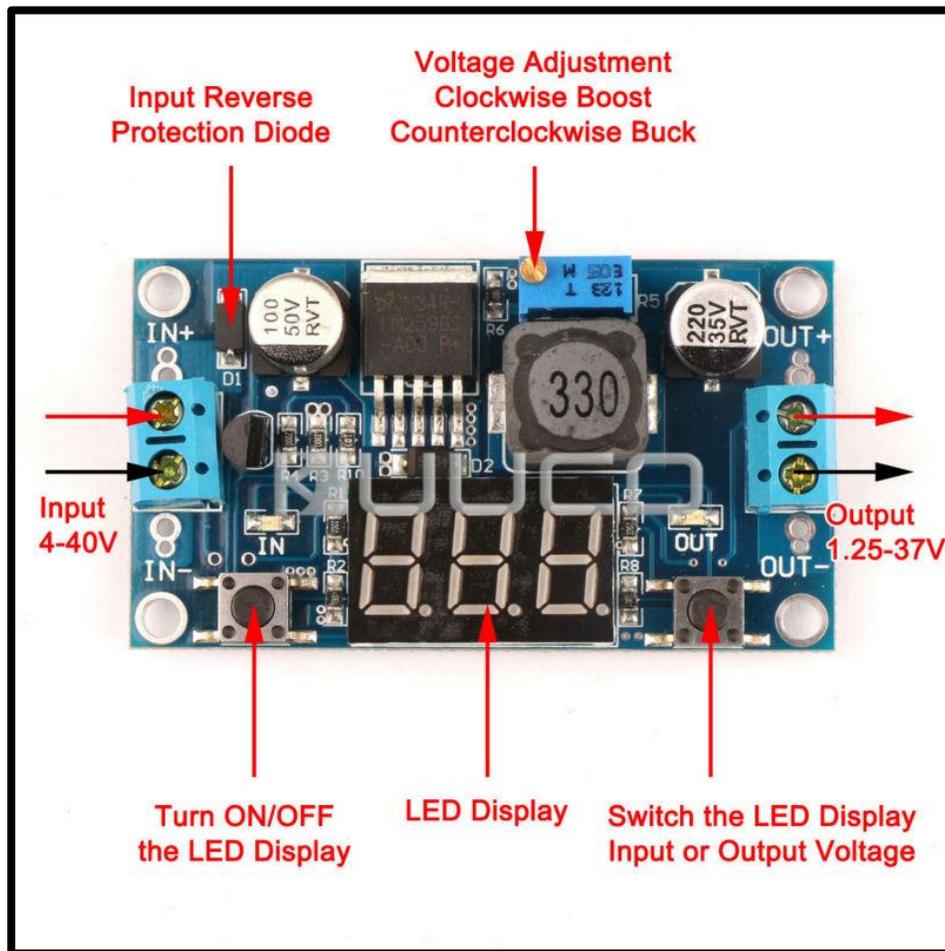
## Arduino UNO





L298N Module





DEVICE PARAMETERS					
$I_b$	Feedback Bias Current	Adjustable Version Only, $V_{FB}=1.3V$	10	50/100	nA nA (max)
$f_o$	Oscillator Frequency	(Note 6)	150	127/110 173/173	kHz (min) kHz (max)
$V_{SAT}$	Saturation Voltage	$I_{OUT}=3A$ (Notes 7, 8)	1.16	1.4/1.5	V V (max)
DC	Max Duty Cycle (ON) Min Duty Cycle (OFF)	(Note 8) (Note 9)	100 0		%
$I_{CL}$	Current Limit	Peak Current (Notes 7, 8)	4.5	3.6/3.4 6.9/7.5	A A (min) A (max)
$I_L$	Output Leakage Current	Output=0V (Notes 7, 9) Output=-0.9V (Note 10)	10	50 30	$\mu$ A (max) mA mA (max)
$I_Q$	Quiescent Current	(Note 9)	5	10	mA mA (max)
$I_{STBY}$	Standby Quiescent Current	ON/OFF pin=5V (OFF) (Note 10)	80	200/250	$\mu$ A $\mu$ A (max)
$\theta_{JC}$	Thermal Resistance	TO-220 or TO-263 Package, Junction to Case	2		$^{\circ}C/W$
$\theta_{JA}$		TO-220 Package, Junction to Ambient (Note 11)	50		$^{\circ}C/W$
$\theta_{JA}$		TO-263 Package, Junction to Ambient (Note 12)	50		$^{\circ}C/W$
$\theta_{JA}$		TO-263 Package, Junction to Ambient (Note 13)	30		$^{\circ}C/W$
$\theta_{JA}$		TO-263 Package, Junction to Ambient (Note 14)	20		$^{\circ}C/W$

## ANEXO B: CODIGO DE CONTROL DE VEHICULO

```
const int EnableL = 5;
const int HighL = 6;      // LEFT SIDE MOTOR
const int LowL =7;

const int EnableR = 10;
const int HighR = 8;     //RIGHT SIDE MOTOR
const int LowR =9;

const int D0 = 0;        //raspberry pin 21      LSB
const int D1 = 1;        //raspberry pin 22
const int D2 = 2;        //raspberry pin 23
const int D3 = 3;        //raspberry pin 24      MSB

int a,b,c,d,data;

void setup() {

pinMode(EnableL, OUTPUT);
pinMode(HighL, OUTPUT);
pinMode(LowL, OUTPUT);

pinMode(EnableR, OUTPUT);
pinMode(HighR, OUTPUT);
pinMode(LowR, OUTPUT);

}

void Data ()
{
    a = digitalRead(D0);
    b = digitalRead(D1);
    c = digitalRead(D2);
    d = digitalRead(D3);

    data = 8*d+4*c+2*b+a;
}
```

```
digitalWrite(HighL, LOW);
digitalWrite(LowL, HIGH);
analogWrite(EnableL, 255);

}

void Backward()
{
    digitalWrite(HighL, HIGH);
    digitalWrite(LowL, LOW);
    analogWrite(EnableL, 255);

}

void Stop()
{
    digitalWrite(HighL, LOW);
    digitalWrite(LowL, HIGH);
    analogWrite(EnableL, 0);

}

void Left1 ()
{
    digitalWrite(HighL, LOW);
    digitalWrite(LowL, HIGH);
    analogWrite(EnableL, 200);

    digitalWrite(HighR, LOW);
    digitalWrite(LowR, HIGH);
    analogWrite(EnableR, 255);

}
```

## ANEXO C: CODIGO DE PROCESAMIENTO DE IMAGENES

```
import pandas as pd
import numpy as np
import os
from matplotlib import pyplot as plt
from sklearn.utils import shuffle
import matplotlib.image as mpimg
from imgaug import augmenters as iaa
import cv2
import random

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D, Flatten, Dense
from tensorflow.keras.optimizers import Adam

def getName(filePath):
    return filePath.split('\\')[-1]

def importDataInfo(path):
    columns = ['center', 'Left', 'Right', 'Steering', 'Throttle', 'Brake', 'Speed']
    data = pd.read_csv(os.path.join(path, 'driving_log.csv'), names=columns)
    #print(data.head())
    #print(data['center'][0])
    #print(getName(data['center'][0]))
    data['center'] = data['center'].apply(getName)
    #print(data.head())
    print('Total Images Imported', data.shape[0])
    return data

def balanceData(data, display=True):
    nBins = 31
    samplesPerBin = 1000
    hist, bins = np.histogram(data['Steering'], nBins)
    #print(bins)
```

```
#print(bins)
if display:
    center = (bins[:-1] + bins[1:])*0.5
    #print(center)
    plt.bar(center, hist, width=0.06)
    plt.plot((-1, 1), (samplesPerBin, samplesPerBin))
    plt.show()

removeIndexList = []
for j in range(nBins):
    binDataList = []
    for i in range(len(data['Steering'])):
        if data['Steering'][i] >= bins[j] and data['Steering'][i] <= bins[j+1]:
            binDataList.append(i)
    binDataList = shuffle(binDataList)
    binDataList = binDataList[samplesPerBin:]
    removeIndexList.extend(binDataList)
print('Removed Images : ', len(removeIndexList))
data.drop(data.index[removeIndexList], inplace=True)
print('Remaining Images : ', len(data))

if display:
    hist, _ = np.histogram(data['Steering'], nBins)
    plt.bar(center, hist, width=0.06)
    plt.plot((-1, 1), (samplesPerBin, samplesPerBin))
    plt.show()

return data

def loadData(path, data):
    imagesPath = []
    steering = []

    for i in range(len(data)):
```

```

def augmentImage(imgPath, steering):
    img = mpimg.imread(imgPath)

    ##pan
    if np.random.rand() < 0.5:
        pan = iaa.Affine(translate_percent={'x':(-0.1,0.1), 'y':(-0.1,0.1)})
        img = pan.augment_image(img)

    ##zoom
    if np.random.rand() < 0.5:
        zoom = iaa.Affine(scale=(1,1.2))
        img = zoom.augment_image(img)

    ##brightness
    if np.random.rand() < 0.5:
        brightness = iaa.Multiply((0.4,1.2))
        img = brightness.augment_image(img)

    ##flip
    if np.random.rand() < 0.5:
        img = cv2.flip(img,1)
        steering = -steering
    return img, steering

#imgRe, st = augmentImage('test.jpg', 0)
#plt.imshow(imgRe)
#plt.show()

def preProcessing(img):
    img = img[60:135, :, :]
    img = cv2.cvtColor(img, cv2.COLOR_RGB2YUV)
    img = cv2.GaussianBlur(img, (3,3), 0)
    img = cv2.resize(img, (200, 66))
    img = img/255
    return img

```

## ANEXO D: CODIGO RED NEURONAL NVIDIA

```

def creatModel():
    model = Sequential()

    model.add(Convolution2D(24, (5, 5), (2, 2), input_shape=(66, 200, 3), activation='elu'))
    model.add(Convolution2D(36, (5, 5), (2, 2), activation='elu'))
    model.add(Convolution2D(48, (5, 5), (2, 2), activation='elu'))
    model.add(Convolution2D(64, (3, 3), activation='elu'))
    model.add(Convolution2D(64, (3, 3), activation='elu'))

    model.add(Flatten())
    model.add(Dense(100, activation='elu'))
    model.add(Dense(50, activation='elu'))
    model.add(Dense(10, activation='elu'))
    model.add(Dense(1))

    model.compile(Adam(learning_rate=0.0001), loss='mse')

    return model

```

## ANEXO E: CODIGO DE MAESTRO RASPBERRY PI 4

```
using namespace std;
using namespace cv;
using namespace raspicam;

Mat frame, Matrix, framePers, frameGray, frameThresh, frameEdge, frameFinal, frameFinalDuplicate, frameFinalDuplicate1;
Mat ROIlane, ROIlaneEnd;
int LeftLanePos, RightLanePos, frameCenter, laneCenter, Result, laneEnd;

RaspICam_Cv Camera;

stringstream ss;

|
vector<int> histogramLane;
vector<int> histogramLaneEnd;

Point2f Source[] = {Point2f(40,135),Point2f(360,135),Point2f(0,185), Point2f(400,185)};
Point2f Destination[] = {Point2f(100,0),Point2f(280,0),Point2f(100,240), Point2f(280,240)};

void Setup ( int argc, char **argv, RaspICam_Cv &Camera )
{
    Camera.set ( CAP_PROP_FRAME_WIDTH, ( "-w",argc,argv,400 ) );
    Camera.set ( CAP_PROP_FRAME_HEIGHT, ( "-h",argc,argv,240 ) );
    Camera.set ( CAP_PROP_BRIGHTNESS, ( "-bn",argc,argv,50 ) );
    Camera.set ( CAP_PROP_CONTRAST, ( "-cb",argc,argv,50 ) );
    Camera.set ( CAP_PROP_SATURATION, ( "-sa",argc,argv,50 ) );
    Camera.set ( CAP_PROP_GAIN, ( "-g",argc,argv,50 ) );
    Camera.set ( CAP_PROP_FPS, ( "-fps",argc,argv,0) );
}

void Capture()
{
```

```
int main(int argc, char **argv)
{
    wiringPiSetup();
    pinMode(21, OUTPUT);
    pinMode(22, OUTPUT);
    pinMode(23, OUTPUT);
    pinMode(24, OUTPUT);

    Setup(argc, argv, Camera);
    cout<<"Connecting to camera"<<endl;
    if (!Camera.open())
    {
        cout<<"Failed to Connect"<<endl;
    }

    cout<<"Camera Id = "<<Camera.getId()<<endl;

    while(1)
    {
        auto start = std::chrono::system_clock::now();

        Capture();
        Perspective();
        Threshold();
        Histogram();
        LaneFinder();
        LaneCenter();

        if (LaneEnd > 3000)
```

```

        digitalWrite(21, 1);
digitalWrite(22, 1); //decimal = 7
digitalWrite(23, 1);
digitalWrite(24, 0);
cout<<"Lane End"<<endl;
}

if (Result == 0)
{
digitalWrite(21, 0);
digitalWrite(22, 0); //decimal = 0
digitalWrite(23, 0);
digitalWrite(24, 0);
cout<<"Forward"<<endl;
}

else if (Result >0 && Result <10)
{
digitalWrite(21, 1);
digitalWrite(22, 0); //decimal = 1
digitalWrite(23, 0);
digitalWrite(24, 0);
cout<<"Right1"<<endl;
}

else if (Result >=10 && Result <20)
{
digitalWrite(21, 0);
digitalWrite(22, 1); //decimal = 2
digitalWrite(23, 0);
digitalWrite(24, 0);
cout<<"Right2"<<endl;
}

```

```

if (laneEnd > 3000)
{
ss.str(" ");
ss.clear();
ss<<" Lane End";
putText(frame, ss.str(), Point2f(1,50), 0,1, Scalar(255,0,0), 2);
}

else if (Result == 0)
{
ss.str(" ");
ss.clear();
ss<<"Result = "<<Result<<" Move Forward";
putText(frame, ss.str(), Point2f(1,50), 0,1, Scalar(0,0,255), 2);
}

else if (Result > 0)
{
ss.str(" ");
ss.clear();
ss<<"Result = "<<Result<<" Move Right";
putText(frame, ss.str(), Point2f(1,50), 0,1, Scalar(0,0,255), 2);
}

else if (Result < 0)
{
ss.str(" ");
ss.clear();
ss<<"Result = "<<Result<<" Move Left";
putText(frame, ss.str(), Point2f(1,50), 0,1, Scalar(0,0,255), 2);
}

```